GENERATING ALL-ATOM PROTEIN STRUCTURE FROM SEQUENCE-ONLY TRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Using generative models for protein design is gaining interest for their potential scientific impact. However, biological processes are mediated by many modalities, and simultaneous generating multiple biological modalities is a continued challenge. We propose **PLAID (Protein Latent Induced Diffusion)**, whereby multimodal biological generation is achieved by learning and sampling from the *latent space of a predictor* from a more abundant data modality (e.g. sequence) to a less abundant data modality (e.g. crystallized structure). Specifically, we examine the *all-atom* structure generation setting, which requires producing both the 3D structure and 1D sequence, to specify how to place sidechain atoms that are critcial to function. Crucially, since **only sequence inputs are required to obtain the latent representation during training**, we can use sequence-only databases, thus augmenting the sampleable data distribution by $10^2\times$ to $10^4\times$ compared to experimental structure databases. Using sequence-only training further also unlocks more annotations that can be used to control and condition the model. As a demonstration, we use two conditioning variables: 2219 function keywords from Gene Ontology, and 3617 organisms across the tree of life. Despite not receiving structure inputs during training, model generations nonetheless exhibit strong performance on structure quality, diversity, novelty, and cross-modal consistency metrics. Analysis of function-conditioned samples show that generated structures preserve non-adjacent catalytic residues at active sites, and learn the hydrophobicity pattern of transmembrane proteins, while exhibiting overall sequence diversity. Model weights and code are publicly accessible at `[redacted]`.

1 INTRODUCTION

Generative protein models propose designs and can accelerate innovation in bioengineering. Many protein functions are mediated by their structure, including the identity, placement, and biophysical properties of both sidechain and backbone atoms, known as the *all-atom structure*. However, to know which sidechain atoms to place, one must first know the *sequence*; all-atom structure generation thus can be seen as a multimodal problem that requires simultaneous generation of sequence and structure.

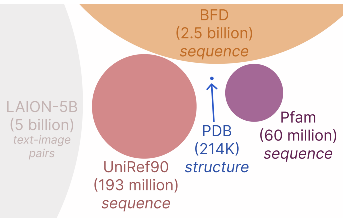


Figure 1: Compared to structural databases, protein sequence databases offer better distribution coverage, and can approach sizes of internet-scale datasets.

While generative modeling for protein structures has seen rapid recent progress, important challenges remain: **(1)** Existing protein structure and sequence generation methods often treat sequence and structure as *separate modalities*; structure-generation methods often only provide backbone atoms. **(2)** Methods that do address all-atom design often require alternating between folding and inverse-folding steps using an extraneous prediction model. **(3)** Evaluations often emphasize *in silico* oracle-based designability, or structure-conditioning, with limited progress towards

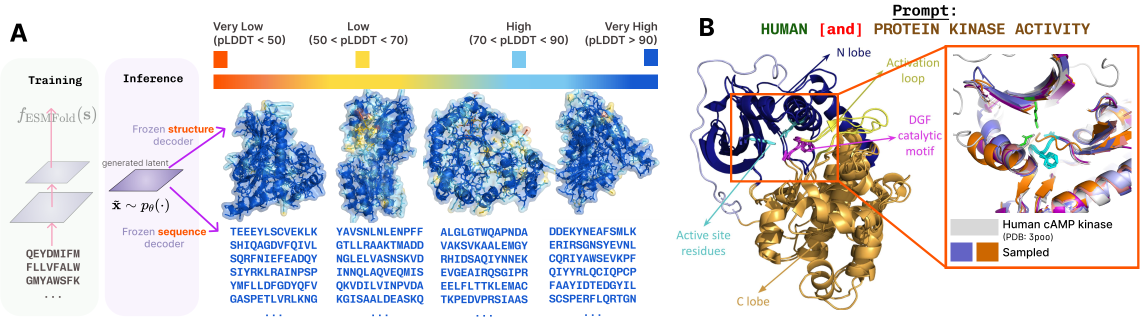


Figure 2: **(A)** Using the PLAID paradigm to sample from the latent space of ESMFold unconditionally generates high quality all-atom structure and sequence **despite using only sequence input** to train the generative model. **(B)** Since sequence-only databases has more annotations, we can compositionally **condition by function and expression organism**. Function-conditioned proteins can **preserve known catalytic residues** (example shown for `HUMAN` and `PROTEIN KINASE ACTIVITY`). An example is shown for generating human kinases; generations preserves the known DFG catalytic motif, despite these residues being non-adjacent in sequence space. The global N-terminal and C-terminal lobes characteristic of human MAP kinases [1] is also preserved, despite sharing only 48% global sequence identity to the generation. Generated samples are classified as being in active kinase conformation by the Kincore predictor [2].

other forms of flexible controllability. **(4)** Methods that rely on experimentally-resolved structure databases have a strong bias towards crystallizable proteins. **(5)** Methods sometimes ignore scalability and flexibility; models that ingest structure as inputs have more restrictions on architecture, and is harder to leveraging progress in hardware-aware mechanisms for more scalable large language models.

Contributions Towards resolving these challenges, we introduce **PLAID (Protein Latent Induced Diffusion)**. Our principal demonstration is that multimodal generation in biology can be achieved by learning the latent space of a predictor from a more abundant data modality (e.g. sequence) to a less abundant data modality (e.g. crystallized structure). In particular, we introduce a controllable diffusion model capable of **sequence and all-atom structure generation, while requiring only sequence inputs during training**. Because training dataset can be defined by sequence databases rather than structural ones, this provides better coverage of the viable protein space traversed by evolution. It furthermore allows us to leverage structural information encoded in the *pretrained weights* rather than training data. Finally, it increases the availability of labels and natural language annotations for controllable generation. As a motivating demonstration, we examine compositional control across the axes of *function* and *organism*. Though we focus on ESMFold [3] and all-atom structure generation in this work, the method is designed to scale readily to expanding sequence datasets, improved infrastructure for Transformer-based models, and capitalize on the ever-expanding capabilities of structure-prediction models to include more modalities, such as nucleic acids and molecular ligand binding [4, 5].

2 RELATED WORKS

Latent space diffusion models Diffusing in the latent space of pixel representations has been successful in generating high-fidelity and resolution image samples [10, 11], as it can reduce compute constraints, improve sampling speed, and improve quality. In images, the encoding to latent space can be seen as a “perceptual

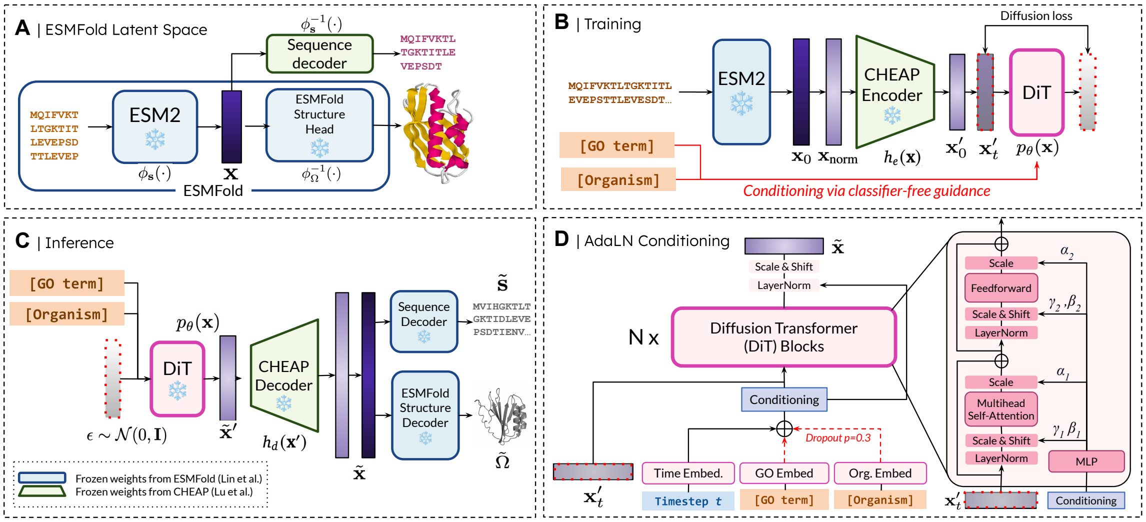


Figure 3: Overview of PLAID. **(A) ESMFold [3] latent space**. The latent space $p(\mathbf{x})$ can be considered a joint embedding of sequence and structure.**(B) Latent diffusion training.** Our goal is to learn and sample from $p_\theta(\mathbf{x})$, following the diffusion [6] formulation. To improve learning efficiency, the embedding $\mathbf{x}$ is compressed using the CHEAP [7] autoencoder h_e. We then iteratively noise and denoise from $p_\theta(h_e(\mathbf{x}))$, **(C) Inference.** To obtain both sequence and structure at inference time, we can sample and uncompress to obtain $\tilde{\mathbf{x}} = h_d(\mathbf{x}')$ where $\tilde{\mathbf{x}}' \sim p_\theta(\tilde{\mathbf{x}})$, and use the frozen structure structure decoder (trained in ESMFold [3]) and the frozen sequence decoder (trained in CHEAP [7]) to obtain the all-atom structure. **(D) DiT block architecture.** We use the Diffusion Transformer (DiT) [8] architecture, which uses AdaLN blocks to incorporate conditioning information. Classifier-free guidance is used to incorporate the function (i.e. GO term) and organism class label embeddings; with $p = 0.3$, the token is replaced by a $\varnothing$ token denoting the unconditional condition [9].

compression" stage where high-frequency and unimportant details are moved. Latent space diffusion can be used with gradient-based control [12] or multimodal conditioning using a CLIP [13]-like biencoder [14, 15], but is also compatible with classifier-free guidance [9].

Generative Modeling for Proteins State-of-the-art diffusion models for designing protein structure have thus far focused on generating *novel backbone folds*, with conditioning controllability typically governed by secondary structure, or for generating scaffolding for a known motif [16, 17, 18, 19]. Evaluation and design of these models focus on fold stability and novelty, and often involve using oracle models [20, 3, 21, 22] for folding or inverse folding. However, to synthesize the protein, the sequence is required, and not all sampled structures might have a corresponding sequence. To address this, “designability” has been posited as a metric, which assesses the correspondence between the original structure and the sequence predicted for that structure. However, there are few mechanisms to enforce designability during training. Methods also exist for designing sequence [23, 24, 25, 26], sometimes conditioned by the structure [27]. Structure can be constructed from these generations using a protein folding model, but models do not explicitly produce atomic positions.

Multimodal Sequence-Structure and All-Atom Generation All-atom generation can thus be viewed as a multimodal generation problem, where the 1D protein sequence and 3D protein structure are jointly produced. Existing works [28, 29] often generate only one of structure or sequence at each diffusion step, and rely on an external predictor to produce the other modality. Multiflow [30] performs co-generation without an external tool, but does not produce side chain positions. Some works have focused on specific protein subclasses, such as antibody design [31, 32]. While these models achieve success within their specialized domains, antibodies represent a narrow subset of protein space and such models often struggle with out-of-distribution generalization when extended to the broader protein universe. Concurrently developed with this work, ESM3 [33] also uses generates in the shared sequence-structure space, and is conditioned on Interpro (many of which are derived from GO terms) for controllability. However, the ESM3 tokenizer is trained on structure datasets, rather than sequence databases, and cannot perform all-atom generation.

3 PLAID: PROTEIN LATENT INDUCED DIFFUSION

Notation A protein is composed of component amino acids. A protein sequence $\mathbf{s} := \{r_i\}_{i=1}^{L}$ is often shown as a string of characters, with each character denoting the identity of an amino acid residue $r \in \mathcal{R}$, with $|\mathcal{R}| = 20$. Each unique residue r can be mapped to a set of atoms as $\mathbf{r} := \{\mathbf{a}_i\}_i^M$, where $\mathbf{a} \in \mathbb{R}^3$ is the 3D coordinates of the atom, and the number of atoms M in each residue $\mathbf{r}$ may be different depending on the identity. A protein structure $\Omega := \{\mathbf{r}_i\}_{j=1}^{L}$ consists of all atoms in the protein.[1]

From above definitions, we see that the *all-atom structure* Ω requires knowledge of the amino acid identities at each position in order to specify the side chain atoms. To reduce complexity, protein structure designers sometimes work with the backbone atoms $\Omega_{CC} \subset \Omega$ only, which only include the N, C, C_α atoms only, and are generally sufficient to define the protein fold.[2]

3.1 DEFINING p(SEQUENCE,STRUCTURE)

We begin with the motivation that sampling directly from $p(\mathbf{s}, \Omega)$ without implicitly factorizing it into $p(\Omega)p(\mathbf{s}|\Omega)$ (e.g., Protpardelle [28]) or $p(\mathbf{s})p(\Omega|\mathbf{s})$ (e.g., ProteinGenerator [29]) circumvents the difficulty in all-atom generation of not knowing which side chain atoms to place; one can choose a latent manifold where residues do not need to be explicit specified during iterative generation. Avoiding reliance on external prediction tools is computationally cheaper, and avoids amplifying errors.

Our goal is to characterize a distribution $p(\mathbf{x})$ over $\mathcal{X}$ that encapsulates both sequence and structure information, such that there is a mapping $\mathbf{x} = \phi_{\mathbf{s},\Omega}(\mathbf{s}, \Omega)$. To do this, we follow the definition of joint embedding of sequence and structure in Lu et al. [7]: if we decompose $\mathbf{x} = \phi_{\mathbf{s},\Omega}(\mathbf{s}, \Omega) = \phi_{\mathbf{s}}(\mathbf{s}) \circ \phi_{\Omega}(\Omega)$, we can look for a space where some deterministic mapping will map sequence $\mathbf{s}$ and its corresponding structure Ω to the same latent embedding $\mathbf{x} \in \mathcal{X}$. One way to do so is by defining $\mathbf{x}$ as the latent space of a protein folding model $p(\Omega|\mathbf{s})$. The trunk of the model provides $\mathbf{x} = \phi_{\text{ESM}}(\mathbf{s})$, and the structure head provides $\Omega = \phi_{\text{Structure Module}}(\mathbf{x})$. If we consider there to be an implicit inverse function of the Structure Module such that $\mathbf{x} = \phi^{-1}_{\text{Structure Module}}(\Omega)$, then this provides the mappings for $\mathbf{x} = \phi_{\mathbf{s},\Omega}(\mathbf{s}, \Omega) = \phi_{\mathbf{s}}(\mathbf{s}) \circ \phi_{\Omega}(\Omega)$ that we are looking for.

3.2 OVERVIEW OF ESMFOLD

Briefly, ESMFold [3] has two main components: a protein language model component $\mathbf{x} = \phi_{\text{ESM2}}(\mathbf{s})$ that captures evolutionary priors via the masked language modeling loss (MLM), and a structure module

---

[1] In practice, to make use of array broadcasting, a standard M is selected for all residues, with an associated one-hot mask to specify which atoms are present for a given residue, and we treat each structure as a matrix $\Omega \in \mathbb{R}^{L \times M \times 3}$. Following prior work [34, 3], we use the `atom14` representation where $M = 14$.

[2] The three torsion angles in backbone-only structures induces 3^L degrees of freedom; depending on the residue identity, there may be 0 to 4 additional rotamer angles associated with the sidechains. Therefore, even when the sequence is known, there may up to 4^L additional degrees of freedom necessary for all-atom structure prediction.

component $\Omega = \phi_{\text{Structure Module}}(\mathbf{x})$ that decodes these latent embeddings into a 3D structure. For the rest of this work, "latent space of ESMFold" refers to the $\mathbf{x} \in \mathbb{R}^{L\times 1024}$ representation at the layer just prior to the Structure Module, where L is the length of a given protein. We choose this layer due to the observations in Lu et al. [7] (also see Section 3.3) that the pairwise input at inference time to the Structure Module is initialized to zeros, such that this sequence embedding contains all information for structure prediction (Figure 3A and Appendix B).

3.3 SAMPLING ALL-ATOM STRUCTURE

Latent Generation Our goal is to learn $p_\theta(\mathbf{x}) \approx p(\mathbf{x})$, where θ are parameters of the model learned through diffusion training (Figure 3C). Then, after training, we can sample $\tilde{\mathbf{x}} \sim p_\theta(\mathbf{x})$ (Figure 3C). To do so, we use diffusion models [6, 35], with some modifications (described in ablation Table **??**). To obtain structure from the sampled latent embedding, we can simply use frozen ESMFold structure module weights to obtain $\tilde{\Omega} = \phi_{\text{Structure Module}}(\tilde{\mathbf{x}})$ (Figure 3C). Since the output of $\phi_{\text{Structure Module}}$ is all-atom, the sampled $\tilde{\Omega}$ is also all-atom.

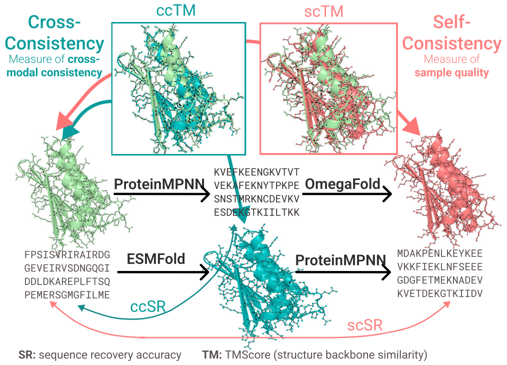


Figure 4: Schematic describing the *cross-consistency* metric we use for assessing multimodal generation consistency, and the *self-consistency* metric we use for assessing the quality of uni-modal generated structures and sequences, independent of the generation quality of other modalities.

Sequence Decoder To obtain the sequence, we need an "inverse mapping of ESM2" to get $\tilde{\mathbf{s}} = \phi^{-1}_{\text{ESM}}(\tilde{\mathbf{x}})$. This inverse mapping is straightforward to train, since $\mathbf{x}$ is a linearly projected version of the ESM2 embedding, which was trained via the MLM loss. This sequence decoder ϕ^{-1}_{ESM} is also trained and provided in Lu et al. [7], with validation accuracy on a heldout partition of UniRef [36] reaching 99.7% [7]. Note that $\tilde{\mathbf{s}}$ must be decoded first, which determines the side-chain atoms to be placed in $\tilde{\Omega}$.

Latent Space Compression In initial experiments, we found that directly learning $p(\mathbf{s})$ performed poorly (results shown in Appendix **??**). We suspected that this might be due to the dimensions of $\mathbf{x} \in \mathbb{R}^{L\times\ 1024}$. For proteins with length $L \times 512$, this maps to a high-resolution synthesis problem in image diffusion literature. We therefore adopt a similar technique as in high-resolution image synthesis, where diffusion is performed in the latent space of an autoencoder $\mathbf{s}' = h_e(\mathbf{x})$ such that the array dimensions of $\mathbf{x}'$ is much smaller [11]. We use the CHEAP autoencoder [7], such that diffusion training becomes $p_\theta(\mathbf{x}') = h_e(\mathbf{x})$. Noise is added and denoised from $p(\mathbf{x}')$. At inference time, we first sample the compressed latent $\tilde{x}' \sim p_\theta(\mathbf{x}')$, then "uncompress" it to $\tilde{\mathbf{x}} = h_d(\mathbf{x}')$, followed by using frozen decoders to obtain $\tilde{\mathbf{s}} = \phi^{-1}_{\text{ESM}}(\tilde{\mathbf{x}})$ and $\tilde{\Omega} = \phi_{\text{Structure Module}}(\tilde{\mathbf{x}})$. More information on CHEAP can be found in Lu et al. [7] and Appendix B.

Figure 10A offers clues to why our initial experiments without compression was difficult; prior to the normalization and compression steps in CHEAP, noise added in the latent space does not affect sequence and structure until the final timesteps in forward diffusion, despite using a cosine schedule (SNR and log-SNR curves shown below), meaning that the denoising task would be trivial for most sampled timesteps.

3.4 DATA AND TRAINING

Choice of Sequence Database The general paradigm in PLAID can be used on any sequence database. As of 2024, sequence-only database sizes can range from UniRef90 [36] (193 million sequences) to metagenomic datasets such as BFD [37] (2.5 billion sequences) and OMG [38] (3.3 billion sequences). We use Pfam because it provides more annotations for *in silico* evaluation, and because protein domains are the main units of structure-mediated functions. More information can be found in Appendix C.

Compositional Conditioning by Function and Organism Gene Ontology (GO) is a structured hierarchical vocabulary for annotating gene functions, biological processes, and cellular components across species [39, 40]. We examine all Pfam domains for which there exists a Gene Ontology mapping; there are 2219 GO terms compatible with our model (an abbreviated list is listed in Appendix **??**). We also examine all unique organisms in our dataset, and find 3617 organisms. Models are trained with classifier-free guidance [9]. The conditioning architecture is described in Figure 3D. More details can be found in Appendix A.

Architecture We use a Diffusion Transformer [8] (DiT) for the denoising task. This enables more flexible options for finetuning on mixed input modalities, as protein structure prediction models begin expanding to complexes with nucleic acids and small molecular ligands. It also makes better use of Transformer training infrastructure [41, 42, 43, 44, 45]. In early experiments, we found that proportioning available memory to a larger DiT model was more helpful than using triangular self-attention [20]. We train our models using the xFormers [41] implementation of [46], which provided a 55.8% speedup with a 15.6% reduction in GPU memory usage in our inference-time benchmarking experiments compared to a vanilla implementation using PyTorch primitives (Appendix G). We train two versions of the model with 100 million and 2 billion parameters respectively, both for 800K steps. More details are in Appendix A.

Diffusion Training and Inference-Time Sampling We use the discrete-time diffusion definition proposed in Ho et al. [6], using 1000 timesteps. Additional strategies are used to stabilize training and improve performance: min-SNR reweighting [47], v-diffusion [48, 49], self-conditioning [50, 51], and a Sigmoid noise schedule [52], and EMA (exponential moving average) decay. Ablation results are shown in Table **??**. For sampling, unless otherwise noted, all results use the DDIM sampler [12, 35] with 500 timesteps. We use $c = 3$ as the conditioning strength for conditional generation; however, we find (Figure 10C) that sample quality is not strongly affected by this hyperparameter. We also find that that DPM-Solvers [53] can attain comparable results with $10\times$ fewer steps in cases where speed is of concern (Appendix Figure 12), but here prioritize sample quality. More details are in Appendix D.

Table 1: Ablation results (see Section 3.4). Metrics are defined in Section 4.

	Configuration	ccTM	scTM	Ppl.	Seq. Div. %	Struct. Div.%
A	cosine noise sched.& pred. noise	0.54	0.55	16.97	**0.98**	0.86
B	A + v-diffusion	0.52	0.53	17.37	**0.98**	**0.89**
C	A + MinSNR	0.59	0.59	16.76	0.97	0.86
D	A +B + C + sigmoid noise sched.	0.56	0.58	16.88	0.92	0.86
E	D + self-conditioning	**0.70**	**0.65**	**15.38**	0.93	0.76
F	E + no cond drop	0.57	0.57	17.28	0.97	0.85

4 EVALUATION

Following previous works and to address the unique challenges of all-atom generation, we examine the following metrics. *More details on how metrics are calculated can be found in Appendix E.*

1. **Multimodal Cross-Consistency:** Do the simultaneously generated structure and sequence accord with each other? When the generated sequence is refolded using Omegafold [56], does it match the generated structure? *[Cross-consistency TM-Score **(ccTM)**, cross-consistency RMSD **(ccRMSD)**.]* When the generated structure is inverse-folded into a sequence using ProteinMPNN [21], does it match the generated sequence? *[Cross-consistency sequence recovery **(ccSR)**.]* What percentage of generated samples are designable? *[**ccRMSD < 2Å**.]*
2. **Uni-modal Sample Quality**: When structure and sequence are separately considered, do samples exhibit high quality?

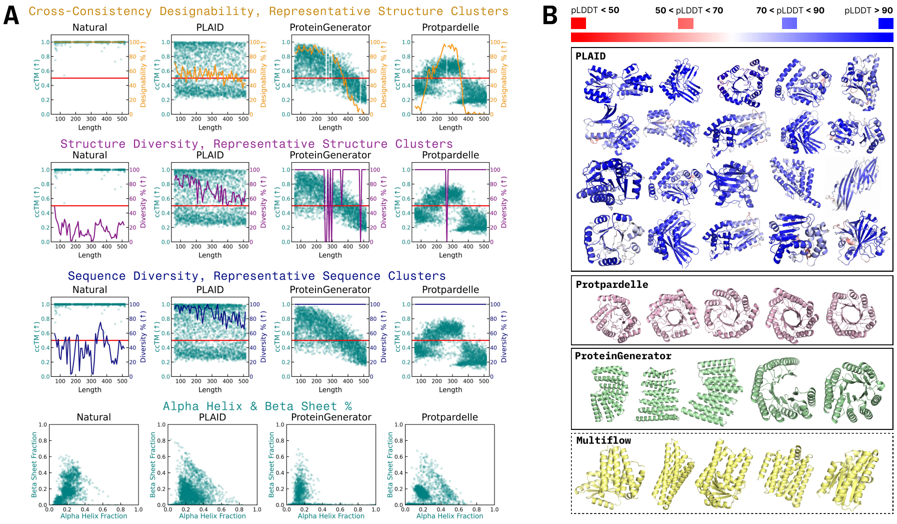


Figure 5: **By-length analysis of quality, sample diversity, and secondary structures**. Additional Figures can be found in Appendix Figure 9. **(A)** For each protein between lengths {64, 72, 80, ..., 508, 512}, we co-generate 64 proteins, and cluster the generated sequences using MMseqs2 [54] and Foldseek [55]. We then plot each representative cluster. The red line is the TM-Score$= 0.5$ threshold that is used in prior work to refer to designability [28, 16]. At each protein length, we plot: **(1)** The fraction of designable samples that have ccTM > 0.5; **(2)** The ratio of unique structure clusters to samples, as a measure of structural diversity; **(3)** The ratio of unique sequence clusters to samples, as a measure of sequence diversity; and **(4)** The beta sheet and alpha helix percentage of generations, follow prior work that demonstrate that protein generative models often produce more alpha helices than beta sheets. At higher sequence lengths, PLAID can produce higher quality samples, whereas baseline methods often struggle, and/or exhibit mode collapse. **(B)** Unconditional generation results on proteins with length 256 using PLAID. Protpardelle [28] and ProteinGenerator [29] suffer mode collapse at this length towards TIM barrels and alpha helix bundles.

   (a) ***Structure.*** Do the inverse-folded sequences of a given structure fold back into itself? *[Self-consistency TM-Score **(scTM)**, self-consistency RMSD **(scRMSD)**.]*
   (b) ***Sequence.*** Do the inverse-folded results from the predicted structure of a generated sequence match the original? *[Self-consistency sequence recovery **(scSR)**.]* Do generated sequences have low perplexity on next-token prediction models trained on natural proteins? *[Perplexity **(Ppl.)** under RITA XL [26].]*

3. **Naturalness**: Do samples exhibit sensible biophysical parameters for real-world characterization? What is the Wasserstein Distance between `ProtParam` properties provided by the Biopython [57] package? In other words, how similar are the distributions of biophysical properties between generated proteins and real proteins? *[**Distributional conformity** [23] scores.]*
4. **Diversity**: Are the designable proposals by the model actually diverse in sequence and structural space? At the default clustering threshold for popular bioinformatics tools [54, 55], how many distinct clusters do we observe? *[**# Des. seq. clusts., # Des. struct. clusts.**.]*
5. **Novelty**: Do generated structures differ from those found in nature? How similar is the structure to its closest structural match? *[**Foldseek TMScore**.]* How similar is the generated sequence to its closest sequence match? *[**MMseqs seq id. %**.]*

5 EXPERIMENTS

Table 2: Comparison of model performance across **consistency and quality metrics**. Arrows indicate whether higher (↑) or lower (↓) values are better. Bold values show best performance among all-atom generation models. pLDDT refers to the confidence score directly returned by the structure trunk of the generative model; for models which do not return a PLDDT metric, N/A is used. Heavy asterisk (*****) indicates Multiflow, which generates backbone structure and residue identities without sidechain positions. Italic values represent natural/reference measurements.

	Cross-Modal Consistency				Structure Quality		Sequence Quality	
Model	ccTM (↑)	ccRMSD (↓)	ccSR (↑)	ccRMSD < 2Å(↑)	scTM (↑)	pLDDT (↑)	scSR (↑)	Ppl. (↓)
PG	0.58	11.86	**0.28**	8.00%	**0.72**	**69.00**	0.40	**8.60**
Protpardelle	0.44	24.28	0.22	0.00%	0.57	N/A	**0.44**	8.86
PLAID	**0.69**	**9.47**	0.26	**32.00%**	0.64	59.46	0.27	14.61
Multiflow*	0.92*	2.45*	0.52*	78%*	0.91*	N/A	0.61*	8.1*
Natural	*1.00*	*0.07*	*0.39*	*100.00%*	*0.84*	*84.51*	*0.39*	*7.40*

Table 3: **Diversity, novelty, and distributional conformity [23]** metrics across models. Metrics are described in Section 4 As with Table 2, asterisk (*) indicates methods which generate backbone structure and sequence without sidechain positions, bold indicates best performance across all-atom generation methods, and italic indicates performance on a reference set of natural sequences.

	Diversity			Novelty			Distributional Conformity				
	# Des. (↑)	# Des. Seq. Clusts. (↑)	# Des. Struct. Clusts. (↑)	MMseqs Seq Id % (↓)	Foldseek TMScore (↓)	Avg. MW. (↓)	Aromaticity (↓)	Instability Index (↓)	Isoelectric Point (↓)	GRAVY (↓)	Charge pH=7 (↓)
PG	309	309	309	0.57	**0.57**	9.54	0.07	14.55	1.42	0.31	6.12
Protpardelle	0	0	0	0.56	0.72	10.4	0.07	8.61	1.99	0.37	8.58
PLAID	**1171**	**809**	**522**	0.60	0.67	**0.62**	**0.01**	**1.98**	**0.49**	**0.28**	**2.71**
Multiflow*	2812*	2452*	460*	**0.45***	0.68*	5.43*	0.07*	4.11*	1.59*	0.3*	7.55*
Natural	*3570*	*1362*	*600*	*0.81*	*0.87*	0	0	0	0	0	0

5.1 UNCONDITIONAL GENERATION

Following prior work demonstrating the effect of protein length on performance [16, 28, 30], we sample 64 proteins for each protein length between $\{64, 72, 80, ..., 496, 504, 512\}$, for a total of 3648 samples. Results in Figure 5 and Tables 2 and 3 show that while PLAID performance also decreases at longer lengths, this degradation is less pronounced, and at longer lengths, PLAID can better balance quality and diversity. This may be due to the fact that the expanded dataset means that there are more samples available for lengths that are less commonly seen in the dataset. Despite not seeing structures when training the diffusion model, PLAID is able to achieve high cross-modal consistency between generated sequences and structures. Table 3 shows that the distribution of biophysical features for PLAID generations are closer to that of natural proteins, potentially due to the removed biases towards structure in its training data.

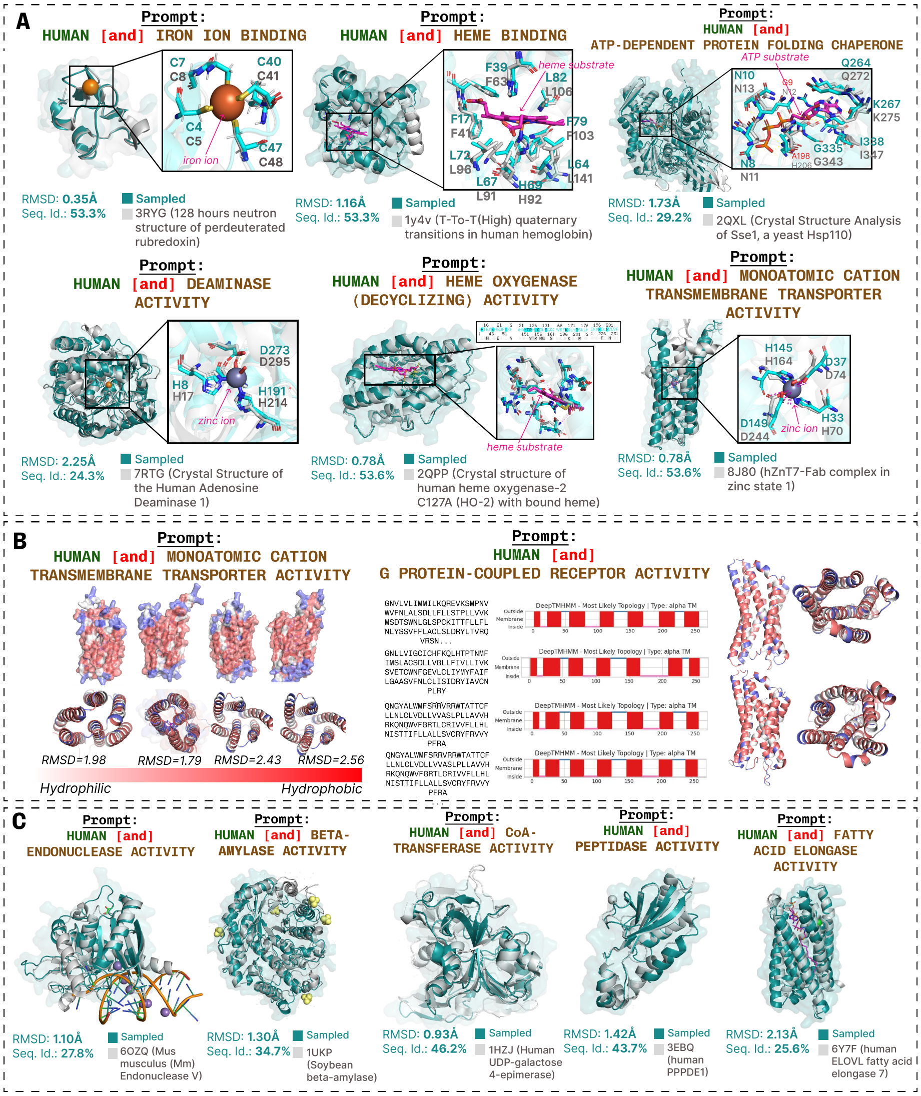

A
Prompt:
HUMAN [and] IRON ION BINDING
C7
C8
C40
C41
C4
C5
C47
C48
iron ion
RMSD: 0.35Å
Seq. Id.: 53.3%
Sampled
3RYG (128 hours neutron structure of perdeuterated rubredoxin)
Prompt:
HUMAN [and] HEME BINDING
F39
F63
heme substrate
L82
L106
F17
F41
F79
F103
L72
L96
L64
L141
L67
L91
H69
H92
RMSD: 1.16Å
Seq. Id.: 53.3%
Sampled
1y4v (T-To-T(High) quaternary transitions in human hemoglobin)
Prompt:
HUMAN [and]
ATP-DEPENDENT PROTEIN FOLDING CHAPERONE
ATP substrate
N10
N13
Q264
Q272
K267
K275
G335
G343
I338
I347
A198
H206
N8
N11
RMSD: 1.73Å
Seq. Id.: 29.2%
Sampled
2QXL (Crystal Structure Analysis of Sse1, a yeast Hsp110)
Prompt:
HUMAN [and] DEAMINASE ACTIVITY
D273
D295
H8
H17
H191
H214
zinc ion
RMSD: 2.25Å
Seq. Id.: 24.3%
Sampled
7RTG (Crystal Structure of the Human Adenosine Deaminase 1)
Prompt:
HUMAN [and] HEME OXYGENASE (DECYCLIZING) ACTIVITY
heme substrate
RMSD: 0.78Å
Seq. Id.: 53.6%
Sampled
2QPP (Crystal structure of human heme oxygenase-2 C127A (HO-2) with bound heme)
Prompt:
HUMAN [and] MONOATOMIC CATION TRANSMEMBRANE TRANSPORTER ACTIVITY
H145
H164
D37
D74
D149
D244
H33
H70
zinc ion
RMSD: 0.78Å
Seq. Id.: 53.6%
Sampled
8J80 (hZnT7-Fab complex in zinc state 1)
B
Prompt:
HUMAN [and] MONOATOMIC CATION TRANSMEMBRANE TRANSPORTER ACTIVITY
RMSD=1.98
RMSD=1.79
RMSD=2.43
RMSD=2.56
Hydrophilic
Hydrophobic
Prompt:
HUMAN [and]
G PROTEIN-COUPLED RECEPTOR ACTIVITY
GNVLVLIMMILKQREVKSMPNV WVFNLALSDLLFLLSTPLLVVK MSDTSWNLGLSPCKITTFLLFL NLYSSVFFLACLSLDRYLTVRQ VRSN...
GNLLVIGCICHFKQLHTPTNMF IMSLACSDLLVGLLFIVLLIVK SVETCWNFGEVLCLIYMYFAIF LGAASVFNLCLISIDRYIAVCN PLRY
QNGYALWMFSRRVRRWTATTCF LLNLCLVDLLVVASLPLLAVVH RKQNQWVFGRTLCRIVVFLLHL NISTTIFLLALLSVCRYFRVVY PFRA
QNGYALWMFSRRVRRWTATTCF LLNLCLVDLLVVASLPLLAVVH RKQNQWVFGRTLCRIVVFLLHL NISTTIFLLALLSVCRYFRVVY PFRA
DeepTMHMM - Most Likely Topology | Type: alpha TM
Outside
Membrane
Inside
0
50
100
150
200
250
C
Prompt:
HUMAN [and] ENDONUCLEASE ACTIVITY
RMSD: 1.10Å
Seq. Id.: 27.8%
Sampled
6OZQ (Mus musculus (Mm) Endonuclease V)
Prompt:
HUMAN [and] BETA-AMYLASE ACTIVITY
RMSD: 1.30Å
Seq. Id.: 34.7%
Sampled
1UKP (Soybean beta-amylase)
Prompt:
HUMAN [and] CoA-TRANSFERASE ACTIVITY
RMSD: 0.93Å
Seq. Id.: 46.2%
Sampled
1HZJ (Human UDP-galactose 4-epimerase)
Prompt:
HUMAN [and] PEPTIDASE ACTIVITY
RMSD: 1.42Å
Seq. Id.: 43.7%
Sampled
3EBQ (human PPPDE1)
Prompt:
HUMAN [and] FATTY ACID ELONGASE ACTIVITY
RMSD: 2.13Å
Seq. Id.: 25.6%
Sampled
6Y7F (human ELOVL fatty acid elongase 7)

Figure 6: **Function conditioned generation for human proteins. (A) PLAID generations preserve catalytic motifs at non-adjacent residues, despite maintaining low sequence identity.** For each generation, we examine the closest Foldseek neighbor in the PDB [58] that was crystallized in complex with a ligand, to analyze residue behaviors at the active site. RMSD is the global structural alignment between generation and target. Sequence identity is the sequence lap in the aligned structural regions. **(B) Generated membrane proteins recapitulate known hydrophobicity patterns**. *(Left)* Generated samples match known hydrophobicity patterns of membrane proteins, where hydrophobic residues are found in the transmembrane portions that span the lipid bilayer's hydrophobic core, hydrophilic residues at the ends which interact with the aqueous environment, and for ion transporters, coordinated hydrophilic residues at the core for mediating ion interactions. *(Right)* For GPCR samples, structures exhibit the expected 7-helix structure. DeepTMMHMM [59] predictions on sequences classifies generations as alpha transmembrane proteins, matching the known topology of GPCRs. **(C)** Additional generations. Samples consistently exhibit high structural conservation that indicates preservation of function, yet can attain high degrees of sequence diversification.

5.2 CONDITIONAL GENERATION

Computational evaluation of function- and organism-conditioned generative models presents a conundrum: lower similarity is a favorable heuristic in machine learning, since it is indicates that the generative model did not merely memorize the training data. From a bioinformatics perspective, however, conservation is key to function; taxonomic membership can be difficult to validate, given the high degree of similarity between homologs. In our case study experiments, we look for **high structural similarity to evaluate for function conditioning**, and **low sequence similarity to penalize exact memorization**. For organisms in particular, differences are more likely to manifest at the sequence rather than the structural level. Case studies shown in Figure 6 show that function-conditioned proteins possess known biological characteristics, such as conserved active site motifs, and membrane hydrophobicity patterns. Global sequence diversity is low despite high levels of conservation at catalytic sites, suggesting the the model has learned key biochemical features associated with the function prompt without direct memorization.

We further by examining the Sinkhorn distance between generated latents and a reference distribution, taken from a heldout validation set unseen during training (Figure 10D). This assess conditional generations indepedent of the sequence and structural decoders. For comparison, the Sinkhorn distance between random real proteins from the validation set and the function-conditioned generations are also evaluated. Conditional generations generally have lower Sinkhorn distances than random samples, suggesting that the desired latent information has been captured in the embedding. Figure10B shows tSNE plots colored by organism. Organisms that are further away phylogenetically (e.g. soybean, E. coli) form more distinct clusters than those closer evolutionarily (e.g. human, mouse). These all serve to demonstrate that our function and organism conditioned samples have been imbued with desired characteristics.

6 DISCUSSION

We proposed PLAID, a paradigm for multi-modal, controllable generation of proteins by diffusing in the latent space of a prediction model that maps single sequences to the desired modality. Our method is designed to adhere to progress in **data availability, model scalability, and sequence-to-structure prediction capabilities**. To this end, we chose an architecture and implementation that leverages fast attention kernels [41], and chose GO terms as a proxy for the vast quantities of language annotation that are paired with sequence databases (but are more scarce for structural ones).

It is straightforward to expand PLAID to many downstream capabilities. First, though we do not examine motif scaffolding or binder design explicitly in this current work, this is easy to build into PLAID by holding some input residues constant. Second, though we examine ESMFold [3] in this work, the method can be applied to any prediction model. There is rapid progress [5, 4, 60, 61, 62] in predicting complexes from structure, owing to the vast differential in data access costs between sequences and experimentally-resolved complexes, and diffusing in the latent space of such models enables us to use the frozen decoder to obtain more modalities than just all-atom structure.

A limitation of PLAID is that performance is limited by prediction model from which the frozen decoders are derived. Here, we rely on the optimism that such models will continue to improve. With explicit finetuning for latent generation (e.g. training CHEAP and the structure decoder end-to-end), model performance can likely be improved. Furthermore, since the structure decoder is deterministic, it is unable to sample different conformations in its current form. One solution is to diffuse in the latent space of a model that returns a distribution over structural conformations instead. Additionally, the GO term one-hot encoding used here does not take into account the hierarchical nature of the Gene Ontology vocabulary, nor that a protein might have several relevant GO terms. Finally, the classifier-free guidance scale can be separated for the organism and function conditions, since the two may require different guidance strengths to produce a desired sample. These limitations are relatively simple to resolve, and will be addressed in future work.

REFERENCES

[1] Jeffrey F Ohren, Huifen Chen, Alexander Pavlovsky, Christopher Whitehead, Erli Zhang, Peter Kuffa, Chunhong Yan, Patrick McConnell, Cindy Spessard, Craig Banotai, et al. Structures of human map kinase kinase 1 (mek1) and mek2 describe novel noncompetitive kinase inhibition. *Nature structural & molecular biology*, 11(12):1192–1197, 2004.

[2] Vivek Modi and Roland L Dunbrack Jr. Kincore: a web resource for structural classification of protein kinases and their inhibitors. *Nucleic acids research*, 50(D1):D654–D664, 2022.

[3] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[4] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[5] Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[7] Amy X Lu, Wilson Yan, Kevin K Yang, Vladimir Gligorijevic, Kyunghyun Cho, Pieter Abbeel, Richard Bonneau, and Nathan Frey. Tokenized and continuous embedding compressions of protein sequence and structure. *bioRxiv*, pages 2024–08, 2024.

[8] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[10] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[14] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[15] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[16] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620:1089–1100, 2023.

[17] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[18] Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion. *arXiv*, 2209.15611, 2022.

[19] Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[20] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[21] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378:49–56, 2022.

[22] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *Proceedings of the 39th International Conference on Machine Learning*, 162:8946–8970, 2022.

[23] Nathan C Frey, Daniel Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, et al. Protein discovery with discrete walk-jump sampling. *arXiv preprint arXiv:2306.12360*, 2023.

[24] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13:4348, 2022.

[25] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106, 2023.

[26] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. RITA: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

[27] Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

[28] Alexander E Chu, Lucy Cheng, Gina El Nesr, Minkai Xu, and Po-Ssu Huang. An all-atom protein generative model. *bioRxiv*, 2023.

[29] Sidney Lyayuga Lisanza, Jacob Merle Gershon, Sam Wayne Kenmore Tipps, Lucas Arnoldt, Samuel Hendel, Jeremiah Nelson Sims, Xinting Li, and David Baker. Joint generation of protein sequence and structure with RoseTTAFold sequence space diffusion. *bioRxiv*, 2023.

[30] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[31] Karolis Martinkus, Jan Ludwiczak, Wei-Ching Liang, Julien Lafrance-Vanasse, Isidro Hotzel, Arvind Rajpal, Yan Wu, Kyunghyun Cho, Richard Bonneau, Vladimir Gligorijevic, et al. Abdiffuser: full-atom generation of in-vitro functioning antibodies. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

[33] Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07, 2024.

[34] Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods*, pages 1–11, 2024.

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[36] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[37] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature methods*, 16(7):603–606, 2019.

[38] Andre Cornman, Jacob West-Roberts, Antonio Pedro Camargo, Simon Roux, Martin Beracochea, Milot Mirdita, Sergey Ovchinnikov, and Yunha Hwang. The omg dataset: An open metagenomic corpus for mixed-modality genomic language modeling. *bioRxiv*, pages 2024–08, 2024.

[39] TGO Consortium, Suzi A Aleksander, James Balhoff, Seth Carbon, JM Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, and Nomi L Harris. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

[40] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[41] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

[42] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[43] Reza Yazdani Aminabadi, Samyam Rajbhandari, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Olatunji Ruwase, Shaden Smith, Minjia Zhang, Jeff Rasley, et al. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2022.

[44] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[45] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35: 16344–16359, 2022.

[46] Markus N Rabe and Charles Staats. Self-attention does not need o(n2) memory. *arXiv preprint arXiv:2112.05682*, 2021.

[47] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7441–7451, 2023.

[48] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024.

[49] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[50] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[51] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

[52] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

[53] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[54] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[55] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[56] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[57] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

[58] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[59] Jeppe Hallgren, Konstantinos D Tsirigos, Mads Damgaard Pedersen, José Juan Almagro Armenteros, Paolo Marcatili, Henrik Nielsen, Anders Krogh, and Ole Winther. Deeptmhmm predicts alpha and beta transmembrane proteins using deep neural networks. *BioRxiv*, pages 2022–04, 2022.

[60] Chai Discovery, Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life. *bioRxiv*, pages 2024–10, 2024.

[61] Lihang Liu, Shanzhuo Zhang, Yang Xue, Xianbin Ye, Kunrui Zhu, Yuxin Li, Yang Liu, Wenlai Zhao, Hongkun Yu, Zhihua Wu, et al. Technical report of helixfold3 for biomolecular structure prediction. *arXiv preprint arXiv:2408.16975*, 2024.

[62] Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, et al. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, pages 2024–11, 2024.

[63] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021.

[64] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

APPENDIX

A ADDITIONAL TRAINING DETAILS

Our 2B model is trained with the memory efficient attention implementation in xFormers. It is trained with float32 precision; at inference time, sequence lengths must be a multiple of 4. All other models are trained with mixed precision (bfloat16 and float32). All models were trained with a learning rate of 1e-4 with a cosine annealing applied over 1,000,000 steps.

For the 2B model used in most evaluations, since the specialized xFormers memory efficient attention kernel was used, lengths must be a multiple of 8 – that is, the latent embedding length must be a multiple of 4, which is upsampled to a multiple of 8 after the decoder is applied. We selected 512 based on the distribution of sequences in Pfam and a shorten factor of 2 based on results in Lu et al. [7].

Following Ho and Salimans [9], we use the same hyperparameters and with $p_{\text{uncond}} = 0.3$, the class label is replaced with the $\varnothing$ unconditional token. sampled separately for both function and organism. Note that not all data samples will have an associated GO term; we use the $\varnothing$ token for those cases as well. At inference time, to perform generate unconditionally (for either or both of function and/or organism), we use the $\varnothing$ token for conditioning.

B CHEAP COMPRESSION DETAILS

Briefly, the CHEAP encoder and decoder uses an Hourglass Transformer [63] architecture that downsamples lengthwise, as well as downprojects the channel dimension, to create a bottleneck layer, the output of which is our compressed embedding. The entire model is trained with the reconstruction loss $MSE(\mathbf{x}, \hat{\mathbf{x}})$. Authors show that structural and sequence information in ESMFold latent spaces are in fact highly compressible, and despite using very small bottleneck dimensions, reconstruction performance can be nonetheless maintained when evaluated in sequence or structure space.

Based on reconstruction results in Lu et al. [7], we choose $\mathbf{x}' \in \mathbb{R}^{\frac{L}{2} \times 32}$ with $L = 512$, which balances reconstruction quality at a resolution comparable to the size of latent spaces in image diffusion models [11]. Dividing the length in half allows us to better leverage the scalability and performance of Transformers, while managing its $\mathcal{O}(L)$ memory needs.

The CHEAP module involves a channel normalization step prior to the forward pass through the autoencoder. We find that the distribution of embedding values is fairly “smooth" here (Figure 7). Though the original Rombach et al. [11] paper was trained with a KL constraint to a Gaussian distribution, we use the embedding output as is. CHEAP embeddings were also trained with a $tanh$ layer at the output of the bottleneck; this allows us to clip our samples between $[-1, 1]$ at each diffusion iteration, as was done in original image diffusion works [6, 9, 12, 64]. We found in early experiments being able to clip the output values were very helpful for improving performance.

Without using the CHEAP compression prior to diffusion, sample quality was poor, even on short ($L = 128$) generations.

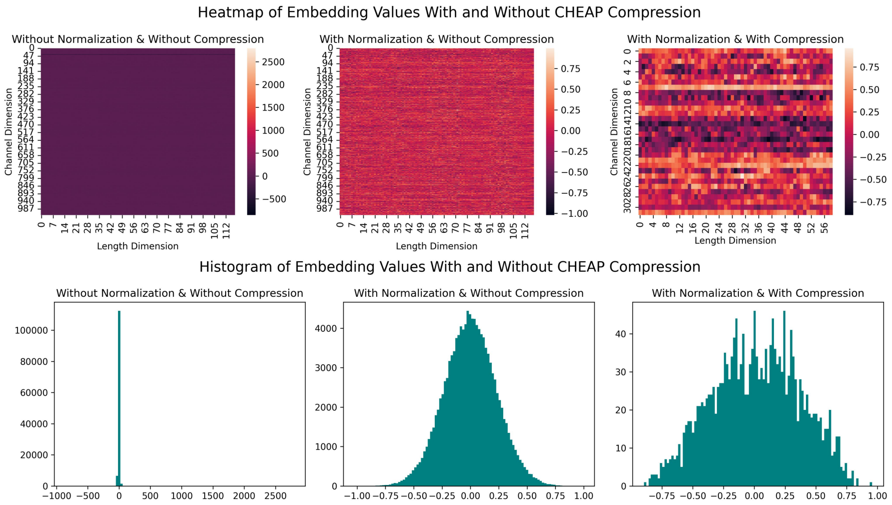
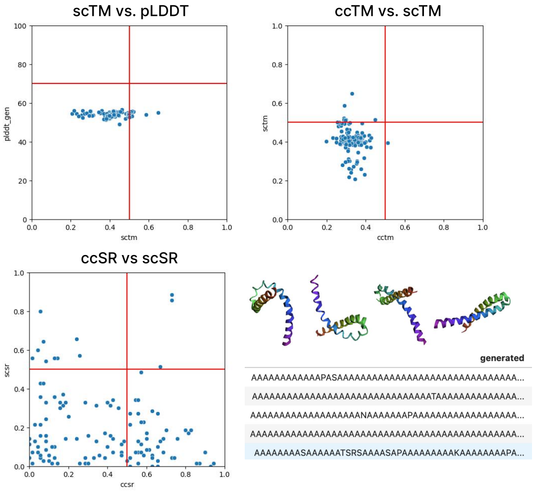


Figure 7: Visualizing the original ESMFold latent space before normalization, after per-channel normalization, and after compression. The value distribution of $p(\mathbf{x}')$ is fairly smooth and "Gaussian-like", making it amenable to diffusion.



generated
AAAAAAAAAAAAAPASAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAATAAAAAAAAAAAAAA...
AAAAAAAAAAAAAAAAAAAAANAAAAAAAPAAAAAAAAAAAAAAAAAAAA...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
AAAAAAAAASAAAAAAATSRSAAAASAPAAAAAAAAAAKAAAAAAAAAPA...

Figure 8: Results when running PLAID on the ESMFold latent space naively without CHEAP compression, for proteins of length 128. There is a tendency to generate repeated sequences, and

C DATA

We use the September 2023 Pfam release, consisting of dataset, consisting of 57,595,205 sequences and 20,795 families. PLAID is fully compatible with larger sequence databases such as UniRef or BFD (roughly 2 billion sequences), which would offer even better coverage. We elect to use Pfam because sequence domains have more structure and functional labels, which is easier for *in silico* evaluation of generated samples. We also hold out about 15% of the data for validation.

Approximately 46.7% of the dataset (N=24,637,236) is annotated with a GO term. Using the publicly available mapping as of July 1, 2024, we take a count of all GO occurrences; for each Pfam entry with multiple GO entries, we pick the one with the fewest GO occurences to encourage more descriptive and distinct GO labels.

The `Pfam-A.fasta` file available from the Pfam FTP server includes the UniRef code of the source organism from which the Pfam domain is from. The UniRef code further more includes a 5 letter “mneumonic" to denote the organism. We examine all unique organisms in our dataset, and find 3617 organisms.

D SAMPLING

Inference-time sampling hyperparameters provides the user with additional control over quality and sampling speed trade-off. PLAID supports the DDPM sampler [6] and the DDIM sampler [12], as well as the improved speed samplers from DPM++ [53]. We find that using the DDIM sampler with 500 timesteps using either the sigmoid or cosine schedulers works best during inference, and reasonable samples can be obtained using the DPM++2M-SDE sampler with only 20 steps. Experiments shown here uses DDIM sampler with the sigmoid noise schedule at 500 timesteps.

Note that the performance bottleneck is found mostly during the latent sampling and structure decoding (which depends on the number of recycling iterations [20, 3] used); however, these two processes can be easily decoupled and parallelized, which cannot be done in existing protein diffusion methods. Furthermore, it allows us to prefilter which latents to decode using heuristic methods, and decode only those latents to structure, which would boost performance for nearly the same computational cost. We do not empirically explore this in this paper to provide a fair comparison, and because the filtering criteria would vary greatly by downstream use.

E EVALUATION DETAILS

For all benchmarks and models, we use default settings provided in their open-source code. For ProteinMPNN [21], we use the `v_48_002` model with a sampling temperature of 0.1 and generate 8 sequences for protein, from which the best performing sequence use chosen. To calculate self-consistency, we fold sequences using OmegaFold [56] rather than ESMFold.

Though our models generate all-atom structure, we examine C_α RMSD rather than all-atom RMSD, to avoid mis-attributing sequence generation under-performance to structure generation failures. Also, since there are usually differences in the sequence that is generated, different number of atoms make it difficult to assess all-atom RMSD.

For the hold-out natural reference dataset, we use sequences from Pfam and keep length distributions similar to that of the sampled proteins. Specifically, for each sequence bin between $\{64, 72, ..., 504, 512\}$, we take 64 natural sequences of that length. For the experiment in Figure 10D, we use the Sinkhorn Distance rather than the Frechet Distance used commonly in images and video. Since not all functions have a large number of samples, we elected to use a metric that works better in low sample settings.

Structure novelty is obtained by searching samples to PDB100 using Foldseek [55] `easy-search`. We examine the TM-score to the closest neighbor. For Foldseek and MMseqs experiments, all clustering experiments are performed by length. We use default settings for both tools. Though we report the average TMScore to top neighbor for Foldseek, we run `easy-search` in 3Di mode. For sequences, we use MMseqs2 [54] to see if sequences have a homolog in UniRef50, using default sensitivity settings. For samples with homologs, we further calculate the average sequence identity to the closet neighbor to assess novelty (Seq ID %).

F SAMPLING SPEEDS

We examine the amount of time necessary for generating a simple sample. We first explore the time necessary to generate 100 sequences with $L = 600$. Multiflow and ProteinGenerator does not support batched generation in its default implementation, so in this experiment, we simply generate one sample at a time for a total of 100 samples. We report the amount of time per sample. For comparison, we also run an experiment where we only generate a single sample, such that none of the methods can make use of any improvements from batching.

Table 4: Time required to sample **proteins with 600 residues**. We assess time required both for sampling $N = 100$ samples in batches whenever possible, and when generating a single sequence. Experiments are run on Nvidia A100. Methods marked by (*) do not support batching in the default implementation

	seconds/sample, batched		seconds/sample, unbatched	
	Sample Latent	Decode	Sample Latent	Decode
Protpardelle	11.21	-	17.16	-
Multiflow*	231.32	-	277.11	-
ProteinGenerator*	343.32	-	342.28	-
PLAID (100M)	1.64	15.12	27.63	1.07
PLAID (2B)	19.34	15.07	49.03	0.9

G ATTENTION SPEED

Forward pass benchmark of vanilla multihead attention compared to the optimized xFormers implementation of memory-efficient attention [46] and FlashAttention-2 [42]. Though FlashAttention2 performed best in our benchmarks, a fused kernel implementation with key padding was not yet available at the time of writing. Since our data contained different lengths (as compared to most image diffusion use cases, or language use-cases that can make use of the implemented causal masking), we instead use the xFormers implementation. We expect that sampling speed results would improve once this feature is becomes available in the FlashAttention package.

Method	**Mean Time (s)**	**Mean Memory (GB)**
Standard Multihead Attention	0.0946 ± 9.23e-4	76.0 ± 0.409
xFormers Memory Efficient Attention	0.0519 ± 4.33e-05	64.0 ± 0.409
Flash Attention	0.0377 ± 1.91e-3	49.2 ± 0.783

H ADDITIONAL RESULTS

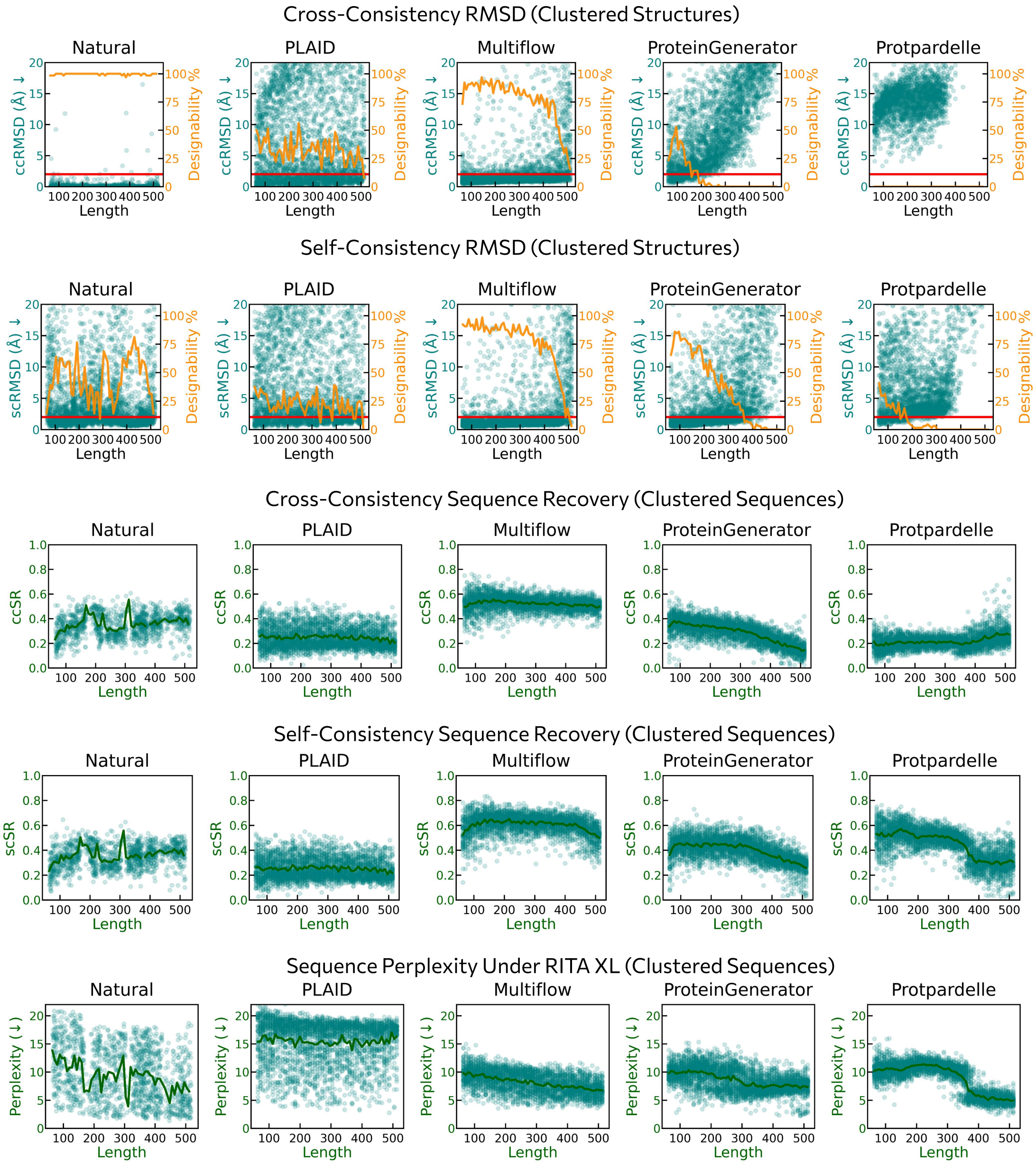


Figure 9: More comparison results between PLAID and baselines.

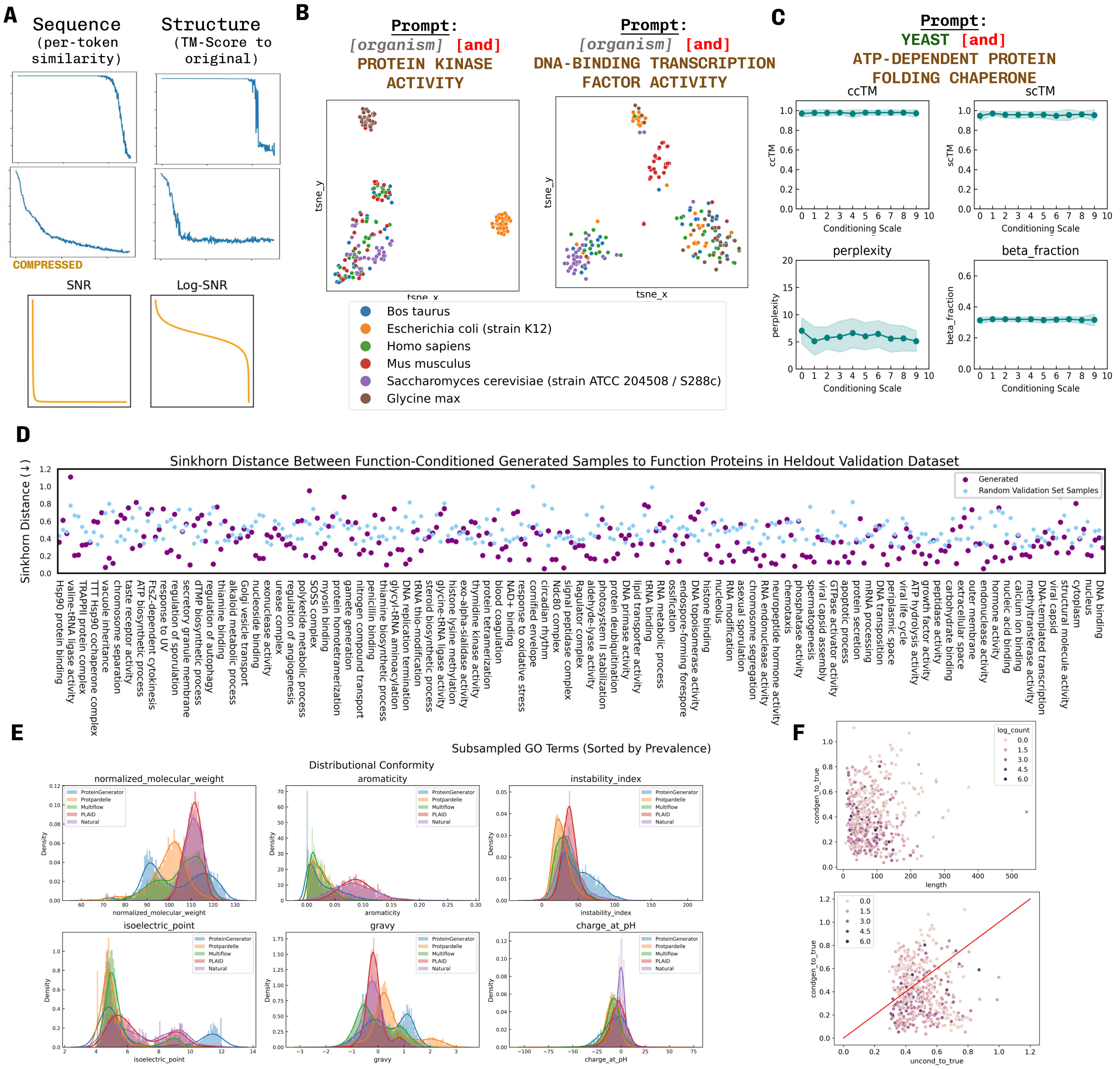


Figure 10: **(A)** Without compression, noise added in the latent space (SNR and logSNR curves visualized in yellow) does not map to corruptions in the sequence and structure spaces until the final diffusion timesteps. This means that at most sampled diffusion timesteps during training, there is negligible noise added to the structure or sequence itself. After compression, corruptions in the sequence and structure space maps better to the intended signal-to-noise ratio. **(B)** T-SNE embeddings of latent generations, colored by organism. **(C)** Examining the effect of conditioning scale on the output quality. **(D)** Sinkhorn distance between GO term conditioned latent and validation set samples of real proteins with a a given GO term. Blue diamonds represent Sinkhorn distance between a random set of real proteins in the hold out set (i.e. not actually annotated with the GO term we are looking at) to the set of real samples *with* the GO term. **(E)** Visualizing distribution conformity. For each biophysical property, and for each method, we plot the biophysical property distribution for samples. PLAID samples adhere best to natural proteins. **(F)** Analyzing what factors might contribute to a greater δ difference between the Sinkhorn distance of samples-to-real-functional-proteins vs random-proteins-to-real-functional-proteins.

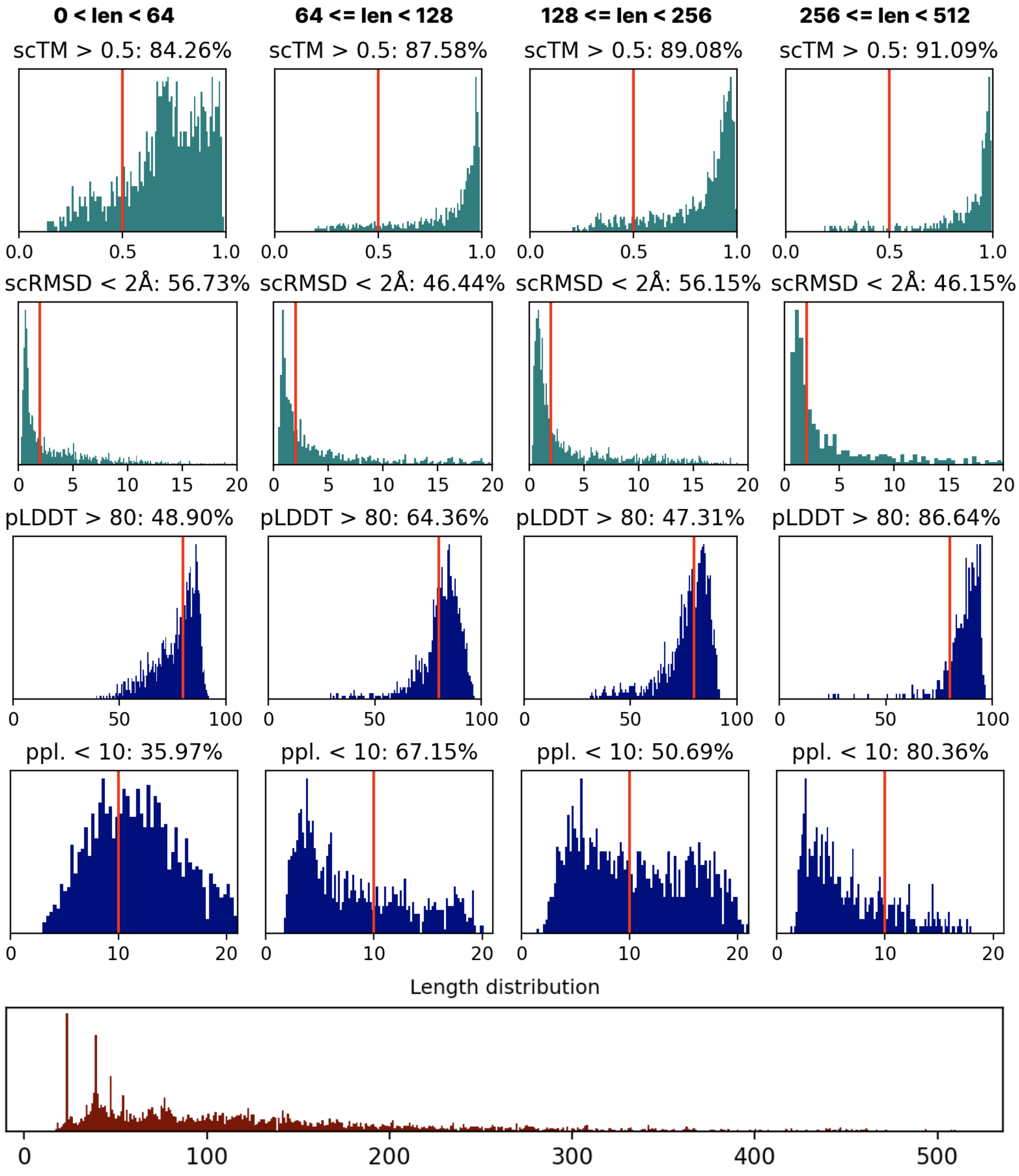


Figure 11: To examine the degree to which co-generation methods are overfitting to structure-based metrics, we examine properties on natural proteins.

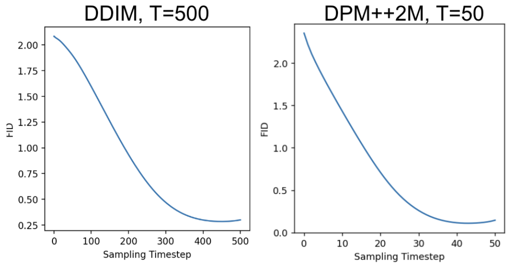


Figure 12: FID across sampling (reverse diffusion) timesteps for the DDIM [12] sampler and the DPM++2M [53] sampler. For both, sample quality decreases steadily over time before plateauing. DPM++2M can achieve low FID results with only 10% of the original number of steps, but final results are still slightly worse.