DISTILLING STRUCTURAL REPRESENTATIONS INTO PROTEIN SEQUENCE MODELS

Jeffrey Ouyang-Zhang, Chengyue Gong, Yue Zhao, Philipp Krähenbühl, Adam R. Klivans, Daniel J. Diaz
University of Texas at Austin
{jozhang,cygong17,yzhao,philkr,klivans,danny.diaz}@utexas.edu

ABSTRACT

Protein language models, like the popular ESM2, are widely used tools for extracting evolution-based protein representations and have achieved significant success on downstream biological tasks. Representations based on sequence and structure models, however, show significant performance differences depending on the downstream task. A major open problem is to obtain representations that best capture both the evolutionary and structural properties of proteins in general. Here we introduce ***I**mplicit **S**tructure **M**odel* (*ISM*), a sequence-only input model with structurally-enriched representations that outperforms state-of-the-art sequence models on several well-studied benchmarks including mutation stability assessment and structure prediction. Our key innovations are a microenvironment-based autoencoder for generating structure tokens and a self-supervised training objective that distills these tokens into ESM2's pre-trained model. We have made *ISM*'s structure-enriched weights easily available: integrating ISM into any application using ESM2 requires changing only a single line of code. Our code is available at https://github.com/jozhang97/ISM.

1 INTRODUCTION

Protein language models (pLMs) are versatile feature extractors with proven success across numerous downstream applications (Elnaggar et al., 2021; Brandes et al., 2022; Rives et al., 2019; Lin et al., 2022). Their accessibility has significantly democratized protein research, enabling biologists with limited computational expertise to apply advanced machine learning techniques to their specific protein domain. The method's success comes from its exclusive use of sequences, bypassing costly, unreliable, or infeasible structure computations and sophisticated data-engineering pipelines.

The tradeoff is that pLMs are often lack structural context and underperform (relative to structure-based models) on tasks that typically require structural insight (Su et al., 2023; Yang et al., 2023; Zhang et al., 2024; Gaujac et al., 2024; Frolova et al., 2024; Li et al., 2024; Kulikova et al., 2023; Allman et al., 2024). Longstanding biological research (Anfinsen, 1973) does suggest that the amino acid sequence is solely responsible for the folding of the structure. Indeed, sequence-only models trained using masked language modeling learn to extract structure features encoded in evolutionary co-variations (Lin et al., 2022). However, current state-of-the-art frameworks, such as AlphaFold, require the protein's evolutionary history as an additional input, demonstrating that sequence-only models fail to extract all the structural information within a multiple sequence alignment (MSA). Building a *single-sequence* model (without additional MSA input) that leads to structurally-informed representations remains a challenging open problem.

In this paper, we introduce ***I**mplicit **S**tructure **M**odel* (*ISM*), a sequence-only protein language model that is trained to *implicitly* capture structural information. Our key contribution is a new self-supervised pre-training objective, *structure-tuning*, where the sequence model learns to distill features derived from structure-based models (see Figure 1). As a result, *ISM* outperforms sequence-only models and is competitive with pLM frameworks that *explicitly* take the protein structure as an additional input. For example, on the CAMEO protein structure prediction benchmark *ISM* outperforms its ESM2 counterpart with a GDT-TS score of 0.67 versus 0.64 (see Table 1). For S669 $\Delta\Delta G$ prediction, *ISM* surpasses ESM2 in AUC (0.76 vs 0.72) and even matches specialized models

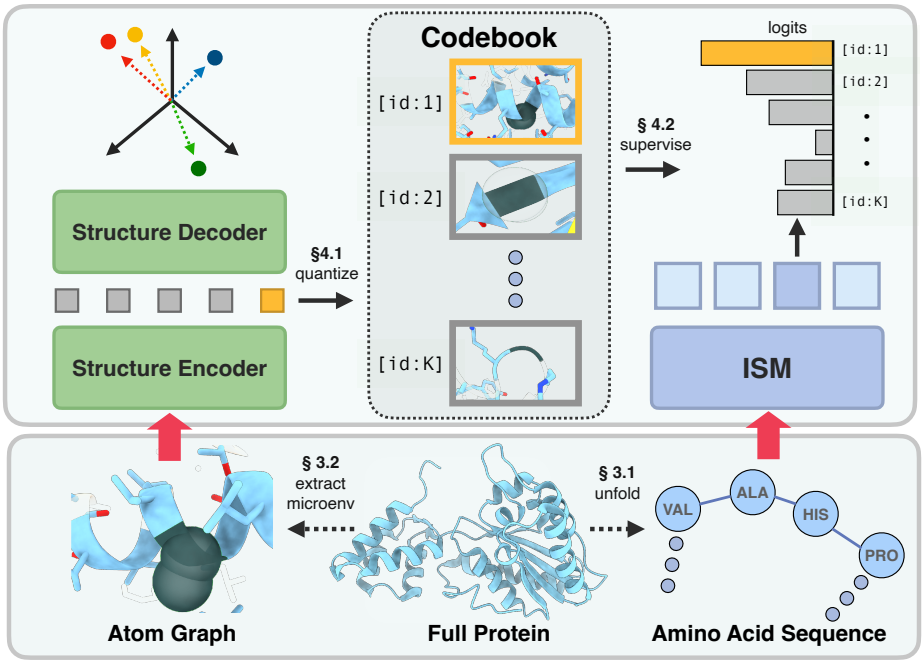


Figure 1: **Structure-tuning a protein language model.** ***I****mplicit* ***S****tructure* ***M****odel* (*ISM*) is a sequence-only protein model (right) supervised by structure tokens derived from a structure model (left). For every residue, a structure encoder takes the atoms of a residue's microenvironment as input and produces a structural representation. We discretize these representations into tokens using a codebook extracted via k-means clustering. The *ISM* sequence model learns to predict these structure tokens.

that process atomic environments (0.76 vs 0.75, see Table 2). Our results align with prior works that show multiple modalities enhance model performance (Gong et al., 2024; Hayes et al., 2024).

Structure-tuning is a fine-tuning technique where a sequence-only model is trained to predict structure tokens – rather than masked amino acids – for each protein residue (see Figure 1). Our structure tokens, derived from our Atomic Autoencoder and MutRank (Gong et al., 2024), capture key chemical interactions that underpin the protein's tertiary structure. Structure-tuning distills these structural representations into *ISM*, as demonstrated by the significant improvement in predicting long-range tertiary interactions (0.49 vs 0.35, see Table 1).

2 Related Work

Protein Language Models. These models take an amino acid sequence as input and produce a deep representation for each amino acid conditioned on the entire sequence. Commonly-used models such as ProtBERT, ProteinBERT, ESM1b, and ESM2 use transformer-based architectures and are trained to maximize wildtype accuracy (*i.e.,* reconstruct masked amino acids) (Elnaggar et al., 2021; Brandes et al., 2022; Rives et al., 2019; Lin et al., 2022).

One of the motivations behind ESM2 was to build a single-sequence variant of AlphaFold that does not require the computationally expensive task of generating MSAs. The resulting model, ESMFold, is a widely used tool but generally underperforms when compared to AlphaFold in terms of predicted structural quality. This demonstrates that ESM2 does not fully capture the epistatic landscape induced during evolution. This has motivated research on augmenting sequence models with a structural modality, and we describe some of these works below.

Sequence models with structure loss. The ESM2-s sequence model incorporates structural information by fine-tuning ESM2 to predict a protein's structural fold (Zhang et al., 2024). The fold of a protein, however, is biologically coarse-grained information. *ISM* achieves superior performance by using the more fine-grained approach of training at the residue level. More specifically, in our training objective, each residue is tasked with predicting its corresponding local structural environment.

S-PLM and "Structure-infused protein language models (SIPLM)" use a type of CLIP training to align sequence and structural features (Wang et al., 2023; Peñaherrera & Koes, 2023). This technique is also coarse-grained because its training objective does not operate at a residue level (we do not include SIPLM in our tables of results due to its relatively weak performance on our benchmarks).

AlphaFold also learns structural representations from sequences (Jumper et al., 2021). However, it requires a multiple sequence alignment as input, which is expensive to compute and often unavailable for many practical applications. Furthermore, prior works have shown that Evoformer, the feature extractor for AlphaFold, underperforms ESM2 on various downstream tasks that involve less structural information (Hu et al., 2022). On these tasks, *ISM* still achieves comparable performance to ESM2.

Sequence models with structure inputs. These models extend sequence models by using the structure as an additional input. SaProt (Su et al., 2023) and ProstT5 (Heinzinger et al., 2023) use the VQVAE from FoldSeek (van Kempen et al., 2022) to extract per-residue structure tokens as additional inputs to a protein language model. MULAN (Frolova et al., 2024) extends these works to include structural features (torsion angles) as additional inputs. Similarly, ProSST (Li et al., 2024) also takes structural tokens as inputs. However, instead of using FoldSeek tokens, ProSST trains a Denoising Autoencoder to extract per-residue features, which are then tokenized into a structure sequence using K-means clustering. All these models require a protein structure as input at inference time. There are well-known drawbacks to frameworks requiring structure as input. In addition to requiring a more sophisticated data engineering pipeline, there are some cases where the structure has not been experimentally resolved and cannot be accurately modeled using computational tools (*e.g.,* antibody-antigen complexes, conformer specific protein-protein interactions, post-translation modification-dependent conformations, interfaces, etc).

Protein Structure Autoencoders. These autoencoders are structure-based models that take the backbone atom coordinates as input and encode each residue into a discrete token (Gaujac et al., 2024; Hayes et al., 2024). The sequence of discrete tokens is used to reconstruct the positions of backbone atoms using coordinate losses (*e.g.,* frame aligned point error, distogram classification). Protein structure denoising Autoencoders take a noisy variant of the protein backbone as input and then learn a latent embedding that decodes the backbone atoms (Peñaherrera & Koes, 2023; Li et al., 2024). Foldseek (van Kempen et al., 2022) extracts features for a residue given the backbone geometry of its nearest neighbors. Unlike our approach, these works use only the protein backbone as input. We also train a structural autoencoder, but instead of reconstructing the local backbone of a protein, we reconstruct the coordinates of all atoms within the local chemical environment surrounding a masked residue (masked microenvironment).

3 PRELIMINARIES

Let $\boldsymbol{x}_{\text{seq}} = (x_1, ..., x_L)$ be a protein sequence of L amino acids where each amino acid residue $x_l \in \{\text{A}, \text{C}, ..., \text{Y}\}$. The atoms defined by this sequence fold into an energetically favorable 3-dimensional structure $\boldsymbol{x}_{\text{struct}} = \{(p_i, e_i, \boldsymbol{c}_i)\}_{i=1}^{N}$ where each atom i consists of residue sequence position $p_i \in \{1, ..., L\}$, an element type $e_i \in \{\text{C}, \text{H}, \text{N}, \text{O}, \text{P}, \text{S}, \text{X}\}$ and coordinates $\boldsymbol{c}_i \in \mathbb{R}^3$.

3.1 PROTEIN SEQUENCE MODELS

A protein language model $\mathbf{pLM}$ takes a protein sequence $\boldsymbol{x}_{\text{seq}}$ as input and produces a latent representation $\mathbf{pLM}(\boldsymbol{x}_{\text{seq}}) \in \mathbb{R}^{L \times D}$ for downstream tasks. Most models use a transformer architecture and are pre-trained via a masked language modeling (MLM) loss. During training, a subset $\mathbb{M} \subset \{1, ..., L\}$ of the sequence is replaced with the `[mask]` token $\tilde{x}_i = \begin{cases} \texttt{[mask]} & \text{if } i \in \mathbb{M} \\ x_i & \text{otherwise} \end{cases}$ with $\tilde{\boldsymbol{x}}_{\text{seq}} = (\tilde{x}_1, ..., \tilde{x}_L)$. The model learns to reconstruct the masked tokens with

$$\mathcal{L}_{\text{MLM}} = \frac{1}{|\mathbb{M}|} \sum_{i \in \mathbb{M}} \ell_{\text{CE}}(\boldsymbol{C}_{\text{MLM}}^{\top} \mathbf{pLM}(\tilde{\boldsymbol{x}}_{\text{seq}})_i, x_i), \tag{1}$$

for the cross entropy loss ℓ_{CE}, indexed feature $\mathbf{pLM}(\tilde{\boldsymbol{x}}_{\text{seq}})_i \in \mathbb{R}^D$ at position i, and a linear classification head $\boldsymbol{C}_{\text{MLM}}$ that predicts the amino acid type. While the backbone $\mathbf{pLM}$ is used for downstream tasks, $\boldsymbol{C}_{\text{MLM}}$ is only used for pre-training.

3.2 Protein Structure Models

An all-atom protein structure model **pSM** computes an atom-level feature representation from the local geometric description of each residue. It starts from a microenvironment $\boldsymbol{x}^l_{\text{microenv}}$ that contains all atoms in a radius $r = 10$Å around $\boldsymbol{\alpha}_l \in \mathbb{R}^3$, the coordinates of the α-carbon of residue l:

$$\boldsymbol{x}^l_{\text{microenv}} = \{(e_i, \boldsymbol{c}_i) : \forall i \in \{1, ..., N\} \text{ such that } ||\boldsymbol{c}_i - \boldsymbol{\alpha}_l|| < r\}.$$

A common backbone for protein structure models is a Graph Transformer G (Ying et al., 2021). The graph transformer $G(\boldsymbol{x}^l_{\text{microenv}})$ embeds each atom's element type e_i in a set $\mathbf{e} = \{e_1, ..., e_{n'}\}$, where n' is the number of atoms in the microenvironment. In attention updates, the graph transformer adds an attention bias $B^l_{ij} = \|\boldsymbol{c}_i - \boldsymbol{c}_j\|$ based on the pairwise distance between atoms i and j. This attention bias B^l is the only structural information given to the transformer. The graph transformer then produces a set of output features $\{\boldsymbol{z}^l_1, ..., \boldsymbol{z}^l_{n'}\} = G(\boldsymbol{x}^l_{\text{microenv}})$, one per input atom e_i. The graph transformer is commonly trained on the downstream task using a supervised learning objective (Ying et al., 2021). In this work, we use the Graph Transformer directly to train a structure model on atomic reconstructions of proteins in our pre-training dataset.

MutComputeX-GT (Diaz et al., 2024) pre-trains a Graph Transformer using a structural analog of masked language modeling. They define a masked microenvironment $\boldsymbol{x}^l_{\text{masked-microenv}}$ that contains all atoms of other residues $p_i \neq l$

$$\boldsymbol{x}^l_{\text{masked-microenv}} = \{(e_i, \boldsymbol{c}_i) : \forall i \in \{1, ..., N\} \text{ such that } p_i \neq l \text{ and } ||\boldsymbol{c}_i - \boldsymbol{\alpha}_l|| < r\},$$

and pool all-atom level features into a single residue level embedding $\boldsymbol{z}^l = \frac{1}{n}\sum_i \boldsymbol{z}^l_i$ for $\{\boldsymbol{z}^l_1, ..., \boldsymbol{z}^l_n\} = G(\boldsymbol{x}^l_{\text{masked-microenv}})$ where n is the number of atoms in the masked microenvironment. They then predict the masked-out amino acid type x_l:

$$\mathcal{L}^l_{\text{AA}} = \ell_{\text{CE}}(\boldsymbol{C}^\top_{\text{AA}} \boldsymbol{z}^l, x_l). \tag{2}$$

where $\boldsymbol{C}_{\text{AA}}$ is a linear classification head.

MutRank (Gong et al., 2024) uses the EvoRank self-supervised training objective to learn the evolutionary mutational landscape of a residue from the masked microenvironment. More specifically, it learns to predict an evolutionary score derived from the protein's multiple sequence alignment.

4 Method

ISM is a sequence model that takes as input only an amino acid sequence $\boldsymbol{x}_{\text{seq}} = (x_1, ..., x_L)$ but is trained to implicitly capture structural information. We start by training an Atomic Autoencoder, based on a Graph Transformer, on protein structures. The autoencoder is trained with a geometric reconstruction loss and the MutComputeX-GT objective $\mathcal{L}^l_{\text{AA}}$. We then cluster the resultant features into one of K structure tokens. We use the sequence $\boldsymbol{s} = (s_1, ..., s_L)$ of structure tokens $s_l \in \{1, ..., K\}$ as additional supervisory signal for the sequence-only ***I**mplicit **S**tructure **M**odel* (*ISM*).

4.1 Atomic Autoencoder

Atomic Autoencoder uses an encoder-decoder architecture with a Graph Transformer encoder and a plain transformer decoder. The encoder takes the masked microenvironment $\boldsymbol{x}^l_{\text{masked-microenv}}$ as input and produces atomic representations $\{\boldsymbol{z}^l_1, ..., \boldsymbol{z}^l_n\}$. The decoder takes atomic representations in and produces features $\{\boldsymbol{f}^l_1, ..., \boldsymbol{f}^l_n\}$ which linearly project to atomic coordinates $\{\hat{\boldsymbol{c}}^l_1, ..., \hat{\boldsymbol{c}}^l_n\}$ (See Figure 2). This might seem like a trivial task, after all the inputs $\boldsymbol{x}^l_{\text{masked-microenv}}$ contain the regression targets. However, since the Graph Transformer only uses relative positions, and only in an attention bias B^l, the prediction tasks are quite difficult and require reasoning about the local structure of the microenvironment.

To obtain a residue-level feature representation, we average the atom-level features of the Graph Transformer $\boldsymbol{z}^l = \frac{1}{n}\sum_i \boldsymbol{z}^l_i$ following Diaz et al. (2024). To train this representation, we add $\boldsymbol{z}^l$ into all atomic representations before the decoder. Mathematically, the transformer decoder takes $\{\boldsymbol{z}^l_1 + \boldsymbol{z}^l, ..., \boldsymbol{z}^l_n + \boldsymbol{z}^l\}$ as input. We also found that adding this $\boldsymbol{z}^l$ directly to the decoder architecture improves training stability. See Figure 5 for full architecture.

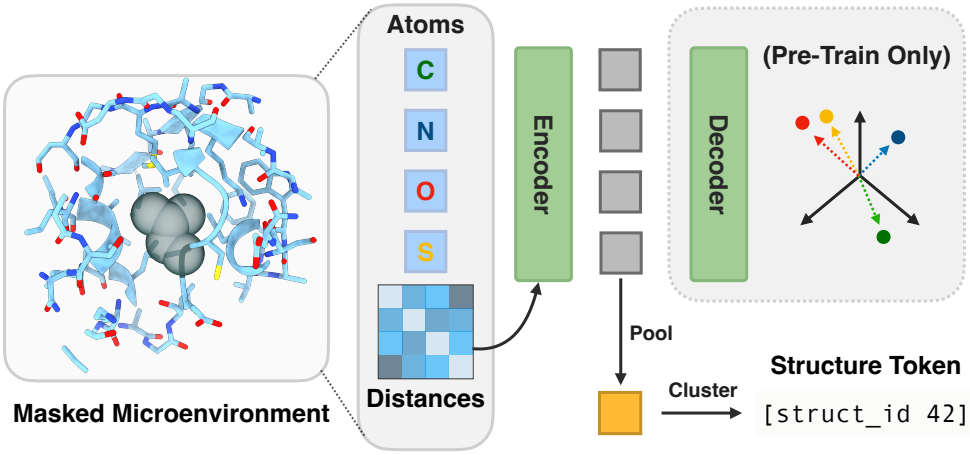


Figure 2: **Atomic Autoencoder learns a structural representation of a residue's microenvironment.** The Autoencoder takes atom element types and pairwise distances as input and reconstructs all atomic coordinates. The encoder is a graph transformer that uses the pairwise distances to bias the attention mechanism to learn rich atomic representations. The atomic representations are pooled to form a microenvironment embedding. The decoder takes the atomic representations and microenvironment embedding as input to decode the coordinates for all atoms. The learned microenvironment embeddings are discretized via K-means into structure tokens, which supervise the fine-tuning of a protein language model. See Figure 5 for architectural details.

Training objective. One major challenge is that microenvironments lack robust protein backbone coordinate frames that underpin full protein models (Jumper et al., 2021; Hayes et al., 2024; Dauparas et al., 2022). Unsurprisingly, we empirically observe that vanilla MSE loss $\mathcal{L}^l_{\text{MSE}} = \frac{1}{n}\sum_i \|\hat{\boldsymbol{c}}^l_i - \boldsymbol{c}^l_i\|$ does not take the coordinate frame into account and overestimates the loss. Thus, we optimize the MSE loss after global alignment. First, we employ the Kabsch algorithm (Kabsch, 1976; Umeyama, 1991) to analytically compute the rotation and translation that minimize MSE loss. Then the loss is calculated using the transformed ground truth coordinates. Formally,

$$\mathcal{L}^l_{\text{MSE-aligned}} = \min_{\boldsymbol{R}\in SE(3),\boldsymbol{T}\in\mathbb{R}^3} \frac{1}{n}\sum_i \|\hat{\boldsymbol{c}}^l_i - (\boldsymbol{R}\boldsymbol{c}^l_i + \boldsymbol{T})\|.$$

During training, we observe that naive optimization of the MSE-aligned loss results in convergence to a local optimum where all predicted coordinates lie on a 2-dimensional plane. Following AlphaFold (Jumper et al., 2021), we addressed the issue using a distogram loss. Here, we use ESM3's distogram head by first computing $\boldsymbol{f}^l_{ij} = \boldsymbol{W}_a\boldsymbol{f}^l_i - \boldsymbol{W}_b\boldsymbol{f}^l_j$, where $\boldsymbol{W}_a, \boldsymbol{W}_b$ are linear adapters. We then apply a binned distance loss

$$\mathcal{L}^l_{\text{disto}} = \frac{1}{n^2}\sum_{i,j} \ell_{\text{CE}}(\boldsymbol{C}^T_{\text{disto}}\boldsymbol{f}^l_{ij}, d^{\text{bin},l}_{ij}).$$

where $\boldsymbol{C}_{\text{disto}}$ is a linear classification head that predicts the distance bin $d^{\text{bin},l}_{ij}$ between atoms i and j at residue position l. During the first stage of training, we train with the distogram and masked modeling losses, $\mathcal{L}^l_{disto} + \mathcal{L}^l_{\text{AA}}$. During the second stage, we additionally include $\mathcal{L}^l_{\text{MSE-aligned}}$.

Generating Structure Tokens. Given a protein structure $\boldsymbol{x}_{\text{struct}}$, we start by generating the masked microenvironment for all L residues, namely $(\boldsymbol{x}^1_{\text{masked-microenv}}, ..., \boldsymbol{x}^L_{\text{masked-microenv}})$. We feed each masked microenvironment into our Graph Transformer encoder to extract a residue-level feature representation at each position, $(\boldsymbol{z}^1, ..., \boldsymbol{z}^L)$. We quantize $\boldsymbol{z}^l$ for every residue in the protein using K-means (Lloyd, 1982) to generate a structure sequence $\boldsymbol{s} = (s_1, ..., s_L)$. In addition to our autoencoder, we also extract features $(\boldsymbol{z}^{1\prime}, ..., \boldsymbol{z}^{L\prime})$ from MutRank (Gong et al., 2024) and generate a second structure sequence $\boldsymbol{s}' = (s'_1, ..., s'_L)$, both of which are used identically to fine-tune the protein sequence model. Both models are trained on a smaller dataset of experimental structures and are used to generate structure tokens on a large dataset of AlphaFold structures.

Table 1: **Comparisons on structural benchmarks.** We freeze all protein models to assess the learned representation. *ISM* is structure-tuned on the AlphaFold structures of Uniclust30 while *ISM*† undergoes additional structure-tuning on PDB structures. SaProt* takes the protein structure as input. All other methods take a sequence as their only input. For contact, secondary structure, and binding residue prediction, the proteins in the training and test sets have at most 30% sequence similarity.

	Strucure Prediction (CAMEO)			Contact			SS	Binding	
Method	GDT-TS	GDT-HA	LDDT	Short	Med	Long	Acc	F1	MCC
Evolutionary pLM									
Amplify (Fournier et al., 2024)	-	-	-	0.38	0.36	0.23	0.82	0.22	0.26
ESM2 (Lin et al., 2022)	0.64	0.47	0.82	0.45	0.45	0.35	0.86	0.31	0.34
ESM2 (fine-tuned)	0.64	0.47	0.82	0.45	0.45	0.35	0.86	0.32	0.34
Structural pLM									
ESM2-S (Zhang et al., 2024)	0.61	0.43	0.79	0.46	0.47	0.36	0.85	0.32	0.35
S-PLM (Wang et al., 2023)	0.61	0.44	0.80	0.48	0.49	0.36	0.86	0.29	0.32
SaProt* (Su et al., 2023)	-	-	-	0.57	0.53	0.48	0.86	0.36	**0.38**
ISM (Ours)	**0.67**	**0.50**	0.83	0.61	**0.60**	**0.49**	**0.89**	0.35	0.37
ISM † (Ours)	**0.67**	**0.50**	**0.84**	**0.62**	**0.60**	0.48	**0.89**	**0.37**	**0.38**

4.2 Structure-Tuning the Protein Sequence Model

We initialize a sequence-only protein language model trained using masked language modeling (*i.e.,* ESM2) and fine-tune it to predict the structure tokens. We call this training **structure-tuning** and the resulting model ***I**mplicit **S**tructure **M**odel* (*ISM*). We append a linear classification head $\boldsymbol{C}_{\text{struct}}$ to the output of the $\mathbf{pLM}$ backbone to predict the structural token. The structure prediction loss function is

$$\mathcal{L}_{\text{Struct}} = \frac{1}{|\mathbb{S}|} \sum_{i \in \mathbb{S}} \ell_{\text{CE}}(\boldsymbol{C}_{\text{struct}}^{\top} \mathbf{pLM}(\tilde{\boldsymbol{x}}_{\text{seq}})_i, s_i),$$

where $\tilde{\boldsymbol{x}}_{\text{seq}}$ is the amino acid sequence with masked residues, $\mathbf{pLM}$ is the protein language model backbone, $\mathbf{pLM}(\tilde{\boldsymbol{x}}_{\text{seq}})_i$ is the representation for residue i, s_i is the structure token at residue i, and $\mathbb{S}$ are the positions at which the loss is computed. In standard MLM, the loss is computed for all masked positions (*i.e.,* $\mathbb{S} = \mathbb{M}$). We found that predicting structure tokens at *all* positions (*i.e.,* $\mathbb{S} = \{1, ..., L\}$), and not just masked positions, better distills structural representations.

We structure-tune our model on AlphaFold protein structures. AlphaFold sometimes produces inaccurate structures with poorly folded areas showing few interactions. Our structure token visualization reveals that many of these problematic residues are grouped into a single token s^* (`[struct id 17]` in Figure 3). To ensure data quality, we exclude microenvironments assigned the s^* token from sequence model training. We compute $\mathcal{L}_{\text{Struct}}$ at positions $\mathbb{S} = \{i : i \in \{1, ..., L\}$ and $s_i \neq s^*\}$.

The final training objective for structure-tuning is the sum of structure token(s) and amino acid cross-entropy losses (see Section 3.1), namely $\mathcal{L} = \mathcal{L}_{\text{Struct}} + \mathcal{L}_{\text{MLM}}$.

5 Results

5.1 Implementation Details

Atomic Autoencoder. Our microenvironment-based Atomic Autoencoder is a Graph Transformer encoder with 4 layers and a vanilla Transformer decoder with 2 layers. Our autoencoder training dataset contains 35K proteins from the Protein Data Bank(PDB). We train both stages for 5 epochs with a learning rate of 1×10^{-3}. See Table 5a for a list of hyperparameters.

Distillation Dataset. Once our autoencoder is fully trained, we extract per-residue microenvironment features for 5.8M proteins from Uniclust30 with AlphaFold structures (Mirdita et al., 2017), along with 35K PDB proteins. We identify cluster centroids by applying K-means clustering to features from the PDB database, then assign features to tokens based on their distances to these centroids. The number of clusters, $K = 64$, is chosen using the elbow method. Additionally, we extract per-residue microenvironment features from MutRank and cluster the features into one of $K = 512$ tokens (see Section 3.2).

Table 2: **Comparisons on S669 Single Mutation Thermodynamic Stability prediction.** We compare *ISM* to state-of-the-art methods that take various modalities as input. The middle and bottom block approaches are fine-tuned on cDNA117K, which consists of mini-proteins that have at most 30% sequence similarity with those in S669. UR50: UniRef-50 used in ESM2 pretraining, UR100: UniRef-100, PDB: Protein data bank, UC30: Uniclust30. OAS: Observed Antibody Space. SCOP Structural Classification of Proteins. r_s: Spearman correlation coefficient.

Method	PreTrain Data	r_s	AUC	MCC	RMSE$_\downarrow$
FoldX (Schymkowitz et al., 2005)	N/A	0.27	0.62	0.14	2.35
PROSTATA (Umerenkov et al., 2022)	UR-50	0.50	0.73	0.28	1.44
Amplify (Fournier et al., 2024)	UR100,OAS,SCOP	0.42	0.66	0.21	1.52
S-PLM (Wang et al., 2023)	UR50,SwissProt	0.41	0.68	0.18	1.53
Stability Oracle (Diaz et al., 2024)	PDB	0.53	0.75	0.34	1.44
MutateEverything (ESM) (Ouyang-Zhang et al., 2024)	UR-50	0.47	0.72	0.31	1.48
MutateEverything (AF) (Ouyang-Zhang et al., 2024)	PDB	**0.56**	**0.76**	0.35	**1.38**
ESM (fine-tuned)	UR-50,PDB+UC30	0.49	0.72	0.25	1.47
ISM	UR50,UC30	0.49	0.73	0.33	1.47
ISM	UR50,PDB	0.52	0.74	0.30	1.45
ISM (Ours)	UR50,PDB+UC30	0.53	**0.76**	**0.40**	1.44

Structure-tuning. We structure-tune the 650M parameter ESM2 for 20 epochs using a cosine learning rate schedule with 4 warmup epochs. We use a total batch size of 1536 proteins cropped to a maximum sequence length of 512 amino acids. We use AdamW optimizer with a learning rate of 1×10^{-4} and weight decay of 5×10^{-3}. Training takes 26 wall-clock hours on 32 GH200 GPUs. See Table 5b for a complete list of hyperparameters.

5.2 Comparisons on structure tasks

Rich sequence representations should inherently capture a protein's fold. In Table 1, we evaluate the structure-enriched representation of *ISM* against established methods on several structure-based downstream tasks, including structure, contact, secondary structure, and binding residue prediction. We evaluate all models as frozen feature extractors and learn a decoding head. For structure prediction, we initialize from pre-trained SoloSeq (Ahdritz et al., 2022), replace the ESM2 backbone model with a frozen protein model, and tune the folding head. For other downstream tasks, we freeze the backbone model and train a shallow head. Contact, secondary structure, and binding residue prediction are evaluated using sequence similarity splits of 30%, 25%, and 20% respectively. More dataset descriptions are listed in Section C. ESM (fine-tuned) follows the same training regimen as *ISM*, but is trained only with masked language modeling. We report results for *ISM* trained on Uniclust30 alone and Uniclust30+PDB.

Our model outperforms all sequence-only models and matches structure-sequence models on all structure-based benchmarks. Notably, on long-range contact prediction, *ISM* outperforms ESM2 by 40%, with a precision of 0.49 against 0.35. This matches the performance of SaProt (0.48), which explicitly requires the structure as input while *ISM* is a sequence-only model. On structure prediction, *ISM* outperforms ESM2 by 5% on the GDT-TS metric (0.67 vs 0.64). On binding residue prediction F1 metric, *ISM* performs similarly with SaProt's 0.36, achieving 0.35 when trained on Uniclust30 and 0.37 when trained on Uniclust30+PDB. Overall, the structure-enriched representations of *ISM* improve performance on various structure-based downstream tasks compared to sequence-only pLMs and structural pLMs.

5.3 Comparisons on Mutation Stability effect

Thermodynamic stability is an important phenotype that often needs to be improved during the engineering of a commercially viable protein (Diaz et al., 2023; Liu et al., 2024; Carceller et al., 2024). We evaluate how effectively *ISM* predicts the impact of single mutations on a protein's thermodynamic stability ($\Delta\Delta G$) on the S669 dataset (Pancotti et al., 2022) in Table 2. We evaluate all pLMs (ESM, Amplify, S-PLM, *ISM*) identically by fine-tuning the model with a shallow decoder head as in MutateEverything (Ouyang-Zhang et al., 2024). We fine-tune on the cDNA117K dataset

Table 3: **System-level Comparisons of transfer learning to various functional benchmarks.** We fine-tune all models with a shallow head for each benchmark (except HumanPPI, in which we freeze *ISM* due to overfitting). * reports the best checkpoint found during training.

Method	Thermostability	HumanPPI	Metal Bind	EC	GO			DeepLoc	
					MF	BP	CC	Subcell.	Binary
	Spearman ρ	Acc	Acc	Fmax	Fmax	Fmax	Fmax	Acc	Acc
ESM1b	0.71	0.82	0.74	0.87	0.66	0.45	0.47	0.80	0.92
MIF-ST	0.69	0.76	0.75	0.81	0.63	0.38	0.32	0.79	0.92
ESM2*	0.70	0.88	0.74	0.87	**0.67**	**0.49**	0.51	**0.85**	**0.94**
SaProt*	**0.72**	0.88	**0.79**	**0.88**	0.65	**0.49**	0.51	**0.85**	0.93
ISM *	0.71	**0.89**	0.75	**0.88**	**0.67**	0.47	**0.52**	0.84	0.93

from Diaz et al. (2024), a subset of the cDNA display proteolysis dataset (Tsuboyama et al., 2023) where all proteins have at most 30% sequence similarity to those in S669.

ISM outperforms all existing models that take a single sequence as input, achieving a Spearman correlation of 0.53 compared to Mutate Everything (ESM)'s 0.49, and an AUC of 0.76 compared to Mutate Everything (ESM)'s 0.72. Additionally, *ISM* matches the performance of state-of-the-art models while only using the amino acid sequence input, achieving an AUC of 0.76, while Mutate Everything (AF) and Stability Oracle achieve AUCs of 0.76 and 0.75, respectively. Note that Stability Oracle (Diaz et al., 2024) takes the atomic microenvironment as input and Mutate Everything (AF) (Ouyang-Zhang et al., 2024) takes a multiple sequence alignment as input.

We conducted an ablation study on the datasets used for structure-tuning and were surprised to find that training on the smaller PDB dataset enhances downstream $\Delta\Delta G$ performance more than training on the larger Uniclust30 dataset. Specifically, *ISM* achieves a Spearman correlation of 0.49 when trained on UniClust30, compared to 0.52 when trained on PDB. Even though the supervision signal during structure-tuning is derived solely from the atomic coordinates in the structure and not $\Delta\Delta G$ labels, we suspect the PDB dataset has some overlap with the structures in the S669 dataset, resulting in performance similar to that of structure-input models. Overall, on the S669 $\Delta\Delta G$ test set, *ISM* is competitive and even outperforms SOTA structure-based methods and AlphaFold's representations, a feat sequence-only pLMs have yet to achieve.

5.4 Comparisons on a diverse set of functional phenotypes

Functional characterization of proteins through biochemical techniques is typically the most resource-intensive type of labeled data to generate, making accurate transfer learning predictions particularly valuable for downstream bioinformatics and protein engineering and design tasks (Yu et al., 2023; Allman et al., 2024; Kulikova et al., 2021). In Table 3, we evaluate *ISM* on the PEER (Xu et al., 2022) and FLIP (Dallago et al., 2021) benchmarks, which encompass tasks that benefit from structural representations (*e.g.,* thermostability), evolutionary representations (*e.g.,* biological process), or both (*e.g.,* EC). We fine-tune all models with a shallow readout head on all benchmarks, except HumanPPI, for which we perform linear probing on *ISM* to prevent overfitting. We observed that longer training leads to overfitting, therefore, we evaluate various training checkpoints and report the highest performance for ESM2, SaProt, and *ISM*. ESM1b (Rives et al., 2019) and MIF-ST (Yang et al., 2023) results are sourced from SaProt (Su et al., 2023).

ISM performance remains competitive with ESM2 and other pLMs on functionally diverse tasks and does not stand out. For example, for predicting gene ontology - molecular function (GO-MF), both *ISM* and ESM2 achieve 67% accuracy while SaProt achieves 65%. This finding aligns with prior work (Hu et al., 2022), which demonstrates that ESM2 outperforms Evoformer, the feature extractor for AlphaFold, on some functional tasks. It seems that for these functional tasks, the evolutionary signal from masked language modeling is sufficient and does not necessarily benefit from AlphaFold representations. Nonetheless, these experiments demonstrate that the structure-enriched representations of *ISM* do not corrupt ESM2's evolutionary representation on various function-based downstream tasks while enhancing ESM2's structural understanding.

Table 4: ***ISM* ablation experiments.** Default settings are marked in grey. See Section 6.1. ss: Secondary Structure prediction, mc: MutCompute, mr: MutRank, ae: Atomic Autoencoder

(a) **Other Structure Tokens**

tokenizer	contact	ss	bind
foldseek	0.42	0.88	0.32
esm3	0.18	0.85	0.11
mc+mr	0.45	0.88	0.36
ae+mr	**0.48**	**0.89**	**0.37**

(b) **Our Structure Tokens**

tokenizer	contact	ss	bind
ae	0.38	0.88	0.35
mr	0.46	0.88	0.34
ae+mr	**0.48**	**0.89**	**0.37**

(c) **Number of clusters**

K	contact	ss	bind
32	0.27	0.84	0.33
64	**0.48**	**0.89**	**0.37**
128	0.42	0.85	**0.37**

(d) **Pre-training Crop length**

crop	val acc	contact
32	0.27	0.27
128	0.36	0.42
512	**0.40**	**0.48**

(e) **Label Type**

label	contact	r_s ($\Delta\Delta G$)
features	0.36	0.49
tokens	**0.46**	**0.51**

(f) **Initialization**

init	val acc	contact
random	0.36	0.10
esm2	**0.40**	**0.48**

6 Analysis

6.1 Ablations

We ablate key design decisions by reporting long-range Precision at L (P@L) for contact prediction, accuracy for secondary structure prediction, F1 for binding residue prediction, and Spearman correlation for $\Delta\Delta G$ prediction in Table 4. We also report the validation accuracy, indicating how often the *ISM* variant correctly predicts the structure token derived from Atomic Autoencoder.

Structure Tokens. In Table 4a, we distill from various structure models from the literature. We compare against a variant using both MutComputeX-GT (mc) and MutRank (mr) structure models. Since Atomic Autoencoder uses the MLM loss $\mathcal{L}^l_{\text{AA}}$ from MutComputeX-GT, this variant determines the effect of dropping the autoencoder from structure-tuning *ISM*. Our model outperforms MutRank and MutComputeX-GT, indicating that the autoencoder provides important structural information.

We found that structure-tuning with ESM3's VQVAE (Hayes et al., 2024) structure tokens do not produce robust structural representations. A model structure-tuned with ESM3 achieves 0.18 and 0.11 on contact and binding residue prediction, compared to 0.48 and 0.37 for *ISM*, respectively. We observe that the accuracy of ESM3 structure token prediction on a held-out validation accuracy on UniClust30 is $\sim$8%, while Atomic Autoencoder accuracy is $\sim$40% and MutRank accuracy is $\sim$47%. We suspect that the large vocabulary of ESM3's VQVAE (4096 structure tokens) results in redundant and overlapping tokens that are difficult to discern and complicate loss optimization.

We also evaluate the performance of our sequence model structure-tuned on FoldSeek VQVAE structure tokens (van Kempen et al., 2022). We train on a larger subset of UniClust30 obtained from SaProt (Su et al., 2023) for the same number of iterations as in *ISM*. The model achieves a long-range contact P@L of 0.42 and a binding residue F1 score of 0.32, which are improvements over ESM3 structure tokens and surpasses the ESM2 baseline (F1 scores of 0.35 and 0.31, respectively). However, representations learned from FoldSeek's VQVAE structure tokens lag behind *ISM* (0.48 and 0.37). Thus, the structure tokens from Atomic Autoencoder and MutRank produce better structure representations, their combination being the most effective (see Table 4b).

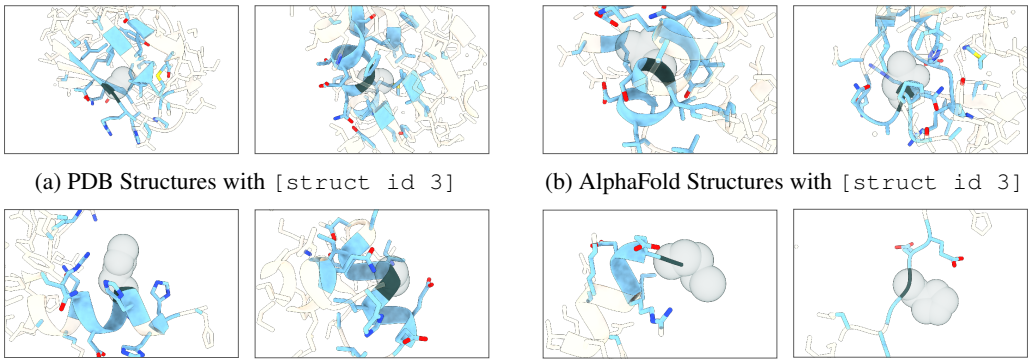
Training parameters. We evaluate how the maximum length of a sequence during structure-tuning affects the accuracy and downstream performance in Table 4d. When the crop length is dropped to 128 and 32 amino acids, the contact long-range P@L drops from 0.48 to 0.42 and 0.27 respectively. This shows that training with longer sequences is essential for learning long-range contacts.

Additionally, we evaluate the effectiveness of clustering MutRank representations $\boldsymbol{z} = (\boldsymbol{z}^1, ..., \boldsymbol{z}^L) \in \mathbb{R}^{L \times D}$ into tokens $\boldsymbol{s} = (s_1, ..., s_L) \in \{1, ..., K\}^L$ in Table 4e (excluding Atomic Autoencoder supervision). Our model variant uses a linear head to predict the MutRank representations $\boldsymbol{z}$ and is trained with normalized MSE loss. Direct MutRank representation prediction achieves 0.36 P@L, while token ID prediction reaches 0.46 P@L on long-range contact prediction. Clustering the MutRank representations potentially removes superfluous high-frequency noise.

Evolutionary Pre-Training. We evaluate the significance of training with MLM before structure tuning in Table 4f by initializing with random weights. This approach resulted in decreased accuracy of structure tokens from 40% to 36%. On downstream contact prediction, training from scratch drops long-range P@L from 0.48 to 0.1. This highlights the importance of structure-tuning a pretrained ESM2 as opposed to structure-tuning from scratch.

6.2 Qualitative Visualizations

In Figure 3, we visualize atomic structures of microenvironments grouped by structure token id. Specifically, we examine tokens `[struct id 3]` and `[struct id 17]`, which are the least and most frequently observed tokens in Uniclust30, respectively. We find that microenvironments of the same structure token are semantically related. For example, `[struct id 3]` contains semi-exposed residues. Interestingly, `[struct id 17]` includes both solvent-exposed residues from experimental structures and unfolded residues from AlphaFold structures. These findings motivate us to exclude `[struct id 17]` from our structure-tuning training objective (see Section 4.2). Additional visualizations and analysis are provided in Section D.



(a) PDB Structures with `[struct id 3]`

(b) AlphaFold Structures with `[struct id 3]`

(c) PDB Structures with `[struct id 17]`

(d) AlphaFold Structures with `[struct id 17]`

Figure 3: **Cluster-based Microenvironment Visualizations**. Residues in sky blue are within the microenvironment, while white residues are outside and included for context. The grey density indicates the masked-out amino acid. Nodes are colored by element: blue for nitrogen, red for oxygen, and yellow for sulfur. The left two columns display structures from the PDB, while the right two columns show protein sequences from Uniclust30, folded using AlphaFold. `[struct id 3]` contains semi-solvent exposed residues. `[struct id 17]` contains solvent exposed residues.

6.3 Runtime

We compare our runtime against SaProt (Su et al., 2023) on three proteins with 91, 355, and 689 amino acids. The transformer is run on an A40 GPU. Colabfold structure prediction (Mirdita et al., 2022) dominates the runtime. Even with structures, the *ISM* runs 2.4× faster than SaProt which additionally runs FoldSeek (van Kempen et al., 2022) to tokenize the structure.

	SaProt	*ISM* (Ours)
ColabFold	418 s	-
FoldSeek	43 ms	-
Transformer	28 ms	28ms

Figure 4: **Runtime comparison.**

7 Discussion

In this paper, we trained *ISM*, a protein language model with enriched structure representations while not requiring explicit structural coordinates and complex data engineering pipelines during inference. *ISM* achieves this with structure-tuning: a multi-modal fine-tuning paradigm that distills the representation of the tertiary structure surrounding a residue from a structure-based model into a sequence-based model. Structure-tuning augments the standard masked language modeling (MLM) loss with additional cross-entropy losses where the labels are structure tokens derived from discretizing the embeddings of structure-based model(s).

Here, we explored structure-tuning ESM2 with structure tokens derived from FoldSeek VQVAE, ESM3 Structural VQVAE, MutComputeX, and MutRank structure models (see Section 6.1). Additionally, we developed Atomic Autoencoder to bridge the gap between current VQVAEs and masking-based self-supervised frameworks. Atomic AutoEncoder learns richer all-atom structural details than existing backbone-based VQVAEs and the reconstruction loss prioritizes learning patterns

within the entire all-atom local tertiary structure compared to traditional self-supervised masked modeling techniques.

ISM demonstrates enhanced structural understanding by achieving state-of-the-art performance on both downstream structure prediction tasks (Table 1) and mutational tasks (*i.e.,* $\Delta\Delta G$ prediction in Table 2) when structure-tuned with both Atomic Autoencoder and MutRank structure tokens. *ISM* maintains performance on tasks that do not appear to benefit from structure-enriched representations. Thus, we believe *ISM*'s structurally-enriched representations will benefit other protein applications where structure is important without compromising performance in other downstream tasks.

We observe the best structure-tuning performance when training with structure tokens from Atomic Autoencoder and MutRank. Both Atomic Autoencoder and MutRank structure models are all-atom microenvironment-based graph transformers trained on a sequence-balanced dataset of experimental structures. They differ in their training objective – reconstruction vs EvoRank. The trained models were run on 5.8M AlphaFold structures from UniClust30 and their representations were discretized into tokens. During this work, we observed sufficient generalization from experimental to computational structures for the structure-tuning of *ISM*. However, upon visualizing a token's microenvironments in experimental and computational structures, we noticed subtle distribution shifts that are primarily due to AlphaFold artifacts (see Section D for detailed analysis). We hypothesize that training the Atomic Autoencoder and MutRank on computational structures will enable further downstream tasks.

Although structure-tuning can be applied to any pre-trained pLM, we built *ISM* on ESM2 due to its popularity in the protein-ML and bioinformatics communities. Thus, *ISM* uses the exact same architecture and interface as ESM2, making it a drop-in replacement to all frameworks built on ESM2. To use *ISM* instead of ESM2, end users require a single line of code: `model.load_state_dict(torch.load("/path/to/ism/weights.pth"))`. We make *ISM*'s weights available for both the 650M and 3B parameter ESM2 models: https://github.com/jozhang97/ISM. All results presented in this manuscript are based on the 650M parameter *ISM* model.

8 Acknowledgements

This work is supported by the NSF AI Institute for Foundations of Machine Learning (IFML) and UT-Austin Center for Generative AI. We would like to thank AMD for the donation of computational hardware and support resources from its HPC Fund. We acknowledge the Texas Advanced Computing Center at The University of Texas at Austin for providing computational resources (Vista cluster) to support this work.

References

Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, Arkadiusz Nowaczynski, Bei Wang, Marta M Stepniewska-Dziubinska, Shang Zhang, Adegoke Ojewole, Murat Efe Guney, Stella Biderman, Andrew M Watkins, Stephen Ra, Pablo Ribalta Lorenzo, Lucas Nivon, Brian Weitzner, Yih-En Andrew Ban, Peter K Sorger, Emad Mostaque, Zhao Zhang, Richard Bonneau, and Mohammed AlQuraishi. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *bioRxiv*, 2022. doi: 10.1101/2022.11.20.517210. 7, 16

Brent Allman, Luiz Vieira, Daniel J Diaz, and Claus O Wilke. A systematic evaluation of the language-of-viral-escape model using multiple machine learning frameworks. *bioRxiv*, 2024. 1, 8

Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096): 223–230, 1973. 1

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8), 2022. 1, 2

Jose M Carceller, Bhumika Jayee, Claire G Page, Daniel G Oblinsky, Gustavo Mondragón-Solórzano, Nithin Chintala, Jingzhe Cao, Zayed Alassad, Zheyu Zhang, Nathaniel White, Daniel J Diaz,

Andrew D Ellington, Gregory D Scholes, Sijia S Dong, and Todd K Hyster. Engineering a photoenzyme to use red light. *Chem*, 2024. 7

Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021–11, 2021. 8

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(6615):49–56, 2022. 5

D. J. Diaz, C. Gong, J. Ouyang-Zhang, J. M. Loy, J. Wells, D. Yang, A. D. Ellington, A. G. Dimakis, and A. R. Klivans. Stability oracle: A structure-based graph-transformer for identifying stabilizing mutations. *Nature Communications*, 15:6170, 2024. doi: 10.1038/s41467-024-49780-2. 4, 7, 8

Daniel J Diaz, Anastasiya V Kulikova, Andrew D Ellington, and Claus O Wilke. Using machine learning to predict the effects and consequences of mutations in proteins. *Current Opinion in Structural Biology*, 78, 2023. 7

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021. 1, 2

Quentin Fournier, Robert M Vernon, Almer van der Sloot, Benjamin Schulz, Sarath Chandar, and Christopher James Langmead. Protein language models: Is scaling necessary? *bioRxiv*, pp. 2024–09, 2024. 6, 7, 17

Daria Frolova, Marina Pak, Anna Litvin, Ilya Sharov, Dmitry Ivankov, and Ivan Oseledets. Mulan: Multimodal protein language model for sequence and structure encoding. *bioRxiv*, pp. 2024–05, 2024. 1, 3

Benoit Gaujac, Jérémie Donà, Liviu Copoiu, Timothy Atkinson, Thomas Pierrot, and Thomas D Barrett. Learning the language of protein structure. *arXiv preprint arXiv:2405.15840*, 2024. 1, 3

Chengyue Gong, Adam Klivans, James Madigan Loy, Tianlong Chen, Daniel Jesus Diaz, et al. Evolution-inspired loss functions for protein representation learning. In *Forty-first International Conference on Machine Learning*, 2024. 2, 4, 5

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*, pp. 2024–07, 2024. 2, 3, 5, 9

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *bioRxiv*, pp. 2023–07, 2023. 3

Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems*, 35:38873–38884, 2022. 3, 8

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. 3, 5

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 5

Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Soenderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6): 520–527, 2019. 16

Anastasiya V Kulikova, Daniel J Diaz, James M Loy, Andrew D Ellington, and Claus O Wilke. Learning the local landscape of protein structures with convolutional neural networks. *Journal of Biological Physics*, 47(4):435–454, 2021. 8

Anastasiya V Kulikova, Daniel J Diaz, Tianlong Chen, T Jeffrey Cole, Andrew D Ellington, and Claus O Wilke. Two sequence-and two structure-based ml models have learned different aspects of protein biochemistry. *Scientific Reports*, 13(1):13280, 2023. 1

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Liang Hong, and Pan Tan. Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv*, pp. 2024–04, 2024. 1, 3

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. 1, 2, 6, 17

Maria Littmann, Michael Heinzinger, Christian Dallago, Konstantin Weissenow, and Burkhard Rost. Protein embeddings and deep learning predict binding residues for various ligand classes. *Scientific Reports*, 11(1):23916, 2021. 17

Yi Liu, Sophie G Bender, Damien Sorigue, Daniel J Diaz, Andrew D Ellington, Greg Mann, Simon Allmendinger, and Todd K Hyster. Asymmetric synthesis of α-chloroamides via photoenzymatic hydroalkylation of olefins. *Journal of the American Chemical Society*, 146(11), 2024. 7

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982. 5

Milot Mirdita, Lars Von Den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017. 6

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022. 10

Jeffrey Ouyang-Zhang, Daniel Diaz, Adam Klivans, and Philipp Krähenbühl. Predicting a protein's stability under a million mutations. *Advances in Neural Information Processing Systems*, 36, 2024. 7, 8

Corrado Pancotti, Silvia Benevenuta, Giovanni Birolo, Virginia Alberini, Valeria Repetto, Tiziana Sanavia, Emidio Capriotti, and Piero Fariselli. Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset. *Briefings in Bioinformatics*, 23(2):bbab555, 2022. 7

Daniel Peñaherrera and David Ryan Koes. Structure-infused protein language models. *bioRxiv*, 2023. 3

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019. 16

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. 1, 2, 8

Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005. 7

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, 2023. 1, 3, 6, 8, 9, 10, 16, 17

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 2023. doi: 10.1038/s41586-023-06328-6. URL https://doi.org/10.1038/s41586-023-06328-6. 8

Dmitriy Umerenkov, Tatiana I Shashkova, Pavel V Strashnov, Fedor Nikolaev, Maria Sindeeva, Nikita V Ivanisenko, and Olga L Kardymon. Prostata: Protein stability assessment using transformers. *bioRxiv*, pp. 2022–12, 2022. 7

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 5

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*, pp. 2022–02, 2022. 3, 9, 10

Duolin Wang, Mahdi Pourmirzaei, Usman L Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, et al. S-plm: Structure-aware protein language model via contrastive learning between sequence and structure. *bioRxiv*, pp. 2023–08, 2023. 3, 6, 7, 17

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022. 8, 16

Jianyi Yang, Ambrish Roy, and Yang Zhang. Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research*, 41(D1):D1096–D1103, 2012. 17

Kevin K Yang, Niccolò Zanichelli, and Hugh Yeh. Masked inverse folding with sequence transfer for protein representation learning. *Protein Engineering, Design and Selection*, 36, 2023. 1, 8

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021. 4

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023. 8

Zuobai Zhang, Jiarui Lu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Structure-informed protein language model. *arXiv preprint arXiv:2402.05856*, 2024. 1, 2, 6

A Atomic Autoencoder Architecture Details

In Figure 5, we visualize the details of our Atomic Autoencoder architecture. We use a GraphTransformer encoder and a vanilla transformer decoder.

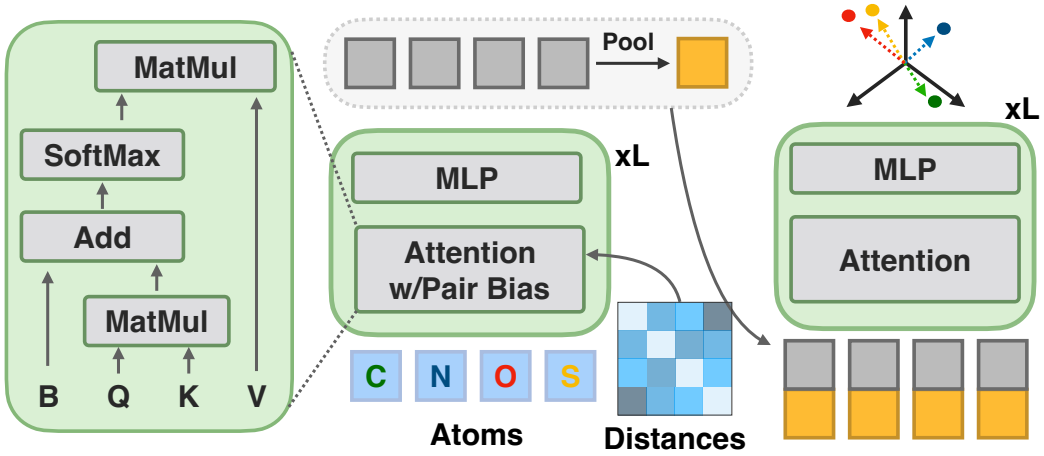


Figure 5: **Atomic Autoencoder Architecture Details.** The autoencoder takes atom element types and pairwise distances as input and reconstructs all atomic coordinates. The encoder is a graph transformer that uses the pairwise distances to bias the attention mechanism to learn rich atomic representations. The atomic representations are pooled to form a microenvironment embedding. The decoder takes the atomic representations and microenvironment embedding as input and produces coordinates for each atom. The learned microenvironment embeddings are discretized via K-means into structure tokens, which supervise the fine-tuning of a protein language model.

B Atomic Autoencoder Training and *ISM* Structure-tuning

Table 5 lists the hyperparameters used for training the Atomic Autoencoder (see Section 4.1) and structure-tuning the PLM (see Section 4.2).

Table 5: **Model Hyperparameters.**

(a) **Atomic Autoencoder Training**

Hyperparameter	Stage 1	Stage 2
optimization		
total batch size	2048	2048
optimizer	AdamW	AdamW
learning rate	1e-3	1e-3
weight decay	1e-5	1e-5
epochs	5	5
warmup epochs	1	1
clip max norm	1.0	1.0
modeling		
layers	4	4
max atoms	512	512
max atom distance	10.0	10.0
losses		
λ_{AA}	1.0	1.0
$\lambda_{\text{Distogram}}$	1.0	1.0
$\lambda_{\text{MSE-aligned}}$	0	1.0
number of GPUs	8	8
runtime	∼12hr	∼12hr

(b) **Protein Language Model Structure-tuning**

Hyperparameter	Structure-tuning
optimization	
total batch size	1536
optimizer	AdamW
learning rate	1e-4
weight decay	5e-3
epochs	20
warmup epochs	4
clip max norm	5.0
modeling	
layers	33
mask ratio	15%
crop length	512
losses	
λ_{MLM}	1.0
λ_{struct1}	1.0
λ_{struct2}	1.0
number of GPUs	32
runtime	26hr

Table 6: **Structural Dataset Statistics.** We report the primary metrics and number of proteins. The split similarity is the maximum allowed sequence similarity between any protein in the training set and any protein in the validation or test sets.

Dataset	Metrics	Train	Valid	Test	Split Similarity
Structure Prediction	GDT-TS	121,481	-	185	-
Contact Prediction	Long Range Precision	25,299	224	40	30%
Secondary Structure Prediction	Accuracy	8,678	2170	513	25%
Binding Residue Prediction	F1	1,014	-	300	20%

Table 7: **Hyperparameters on downstream structural benchmarks.** *: we find that training converges and terminate training early.

Hyperparameter	Structure	Contact	Secondary Structure	Binding Residues
optimization				
total batch size	128	16	16	32
optimizer	LION	AdamW	AdamW	AdamW
learning rate	1e-4	0.01	3e-4	1e-4
weight decay	5e-3	0.01	0.5	0.5
epochs	20	30	10	10
warmup epochs	4	-	2	2
clip max norm	5.0	-	5.0	5.0
freeze backbone	True	True	True	True
number of GPUs	32	8	4	8
runtime	20hr	40m*	35m	5m

C Downstream Structural Benchmark Details

We summarize our structural datasets in Table 6. In Table 7, we report the hyperparameters used for fine-tuning on different downstream benchmarks. Additionally, we report all additional metrics for contact prediction and binding residue prediction in Table 8 and Table 9 respectively.

C.1 Structure Prediction

We train on proteins in the PDB and evaluate our model on the CAMEO dataset. Notably, unlike most benchmarks, CAMEO evaluations customarily do not include a sequence similarity split.

We initialize our model from SoloSeq Ahdritz et al. (2022) and freeze our *ISM* backbone. We fine-tune the folding trunk for 10 epochs using a cosine learning rate schedule with 2 warmup epochs. We use a batch size of 128 proteins. We use LION optimizer with a learning rate of 1×10^{-4} and weight decay of 0.01.

C.2 Contact Prediction

We follow the experimental setting as in SaProt (Su et al., 2023), which uses the contact prediction benchmark proposed by Rao et al. (2019) and Xu et al. (2022). In this benchmark, the goal is to predict whether a pair of residues is within a certain distance of one another. We evaluate our model on the ProteinNet CASP12 test set which contains at most 30% sequence identity to those in the training set.

In the main paper, we report precision at L (P@L) for long-range contacts at least 24 amino acids away. In Table 8, we thoroughly evaluate precision at L, $L/2$, $L/5$ on short, medium, and long-range intervals of [6,12], [12,24],[24,∞] amino acids respectively. The results of our baseline Amplify model closely align with those reported in their paper.

C.3 Secondary Structure

We use the secondary structure prediction benchmark from Xu et al. (2022). The protein's secondary structures are labeled one of three states - coil, strand, or helix. The training set is taken from Klausen

Table 8: **Comparisons to prior work on contact prediction.** *ISM* is structure-tuned on Uniclust30 while *ISM* † is additionally trained on the PDB. SaProt* takes the structure as input. The proteins in the training and test sets have at most 30% sequence similarity.

	Short Range			Medium Range			Long Range		
Method	P@L	P@L/2	P@L/5	P@L	P@L/2	P@L/5	P@L	P@L/2	P@L/5
ESM-2	0.45	0.45	0.50	0.45	0.45	0.54	0.35	0.42	0.52
ESM-2S	0.46	0.46	0.50	0.46	0.47	0.54	0.36	0.43	0.52
Amplify	0.38	0.38	0.41	0.36	0.35	0.40	0.23	0.28	0.35
S-PLM	0.49	0.49	0.55	0.48	0.49	0.57	0.36	0.43	0.54
SaProt*	0.57	0.57	0.64	0.53	0.55	0.66	0.48	**0.60**	**0.74**
ISM (Ours)	**0.62**	**0.62**	0.67	**0.60**	**0.61**	**0.68**	**0.49**	0.57	0.69
ISM † (Ours)	**0.62**	**0.62**	**0.68**	**0.60**	0.60	**0.68**	0.48	0.56	0.67

et al. (2019), which contains proteins with no more than 25% sequence similarity. The proteins in the test set have at most 25% sequence similarity to those in the training set. We evaluate the model's classification accuracy.

We freeze *ISM* and train a 2-layer classifier for 10 epochs using a cosine learning rate schedule with 2 warmup epochs. We use a batch size of 32 proteins. We use AdamW optimizer with a learning rate of 1×10^{-4} and weight decay of 0.5.

C.4 Binding Residues

We use the binding residues benchmark extracted from BioLip (Yang et al., 2012) prepared in the bindEmbed21 method (Littmann et al., 2021). At the time of dataset generation, they found 104,733 structures corresponding to 14,894 sequences in BioLiP. Upon deduplication at 20% sequence similarity, they ended up with 1314 proteins, of which 1014 were used for training and 300 were used for testing. We evaluate on the binary classification of whether a residue is within < 2.5Å of a metal ion, nucleic acid, or a small ligand (Littmann et al., 2021).

We freeze *ISM* and train a 2-layer classifier for 10 epochs using a cosine learning rate schedule with 2 warmup epochs. We use a batch size of 32 proteins. We use AdamW optimizer with a learning rate of 3×10^{-4} and weight decay of 0.5. Full results with all metrics are available in Table 9.

Table 9: **Comparisons to prior work on binding residue prediction.** *ISM* is structure-tuned on Uniclust30 while *ISM* † is additionally trained on the PDB. SaProt* takes the structure as input. The proteins in the training and test sets have at most 20% sequence similarity.

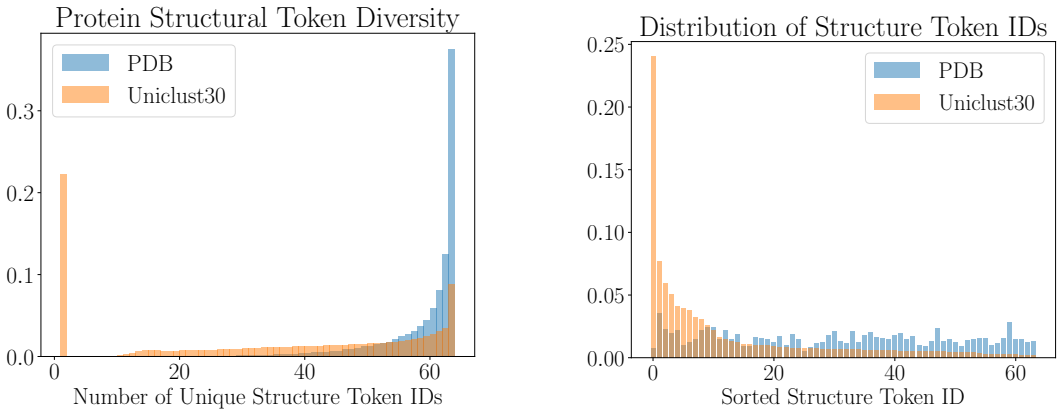
	Test			Independent		
Method	F1	MCC	AUC	F1	MCC	AUC
ESM (Lin et al., 2022)	0.31	0.34	0.84	0.28	0.28	0.82
ESM-2S	0.32	0.35	0.84	0.28	0.28	0.83
Amplify (Fournier et al., 2024)	0.22	0.26	0.81	0.19	0.18	0.79
S-PLM (Wang et al., 2023)	0.35	0.36	0.83	0.35	0.33	0.82
SaProt* (Su et al., 2023)	0.36	**0.38**	**0.87**	**0.35**	**0.33**	**0.87**
ISM (Ours)	0.35	0.37	0.86	0.33	0.31	0.85
ISM † (Ours)	**0.37**	**0.38**	0.86	0.34	0.32	0.85

D QUALITATIVE ANALYSIS ON THE CLUSTERING RESULTS.

We qualitatively evaluate our clusters both on the experimental structures in PDB and the AlphaFold structures in Uniclust30. First, we measured how many unique token IDs occurred in each protein in Figure 6a. Surprisingly, we observed that over 20% of the proteins contained the same token ID (token `[17]`) for every residue in the sequence. We then measured the number of times each token appeared in the entire Uniclust30 dataset and found that one token appeared over 20% in total (see Figure 6b). This turns out to be token `[17]` in Figure 7 which contains disordered regions with little or no secondary or tertiary structures. Interestingly, the microenvironments in PDB with token `[17]` do contain more sparse environments. This motivated us to remove training on the special token $s^* =$`[17]`.

We also looked at a few tokens in Figure 7 that either occurred the most/least and report our intuition below. Note that while our intuition can offer some rationale about the clusters, the model may capture relevant microenvironment features that are difficult for humans to interpret.

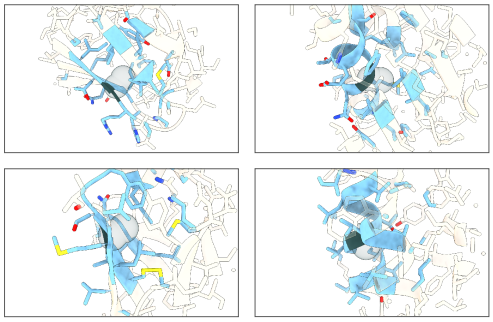
- `[id:3]`: In PDB proteins, this cluster consists primarily of semi-solvent exposed microenvironments with masked alanines. In Alphafold proteins, the cluster still contains semi-solvent exposed microenvironments but is not as heavily biased towards alanine. This is the least frequently seen structure token in Uniclust30.
- `[id:14]`: In PDB proteins, this cluster consists primarily of glycine residues that are solvent-exposed and mainly present in highly dynamic loops, often with little local secondary structure. In Alphafold proteins, we observe similar microenvironments, though not as heavily biased towards glycine. This is the second most frequently seen structure token in Uniclust30. It is the most frequently seen token ID in PDB.
- `[id:17]` In PDB proteins, this cluster consists primarily of residues that are solvent-exposed. However, in Alphafold proteins, this cluster corresponds to poorly folded regions (*e.g.,* N- and C-terminus residues and low pLDDT regions). This is the most frequent structure token in Uniclust30 and the second least frequent structure token in PDB. Because this token accurately captures poorly folded regions in computational structures, we drop this token during training on the Uniclust30 dataset.
- `[id:25]`: In PDB proteins, this cluster primarily consists of the tertiary interactions centered on disulfide bridges. In Alphafold proteins, this cluster also captures tertiary interactions of small amino acids, primarily glycine. We suspect that since AlphaFold does not explicitly model post-translation modifications, this cluster is not biased towards compact tertiary structures formed by disulfide bridges, as observed in the PDB. This is the least frequently seen structure token in PDB proteins.



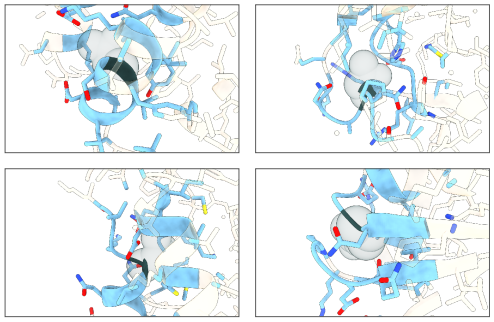
(a) **Histogram of Unique Structure Token IDs per Protein**

(b) **Distribution of tokens across the entire dataset**

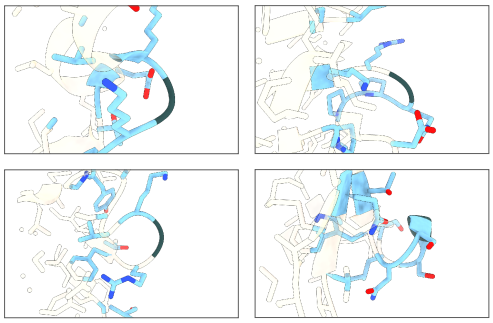
Figure 6: **Measuring the diversity of tokens in both PDB and Uniclust30.**



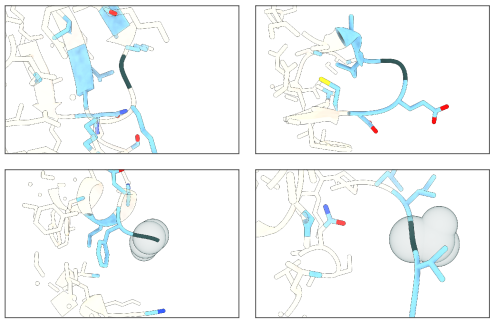
(a) PDB Structures with `[struct id 3]`



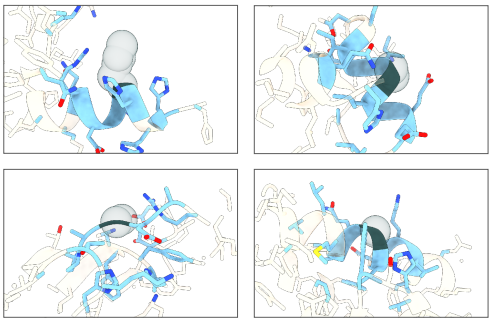
(b) AlphaFold Structures with `[struct id 3]`



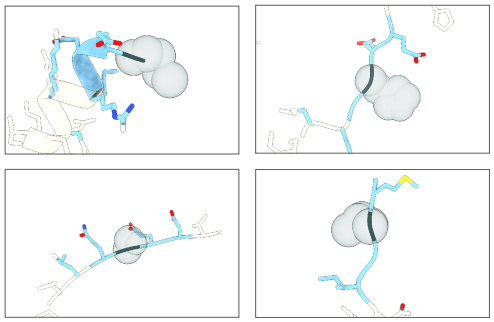
(c) PDB Structures with `[struct id 14]`



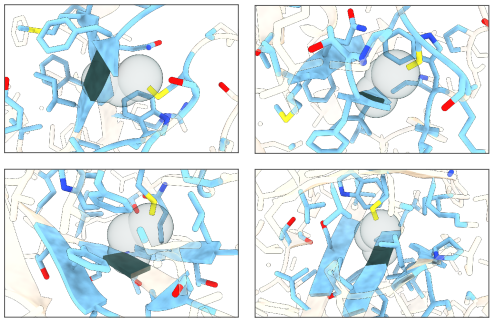
(d) AlphaFold Structures with `[struct id 14]`



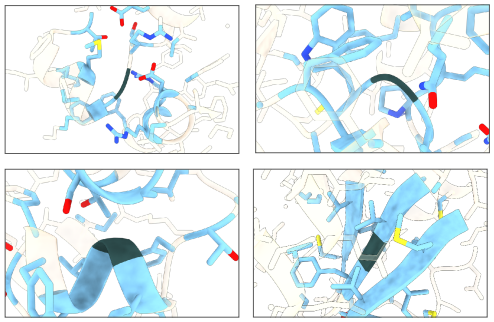
(e) PDB Structures with `[struct id 17]`



(f) AlphaFold Structures with `[struct id 17]`



(g) PDB Structures with `[struct id 25]`



(h) AlphaFold Structures with `[struct id 25]`

Figure 7: **More Cluster-based Microenvironment Visualizations**. Residues in sky blue are within the microenvironment, while white residues are outside and included for context. The grey density indicates the masked-out amino acid. Nodes are colored by element: blue for nitrogen, red for oxygen, and yellow for sulfur. The left two columns display structures from the PDB, while the right two columns show protein sequences from Uniclust30, folded using AlphaFold.