Trainable subnetworks reveal insights into structure knowledge organization in protein language models

Ria Vinod[1], Ava P. Amini[2], Lorin Crawford[2,*], and Kevin K. Yang[2,*]

[1]Center for Computational Molecular Biology, Brown University, Providence, RI, USA
[2]Microsoft Research, Cambridge, MA, USA
[*]Corresponding Email(s): lcrawford@microsoft.com; yang.kevin@microsoft.com

Abstract

Protein language models (PLMs) pretrained via a masked language modeling objective have proven effective across a range of structure-related tasks, including high-resolution structure prediction. However, it remains unclear to what extent these models factorize protein structural categories among their learned parameters. In this work, we introduce trainable subnetworks, which mask out the PLM weights responsible for language modeling performance on a structural category of proteins. We systematically train 36 PLM subnetworks targeting both sequence- and residue-level features at varying degrees of resolution using annotations defined by the CATH taxonomy and secondary structure elements. Using these PLM subnetworks, we assess how structural factorization in PLMs influences downstream structure prediction. Our results show that PLMs are highly sensitive to sequence-level features and can predominantly disentangle extremely coarse or fine-grained information. Furthermore, we observe that structure prediction is highly responsive to factorized PLM representations and that small changes in language modeling performance can significantly impair PLM-based structure prediction capabilities. Our work presents a framework for studying feature entanglement within pretrained PLMs and can be leveraged to improve the alignment of learned PLM representations with known biological concepts.

Introduction

Understanding protein structure is essential for deciphering biological function, as a protein's structure governs its molecular stability, interactions, and activity. Recent protein language models (PLMs) trained purely on sequence data have been shown to learn representations that implicitly encode structural information [1–4]. These models have proven effective in a wide range of protein engineering tasks including structure prediction [5–7], function annotation [8, 9], mutation effect estimation [10], and even the design of novel proteins [11–13].

Many PLMs are pretrained with a self-supervised masked language modeling (MLM) objective, where models are tasked with predicting the amino acid identities of randomly masked tokens in a sequence [1, 2, 14, 15]. Since protein structure is fundamentally determined by amino acid sequence, PLMs can implicitly encode structural information in their weights. ESM-2, a family of protein language models trained on evolutionary-scale data, showed that performance on the language modeling task can predict the quality of a PLM-predicted structure [1, 7]. PLM representations are now widely used as inputs to models that predict structural coordinates [16], and scaling analyses have shown that improvements in MLM performance improve single-sequence structure prediction accuracy [7, 17].

The simplicity and efficacy of pretraining language models on protein sequences have led to growing interest in understanding their internal mechanisms, establishing interpretability as an important research direction. With respect to the structure features that PLMs learn, prior

work has shown that protein contact information is stored in attention maps [18], that PLMs learn co-evolving motifs which emulate an understanding of fundamental biophysics [19], and that sparse latent features from PLMs capture known functional biophysical properties and motifs [20, 21].

To date, however, it remains unclear (i) if and how structure information is factorized and stored in the learned PLM weights, and (ii) whether this factorization affects performance on downstream structure prediction tasks. One approach to uncover whether such a factorization of concepts in pretrained language models exists is via subnetwork discovery [22]. Subnetworks are sparse computational subgraphs of a pretrained model's weights that are responsible for performance on a specific task or class of inputs. In the natural language domain, subnetwork discovery has been widely used to uncover linguistic properties—such as semantics, syntax, and relational entities—learned during pretraining by defining concepts and localizing weights, neurons, or layers that encode them [23–27].

In this work, we leverage subnetworks to interrogate the factorization of structural categories of proteins in the pretrained PLM, ESM-2 (Fig. 1). Our goal is to find PLM subnetworks—sparse subgraphs of the original model weights—that when isolated, suppress the model's ability to make correct predictions on one class of inputs while preserving MLM performance on all other classes of inputs. We systematically train 36 such PLM subnetworks to suppress either residue- or sequence-level structural information across different scales of the CATH hierarchy. Our results reveal that structural categories are indeed encoded in a factorized manner within PLM weights and that, although non-suppressed inputs achieve ESM-2-level perplexity, structure prediction is still statistically significantly perturbed. Together, our subnetworks approach and the suite of analyses we perform provide insight into how PLMs organize structural features among learned parameters.

Methods

Suppression and maintenance inputs are defined by structural annotations

Protein secondary structure refers to the local spatial arrangement of amino acid residue backbone atoms. Each residue in a protein sequence adopts a secondary structure, which can be classified as an alpha helix, beta sheet, or loop. The resulting structural arrangements enable further categorization of proteins at the sequence level, grouping them by shared architectural and evolutionary features (Fig. 1A). In this work, we define structural categories at the residue level (alpha helix, beta sheet, loop) using DSSP [28] and at the sequence-level (Class, Architecture, Topology, Homologous Superfamily) using the CATH taxonomy (Fig. 1D).

Notation

We denote a pretrained PLM as $f(\mathbf{x}, \boldsymbol{\theta})$, where $\mathbf{x}$ represents the input sequence and $\boldsymbol{\theta}$ are the model weights. The objective of our training procedure is to predict a binary mask, $\mathbf{m} \in \{0, 1\}^K$, where K denotes the number of parameters in the PLM that we seek to learn the mask over. A subnetwork can be obtained by taking the Hadamard product between the binary mask and pretrained PLM weights, $f(\mathbf{x}, \mathbf{m} \odot \boldsymbol{\theta})$. Subnetworks are trained independently to suppress a structural category of sequence-level or residue-level inputs (i.e., all suppressed inputs belong to the same category defined by structural annotation). We define a suppression input sequence as $\mathbf{x}^*$ and all other input sequences (i.e., maintenance inputs) as $\mathbf{x}'$ (Fig. 1A; suppressed inputs are red, maintained inputs are black). For residue-level suppression, let $\mathcal{A}$ be an annotation from DSSP that contains a set of J positions corresponding to residues in {alpha helix, beta sheet, loop}. We define the J suppression inputs for this DSSP annotation as $\mathbf{x}_{(J^*)}$. The maintenance inputs are all remaining complementary positions of residues with annotation $\mathcal{A}$, which are then defined as $\mathbf{x}_{(J')}$.

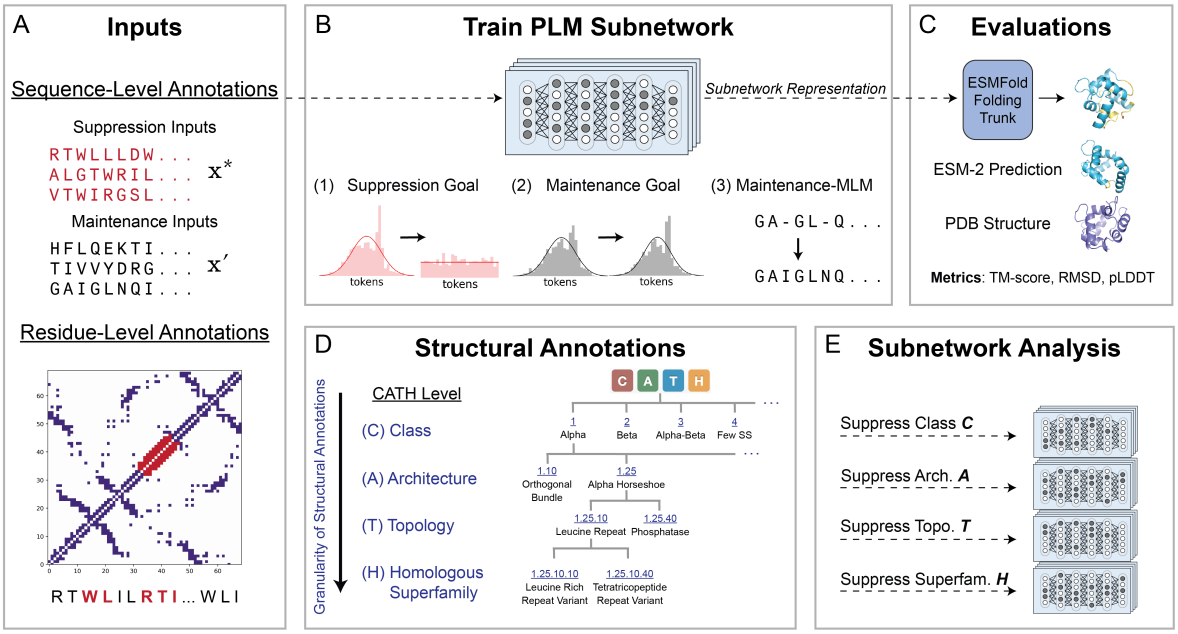


Figure 1. Summary of subnetwork training procedure, evaluations, and analysis. **(A)** Suppression inputs can be defined at either the protein sequence-level (based on CATH annotations) or the residue-level (based on 3-way secondary structure annotations). Suppressed inputs are red; maintained inputs are black. **(B)** PLM subnetworks are trained via a weighted multi-objective differentiable mask learning scheme, which defines suppression and maintenance goals for a binary mask learning procedure. **(C)** The subnetwork sequence representation is passed as input to the ESMFold folding trunk to predict a 3D structure. The resulting 3D structure can be compared to the ESM-2 prediction for the same sequence, both of which are independently aligned to ground-truth structures from the Protein Data Bank (PDB). **(D)** The CATH database offers a hierarchical classification of protein structural annotations across four levels, grouping proteins by their fold patterns and evolutionary relationships. Each successive level provides increasingly fine-grained structural annotations. **(E)** Independent subnetworks are trained to suppress different structural categories.

Subnetwork training objective

To obtain a subnetwork $f(\mathbf{x}, \mathbf{m} \odot \boldsymbol{\theta})$, we learn a binary mask $\mathbf{m}$ using a weighted loss that consists of the following components (Fig. 1B):

1. **Suppression goal.** If the mask is working appropriately, the subnetwork should struggle to reconstruct its suppressed inputs accurately. In other words, a well-calibrated predictive distribution over the suppressed input tokens should be approximately uniformly distributed with respect to the vocabulary V. As a result, we define the suppression loss as minimizing the Kullback–Leibler (KL) divergence between (i) the predictive distribution of the subnetwork over the tokens corresponding to the suppression inputs and (ii) a uniform reference distribution over tokens in the vocabulary (denoted as $\mathcal{U}_V$). For sequence-level suppression, this corresponds to

$$\mathcal{L}_{\text{supp}} = \text{KL}\left(f(\mathbf{x}^*, \mathbf{m} \odot \boldsymbol{\theta}), \mathcal{U}_V\right). \tag{1}$$

   Similarly, for residue-level suppression,

$$\mathcal{L}_{\text{supp}} = \text{KL}\left(f(\mathbf{x}_{(J^*)}, \mathbf{m} \odot \boldsymbol{\theta}), \mathcal{U}_V\right) \tag{2}$$

   where, again, the index J^* is used to represent all the positions corresponding to residues with DSSP annotation $\mathcal{A}$.

2. **Maintenance goal.** Even in the presence of a mask, the subnetwork should preserve the predictive behavior of the full PLM on maintenance inputs. Therefore, as a maintenance goal, we also aim to minimize the KL divergence between (i) the predictive distribution of the subnetwork over the tokens corresponding to the maintenance inputs and (ii) the predictive distribution for the pretrained PLM over the same elements. For sequence-level suppression, this is expressed as the following

$$\mathcal{L}_{\text{maint}} = \text{KL}\left(f(\mathbf{x}', \mathbf{m} \odot \boldsymbol{\theta}), f(\mathbf{x}', \boldsymbol{\theta})\right). \tag{3}$$

Similarly, for the residue-level suppression, we write

$$\mathcal{L}_{\text{maint}} = \text{KL}\left(f(\mathbf{x}_{(J')}, \mathbf{m} \odot \boldsymbol{\theta}), f(\mathbf{x}_{(J')}, \boldsymbol{\theta})\right) \tag{4}$$

where, again, the index J' denotes the set of positions not included in a given residue annotation $\mathcal{A}$ from DSSP.

3. **Maintenance masked language modeling (MLM) goal.** To ensure that the subnetwork retains performance on the MLM task, we include an MLM objective on the maintenance inputs. We randomly select 15% of sequence positions over which the MLM loss is computed. Of these M positions, 80% are replaced with a mask token, 10% are randomly replaced, and 10% are unchanged. Namely,

$$\mathcal{L}_{\text{MLM}} = \sum_{i \in M} \log p_{\mathbf{m} \odot \boldsymbol{\theta}}\left(x'_i \mid \mathbf{x}'_{(M')}\right) \tag{5}$$

where M' are unmasked positions (i.e., true amino acid identities). To perform residue-level suppression, we again randomly choose a set of M masked positions with the same mask-and-mutate scheme; however, this time, we compute the MLM loss with respect to the masked indices that overlap with the positions corresponding to a particular residue-level annotation $\mathcal{A}$, which we denote as $M \cap J$. This can be expressed as

$$\mathcal{L}_{\text{MLM}} = \sum_{i \in M \cap J'} \log p_{\mathbf{m} \odot \boldsymbol{\theta}}\left(x_i \mid \mathbf{x}_{(M' \cap J')}\right) \tag{6}$$

where $M \cap J'$ denotes intersection of indices that are in M and not in J, and $M' \cap J'$ represents the indices that are not in either set.

Overall, the final weighted training objective is then represented as the sum

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{supp}} + \lambda_2 \mathcal{L}_{\text{maint}} + \lambda_3 \mathcal{L}_{\text{MLM}} \tag{7}$$

where $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \lambda_3\}$ are hyperparameters. A description of how we select training hyperparameters can be found in Appendix S1.1.1.

Differentiable weight masking for subnetworks

Following previous work [23, 24, 29], we adopt a differentiable weight masking scheme to learn $\mathbf{m}$. Here, each binary mask parameter $m_i \in \{0, 1\}$ is sampled from a Gumbel distribution. We learn a logit $l_i \in \mathbb{R}$ for every i-th parameter and obtain a continuous mask score over the unit interval via the following Gumbel softmax transformation

$$s_i = \sigma\left\{-\frac{1}{\tau}\left[l_i + \log\left(\frac{U_1}{1 - U_1}\right)\right]\right\} \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, τ is a temperature scaling hyperparameter, and $U_1 \sim \mathcal{U}(0, 1)$ is a random variable drawn from a standard uniform distribution. This Gumbel noise introduces

stochasticity into the logit sampling process which results in exploring different binary mask configurations during training. We backpropagate through the collection of continuous mask scores $\mathbf{s}$ and threshold values to obtain binarized mask values $\mathbf{m}$ using the following

$$m_i = [\mathbb{1}(s_i > T) - s_i]_{\text{detach}} + s_i, \tag{9}$$

where $\mathbb{1}(\cdot)$ is an indicator function, T is the mask score threshold, and $[\cdot]_{\text{detach}}$ prevents back-propagation through the discrete values in $\mathbf{m}$. This thresholding operation enables differentiable training by allowing gradients to flow through the continuous mask scores, while still computing the loss with respect to binary predictions [26, 30]. The sparsity is defined as the proportion of zeros in the learned binary mask—and therefore in the subnetwork—which is calculated as

$$\text{sparsity}(\mathbf{m}) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}(m_i = 0), \tag{10}$$

where $\mathbf{m} = [m_1, \ldots, m_K]^\top \in \{0, 1\}^K$ and again, K is the total number of mask parameters. In our results, we represent this number as a percentage and multiply the value above by 100%. A description of how we select weight masking hyperparameters can be found in Appendix S1.1.2.

Model architecture and datasets

We learn subnetworks in the ESM-2 650 million parameter model [7]. Recent work has shown that most layers in this model encode secondary structure information [21], so we choose to mask layers 6–33, corresponding to the latest 80% of transformer layers. While other binary masking approaches often focus on the final layers to capture fine-grained concepts or properties, our broader masking prevents over-reliance on weak early-layer signals and spurious late-layer correlations. Details on the weight masking scheme, choice of which hidden layers to mask, hyperparameters, and mask initialization values can be found in Appendix S1.

For training and evaluation, we use the CATH S20 version 4.3.0 release. This dataset is comprised of CATH domains, which are sequences clustered at a maximum of 20% pairwise sequence identity with at least 60% alignment overlap. This ensures low redundancy while maintaining structural and functional diversity across domains. Every domain has an annotation at successive levels of the CATH hierarchy: Class (C), Architecture (A), Topology (T), and Homologous Superfamily (H). We use DSSP [28] to obtain reduced 3-way residue-level secondary structure categories, which is the procedure from Ahdritz et al. [31]. During training, we filter down to the set of CATH domains with length between 64 and 1024 residues (the maximum context window of ESM-2) and with available PDB structures. Altogether, this leaves 8886 CATH domains. For each subnetwork, we randomly split this data into 70% for training, 20% for validation, and 10% for a held out set. Subnetworks are learned according to the suppression and maintenance inputs present in the train split. At evaluation time, we perform MLM evaluations over all splits, which allows for aggregating performance on suppression and maintenance inputs that are both seen during training via the train split, but also unseen inputs in the validation and test splits. We perform the structure prediction evaluations with the ESMFold folding trunk on the validation and test sets to limit computational cost. We provide details on CATH data and annotation frequencies in Fig. S1.

Results

We use trained subnetworks to assess whether structural categories are factorized in PLM weights and, in turn, whether this factorization affects downstream structure prediction. To this end, we first evaluate 36 trained subnetworks to assess MLM performance on suppression and maintenance inputs. Second, we pass subnetwork representations as inputs to the ESMFold

folding trunk to evaluate the effect of the suppression on structure prediction capabilities (see Fig. 1C). In both sets of evaluations, we compare the effects of sequence-level suppression and residue-level suppression.

Sparse subnetworks enable successful factorizations in ESM-2 weights

A structural category of proteins is considered to be factorized in PLM weights if we can identify a subnetwork that selectively reduces MLM performance on suppression inputs while maintaining performance on maintenance inputs, relative to the full PLM. We measure MLM performance using perplexity (i.e., the exponential of the negative log-likelihood), computed separately on the suppressed and maintained inputs for both the subnetworks and the pretrained ESM-2 model (Fig. 2A).

By our training procedure, a subnetwork can achieve this differential performance by identifying a sparse subgraph of PLM weights where parameters that encode information about the suppressed input category are zeroed out. As the subnetwork training procedure does not explicitly promote any sparse regularization, the percentage of learned sparsity is intrinsic to any discovered implicit factorization. We quantify the mean learned sparsity (defined in Eq. (10) and reported as a percentage) across structural levels in Fig. 2B. As suppressed categories become more fine-grained within the CATH hierarchy, the learned sparsity decreases, indicating that smaller structural categories can be factorized by zeroing out fewer parameters. This trend suggests a correlation between the learned percent sparsity of the subnetwork and the frequency of structural annotations (i.e., annotation granularity) in the data.

We observe that the trained subnetworks consistently yield higher perplexity on the suppression inputs, implying that structure-relevant representations can be selectively factorized in the PLM weight space (Fig. 2C). Targeted suppression of residues belonging to a secondary structure (i.e., alpha helices or beta strands) results in an approximate two-fold increase in perplexity of these residues. Suppression of broader sequence-level annotations, such as CATH Class where the suppression set sizes are largest, yields predictive distributions resembling random noise, with perplexity approaching 20—the value expected from uniform guessing over 20 amino acids. A similar increase in perplexity is observed when suppressing Homologous Superfamilies, despite much smaller suppression sets (< 100 sequences), suggesting that factorization is strongest at the coarsest and finest levels of annotation granularity. In contrast, structural categories are more weakly factorized at intermediate CATH levels like Architecture and Topology, as evidenced by lower perplexities on suppression inputs as compared to that of other CATH levels. As a baseline, we include subnetworks trained to suppress N randomly selected sequences. These fail to produce differential performance on the suppressed inputs. The configurations of all trained subnetworks are reported in Table S1, and the corresponding MLM performance is reported in Table S2. Together, the percent sparsity and MLM evaluations show that subnetworks can increase perplexity in a targeted manner by identifying sparse factorizations of structural information in PLMs, with the strongest effects observed at the coarsest and finest levels of annotation granularity.

Subnetworks perturb structure prediction accuracy on both suppressed and maintained inputs

Having evaluated subnetworks on the language modeling task, we then assessed how suppression via a subnetwork affects structure prediction accuracy. For each input sequence, we extract sequence representations from the subnetwork and from ESM-2, and we pass them as separate inputs to the ESMFold folding trunk for structure prediction. Both predicted structures are independently aligned to the ground truth PDB structure to obtain a template modeling (TM) score, root mean square deviation (RMSD), and predicted local distance difference test

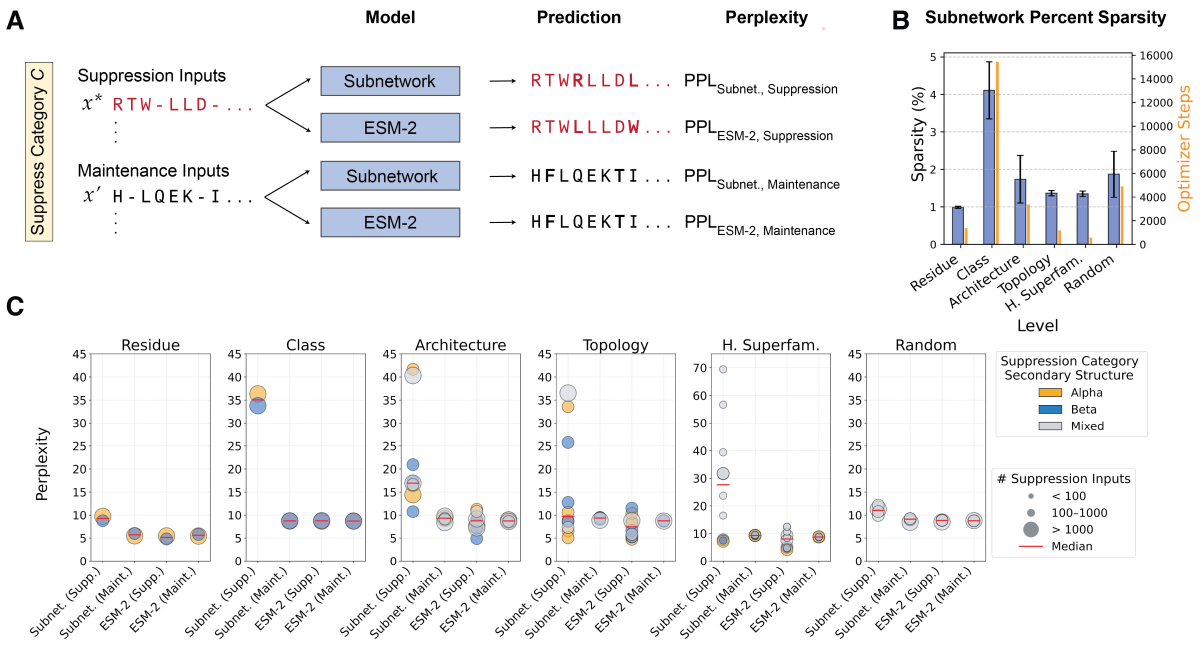


Figure 2. Ablating less than 5% of ESM-2 parameters increases perplexity on suppressed inputs and reveals successful factorizations of structural categories in PLM weights. (A) For a given subnetwork, suppression and maintenance inputs are defined by their structural annotation. At inference time, the same sequence is fed as separate inputs to the subnetwork and ESM-2. ESM-2 performance on the same suppression and maintenance inputs, labeled as "ESM-2 Supp." and "ESM-2 Maint." respectively, provides a baseline for the subnetwork. (B) Learned sparsity of each subnetwork across categories of structural annotations (reported mean with ± standard deviation depicted by the error bars). The right y-axis shows the average number of optimization steps in yellow. **(C)** Perplexity (y-axis) on test sequences across suppression and maintenance input categories (x-axis). Each point represents an independently trained subnetwork, where marker size indicates number of suppression inputs in the structural category and color indicates its predominant secondary structure type. Suppression and maintenance inputs are defined per subnetwork; ESM-2 points reflect baseline perplexity on those same inputs.

(pLDDT) for each model's prediction. This procedure is repeated on all suppression inputs $\mathbf{x}^*$ and all maintenance inputs $\mathbf{x}'$ in the validation datasets (Fig. 3A).

First, we evaluate whether the subnetwork leads to a decrease in structure prediction accuracy relative to the ESM-2 performance on these inputs (Fig. 3B). For each input, we compute the change in RMSD where $\Delta\text{RMSD} = \text{RMSD}_{\text{Subnet., PDB}} - \text{RMSD}_{\text{ESM-2, PDB}}$. These differences are consistently larger on suppression inputs and minor on maintenance inputs (Fig. 3B). This suggests that the subnetwork is able to ablate structural information pertaining to the suppressed inputs in PLM weights. We then perform separate paired t-tests for the inputs in the suppression and maintenance sets, comparing $\text{RMSD}_{\text{Subnet., PDB}}$ and $\text{RMSD}_{\text{ESM-2, PDB}}$. Both tests result in significant p-values, suggesting that despite targeting only suppression inputs, the subnetwork also significantly affects structure prediction on maintenance inputs. That is, $\Delta\text{RMSD}_{\text{supp.}}$ and $\Delta\text{RMSD}_{\text{maint.}}$ are both statistically significant. We repeat this procedure using the TM-score and pLDDT and observe a similar trend (Fig. S2A-B). All structure prediction metrics and p-values are reported in Table S3.

Next, we assess whether the magnitude of these differences is greater for suppression inputs than for maintenance inputs across CATH and secondary structure categories (Fig. 3C). That is, we seek to confirm that $|\Delta\text{RMSD}_{\text{supp.}}| - |\Delta\text{RMSD}_{\text{maint.}}| > 0$ (Fig. 3C). We find that these differ-

ences are consistently positive, suggesting that a subnetwork always results in a greater increase in RMSD on the suppression inputs when compared to maintenance inputs. We also confirm that the differences in magnitude are consistently statistically significant by applying a two-sample Kolmogorov–Smirnov (KS) test between $|\Delta\text{RMSD}_{\text{supp.}}|$ and $|\Delta\text{RMSD}_{\text{maint.}}|$, rejecting the null hypothesis that the suppression and maintenance inputs exhibit the same distributions of difference in magnitudes of RMSD increase. We repeat the same analysis for TM-score and pLDDT and observe the same trend (Fig. S2C–D). All structure prediction metrics and p-values are reported in Table S3.

Structure prediction is more sensitive to suppressed sequence features

Alpha helices and beta sheets are local secondary structure elements that, when predominant in a sequence, define the overall CATH Class of a protein domain. Sequences consisting of majority alpha helices form "Mainly Alpha" domains, while those rich in beta sheets form "Mainly Beta" domains at the CATH Class level. This connection allows us to directly compare the change in pLDDT between structures predicted by ESM-2 and those predicted by subnetworks trained to suppress (i) sequences labeled by CATH Class (e.g., Mainly Alpha or Mainly Beta), and (ii) residues belonging to specific secondary structures (e.g., alpha helices or beta sheets). We illustrate predicted structures for alpha suppression subnetworks in Fig. 3D and beta suppression subnetworks in Fig. 3E, where in each panel the top row is a suppressed input and the bottom row is a maintained input. While suppressing at the residue level based on secondary structure annotations only affects the mean pLDDT of the full predicted structure, suppressing sequences leads to no clear predicted fold or structure pattern. Since pLDDT is a per residue metric, we compute residue-specific pLDDT of predicted structures for both residue- and sequence-level suppression with alpha type (Fig. 3F) and beta type (Fig. 3G). Suppressing sequences still leads to a lower residue-specific pLDDT of the target residue type for both alpha and beta suppression subnetworks. This suggests that factorization of structural information in PLM weights is more sensitive to sequence-level suppression, and, by extension, that PLMs more effectively model distributions at the sequence level.

Discussion

We introduce subnetwork discovery as a post-hoc mechanistic interpretability technique for pretrained protein language models. We train and evaluate 36 subnetworks in ESM-2 using differentiable weight masking to probe how structural information is factorized in the pretrained model. Because subnetworks are extracted without any fine-tuning, they only expose relationships already present in the pretrained weights. The method is therefore a lightweight alternative to concept-level pretraining and other supervision-heavy interpretable pretraining schemes [32]. Our experiments show that PLMs can effectively disentangle both coarse and fine-grained CATH structural categories. Since coarse-grained annotations of CATH Classes are defined by secondary structure content, our work supports early findings that PLM representations cluster by secondary structure categories in low-dimensional space [15]. Notably, our approach reveals that, though non-suppressed sequence inputs can retain accuracy on the masked language modeling task relative to ESM-2, accuracy suffers when structures are predicted using representations from a subnetwork. Furthermore, sequences are factorized more effectively than secondary-structure residues within PLM weights, underscoring the importance of broader sequence context in pretrained PLMs. In contrast, protein structure models have been shown to demonstrate high generalization capabilities despite CATH-class level ablations of training data [31]. However, we find that suppressing beta-sheet content leads to larger reductions in structure prediction accuracy than alpha-helix suppression, consistent with OpenFold's observation that beta-sheet-containing proteins are more important for generalization [31].

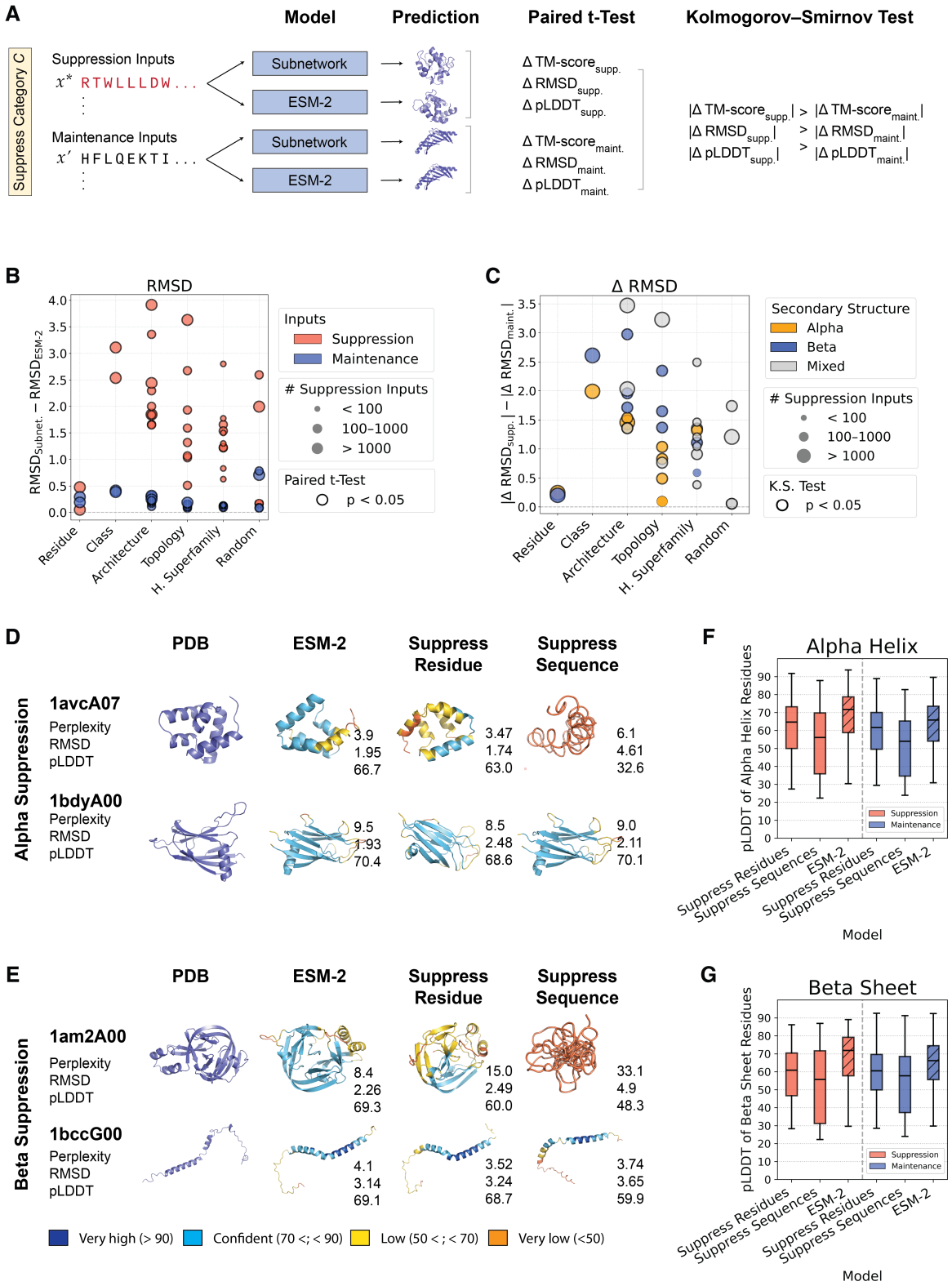

A
Model
Prediction
Paired t-Test
Kolmogorov–Smirnov Test
Suppress Category C
Suppression Inputs
x* RTWLLLDW...
Maintenance Inputs
x′ HFLQEKTI...
Subnetwork
ESM-2
Subnetwork
ESM-2
Δ TM-score supp.
Δ RMSD supp.
Δ pLDDT supp.
Δ TM-score maint.
Δ RMSD maint.
Δ pLDDT maint.
|Δ TM-score supp.| > |Δ TM-score maint.|
|Δ RMSD supp.| > |Δ RMSD maint.|
|Δ pLDDT supp.| > |Δ pLDDT maint.|
B
RMSD
RMSD Subnet. − RMSD ESM-2
Residue
Class
Architecture
Topology
H. Superfamily
Random
Inputs
Suppression
Maintenance
Suppression Inputs
< 100
100–1000
> 1000
Paired t-Test
p < 0.05
C
Δ RMSD
|Δ RMSD supp.| − |Δ RMSD maint.|
Residue
Class
Architecture
Topology
H. Superfamily
Random
Secondary Structure
Alpha
Beta
Mixed
Suppression Inputs
< 100
100–1000
> 1000
K.S. Test
p < 0.05
D
PDB
ESM-2
Suppress Residue
Suppress Sequence
Alpha Suppression
1avcA07
Perplexity
RMSD
pLDDT
3.9
1.95
66.7
3.47
1.74
63.0
6.1
4.61
32.6
1bdyA00
Perplexity
RMSD
pLDDT
9.5
1.93
70.4
8.5
2.48
68.6
9.0
2.11
70.1
F
Alpha Helix
pLDDT of Alpha Helix Residues
Suppression
Maintenance
Suppress Residues
Suppress Sequences
ESM-2
Model
E
PDB
ESM-2
Suppress Residue
Suppress Sequence
Beta Suppression
1am2A00
Perplexity
RMSD
pLDDT
8.4
2.26
69.3
15.0
2.49
60.0
33.1
4.9
48.3
1bccG00
Perplexity
RMSD
pLDDT
4.1
3.14
69.1
3.52
3.24
68.7
3.74
3.65
59.9
Very high (> 90)
Confident (70 <; < 90)
Low (50 < ; < 70)
Very low (<50)
G
Beta Sheet
pLDDT of Beta Sheet Residues
Suppression
Maintenance
Suppress Residues
Suppress Sequences
ESM-2
Model

Figure 3. Subnetworks consistently impair structure prediction capabilities on suppressed inputs. (A) Sequence representations for a given input are obtained from the subnetwork and ESM-2 and fed as inputs to the ESMFold folding trunk to produce two predicted structures. Both structures are independently aligned to the ground truth PDB structure, and metrics are calculated for evaluations. **(B)** RMSD differences between the subnetwork and ESM-2 baseline (y-axis) across structural levels (x-axis). Each point represents the RMSD increase for suppression inputs (red) or maintenance inputs (blue) relative to ESM-2. Marker size indicates the number of suppression inputs for each subnetwork. Bold outlines of markers indicate significant paired t-test p-values ($p < 0.05$). **(C)** Difference in absolute RMSD changes from the ESM-2 baseline (y-axis), stratified by structural level (x-axis). Each point corresponds to an individual subnetwork and shows the difference between the magnitudes of suppression and maintenance ΔRMSD values. Marker size reflects the number of suppressed inputs in the subnetwork, and color indicates secondary structure type. Bold outlines of markers indicate significance by Kolmogorov–Smirnov (KS) test ($p < 0.05$). **(D–E)** Visualizations and evaluation metrics for subnetwork predictions on select validation targets under (D) alpha suppression and (E) beta suppression. For each example, the ground truth PDB structure, ESM-2 prediction, and subnetwork predictions under residue-level and sequence-level suppression are shown. Each prediction is annotated with perplexity, mean RMSD, and mean pLDDT scores. **(F–G)** Distributions of predicted per-residue pLDDT scores across (F) alpha helix and (G) beta sheet residues across suppression and maintenance conditions, for subnetworks and baseline ESM-2 predictions. Each box shows the median and interquartile range.

Despite these successes, our work has some limitations. First, while thorough, our analysis is limited to the CATH dataset and the fundamental secondary structures of residues. Because our approach depends on structural annotations, it is restricted to labeled sequence datasets, which can be costly to acquire. Second, PLM size [7], pretraining strategy [33], and training data distribution [34] also influence learned representations and, therefore, any subsequent factorizations that subnetworks can uncover. We use ESM-2 650M due to its prevalence in the scientific community, which is driven by its high accuracy and moderate compute cost. We do not evaluate larger ESM-2 versions or any other MLM-based PLMs; since prior work suggests that downstream effects scale with model size [7, 17], we expect similarly scaled results of our approach with these models, though this remains to be determined empirically. Finally, there exist other single-sequence, structure prediction models, and it is known that model choice can influence structure prediction outcomes [5, 6]. Though our work is, to our knowledge, the first study of disentangling learned protein structural categories in PLMs, future work could examine how different pretraining approaches affect learned factorizations.

We envision that our approach will help accelerate PLM development and interpretability, as our framework can be readily extended to any labeled protein sequence dataset. It provides a lens to understand how factorizations affect single-sequence downstream prediction tasks. As masked language model-based PLMs scale in size and availability, understanding how PLMs synthesize biological information becomes increasingly important. Our approach, along with other mechanistic interpretability methods, can help identify PLM success and failure modes as well as align pretraining methods and evaluations with known biological patterns.

Code and data availability

All code is available under an open-source MIT license at `https://github.com/microsoft/plm_subnetworks`. Training data, model configurations and checkpoints, and results are available at `https://zenodo.org/records/15514844`.

Acknowledgments

This research was conducted using a combination of computational resources and services provided by Microsoft Research and the Center for Computation and Visualization at Brown University. This research was also supported in part by a David & Lucile Packard Fellowship for Science and Engineering awarded to LC. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the funders.

Author contributions

RV, APA, LC, and KKY conceived the study. RV developed the methods, software, and performed the analyses. LC provided resources. APA, LC, and KKY supervised the project. All authors interpreted the results and wrote the manuscript.

Declaration of interests

The authors have declared that no competing interests exist.

References

[1] 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. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, April 2021. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/doi/10.1073/pnas.2016239118`. Publisher: Proceedings of the National Academy of Sciences.

[2] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning, May 2021. URL `https://www.biorxiv.org/content/10.1101/2020.07.12.199554v3`. Pages: 2020.07.12.199554 Section: New Results.

[3] Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F. Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA Transformer, August 2021. URL `https://www.biorxiv.org/content/10.1101/2021.02.12.430858v3`. Pages: 2021.02.12.430858 Section: New Results.

[4] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):723, December 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3220-8.

[5] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence, July 2022. URL `https://www.biorxiv.org/content/10.1101/2022.07.21.500999v1`. Pages: 2022.07.21.500999 Section: New Results.

[6] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M. Church, Peter K. Sorger, and Mohammed AlQuraishi. Single-sequence protein structure

prediction using language models from deep learning, August 2021. URL https://www.biorxiv.org/content/10.1101/2021.08.02.454840v1. Pages: 2021.08.02.454840 Section: New Results.

[7] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, March 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574. Publisher: American Association for the Advancement of Science.

[8] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating Protein Transfer Learning with TAPE, June 2019. URL http://arxiv.org/abs/1906.08230. arXiv:1906.08230 [cs].

[9] Samir Char, Nathaniel Corley, Sarah Alamdari, Kevin K Yang, and Ava P Amini. Protnote: a multimodal method for protein–function annotation. *Bioinformatics*, 41(5):btaf170, 04 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf170. URL https://doi.org/10.1093/bioinformatics/btaf170.

[10] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function, November 2021. URL https://www.biorxiv.org/content/10.1101/2021.07.09.450648v2. Pages: 2021.07.09.450648 Section: New Results.

[11] Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the Boundaries of Protein Language Models, June 2022. URL http://arxiv.org/abs/2206.13517. arXiv:2206.13517 [cs].

[12] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Neil Tenenholtz, Robert Strome, Alan M. Moses, Alex X. Lu, Nicolò Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need, November 2024. URL https://www.biorxiv.org/content/10.1101/2023.09.11.556673v2. Pages: 2023.09.11.556673 Section: New Results.

[13] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model, December 2024. URL https://www.biorxiv.org/content/10.1101/2024.07.01.600583v2. Pages: 2024.07.01.600583 Section: New Results.

[14] Kevin K. Yang, Nicolo Fusi, and Alex X. Lu. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Systems*, 15(3):286–294.e2, March 2024. ISSN 2405-4712. doi: 10.1016/j.cels.2024.01.008. URL https://www.sciencedirect.com/science/article/pii/S2405471224000292.

[15] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, December 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0598-1.

[16] Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Le Song. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein, December 2024. URL http://arxiv.org/abs/2401.06199. arXiv:2401.06199 [q-bio].

[17] Francesca-Zhoufan Li, Ava P. Amini, Yisong Yue, Kevin K. Yang, and Alex X. Lu. Feature Reuse and Scaling: Understanding Transfer Learning with Protein Language Models, February 2024. URL https://www.biorxiv.org/content/10.1101/2024.02.05.578959v2. Pages: 2024.02.05.578959 Section: New Results.

[18] Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. BERTology Meets Biology: Interpreting Attention in Protein Language Models, March 2021. URL http://arxiv.org/abs/2006.15222. arXiv:2006.15222 [cs].

[19] Zhidian Zhang, Hannah K. Wayment-Steele, Garyk Brixi, Haobo Wang, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45): e2406285121, November 2024. doi: 10.1073/pnas.2406285121. URL https://www.pnas.org/doi/10.1073/pnas.2406285121. Publisher: Proceedings of the National Academy of Sciences.

[20] Elana Simon and James Zou. InterPLM: Discovering Interpretable Features in Protein Language Models via Sparse Autoencoders, January 2025. URL https://www.biorxiv.org/content/10.1101/2024.11.14.623630v2. Pages: 2024.11.14.623630 Section: New Results.

[21] Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From Mechanistic Interpretability to Mechanistic Biology: Training, Evaluating, and Interpreting Sparse Autoencoders on Protein Language Models, February 2025. URL https://www.biorxiv.org/content/10.1101/2025.02.06.636901v1. Pages: 2025.02.06.636901 Section: New Results.

[22] Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. Discovering Knowledge-Critical Subnetworks in Pretrained Language Models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6549–6583, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.376. URL https://aclanthology.org/2024.emnlp-main.376/.

[23] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.146. URL https://aclanthology.org/2021.acl-long.146/.

[24] Victor Sanh, Thomas Wolf, and Alexander M. Rush. Movement Pruning: Adaptive Sparsity by Fine-Tuning, October 2020. URL http://arxiv.org/abs/2005.07683. arXiv:2005.07683 [cs].

[25] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights, March 2018. URL http://arxiv.org/abs/1801.06519. arXiv:1801.06519 [cs].

[26] Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are Neural Nets Modular? Inspecting Functional Modularity Through Differentiable Weight Masks, March 2021. URL http://arxiv.org/abs/2010.02066. arXiv:2010.02066 [cs].

[27] Xiongyi Zhang, Jan-Willem van de Meent, and Byron Wallace. Disentangling Representations of Text by Masking Transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 778–791, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.60. URL https://aclanthology.org/2021.emnlp-main.60/.

[28] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, December 1983. ISSN 0006-3525. doi: 10.1002/bip.360221211.

[29] Steven Cao, Victor Sanh, and Alexander M. Rush. Low-Complexity Probing via Finding Subnetworks, April 2021. URL http://arxiv.org/abs/2104.03514. arXiv:2104.03514 [cs].

[30] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, August 2013. URL http://arxiv.org/abs/1308.3432. arXiv:1308.3432 [cs].

[31] Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, 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, August 2023. URL https://www.biorxiv.org/content/10.1101/2022.11.20.517210v3. Pages: 2022.11.20.517210 Section: New Results.

[32] Aya Abdelsalam Ismail, Tuomas Oikarinen, Amy Wang, Julius Adebayo, Samuel Stanton, Taylor Joren, Joseph Kleinhenz, Allen Goodman, Héctor Corrada Bravo, Kyunghyun Cho, and Nathan C. Frey. Concept Bottleneck Language Models For protein design, December 2024. URL http://arxiv.org/abs/2411.06090. arXiv:2411.06090 [cs].

[33] Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. Learning meaningful representations of protein sequences. *Nature Communications*, 13(1):1914, April 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29443-w. URL https://www.nature.com/articles/s41467-022-29443-w. Publisher: Nature Publishing Group.

[34] Quentin Fournier, Robert M. Vernon, Almer van der Sloot, Benjamin Schulz, Sarath Chandar, and Christopher James Langmead. Protein Language Models: Is Scaling Necessary?, September 2024. URL https://www.biorxiv.org/content/10.1101/2024.09.23.614603v1. Pages: 2024.09.23.614603 Section: New Results.

Supplemental Information

S1 Subnetwork Training Details

S1.1 Hyperparameters

For each subnetwork, we conducted a small-scale grid search to select optimal hyperparameter values to achieve suppression. The specific configuration used for each subnetwork is provided in the code in separate configuration files (see `https://github.com/microsoft/plm_subnetworks` for more information).

S1.1.1 Training Hyperparameters

- **Suppression lambda (λ_1):** Set at 10 for sequence-suppression and 50 for residue-suppression.
- **Maintenance lambda (λ_2):** Set at 7 for sequence-suppression and 100 for residue-suppression.
- **Maintenance MLM lambda (λ_3):** Set between (1, 1.5).

S1.1.2 Masking Hyperparameters

- **Mask temperature.** We apply a cosine decay schedule to the temperature τ, decreasing it smoothly from a maximum of $\tau_{\text{init}} = 3.2$ to $\tau_{\text{final}} = 0.01$ over 300 epochs. This schedule enables high-entropy exploration early in training and progressively sharper binarization of the mask as training progresses.
- **Mask threshold.** In practice, the binarization threshold T in Eq. (9) is set to (0.37, 0.40), which encourages sufficiently sparse and selective subnetworks while preserving gradient flow.
- **Mask initialization.** Each mask logit l_i is initialized such that $\sigma(l_i) = p_{\text{keep},0}$, where $p_{\text{keep},0}$ is the desired initial retention probability (e.g., 0.96) of the i-th parameter. This is achieved by setting $l_i = \log\left[p_{\text{keep},0}/(1 - p_{\text{keep},0})\right]$, so that each parameter begins training with a high probability of being retained. This initialization enables the model to start from a near-complete network and learn to suppress parameters selectively over time. We choose $p_{\text{keep},0} = 0.96$ to ensure that the subnetwork MLM loss initially matches the ESM-2 maintenance loss, while introducing less than 1% sparsity. This yields a mask score distribution centered near zero, placing most sigmoid activations in a high-sensitivity regime and allowing flexible updates toward maintenance and suppression goals.

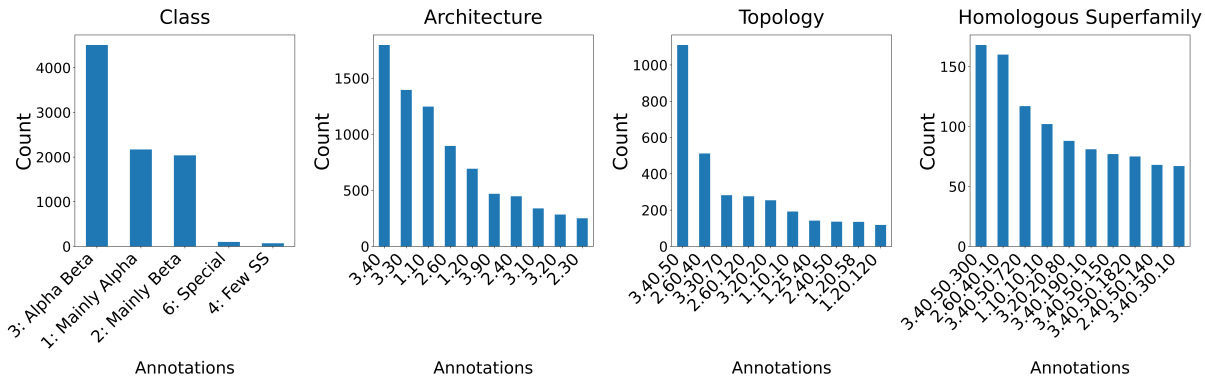
S1.2 Learning Rate

We use a maximum learning rate of `peak_lr` with a reverse OneCycle learning rate schedule. The learning rate remains constant at `peak_lr` during an initial plateau phase, defined by `1 / lr_plateau_epochs`, and then decays to `peak_lr` divided by 30 using cosine annealing. The learning rate schedule spans the full number of estimated training steps, as determined by PyTorch Lightning. Only parameters in the mask learner are optimized, using the AdamW optimizer with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-6}$, and zero weight decay. The maximum value of `peak_lr` observed in training is 10^{-1}.

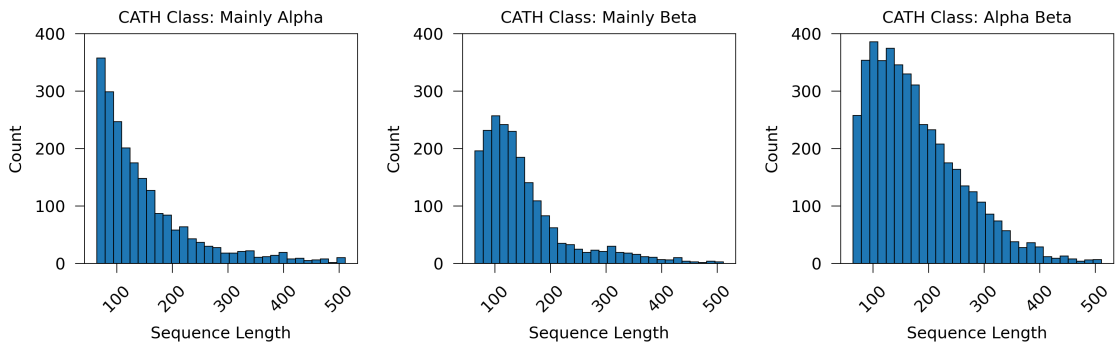
S1.3 Compute

We train each subnetwork on one NVIDIA H100 GPU. We curate training batches of size 16 with a 0.25 frequency of suppression inputs in a batch such that subnetworks training converges over fewer epochs. We report the number of suppression inputs per category and the number training steps in Table S1.

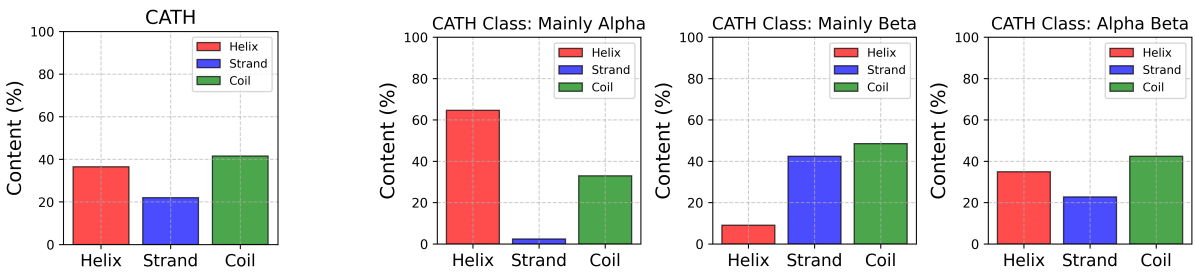
Supplemental Figures



(A) Annotation frequencies by CATH levels. Each CATH domain is annotated with a label at the Class, Architecture, Topology, and Homologous Superfamily levels. Bar plots show the counts (y-axis) of the top 10 most frequent annotations (x-axis) at each level of the CATH hierarchy.



(B) Sequence length distributions stratified by CATH level. Histograms show the counts (y-axis) of domain sequence lengths (x-axis) for each CATH Class: Mainly Alpha, Mainly Beta, Alpha Beta.



(C) Secondary structure composition of CATH domains. (Left) Average fraction of residues annotated as helix, strand, or coil across all CATH domains, based on DSSP annotations [28]. (Right) Breakdown of secondary structure content stratified by CATH class. DSSP 8-state annotations are mapped to 3-state labels: H, G, I → H; E, B → E; T, S, - → L. Helix is H, beta strand is E, and loop is L.

Figure S1. CATH annotation characteristics.

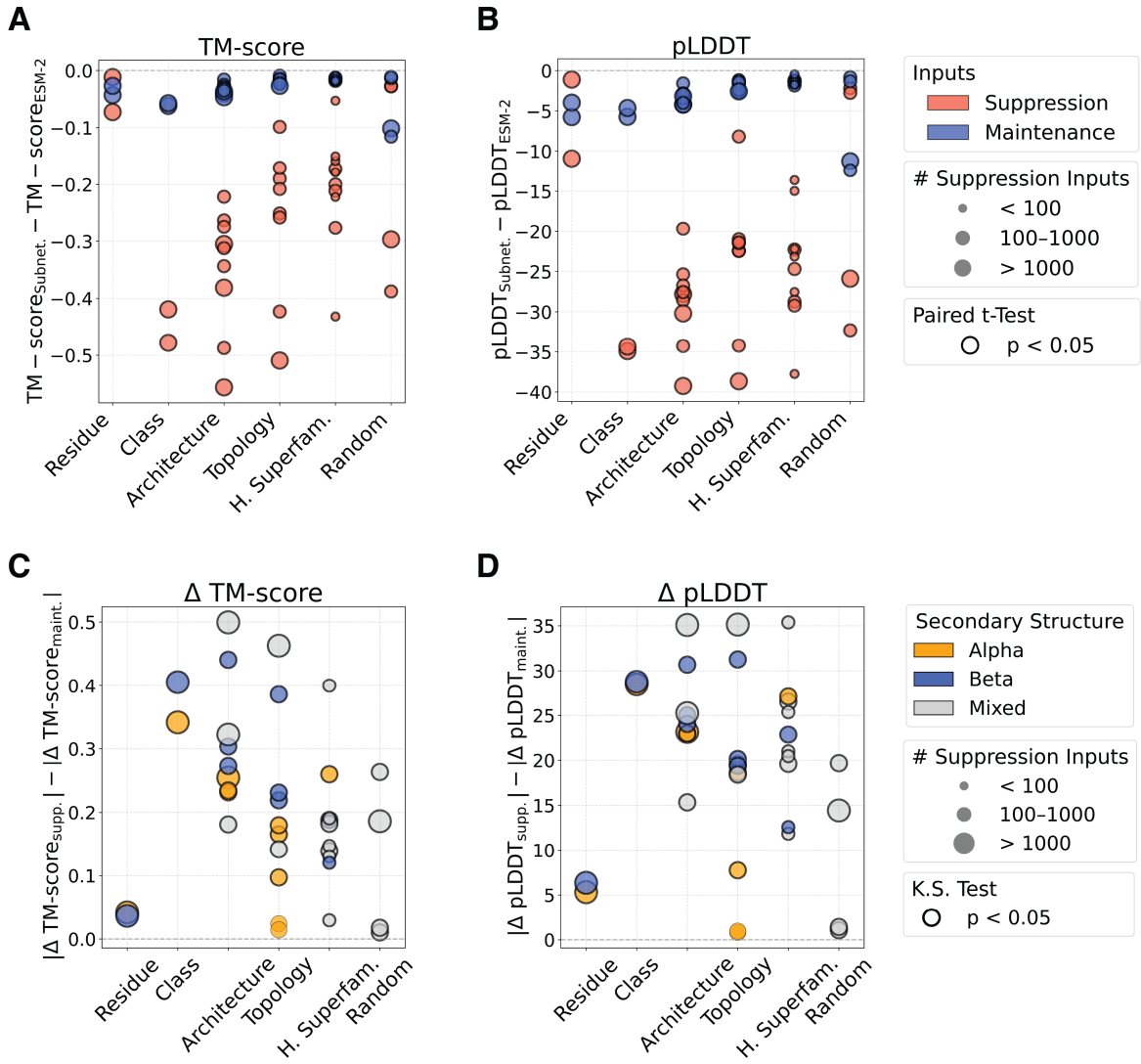


Figure S2. Effect of PLM representation factorization on predicted TM-score and pLDDT. (A–B) Structural prediction differences (y-axis) between subnetworks and the ESM-2 baseline shown for **(A)** TM-score and **(B)** pLDDT across structural levels (x-axis). Each point represents change in metrics for suppression inputs (red) or maintenance inputs (blue) for a subnetwork relative to ESM-2. Marker size indicates the number of suppression inputs for each subnetwork. Bold outlines of markers indicate statistically significant paired t-test p-values ($p < 0.05$). **(C–D)** Difference in absolute structure prediction metric changes from the ESM-2 baseline (y-axis), stratified by structural level, for **(C)** TM-score and **(D)** pLDDT. Each point corresponds to an individual subnetwork and shows the difference between suppression and maintenance Δ-values. Marker size reflects the number of suppressed inputs in the subnetwork, and color indicates secondary structure type. Bold outlines of markers indicate statistical significance by Kolmogorov–Smirnov (KS) test ($p < 0.05$).

Supplemental Tables

Table S1. Overview of subnetworks. For each subnetwork trained to suppress a structural category, we report the category, the number of sequences in the category, the learned percent sparsity (Eq. 10), the number of training steps, and predominant type of secondary structure content. We report all performance with respect to the model checkpoint that achieves the maximum difference in validation $\text{perplexity}_{\text{supp.}} - \text{perplexity}_{\text{maint.}}$ while the subnetwork $\text{perplexity}_{\text{maint.}}$ matches that of ESM-2. See Fig. S1 for residue frequencies. We provide all model configurations in our accompanying code and data (see `https://zenodo.org/records/15514844`).

Category	Target	# Seqs.	Sparsity (%)	# Training Steps	Type
Residue	Alpha Helix	—	1.005	1,568	alpha
	Beta Strand	—	0.966	1176	beta
Class	1	2,169	4.642	18,048	alpha
	2	2038	3.567	12816	beta
Architecture	1.10	1247	2.449	4,400	alpha
	1.25	182	1.325	720	alpha
	1.20	693	1.094	2,560	alpha
	2.30	251	1.058	1,536	beta
	2.40	447	1.123	2,240	beta
	2.60	896	2.189	3,200	beta
	3.30	1,396	2.651	7,320	mixed
	3.40	1797	2.196	6,400	mixed
	3.90	469	1.514	1,968	mixed
Topology	1.10.10	192	1.462	960	alpha
	1.10.287	111	1.453	704	alpha
	1.20.58	135	1.382	576	alpha
	1.20.120	118	1.360	600	alpha
	1.25.40	142	1.336	672	alpha
	2.40.50	136	1.345	576	beta
	2.60.40	512	1.433	1,656	beta
	2.60.120	276	1.241	576	beta
	3.30.70	282	1.332	1,008	mixed
	3.40.50	1110	1.302	4,608	mixed
H. Superfam.	3.40.50.300	168	1.370	672	mixed
	2.60.40.10	160	1.308	504	beta
	3.40.50.720	117	1.202	720	mixed

Category	Target	# Seqs.	Sparsity (%)	# Training Steps	Type
	1.10.10.10	102	1.366	648	alpha
	3.20.20.80	88	1.273	384	mixed
	3.40.190.10	81	1.419	576	mixed
	3.40.50.150	77	1.377	560	mixed
	3.40.50.1820	75	1.307	504	mixed
	2.40.50.140	68	1.395	480	beta
	3.40.30.10	67	1.452	672	mixed
Random Baseline	random seqs.	2,000	2.529	12,000	mixed
	random seqs.	1,000	2.255	6,048	mixed
	random seqs.	200	1.356	936	mixed
	random seqs.	100	1.336	672	mixed

Table S2. Performance of subnetworks and ESM-2 on the masked language modeling task. We perform this evaluation on the full dataset, including all train, validation, and test sequences. We do this in order to assess whether the subnetwork truly suppressed inputs that it was trained to suppress. We perform 10 inference passes at evaluation time, such that each pass produces a different random mask applied to the same sequence. We aggregate performance on all input sequences below and report the mean ± standard deviation within each category of suppression and maintenance inputs. For residue-level suppression, we report the suppression perplexity with respect to only the masked amino acid identities of the suppression type; for sequence-level suppression we average over all masked amino acid predicted identities in the sequence. The evaluation scheme is illustrated in Fig 2A. We evaluate the random baseline sequence-suppression models at the average training epoch reached by subnetworks with comparable suppression-set sizes. We do not include an additional random baseline for residue-level suppression, as randomly masking residues is equivalent to the standard MLM pretraining procedure. Thus, a random baseline for residue-level suppression is the ESM-2 performance itself on the same inputs. We provide the perplexity results of each input in each subnetwork evaluation via CSVs in our accompanying code and data (see `https://zenodo.org/records/15514844`).

Category	Target	#Seqs	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.
Residue	Alpha Helix	—	9.72 ± 5.52	5.47 ± 3.91	5.45 ± 4.01	5.43 ± 3.90
	Beta Strand	—	8.76 ± 6.99	5.97 ± 3.50	4.80 ± 5.10	5.78 ± 3.45
Class	1	2169	36.33 ± 105.75	8.69 ± 18.77	8.66 ± 16.09	8.71 ± 14.71
	2	2038	33.76 ± 88.61	8.75 ± 16.40	8.79 ± 16.29	8.67 ± 14.67
Architecture	1.10	1247	14.43 ± 38.04	9.33 ± 17.08	7.68 ± 16.36	8.86 ± 14.83
	1.20	693	16.50 ± 48.18	8.98 ± 17.26	9.10 ± 13.25	8.66 ± 15.20
	1.25	182	41.65 ± 71.07	9.12 ± 17.74	11.26 ± 18.95	8.64 ± 14.96
	2.30	251	10.74 ± 12.44	9.37 ± 17.23	4.87 ± 4.83	8.81 ± 15.24
	2.40	447	17.33 ± 39.71	9.29 ± 16.67	7.16 ± 11.07	8.78 ± 15.24
	2.60	896	20.93 ± 38.74	9.30 ± 16.78	8.73 ± 14.39	8.69 ± 15.13
	3.30	1396	16.94 ± 36.69	9.77 ± 20.09	7.22 ± 11.01	8.97 ± 15.68
	3.40	1797	40.31 ± 74.01	8.34 ± 16.90	9.14 ± 14.88	8.58 ± 15.10
	3.90	469	16.64 ± 25.56	9.12 ± 17.93	10.80 ± 19.12	8.58 ± 14.79
Topology	1.10.10	192	6.53 ± 7.23	9.52 ± 17.15	4.74 ± 7.23	8.79 ± 15.17
	1.10.287	111	5.06 ± 6.48	9.33 ± 16.04	5.30 ± 6.38	8.74 ± 15.13
	1.20.120	118	10.64 ± 14.95	9.32 ± 16.90	8.18 ± 8.87	8.70 ± 15.12
	1.20.58	135	8.83 ± 11.92	9.24 ± 16.29	8.52 ± 11.34	8.70 ± 15.11
	1.25.40	142	33.57 ± 59.49	9.37 ± 17.14	10.35 ± 18.35	8.67 ± 15.00
	2.40.50	136	8.58 ± 9.09	9.39 ± 17.07	5.22 ± 6.10	8.75 ± 15.15
	2.60.120	276	25.73 ± 40.16	9.00 ± 16.88	11.51 ± 15.73	8.61 ± 15.03
	2.60.40	512	12.74 ± 17.24	9.40 ± 17.95	6.67 ± 8.95	8.82 ± 15.34
	3.30.70	282	7.29 ± 7.05	9.25 ± 16.90	5.78 ± 6.51	8.79 ± 15.25
	3.40.50	1110	36.53 ± 60.43	8.85 ± 17.03	8.79 ± 11.26	8.69 ± 15.53
H. Superfam.	1.10.10.10	102	7.14 ± 7.50	9.26 ± 16.41	4.11 ± 4.13	8.75 ± 15.13
	2.40.50.140	68	7.43 ± 8.95	9.36 ± 17.22	4.45 ± 3.60	8.73 ± 15.11
	2.60.40.10	160	7.69 ± 6.62	9.20 ± 16.04	4.94 ± 4.38	8.77 ± 15.18
	3.20.20.80	88	56.64 ± 67.67	9.08 ± 16.61	12.33 ± 14.05	8.66 ± 15.06

Category	Target	#Seqs.	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.
	3.40.190.10	81	23.67 ± 29.67	9.43 ± 17.03	8.47 ± 8.30	8.70 ± 15.11
	3.40.30.10	67	16.47 ± 17.64	9.43 ± 16.48	4.76 ± 3.90	8.73 ± 15.11
	3.40.50.150	77	39.34 ± 58.62	9.24 ± 16.63	10.60 ± 13.86	8.68 ± 15.07
	3.40.50.1820	75	69.39 ± 95.57	9.19 ± 17.11	12.41 ± 12.44	8.67 ± 15.07
	3.40.50.300	168	31.61 ± 42.36	9.37 ± 16.46	9.28 ± 10.37	8.69 ± 15.13
	3.40.50.720	117	31.73 ± 62.95	9.26 ± 16.87	7.57 ± 8.09	8.71 ± 15.13
Random Baseline	random	100	10.75 ± 21.04	9.22 ± 17.11	8.82 ± 14.72	8.74 ± 14.39
	random	200	9.89 ± 13.32	9.24 ± 16.93	8.88 ± 15.46	8.74 ± 14.37
	random	1000	12.15 ± 21.11	8.95 ± 16.95	8.80 ± 14.89	8.73 ± 14.34
	random	2000	11.29 ± 17.63	8.40 ± 17.81	8.50 ± 13.77	8.79 ± 14.52

Table S3. Evaluation of structure prediction capabilities of the factorized PLM representations using the ESMFold (650M) folding trunk on the validation datasets. For TM-score, RMSD, and pLDDT, we report the mean ± standard deviation of the subnetwork performance across all sequences within each category of suppression and maintenance inputs. ESM-2 (650M) performance on the same categories is reported as the PLM baseline. To quantify the differences in subnetwork and PLM performance, we perform a paired t-test on all (i) suppression inputs and (ii) maintenance inputs, computing the difference in $\Delta\text{metric} = \text{metric}_{\text{Subnet.}} - \text{metric}_{\text{ESM-2.}}$. We then perform a Kolmogorov–Smirnov (KS) test on these differences to assess whether the distribution of $|\Delta_{\text{metric,supp}}|$ is significantly greater than that of $|\Delta_{\text{metric,maint}}|$. Our evaluation scheme is illustrated in Fig. 3A. We report p-values for both statistical tests for each subnetwork; significant p-values are in bold. For residue-level suppression, we evaluate structure prediction on CATH class categories of mainly alpha and mainly beta sequences as a proxy for evaluating alpha helix and beta strand performance. We report performance on the only residue-specific metric of pLDDT in Fig. 3F-G. All structure prediction metrics are provided via CSVs in our accompanying code and data (see `https://zenodo.org/records/15514844`).

Category	CATH Target	Metric	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.	p-value t-test supp.	p-value t-test maint.	p-value KS test
Residue	1	RMSD	3.37 ± 1.18	2.98 ± 1.06	2.93 ± 1.12	2.81 ± 1.08	**5.3e-47**	**6.3e-57**	**1.9e-30**
		TM-score	0.59 ± 0.20	0.69 ± 0.19	0.66 ± 0.20	0.71 ± 0.20	**1.4e-36**	**2.8e-38**	**2.6e-16**
		pLDDT	57.68 ± 14.66	62.01 ± 12.61	66.32 ± 12.80	65.34 ± 12.71	**3.5e-91**	**3.2e-164**	**1.6e-64**
	2	RMSD	3.40 ± 1.25	3.06 ± 1.10	2.92 ± 1.14	2.87 ± 1.11	**4.9e-59**	**1.2e-63**	**5.7e-33**
		TM-score	0.62 ± 0.20	0.66 ± 0.20	0.69 ± 0.19	0.69 ± 0.20	**2.0e-46**	**4.0e-49**	**4.0e-19**
		pLDDT	52.39 ± 13.16	61.65 ± 12.40	63.32 ± 13.79	65.64 ± 12.62	**9.0e-121**	**9.0e-188**	**1.4e-87**
Class	1	RMSD	5.42 ± 1.42	3.27 ± 1.04	2.88 ± 1.06	2.85 ± 1.07	**1.1e-141**	**5.5e-104**	**1.5e-171**
		TM-score	0.25 ± 0.11	0.65 ± 0.19	0.67 ± 0.20	0.71 ± 0.19	**7.6e-122**	**1.6e-97**	**7.9e-143**
		pLDDT	31.99 ± 7.51	59.27 ± 12.45	66.91 ± 12.56	65.00 ± 12.54	**1.2e-185**	**8.6e-161**	**1.9e-238**
	2	RMSD	5.97 ± 1.20	3.25 ± 1.12	2.86 ± 1.14	2.86 ± 1.11	**5.8e-162**	**2.6e-91**	**4.5e-198**
		TM-score	0.21 ± 0.08	0.64 ± 0.20	0.69 ± 0.19	0.70 ± 0.20	**8.4e-143**	**2.0e-90**	**1.5e-203**
		pLDDT	29.69 ± 5.67	61.45 ± 12.77	64.04 ± 13.14	66.10 ± 12.62	**6.5e-167**	**7.2e-133**	**1.3e-209**
Architecture	1.10	RMSD	4.69 ± 1.34	3.09 ± 1.12	2.84 ± 1.11	2.79 ± 1.08	**5.0e-58**	**3.8e-77**	**1.2e-80**
		TM-score	0.34 ± 0.16	0.67 ± 0.20	0.64 ± 0.21	0.71 ± 0.20	**5.2e-53**	**6.1e-77**	**1.9e-72**
		pLDDT	38.56 ± 12.31	61.65 ± 14.13	66.36 ± 13.09	65.86 ± 12.96	**1.1e-78**	**2.8e-104**	**1.9e-119**
	1.20	RMSD	4.73 ± 1.37	3.05 ± 1.18	3.07 ± 1.13	2.83 ± 1.12	**6.3e-09**	**4.3e-33**	**3.2e-21**
		TM-score	0.37 ± 0.19	0.67 ± 0.21	0.65 ± 0.20	0.70 ± 0.20	**2.4e-10**	**5.6e-24**	**3.6e-20**
		pLDDT	39.16 ± 13.43	62.33 ± 14.53	65.90 ± 13.75	65.59 ± 12.87	**3.5e-11**	**5.2e-69**	**2.0e-25**
	1.25	RMSD	4.47 ± 1.33	3.00 ± 1.11	2.63 ± 1.04	2.89 ± 1.12	**2.3e-32**	**5.1e-58**	**5.4e-55**

Category	CATH Target	Metric	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.	p-value t-test supp.	p-value t-test maint.	p-value KS test
		TM-score	0.52 ± 0.22	0.67 ± 0.20	0.79 ± 0.16	0.69 ± 0.20	**2.3e-28**	**7.7e-58**	**3.7e-42**
		pLDDT	47.96 ± 15.82	63.30 ± 12.87	73.32 ± 11.45	64.90 ± 12.77	**9.1e-41**	**1.9e-81**	**3.4e-71**
	2.30	RMSD	4.96 ± 1.06	3.08 ± 1.16	2.96 ± 1.02	2.91 ± 1.14	**5.7e-13**	**1.2e-50**	**3.6e-28**
		TM-score	0.31 ± 0.14	0.66 ± 0.20	0.62 ± 0.20	0.69 ± 0.20	**7.4e-13**	**1.0e-41**	**1.3e-28**
		pLDDT	33.97 ± 7.27	62.23 ± 13.97	62.54 ± 13.35	65.18 ± 13.08	**1.7e-15**	**3.0e-91**	**2.7e-30**
	2.40	RMSD	5.38 ± 1.27	3.08 ± 1.18	3.09 ± 1.16	2.88 ± 1.13	**8.5e-23**	**3.1e-62**	**3.1e-36**
		TM-score	0.29 ± 0.15	0.66 ± 0.21	0.64 ± 0.20	0.69 ± 0.20	**3.1e-22**	**9.9e-52**	**7.3e-38**
		pLDDT	33.69 ± 10.16	62.39 ± 14.06	61.24 ± 13.40	65.22 ± 13.08	**2.8e-27**	**2.6e-96**	**4.3e-51**
	2.60	RMSD	6.11 ± 1.03	3.05 ± 1.12	2.75 ± 1.08	2.83 ± 1.10	**4.3e-83**	**1.2e-65**	**2.5e-135**
		TM-score	0.23 ± 0.11	0.66 ± 0.20	0.72 ± 0.16	0.70 ± 0.20	**1.1e-83**	**6.3e-57**	**1.8e-145**
		pLDDT	29.36 ± 5.34	62.93 ± 13.22	63.65 ± 13.15	65.81 ± 12.79	**1.3e-77**	**1.0e-124**	**4.2e-137**
	3.30	RMSD	5.38 ± 1.18	3.15 ± 1.11	2.94 ± 1.14	2.84 ± 1.07	**1.1e-79**	**1.0e-85**	**2.8e-106**
		TM-score	0.27 ± 0.12	0.66 ± 0.20	0.65 ± 0.21	0.70 ± 0.20	**5.5e-78**	**1.9e-76**	**2.4e-109**
		pLDDT	33.41 ± 9.14	61.92 ± 13.29	63.61 ± 14.04	66.13 ± 12.44	**2.9e-90**	**6.6e-123**	**1.0e-122**
	3.40	RMSD	6.48 ± 1.14	3.13 ± 1.12	2.56 ± 0.95	2.88 ± 1.15	**3.3e-169**	**4.8e-63**	**8.3e-237**
		TM-score	0.23 ± 0.11	0.64 ± 0.19	0.78 ± 0.15	0.68 ± 0.20	**8.5e-159**	**1.5e-57**	**2.3e-241**
		pLDDT	29.60 ± 6.59	61.68 ± 13.29	68.85 ± 10.55	64.86 ± 13.30	**6.2e-174**	**4.7e-106**	**1.0e-247**
	3.90	RMSD	4.95 ± 1.62	3.07 ± 1.21	3.31 ± 1.26	2.84 ± 1.12	**7.4e-16**	**1.3e-83**	**4.3e-18**
		TM-score	0.42 ± 0.22	0.66 ± 0.21	0.64 ± 0.22	0.69 ± 0.20	**1.7e-15**	**3.8e-60**	**3.0e-18**
		pLDDT	40.85 ± 14.28	61.70 ± 15.10	60.53 ± 12.75	65.71 ± 13.06	**3.4e-22**	**9.4e-173**	**1.5e-25**
Topology	1.10.10	RMSD	3.52 ± 1.06	3.04 ± 1.11	2.49 ± 0.90	2.89 ± 1.11	**1.6e-07**	**1.6e-49**	**5.9e-12**
		TM-score	0.46 ± 0.17	0.67 ± 0.20	0.65 ± 0.18	0.69 ± 0.20	**4.3e-07**	**1.4e-40**	**5.0e-10**
		pLDDT	47.74 ± 14.04	62.65 ± 13.80	70.16 ± 12.34	65.36 ± 13.08	**4.2e-12**	**2.1e-118**	**2.1e-24**
	1.10.287	RMSD	2.85 ± 0.86	2.95 ± 1.08	2.70 ± 1.02	2.85 ± 1.09	1.7e-01	**2.9e-42**	2.5e-01
		TM-score	0.60 ± 0.18	0.68 ± 0.20	0.62 ± 0.19	0.70 ± 0.20	2.9e-01	**2.7e-26**	9.8e-01
		pLDDT	73.14 ± 10.41	64.02 ± 13.16	74.28 ± 10.41	65.23 ± 12.84	2.0e-01	**5.6e-73**	7.0e-01
	1.20.120	RMSD	3.87 ± 1.29	2.97 ± 1.11	3.35 ± 1.40	2.85 ± 1.12	1.4e-01	**1.6e-19**	3.0e-01
		TM-score	0.50 ± 0.22	0.68 ± 0.20	0.60 ± 0.24	0.70 ± 0.20	3.8e-01	**9.5e-17**	8.5e-02
		pLDDT	53.95 ± 15.15	63.90 ± 13.29	62.13 ± 13.63	65.43 ± 13.12	**5.4e-03**	**4.1e-101**	1.8e-01
	1.20.58	RMSD	2.64 ± 1.17	2.93 ± 1.09	2.55 ± 1.05	2.85 ± 1.09	**1.1e-02**	**3.0e-48**	**2.9e-03**

Category	CATH Target	Metric	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.	p-value t-test supp.	p-value t-test maint.	p-value KS test
		TM-score	0.69 ± 0.19	0.68 ± 0.20	0.71 ± 0.19	0.69 ± 0.20	**2.2e-02**	**1.4e-27**	**3.9e-02**
		pLDDT	69.69 ± 13.61	63.99 ± 13.04	72.30 ± 12.29	65.48 ± 13.04	**1.6e-02**	**8.8e-89**	**1.8e-03**
	1.25.40	RMSD	4.19 ± 1.14	2.97 ± 1.12	2.87 ± 1.04	2.84 ± 1.10	**2.2e-06**	**1.3e-45**	**5.3e-13**
		TM-score	0.52 ± 0.21	0.68 ± 0.20	0.73 ± 0.16	0.70 ± 0.20	**3.9e-07**	**5.4e-32**	**7.4e-15**
		pLDDT	48.87 ± 15.37	63.92 ± 13.05	69.86 ± 12.96	65.54 ± 12.58	**6.1e-08**	**4.0e-71**	**7.1e-16**
	2.40.50	RMSD	4.25 ± 1.44	2.95 ± 1.11	2.66 ± 0.85	2.83 ± 1.11	**4.4e-07**	**6.0e-40**	**1.1e-11**
		TM-score	0.44 ± 0.20	0.68 ± 0.20	0.69 ± 0.15	0.70 ± 0.20	**2.7e-06**	**8.4e-30**	**3.8e-11**
		pLDDT	42.92 ± 15.97	64.11 ± 13.05	65.38 ± 10.69	65.70 ± 12.93	**3.1e-08**	**8.3e-61**	**2.0e-15**
	2.60.120	RMSD	4.92 ± 1.83	2.94 ± 1.11	2.99 ± 1.04	2.84 ± 1.09	**1.9e-46**	**3.0e-44**	**8.3e-89**
		TM-score	0.50 ± 0.25	0.68 ± 0.20	0.76 ± 0.14	0.70 ± 0.20	**4.2e-44**	**4.6e-35**	**6.5e-75**
		pLDDT	41.24 ± 14.90	64.36 ± 13.34	62.72 ± 11.15	65.77 ± 12.87	**2.1e-50**	**6.9e-77**	**1.2e-94**
	2.60.40	RMSD	5.25 ± 1.19	3.00 ± 1.14	2.57 ± 1.01	2.85 ± 1.12	**1.3e-14**	**1.0e-40**	**3.4e-30**
		TM-score	0.32 ± 0.17	0.67 ± 0.20	0.74 ± 0.15	0.69 ± 0.20	**6.4e-14**	**1.3e-25**	**4.7e-21**
		pLDDT	32.33 ± 7.79	63.24 ± 13.73	66.56 ± 12.49	65.58 ± 12.87	**2.5e-18**	**3.1e-63**	**5.6e-44**
	3.30.70	RMSD	3.74 ± 0.90	3.01 ± 1.11	2.68 ± 1.03	2.85 ± 1.10	**8.9e-10**	**4.6e-49**	**3.6e-14**
		TM-score	0.47 ± 0.13	0.67 ± 0.20	0.64 ± 0.19	0.70 ± 0.20	**1.7e-10**	**1.4e-38**	**4.9e-13**
		pLDDT	44.27 ± 11.72	63.09 ± 13.19	65.65 ± 13.79	65.50 ± 12.77	**3.2e-11**	**1.3e-95**	**8.5e-21**
	3.40.50	RMSD	6.08 ± 1.14	3.06 ± 1.14	2.45 ± 0.81	2.87 ± 1.12	**7.2e-124**	**3.2e-40**	**1.5e-186**
		TM-score	0.29 ± 0.14	0.66 ± 0.20	0.80 ± 0.12	0.68 ± 0.20	**2.3e-115**	**1.1e-34**	**1.8e-191**
		pLDDT	31.58 ± 6.41	62.52 ± 13.45	70.25 ± 8.82	65.11 ± 13.23	**6.7e-137**	**4.7e-71**	**1.8e-220**
H. Superfam.	1.10.10.10	RMSD	3.59 ± 1.27	2.99 ± 1.12	2.05 ± 0.91	2.90 ± 1.16	**4.9e-10**	**1.6e-50**	**4.0e-14**
		TM-score	0.49 ± 0.20	0.68 ± 0.20	0.77 ± 0.14	0.69 ± 0.20	**2.9e-10**	**1.1e-32**	**3.3e-15**
		pLDDT	45.02 ± 15.02	63.94 ± 13.14	74.26 ± 12.02	65.37 ± 13.13	**8.4e-17**	**7.2e-94**	**3.9e-32**
	2.40.50.140	RMSD	3.62 ± 1.40	2.98 ± 1.13	2.79 ± 0.46	2.85 ± 1.11	**1.8e-06**	**4.2e-43**	**8.2e-13**
		TM-score	0.53 ± 0.20	0.68 ± 0.20	0.68 ± 0.10	0.70 ± 0.20	**2.7e-05**	**6.4e-30**	**1.7e-07**
		pLDDT	51.56 ± 15.70	63.67 ± 13.17	66.54 ± 7.95	65.41 ± 13.01	**9.4e-10**	**1.3e-58**	**4.2e-23**
	2.60.40.10	RMSD	3.52 ± 1.39	2.97 ± 1.13	2.30 ± 0.80	2.86 ± 1.11	**2.0e-06**	**7.2e-31**	**7.3e-12**
		TM-score	0.56 ± 0.20	0.68 ± 0.20	0.76 ± 0.11	0.70 ± 0.20	**8.8e-07**	**1.6e-25**	**1.6e-12**
		pLDDT	46.95 ± 17.01	63.97 ± 13.35	71.64 ± 10.48	65.35 ± 13.06	**4.7e-11**	**1.3e-36**	**8.9e-34**
	3.20.20.80	RMSD	3.40 ± 0.84	2.94 ± 1.14	2.77 ± 0.58	2.86 ± 1.15	**9.1e-07**	**2.1e-27**	**4.9e-16**

Category	CATH Target	Metric	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.	p-value t-test supp.	p-value t-test maint.	p-value KS test
	3.40.190.10	TM-score	0.81 ± 0.07	0.68 ± 0.20	0.87 ± 0.05	0.69 ± 0.20	**1.8e-06**	**2.4e-18**	**1.8e-11**
		pLDDT	51.85 ± 8.13	64.29 ± 13.17	65.48 ± 6.18	65.39 ± 13.08	**1.6e-08**	**4.8e-77**	**1.7e-17**
		RMSD	4.67 ± 1.44	2.98 ± 1.10	2.90 ± 1.09	2.85 ± 1.10	**3.0e-05**	**1.4e-21**	**5.9e-07**
	3.40.30.10	TM-score	0.40 ± 0.18	0.67 ± 0.20	0.63 ± 0.15	0.69 ± 0.20	**1.7e-05**	**2.9e-14**	**2.1e-08**
		pLDDT	37.28 ± 8.27	64.29 ± 13.29	64.82 ± 10.05	65.46 ± 12.99	**4.6e-10**	**9.3e-77**	**2.2e-30**
		RMSD	5.21 ± 0.87	2.97 ± 1.12	2.41 ± 0.58	2.84 ± 1.13	**2.2e-07**	**5.0e-61**	**1.2e-14**
	3.40.50.150	TM-score	0.34 ± 0.14	0.68 ± 0.20	0.77 ± 0.09	0.70 ± 0.20	**1.8e-07**	**2.1e-36**	**2.9e-16**
		pLDDT	32.32 ± 3.20	63.94 ± 13.23	70.08 ± 8.94	65.71 ± 13.06	**1.8e-11**	**6.5e-54**	**4.6e-35**
		RMSD	3.58 ± 1.55	2.99 ± 1.14	2.29 ± 0.48	2.87 ± 1.15	**1.0e-03**	**5.4e-43**	**2.1e-08**
	3.40.50.1820	TM-score	0.68 ± 0.21	0.68 ± 0.20	0.86 ± 0.07	0.69 ± 0.21	**2.5e-03**	**3.6e-33**	**8.2e-07**
		pLDDT	49.78 ± 14.07	63.91 ± 13.44	72.93 ± 6.84	65.31 ± 13.26	**3.8e-06**	**2.4e-78**	**1.4e-23**
		RMSD	3.88 ± 1.59	3.00 ± 1.14	2.38 ± 0.70	2.88 ± 1.13	**7.5e-04**	**1.3e-38**	**4.0e-07**
	3.40.50.300	TM-score	0.71 ± 0.18	0.68 ± 0.20	0.87 ± 0.07	0.69 ± 0.20	**8.1e-04**	**1.8e-32**	**6.2e-08**
		pLDDT	48.12 ± 14.61	64.57 ± 13.51	70.33 ± 7.40	65.05 ± 13.31	**1.3e-05**	**3.3e-13**	**1.9e-21**
		RMSD	3.86 ± 1.29	3.00 ± 1.12	2.65 ± 0.74	2.88 ± 1.14	**1.3e-02**	**1.2e-46**	**3.4e-05**
	3.40.50.720	TM-score	0.62 ± 0.20	0.68 ± 0.20	0.79 ± 0.13	0.69 ± 0.20	**3.4e-02**	**7.9e-36**	1.0e-01
		pLDDT	45.56 ± 12.07	63.58 ± 12.98	67.84 ± 8.49	65.43 ± 12.97	**1.8e-03**	**5.0e-102**	**1.6e-07**
		RMSD	4.08 ± 1.45	2.96 ± 1.12	2.42 ± 0.88	2.85 ± 1.10	**6.8e-05**	**1.7e-47**	**4.0e-16**
		TM-score	0.60 ± 0.22	0.68 ± 0.20	0.81 ± 0.13	0.70 ± 0.20	**2.9e-04**	**2.5e-39**	**2.5e-14**
		pLDDT	44.47 ± 13.80	64.47 ± 12.90	73.20 ± 9.51	65.65 ± 12.82	**2.4e-06**	**1.6e-95**	**2.5e-18**
Random	100 seqs.	RMSD	3.01 ± 1.19	2.97 ± 1.09	2.84 ± 1.16	2.85 ± 1.11	0.0e+00	**4.8e-146**	**7.9e-145**
		TM-score	0.68 ± 0.20	0.68 ± 0.20	0.71 ± 0.19	0.70 ± 0.20	0.0e+00	**3.0e-139**	**8.2e-141**
		pLDDT	62.92 ± 13.61	63.75 ± 12.98	65.69 ± 13.07	65.54 ± 12.91	0.0e+00	**5.4e-250**	**1.2e-166**
	1000 seqs.	RMSD	5.49 ± 1.30	3.66 ± 1.41	2.89 ± 1.13	2.85 ± 1.11	**1.4e-320**	**1.3e-173**	**2.6e-189**
		TM-score	0.29 ± 0.13	0.58 ± 0.23	0.68 ± 0.21	0.70 ± 0.20	**5.7e-295**	**6.7e-168**	**2.4e-186**
		pLDDT	32.78 ± 5.70	52.82 ± 15.74	65.09 ± 13.25	65.59 ± 12.87	0.0e+00	**2.0e-286**	**5.7e-198**
	200 seqs.	RMSD	2.89 ± 1.03	2.96 ± 1.08	2.73 ± 1.01	2.85 ± 1.11	**2.7e-08**	**2.1e-32**	**1.3e-02**
		TM-score	0.70 ± 0.19	0.68 ± 0.20	0.73 ± 0.18	0.69 ± 0.20	**5.6e-06**	**1.4e-18**	**3.6e-02**
		pLDDT	65.03 ± 12.71	64.43 ± 12.88	67.21 ± 11.96	65.50 ± 12.93	**6.2e-10**	**1.5e-36**	**5.8e-06**
	2000 seqs.	RMSD	4.79 ± 1.55	3.58 ± 1.33	2.79 ± 1.09	2.87 ± 1.11	**5.9e-05**	**2.3e-24**	**6.2e-04**

Category	CATH Target	Metric	Subnet. Supp.	Subnet. Maint.	ESM-2 Supp.	ESM-2 Maint.	p-value t-test supp.	p-value t-test maint.	p-value KS test
		TM-score	0.41 ± 0.21	0.59 ± 0.21	0.70 ± 0.20	0.69 ± 0.20	**1.4e-04**	**1.7e-15**	**2.3e-02**
		pLDDT	40.23 ± 13.68	54.17 ± 14.81	66.12 ± 12.73	65.37 ± 12.96	**1.6e-09**	**2.0e-90**	**6.3e-04**