Feature Reuse and Scaling: Understanding Transfer Learning with Protein Language Models

Francesca-Zhoufan Li [1] [2] **Ava P. Amini** [3] **Yisong Yue** [4] **Kevin K. Yang** [3] **Alex X. Lu** [3]

Abstract

Large pretrained protein language models (PLMs) have improved protein property and structure prediction from sequences via transfer learning, in which weights and representations from PLMs are repurposed for downstream tasks. Although PLMs have shown great promise, currently there is little understanding of how the features learned by pretraining relate to and are useful for downstream tasks. We perform a systematic analysis of transfer learning using PLMs, conducting 370 experiments across a comprehensive suite of factors including different downstream tasks, architectures, model sizes, model depths, and pretraining time. We observe that while almost all downstream tasks do benefit from pretrained models compared to naive sequence representations, for the majority of tasks performance does not scale with pretraining, and instead relies on low-level features learned early in pretraining. Our results point to a mismatch between current PLM pretraining paradigms and most applications of these models, indicating a need for better pretraining methods.

1. Introduction

Proteins perform a myriad of critical biological functions, and thus the ability to design proteins has vast impacts on healthcare, environment, and industry (Lutz & Iamurri, 2018). Since a protein's function is largely determined by its amino acid sequence, specifying a sequence that will yield a desired function is feasible in principle. However, the relationship between amino acid sequence and function remains poorly understood, and most experimental methods for measuring function are costly and low-throughput (Maynard Smith, 1970; Romero & Arnold, 2009). To overcome the challenge presented by limited labelled data, researchers have sought to use transfer learning, in which models are pretrained in a self-supervised fashion on large public datasets in the hope that the pretrained features or model weights will improve performance on downstream tasks where supervised data is limited (Fig. 1a-b).

Protein language models (PLMs) have emerged as the most popular framework for transfer learning for proteins (Rives et al., 2021; Yang et al., 2022; Elnaggar et al., 2022; Brandes et al., 2022; Alley et al., 2019; Elnaggar et al., 2023; Lin et al., 2023). Most PLMs pretrain using the masked language modeling (MLM) task, in which the model is trained to predict the original identity of masked or corrupted amino acids. PLMs have been effective at improving performance on many protein function prediction tasks, and some are now integrated into bioinformatics and structure prediction tools (Teufel et al., 2022; Thumuluri et al., 2022; Wu et al., 2022; Flamholz et al., 2024). Despite their widespread adoption, it is not understood how or why PLMs improve performance on downstream tasks.

Drawing from other domains like computer vision where investigations of transfer learning are more established, we synthesize a set of possible hypotheses to explain improvement in downstream tasks, and design and conduct a comprehensive series of experiments to test them. We structure our study around the following hypotheses:

Feature reuse (Fig. 1c-i). One popular hypothesis is that MLM pretraining learns general features of protein biology, and that these features can be re-used across tasks. Previous work has shown that transfer learning improves performance across diverse downstream tasks (Rao et al., 2019; Dallago et al., 2021). However, the degree of feature reuse is also important: ideally, the pretrain and downstream tasks should be *aligned*, such that transferring PLM representations improves downstream function prediction accuracy and that this improvement increases with larger model sizes, deeper layers, and better pretraining performance.

[1] Department of Bioengineering, California Institute of Technology, California, USA [2] Initial work was conducted while F.Z.L. was an intern at Microsoft. [3] Microsoft Research, Massachusetts, USA [4] Department of Computing and Mathematical Sciences, California Institute of Technology, California, USA. Correspondence to: Francesca-Zhoufan Li <fzl@caltech.edu>, Alex X. Lu <lualex@microsoft.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

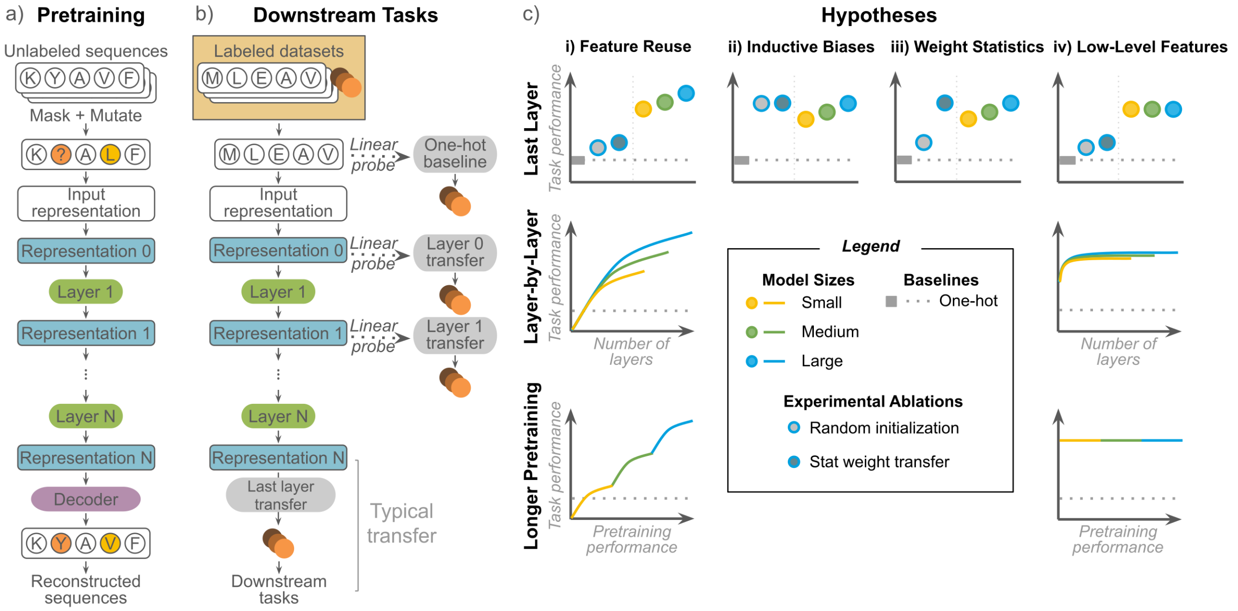

Figure 1: Summary of the transfer learning procedure and our analyses. a) PLMs are pretrained using masked language modeling. b) Typically, transfer learning uses respresentations from the last layer of the PLM for downstream tasks. We evaluate downstream task performance at every layer in the model. c) We compare to baselines and ablations and evaluate the effects of PLM size, model depth, and pretraining time. These experiments characterize behavior consistent with either feature reuse (i), or an alternative hypothesis (inductive biases/overparameterization - ii, weight statistics - iii, or reuse of low-level features only - iv).

If this does not occur, it suggests that pretraining primarily learns features that cannot be reused on downstream tasks. To determine whether or not this is the case for PLMs, we explore three alternative hypotheses.

Inductive biases and overparameterization (Fig. 1c-ii). The large number of parameters in pretrained models may lead to some alignment with useful signals by chance (Raghu et al., 2019). If inductive biases are sufficient, then transferring from randomly-initialized version of the same model architecture should perform similarly.

Statistics of pretrained weights (Fig. 1c-iii). The primary benefit of pretraining may be initializing weights to a sensible scale (Raghu et al., 2019; Matsoukas et al., 2022). If pretraining primarily provides better weight initialization, resampling weights from the empirical distribution after pretraining should provide similar performance.

Reuse of low-level features (Fig. 1c-iv). It is possible for only less complex features learned early in pretraining to contribute to transfer learning (Neyshabur et al., 2021). If low-level features are sufficient, then features extracted from earlier layers of the pretrained model may provide better or similar performance to those extracted from the last layer. Similarly, earlier pretraining checkpoints or smaller, less performant models should provide similar performance to the full-size, fully-pretrained model.

Critically, while all three alternative hypotheses can still lead to improvements in downstream task performance, they do not predict that downstream task performance can be improved by transferring representations from larger, better-trained models (Raghu et al., 2019; Abnar et al., 2022).

Contributions

Our work evaluates the scalability of transfer learning for PLMs and makes the following contributions:

1. The most comprehensive evaluation, to date and to the best of our knowledge, of transfer learning with PLMs, spanning 370 experiments over a diverse suite of downstream tasks.
2. The discovery that current MLM pretraining paradigms underserve many aspects of protein biology, as supported empirically by evidence from both structure and function prediction tasks.
3. Systematic evidence that performance on many protein property prediction tasks does not scale with PLM size or pretraining. Our results uncouple improvements in downstream performance from scaling properties.

Together, our results predict that scaling PLMs with current MLM pretraining paradigms may not scale performance on many protein function prediction tasks, but provides an evaluation framework for identifying if future pretraining efforts are scalable across more aspects of protein biology.

2. Related Work

2.1. Pretrained Protein Language Models

While numerous pretrained PLMs have been proposed in the past few years (Rives et al., 2021; Yang et al., 2022; Elnaggar et al., 2022; Brandes et al., 2022; Alley et al., 2019; Rao et al., 2019; Elnaggar et al., 2023; Lin et al., 2023), these works primarily focus on validating that pretraining improves performance on downstream tasks. In contrast, our work primarily seeks to understand the factors impacting transfer learning, which have not been rigorously studied to date for PLMs. Most PLM studies include comparisons to models with randomly initialized weights (Rives et al., 2021; Yang et al., 2022) to confirm that pretrained models do not improve downstream task performance due to overparameterization or inductive biases alone. Other studies show that under some circumstances, PLMs yield no detectable improvement over a simple one-hot representation of sequences (Wittmann et al., 2021; Hsu et al., 2022; Dallago et al., 2021). Compared to these individual baselines and benchmarks, our paper conducts a systematic analysis over many different factors impacting transfer learning.

The most similar work to ours is Detlefsen et al. (2022), which analyzes the effects of model architecture, fine-tuning, and different pooling schemes on transfer learning performance. However, we use MLMs trained on complete sequences instead of autoregressive models trained on Pfam domains. While they train proprietary, unreleased models for analysis, we use established models in the public domain. This makes our analysis more relevant to applications currently using these models and also improves documentation around these models. For example, neither their paper nor their released code describes the pretrained models in detail, so it is uncertain what the size of their model is, whereas we systematically vary the model size. More importantly, we evaluate a larger and more diverse set of downstream tasks with experiments designed to differentiate possible mechanisms by which transfer learning improves performance on downstream tasks. Critically, our systematic analysis identifies cases where transfer from PLMs is empirically effective in improving downstream task performance but the improvement is due to factors that are not expected to scale with further pretraining or larger models. However, to the extent that our analyses reach similar conclusions (e.g. both our studies observe that performance on the pretraining task does not always correlate with downstream task performance), we view our work as complementary: Detlefsen et al. (2022) use different architectures, pretraining tasks, and pretraining datasets than us, suggesting that our observations are general across more factors than what either paper analyzes independently.

2.2. Understanding Transfer Learning

While our analysis is differentiated as we focus on protein sequences, we take inspiration from computer vision studies that have sought to understand factors underlying successful transfer learning. Many are motivated by the observation that ImageNet-trained models are effective when transferred to medical images, raising the question of whether transfer performance is really due to reuse of features (given the extreme mismatch in domain), or due to more trivial factors. Raghu et al. (2019) compare pretrained models against random initialization to demonstrate that in some situations transfer performance is due to overparameterization. By randomly initializing models to match the weight statistics of pretrained models, the authors further demonstrate that improvements from pretraining may arise from good weight scalings rather than learning reusable features. Similarly, He et al. (2019) show that hyperparameter tuning can often explain improvements from transfer learning. By scrambling input images, Neyshabur et al. (2021) show that improvements from transfer learning can at least partially be attributed to the pretrained models learning low-level statistics of data rather than more sophisticated feature use. Matsoukas et al. (2022) further demonstrate that these factors vary depending upon downstream task dataset and model architecture.

Beyond models pretrained on ImageNet, some papers have looked at factors more specific to self-supervised pretraining. Abnar et al. (2022) show that improvements on the self-supervised pretraining task do not necessarily translate to improved performance on downstream tasks, and in some cases, are even anti-correlated. Pioneering work in generative self-supervised models also demonstrates that these models often saturate in downstream task performance in an intermediate layer of the model and degrade after (Jing & Tian, 2020). This is reinforced by empirical studies showing that the representations learned by self-supervised models versus supervised models rapidly diverge in the last few layers (Grigg et al., 2021), consistent with previous observations that later layers may be more specialized for the original task (Yosinski et al., 2014), underscoring the importance of a layer-by-layer evaluation.

Beyond computer vision, transfer learning has been extensively studied in natural language processing, and particularly in "BERTology" (Rogers et al., 2021), which seeks to understand what self-supervised Transformer models learn and their transferability. Among other factors, studies have similarly analyzed the effects of pretraining (Kovaleva et al., 2019), overparameterization (Gordon et al., 2020), and layer-by-layer content (Lin et al., 2019).

3. Datasets and Pretrained Models

To understand why and when transfer learning with PLMs improves downstream performance and how the improvements scale with increasingly large PLMs, we conducted 370 experiments on a diverse suite of downstream tasks with PLMs of different sizes, architectures, and at different checkpoints in training. The downstream tasks are summarized in Tables 1 and A1.

3.1. Downstream Tasks

We test a diverse set of tasks covering both property and structure prediction, different types of distribution shift relevant to protein engineering, and global versus local variation over the sequence (Supplementary section B.2).

Structure prediction. We use the three-class secondary structure (SS3) task from TAPE with three independent test sets, SS3 – CB513 (Cuff & Barton, 1999), SS3 – TS115 (Yang et al., 2016), and SS3 – CASP12 (Moult et al., 2018), where the objective is to predict whether each residue belongs to an α-helix, β-strand, or coil (Rao et al., 2019).

Property prediction. We use the thermostability, subcellular localization, GB1, and AAV datasets from FLIP (Dallago et al., 2021).

Thermostability and subcellular localization are global protein properties measured for sequences spanning different functional families and domains of life. The thermostability dataset measures the melting temperature of 48,000 proteins across 13 species (Jarzab et al., 2020). Subcellular localization is a classification task predicting the cell compartment to which a eukaryotic protein localizes (Armenteros et al., 2017; Stärk et al., 2021).

In contrast, the GB1 and AAV datasets measure the effects of local sequence variation. GB1 is the 56 amino-acid B1 domain of protein G, an immunoglobulin-binding protein. The GB1 dataset covers binding measurements for simultaneous mutations of up to 4 interactive sites (Wu et al., 2016). VP1 is an adeno-associated virus (AAV) capsid protein, over 700 amino acids long (Bryant et al., 2021). The AAV dataset measures the effects of sparsely sampled mutations across a contiguous 28 amino-acid region over the binding interface on viral viability.

For GB1 and AAV, FLIP provides different train-test splits with different distribution shifts, including sampled (in-distribution) and out-of-distribution splits, as described in Table A1. Out-of-distribution splits more closely resemble protein engineering applications where a few low-functioning variants with a limited number of mutations are initially generated, but high-functioning variants across the larger sequence space are the engineering end goal. For GB1, we test three splits, in order of increasing difficulty:

- **Sampled:** Sequences randomly partitioned between 80% training and 20% testing.
- **Low vs high:** Models are trained on mutants with function worse than the parent and tested on those with better function.
- **Two vs rest:** Models are trained on single and double mutants and tested on triple and quadruple mutants.

For AAV, we test two splits, in order of increasing difficulty:

- **Two vs many:** Models are trained on single and double mutants and tested on variants with three or more mutations.
- **One vs many:** Models are trained on single mutants and tested on variants with more mutations.

3.2. Transfer Learning with Protein Language Models

While a number of pretraining tasks have been proposed for protein sequences, we focused on models trained using the popular BERT (Devlin et al., 2019) masked language modeling (MLM) task. During pretraining, 15% of tokens are randomly selected. Of the 15%, 10% are replaced with a special masking token, 2.5% are randomly changed to another token, and the remaining 2.5% are unperturbed to encourage the model to preserve the input sequence. The corrupted sequence is passed to the model, which is trained to maximize the probability of the original tokens at the selected locations.

To evaluate the effect of model architecture, we chose two families of protein MLMs with comparable model sizes trained on UniRef50 (Suzek et al., 2015): the ESM (Rives et al., 2021) family of transformer models and the Convolutional Autoencoding Representations of Proteins (CARP) (Yang et al., 2022) family of convolutional models (Supplementary section B.1). Due to the sequence length limit of the ESM-1b transformer model, the first and last 511 amino acids were taken for all sequences exceeding 1022 amino acids. This length restriction chiefly impacts the subcellular localization dataset: targeting signals often occur at the N- or C-terminal, and we reason that taking both terminals preserves biologically-relevant signals.

Following standard protein transfer learning practice when resources for full finetuning are not available (Dallago et al., 2021), we pass representations from each PLM layer to a linear model and compare the performance to a linear model on the one-hot encoding of the sequence for each task (Fig. 1b). In addition to linear models, we also tested a learned attention pooling followed by a shallow multi-layer perceptron. However, we found last layer performance to be inferior to the linear models across almost all downstream tasks (Supplementary Figure A1), so we focus on the linear models in our analyses. For the SS3 and subcellular localization tasks, we train linear classifiers with mini-batches

Table 1: Summary of downstream prediction tasks

Dataset	Description	Tasks	Task type
SS3	Secondary structure	CB513, TS115, CASP12	Residue-level classification
Thermostability	Melting temperature	Thermostability	Regression
Subcellular localization	Cellular location	Subcellular localization	Classification
GB1	Immunoglobulin binding	Sampled, low vs. high, two vs. rest	Regression
AAV	Viral viability	Two vs. many, one vs. many	Regression

in PyTorch and perform early stopping based on the validation set. For the regression tasks, we train ridge regression models with Scikit-learn (Buitinck et al., 2013), using a grid search on the validation set to tune the regularization strength. For all tasks except secondary structure prediction, we mean pool the representations over the length dimension from each layer. Secondary structure prediction requires a representation for every residue, so no pooling is performed (Supplementary section B.3).

As protein engineers often seek to identify top-ranked mutants as opposed to predicting the absolute function of mutations, we use ranking metrics, Spearman's rank correlation and Normalized Discounted Cumulative Gain (NDCG), as the primary metrics for the regression tasks. We report Spearman's rank correlation for regression tasks and accuracy for classification tasks in the main text. Additional metrics, including mean square error, cross-entropy loss, and ROC-AUC are in the Supplemental Materials.

4. Experimental Setup

Baseline and ablations. We conduct baselines and model ablations to determine when transfer learning improves downstream task performance and whether improvements in downstream task performance can be attributed to mechanisms other than feature reuse (Fig. 1b-ii and 1b-iii).

- **One-hot baseline (■).** To determine whether transfer learning with PLMs improves performance, we test if representations from pretrained models perform better than a one-hot representation.
- **Random init (◎).** To evaluate whether the effect of transfer learning is due to overparameterization and/or the inductive biases of the PLM architecture, we test the impact of randomly initialized weights.
- **Stat transfer (●).** To evaluate whether the effect of transfer learning is due to weight statistics and/or initializing the weights to a sensible scale, we test the impact of randomly initialized weights matching the weight distribution of the pretrained PLM by randomly permuting the pretrained weights.

For random init and stat transfer, we initialize models with 3 random seeds. We consider transfer learning from a PLM to improve performance over baselines if it has a one-tailed p-value < 0.05 in a one-sample t-test.

Scaling experiments. To further understand if the MLM pretraining task is aligned with downstream tasks, we sought to understand if improving PLM performance by scaling across three factors also improves transfer learning performance on downstream tasks (Fig. 1b-iv):

- **Model size.** For both CARP (●) and ESM (✖), we test models with different numbers of layers and parameters (Table A2). For concision, we refer to CARP-38M and ESM-43M as the "small" models (🟡), CARP-76M and ESM-85M as the "medium" models (🟢), and CARP-640M and ESM-650M (ESM-1b) as the "large" models (🔵).
- **Model depth.** For each architecture (CARP: —, ESM: - -) and model size (— — —), we test whether downstream task performance improves as we transfer deeper layers by determining whether the Spearman rank correlation between layer number and performance is greater than 0.9 (Table A6). This experiment allows us to understand if tasks primarily reuse low-level features early in the pretrained models, or if more complex features deeper in the models also contribute to downstream task performance. Convolutional neural networks (CNNs) induce a stronger correlation between the depth of the layer and the complexity of the features than transformers, leading to different patterns of feature reuse in previous transfer learning studies (Matsoukas et al., 2022). However, we find little empirical difference between CNNs (CARP) and transformers (ESM) in our analyses.
- **Model checkpoint.** For each model size (🟡 🟢 🔵), we test the effect of using checkpoints from earlier in pretraining. We order these checkpoints based upon their pretraining performance (perplexity, calculated on a held-out test set of 210k sequences from Uniref50 not used to train CARP), as earlier checkpoints have higher losses on the MLM pretraining task (Table A7). We evaluate whether features from later in pretraining improve transfer learning by determining whether the Spearman rank correlation between the negative pretrain loss and downstream performance is greater than 0.9 (Table A8). Unfortunately, checkpoints are

only publicly available for CARP, so we cannot run this analysis with ESM.

Estimating error for our scaling experiments is infeasible as it would require re-training multiple PLMs from scratch. Thus, we chose arbitrary thresholds, which may impact the way we have categorized downstream tasks in our interpretation. For transparency, we plotted performance for all PLMs in each of our scaling experiments in Figures 3, 4, and 5.

We define the MLM pretraining task to be *aligned* with a downstream task if transferring PLM representations improves downstream task performance over the baseline and ablations *and* this improvement scales with improvements to pretraining. Code for all experiments is available at `https://github.com/microsoft/protein-transfer`

	SS3 - CB513	SS3 - TS115	SS3 - CASP12	Subcellular localization	Thermostability	AAV - two vs many	GB1 - low vs high	GB1 - sampled	AAV - one vs many	GB1 - two vs rest
Transfer > One-hot	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Transfer > Random init	✓	✓	✓	✓	✓	✓	✓	✓	~	~
Transfer > Stat transfer	✓	✓	✓	✓	✓	✓	✓	✓	~	✓
Scale with PLM sizes	✓	✓	✓	✓	✓	✓	✗	~	✗	✗
Scale with layer depths	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Scale with pretrain losses	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗

Figure 2: Downstream task result summary. ✓ indicates true, ∼ indicates true for only one architecture, and ✗ indicates false.

5. Results

Overall, our analyses reveal three clusters of transfer learning behavior across downstream tasks (Figure 2). First, we find that within our set of benchmarks, secondary structure prediction tasks are the only tasks where pretraining improves downstream performance and the pretrain and downstream tasks are aligned. Second, we observe that transfer learning improves performance for many downstream tasks despite the pretrain and downstream tasks not being well-aligned, indicating that performance on these tasks may not improve as PLMs scale on the axes we tested. Third, we observe that although transfer learning improves performance on almost all downstream tasks, for some tasks this improvement can be attributed to overparameterization, inductive biases, or sensible weight initialization. In subsequent sections, we expand on each of these clusters of observations in detail.

5.1. Structure Prediction Benefits from Transfer Learning Because It Is Well-Aligned with MLM pretraining

For all three residue-level secondary structure prediction tasks, Fig. 3a and Table A3 show that PLM embeddings outperform the one-hot baseline as well as the random init and stat transfer ablations, demonstrating that transfer learning improves secondary structure prediction performance and that the improvement is not due to the inductive biases or weight statistics of the models. Secondary structure prediction performance improves when transferring deeper PLM features (Fig. 3b), indicating that more complex features from later layers continue to improve performance. Furthermore, transfer learning with features from larger models and from later in pretraining improve secondary structure prediction (Fig. 3a and 3c), as previously observed by Rives et al. (2021), Elnaggar et al. (2022), and Yang et al. (2022). We therefore conclude that MLM pretraining is well-aligned to structure prediction, allowing PLM features to be reused when predicting secondary structure from sequence.

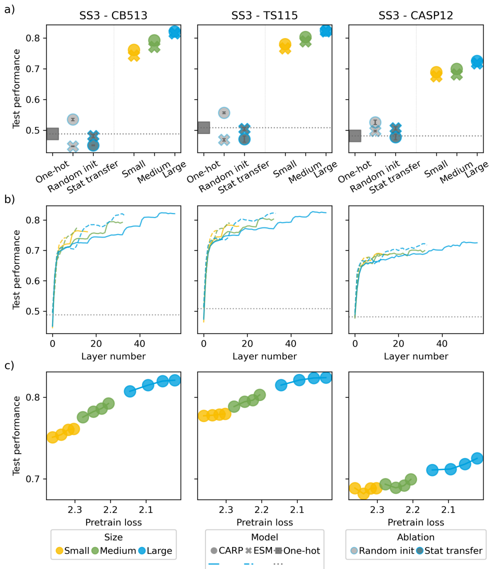

Figure 3: Results for secondary structure prediction. a) Performance on downstream tasks when transferring the final layer representation from various sizes of ESM and CARP compared to baselines and ablations. b) Downstream task performance by depth of layer transferred. c) Downstream task performance by pretraining loss. Each dot is a model checkpoint. For all subplots, downstream task test performance is quantified using accuracy.

5.2. Many Tasks Benefit from Transfer Learning Despite Lack of Alignment with MLM Pretraining

Next, we observe a cluster of five downstream tasks (subcellular localization, thermostability, AAV – two vs many, GB1 – low vs high, and GB1 – sampled) where transfer learning improves performance over baselines even though the tasks do not align well with the pretraining task (Fig. 4). For these tasks, transfer learning improves performance over both the random init and stat transfer ablations, indicating that transfer learning confers at least some benefit over the inductive biases, parameterization, or weight statistics of the models alone (Fig. 4a and Table A4). However, for all of these tasks, downstream task performance does not improve as features from deeper layers are transferred (Fig. 4b) or as the PLMs improve their pretraining loss over checkpoints (Fig. 4c), suggesting that these tasks may rely upon low-level features learned early in pretraining. Notably, the performance on these tasks typically saturates below fully fine-tuned models of the same architecture (and always below the the best-performing model for each task) trained by Yang et al. (2022) (Table A4), indicating that the saturation is not because downstream task performance has hit an upper bound.

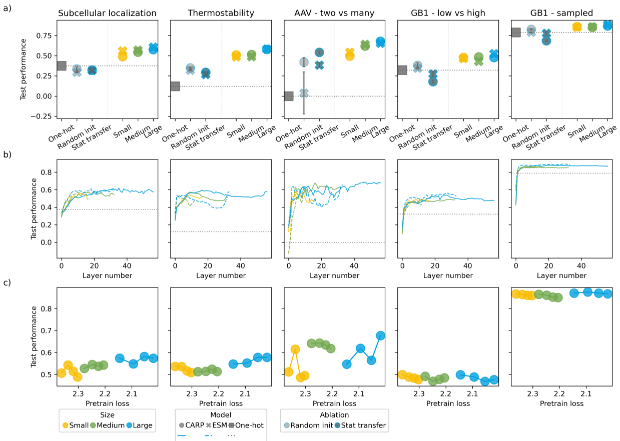

Figure 4: Results for tasks where transfer learning improves downstream task performance, but the pretrain and downstream tasks are not aligned. a) Performance on downstream tasks when transferring the final layer representation from various sizes of ESM and CARP compared to baselines and ablations. b) Downstream task performance by depth of layer transferred. c) Downstream task performance by pretraining loss. Each dot is a model checkpoint. For all subplots, downstream task test performance is quantified using Spearman's rank correlation.

To supplement our quantitative cut-offs for alignment, we qualitatively assess trends in layer-by-layer performance across tasks. We observe that for all tasks where transfer learning improves performance over the baselines (including the secondary structure prediction tasks), the largest gains in performance occur in the first 3-5 layers of both the ESM and CARP models, across model sizes (Fig. 3b and 4b). However, unlike the secondary structure prediction tasks, which continue to improve in performance past this initial peak, improvement on the downstream tasks in this cluster generally plateaus (e.g. for the GB1 – low vs high task), supporting our interpretation that features contributing to these tasks are already present within the first few layers of pretrained PLMs.

Interestingly, although none of these tasks scale with model depth or pretraining loss, three downstream tasks (subcellular localization, thermostability, and AAV – two vs many) scale with PLM size (Fig. 4a). We reasoned that while our random init ablation rules out that improvements in downstream task performance is entirely due to parameterization, parameterization may still partially contribute to performance independently of feature reuse. To test this, we additionally evaluated the performance of small and medium randomly initialized models. Indeed, we observe that both types of randomly initialized models scale in performance with ESM-1 model sizes for two of the tasks, and in similar proportions to the improvements for the pretrained models (Table A4). Together, this suggests observing downstream task performance scales with model size alone is not sufficient to conclude that pretraining and downstream tasks are aligned, and that demonstrating scaling across other axes (such as model depth and checkpoints in training, as we propose here) is necessary.

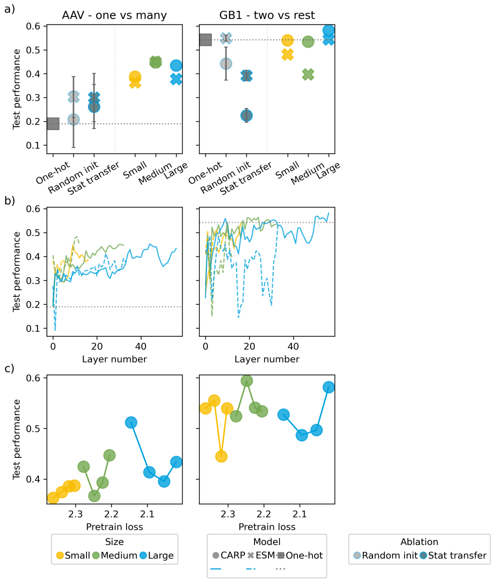

Figure 5: Results for task where pretraining does not improve downstream task performance. a) Performance on downstream tasks when transferring the final layer representation from various sizes of ESM and CARP compared to baselines and ablations. b) Downstream task performance by depth of layer transferred. c) Downstream task performance by pretraining loss. Each dot is a model checkpoint. For subcellular localization, the downstream classification task performance is quantified using accuracy. For other tasks, the downstream regression task performance is quantified using Spearman's rank correlation.

5.3. Some Tasks Do Not Benefit from MLM Pretraining

Finally, we observe two downstream tasks (AAV – one vs many, and GB1 – two vs rest) where pretraining does not improve transfer learning performance (Fig. 5). For, the AAV – one vs many task, although transfer learning improves over a one-hot representation, pretrained models do not significantly outperform randomly initialized models, suggesting that the improvement can be entirely attributed to inductive biases and parameterization. In contrast, transfer learning fails to outperform a one-hot representation on the GB1 – two vs rest task (Fig. 5a and Table A5). We hypothesize that GB1 – two vs rest is too challenging for any pretrained model, given it is an out-of-distribution split with only about 400 training samples.

Intriguingly, our stats transfer ablation decreases performance for all GB1 tasks, including the GB1 – low vs high task in the previous section, compared to the one-hot and random initialization baselines (Fig. 4b and 5b; Tables A4 and A5). We hypothesize that this is because the GB1 dataset is a highly local task, depending on finding interactions between just four mutated positions in a sequence.

6. Discussion

In this work, we systematically evaluate the mechanisms via which transfer learning from large pretrained protein language models improve performance on downstream protein function and structure prediction tasks. While most downstream tasks benefit from transfer learning, of the tasks we evaluated, structure prediction is the only task where we observe pretrain-downstream alignment. Our results are consistent with previous studies that show MLM pretraining imparts information about protein structure. Previous work has shown that the attention matrices in pretrained PLMs recapitulate contact maps (Vig et al., 2020; Rao et al., 2021), that it is possible to extract contact maps by perturbing the inputs to PLMs (Zhang et al., 2024), and that PLM representations contain similar co-evolution information as multiple sequence alignments (Chowdhury et al., 2022; Lin et al., 2023; Wu et al., 2022).

Other works have argued that larger models will not benefit fitness prediction when using zero-shot likelihoods from generative models (Nijkamp et al., 2022; Weinstein et al., 2023), as some degree of misspecification is important for generalizing from natural protein distributions to mutant variants. Our results, which show that fitness prediction tasks do not scale with pretraining, are consistent with these prior works, but we show this holds true even when transferring embeddings and even with some degree of fine-tuning. At the same time, by showing that structural prediction tasks do scale with pretraining, our results suggest that these prior results may not be general past fitness prediction.

Our primary contribution is showing that scaling pretraining does not improve performance on prediction tasks that are less reliant on coevolutionary patterns, and that outperforming the one-hot and randomly initialized baselines does not imply that downstream task performance will scale with pretraining performance. By providing the observation that current PLMs trained on MLM fail to scale on many downstream tasks, we provide a means for future models to improve on these limitations: we believe that assessing different pretraining tasks, architectures, datasets, and fine-tuning methods with our evaluation framework may produce insights on how to build more general models.

Limitations. There are factors known to impact transfer that we could not test for PLMs due to a lack of public models or computational expense. First, pretraining dataset is important, both in terms of distance between the pretraining and downstream task data domains (Cherti & Jitsev, 2022) and data size (Abnar et al., 2022). PLMs pretrain on large databases of natural sequences. In principle, this means that some downstream tasks may be out-of-distribution (e.g. those involving artificial variation or non-natural function), or subject to biases in data collection (e.g. taxonomies less-represented in UniProt (Consortium, 2019)). Previous studies have shown differences in pretraining performance by taxonomy (Almagro Armenteros et al., 2020), and that model likelihoods are biased towards more frequent species in UniProt (Ding & Steinhardt, 2024). Meier et al. (2021) trained versions of ESM on UniRef100 instead of UniRef50, and Dallago et al. (2021) show that they perform very similarly on function prediction tasks. However, subsampling pretraining sequence datasets has not been explored beyond downsampling redundant sequences, making the impact of data difficult to evaluate for pretrained PLMs.

Second, a variety of other pretraining tasks have been proposed for protein transfer learning, such as autoregressive next-token prediction (Madani et al., 2023; Ferruz et al., 2022; Hesslow et al., 2022). Different pretraining tasks could potentially learn different aspects of protein biology, and thus have different patterns of scaling. While we only evaluated the MLM pretraining objective, future work that tests other pretraining tasks under our evaluation framework will be critical. However, from principle, we remain uncertain if existing tasks in literature will result in significant differences from MLMs. Many pretraining tasks still aim to reconstruct natural sequences (He et al., 2021; Notin et al., 2022; Tan et al., 2023; Ma et al., 2023) and so are also likely to primarily learn coevolutionary patterns. Other tasks use structure as an additional input or target, but they generally make only modest improvements on function prediction tasks (Mansoor et al., 2021; Wang et al., 2022; Yang et al., 2023; Su et al., 2023). Supporting the idea that predicting structure may not improve function prediction, Hu et al. (2022) show that transfer learning using the AlphaFold2 (Jumper et al., 2021) structure module is less effective for function prediction than transferring PLMs. Finally, Brandes et al. (2022) and Xu et al. (2023) predict both sequence and function but also find that downstream performance does not always scale with pretraining time.

Finally, we only test linear probes or small neural networks built on top of frozen models to limit computational cost, but previous work shows that for many tasks finetuning the PLM end-to-end perform better (Dallago et al., 2021; Yang et al., 2022), and that mean-pooling is rarely optimal (Detlefsen et al., 2022; Goldman et al., 2022). In computer vision, models trained on different datasets (Cherti & Jitsev, 2022) and pretraining tasks (Grigg et al., 2021) exhibit different finetuning dynamics, and there is some evidence for this in proteins as well (Detlefsen et al., 2022). More sophisticated approaches to finetuning (such as using automated ML to select architecture of probe) could further improve performance, and introduce different patterns of alignment.

Besides studying further factors that may impact transfer learning, improvements to our evaluation could better understand mechanisms underlying transfer learning. Transformer PLMs often learn sparse attention matrices (Vig et al., 2020; Rao et al., 2021), so one question is if it is the sparsity that drives performance, or if pairwise attention must be placed on the correct pairs of residues (as opposed to any pairs). However, our current baselines do not permit this understanding: both random init and stat transfer do not guarantee that sparsity in attention matrices is preserved.

Finally, although we require that downstream task performance scale with improvements on the pretraining task and specifically analyze three axes of scaling (model size, depth of model, and checkpoint in pretraining), it is unclear if all three axes are necessary. For example, self-supervised computer vision representations often diverge in their final layers relative to supervised models (Grigg et al., 2021), suggesting that downstream task performance may not always improve monotonically with layer depth, even when pretraining is effective. However, in our context, we chose to include scaling with layer depth because we thought it to be an explanation of potential mechanism for why scaling other axes does not translate into downstream task performance. We observed that even though larger models and models pretrained for longer generally achieve better performance on the pretraining task, this often does not translate into downstream task performance. Our layer depth experiments show that downstream task performance often saturates very early: for the tasks where pretraining improves downstream task performance, but this improvement does not scale, downstream task performance usually saturates within the first 3-5 layers of models (Fig. 4), even in models with 30+ layers. This observation suggests that PLMs are currently burning the majority of their parameters on modeling the pretraining task, with very few of the learned features contributing to both the pretraining and downstream tasks jointly. Ultimately, this means while we offer an oversimplified definition of “alignment”, future work should further analyze the interaction between different axes of scaling.

Implications for future work. Together, the number of factors that can potentially impact transfer learning means that there are many opportunities for future work to address the limitations in scaling that we identified in our work. Towards this goal, our work provides an improved evaluation standard for PLMs. We show that checking for improved performance over baselines may overestimate the generality of PLMs across applications in protein biology, as it does not rule out that improvement may be due to alternate hypotheses that do not scale. However, most current works rely on comparisons to baselines to argue that PLMs are widely applicable, and to the extent scaling has been studied, most only use scaling on structure prediction accuracy alone to justify training larger models (Rives et al., 2021; Elnaggar et al., 2022; Lin et al., 2023; Chen et al., 2024). Future PLM evaluation should therefore assess scaling on diverse downstream function prediction and engineering tasks, and not just structure alone, to validate the generality of models.

Second, synthesizing our empirical results with how the current landscape of protein sequence pretraining tasks primarily align with structure prediction, our work points to a need for new pretraining tasks. For many downstream tasks, the lack of alignment prevents transfer learning from taking full advantage of the pretrained model, as features from deep in the PLM perform no better than features from early layers in the PLM. Likewise, for these tasks, simply scaling to larger PLMs trained for more steps on more data may not improve performance. Our study suggests that the field needs to explore diversified pretraining strategies instead of further scaling existing strategies in order to reach aspects of protein biology that are not currently well-served by PLMs.

Impact Statement

This paper exposes current limitations in protein language models, which are routinely used in protein engineering and bioinformatics. Protein language models scale from year to year, with current models reaching hundreds of billions of parameters (Chen et al., 2024). By showing that current model pretraining paradigms fail to confer benefits on many aspects of protein biology, we caution against uncritically investing compute resources into scaling these models, which we hope will translate to impact through reduced carbon emissions. Additionally, by showing what kinds of tasks protein language models currently fail to scale on, we hope our work leads to the development of pretrained models that improve bioinformatics and protein design predictions beyond those currently well-served by protein language models. If so, we anticipate both positive and negative impacts from an expanded capability to design new proteins.

Acknowledgements and Funding

The authors thank Sarah Alamdari, Frances Arnold, Sébastien Bubeck, Kianoush Falahkheirkhah, Nicolo Fusi, Ivan Jimenez, Kadina Johnston, Alexander Lin, David Alvarez Melis, Ariane Mora, Nitya Thakkar, Amy Wang, Sean Whitzell, Bruce Wittmann, Kevin Wu, and Wenjun Wu for ideas and discussions that helped us improve our work. F.Z.L was partially supported by the National Science Foundation Graduate Research Fellowship and Amazon AI4Science Fellowship at Caltech. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award Number DE-SC0022218. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

Abnar, S., Dehghani, M., Neyshabur, B., and Sedghi, H. Exploring the limits of large scale pre-training. *ICLR*, 2022.

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 2019. doi: 10.1038/s41592-019-0598-1.

Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx431. URL https://doi.org/10.1093/bioinformatics/btx431.

Almagro Armenteros, J. J., Johansen, A. R., Winther, O., and Nielsen, H. Language modelling for biological sequences–curated datasets and baselines. *BioRxiv*, pp. 2020–03, 2020.

Armenteros, J. J. A., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387—-3395, 2017. doi: 10.1093/bioinformatics/btx431.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 02 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac020. URL https://doi.org/10.1093/bioinformatics/btac020.

Bryant, D. H., Bashir, A., Sinai, S., Jain, N. K., Ogden, P. J., Riley, P. F., Church, G. M., Colwell, L. J., and Kelsic, E. D. Deep diversification of an AAV capsid protein by machine learning. *Nature Biotechnology*, 2021. doi: https://doi.org/10.1038/s41587-020-00793-4.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. *CoRR*, abs/1309.0238, 2013. URL http://arxiv.org/abs/1309.0238.

Büning, H., Huber, A., Zhang, L., Meumann, N., and Hacker, U. Engineering the aav capsid to optimize vector–host-interactions. *Current opinion in pharmacology*, 24: 94–104, 2015.

Chen, B., Cheng, X., Li, P., Geng, Y.-a., Gong, J., Li, S., Bei, Z., Tan, X., Wang, B., Zeng, X., et al. xTrimoPGLM: unified 100B-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

Cherti, M. and Jitsev, J. Effect of pre-training scale on intra- and inter-domain, full and few-shot transfer learning for natural and x-ray chest images. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2022.

Chowdhury, R., Bouatta, N., Biswas, S., Floristean, C., Kharkar, A., Roy, K., Rochereau, C., Ahdritz, G., Zhang, J., Church, G. M., et al. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40(11):1617–1623, 2022.

Consortium, U. Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–d515, 2019.

Cuff, J. A. and Barton, G. J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 1999. doi: 10.1002/(SICI)1097-0134(19990301)34:4⟨508::AID-PROT10⟩3.0.CO;2-4.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, 2021. doi: 10.1101/2021.11.09.467890. URL https://www.biorxiv.org/content/early/2021/11/11/2021.11.09.467890.

Detlefsen, N. S., Hauberg, S., and Boomsma, W. Learning meaningful representations of protein sequences. *Nature communications*, 13(1):1914, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding, 2019.

Ding, F. and Steinhardt, J. N. Protein language models are biased by unequal sequence sampling across the tree of life. *bioRxiv*, pp. 2024–03, 2024.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis &; Machine Intelligence*, 44(10):7112–7127, oct 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3095381.

Elnaggar, A., Essam, H., Salah-Eldin, W., Moustafa, W., Elkerdawy, M., Rochereau, C., and Rost, B. Ankh: Optimized protein language model unlocks general-purpose modelling, 2023.

Ferruz, N., Schmidt, S., and Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Flamholz, Z. N., Biller, S. J., and Kelly, L. Large language models improve annotation of viral proteins. *Nature biotechnology*, 2024. doi: 10.1038/s41564-023-01584-8.

Goldman, S., Das, R., Yang, K. K., and Coley, C. W. Machine learning modeling of family wide enzyme-substrate specificity screens. *PLoS computational biology*, 18(2): e1009853, 2022.

Gordon, M. A., Duh, K., and Andrews, N. Compressing bert: Studying the effects of weight pruning on transfer learning. *arXiv preprint arXiv:2002.08307*, 2020.

Grigg, T. G., Busbridge, D., Ramapuram, J., and Webb, R. Do self-supervised and supervised methods learn similar visual representations? *arXiv preprint arXiv:2110.00528*, 2021.

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4918–4927, 2019.

He, L., Zhang, S., Wu, L., Xia, H., Ju, F., Zhang, H., Liu, S., Xia, Y., Zhu, J., Deng, P., et al. Pre-training co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527*, 2021.

Hesslow, D., Zanichelli, N., Notin, P., Poli, I., and Marks, D. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature biotechnology*, 40(7):1114–1122, 2022.

Hu, M., Yuan, F., Yang, K. K., Ju, F., Su, J., Wang, H., Yang, F., and Ding, Q. Exploring evolution-aware & -free protein language models as protein function predictors. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=U8k0QaBgXS.

Jarzab, A., Kurzawa, N., Hopf, T., Moerch, M., Zecha, J., Leijten, N., Bian, Y., Musiol, E., Maschberger, M., Stoehr, G., et al. Meltome atlas—thermal proteome stability across the tree of life. *Nature methods*, 17(5):495–503, 2020.

Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 4037–4058, 2020.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., and Kavukcuoglu, K. Neural machine translation in linear time, 2017.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

Lin, Y., Tan, Y. C., and Frank, R. Open sesame: Getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*, 2019.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574. Earlier versions as preprint: bioRxiv 2022.07.20.500902.

Lutz, S. and Iamurri, S. M. Protein engineering: Past, present, and future. *Methods in Molecular Biology*, 2018. doi: 10.1007/978-1-4939-7366-8_1.

Ma, C., Zhao, H., Zheng, L., Xin, J., Li, Q., Wu, L., Deng, Z., Lu, Y., Liu, Q., and Kong, L. Retrieved sequence augmentation for protein representation learning. *bioRxiv*, pp. 2023–02, 2023.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pp. 1–8, 2023.

Mansoor, S., Baek, M., Madan, U., and Horvitz, E. Toward more general embeddings for protein design: Harnessing joint representations of sequence and structure. *bioRxiv*, 2021.

Matsoukas, C., Haslum, J. F., Sorkhei, M., Söderberg, M., and Smith, K. What makes transfer learning work for medical images: Feature reuse & other factors, 2022.

Maynard Smith, J. Natural selection and the concept of a protein space. *Nature*, 225(5232):563–564, 1970. doi: 0.1038/225563a0.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., and Dauphin, Y. (eds.), *Advances in Neural Information Processing Systems 34*, 2021.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. Critical assessment of methods of protein structure prediction (CASP) – round XII. *Proteins*, 2018. doi: 10.1002/prot.25415.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning?, 2021.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. Progen2: Exploring the boundaries of protein language models, 2022.

Notin, P., Dias, M., Frazer, J., Hurtado, J. M., Gomez, A. N., Marks, D., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022.

Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. Transfusion: Understanding transfer learning for medical imaging, 2019.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating protein transfer learning with TAPE, 2019.

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=fylclEqgvgd.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/doi/full/10.1073/pnas.2016239118. bioRxiv 10.1101/622803.

Rogers, A., Kovaleva, O., and Rumshisky, A. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

Romero, P. A. and Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866–876, 2009. doi: 10.1038/nrm2805.

Stärk, H., Dallago, C., Heinzinger, M., and Rost, B. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 11 2021. ISSN 2635-0041. doi: 10.1093/bioadv/vbab035. URL https://doi.org/10.1093/bioadv/vbab035.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. SaProt: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Tan, Y., Li, M., Tan, P., Zhou, Z., Yu, H., Fan, G., and Hong, L. PETA: Evaluating the impact of protein transfer learning with sub-word tokenization on downstream applications. *arXiv preprint arXiv:2310.17415*, 2023.

Teufel, F., Almagro Armenteros, J. J., Johansen, A. R., Gíslason, M. H., Pihl, S. I., Tsirigos, K. D., Winther, O., Brunak, S., von Heijne, G., and Nielsen, H. SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nature biotechnology*, 40(7):1023–1025, 2022.

Thumuluri, V., Almagro Armenteros, J. J., Johansen, A. R., Nielsen, H., and Winther, O. DeepLoc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic Acids Research*, 2022.

Vandenberghe, L., Wilson, J., and Gao, G. Tailoring the aav vector capsid for gene therapy. *Gene therapy*, 16(3): 311–319, 2009.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R., and Rajani, N. F. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

Wang, Z., Combs, S. A., Brand, R., Rebollar, M. C., Xu, P., Price, G., Golovach, N., Salawu, E. O., Wise, C., Ponnapalli, S. P., and Clark, P. M. LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. *Scientific Reports*, 12, 2022.

Weinstein, E. N., Amin, A. N., Frazer, J., and Marks, D. S. Non-identifiability and the blessings of misspecification in models of molecular fitness. *bioRxiv*, 2023. doi: 10.1101/2022.01.29.478324. URL https://www.biorxiv.org/content/early/2023/02/08/2022.01.29.478324.

Wittmann, B. J., Yue, Y., and Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 2021. doi: 10.1016/j.cels.2021.07.008.

Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O., and Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965, jul 2016. ISSN 2050-084X. doi: 10.7554/eLife.16965. URL https://doi.org/10.7554/eLife.16965.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., Ma, J., and Peng, J. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022. doi: 10.1101/2022.07.21.500999. URL https://www.biorxiv.org/content/early/2022/07/22/2022.07.21.500999.

Xu, M., Yuan, X., Miret, S., and Tang, J. ProtST: multi-modality learning of protein sequences and biomedical texts. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yang, K. K., Fusi, N., and Lu, A. X. Convolutions are competitive with transformers for protein sequence pre-training. *bioRxiv*, pp. 2022–05, 2022.

Yang, K. K., Zanichelli, N., and Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *Protein Engineering, Design and Selection*, 36: gzad015, 2023.

Yang, K. K., Fusi, N., and Lu, A. X. Convolutions are competitive with transformers for protein sequence pre-training. *Cell Systems*, 15(3):286–294.e2, 2024. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2024.01.008. URL https://www.sciencedirect.com/science/article/pii/S2405471224000292.

Yang, Y., Gao, J., Wang, J., Heffernan, R., Hanson, J., Paliwal, K., and Zhou, Y. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, 19(3):482–494, 12 2016. ISSN 1477-4054. doi: 10.1093/bib/bbw129. URL https://doi.org/10.1093/bib/bbw129.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Zhang, Z., Wayment-Steele, H. K., Brixi, G., Wang, H., Peraro, M. D., Kern, D., and Ovchinnikov, S. Protein language models learn evolutionary statistics of interacting sequence motifs. *bioRxiv*, 2024. doi: 10.1101/2024.01.30.577970. URL https://www.biorxiv.org/content/early/2024/01/31/2024.01.30.577970.

A. Additional Tables and Figures

Table A1: Downstream functional and structural tasks

Task	Task type	Split type	n Train sequences	n Val sequences	n Test sequences	Sequence length [min, max]	Model type (n classes)
SS3 – CB513	Residue-level secondary structure	Minimal homology	8678	2170	513	[20, 1632]	PyTorch linear classifier (3)
SS3 – TS115	Residue-level secondary structure	Released in 2016, from 43 to 1085 residues	8678	2170	115	[20, 1632]	PyTorch linear classifier (3)
SS3 – CASP12	Residue-level secondary structure	CASP12 targets, mostly more than 400 residues	8678	2170	21	[20, 1632]	PyTorch linear classifier (3)
Thermostability	Global property	In-distribution	22335	2482	3134	[20, 35213]	Scikit-learn ridge regression
Subcellular localization (scl)	Global property	In-distribution	9503	1678	385	[40, 13100]	PyTorch linear classifier (10)
GB1 – sampled	Local property	In-distribution	6289	699	1745	[265]	Scikit-learn ridge regression
GB1 – low vs high	Local property	Out-of-distribution	4580	509	3644	[265]	Scikit-learn ridge regression
GB1 – two vs rest	Local property	Out-of-distribution (fewer training samples)	381	43	8309	[265]	Scikit-learn ridge regression
AAV – two vs many	Local property	Out-of-distribution	28626	3181	50776	[734, 750]	Scikit-learn ridge regression
AAV – one vs many	Local property	Out-of-distribution (fewer training samples)	1053	117	81413	[734, 750]	Scikit-learn ridge regression

Table A2: Pretrained models

Name	Size	Name in code	Layers	Parameters	Embedding dimension
ESM-43M	Small	esm1_t6_43M_UR50S	6	43M	768
ESM-85M	Medium	esm1_t12_85M_UR50S	12	85M	768
ESM-670M	-	esm1_t34_670M_UR50S	34	670M	1280
ESM-650M	Large	esm1b_t33_650M_UR50S	33	650M	1280
CARP-600k	Tiny	carp_600k	16	600k	128
CARP-38M	Small	carp_38M	16	38M	1024
CARP-76M	Medium	carp_76M	32	76M	1024
CARP-640M	Large	carp_640M	56	640M	1280

Table A3: Last layer transfer learning performance for tasks that are aligned with MLM pretraining. Values are accuracy.

Task	Model	Ablation		
		pretrain	rand	stat
SS3 - CB513	onehot	0.49	-	-
	carp_600k	0.71	0.49±0.01	0.45±0.00
	carp_38M	0.76	0.57±0.00	0.48±0.00
	carp_76M	0.79	0.55±0.00	0.43±0.02
	carp_640M	0.82	0.53±0.00	0.45±0.00
	esm1_t6_43M_UR50S	0.74	0.52±0.00	0.51±0.00
	esm1_t12_85M_UR50S	0.77	0.52±0.00	0.50±0.01
	esm1_t34_670M_UR50S	0.80	0.52±0.00	0.51±0.00
	esm1b_t33_650M_UR50S	0.82	0.45±0.00	0.48±0.01
SS3 - TS115	onehot	0.51	-	-
	carp_600k	0.74	0.50±0.00	0.46±0.00
	carp_38M	0.78	0.59±0.00	0.49±0.01
	carp_76M	0.80	0.57±0.00	0.46±0.01
	carp_640M	0.82	0.56±0.00	0.47±0.01
	esm1_t6_43M_UR50S	0.77	0.57±0.00	0.54±0.00
	esm1_t12_85M_UR50S	0.79	0.57±0.00	0.54±0.01
	esm1_t34_670M_UR50S	0.81	0.57±0.00	0.55±0.00
	esm1b_t33_650M_UR50S	0.82	0.47±0.00	0.50±0.01
SS3 - CASP12	onehot	0.48	-	-
	carp_600k	0.66	0.49±0.01	0.45±0.00
	carp_38M	0.69	0.56±0.01	0.50±0.00
	carp_76M	0.70	0.54±0.01	0.45±0.03
	carp_640M	0.73	0.53±0.01	0.48±0.01
	esm1_t6_43M_UR50S	0.68	0.55±0.01	0.53±0.01
	esm1_t12_85M_UR50S	0.68	0.55±0.01	0.53±0.01
	esm1_t34_670M_UR50S	0.71	0.55±0.00	0.53±0.00
	esm1b_t33_650M_UR50S	0.72	0.50±0.00	0.51±0.01

Table A4: Last layer transfer learning performance for tasks where transfer learning improves performance but the pretrain and downstream tasks are not aligned. Values are Spearman rank correlation. We include linear and attention probes for the pretrained models. The “Yang” column indicates results for the best-performing baseline for the PLM from Yang *et al.* 2024 (Yang et al., 2024).

Task	Model	Ablation				Yang
		linear	attention	rand	stat	
Subcellular localization	onehot	0.37	-	-	-	-
	carp_600k	0.45	0.52 ± 0.01	0.30±0.01	0.29±0.00	-
	carp_38M	0.49	0.52 ± 0.02	0.33±0.01	0.33±0.02	-
	carp_76M	0.54	0.48 ± 0.02	0.34±0.01	0.32±0.02	-
	carp_640M	0.57	0.54 ± 0.04	0.34±0.03	0.32±0.00	-
	esm1_t6_43M_UR50S	0.56	-	0.34±0.01	0.33±0.01	-
	esm1_t12_85M_UR50S	0.57	-	0.35±0.02	0.36±0.01	-
	esm1_t34_670M_UR50S	0.62	-	0.35±0.02	0.37±0.02	-
	esm1b_t33_650M_UR50S	0.61	-	0.30±0.00	0.32±0.02	-
Thermostability	onehot	0.12	-	-	-	-
	carp_600k	0.45	0.51 ± 0.01	0.32±0.00	0.29±0.03	-
	carp_38M	0.51	0.51 ± 0.01	0.37±0.01	0.30±0.01	-
	carp_76M	0.51	0.42 ± 0.03	0.36±0.02	0.29±0.01	-
	carp_640M	0.58	0.49 ± 0.03	0.35±0.01	0.30±0.01	0.54
	esm1_t6_43M_UR50S	0.48	-	0.36±0.00	0.35±0.01	-
	esm1_t12_85M_UR50S	0.49	-	0.36±0.01	0.36±0.01	-
	esm1_t34_670M_UR50S	0.58	-	0.38±0.00	0.37±0.01	-
	esm1b_t33_650M_UR50S	0.58	-	0.32±0.01	0.27±0.01	0.67 ± 0.01
AAV - two vs many	onehot	-0.00	-	-	-	-
	carp_600k	0.36	0.37 ± 0.15	0.34±0.07	0.33±0.06	-
	carp_38M	0.49	0.56 ± 0.07	0.40±0.04	0.49±0.05	-
	carp_76M	0.62	0.55 ± 0.10	0.34±0.08	0.55±0.02	-
	carp_640M	0.68	0.56 ± 0.17	0.42±0.06	0.54±0.02	0.81 ± 0.03
	esm1_t6_43M_UR50S	0.54	-	-0.17±0.01	0.22±0.05	-
	esm1_t12_85M_UR50S	0.64	-	-0.15±0.02	-0.00±0.19	-
	esm1_t34_670M_UR50S	0.46	-	-0.11±0.09	0.08±0.21	-
	esm1b_t33_650M_UR50S	0.65	-	0.04±0.26	0.38±0.03	0.61 ± 0.04
GB1 - low vs high	onehot	0.32	-	-	-	-
	carp_600k	0.24	0.08 ± 0.08	0.24±0.02	0.14±0.01	-
	carp_38M	0.48	0.25 ± 0.04	0.38±0.02	0.25±0.03	-
	carp_76M	0.48	0.15 ± 0.04	0.38±0.02	0.18±0.03	-
	carp_640M	0.48	0.15 ± 0.06	0.38±0.03	0.18±0.02	0.43 ± 0.04
	esm1_t6_43M_UR50S	0.46	-	0.34±0.01	0.35±0.01	-
	esm1_t12_85M_UR50S	0.43	-	0.34±0.01	0.35±0.00	-
	esm1_t34_670M_UR50S	0.51	-	0.34±0.01	0.36±0.01	-
	esm1b_t33_650M_UR50S	0.52	-	0.35±0.00	0.27±0.04	0.53 ± 0.03
GB1 - sampled	onehot	0.79	-	-	-	-
	carp_600k	0.79	0.43 ± 0.08	0.75±0.01	0.67±0.02	-
	carp_38M	0.86	0.78 ± 0.03	0.83±0.00	0.77±0.01	-
	carp_76M	0.85	0.69 ± 0.05	0.83±0.00	0.72±0.01	-
	carp_640M	0.87	0.74 ± 0.03	0.83±0.01	0.69±0.01	-
	esm1_t6_43M_UR50S	0.85	-	0.80±0.00	0.81±0.00	-
	esm1_t12_85M_UR50S	0.86	-	0.79±0.00	0.81±0.00	-
	esm1_t34_670M_UR50S	0.87	-	0.80±0.00	0.82±0.00	-
	esm1b_t33_650M_UR50S	0.88	-	0.79±0.00	0.78±0.02	-

Table A5: Last layer transfer learning performance for tasks where transfer learning does not improve performance. Values are Spearman rank correlation for the GB1 tasks and accuracy for subcellular localization. We include linear and attention probes for the pretrained models. The "Yang" column indicates results for the best-performing baseline for the PLM from Yang *et al.* 2024 (Yang et al., 2024).

Task	Model	Ablation				Yang
		linear	attention	rand	stat	
AAV - one vs many	onehot	0.19	-	-	-	-
	carp_600k	0.52	0.18 ± 0.12	0.44±0.06	0.41±0.07	-
	carp_38M	0.39	0.52 ± 0.11	0.24±0.13	0.32±0.06	-
	carp_76M	0.45	0.40 ± 0.08	0.24±0.07	0.24±0.10	-
	carp_640M	0.43	0.51 ± 0.21	0.21±0.12	0.26±0.09	0.73 ± 0.05
	esm1_t6_43M_UR50S	0.36	-	0.41±0.05	0.40±0.05	-
	esm1_t12_85M_UR50S	0.45	-	0.45±0.04	0.38±0.05	-
	esm1_t34_670M_UR50S	0.36	-	0.39±0.06	0.36±0.10	-
	esm1b_t33_650M_UR50S	0.38	-	0.30±0.09	0.30±0.10	0.18 ± 0.01
GB1 - two vs rest	onehot	0.54	-	-	-	-
	carp_600k	0.56	-0.16 ± 0.18	0.42±0.10	0.23±0.09	-
	carp_38M	0.54	0.24 ± 0.20	0.38±0.05	0.35±0.07	-
	carp_76M	0.53	0.07 ± 0.29	0.41±0.04	0.32±0.02	-
	carp_640M	0.58	0.33 ± 0.07	0.44±0.07	0.22±0.03	0.73 ± 0.03
	esm1_t6_43M_UR50S	0.48	-	0.58±0.04	0.61±0.06	-
	esm1_t12_85M_UR50S	0.40	-	0.57±0.04	0.59±0.03	-
	esm1_t34_670M_UR50S	0.51	-	0.57±0.02	0.55±0.02	-
	esm1b_t33_650M_UR50S	0.54	-	0.55±0.01	0.39±0.02	0.67 ± 0.07

Table A6: Spearman's rank correlation (ρ) between downstream task performance and layer depth

Task	CARP-640M		ESM-650M	
	ρ	p	ρ	p
SS3 - CB513	0.989	5.850×10^{-47}	0.954	2.511×10^{-18}
SS3 - TS115	0.985	6.109×10^{-44}	0.953	4.197×10^{-18}
SS3 - CASP12	0.991	2.136×10^{-49}	0.957	1.063×10^{-18}
Subcellular localization	0.694	2.085×10^{-9}	0.621	8.896×10^{-5}
Thermostability	-0.090	5.042×10^{-1}	-0.432	1.068×10^{-2}
AAV - two vs many	0.809	2.583×10^{-14}	0.014	9.378×10^{-1}
GB1 - low vs high	0.289	2.922×10^{-2}	0.814	4.817×10^{-9}
GB1 - sampled	0.325	1.362×10^{-2}	0.850	1.961×10^{-10}
AAV - one vs many	0.853	3.966×10^{-17}	0.757	2.160×10^{-7}
GB1 - two vs rest	0.436	7.023×10^{-4}	-0.267	1.266×10^{-1}

Table A7: Pretrained CARP checkpoints

Name	Fraction	Loss	Accuracy	Step
carp_600k	1	2.505	0.240	4.889×10^5
carp_600k	0.5	2.512	0.239	2.393×10^5
carp_600k	0.25	2.518	0.237	1.143×10^5
carp_600k	0.125	2.527	0.234	5.204×10^4
carp_38M	1	2.303	0.300	1.027×10^6
carp_38M	0.5	2.319	0.295	5.176×10^5
carp_38M	0.25	2.339	0.289	2.569×10^5
carp_38M	0.125	2.363	0.282	1.296×10^5
carp_76M	1	2.206	0.328	6.545×10^5
carp_76M	0.5	2.225	0.322	3.280×10^5
carp_76M	0.25	2.248	0.315	1.630×10^5
carp_76M	0.125	2.278	0.307	8.318×10^4
carp_640M	1	2.019	0.382	6.220×10^5
carp_640M	0.5	2.054	0.372	3.118×10^5
carp_640M	0.25	2.094	0.360	1.547×10^5
carp_640M	0.125	2.146	0.345	7.881×10^4

Table A8: Spearman's rank correlation (ρ) between downstream task performance and CARP pretrain loss

Task	ρ	p
SS3 - CB513	1.000	0.000
SS3 - TS115	1.000	0.000
SS3 - CASP12	0.949	2.000×10^{-6}
Subcellular localization	0.832	7.980×10^{-4}
Thermostability	0.552	6.251×10^{-2}
AAV - two vs many	0.483	1.121×10^{-1}
GB1 - low vs high	-0.392	2.081×10^{-1}
GB1 - sampled	0.441	1.517×10^{-1}
AAV - one vs many	0.727	7.355×10^{-3}
GB1 - two vs rest	-0.084	7.954×10^{-1}

Table A9: Last layer random init or stat transfer replicates transfer learning performance for tasks that are aligned with MLM pretraining. We initialize models with N random seeds. We consider transfer learning from a PLM to improve performance over these baselines if it has a one-tailed p-value < 0.05 in a one-sample t-test using the sample mean and standard deviation across random init or stat transfer models. Values are accuracy. See A3 for pretrained results with linear or non-linear (attention-based) probes.

Task	Ablation	Model	Mean	Std	N	T-Statistic	P-Value
SS3 – CB513	rand	carp_600k	0.489	0.007	3	-58.874	0.000
		carp_38M	0.571	0.003	3	-102.875	0.000
		carp_76M	0.546	0.003	3	-151.050	0.000
		carp_640M	0.535	0.004	3	-125.517	0.000
		esm1_t6_43M_UR50S	0.519	0.001	3	-484.965	0.000
		esm1_t12_85M_UR50S	0.519	0.001	3	-347.688	0.000
		esm1_t34_670M_UR50S	0.517	0.001	3	-434.876	0.000
		esm1b_t33_650M_UR50S	0.448	0.001	3	-517.805	0.000
	stat	carp_600k	0.450	0.001	3	-398.907	0.000
		carp_38M	0.476	0.003	3	-152.133	0.000
		carp_76M	0.430	0.017	3	-37.539	0.000
		carp_640M	0.451	0.001	3	-433.787	0.000
		esm1_t6_43M_UR50S	0.506	0.002	3	-194.808	0.000
		esm1_t12_85M_UR50S	0.504	0.007	3	-62.703	0.000
		esm1_t34_670M_UR50S	0.507	0.003	3	-173.613	0.000
		esm1b_t33_650M_UR50S	0.481	0.005	3	-107.336	0.000
SS3 – TS115	rand	carp_600k	0.504	0.003	3	-149.041	0.000
		carp_38M	0.592	0.004	3	-76.825	0.000
		carp_76M	0.567	0.003	3	-121.379	0.000
		carp_640M	0.557	0.003	3	-158.932	0.000
		esm1_t6_43M_UR50S	0.567	0.001	3	-312.525	0.000
		esm1_t12_85M_UR50S	0.566	0.001	3	-449.186	0.000
		esm1_t34_670M_UR50S	0.566	0.000	3	-1933.284	0.000
		esm1b_t33_650M_UR50S	0.469	0.004	3	-148.865	0.000
	stat	carp_600k	0.464	0.002	3	-209.013	0.000
		carp_38M	0.494	0.010	3	-50.674	0.000
		carp_76M	0.461	0.012	3	-50.636	0.000
		carp_640M	0.471	0.010	3	-60.832	0.000
		esm1_t6_43M_UR50S	0.544	0.004	3	-106.369	0.000
		esm1_t12_85M_UR50S	0.540	0.011	3	-38.783	0.000
		esm1_t34_670M_UR50S	0.549	0.004	3	-108.114	0.000
		esm1b_t33_650M_UR50S	0.504	0.005	3	-106.479	0.000
SS3 – CASP12	rand	carp_600k	0.486	0.006	3	-47.378	0.000
		carp_38M	0.558	0.005	3	-45.221	0.000
		carp_76M	0.543	0.005	3	-51.011	0.000
		carp_640M	0.526	0.007	3	-47.105	0.000
		esm1_t6_43M_UR50S	0.548	0.005	3	-42.335	0.000
		esm1_t12_85M_UR50S	0.550	0.006	3	-35.166	0.000
		esm1_t34_670M_UR50S	0.548	0.002	3	-175.136	0.000
		esm1b_t33_650M_UR50S	0.498	0.003	3	-151.384	0.000
	stat	carp_600k	0.450	0.003	3	-132.755	0.000
		carp_38M	0.501	0.004	3	-79.816	0.000
		carp_76M	0.449	0.035	3	-12.464	0.003
		carp_640M	0.478	0.009	3	-47.290	0.000
		esm1_t6_43M_UR50S	0.528	0.008	3	-31.438	0.001
		esm1_t12_85M_UR50S	0.531	0.008	3	-30.937	0.001
		esm1_t34_670M_UR50S	0.528	0.001	3	-218.247	0.000
		esm1b_t33_650M_UR50S	0.506	0.005	3	-69.137	0.000

Table A10: Last layer random init or stat transfer replicates transfer learning performance for tasks where transfer learning improves performance but the pretrain and downstream tasks are not aligned. We initialize models with N random seeds. We consider transfer learning from a PLM to improve performance over these baselines if it has a one-tailed p-value < 0.05 in a one-sample t-test using the sample mean and standard deviation across random init or stat transfer models. Values are accuracy. See A4 for pretrained results.

Task	Ablation	Model	Mean	Std	N	T-Statistic	P-Value
Subcellular localization	rand	carp_600k	0.300	0.008	3	-32.694	0.000
		carp_38M	0.332	0.011	3	-25.100	0.001
		carp_76M	0.338	0.004	3	-79.000	0.000
		carp_640M	0.336	0.026	3	-15.851	0.002
		esm1_t6_43M_UR50S	0.344	0.007	3	-56.895	0.000
		esm1_t12_85M_UR50S	0.350	0.017	3	-22.720	0.001
		esm1_t34_670M_UR50S	0.350	0.022	3	-20.558	0.001
		esm1b_t33_650M_UR50S	0.295	0.004	3	-135.311	0.000
	stat	carp_600k	0.288	0.003	3	-107.387	0.000
		carp_38M	0.326	0.019	3	-14.647	0.002
		carp_76M	0.318	0.022	3	-18.028	0.002
		carp_640M	0.321	0.001	3	-292.000	0.000
		esm1_t6_43M_UR50S	0.329	0.012	3	-32.375	0.000
		esm1_t12_85M_UR50S	0.358	0.009	3	-40.113	0.000
		esm1_t34_670M_UR50S	0.365	0.016	3	-27.308	0.001
		esm1b_t33_650M_UR50S	0.318	0.021	3	-23.714	0.001
Thermostability	rand	carp_600k	0.322	0.003	3	-80.860	0.000
		carp_38M	0.373	0.008	3	-29.264	0.001
		carp_76M	0.358	0.018	3	-15.127	0.002
		carp_640M	0.349	0.012	3	-34.318	0.000
		esm1_t6_43M_UR50S	0.361	0.001	3	-198.656	0.000
		esm1_t12_85M_UR50S	0.361	0.007	3	-32.457	0.000
		esm1_t34_670M_UR50S	0.377	0.004	3	-84.831	0.000
		esm1b_t33_650M_UR50S	0.321	0.006	3	-78.881	0.000
	stat	carp_600k	0.286	0.027	3	-10.387	0.005
		carp_38M	0.302	0.013	3	-28.463	0.001
		carp_76M	0.286	0.011	3	-36.804	0.000
		carp_640M	0.296	0.007	3	-65.367	0.000
		esm1_t6_43M_UR50S	0.355	0.008	3	-29.599	0.001
		esm1_t12_85M_UR50S	0.355	0.007	3	-32.119	0.000
		esm1_t34_670M_UR50S	0.374	0.007	3	-50.864	0.000
		esm1b_t33_650M_UR50S	0.267	0.009	3	-61.739	0.000

Continued on next page

Table A10: (continued)

Task	Ablation	Model	Mean	Std	N	T-Statistic	P-Value
AAV – two vs many	rand	carp_600k	0.341	0.073	3	-0.465	0.344
		carp_38M	0.403	0.036	3	-4.411	0.024
		carp_76M	0.345	0.080	3	-5.910	0.014
		carp_640M	0.421	0.058	3	-7.670	0.008
		esm1_t6_43M_UR50S	-0.175	0.007	3	-176.614	0.000
		esm1_t12_85M_UR50S	-0.151	0.015	3	-91.025	0.000
		esm1_t34_670M_UR50S	-0.111	0.092	3	-10.659	0.004
		esm1b_t33_650M_UR50S	0.038	0.261	3	-4.077	0.028
	stat	carp_600k	0.330	0.058	3	-0.912	0.229
		carp_38M	0.489	0.055	3	-0.188	0.434
		carp_76M	0.546	0.018	3	-7.002	0.010
		carp_640M	0.542	0.017	3	-13.588	0.003
		esm1_t6_43M_UR50S	0.220	0.048	3	-11.714	0.004
		esm1_t12_85M_UR50S	-0.004	0.193	3	-5.795	0.014
		esm1_t34_670M_UR50S	0.080	0.207	3	-3.162	0.044
		esm1b_t33_650M_UR50S	0.384	0.033	3	-14.290	0.002
GB1 – low vs high	rand	carp_600k	0.244	0.016	3	0.287	0.600
		carp_38M	0.384	0.024	3	-6.699	0.011
		carp_76M	0.384	0.017	3	-10.299	0.005
		carp_640M	0.382	0.029	3	-5.672	0.015
		esm1_t6_43M_UR50S	0.337	0.006	3	-39.774	0.000
		esm1_t12_85M_UR50S	0.336	0.009	3	-17.681	0.002
		esm1_t34_670M_UR50S	0.341	0.012	3	-25.026	0.001
		esm1b_t33_650M_UR50S	0.345	0.000	3	-1445.546	0.000
	stat	carp_600k	0.135	0.013	3	-14.517	0.002
		carp_38M	0.251	0.034	3	-11.504	0.004
		carp_76M	0.181	0.034	3	-15.653	0.002
		carp_640M	0.179	0.019	3	-27.147	0.001
		esm1_t6_43M_UR50S	0.351	0.005	3	-38.536	0.000
		esm1_t12_85M_UR50S	0.350	0.001	3	-102.890	0.000
		esm1_t34_670M_UR50S	0.359	0.014	3	-18.492	0.001
		esm1b_t33_650M_UR50S	0.274	0.036	3	-12.122	0.003
GB1 – sampled	rand	carp_600k	0.747	0.011	3	-5.859	0.014
		carp_38M	0.829	0.004	3	-14.596	0.002
		carp_76M	0.832	0.004	3	-8.754	0.006
		carp_640M	0.826	0.008	3	-8.807	0.006
		esm1_t6_43M_UR50S	0.796	0.002	3	-47.192	0.000
		esm1_t12_85M_UR50S	0.795	0.002	3	-54.522	0.000
		esm1_t34_670M_UR50S	0.799	0.002	3	-65.882	0.000
		esm1b_t33_650M_UR50S	0.790	0.000	3	-661.876	0.000
	stat	carp_600k	0.673	0.015	3	-12.667	0.003
		carp_38M	0.768	0.013	3	-12.457	0.003
		carp_76M	0.725	0.008	3	-26.511	0.001
		carp_640M	0.686	0.011	3	-29.883	0.001
		esm1_t6_43M_UR50S	0.813	0.004	3	-15.683	0.002
		esm1_t12_85M_UR50S	0.813	0.001	3	-61.208	0.000
		esm1_t34_670M_UR50S	0.820	0.003	3	-25.833	0.001
		esm1b_t33_650M_UR50S	0.779	0.025	3	-7.387	0.009

Table A11: Last layer random init or stat transfer replicates transfer learning performance for tasks where transfer learning does not improve performance. We initialize models with N random seeds. We consider transfer learning from a PLM to improve performance over these baselines if it has a one-tailed p-value < 0.05 in a one-sample t-test using the sample mean and standard deviation across random init or stat transfer models. Values are accuracy. See A5 for pretrained results. Values are Spearman rank correlation for the GB1 tasks and accuracy for subcellular localization.

Task	Ablation	Model	Mean	Std	N	T-Statistic	P-Value
AAV – one vs many	rand	carp_600k	0.437	0.058	3	-2.329	0.073
		carp_38M	0.237	0.127	3	-2.045	0.089
		carp_76M	0.244	0.069	3	-5.098	0.018
		carp_640M	0.208	0.118	3	-3.304	0.040
		esm1_t6_43M_UR50S	0.410	0.049	3	1.675	0.882
		esm1_t12_85M_UR50S	0.449	0.039	3	-0.044	0.484
		esm1_t34_670M_UR50S	0.390	0.062	3	0.777	0.741
		esm1b_t33_650M_UR50S	0.302	0.086	3	-1.510	0.135
	stat	carp_600k	0.405	0.070	3	-2.738	0.056
		carp_38M	0.318	0.063	3	-1.918	0.098
		carp_76M	0.236	0.104	3	-3.507	0.036
		carp_640M	0.262	0.092	3	-3.236	0.042
		esm1_t6_43M_UR50S	0.402	0.047	3	1.461	0.859
		esm1_t12_85M_UR50S	0.378	0.054	3	-2.295	0.074
		esm1_t34_670M_UR50S	0.360	0.100	3	-0.041	0.486
		esm1b_t33_650M_UR50S	0.300	0.101	3	-1.325	0.158
GB1 – two vs rest	rand	carp_600k	0.423	0.100	3	-2.418	0.068
		carp_38M	0.379	0.046	3	-6.020	0.013
		carp_76M	0.415	0.043	3	-4.795	0.020
		carp_640M	0.442	0.069	3	-3.481	0.037
		esm1_t6_43M_UR50S	0.580	0.040	3	4.354	0.976
		esm1_t12_85M_UR50S	0.575	0.036	3	8.660	0.993
		esm1_t34_670M_UR50S	0.568	0.018	3	6.072	0.987
		esm1b_t33_650M_UR50S	0.549	0.012	3	0.840	0.755
	stat	carp_600k	0.235	0.094	3	-6.040	0.013
		carp_38M	0.355	0.074	3	-4.354	0.024
		carp_76M	0.317	0.024	3	-15.801	0.002
		carp_640M	0.225	0.029	3	-21.448	0.001
		esm1_t6_43M_UR50S	0.609	0.058	3	3.827	0.969
		esm1_t12_85M_UR50S	0.589	0.030	3	11.153	0.996
		esm1_t34_670M_UR50S	0.545	0.019	3	3.547	0.964
		esm1b_t33_650M_UR50S	0.392	0.022	3	-12.015	0.003

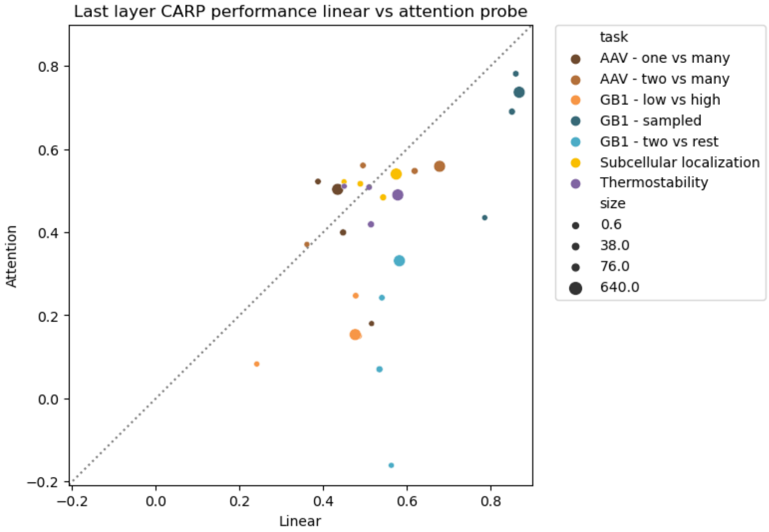

Figure A1: Comparison of linear global average probe versus a learned attention pooling followed by a shallow multi-layer perceptron performance across downstream tasks. Non-linear probe last layer performances are inferior to the linear models across almost all downstream tasks.

B. Supplementary Details

B.1. Architectures

ESM. The ESM family are BERT-style Transformer models (Vaswani et al., 2023; Devlin et al., 2019). Rives et al. (2021) process input protein sequences as strings of amino acids. The ESM model takes a sequence of tokens passing through a series of Transformer encoder blocks. For each block, there is a scaled dot-product multi-head self-attention layer computing position-position interactions across the sequence followed by a feed-forward network independent of position, with residual connections and final layer normalization.

CARP. The CARP family integrates ByteNet, a dilated CNN architecture (Kalchbrenner et al., 2017), with input embedding and output decoding layers (Yang et al., 2024). The CARP embedding process begins with a down-embedding layer, followed by an up-embedding linear mapping. This prepared input then passes through a series of n ByteNet blocks, with residual connections and final layer normalization.

B.2. Datasets

Information about the number of samples in training/validation/test datasets, and strategies used to split test data can be found in Supplementary Table A1.

Secondary Structure. The secondary structure datasets (originally proposed by Rao et al. (2019)) predict if individual residues in proteins belong to secondary structural elements (n α-helix, β-strand, or coil). These secondary structure elements are local aspects of structure, that serve as the building blocks for the overall structure of a protein.

Thermostability. The thermostability dataset measures the melting temperature of 48,000 proteins across 13 species, and was originally published by Jarzab et al. (2020). Thermostable proteins (e.g. proteins that have a higher melting temperature) are capable of maintaining their activity even at high temperatures) can be important to engineer for applications that must occur at high temperatures (e.g. industrial processes, PCR, etc.)

Subcellular Localization. The subcellular localization dataset (originally introduced by Almagro Armenteros et al. (2017)) presents a classification task, predicting what cell compartment an eukaryotic protein localizes. This dataset is a multi-way classification problem between the nucleus, cytoplasm, extracellular, mitochondria, cell membrane, endoplasmic reticulum, plastid, Golgi apparatus, lysosome/vacuole, and peroxisome. Predicting where a protein localizes in a cell can be important for understanding its biological function: for example, proteins in the nucleus will often have a role in DNA or RNA regulation.

GB1. The GB1 dataset, originally introduced in the FLIP benchmark dataset (Dallago et al., 2021), measures the effects of combinatorial mutations at four different sites in the GB1 domain of Protein G, an immunoglobulin-binding protein. Here, the goal is to optimize a sequence that is stable (produces a high fraction of folded proteins) and improves the binding affinity to immunoglobulin. One of the aims of the dataset is to understand epistasis, or non-additive interactions between mutations at different amino acid positions, and this phenomena also makes building machine learning predictors difficult, because models must learn combinatorial interactions between mutations (instead of treating mutation effects as independent).

AAV. The AAV dataset measures the effects of mutations on the adeno-associated virus (AAV) capsid protein. AAV capsid proteins are responsible for helping the virus integrate a DNA payload into a target cell (Vandenberghe et al., 2009), so engineering variants of these proteins can lead to useful products for gene therapy (Büning et al., 2015). The AAV dataset was originally introduced in the FLIP benchmark dataset (Dallago et al., 2021), and measures the fitness (or ability to perform its function) of mutant variants.

B.3. Methods

Additional experimental details. For the regression linear probes implemented in scikit-learn, we performed a grid search over alpha values (controlling regulation strength) of $[1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01]$. Output predicted fitness values are scaled with StandardScaler().

For the classification linear probes implemented in PyTorch, we use Adam optimizer and set learning rate to $1.e-4$ with a decay rate of 0.1. Batch size of 256 over 120 epoches was used for annotation tasks and a batch size of 120 over 100 epoches for the secondary structure tasks. We implement early stopping on validation loss with a tolerance of 10 (after a minimum of 5 epochs).

Non-linear probe. For the non-linear probe, a shallow neural net with learned aggregation is applied with 5 random seeds on 3-5 checkpoints.

B.4. Definitions

- **Alignment:** We define the MLM pretraining task to be aligned with a downstream task if transferring PLM representations improves downstream task performance over the baseline and ablations and this improvement scales with improvements to pretraining.