Evaluating Protein Transfer Learning with TAPE

Roshan Rao*[1] Nicholas Bhattacharya*[1] Neil Thomas*[1]
Yan Duan[2] Xi Chen[2] John Canny[1,3] Pieter Abbeel[1,2] Yun S. Song[1,4]

Abstract

Protein modeling is an increasingly popular area of machine learning research. Semi-supervised learning has emerged as an important paradigm in protein modeling due to the high cost of acquiring supervised protein labels, but the current literature is fragmented when it comes to datasets and standardized evaluation techniques. To facilitate progress in this field, we introduce the Tasks Assessing Protein Embeddings (TAPE), a set of five biologically relevant semi-supervised learning tasks spread across different domains of protein biology. We curate tasks into specific training, validation, and test splits to ensure that each task tests biologically relevant generalization that transfers to real-life scenarios. We benchmark a range of approaches to semi-supervised protein representation learning, which span recent work as well as canonical sequence learning techniques. We find that self-supervised pretraining is helpful for almost all models on all tasks, more than doubling performance in some cases. Despite this increase, in several cases features learned by self-supervised pretraining still lag behind features extracted by state-of-the-art non-neural techniques. This gap in performance suggests a huge opportunity for innovative architecture design and improved modeling paradigms that better capture the signal in biological sequences. TAPE will help the machine learning community focus effort on scientifically relevant problems. Toward this end, all data and code used to run these experiments are available at `https://github.com/songlab-cal/tape`.

1 Introduction

New sequencing technologies have led to an explosion in the size of protein databases over the past decades. These databases have seen exponential growth, with the total number of sequences doubling every two years [1]. Obtaining meaningful labels and annotations for these sequences requires significant investment of experimental resources, as well as scientific expertise, resulting in an exponentially growing gap between the size of protein sequence datasets and the size of annotated subsets. Billions of years of evolution have sampled the portions of protein sequence space that are relevant to life, so large unlabeled datasets of protein sequences are expected to contain significant biological information [2–4]. Advances in natural language processing (NLP) have shown that self-supervised learning is a powerful tool for extracting information from unlabeled sequences [5–7], which raises a tantalizing question: can we adapt NLP-based techniques to extract useful biological information from massive sequence datasets?

To help answer this question, we introduce the Tasks Assessing Protein Embeddings (TAPE), which to our knowledge is the first attempt at systematically evaluating semi-supervised learning on protein sequences. TAPE includes a set of five biologically relevant supervised tasks that evaluate the performance of learned protein embeddings across diverse aspects of protein understanding.

We choose our tasks to highlight three major areas of protein biology where self-supervision can facilitate scientific advances: structure prediction, detection of remote homologs, and protein en-

*Equal Contribution [1]UC Berkeley [2]covariant.ai [3]Google [4]Chan Zuckerberg Biohub
Correspondence to: {roshan_rao,nick_bhat,nthomas,yss}@berkeley.edu

gineering. We constructed data splits to simulate biologically relevant generalization, such as a model's ability to generalize to entirely unseen portions of sequence space, or to finely resolve small portions of sequence space. Improvement on these tasks range in application, including designing new antibodies [8], improving cancer diagnosis [9], and finding new antimicrobial genes hiding in the so-called "Dark Proteome": tens of millions of sequences with no labels where existing techniques for determining protein similarity fail [10].

We assess the performance of three representative models (recurrent, convolutional, and attention-based) that have performed well for sequence modeling in other fields to determine their potential for protein learning. We also compare two recently proposed semi-supervised models (Bepler et al. [11], Alley et al. [12]). With our benchmarking framework, these models can be compared directly to one another for the first time.

We show that self-supervised pretraining improves performance for almost all models on all downstream tasks. Interestingly, performance for each architecture varies significantly across tasks, highlighting the need for a multi-task benchmark such as ours. We also show that non-deep alignment-based features [13–16] outperform features learned via self-supervision on secondary structure and contact prediction, while learned features perform significantly better on remote homology detection.

Our results demonstrate that self-supervision is a promising paradigm for protein modeling but considerable improvements need to be made before self-supervised models can achieve breakthrough performance. All code and data for TAPE are publically available[1], and we encourage members of the machine learning community to participate in these exciting problems.

2 Background

2.1 Protein Terminology

Proteins are linear chains of amino acids connected by covalent bonds. We encode amino acids in the standard 25-character alphabet, with 20 characters for the standard amino acids, 2 for the non-standard amino acids selenocysteine and pyrrolysine, 2 for ambiguous amino acids, and 1 for when the amino acid is unknown [1, 17]. Throughout this paper, we represent a protein x of length L as a sequence of discrete amino acid characters $(x_1, x_2, \ldots, x_L)$ in this fixed alphabet.

Beyond its encoding as a sequence $(x_1, \ldots, x_L)$, a protein has a 3D molecular structure. The different levels of protein structure include *primary* (amino acid sequence), *secondary* (local features), and *tertiary* (global features). Understanding how primary sequence folds into tertiary structure is a fundamental goal of biochemistry [2]. Proteins are often made up of a few large *protein domains*, sequences that are evolutionarily conserved, and as such have a well-defined fold and function.

Evolutionary relationships between proteins arise because organisms must maintain certain functions, such as replicating DNA, as they evolve. Evolution has selected for proteins that are well-suited to these functions. Though structure is constrained by evolutionary pressures, sequence-level variation can be high, with very different sequences having similar structure [18]. Two proteins with different sequences but evolutionary or functional relationships are called *homologs*.

Quantifying these evolutionary relationships is very important for preventing undesired information leakage between data splits. We mainly rely on *sequence identity*, which measures the percentage of exact amino acid matches between aligned subsequences of proteins [19]. For example, filtering at a 25% sequence identity threshold means that no two proteins in the training and test set have greater than 25% exact amino acid matches. Other approaches besides sequence identity filtering also exist, depending on the generalization the task attempts to test [20].

2.2 Modeling Evolutionary Relationships with Sequence Alignments

The key technique for modeling sequence relationships in computational biology is alignment [13, 16, 21, 22]. Given a database of proteins and a new protein at test-time, an alignment-based method uses either carefully designed scoring systems to perform pairwise comparisons [21, 23], Hidden Markov Model-like probabilistic models [16], or a combination [13] to align the test protein

[1] `https://github.com/songlab-cal/tape`

against the database. If good alignments are found, information from the alignments is either directly sufficient for the task at hand, or can be fed into downstream models for further use [24].

2.3 Semi-supervised Learning

The fields of computer vision and natural language processing have been dealing with the question of how to learn from unlabeled data for years [25]. Images and text found on the internet generally lack accompanying annotations, yet still contain significant structure. Semi-supervised learning tries to jointly leverage information in the unlabeled and labeled data, with the goal of maximizing performance on the supervised task. One successful approach to learning from unlabeled examples is *self-supervised learning*, which in NLP has taken the form of next-token prediction [5], masked-token prediction [6], and next-sentence classification [6]. Analogously, there is good reason to believe that unlabelled protein sequences contain significant information about their structure and function [2, 4]. Since proteins can be modeled as sequences of discrete tokens, we test both next-token and masked-token prediction for self-supervised learning.

3 Related Work

The most well-known protein modeling benchmark is the Critical Assessment of Structure Prediction (CASP) [26], which focuses on structure modeling. Each time CASP is held, the test set consists of new experimentally validated structures which are held under embargo until the competition ends. This prevents information leakage and overfitting to the test set. The recently released ProteinNet [27] provides easy to use, curated train/validation/test splits for machine learning researchers where test sets are taken from the CASP competition and sequence identity filtering is already performed. We take the contact prediction task from ProteinNet. However, we believe that structure prediction alone is not a sufficient benchmark for protein models, so we also use tasks not included in the CASP competition to give our benchmark a broader focus.

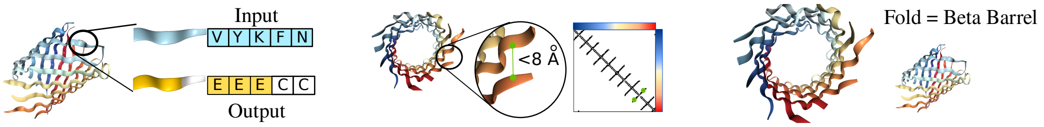
Semi-supervised learning for protein problems has been explored for decades, with lots of work on kernel-based pretraining [28, 29]. These methods demonstrated that semi-supervised learning improved performance on protein network prediction and homolog detection, but couldn't scale beyond hundreds of thousands of unlabeled examples. Recent work in protein representation learning has proposed a variety of methods that apply NLP-based techniques for transfer learning to biological sequences [11, 12, 30, 31]. In a related line of work, Riesselman et al. [32] trained Variational Auto Encoders on aligned families of proteins to predict the functional impact of mutations. Alley et al. [12] also try to combine self-supervision with alignment in their work by using alignment-based querying to build task-specific pretraining sets.

Due to the relative infancy of protein representation learning as a field, the methods described above share few, if any, benchmarks. For example, both Rives et al. [31] and Bepler et al. [11] report transfer learning results on secondary structure prediction and contact prediction, but they differ significantly in test set creation and data-splitting strategies. Other self-supervised work such as Alley et al. [12] and Yang et al. [33] report protein engineering results, but on different tasks and datasets. With such varied task evaluation, it is challenging to assess the relative merits of different self-supervised modeling approaches, hindering efficient progress.

4 Datasets

Here we describe our unsupervised pretraining and supervised benchmark datasets. To create benchmarks that test generalization across large evolutionary distances and are useful in real-life scenarios, we curate specific training, validation, and test splits for each dataset. Producing the data for these tasks requires significant effort by experimentalists, database managers, and others. Following similar benchmarking efforts in NLP [34], we describe a set of citation guidelines in our repository[2] to ensure these efforts are properly acknowledged.

[2] `https://github.com/songlab-cal/tape#citation-guidelines`



(a) Secondary Structure (b) Contact Prediction (c) Remote Homology

Figure 1: Structure and Annotation Tasks on protein KgdM Porin (pdbid: 4FQE). (a) Viewing this Porin from the side, we show secondary structure, with the input amino acids for a segment (blue) and corresponding secondary structure labels (yellow and white). (b) Viewing this Porin from the front, we show a contact map, where entry i, j in the matrix indicates whether amino acids at positions i, j in the sequence are within 8 angstroms of each other. In green is a contact between two non-consecutive amino acids. (c) The fold-level remote homology class for this protein.

4.1 Unlabeled Sequence Dataset

We use Pfam [35], a database of thirty-one million protein domains used extensively in bioinformatics, as the pretraining corpus for TAPE. Sequences in Pfam are clustered into evolutionarily-related groups called *families*. We leverage this structure by constructing a test set of fully heldout families (see Section A.4 for details on the selected families), about 1% of the data. For the remaining data we construct training and test sets using a random 95/5% split. Perplexity on the uniform random split test set measures in-distribution generalization, while perplexity on the heldout families test set measures out-of-distribution generalization to proteins that are less evolutionarily related to the training set.

4.2 Supervised Datasets

We provide five biologically relevant downstream prediction tasks to serve as benchmarks. We categorize these into structure prediction, evolutionary understanding, and protein engineering tasks. The datasets vary in size between 8 thousand and 50 thousand training examples (see Table S1 for sizes of all training, validation and test sets). Further information on data processing, splits and experimental challenges is in Appendix A.1. For each task we provide:

(Definition) A formal definition of the prediction problem, as well as the source of the data.
(Impact) The impact of improving performance on this problem.
(Generalization) The type of understanding and generalization desired.
(Metric) The metric used in Table 2 to report results.

Task 1: Secondary Structure (SS) Prediction (Structure Prediction Task)

(Definition) Secondary structure prediction is a sequence-to-sequence task where each input amino acid x_i is mapped to a label $y_i \in \{\text{Helix, Strand, Other}\}$. See Figure 1a for illustration. The data are from Klausen et al. [36].

(Impact) SS is an important feature for understanding the function of a protein, especially if the protein of interest is not evolutionarily related to proteins with known structure [36]. SS prediction tools are very commonly used to create richer input features for higher-level models [37].

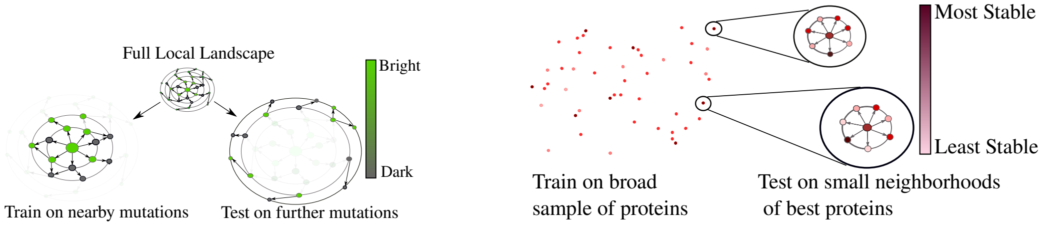
(Generalization) SS prediction tests the degree to which models learn local structure. Data splits are filtered at 25% sequence identity to test for broad generalization.

(Metric) We report accuracy on a per-amino acid basis on the CB513 [38] dataset.

Task 2: Contact Prediction (Structure Prediction Task)

(Definition) Contact prediction is a pairwise amino acid task, where each pair x_i, x_j of input amino acids from sequence x is mapped to a label $y_{ij} \in \{0, 1\}$, where the label denotes whether the amino acids are "in contact" ($<$ 8Å apart) or not. See Figure 1b for illustration. The data are from the ProteinNet dataset [27].

(Impact) Accurate contact maps provide powerful global information; e.g., they facilitate robust modeling of full 3D protein structure [39]. Of particular interest are medium- and long-range contacts, which may be as few as twelve sequence positions apart, or as many as hundreds apart.



(a) Fluorescence

(b) Stability

Figure 2: Protein Engineering Tasks. In both tasks, a parent protein p is mutated to explore the local landscape. As such, dots represent proteins and directed arrow $x \to y$ denotes that y has exactly one more mutation than x away from parent p. (a) The Fluorescence task consists of training on small neighborhood of the parent green fluorescent protein (GFP) and then testing on a more distant proteins. (b) The Stability task consists of training on a broad sample of proteins, followed by testing on one-mutation neighborhoods of the most promising sampled proteins.

(Generalization) The abundance of medium- and long-range contacts makes contact prediction an ideal task for measuring a model's understanding of global protein context. We select the data splits that was filtered at 30% sequence identity to test for broad generalization.

(Metric) We report precision of the $L/5$ most likely contacts for medium- and long-range contacts on the ProteinNet CASP12 test set, which is a standard metric reported in CASP [26].

Task 3: Remote Homology Detection (Evolutionary Understanding Task)

(Definition) This is a sequence classification task where each input protein x is mapped to a label $y \in \{1, \ldots, 1195\}$, representing different possible protein folds. See Figure 1c for illustration. The data are from Hou et al. [40].

(Impact) Detection of remote homologs is of great interest in microbiology and medicine; e.g., for detection of emerging antibiotic resistant genes [41] and discovery of new CAS enzymes [42].

(Generalization) Remote homology detection measures a model's ability to detect structural similarity across distantly related inputs. We hold out entire evolutionary groups from the training set, forcing models to generalize across large evolutionary gaps.

(Metric) We report overall classification accuracy on the fold-level heldout set from Hou et al. [40].

Task 4: Fluorescence Landscape Prediction (Protein Engineering Task)

(Definition) This is a regression task where each input protein x is mapped to a label $y \in \mathbb{R}$, corresponding to the log-fluorescence intensity of x. See Figure 2a for illustration. The data are from Sarkisyan et al. [43].

(Impact) For a protein of length L, the number of possible sequences m mutations away is $O(L^m)$, a prohibitively large space for exhaustive search via experiment, even if m is modest. Moreover, due to epistasis (second- and higher-order interactions between mutations at different positions), greedy optimization approaches are unlikely to succeed. Accurate computational predictions could allow significantly more efficient exploration of the landscape, resulting in better optima. Machine learning methods have already seen some success in related protein engineering tasks [44].

(Generalization) The fluorescence prediction task tests the model's ability to distinguish between very similar inputs, as well as its ability to generalize to unseen combinations of mutations. The train set is a Hamming distance-3 neighborhood of the parent green fluorescent protein (GFP), while the test set has variants with four or more mutations.

(Metric) We report Spearman's ρ (rank correlation coefficient) on the test set.

Task 5: Stability Landscape Prediction (Protein Engineering Task)

(Definition) This is a regression task where each input protein x is mapped to a label $y \in \mathbb{R}$ measuring the most extreme circumstances in which protein x maintains its fold above a concentration threshold (a proxy for intrinsic stability). See Figure 2b for illustration. The data are from Rocklin et al. [45].

Table 1: Language modeling metrics

	Random Families			Heldout Families		
	Accuracy	Perplexity	ECE	Accuracy	Perplexity	ECE
Transformer	**0.45**	**8.89**	**6.01**	**0.30**	**13.04**	**10.04**
LSTM	0.40	**8.89**	6.94	0.16	14.72	15.21
ResNet	0.41	10.16	6.86	0.29	13.55	10.32
Supervised LSTM [11]	0.28	11.62	10.17	0.14	15.28	16.02
UniRep mLSTM [12]	0.32	11.29	9.08	0.12	16.36	16.92
Random	0.04	25	25	0.04	25	25

(Impact) Designing stable proteins is important to ensure, for example, that drugs are delivered before they are degraded. More generally, given a broad sample of protein measurements, finding better refinements of top candidates is useful for maximizing yield from expensive protein engineering experiments.

(Generalization) This task tests a model's ability to generalize from a broad sampling of relevant sequences and to localize this information in a neighborhood of a few sequences, inverting the test-case for fluorescence above. The train set consists of proteins from four rounds of experimental design, while the test set contains Hamming distance-1 neighbors of top candidate proteins.

(Metric) We report Spearman's ρ on the test set.

5 Models and Experimental Setup

Losses: We examine two self-supervised losses that have seen success in NLP. The first is *next-token prediction* [46], which models $p(x_i \mid x_1, \ldots, x_{i-1})$. Since many protein tasks are sequence-to-sequence and require bidirectional context, we apply a variant of next-token prediction which additionally trains the reverse model, $p(x_i \mid x_{i+1}, \ldots, x_L)$, providing full context at each position (assuming a Markov sequence). The second is *masked-token prediction* [6], which models $p(x_{\text{masked}} \mid x_{\text{unmasked}})$ by replacing the value of tokens at multiple positions with alternate tokens.

Protein-specific loss: In addition to self-supervised algorithms, we explore another protein-specific training procedure proposed by Bepler et al. [11]. They suggest that further *supervised* pretraining of models can provide significant benefits. In particular, they propose supervised pretraining on contact prediction and remote homology detection, and show it increases performance on secondary structure prediction. Similar work in computer vision has shown that supervised pretraining can transfer well to other tasks, making this a promising avenue of exploration [47].

Architectures and Training: We implement three architectures: an LSTM [48], a Transformer [49], and a dilated residual network (ResNet) [50]. We use a 12-layer Transformer with a hidden size of 512 units and 8 attention heads, leading to a 38M-parameter model. Hyperparameters for the other models were chosen to approximately match the number of parameters in the Transformer. Our LSTM consists of two three-layer LSTMs with 1024 hidden units corresponding to the forward and backward language models, whose outputs are concatenated in the final layer, similar to ELMo [5]. For the ResNet we use 35 residual blocks, each containing two convolutional layers with 256 filters, kernel size 9, and dilation rate 2.

In addition, we benchmark two previously proposed architectures that differ significantly from the three above. The first, proposed by Bepler et al. [11], is a two-layer bidirectional language model, similar to the LSTM discussed above, followed by three 512 hidden unit bidirectional LSTMs. The second, proposed by Alley et al. [12], is a unidirectional mLSTM [51] with 1900 hidden units. Details on implementing and training these architectures can be found in the original papers.

The Transformer and ResNet are trained with masked-token prediction, while the LSTM is trained with next-token prediction. Both Alley et al. and Bepler et al. are trained with next-token prediction. All self-supervised models are trained on four NVIDIA V100 GPUs for one week.

Table 2: Results on downstream supervised tasks

Method		Structure		Evolutionary	Engineering	
		SS	Contact	Homology	Fluorescence	Stability
No Pretrain	Transformer	0.70	0.32	0.09	0.22	-0.06
	LSTM	0.71	0.19	0.12	0.21	0.28
	ResNet	0.70	0.20	0.10	-0.28	0.61
Pretrain	Transformer	0.73	0.36	0.21	**0.68**	**0.73**
	LSTM	0.75	0.39	**0.26**	0.67	0.69
	ResNet	0.75	0.29	0.17	0.21	**0.73**
Supervised [11]	LSTM	0.73	0.40	0.17	0.33	0.64
UniRep [12]	mLSTM	0.73	0.34	0.23	0.67	**0.73**
Baseline	One-hot	0.69	0.29	0.09	0.14	0.19
	Alignment	**0.80**	**0.64**	0.09	N/A	N/A

Baselines: We evaluate learned features against two baseline featurizations. The first is a one-hot encoding of the input amino acid sequence, which provides a simple baseline. Additionally, most current state-of-the-art algorithms for protein modeling take advantage of alignment or HMM-based inputs (see Section 2.2). Alignments can be transformed into various features, such as mutation probabilities [40] or the HMM state-transition probabilities [36] for each amino acid position. These are concatenated to the one-hot encoding of the amino acid to form another baseline featurization. For our baselines we use alignment-based inputs that vary per task depending on the inputs used by the current state-of-the-art method. See Appendix A.2 for details on the alignment-based features used. We do not use alignment-based inputs for protein engineering tasks - since proteins in the engineering datasets differ by only a single amino acid, and alignment-based methods search for proteins with high sequence identity, the alignment-based methods return the same set of features for all proteins we wish to distinguish between.

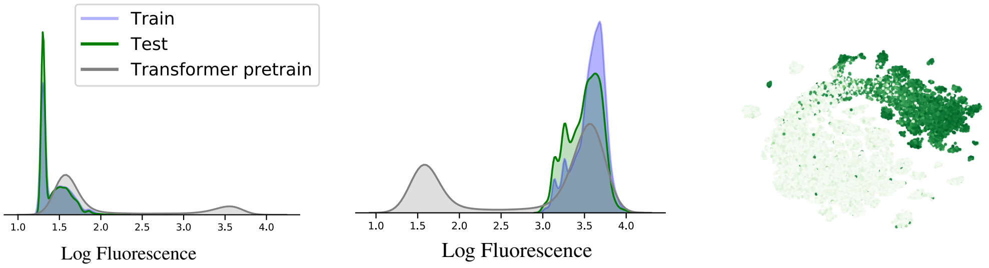
Experimental Setup: The goal of our experimental setup is to systematically compare all featurizations. For each task we select a particular supervised architecture, drawing from state-of-the-art where available, and make sure that fine-tuning on all language models is identical. See Appendix A.2 for details on supervised architectures and training.

6 Results

Table 1 contains accuracy, perplexity, and exponentiated cross entropy (ECE) on the language modeling task for the five architectures we trained with self-supervision as well as a random model baseline. We report metrics on both the random split and the fully heldout families. Supervised LSTM metrics are reported after language modeling pretraining, but before supervised pretraining. Heldout family accuracy is consistently lower than random-split accuracy, demonstrating a drop in the out-of-distribution generalization ability. Note that although some models have lower perplexity than others on both random-split and heldout sets, this lower perplexity does not necessarily correspond to better performance on downstream tasks. This replicates the finding in Rives et al. [31].

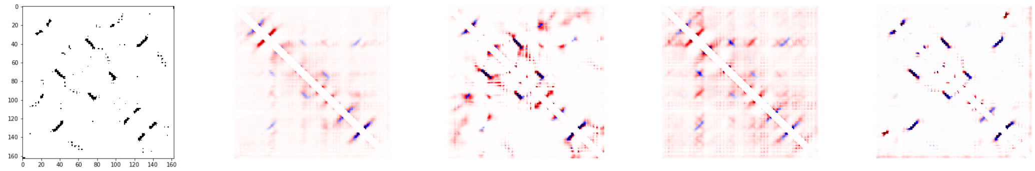
Table 2 contains results for all benchmarked architectures and training procedures on all downstream tasks in TAPE. We report accuracy, precision, or Spearman's ρ, depending on the task, so higher is always better and each metric has a maximum value of 1.0. See Section 4 for the metric reported in each task. Detailed results and metrics for each task are in Appendix A.6.

We see from Table 2 that self-supervised pretraining improves overall performance across almost all models and all tasks. Further analysis reveals aspects of these tasks with more for significant improvement. In the fluorescence task, the distribution is bimodal with a mode of bright proteins and a mode of dark proteins (see Figure 3). Since one goal of using machine learning models in protein engineering is to screen potential variants, it is important for these methods to successfully distinguish between beneficial and deleterious mutations. Figure 3 shows that the model does successfully perform some clustering of fluorescent proteins, but that many proteins are still misclassified.



(a) Dark Mode (b) Bright Mode (c) Embedding t-SNE

Figure 3: Distribution of training, test, and pretrained Transformer predictions on the dark and bright modes, along with t-SNE of pretrained Transformer protein embeddings colored by log-fluorescence.



(a) True Contacts (b) LSTM (c) LSTM Pretrain (d) One Hot (e) Alignment

Figure 4: Predicted contacts for chain 1A of a Bacterioferritin comigratory protein (pdbid: 3GKN). Blue indicates true positive contacts while red indicates false positive contacts. Darker colors represent more certainty from the model.

For the stability task, to identify which mutations a model believes are beneficial, we use the parent protein as a decision boundary and label a mutation as beneficial if its predicted stability is higher than the parent's predicted stability. We find that our best pretrained model achieves 70% accuracy in making this prediction while our best non-pretrained model achieves 68% accuracy (see Table S8 for full results). Improving the ability to distinguish beneficial from deleterious mutations would make these models much more useful in real protein engineering experiments.

In the contact prediction task, long-range contacts are of particular interest and can be hundreds of positions apart. Figure 4 shows the predictions of several models on a protein where the longest range contact occurs between the 8th and 136th amino acids. Pretraining helps the model capture more long-range information and improves the overall resolution of the predicted map. However, the hand-engineered alignment features result in a much sharper map, accurately resolving a smaller number well-spaced of long-range contacts. This increased specificity is highly relevant in the structure prediction pipeline [39, 52] and highlights a clear challenge for pretraining.

7 Discussion

Comparison to state of the art. As shown in Table 2, alignment-based inputs can provide a powerful signal that outperforms current self-supervised models on multiple tasks. Current state-of-the-art prediction methods for secondary structure prediction, contact prediction, and remote homology classification all take in alignment-based inputs. These methods combine alignment-based inputs with other techniques (e.g. multi-task training, kernel regularization) to achieve an additional boost in performance. For comparison, NetSurfP-2.0 [36] achieves 85% accuracy on the CB513 [38] secondary structure dataset, compared to our best model's 75% accuracy, RaptorX [53] achieves 0.69 precision at $L/5$ on CASP12 contact prediction, compared to our best model's 0.49, and DeepSF [40] achieves 41% accuracy on remote homology detection compared to our best model's 26%.

Need for multiple benchmark tasks. Our results support our hypothesis that multiple tasks are required to appropriately benchmark performance of a given method. Our Transformer, which performs worst of the three models in secondary structure prediction, performs best on the fluorescence and stability tasks. The reverse is true of our ResNet, which ties the LSTM in secondary structure prediction but performs far worse for the fluorescence task, with a Spearman's ρ of 0.21 compared to the LSTM's 0.67. This shows that performance on a single task does not capture the full extent of a trained model's knowledge and biases, creating the need for multi-task benchmarks such as TAPE.

8 Future Work

Protein representation learning is an exciting field with incredible room for expansion, innovation, and impact. The exponentially growing gap between labeled and unlabeled protein data means that self-supervised learning will continue to play a large role in the future of computational protein modeling. Our results show that no single self-supervised model performs best across all protein tasks. We believe this is a clear challenge for further research in self-supervised learning, as there is a huge space of model architecture, training procedures, and unsupervised task choices left to explore. It may be that language modeling as a task is not enough, and that protein-specific tasks are necessary to push performance past state of the art. Further exploring the relationship between alignment-based and learned representations will be necessary to capitalize on the advantages of each. We hope that the datasets and benchmarks in TAPE will provide a systematic model-evaluation framework that allows more machine learning researchers to contribute to this field.

Acknowledgments We thank Philippe Laban, David Chan, Jeffrey Spence, Jacob West-Roberts, Alex Crits-Cristoph, Aravind Srinivas, Surojit Biswas, Ethan Alley, Mohammed AlQuraishi and Grigory Khimulya for valuable input on this paper. We thank the AWS Educate program for providing us with the resources to train our models. Additionally, we acknowledge funding from Berkeley Deep Drive, Chan-Zuckerberg Biohub, DARPA XAI, NIH, the Packard Fellowship for Science and Engineering, and the Open Philanthropy Project.

References

[1] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 2018.

[2] C B Anfinsen, E Haber, M Sela, F H White, and Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):1309–14, 1961.

[3] C Yanofsky, V Horn, and D Thorpe. Protein Structure Relationships Revealed by Mutational Analysis. *Science (New York, N.Y.)*, 146(3651):1593–4, 1964.

[4] D Altschuh, T Vernet, P Berti, D Moras, and K Nagai. Coordinated amino acid changes in homologous protein families. *Protein engineering*, 2(3):193–9, 1988.

[5] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*, 2018.

[7] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

[8] Timothy A Whitehead, Aaron Chevalier, Yifan Song, Cyrille Dreyfus, Sarel J Fleishman, Cecilia De Mattos, Chris A Myers, Hetunandan Kamisetty, Patrick Blair, Ian A Wilson, et al. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nature biotechnology*, 30(6):543, 2012.

[9] Esther Vazquez, Neus Ferrer-Miralles, Ramon Mangues, Jose L Corchero, Jr Schwartz, Antonio Villaverde, et al. Modular protein engineering in emerging cancer therapies. *Current pharmaceutical design*, 15(8):893–916, 2009.

[10] Nelson Perdigão, Julian Heinrich, Christian Stolte, Kenneth S. Sabir, Michael J. Buckley, Bruce Tabor, Beth Signal, Brian S. Gloss, Christopher J. Hammang, Burkhard Rost, Andrea Schafferhans, and Seán I. O'Donoghue. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences*, 112(52):15898–15903, 2015.

[11] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2019.

[12] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, page 589333, 2019.

[13] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[14] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2):173–175, 2012.

[15] J. Soding, A. Biegert, and A. N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research*, 33(Web Server):W244–W248, 2005.

[16] Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763, 1998.

[17] IUPAC-IUB. Recommendations on nomenclature and symbolism for amino acids and peptides. *Pure Appl. Chem*, 56:595–623, 1984.

[18] Thomas E Creighton. *Proteins: structures and molecular properties*. Macmillan, 1993.

[19] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, 1999.

[20] Steven E Brenner, Cyrus Chothia, and Tim JP Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences*, 95(11):6073–6078, 1998.

[21] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[22] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[23] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[24] Konstantin Arnold, Lorenza Bordoli, Jürgen Kopp, and Torsten Schwede. The swiss-model workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, 2006.

[25] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[26] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, 2018.

[27] Mohammed AlQuraishi. ProteinNet: a standardized data set for machine learning of protein structure. *bioRxiv*, 2019.

[28] Jason Weston, Dengyong Zhou, André Elisseeff, William S Noble, and Christina S Leslie. Semi-supervised protein classification using cluster kernels. In *Advances in neural information processing systems*, pages 595–602, 2004.

[29] Hyunjung Shin, Koji Tsuda, B Schölkopf, A Zien, et al. Prediction of protein function from networks. In *Semi-supervised learning*, pages 361–376. MIT press, 2006.

[30] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian 4 Dallago, Dmitrii Nachaev, Florian Matthes, and & Burkhard Rost. Modeling the Language of Life - Deep Learning Protein Sequences. *bioRxiv*, 2019.

[31] Alexander Rives, Siddharth Goyal, Joshua Meier, 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. *bioRxiv*, 2019.

[32] Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, 2018.

[33] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

[34] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

[35] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, Erik L L Sonnhammer, Layla Hirsh, Lisanna Paladin, Damiano Piovesan, Silvio C E Tosatto, and Robert D Finn. The Pfam protein families database in 2019. *Nucleic Acids Research*, 47(D1):D427–D432, 2019.

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

[37] Alexey Drozdetskiy, Christian Cole, James Procter, and Geoffrey J. Barton. JPred4: a protein secondary structure prediction server. *Nucleic Acids Research*, 43(W1):W389–W394, 2015.

[38] James A Cuff and Geoffrey J Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[39] David E. Kim, Frank DiMaio, Ray Yu-Ruei Wang, Yifan Song, and David Baker. One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins: Structure, Function, and Bioinformatics*, 82:208–218, 2014.

[40] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2017.

[41] Leticia Stephan Tavares, Carolina dos Santos Fernandes da Silva, Vinicius Carius Souza, Vânia Lúcia da Silva, Cláudio Galuppo Diniz, and Marcelo De Oliveira Santos. Strategies and molecular tools to fight antimicrobial resistance: resistome, transcriptome, and antimicrobial peptides. *Frontiers in microbiology*, 4:412, 2013.

[42] Jun-Jie Liu, Natalia Orlova, Benjamin L Oakes, Enbo Ma, Hannah B Spinner, Katherine LM Baney, Jonathan Chuck, Dan Tan, Gavin J Knott, Lucas B Harrington, et al. Casx enzymes comprise a distinct family of rna-guided genome editors. *Nature*, 566(7743):218, 2019.

[43] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397, 2016.

[44] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine learning in protein engineering. *arXiv preprint arXiv:1811.10775*, 2018.

[45] Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

[46] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[47] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Journal of Machine Learning Research*, 2013.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[50] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*, 2016.

[52] Joerg Schaarschmidt, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Alexandre MJJ Bonvin. Assessment of contact predictions in casp12: Co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*, 86:51–66, 2018.

[53] Jianzhu Ma, Sheng Wang, Zhiyong Wang, and Jinbo Xu. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. *Bioinformatics*, 31(21):3506–3513, 2015.

[54] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)—round xii. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, 2018.

[55] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in bioinformatics*, 19(3):482–494, 2016.

[56] Raymond C Stevens. The cost and value of three-dimensional protein structure. *Drug Discovery World*, 4(3):35–48, 2003.

[57] Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2013.

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

[59] Cindy J Castelle and Jillian F Banfield. Major new microbial groups expand diversity and alter our understanding of the tree of life. *Cell*, 172(6):1181–1197, 2018.

[60] Stefan Seemayer, Markus Gruber, and Johannes Söding. CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 2014.

[61] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training Deep Nets with Sublinear Memory Cost. *arXiv*, 2016.

[62] Robbie P Joosten, Tim AH Te Beek, Elmar Krieger, Maarten L Hekkelman, Rob WW Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. A series of pdb related databases for everyday needs. *Nucleic acids research*, 39(suppl_1):D411–D419, 2010.

Table S1: Dataset sizes

Task	Train	Valid	Test
Language Modeling	32,207,059	N/A	2,147,130 (Random-split) / 44,314 (Heldout families)
Secondary Structure	8,678	2,170	513 (CB513) / 115 (TS115) / 21 (CASP12)
Contact Prediction	25,299	224	40 (CASP12)
Remote Homology	12,312	736	718 (Fold) / 1,254 (Superfamily) / 1,272 (Family)
Fluorescence	21,446	5,362	27,217
Stability	53,679	2,447	12,839

A Appendix

A.1 Dataset Details

In Table S1 we show the size of all train, validation, and test sets.

We provide further details about dataset sources, preprocessing decisions, data splitting, and experimental challenges in obtaining labels for each of our supervised tasks below. For ease of reading, each section starts with the following items:

(Dataset) The source of the dataset and creation of train/test splits.
(Labeling) The current approach to acquiring supervised labels for this task.

A.1.1 Secondary Structure Details

(Dataset) We use a training and validation set from Klausen et al. [36], which is filtered such that no two proteins have greater than 25% sequence identity. We use three test sets, CB513 [38], CASP12 [54], and TS115 [55]. The training set is also filtered at the 25% identity threshold with these test sets. This filtering tests the model's ability to generalize in the interesting case where test proteins are not closely related to train proteins.

(Labeling) Determining the secondary structure of a protein experimentally requires high-resolution imaging of the structure, a particularly labor intensive task for structural biologists. Imaging often uses Cryo Electron-Microscopy or X-Ray Crystallography, which can take between weeks and years and can cost over $\$200,000$ [56].

A.1.2 Contact Prediction Details

(Dataset) We use training, validation, and test sets from ProteinNet [27], which uses a test set based on the CASP12 [54] competition, with training and validation sets filtered at the 30% sequence identity threshold. This tests the ability of the model to generalize to proteins that are not closely related to any train proteins.

(Labeling) Determining the contacts of a protein requires knowing its full 3D structure. As with secondary structure, determining the 3D structure requires imaging a protein.

A.1.3 Remote Homology Details

(Dataset) We use a training, validation, and test set from [40], derived from the SCOP 1.75 database [57] of hierarchically classified protein domains. All proteins of a given fold are further categorized into related *superfamilies*. Entire superfamilies are held out from the training set, allowing us to evaluate how the model generalizes across evolutionary distance when structure is preserved.

(Labeling) Each fold is annotated from the structure of the sequence, which SCOP pulls from the Protein DataBank [57, 58]. Finding new superfamilies with the same fold is a challenging task, requiring sequencing in extreme environments as is often done in metagenomics [59].

A.1.4 Fluorescence Details

(Dataset) We use data generated by an experimental technique called Deep Mutational Scanning (DMS) [43]. This technique allows for extensive characterizations of small neighborhoods of a parent protein through mutagenesis. We create training, validation, and test splits ourselves, partitioning the data so that train and validation are in a Hamming distance 3 neighborhood of the original protein, while test data is a sample from the Hamming distance 4-15 neighborhood.

(Labeling) DMS is efficient for local characterization near a single protein, but its samples become vanishingly small once neighborhoods start to expand outside of Hamming distance 2.

A.1.5 Stability Details

(Dataset) We use data generated by a novel combination of parallel DNA synthesis and protein stability measurements [45]. We create training, validation, and test splits ourselves, partitioning the data so that training and validation sets come from four rounds of experimental data measuring stability for many candidate proteins, while our test set consists of seventeen 1-Hamming distance neighborhoods around promising proteins observed in the four rounds of experimentation.

(Labeling) This approach for observing stability is powerful because of its throughput, allowing the authors to find the most stable proteins ever observed for certain classes [45]. The authors observe that the computational methods used to guide their selection at each stage could be improved, meaning that in this case better models could actually lead to better labeled data in a virtuous cycle.

A.2 Supervised Architectures

For each task, we fixed one supervised architecture and tried one-hot, alignment-based, and neural net based features. We did not perform hyperparameter tuning or significant architecture optimization, as the main goal was to compare feature extraction techniques.

For each task we define the supervised architecture below. If this is a state of the art architecture from other work, we highlight any novel training procedure or inputs they take.

A.2.1 Secondary Structure Architecture

We used the NetSurfP2.0 model from Klausen et al [36]. The model consists of two convolutional layers followed by two bidirection LSTM layers and a linear output layer. The convolutional layers have filter size 32 and kernel size 129 and 257, respectively. The bidirectional LSTM layers have 1024 hidden units each.

Our evolutionary features for secondary structure prediction are transition probabilities and state information from HHblits [14], an HMM-HMM alignment method. In the original model, the authors take HHblits outputs in addition to a one-hot encoding of the sequence, giving 50-dimensional inputs at each position. They train the model on multiple tasks including secondary structure prediction (3 and 8 class), bond-angle prediction, and solvent accessibility prediction. For clarity, we only compared to the model trained without the multitask training, which in our experiments contributed an extra one to two percent in test accuracy. In addition to multitask training, they balance the losses between different tasks to achieve maximum accuracy on secondary structure prediction. All features and code to do the full multitask training is available in our repository.

A.2.2 Contact Prediction Architecture

We used a supervised network inspired by the RaptorX-Contact model from Ma et al [53]. Since a contact map is a 2D pairwise prediction, we form a 2D input from our 1D features by concatenating the features at position i and j for all i, j. This 2D input is then passed through a convolutional residual network with. The 2D network contains 30 residual blocks with two convolutional layers each. Each convolution in the residual block has filter size 64 and a kernel size of 3.

Currently our evolutionary features for contact prediction are Position Specific Scoring Matrices (PSSMs) available in ProteinNet [27]. The original RaptorX method has a more complex evolutionary featurization: they construct a Multiple Sequence Alignment for each protein, then pass it through CCMpred [60] - a Markov Random Field based contact prediction method. This outputs a 2D featurization including mutual information and pairwise potential. This, along with 1D HMM alignment features and the one-hot encoding of each amino acid are fed to their network. We are currently recreating this pipeline to use these features instead of PSSMs, as the results reported by RaptorX are better than those with the PSSMs.

A.2.3 Remote Homology Architecture

Remote homology requires a single prediction for each protein. To obtain a sequence-length invariant protein embedding we compute an attention-weighted mean of the amino acid embeddings. More precisely, we predict an attention value for each position in the sequence using a trainable dense layer, then use those attention values to compute an attention-weighted mean protein embedding. This protein embedding is then passed through a 512 hidden unit dense layer, a relu nonlinearity, and a final linear output layer to predict logits for all 1195 classes. We note that Hou et al. [40] propose a deep architecture for this task and report state of the art performance. When we compared the performance of this supervised architecture to that of the attention-weighted mean above, the attention-based embedding performed better for all featurizations. As such, we choose to report results using the simpler attention-based downstream architecture.

Our evolutionary features for remote homology detection are Position Specific Scoring Matrices (PSSMs), following the recent work DeepSF [40]. The current state of the art method in this problem, DeepSF [40], takes in a one-hot encoding of the amino acids, predicted secondary structure labels, predicted solvent accessibility labels, and a 1D alignment-based features. In an ablation study, the authors show that the secondary structure labels are most useful for performance of their model. We report only one-hot and alignment-based results in the main paper to maintain consistency with alignment-based featurizations for other tasks. All input features used by DeepSF are available in our repository.

A.2.4 Protein Engineering Architectures

Protein engineering also requires a single prediction for each protein. Therefore we use the same architecture as we do for remote homology, computing an attention-weighted mean protein embedding, a dense layer with 512 hidden units, a relu nonlinearity and a final linear output layer to predict the quantity of interest (either stability or fluorescence).

Since we create these training, validation, and test splits ourselves, no clear previous state of the art exists. Related work on protein engineering has used a similar architecture by computing a single protein embedding followed by some form of projection (linear or with a small feed forward network) [12, 31]. These methods also do not take in alignment-based features and only use one-hot amino acids as inputs.

A.3 Training Details

Self-supervised models were all trained on four NVIDIA V100 GPUs on AWS for 1 week. Training used a learning rate of 10^{-3} with a linear warm up schedule, the Adam optimizer, and a 10% dropout rate. Since proteins vary in length significantly, we use variable batch sizes depending on the length of the protein. These sizes also differ based on model architecture, as some models (e.g. the Transformer) have significantly higher memory requirements. Specific batch sizes for each model at each protein length are available in our repository.

Supervised models were trained on two NVIDIA Titan Xp GPUs until convergence (no increase in validation accuracy for 10 epochs) with the exception of the memory-intensive Contact Prediction task, which was trained on two NVIDIA Titan RTX GPUs until convergence. Training used a learning rate of 10^{-4} with a linear warm up schedule, the Adam optimizer, and a 10% dropout rate. We backpropagated fully through all models during supervised fine-tuning.

In addition, due to high memory requirements of some downstream tasks (especially contact prediction) we use memory saving gradients [61] to fit more examples per batch on the GPU.

A.4 Pfam Heldout Families

The following Pfam clans were held out during self-supervised training: CL0635, CL0624, CL0355, CL0100, CL0417, CL0630. The following Pfam families were held out during self-supervised training: PF18346, PF14604, PF18697, PF03577, PF01112, PF03417. First, a “clan” is a cluster of families grouped by the maintainers of Pfam based on shared function or evolutionary origin (see [35] for details). We chose holdout clans and families in pairs, where a clan of novel function is held out together with a family that is similar in sequence but different evolutionarily or functionally. This

Table S2: Detailed secondary structure results

		Three-Way Accuracy (Q3)			Eight-Way Accuracy (Q8)		
		CB513	CASP12	TS115	CB513	CASP12	TS115
No Pretrain	Transformer	0.70	0.68	0.73	0.51	0.52	0.58
	LSTM	0.71	0.69	0.74	0.47	0.48	0.52
	ResNet	0.70	0.68	0.73	0.55	0.56	0.61
Pretrain	Transformer	0.73	0.71	0.77	0.59	0.59	0.64
	LSTM	0.75	0.70	0.78	0.59	0.57	0.66
	ResNet	0.75	0.72	0.78	0.58	0.58	0.64
Supervised [11]	LSTM	0.73	0.70	0.76	0.58	0.57	0.65
UniRep [12]	mLSTM	0.73	0.72	0.77	0.57	0.59	0.63
Baseline	One-hot	0.69	0.68	0.72	0.52	0.53	0.58
	Alignment	**0.8**	**0.76**	**0.81**	**0.63**	**0.61**	**0.68**

serves to simultaneously test generalization across large distances (entirely held out families) and between similar looking unseen groups (e.g. the paired holdout clan and holdout family).

A.5 Bepler Supervised Training

We perform supervised pretraining using the same architecture described in Bepler et al. [11]. We train on the same tasks, a paired remote homology task and contact map prediction task. However, in order to accurately report results on downstream secondary structure, contact map, and remote homology datasets, which were filtered by sequence identity, we perform this same sequence identity filtering on the supervised pretraining set. This reduced the supervised pretraining dataset size by 75% which likely reduced the effectiveness of the supervised pretraining. Both filtered and unfiltered supervised pretraining datasets are made available in our repository.

A.6 Detailed Results on Supervised Tasks

Here we provide detailed results on each task, examining multiple metrics and test-conditions to further determine what the models are learning.

A.6.1 Secondary Structure Results

We perform both three-class and eight-class secondary structure classification following the DSSP labeling system [62]. Three way classification tags each position as either Helix, Strand or Other. Eight-way classification breaks these three labels into more specialized classes, for example Helix is broken into 3-turn, 4-turn or 5-turn helix. Table S2 shows results on these tasks. We note that test-set performance is comparable for all three test sets, in particular alignment does better at both eight-way and three-way classification by a large margin.

We follow the standard notation, where Q3 refers to three-way classification accuracy and Q8 refers to eight-way classification accuracy.

A.6.2 Contact Prediction Results

We report all metrics commonly used to capture contact prediction results [53] in tables S4 and S5. The metrics "P@K" are precision for the top K contacts, where all contacts are sorted from highest confidence to lowest confidence. Note that L is the length of the protein, so "P@L/2", for example, denotes the precision for the $L/2$ most likely predicted contacts in a protein of length L. In Table S4 we report all metrics for medium range contacts, which are contacts for positions between five and twelve amino acids apart. In Table S5 we report all metrics for long range contacts, which are contacts for positions greater than 12 amino acids apart.

All results decay as we transition from short range to long range contacts, which we note is *not* the case for many state of the art methods from recent CASP competitions [52, 53].

Table S3: Detailed short-range contact prediction results. Short range contacts are contacts between positions separated by 6 to 11 positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.29	0.25	0.32	0.4
	LSTM	0.23	0.22	0.26	0.33
	ResNet	0.2	0.18	0.24	0.31
Pretrain	Transformer	0.35	0.28	0.35	0.46
	LSTM	0.35	0.26	0.36	0.49
	ResNet	0.32	0.25	0.34	0.46
Supervised [11]	LSTM	0.33	0.27	0.35	0.44
UniRep [12]	mLSTM	0.27	0.23	0.3	0.39
Baseline	One-hot	0.3	0.26	0.34	0.42
	Alignment	**0.51**	**0.35**	**0.5**	**0.66**

Table S4: Detailed medium-range contact prediction results. Medium range contacts are contacts between positions separated by 12 to 23 positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.2	0.18	0.24	0.31
	LSTM	0.13	0.13	0.15	0.19
	ResNet	0.15	0.14	0.18	0.23
Pretrain	Transformer	0.23	0.19	0.25	0.33
	LSTM	0.23	0.2	0.26	0.34
	ResNet	0.23	0.18	0.25	0.35
Supervised [11]	LSTM	0.26	0.22	0.29	0.37
UniRep [12]	mLSTM	0.2	0.17	0.23	0.32
Baseline	One-hot	0.2	0.17	0.23	0.3
	Alignment	**0.45**	**0.32**	**0.45**	**0.59**

Table S5: Detailed long-range contact prediction results. Long range contacts are contacts between positions separated by 24 or more positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.09	0.15	0.17	0.19
	LSTM	0.05	0.1	0.12	0.15
	ResNet	0.06	0.11	0.13	0.15
Pretrain	Transformer	0.1	0.17	0.2	0.24
	LSTM	0.11	0.2	0.23	0.27
	ResNet	0.06	0.1	0.13	0.17
Supervised [11]	LSTM	0.11	0.18	0.22	0.26
UniRep [12]	mLSTM	0.09	0.17	0.2	0.22
Baseline	One-hot	0.07	0.13	0.16	0.22
	Alignment	**0.2**	**0.33**	**0.42**	**0.51**

Table S6: Detailed remote homology prediction results

		Fold		Superfamily		Family	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
No Pretrain	Transformer	0.09	0.21	0.07	0.2	0.31	0.58
	LSTM	0.12	0.28	0.13	0.29	0.68	0.85
	ResNet	0.1	0.24	0.07	0.19	0.39	0.6
Pretrain	Transformer	0.21	0.37	0.34	0.51	0.88	0.94
	LSTM	**0.26**	**0.43**	**0.43**	**0.59**	**0.92**	**0.97**
	ResNet	0.17	0.29	0.31	0.44	0.77	0.87
Supervised [11]	LSTM	0.17	0.30	0.20	0.36	0.79	0.91
UniRep [12]	mLSTM	0.23	0.39	0.38	0.54	0.87	0.94
Baseline	One-hot	0.09	0.21	0.08	0.21	0.39	0.66
	Alignment	0.09	0.21	0.09	0.24	0.53	0.77

Table S7: Detailed fluorescence prediction results. ρ denotes Spearman ρ.

		Full Test Set		Bright Mode Only		Dark Mode Only	
		MSE	ρ	MSE	ρ	MSE	ρ
No Pretrain	Transformer	2.59	0.22	0.08	0.08	3.79	0
	LSTM	2.35	0.21	0.11	0.05	3.43	-0.01
	ResNet	2.79	-0.28	**0.07**	-0.07	4.1	-0.01
Pretrain	Transformer	0.22	**0.68**	0.09	0.60	0.29	**0.05**
	LSTM	**0.19**	0.67	0.12	0.62	**0.22**	0.04
	ResNet	3.04	0.21	0.12	0.05	4.45	0.02
Supervised [11]	LSTM	2.17	0.33	0.08	0.06	3.17	0.02
UniRep [12]	mLSTM	0.20	0.67	0.13	**0.63**	0.24	0.04
Baseline	One-hot	2.69	0.14	0.08	0.03	3.95	0.0

A.6.3 Remote Homology Results

In Table S6, we report results on three remote homology test datasets constructed in Hou et al [40]. Recall that “Fold” has the most distantly related proteins from train, while “Superfamily” and “Family” are increasingly related (see Section A.1.3 for more details). This is reflected in the accuracies in Table S6, which increase drastically as the test sets get easier.

A.6.4 Fluorescence Results

Fluorescence distribution in the train, validation, and test sets is bimodal, with one mode corresponding to bright proteins and one mode corresponding to dark proteins. The dark mode is significantly more diverse in the test set than the train and validation sets, which makes sense as most random mutations will destroy the refined structure necessary for fluorescence. With this in mind, we report Spearman’s ρ and mean-squared-error (MSE) on the whole test-set, on only dark mode, and on only the bright mode in Table S7. The drop in MSE for both modes shows that pretraining helps our best models distinguish between dark and bright proteins. However low Spearman’s ρ for the dark mode suggests that models are not able to rank proteins within this mode.

Table S8: Overall stability prediction results

		Spearman's ρ	Accuracy
No Pretrain	Transformer	-0.06	0.5
	LSTM	0.28	0.6
	ResNet	0.61	0.68
Pretrain	Transformer	**0.73**	**0.70**
	LSTM	0.69	0.69
	ResNet	**0.73**	0.66
Supervised [11]	LSTM	0.64	0.67
UniRep [12]	mLSTM	**0.73**	0.69
Baseline	One-hot	0.19	0.58

A.6.5 Stability Results

Table S9: Stability prediction results broken down by protein topology

		$\alpha\alpha\alpha$		$\alpha\beta\beta\alpha$		$\beta\alpha\beta\beta$		$\beta\beta\alpha\beta\beta$	
		ρ	Acc	ρ	Acc	ρ	Acc	ρ	Acc
No Pretrain	Transformer	-0.39	0.49	-0.41	0.47	0.52	0.5	0.25	0.52
	LSTM	-0.07	0.57	0.39	0.7	-0.43	0.56	-0.34	0.56
	ResNet	0.64	0.69	0.16	0.69	0.63	0.67	0.65	0.67
Pretrain	Transformer	0.66	0.68	**0.48**	0.73	0.65	**0.71**	0.65	0.67
	LSTM	0.71	**0.7**	0.17	0.73	**0.68**	0.67	**0.67**	**0.7**
	ResNet	0.68	0.68	0.15	0.63	0.61	0.68	0.6	0.68
Supervised [11]	LSTM	0.33	0.66	0.24	**0.79**	0.54	0.7	0.58	0.53
UniRep [12]	mLSTM	**0.72**	0.66	0.11	0.76	**0.68**	0.66	0.65	0.67
Baseline	One-hot	0.58	0.59	0.04	0.58	-0.05	0.58	0.54	0.58

The goal of the Rocklin et al. [45] experiment was to find highly stable proteins. In the last stage of this experiment they examine variants of the the most promising candidate proteins. Therefore we wish to measure both whether our model was able to learn the landscape around these candidate proteins, as well as whether it successfully identified those variants with greater stability than the original parent proteins. In Table S8 we report Spearman's ρ to measure the degree to which the landscape was learned. In addition, we report classification accuracy of whether a mutation is beneficial or harmful using the predicted stability of the parent protein as a decision boundary.

In Table S9 report all metrics separately for each of the four protein topologies tested in Rocklin et al [45], where α denotes a helix and β denotes a strand (or β-sheet). We do this because success rates varied significantly by topology in their experiments, so some topologies (such as $\alpha\alpha\alpha$ were much easier to optimize than others (such as $\alpha\beta\beta\alpha$). We find that our prediction success also varies significantly by topology.