TRANSFORMER PROTEIN LANGUAGE MODELS ARE UNSUPERVISED STRUCTURE LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised contact prediction is central to uncovering physical, structural, and functional constraints for protein structure determination and design. For decades, the predominant approach has been to infer evolutionary constraints from a set of related sequences. In the past year, protein language models have emerged as a potential alternative, but performance has fallen short of state-of-the-art approaches in bioinformatics. In this paper we demonstrate that Transformer attention maps learn contacts from the unsupervised language modeling objective. We find the highest capacity models that have been trained to date already outperform a state-of-the-art unsupervised contact prediction pipeline, suggesting these pipelines can be replaced with a single forward pass of an end-to-end model.

1 INTRODUCTION

Unsupervised modeling of protein contacts has an important role in computational protein design (Russ et al., 2020; Tian et al., 2018; Blazejewski et al., 2019) and is a central element of all current state-of-the-art structure prediction methods (Wang et al., 2017; Senior et al., 2020; Yang et al., 2019). The standard bioinformatics pipeline for unsupervised contact prediction includes multiple components with specialized tools and databases that have been developed and optimized over decades. In this work we propose replacing the current multi-stage pipeline with a single forward pass of a pre-trained end-to-end protein language model.

In the last year, protein language modeling with an unsupervised training objective has been investigated by multiple groups (Rives et al., 2019; Alley et al., 2019; Heinzinger et al., 2019; Rao et al., 2019; Madani et al., 2020). The longstanding practice in bioinformatics has been to fit linear models on focused sets of evolutionarily related and aligned sequences; by contrast, protein language modeling trains nonlinear deep neural networks on large databases of evolutionarily diverse and unaligned sequences. High capacity protein language models have been shown to learn underlying intrinsic properties of proteins such as structure and function from sequence data (Rives et al., 2019).

A line of work in this emerging field proposes the Transformer for protein language modeling (Rives et al., 2019; Rao et al., 2019). Originally developed in the NLP community to represent long range context, the main innovation of the Transformer model is its use of self-attention (Vaswani et al., 2017). Self-attention has particular relevance for the modeling of protein sequences. Unlike convolutional and recurrent LSTM models, the Transformer constructs a pairwise interaction map between all positions in the sequence. In principle this mechanism has an ideal form to model residue-residue contacts.

In theory, end-to-end learning with a language model has advantages over the bioinformatics pipeline: (i) it replaces the expensive query, alignment, and training steps with a single forward pass, greatly accelerating feature extraction; and (ii) it shares parameters for all protein families, enabling generalization by capturing commonality across millions of evolutionarily diverse and unrelated sequences.

We demonstrate that Transformer protein language models learn contacts in the self-attention maps with state-of-the-art performance. We compare ESM-1b, a large-scale (650M parameters) Transformer model trained on UniRef50 (Suzek et al., 2007) to the Gremlin (Kamisetty et al., 2013) pipeline which implements a log linear model trained with pseudolikelihood (Balakrishnan et al., 2011; Ekeberg et al., 2013). Contacts can be extracted from the attention maps of the Transformer

model by a sparse linear combination of attention heads identified by logistic regression. ESM-1b model contacts have higher precision than Gremlin contacts. When ESM and Gremlin are compared with access to the same set of sequences the precision gain from the protein language model is significant; the advantage holds on average even when Gremlin is given access to an optimized set of multiple sequence alignments incorporating metagenomics data.

We find a linear relationship between language modeling perplexity and contact precision. We also find evidence for the value of parameter sharing: the ESM-1b model significantly outperforms Gremlin on proteins with low-depth MSAs. Finally we explore the Transformer language model's ability to generate sequences and show that generated sequences preserve contact information.

2 BACKGROUND

Protein Language Modeling Since the introduction of large scale language models for natural language processing (Vaswani et al., 2017; Devlin et al., 2019), there has been considerable interest in developing similar models for proteins. Many of these methods have shown promising initial results, from structure prediction (Rives et al., 2019) to protein engineering (Yang et al., 2018; Biswas et al., 2020). In addition to performance on downstream prediction tasks, Vig et al. (2020) showed that the TAPE Transformer learns to represent some contact information via attention maps.

Multiple Sequence Alignments (MSAs) A multiple sequence alignment consists of a set of evolutionarily related protein sequences. Since real protein sequences are likely to have insertions, deletions, and substitutions, the sequences are *aligned* by minimizing a Levenshtein distance-like metric over all the sequences. In practice heuristic alignment schemes are used. Potential sources of error include (i) failure of prefiltering heuristics; (ii) undetected related sequences; (iii) failure to find an optimal alignment; (iv) suboptimality of the substitution matrix and gap penalty. Tools like Jackhmmer and HHblits can help mitigate these errors by iteratively performing the search and alignment steps (Johnson et al., 2010; Remmert et al., 2012).

Metrics For a protein of length L, we evaluate the precision of the top L, $L/2$, and $L/5$ contacts for short range ($|i-j| \in [6, 12)$), medium range ($|i-j| \in [12, 24)$), and long range ($|i = j| \in [24, \infty)$) contacts. We also separately evaluate local contacts ($|i-j| \in [3, 6)$) for secondary structure prediction in Section 4.5. In general, all contacts provide information about protein structure and important interactions, with shorter-range contacts being useful for secondary and local structure, while longer range contacts are useful for determining global structure (Taylor et al., 2014).

3 MODELS

We compare Transformer models trained on large sequence databases to Potts Models trained on individual MSAs. While Transformers and Potts Models emerged in separate research communities, the two models share core similarities (Wang & Cho, 2019) which we exploit here. Our main result is that just as Gremlin directly represents contacts via its pairwise component (the weights), the Transformer also directly represents contacts via its pairwise component (the self-attention).

3.1 OBJECTIVES

For a set of training sequences, X, Gremlin optimizes the following pseudolikelihood loss, where a single position is masked and predicted from its context. Since inputs to Gremlin are aligned they all have length L:

$$\mathcal{L}_{\text{PLL}}(X;\theta) = \mathop{\mathbb{E}}_{x\sim X} \sum_{i=1}^{L} \log p(x_i|x_{j\neq i};\theta) \tag{1}$$

The masked language modeling (MLM) loss used by the Transformer models can be seen as a generalization of the Potts Model objective when written as follows:

$$\mathcal{L}_{\text{MLM}}(X;\theta) = \mathop{\mathbb{E}}_{x\sim X} \mathop{\mathbb{E}}_{\text{mask}} \sum_{i\in\text{mask}} \log p(x_i|x_{j\notin\text{mask}};\theta) \tag{2}$$

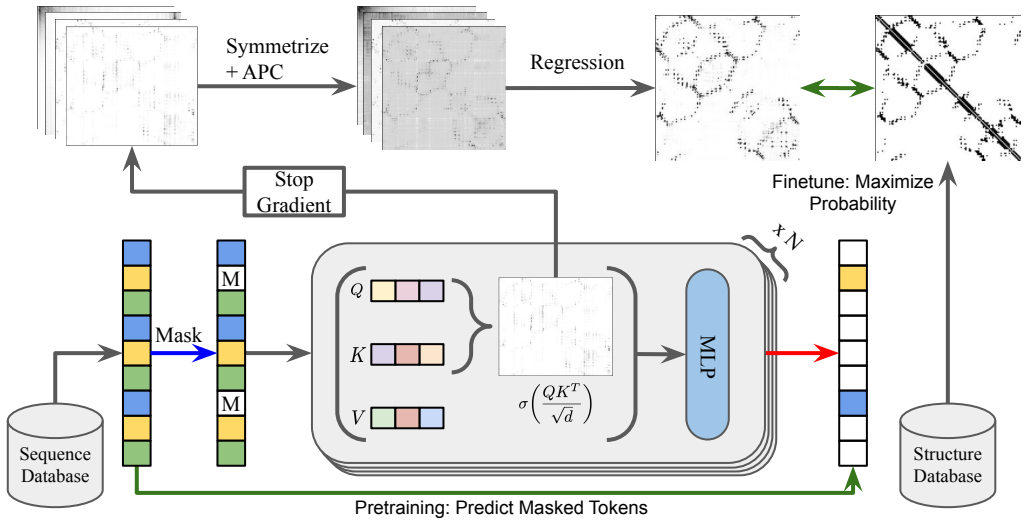


Figure 1: Contact prediction pipeline. The Transformer is first pretrained on sequences from a large database (Uniref50) via Masked Language Modeling. Once finished training, the attention maps are extracted, passed through symmetrization and average product correction, then into a regression. The regression is trained on a small number ($n \leq 20$) of proteins to determine which attention heads are informative. At test time, contact prediction from an input sequence can be done entirely on GPU in a single forward pass.

In contrast to Gremlin, the MLM objective applied by protein language modeling is trained on unaligned sequences. The key distinction of MLM is to mask and predict multiple positions concurrently, instead of masking and predicting one at a time. This enables the model to scale beyond individual MSAs to massive sequence datasets. In practice, the expectation under the masking pattern is computed stochastically using a single sample at each epoch.

3.2 Gremlin

The log probability optimized by Gremlin is described in section A.4. Contacts are extracted from the pairwise Gremlin parameters by taking the Frobenius norm along the amino acid dimensions, resulting in an $L \times L$ coupling matrix. Average product correction (APC) is applied to this coupling matrix to determine the final predictions.

Gremlin takes an MSA as input. The quality of the output predictions are highly dependent on the construction of the MSA. We compare to Gremlin under two conditions. In the first condition, we present Gremlin with all MSAs from the trRosetta training set (Yang et al., 2019). These MSAs were generated from all of Uniref100 and are also supplemented with metagenomic sequences when the depth from Uniref100 is too low. The trRosetta MSAs are a key ingredient in the state-of-the-art protein folding pipeline. See Yang et al. (2019) for a discussion on the significant impact of metagenomic sequences on the final result. In the second setting, we allow Gremlin access only to the same information as the ESM Transformers by generating MSAs via Jackhmmer on the ESM training set (a subset of Uniref50). We used a bitscore threshold of 27 and 8 iterations when searching for sequences.

3.3 Transformers

We evaluate several pre-trained Transformer models, including ESM-1 (Rives et al., 2019), ProtBert-BFD (Elnaggar et al., 2020) and the TAPE Transformer (Rao et al., 2019). The key differences between these models are the datasets, model sizes, and hyperparameters. Liu et al. (2019) previously showed that these changes can have a significant impact on final model performance. In addition to ESM-1, we also evaluate an updated version, ESM-1b, provided to us by the authors. ESM-1b is the result of a hyperparameter sweep and the differences are described in section A.5. The Transformer

Table 1: Average precision on 14842 test structures for Transformer models trained on 20 structures.

	$6 \leq$ sep < 12			$12 \leq$ sep < 24			$24 \leq$ sep		
Model	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Gremlin (ESM Data)	15.2	23.0	37.8	18.1	27.9	44.3	31.3	43.1	55.5
Gremlin (trRosetta Data)	17.2	26.7	44.4	21.1	33.3	52.3	39.3	52.2	62.8
TAPE	9.9	12.3	16.4	10.0	12.6	16.6	11.2	14.0	17.9
ProtBERT-BFD	20.4	30.7	48.4	24.3	35.5	52.0	34.1	45.0	57.4
ESM-1 (6 layer)	11.0	13.2	15.9	11.5	14.6	19.0	13.2	16.7	21.5
ESM-1 (12 layer)	15.2	21.1	30.5	18.1	24.7	34.0	23.7	30.5	39.3
ESM-1 (34 layer)	20.3	30.2	46.0	23.8	34.3	49.2	34.7	44.6	56.0
ESM-1b	**21.6**	**33.2**	**52.7**	**26.2**	**38.6**	**56.4**	**41.1**	**53.3**	**66.1**

processes inputs through a series of blocks alternating multi-head self-attention and feed-forward layers. In each head of a self-attention layer, the Transformer views the encoded representation as a set of query-key-value triples. The output of the head is the result of scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{n}) \cdot V$$

Rather than only computing the attention once, the multi-head approach runs scaled dot-product attention multiple times in parallel and concatenates the output. Since self-attention explicitly constructs pairwise interactions (QK^T) between all positions in the sequence, the model can directly represent residue-residue interactions. In this work, we demonstrate that the QK^T pairwise "self attention maps" indeed capture accurate contacts.

3.4 Logistic Regression

To extract contacts from a Transformer, we first pass the input sequence through the model to obtain the attention maps (one map for each head in each layer). We then symmetrize and apply APC to each attention map independently. The resulting maps are passed through an L_1-regularized logistic regression, which is applied independently at each amino acid pair (i, j). At training time, we only train the weights of the logistic regression; we do not backpropagate through the entire model. At test time, the entire prediction pipeline can be run in a single forward pass, providing a single end-to-end pipeline for protein contact prediction that does not require any retrieval steps from a sequence database. See Figure 1 for an illustration of this pipeline and Section A.6 for a full description of the logistic regression setup.

4 Results

We evaluate models with the 15051 proteins in the trRosetta training dataset (Yang et al., 2019), removing 43 proteins with sequence length greater than 1024, since ESM-1b was trained with a context size of 1024. Of these sequences, Jackhmmer fails on 126 when we attempt to construct MSAs using the ESM training set. This leaves us with 14882 total sequences. We reserve 20 sequences for training, 20 sequences for validation, and 14842 sequences for testing.

Table 1 shows evaluations of Gremlin, ESM-1, ESM-1b as well as the TAPE and ProtBERT-BFD models. Confidence intervals are within 0.5 percentage points for all statistics in Tables 1 and 2. In Table 1, all Transformer model contact predictors are trained with logistic regression on 20 proteins. We find that with only 20 training proteins ESM-1b has higher precision than Gremlin for short, medium, and long range contacts.

4.1 Ablations: Limiting supervision

While the language modeling objective is fully unsupervised, the logistic regression is trained with a small number of supervised examples. In this section, we study the dependence of the results on this supervision, providing evidence that the contacts are indeed learned in the unsupervised phase, and the logistic regression is only necessary to extract the contacts.

Table 2: ESM-1b Ablations with limited supervision and with MSA information. n is the number of logistic regression training proteins. s is the number of sequences ensembled over.

		$6 \leq$ sep < 12			$12 \leq$ sep < 24			$24 \leq$ sep		
Model	Variant	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Gremlin	ESM Data	15.2	23.0	37.8	18.1	27.9	44.3	31.3	43.1	55.5
	trRosetta Data	17.2	26.7	44.4	21.1	33.3	52.3	39.3	52.2	62.8
ESM-1b (Ablations)	top-1 heads	16.8	23.4	34.8	19.8	27.6	40.2	29.3	38.1	50.0
	top-5 heads	19.2	28.5	44.5	23.3	33.8	49.0	35.0	45.2	57.3
	top-10 heads	20.0	30.1	47.4	24.7	36.0	52.2	38.5	49.4	61.1
	n=1, s=1	19.4	29.7	47.1	25.1	37.1	54.0	39.2	50.6	63.0
	n=10, s=1	21.4	32.9	52.3	26.1	38.5	56.4	40.8	52.9	65.7
	n=20, s=1	21.6	33.2	52.7	26.2	38.6	56.4	41.1	53.3	66.1
	MSA, s=1	18.4	28.1	45.5	23.9	36.1	53.7	39.9	51.3	63.0
ESM-1b (s seqs)	n=20, s=16	21.9	33.8	53.6	26.7	39.4	57.5	41.9	54.3	67.3
	n=20, s=32	22.0	34.1	54.0	26.9	39.8	58.1	42.3	54.8	67.8
	n=20, s=64	**22.1**	**34.3**	**54.3**	**27.1**	**40.1**	**58.5**	**42.6**	**55.1**	**68.2**

Top Heads Here we use the logistic regression only to determine the most important heads. Once they are selected, we discard the weights from the logistic regression and simply average the attention heads corresponding to the top-k weight values. By taking the single best head from ESM-1b, we come close to Gremlin performance given the same data, and averaging the top-5 heads allows us to outperform Gremlin. Averaging the top-10 heads outperforms a full logistic regression on all other Transformer models and comes close to Gremlin given optimized MSAs.

Low-N The second variation we consider is to limit the number of supervised examples provided to the logistic regression. We find that with **only a single training example**, the model achieves a long range top-L precision of 39.2, which is statistically indistinguishable from Gremlin ($p > 0.05$). Using only 10 training examples, the model outperforms Gremlin on all the metrics.

MSA Only Finally, we consider supervising the logistic regression only with MSAs instead of real structures. This is the same training data used by the Gremlin baseline. To do this, we first train Gremlin on each MSA. We take the output couplings from Gremlin and mark the top L couplings with sequence separation ≥ 6 in each protein as true contacts, and everything else as false contacts, creating a binary decision problem. When trained on 20 MSAs, we find that this model achieves a long range P@L of 39.9, and generally achieves similar long range performance to Gremlin, while still having superior short and medium range contact precision.

4.2 Ensembling over MSA

Transformer models are fundamentally single-sequence models, but we can further boost performance by ensembling predictions from multiple sequences in the alignment. To do so, we unalign each sequence in the alignment (removing any gaps), pass the resulting sequence through the Transformer and regression, and realign the resulting contact maps to the original aligned indices. For these experiments, we use the logistic regression weights trained on single-sequence inputs, rather than re-training the logistic regression on multi-sequence inputs. We also simply take the first s sequences in the MSA. Table 2 shows performance improvements from averaging over 16, 32, and 64 sequences.

To better understand this result, we return to the single-sequence setting and study the change in prediction when switching between sequences in the alignment. We find that contact precision can vary significantly depending on the exact sequence input to the model, and that the initial query sequence of the MSA does not necessarily generate the highest contact precision (Figure 8).

Lastly, Alley et al. (2019) presented a method of fine-tuning where a pretrained language model is further trained on the MSA of the sequence of interest ('evotuning'). Previously this has only been

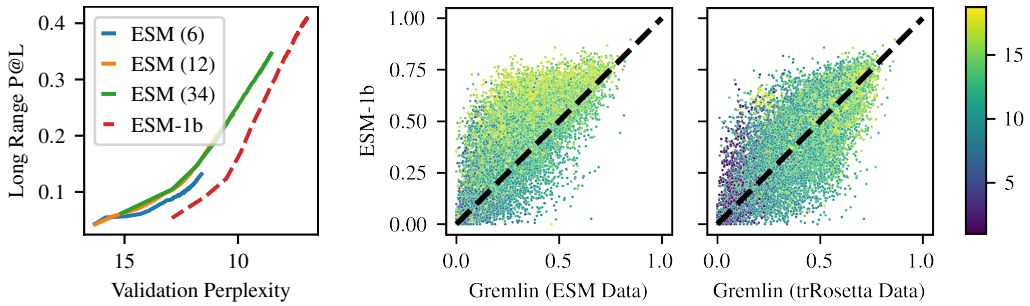


Figure 2: Left: Language modeling validation perplexity on holdout of Uniref50 vs. contact precision over the course of pre-training. ESM-1b was trained with different masking so perplexities between the versions are not comparable. Right: Long range P@L performance distribution of ESM-1b vs. Gremlin. Each point is colored by the log of the number of sequences in the MSA used to train Gremlin.

investigated for function prediction and for relatively low-capacity models. We fine-tune the full ESM-1b model (which has 50x more parameters than UniRep) on 380 protein sequence families. We find that after 30 epochs of fine-tuning, long range P@L increases only slightly, with an average of 1.6 percentage points (Figure 13).

4.3 Performance Distribution

Although our model is, on average, better than Gremlin at detecting contacts, the performance distribution over all sequences in the dataset is still mixed. ESM-1b is consistently better at extracting short and medium range contacts (Figure 6), but only slightly outperforms Gremlin on long range contacts when Gremlin has access to Uniref100 and metagenomic sequences. Figure 2 shows the distribution of long range P@L for ESM-1b vs. Gremlin. Overall, ESM-1b has higher long range P@L on 55% of sequences in the test set.

In addition, we examine the relationship between MSA depth and precision for short, medium, and long range contacts (Figure 3). Although our contact prediction pipeline does not make explicit use of MSAs, there is still some correlation between MSA depth and performance, since MSA depth is a measure of how many related sequences are present in the ESM-1b training set. We again see that ESM-1b consistently outperforms Gremlin at all MSA depths for short and medium range sequences. We also confirm that ESM-1b outperforms Gremlin for long range contact extraction for sequences with small MSAs (depth < 1000). ESM-1b also outperforms Gremlin on sequences with the very largest MSAs (depth > 16000), which is consistent with prior work showing that Gremlin performance plateaus for very large MSAs and suggests that ESM-1b does not suffer from the same issues (Anishchenko et al., 2017).

4.4 Logistic Regression Weights

In Section 4.1 we show that selecting only a sparse subset of the attention heads can yield good results for contact prediction. Overall, the L_1-regularized logistic regression identifies 102 / 660 heads as being predictive of contacts (Figure 5b). Additionally, we train separate logistic regressions to identify contacts at different ranges (local, short, medium, and long range). These regressions identify an overlapping, but non-identical set of useful attention heads. Two attention heads have the top-10 highest weights for detecting contacts at all ranges. One attention head is highly positively correlated with local contacts, but highly negatively correlated with long range contacts. Lastly, we identify a total of 104 attention heads that are correlated (positively or negatively) with contacts at only one of the four ranges, suggesting that particular attention heads specialize in detecting certain types of contacts.

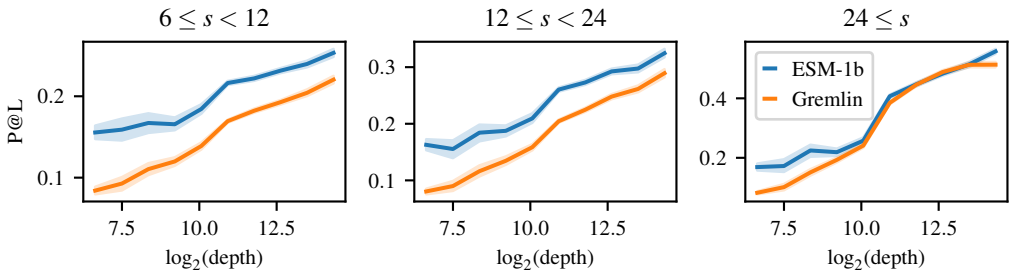


Figure 3: Gremlin (trRosetta) performance binned by MSA depth. For comparison, ESM-1b performance is also shown for the sequences in each bin.

4.5 Secondary Structure

In Section 4.4 we show that some heads that detect local contacts (which often correspond to secondary structure) are actually negatively correlated with long range contacts. We test ESM-1b's ability to detect secondary structure via attention by training a separate logistic regression on the Netsurf dataset (Klausen et al., 2019). As with the logistic regression on contacts, we compute attentions and perform APC + symmetrization. To predict the secondary structure of amino acid i, we feed as input the couplings a_{mhij} for each layer m, for each head h, and for $j \in [i-5, i+5]$, for a total of 7260 input features. Using just 100 of the 8678 training proteins, we achieve 79.3% accuracy on 3-class secondary structure prediction on the CB513 test set (Cuff & Barton, 1999). Figure 9 shows the importance of each layer to predicting the three secondary structure classes. There are spikes in different layers for all three classes, indicating that particular heads within those layers are specializing in detecting specific classes of secondary structure.

4.6 Perplexity vs. Contact Precision

Figure 2 explores the relationship between performance on the masked language modeling task (validataion perplexity) and contact prediction (Long Range P@L). A linear relationship exists between validation perplexity and contact precision for each model. Furthermore, for the same perplexity, the 12-layer ESM-1 model achieves the same long range P@L as the 34 layer ESM-1 model, suggesting that perplexity is a good proxy task for contact prediction. ESM-1 and ESM-1b models are trained with different masking patterns, so their perplexities cannot be directly compared, although a linear relationship is clearly visible in both. ESM-1 and ESM-1b have a similar number of parameters; the key difference is in their hyperparameters and architecture. The models shown have converged in pre-training, with minimal decrease in perplexity (or increase in contact precision) in the later epochs. This provides clear evidence that both model scale and hyperparameters play a significant role in a model's ability to learn contacts.

4.7 Calibration, False Positives, and Robustness

One concern with large neural networks is that, while they may be accurate on average, they can also produce spurious results with high confidence. We investigate this possibility from several perspectives. First, we find that logistic regression probabilities are reasonably well calibrated estimators of true contact probability and can be used directly as a measure of the model's confidence (Figure 10a).

Second, we analyze the false positives that the model does predict. We find that these are very likely to be within a Manhattan distance of 1-4 of a true contact (Figure 10b). This suggests that false positives may arise due to the way a contact is defined (Cb-Cb distance within 8 angstroms), and could be marked as true contacts under a different definition (Zheng & Grigoryan, 2017). Further, when we explore an example where the model's predictions are not near a true contact, we see that the example in question is a homodimer, and that the model is picking up on inter-chain interactions (Figure 11). While these do not determine the structure of the monomer, they are important for its function (Anishchenko et al., 2017).

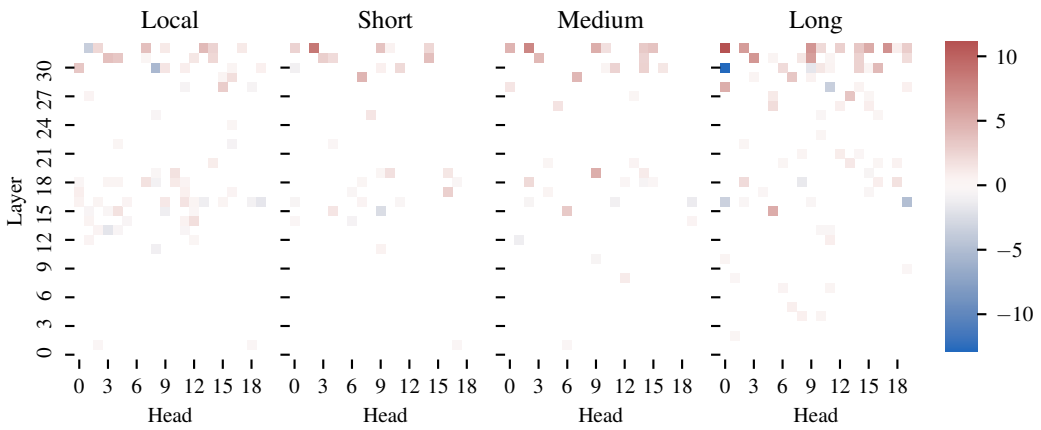


Figure 4: Logistic regression weights trained only on contacts in specific ranges: local [3, 6), short range [6, 12), medium range [12, 24), long range [24, ∞).

Third, we test the robustness of the model to insertions by inserting consecutive alanines at the beginning, middle, or end of 1000 randomly chosen sequences. We find that ESM-1b can tolerate up to 256 insertions at the beginning or end of the sequence and up to 64 insertions in the middle of the sequence before performance starts to significantly degrade. This suggests that ESM-1b learns a robust implicit alignment of the protein sequence. See Section A.10 for more details.

4.8 MSA Generation

Wang & Cho (2019) note that Transformers trained with the MLM objective can be used generatively. Here, we consider whether generations from ESM-1b preserve contact information for a given protein. The ability to generate sequences that preserve this information is a necessary condition for generation of biologically active proteins. We perform this evaluation by taking an input protein, masking out several positions, and re-predicting them. This process is repeated 10000 times to generate a pseudo-MSA for the input sequence (see Algorithm 1). We feed the resulting MSA into Gremlin to predict contacts. Over all sequences from our test set, this procedure results in a long range contact P@L of 14.5. Figure 14 shows one example where the procedure works well, with Gremlin on the pseudo-MSA having long range P@L of 52.2. For comparison, the standard ESM-1b pipeline achieves a contact precision of 76.7 on this example.

5 Discussion

Transformer protein language models trained with an unsupervised objective learn the tertiary structure of a protein sequence in their attention maps. Residue-residue contacts can be extracted from the attention by sparse logistic regression. Attention heads are found that specialize in different types of contacts. An ablation analysis confirms that the contacts are learned in the unsupervised phase, and that the logistic regression is only necessary to extract the part of the model that represents the contacts.

These results have implications for protein structure determination and design. The initial studies proposing Transformers for protein language modeling showed that representation learning could be used to derive state-of-the-art features across a variety of tasks, but were not able to show a benefit in the fully end-to-end setting (Rives et al., 2019; Rao et al., 2019; Elnaggar et al., 2020). For the first time, we show that protein language models can outperform state-of-the-art unsupervised structure learning methods that have been intensively researched and optimized over decades.

Finally, we establish a link between language modeling perplexity and unsupervised structure learning. A similar scaling law has been observed previously for supervised secondary structure prediction (Rives et al., 2019), and parallels observations in the NLP community (Kaplan et al., 2020; Brown et al., 2020). Evidence of scaling laws for protein language modeling support future promise as models and data continue to grow.

REFERENCES

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *Nature Methods*, 12:1315–1322, 3 2019. ISSN 15487105. doi: 10.1101/589333. URL https://www.biorxiv.org/content/10.1101/589333v1.

Ivan Anishchenko, Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. Origins of coevolution between residues distant in protein 3D structures. *Proceedings of the National Academy of Sciences of the United States of America*, 114 (34):9122–9127, 8 2017. ISSN 10916490. doi: 10.1073/pnas.1702664114. URL https://www.pnas.org/content/early/2017/08/03/1702664114https://www.pnas.org/content/early/2017/08/03/1702664114.abstract.

Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G. Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 4 2011. ISSN 08873585. doi: 10.1002/prot.22934. URL http://doi.wiley.com/10.1002/prot.22934.

Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-N protein engineering with data-efficient deep learning. *bioRxiv*, pp. 2020.01.23.917682, 2020. doi: 10.1101/2020.01.23.917682.

Tomasz Blazejewski, Hsing-I Ho, and Harris H Wang. Synthetic sequence entanglement augments stability and containment of genetic information in cells. *Science*, 365(6453):595–598, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

James A. Cuff and Geoffrey J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function and Genetics*, 34 (4):508–519, 3 1999. ISSN 08873585. doi: 10.1002/(SICI)1097-0134(19990301)34:4⟨508::AID-PROT10⟩3.0.CO;2-4. URL https://pubmed.ncbi.nlm.nih.gov/10081963/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 6 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL http://arxiv.org/abs/1810.04805.

S. D. Dunn, L. M. Wahl, and G. B. Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2 2008. ISSN 13674803. doi: 10.1093/bioinformatics/btm604.

Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. *Phys. Rev. E*, 87:012707, Jan 2013. doi: 10.1103/PhysRevE.87.012707. URL https://link.aps.org/doi/10.1103/PhysRevE.87.012707.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. 7 2020. URL http://arxiv.org/abs/2007.06225.

Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling the language of life – Deep Learning Protein Sequences. *bioRxiv*, pp. 614313, 2019. doi: 10.1101/614313. URL https://www.biorxiv.org/content/10.1101/614313v3.

L. Steven Johnson, Sean R. Eddy, and Elon Portugaly. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11(1):431, 8 2010. ISSN 14712105. doi: 10.1186/1471-2105-11-431. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-431.

David T. Jones, Daniel W. A. Buchan, Domenico Cozzetto, and Massimiliano Pontil. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 11 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr638. URL https://doi.org/10.1093/bioinformatics/btr638.

Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1314045110. URL https://www.pnas.org/content/110/39/15674.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjærgaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, and Paolo Marcatili. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 6 2019. ISSN 0887-3585. doi: 10.1002/prot.25674. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25674.

Tomasz Kosciolek and David T. Jones. De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PLOS ONE*, 9(3):1–15, 03 2014. doi: 10.1371/journal.pone.0092197. URL https://doi.org/10.1371/journal.pone.0092197.

Alan S. Lapedes, Bertrand G. Giraud, LonChang Liu, and Gary D. Stormoo. Correlated mutations in models of protein sequences: Phylogenetic and structural effects. *Lecture Notes-Monograph Series*, 33:236–256, 1999. ISSN 07492170. URL http://www.jstor.org/stable/4356049.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 7 2019. URL http://arxiv.org/abs/1907.11692.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language Modeling for Protein Generation. 3 2020. URL http://arxiv.org/abs/2004.03497.

Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PLOS ONE*, 6(12):1–20, 12 2011. doi: 10.1371/journal.pone.0028766. URL https://doi.org/10.1371/journal.pone.0028766.

Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011. ISSN 0027-8424. doi: 10.1073/pnas.1111471108. URL https://www.pnas.org/content/108/49/E1293.

Sergey Ovchinnikov, David E. Kim, Ray Yu-Ruei Wang, Yuan Liu, Frank DiMaio, and David Baker. Improved de novo structure prediction in casp11 by incorporating coevolution information into rosetta. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):67–75, 2016. doi: 10.1002/prot.24974. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.24974.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas,

Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. ISSN 1533-7928. URL http://scikit-learn.sourceforge.net.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating Protein Transfer Learning with TAPE. In *Neural Information Processing Systems*. Cold Spring Harbor Laboratory, 6 2019. doi: 10.1101/676825. URL https://doi.org/10.1101/676825http://arxiv.org/abs/1906.08230.

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, 2 2012. ISSN 1548-7091. doi: 10.1038/nmeth.1818. URL http://www.nature.com/articles/nmeth.1818.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, 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*, pp. 622803, 4 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v3.

William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W.R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 1 2020. ISSN 14764687. doi: 10.1038/s41586-019-1923-7. URL https://doi.org/10.1038/s41586-019-1923-7.

J. I. Sulkowska, F. Morcos, M. Weigt, T. Hwa, and J. N. Onuchic. Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, 109(26):10340–10345, 2012. doi: 10.1073/PNAS.1207864109. URL http://www.pnas.org/cgi/doi/10.1073/pnas.1207864109.

Baris E. Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H. Wu. UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288, 5 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm098. URL http://www.uniprot.org. UniRef50 database licensed under (CC BY 4.0).

Todd J. Taylor, Hongjun Bai, Chin Hsien Tai, and Byungkook Lee. Assessment of CASP10 contact-assisted predictions. *Proteins: Structure, Function and Bioinformatics*, 82(SUPPL.2):84–97, 2 2014. ISSN 08873585. doi: 10.1002/prot.24367. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6961783/.

Pengfei Tian, John M Louis, James L Baber, Annie Aniana, and Robert B Best. Co-evolutionary fitness landscapes for sequence design. *Angewandte Chemie International Edition*, 57(20):5674–5678, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. BERTology Meets Biology: Interpreting Attention in Protein Language Models. *bioRxiv*, pp. 2020.06.26.174417, 6 2020. doi: 10.1101/2020.06.26.174417. URL http://arxiv.org/abs/2006.15222.

Alex Wang and Kyunghyun Cho. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. 2 2019. URL http://arxiv.org/abs/1902.04094.

Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):1–34, 01 2017. doi: 10.1371/journal.pcbi.1005324. URL https://doi.org/10.1371/journal.pcbi.1005324.

Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, David Baker, and John Harvard. Improved protein structure prediction using predicted inter-residue orientations. *bioRxiv*, pp. 846279, 2019. doi: 10.1101/846279. URL https://www.biorxiv.org/content/10.1101/846279v1.

Kevin K. Yang, Zachary Wu, Claire N. Bedbrook, and Frances H. Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 8 2018. ISSN 14602059. doi: 10.1093/bioinformatics/bty178.

Fan Zheng and Gevorg Grigoryan. Sequence statistics of tertiary structural motifs reflect protein stability. *PLoS ONE*, 12(5):e0178272, 5 2017. ISSN 19326203. doi: 10.1371/journal.pone.0178272. URL https://doi.org/10.1371/journal.pone.0178272.

A APPENDIX

A.1 NOTATION

In the figures, we report contact precision in the range of 0.0 to 1.0. In the text and in the tables, we report contact precision in terms of percentages, in the range of 0 to 100.

A.2 AVERAGE PRODUCT CORRECTION (APC)

In protein contact prediction, APC is commonly used to correct for background effects of entropy and phylogeny (Dunn et al., 2008). Given an $L \times L$ coupling matrix F, APC is defined as

$$F_{ij}^{\text{APC}} = F_{ij} - \frac{F_i F_j}{F} \tag{3}$$

Where F_i, F_j, and F are the sum over the i-th row, j-th column, and the full matrix respectively. We apply APC independently to the symmetrized attention maps of each head in the Transformer. These corrected attention maps are passed in as input to a logistic regression.

A.3 STRUCTURE PREDICTION FROM CONTACTS

Given a contact map, several approaches have been described for inferring the all-atom 3D structure of the protein. For example, EVFold (Marks et al., 2011) and DCAFold (Sulkowska et al., 2012) predict co-evolving couplings by using a Potts Model and then generate 3D conformations by directly folding an initial conformation with simulated annealing, using the predicted residue-residue contacts as constraints. Similarly, FragFold (Kosciolek & Jones, 2014) and Rosetta (Ovchinnikov et al., 2016) incorporate constraints from a Potts Model into a fragment assembly based pipeline. While Gremlin (Balakrishnan et al., 2011) uses a pseudolikelihood approach to training the Potts Model (Lapedes et al., 1999), other methods can be used as well, for example inverse covariance (Morcos et al., 2011) or sparse inverse covariance (Jones et al., 2011).

A.4 GREMLIN IMPLEMENTATION DETAILS

Gremlin is trained by optimizing the pseudolikelihood of W and V, which correspond to pairwise and individual amino acid propensities. The pseudolikelihood approximation models the conditional distributions of the original joint distribution and can be written:

$$\log p(x_i^d = a | x_{j \neq i}^d; W_i, V_i) = \log \frac{\exp\left(V_{ia} + \sum_{j=1, j\neq i}^{N} \sum_{b=1}^{20} \mathbb{1}(x_j^d = b) W_{ijab}\right)}{\sum_{c=1}^{20} \exp\left(V_{ic} + \sum_{j=1, j\neq i}^{N} \sum_{b=1}^{20} \mathbb{1}(x_j^d = b) W_{ijcb}\right)} \tag{4}$$

subject to the constraint that $W_{ii} = 0$ for all i, and that W_{ijab} is symmetric in both sequence (i, j) and amino acid (a, b). Additionally, Gremlin uses a regularization parameter that is adjusted based on the depth of the MSA.

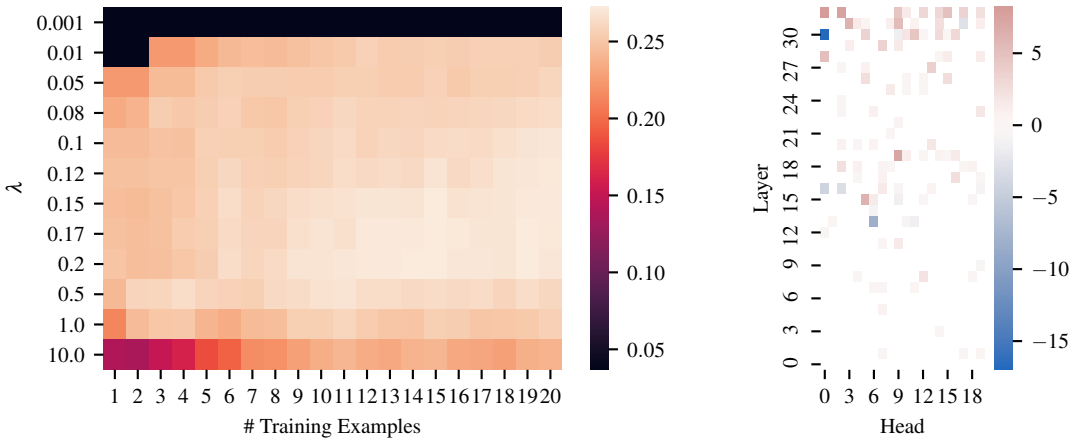
A.5 ESM-1 IMPLEMENTATION DETAILS

The original ESM-1 models were described in (Rives et al., 2019). Recently, the authors have made ESM-1b available, which is a robustly optimized version of the largest ESM-1 model. ESM-1b is the result of an extensive hyperparameter sweep that was performed on smaller 12 layer models. ESM-1b is the result of scaling up that model to 33 layers.

Compared to ESM-1, the main changes in ESM-1b are: higher learning rate; dropout after word embedding; learned positional embeddings; final layer norm before the output; and tied input/output word embeddings. The weights of all ESM models throughout the training process were provided to us by the authors.

A.6 LOGISTIC REGRESSION DETAILS

Given a model with M layers, H heads, and an input sequence x of length L, let A_{mh} be the $L \times L$ contact map from the h-th head in the m-th layer. We first symmetrize this map and apply APC and



(a) Gridsearch (b) Weights

Figure 5: (a) Gridsearch on logistic regression over number of training examples and number regularization penalty. Values shown are long range P@L over a validation set of 20 proteins. (b) Per-head and layer weights of the logistic regression on the best ESM-1b model.

.

let a_{mhij} be the coupling weight between sequence position i and j in the resulting map. Then we define the probability of a contact between positions i and j according to a logistic regression with parameters β:

$$p(c_{ij}^d; \beta) = \frac{1}{1 + \exp\left(-\beta_0 - \sum_{m=1}^{M} \sum_{h=1}^{H} \beta_{mh} a_{mhij}^d \right)} \tag{5}$$

To fit β, let $\mathcal{D}$ be a set of training proteins, k be a minimum sequence separation, and λ be a regularization weight. The objective can then be defined as follows:

$$\mathcal{L}(\mathcal{D}; \beta) = \prod_{d \in \mathcal{D}} \prod_{i=1}^{L_d - k} \prod_{j=i+k}^{L_d} p(c_{ij}^d; \beta) \tag{6}$$

$$\hat{\beta} = \max_{\beta} \mathcal{L}(\mathcal{D}; \beta) + \frac{1}{\lambda} \sum_{m=1}^{M} \sum_{h=1}^{H} |\beta_{mh}| \tag{7}$$

We fit the parameters β via scikit-learn (Pedregosa et al., 2011) and do not backpropagate the gradients through the attention weights. In total, our model learns $MH + 1$ parameters, many of which are zero thanks to the L_1 regularization.

There are three hyperparameters in our training setup: the number of proteins in our training set $\mathcal{D}$, the regularization parameter λ, and the minimum sequence separation of training contacts k. We find that performance improves significantly when increasing the $\mathcal{D}$ from 1 protein to 10 proteins, but that the performance gains drop off when $\mathcal{D}$ increases from 10 to 20 (Figure 1). Through a hyperparameter sweep, we determined that the optimal λ is 0.15. We find that ignoring local contacts ($|i-j| < 6$) is also helpful. Therefore, unless otherwise specified, all logistic regressions are trained with $|\mathcal{D}| = 20, \lambda = 0.15, k = 6$. See Figure 5a for a gridsearch over the number of training proteins and regression penalty. We used 20 training proteins and 20 validation proteins for this gridsearch. Figure 5b shows the weights of the final logistic regression used for ESM-1b.

A.7 Performance Distribution

Figure 6 shows the full distribution of performance of ESM-1b compared with Gremlin. When we provide Gremlin access to Uniref100, along with metagenomic sequences, ESM-1b still consistenly

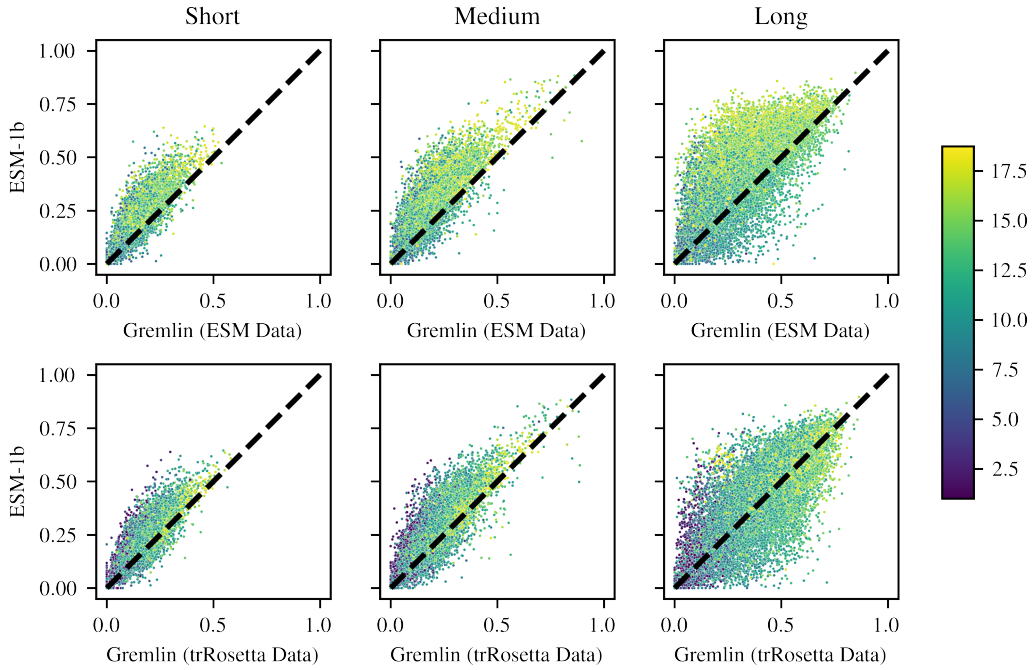


Figure 6: Short, medium, and long range P@L performance distribution of ESM-1b vs. Gremlin. Each point is colored by the $\log_2$ of the number of sequences in the MSA.

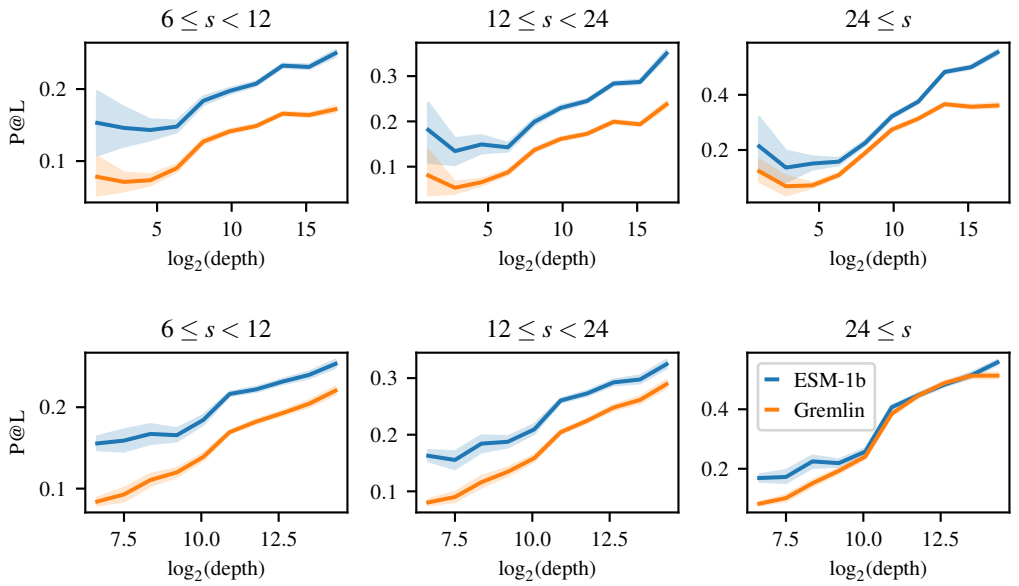


Figure 7: Gremlin performance binned by MSA depth using both ESM (top) and trRosetta (bottom) MSAs. For comparison, ESM-1b performance is also shown for the sequences in each bin.

outperforms Gremlin when extracting short and medium range contacts. For long range contacts, Gremlin is much more comparable, and has higher contact precision on 47% of sequences. With access to the same set of sequences, ESM-1b consistently outperforms Gremlin in detecting short, medium, and long range contacts. This suggests that ESM-1b can much better extract information from the same set of sequences and suggests that further scaling of training data may improve ESM-1b even further.

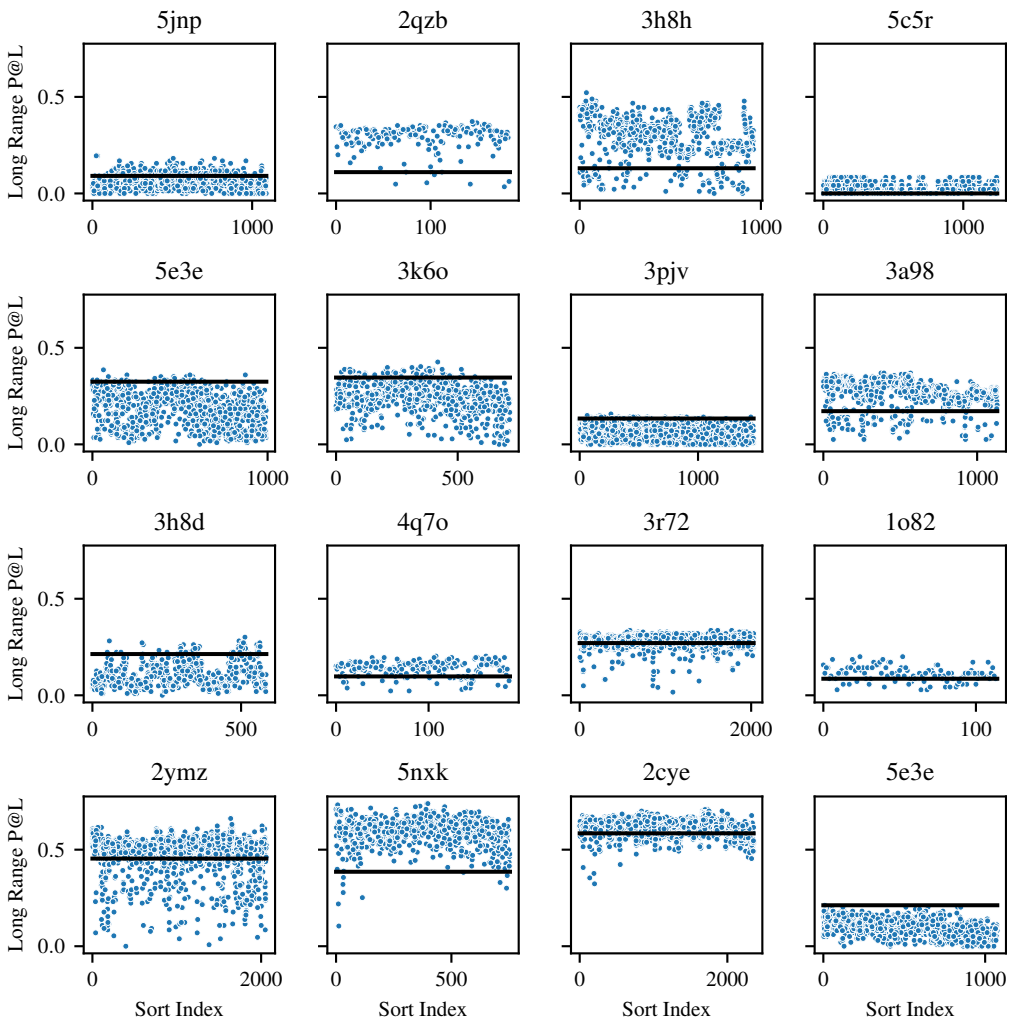


Figure 8: Distribution of contact perplexity when evaluating different sequences from the same MSA. The x-axis shows the index of each sequence, sorted in ascending order by hamming distance from the query sequence (query sequence is always index 0). The y-axis shows long range P@L. The black line indicates Gremlin performance on that MSA.

.

This analysis is further borne out in Figure 7. Given the same set of sequences, ESM-1b outperforms Gremlin on average for short, medium, and long-range contacts regardless of the depth of the MSA generated from the ESM-1b training set.

Additionally, we find that ESM-1b can provide varying contact maps for different sequences in the MSA (Figure 8). This is not possible for Gremlin, which is a family-level model. We leverage this in a fairly simple way to provide a modest boost to the contact precision of ESM-1b (Section 4.2).

A.8 Secondary Structure Weights

Figure 9 shows importance of each transformer layer to predicting each of the three secondary structure classes. We see that, as with contact prediction, the most important layers are in the middle layers (14-20) and the final layers (29-33). Some layers spike more heavily on particular contact classes (e.g. layer 33 is important for all classes, but particularly important for β-strand prediction). This suggests that particular heads within these layers activate specifically for certain types of secondary structure.

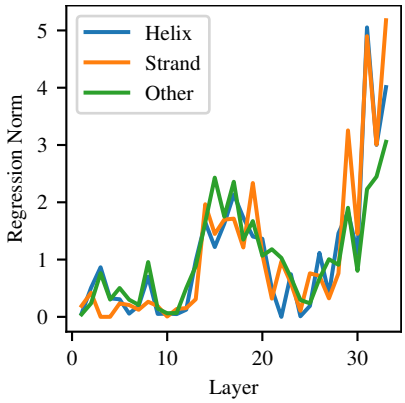
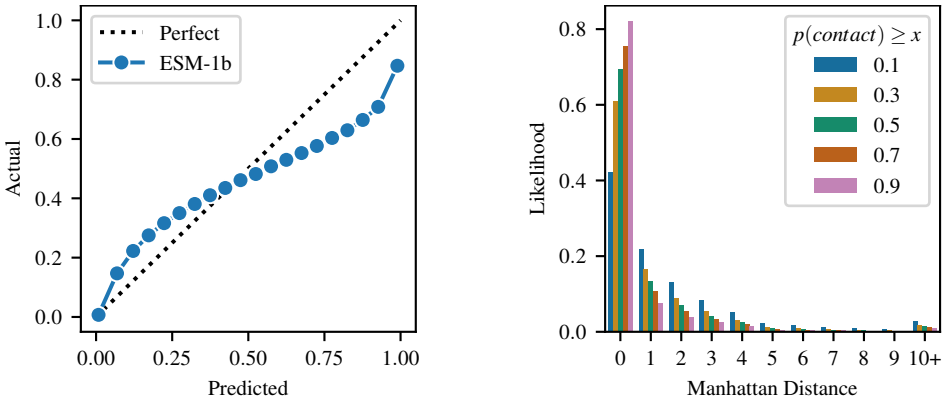


Figure 9: L_2 norm of weights for 3-class secondary structure prediction by Transformer layer.



(a) Predicted probability calibration

(b) Manhattan distance to nearest contact.

Figure 10: (a) Calibrated probability of a real contact given predicted probability of contact over all test proteins. (b) Distribution of Manhattan distance between the coordinates of predicted contacts and the nearest true contact at various thresholds of minimum $p(contact)$. A distance of zero corresponds to a true contact.

A.9 Model Calibration and False Positives

Vig et al. (2020) showed that the attention probability from the TAPE Transformer was a well-calibrated estimator for the probability of a contact. In Figure 10a we examine the same with the logistic regression trained on the ESM-1b model. We note that ESM-1b, in addition to being more accurate overall than Gremlin, also provides actual probabilities. We find that the ESM-1b model probabilities are not perfectly calibrated, although the predicted probabilities are still indicative of actual contact likelihood. It is therefore reasonable to use the logistic regression probability as a measure of the model's confidence.

In the case of false positive contacts we attempt to measure the Manhattan distance between the coordinates of predicted contacts and the nearest true contact (Figure 10b). We observe that the Manhattan distance between the coordinates of false positive contacts are often very close (Manhattan distance between 1-4) to real contacts, and that very few false positives have a Manhattan distance $\geq$ 10 from a true contact. Figure 11 shows one example where the model does appear to hallucinate contacts around residues 215 and 415, which do not appear in the contact map for this protein. However, this protein is a homodimer and these contacts are present in the inter-chain contact map. This suggests that some 'highly incorrect' false positives may instead be picking up on inter-chain contacts.

One hypothesis as to the benefit of large language models as opposed to simpler Potts models is that they may be able to learn an implicit alignment due to their learned positional embedding. For a Potts

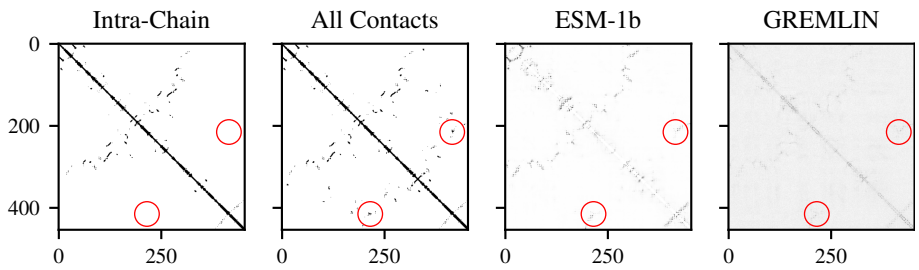


Figure 11: Intra-chain, inter-chain, and predicted contacts for 5mlt, which is a homodimer.

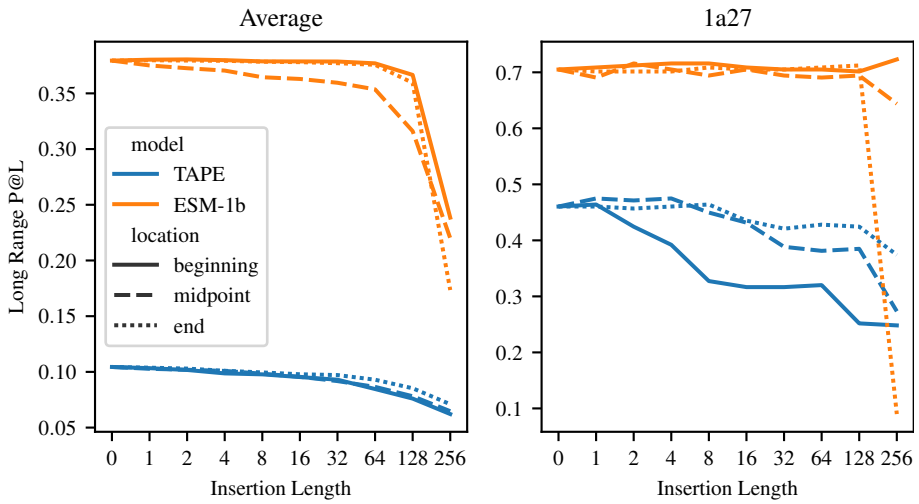


Figure 12: Robustness of ESM-1b and TAPE models to insertions of Alanine at the beginning, middle, and end of sequence

.

Model, an alignment enables a model to relate positions in the sequence given evolutionary context despite the presence of insertions or deletions. We test the robustness of the model to insertions by inserting consecutive alanines at the beginning, middle, or end of 1000 randomly chosen sequences with initial sequence length < 512 (we limit initial sequence length in order to avoid out-of-memory issues after insertion). We find that ESM-1b can tolerate up to 256 insertions at the beginning or end of the sequence and up to 64 insertions in the middle of the sequence before performance starts to significantly degrade. This suggests that ESM-1b learns a robust implicit alignment of the protein sequence.

A.10 Alignment

On the other hand, we find that the TAPE Transformer is less robust to insertions. On one sequence (pdbid: 1a27), we find the TAPE Transformer drops in precision by 12 percentage points after adding just 8 alanines to the beginning of the sequence, while ESM-1b sees minimal degradation until 256 alanines are inserted. We hypothesize that, because TAPE was trained on protein domains, it did not learn to deal with mis-alignments in the input sequence.

A.11 Evolutionary Finetuning Details

We finetuned each model using a learning rate of 1e-4, 16k warmup updates, an inverse square root learning rate schedule, and a maximum of 30 epochs. This resulted in a varying number of total updates depending on the size of the MSA, with larger MSAs being allowed to train for more updates. This should ideally help prevent the model from overfitting too quickly on very small MSAs. We use a variable batch size based on the length of the input proteins, fixing a maximum of 16384 tokens per batch (so for a length 300 protein this would correspond to a batch size of 54). We

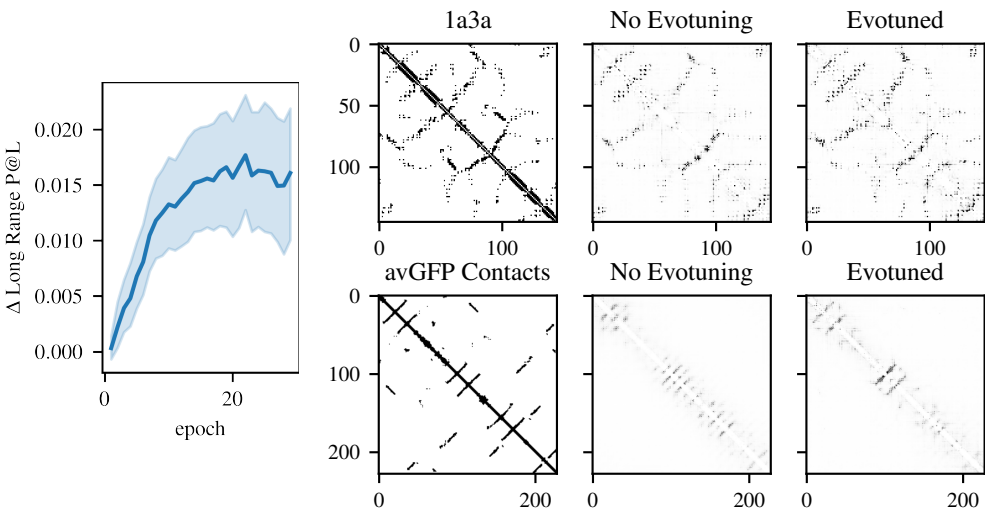


Figure 13: Left: Average change in contact precision vs. number of finetuning epochs over 380 proteins. Right: Real and predicted contacts before and after evolutionary finetuning for 1a3a and avGFP. For 1a3a, long range P@L improves from 54.5 to 61.4. For avGFP, long range P@L improves from 7.9 to 11.4.

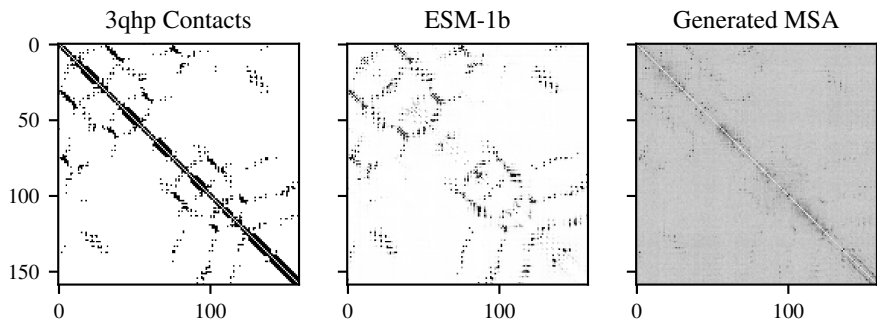


Figure 14: Contacts for 3qhp from Gremlin trained on pseudo-MSA generated by ESM-1b, compared to real and ESM-1b predicted contacts. The generated MSA achieves a long-range P@L of 52.2 while the attention maps achieve a precision of 76.7.

use MSAs from trRosetta for finetuning all proteins with the exception of avGFP, where we use the same set of sequences from Alley et al. (2019).

A.12 MSA Generation

Result: Generated MSA
input `// protein sequence`
curr = input `// optionally, the input can be repeated for batching`
for $0 \leq i < 10000$ **do**
  masked = mask 20% of positions in curr;
  pred = model(masked);
  curr[masked positions] = pred[masked positions];
  MSA.append(curr);
  if $random() < 0.1$ **then**
    curr = input;
  end
end

Algorithm 1: Quickly generate a pseudo-MSA from an input sequence.

Algorithm 1 presents the algorithm used to generate pseudo-MSAs from ESM-1b. Each pseudo-MSA is passed to GREMLIN in order to evaluate the preservation of contact information (Figure 14).