We thank the reviewers for their positive feedback and helpful suggestions for improvement.

Response to Reviewer 1 Both Bepler et al. [11] and Alley et al. [12] were pretrained on the PFAM corpus. We have updated our table to make this clearer. We have expanded the discussion of alignment-based features in the Background section to make it clearer what they are. Since alignment-based features are fed directly into the downstream task models, this is separate from pretraining on the language modeling tasks. The reviewer points out the unexpected result that pretraining decreased contact prediction performance with the ResNet. This was caused by an issue with our training/evaluation setup for contact map prediction, which we discovered after the submission deadline. When we re-ran training/evaluation for all models, we found that pretraining improved accuracy across the board, as shown in Table 1.

Method	Contact
Transformer (No PT)	0.32
LSTM (No PT)	0.19
ResNet (No PT)	0.20
Transformer (PT)	0.36
LSTM (PT)	0.39
ResNet (PT)	0.29
Bepler (PT)	0.40
UniRep (PT)	0.34

Table 1: Updated Contact Results

Response to Reviewer 2 When splitting PFAM, we split at both the family and clan level. We will present these results separately in Table 1 of the paper. We ran BLAST to obtain sequence identity between these sets. We find that that on average the maximum sequence identity between the training set and proteins in the random-split set is 73%, in the family-holdout is 37%, and in the clan-holdout is 22%. Hamming distance is measured at the amino acid level, which we chose because we are trying to highlight the protein engineering setting where the goal is to accurately predict the function of an input molecule specified by a protein designer. We have added this reasoning to the task description.

The reviewer is correct that the LSTM language model is not making a Markovian assumption, and we have removed this line. The reviewer is correct in their characterization of the LSTM, which is the concatenation of the two vectors of dimension 1024. For the single-prediction tasks, we use a learned attention-weighted sum over the amino-acid representations. These points are in the Appendix, and we have added pointers to these details in the main text.

We agree that difference in representation size could make a difference on downstream tasks. Due to cost of pretraining large models, we cannot perform a grid search over number of parameters and representation size for pretrained models. As such, we instead chose to base models on common hyperparameter choices in the literature. In addition, controlling for number of parameters and representation size proved difficult without resorting to extremely odd choices or significant alterations to the network design. For example, lowering the LSTM hidden dimension to 512 requires 10 layers to match the parameters in the Transformer. To address these concerns, we are pretraining three smaller models with a fixed representation size to compare architectures with representation size held constant (see Table 2).

Response to Reviewer 3 We agree with the reviewer that the language models we trained are quite large, which may present a problem for fine-tuning on small datasets. Our tasks are of similar size to most of those in the GLUE benchmark, which has been instrumental in demonstrating the success of self-supervision in NLP. Since the models that were applied to GLUE have tens to hundreds of millions of parameters, we chose to make our models roughly the same size ($\sim$40M parameters). We also believe that showing results of larger models is important to counter the claim that simply scaling these models would allow them to outperform gold-standard features from bioinformatics. We have added this discussion in the manuscript. To further address questions about size, we are pretraining smaller language models with $\sim$3M parameters for a week on Pfam. Some preliminary results on the Transformer and LSTM after two days of pretraining are reported in Table 2.

Method	SS	Stab	Fluor	RH
LSTM	0.73	0.67	0.66	0.18
Transformer	0.70	0.68	0.68	0.13
ResNet	0.73	0.68	0.43	0.11

Table 2: Small Language Model Results

We agree with the reviewer that usability of the benchmark is of paramount concern. The larger models require around 11 GB of memory on a GPU to train, which is on the higher end. To alleviate this, we will upload weights for the smaller models once they are trained.

In Table 2 of the paper we chose to report simple global metrics. However, we report many detailed metrics in the appendix, including some of the metrics the reviewer suggested: precision-recall, AUPRC, and accuracy for long-range contacts. For the other tasks we provided label/task-specific breakdowns in the appendix, and have added more pointers to the appendix to improve the visibility of these results. We have also revised the discussion in the appendix for ease of reading. We also plan to create an online leaderboard, where these other metrics will be more prominent. We agree with the reviewer that CB513 is built from an old dataset and will look into improving the test set for future iterations of TAPE.

Clarity We thank all reviewers for positive comments on clarity. All small edits and suggested improvements have been incorporated. In particular we are grateful for Reviewer 2's catch of the missing beta strand contacts (the contact map was cropped incorrectly, and has been updated). Additionally, as per Reviewer 3's suggestion, we have changed 'Protein Modeling' to 'Machine learning applied to protein sequences' in the abstract and throughout the manuscript.