Parameter-Efficient Fine-Tuning of Protein Language Models Improves Prediction of Protein-Protein Interactions

Samuel Sledzieski [*]
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
samsl@mit.edu

Meghana Kshirsagar
AI for Good Research Lab
Microsoft Corporation
Redmond, WA 98052, USA
meghana.kshirsagar@microsoft.com

Bonnie Berger
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
bab@mit.edu

Rahul Dodhia
AI for Good Research Lab
Microsoft Corporation
Redmond, WA 98052, USA
rahul.dodhia@microsoft.com

Juan Lavista Ferres [†]
AI for Good Research Lab
Microsoft Corporation
Redmond, WA 98052, USA
jlavista@microsoft.com

Abstract

Mirroring the massive increase in the size of transformer-based models in natural language processing, proteomics too has seen the advent of increasingly large foundational protein language models. As model size increases, the computational and memory footprint of fine-tuning expands out of reach of many academic labs and small biotechs. In this work, we apply parameter-efficient fine-tuning (PEFT) to protein language models to predict protein-protein interactions. We show that a model trained with the PEFT method LoRA outperforms full fine-tuning while requiring a reduced memory footprint. We also perform an analysis of which weight matrices in the attention layers to adapt, finding that contrary to in natural language processing, modifying the key and value matrices yields the best performance. This work demonstrates that despite the recent increase in scale, the effective use of protein language models for representation learning is not out of the reach of research groups with fewer computational resources.

1 Introduction

The field of natural language processing has recently seen an explosion of growth in transformer-based language models [Vaswani et al., 2017], enabling massive advances in text generation, sentiment analysis, and other language understanding tasks [**?**]. Language models excel because natural language conforms to the distributional hypothesis: that the function of a word, or token, is dependent

[*] Work partially performed while an intern with AI for Good Lab
[†] Corresponding author

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

on the context in which it appears. Likewise, protein sequences are order- and context- dependent, and proteomics has thus seen a parallel rise in the use of language models, which has transformed the modeling of protein sequence, structure, and function. Typically, when a large language model is tailored to a specific downstream task, the parameters of the pre-trained model are updated in a process known as fine-tuning. However, as the size of foundation models increases, fine-tuning a model for a task of interest is increasingly computationally expensive. This has motivated the development of *parameter-efficient* fine-tuning methods (PEFT), which aim to match fine-tuning performance with only a very small percentage ($< 1\%$) of tunable parameters.

In this work, we explore the application of PEFT to protein language models, comparing parameter efficient tuning to full fine-tuning, as well as using the learned representations unchanged, training only a classification head. Specifically, we look at the task of predicting protein-protein interaction from sequence, using protein language models to generate learned representations which act as input features. To hone in on the impact of language model featurization and fine-tuning, we eschew more complex learning approaches and train a simple classification head. We find that with proper featurization and training, these approaches are competitive with the current state-of-the-art for sequence-based PPI prediction. In addition, we find that using the frozen representations alone is a viable alternative to full- or parameter-efficient fine-tuning on the PPI prediction task—especially when considering compute and memory efficiency.

2 Methods

2.1 Protein Language Model

We focused our efforts on ESM2, a transformer-based protein language model which is presently considered the state-of-the-art in protein language modeling [Lin et al., 2023]. ESM2 has several different model sizes, ranging from eight million to 15 billion parameters. For this study, we focused on the 650 million parameter version of ESM2 (Section S1).

For an amino acid sequence $X = x_1x_2...x_n$, a PLM of dimension d returns a set of embeddings $E \in \mathbb{R}^{d\times n} = e_1e_2...e_n, e_i \in \mathbb{R}^d$. To standardize the size of representations for sequences of dynamic length, a pooling step needs to be undertaken. This is most commonly done either by averaging along the length of the sequences ($e_p \in \mathbb{R}^d = \frac{1}{n}\sum_{i=1}^{n} e_i$) or by selecting the first token of the sequence, a non-amino acid token (`[clsf]`) created specifically for sequence classification. Here, we chose to take the former approach as it explicitly integrates signal across the length of the protein. We note that while this is a commonly used approach, how to best aggregate sequence-length representations into a fixed dimension embedding is an open problem in language modeling. Converting this pooled embedding into a binary ($Y \in \mathbb{R}^2$) or multi-class ($Y \in \mathbb{R}^{18}$) prediction requires an additional classification head. Then, fixed-length embeddings were averaged before being passed to the classification head, as in Szymborski and Emad [2022]. In this study, we tested two different classification heads. The first, applied directly to the pre-trained E without fine-tuning, is a simple multi-layer perceptron (MLP), with the number and size of layers determined by grid search (Section 2.3, Section S2). The second is the `ESMClassificationHead` made available by the authors in the public HuggingFace repository, which consists of two dense layers with dropout and a tanh activation between the layers. We selected classification heads which were demonstrated to yield strong performance in previous work in order to minimize the need for hyper-parameter search in this space.

2.2 Parameter-Efficient Adaptation

Here, we chose to use LoRA [Hu et al., 2021], one of the most widely-adopted parameter-efficient fine-tuning methods. LoRA adds two low-rank matrices A and B to each adapted weight matrix. Given weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA adds new parameters $A \in \mathbb{R}^{r\times k}, B \in \mathbb{R}^{d\times r}, r << d, k$. The normal forward pass of the layer given input $x \in \mathbb{R}^k$ is $h = Wx$, and the forward pass with the LoRA adaptation is $h = Wx + BAx$. Only the weights of A, B are updated during back-propagation, while the weights of W are frozen. BAx is scaled by the quantity $\frac{\alpha}{r}$, where α is a hyper-parameter which is held constant in the original report. While A is initialized with a random Gaussian distribution, by initializing $B = 0$ the first forward pass of the model is equivalent to the pre-trained model without adaptation. Following the recommendations of the original paper, we initially apply LoRA only to the query and value matrices of the attention head.

2.3 Training and Implementation

All fine-tuning and PEFT models were implemented in PyTorch, using the HuggingFace implementations of ESM2 from the `transformers` package and LoRA from the `peft` package. Models were trained on NVIDIA V100 GPUs with 32GB of memory. We used the binary cross-entropy with logits loss to compute error, with an L2 weight decay of 0.01. Model weights were optimized via back-propagation using the Adam optimizer and a cosine decay with restarts learning rate schedule (initial learning rate 0.001). Models were trained with an epoch size of $16,384$, with an on-device batch size of 4 and gradient accumulation every 16 steps, for an effective batch size of 64. Models were trained for 40 epochs and the best model based on validation AUPR was chosen for testing. Except for where otherwise specified, for PEFT models we used LoRA hyperparameter values of $r = 8$, $\alpha = 32$, dropout $p = 0.1$, and an L2 coefficient of 0.01.

The parameters for the MLP on frozen embeddings were chosen by a grid search (implemented in scikit-learn, Section S2). The best performing model had two hidden layers with sizes $(64, 64)$ and ReLU activations, and were optimized with the Adam optimizer for 2000 iterations with a tolerance of 0.0001 and an adaptive learning rate initialized at 0.01.

2.4 Benchmark Data

While creating train/test splits based on filtering homologous proteins is common in machine learning for proteomics, the binary nature of PPI prediction presents a unique challenge because data leakage can still occur if only one protein of an interacting pair appears in both sets. If a so called "hub" protein with many interactions appears in both the training and test set, models can learn that this specific protein is likely to have positive interactions. Then, test set performance will be inflated even if nothing is learned about the actual pairwise interactions. After noting pervasive biases in previous benchmarks relating to sequence similarity and node degree, Bernett et al. [2023] introduced a new gold standard data set for benchmarking PPI. The splits introduced in this benchmark apply a more stringent notion of sequence similarity for pairwise problems as introduced by Park and Marcotte [2012], splitting by C3 similarity. In addition, both the positive and negative data sets are balanced with regard to node degree; as a consequence models cannot learn that proteins *in general* interact just because they are high degree. This data set consists of 163,192/59,246/52,035 training/validation/test edges, with an 1:1 ratio of positives to negatives.

3 Results

3.1 Reduced memory of PEFT enables deeper fine-tuning

The most common approach to fine-tuning is to unfreeze the weights of the last n transformer layers, which we compare with using LoRA to add tunable weights of the last n layers. Table 1 shows the maximum GPU memory used by fine-tuning (FT) or adaptation (PEFT) of different numbers of layers. Both FT and PEFT eventually overflow available GPU memory as the number of layers adapted increases; in the following section we show the results of fine-tuning 4 layers, and using LoRA to adapt 5 layers to demonstrate that using parameter-efficient fine-tuning allows for deeper adaptation. All experiments were performed on a GPU with 32GB of memory, with a batch size of 4 and maximum sequence length of 1024.

Table 1: **Comparing memory usage of PEFT.** Maximum memory usage in GB. *OOM* indicates that the run was killed due to running out of GPU memory. Parameter-efficient fine-tuning enables fine-tuning of deeper model layers.

	Tuning Method	**# Layers = 2**	**4**	**5**	**8**
PPI Prediction					
	PEFT	10.7	17.7	21.8	*OOM*
	FT	14.8	22.7	*OOM*	*OOM*

3.2 ESM2 embeddings enable state-of-the-art PPI prediction

We present results for an MLP classifier trained on frozen embeddings, a model trained with 5 layers of PEFT adaptation, and a model trained with 4 layers of full-parameter fine-tuning. In Table 2, we compare to the best prior scores from Bernett et al. [2023]—either Topsy-Turvy [Singh et al., 2022], or SVM-PCA, a baseline constructed by Bernett et al. [2023] which trains a support vector machine on PCA-reduced sequence similarity vectors. Note that for each benchmark metric, we selected the best score across all methods evaluated, and that no single method achieved the "Best Prior" performance across the board, so this is a significantly higher threshold than comparison to any single method. The PEFT model achieves performance better than the best prior, and is competitive with the frozen embeddings. While the frozen embedding method is faster if embeddings are pre-computed, it requires substantially more disk space and memory to store those embeddings, so if memory or disk space is limited using a PEFT model is preferable. Both result in better performance than full-parameter fine-tuning.

Table 2: **Applying PEFT to train models for protein-protein interaction.** We trained multiple variants of ESM2 to predict protein-protein interactions, and evaluate using the benchmark data sets from Bernett et al. [2023]. **MLP** indicates a multi-layer perceptron trained on embeddings from a frozen model, while **PEFT** and **FT** indicate parameter-efficient fine-tuning and traditional fine-tuning of the transformer layers. Due to the reduced memory footprint of PEFT, we were able to fine-tune an additional layer. For fine-tuning, validation performance rapidly fluctuates between extremely high recall/near zero specificity and the reverse (Supplementary Figure S2).

	Best Prior	**MLP**	**PEFT** (5 Layers)	**FT** (4 Layers)
# Trainable Params.	-	88,769	368,897	78,873,857
Validation				
Accuracy	-	0.595	**0.596**	0.521
F1	-	0.576	0.648	**0.669**
MCC	-	-	**0.201**	0.092
AUPR	-	**0.632**	0.620	0.599
Precision	-	**0.603**	0.574	0.511
Recall	-	0.552	0.742	**0.968**
Specificity	-	-	**0.450**	0.0733
Test				
Accuracy	0.56 (Topsy-Turvy)	**0.631**	0.608	0.508
F1	0.61 (SVM-PCA)	0.632	**0.666**	**0.666**
MCC	0.15 (Topsy-Turvy)	**0.261**	0.230	0.055
AUPR	-	**0.684**	0.600	0.577
Precision	**0.65** (Topsy-Turvy)	0.630	0.580	0.504
Recall	0.77 (SVM-PCA)	0.633	0.780	**0.984**
Specificity	**0.86** (Topsy-Turvy)	0.623	0.436	0.032

3.3 Which matrices should be adapted for protein modeling?

In their original manuscript, Hu et al. [2021] show that for natural language models, adapting only query and value weights (W_Q, W_V) of the attention heads yields the best tradeoff of performance and parameter efficiency. However, the space of natural language is not necessarily the same as that of protein sequence, and we wanted to evaluate to what extent the choice in adapted weight matrices affected performance. Table 3 shows that while performance is relatively robust across all values, adapting the key and value matrices results in the best overall performance—although adapting only the value matrix is also quite strong, and is more parameter efficient, so may be the best choice for parameter efficient fine-tuning of protein language models if memory constraints are especially tight.

Table 3: **Comparison of weight matrix adaptations.** Validation AUPR and test set metrics after applying LoRA to different combinations of weight matrices in the attention layer. Adapting only W_K and W_V gives the best performance, although adapting only W_V also performs well and is more parameter efficient.

Weight Matrix	**Val. AUPR**	**AUPR**	**Acc.**	**F1**	**MCC**	**Prec.**	**Rec.**	**Spec.**
W_Q	0.536	0.529	0.516	0.430	0.034	0.524	0.364	0.669
W_K	0.565	0.562	0.544	0.433	0.095	0.572	0.348	**0.739**
W_V	0.612	0.610	**0.605**	**0.650**	**0.216**	0.583	**0.735**	0.474
W_Q, W_K	0.590	0.576	0.564	0.582	0.129	0.559	0.607	0.521
W_Q, W_V	0.619	0.617	0.599	0.633	0.201	0.583	0.692	0.506
W_K, W_V	**0.637**	**0.633**	0.601	0.630	0.205	**0.587**	0.680	0.522
W_Q, W_K, W_V	0.628	0.613	0.603	0.639	0.210	0.585	0.704	0.502

4 Related Work

Language Modeling in Biology While the first protein language models (Bepler & Berger, UniRep) used recurrent neural networks like the bi-LSTM [Bepler and Berger, 2021, Alley et al., 2019], recent work has also converged around the transformer. Models like ProtBert, ProtT5, [Elnaggar et al., 2021], and ESM [Rives et al., 2021] train transformers on massive sets of protein sequence data in an unsupervised manner, learning meaningful representations which can be applied to replace manual feature engineering, or computationally expensive evolutionary searches and construction of multiple sequence alignments. Most recently, ESM2 [Lin et al., 2023] represents the largest protein language model to date, with models as large as 15 billion parameters. While this is still shy of the largest natural language models, this represents a significant step up in the size of protein language models and their capacity for unsupervised representation learning. While language modeling has seen substantial success in proteomics, language modeling has also expanded to other biological domains. Biochemistry language models learn representations of small molecules [Ross et al., 2021, Fang et al., 2023], most notably with ChemBERTa [Chithrananda et al., 2020]. Likewise, single-cell genomics has been transformed by the release of scGPT [Cui et al., 2023], and language models have even seen direct clinical use, such as with the medical question answering model Med-PaLM [Singhal et al., 2023].

Parameter-Efficient Fine-Tuning Houlsby et al. [2019] introduced adapters, which add parameters in serial to each transformer layer, allowing for every layer of the model to be trained using only a small number of parameters. The current state-of-the-art is low-rank adapters (LoRA), introduced by Hu et al. [2021], which uses low-rank adapter matrices added to the query and value weight matrices of the attention heads. In addition to parameter efficient fine-tuning, Liao et al. [2023] introduce the paradigm of *memory* efficient fine-tuning. While PEFT methods have primarily been used in natural language processing, they have recently begun to be used for tuning large biology foundation models. Yang et al. [2023] use parameter efficient fine-tuning for secondary structure prediction, and Chen et al. [2023] fine-tune their new protein language model using parameter efficient methods. Zhao et al. [2023] provide a review of large language model specialization across several domains, including biology. Dutt et al. [2023] use parameter efficient fine-tuning for medical image analysis.

Protein Interactions Cellular function is driven by interactions between proteins, but the expense and time required for experimental determination motivates the need for computational models of protein interaction. Models such as AlphaFold-Multimer [Evans et al., 2021] have recently been developed to predict the structure of interacting complexes [Zhu et al., 2023]. Quaternary structure prediction is valuable if the pair is already known to interact, but often results in degenerate prediction for pairs which don't interact, and due to the size of the model is difficult to scale to the whole-genome and all possible protein pairs. Methods like PIPR [Chen et al., 2019], D-SCRIPT [Sledzieski et al., 2021], Topsy-Turvy [Singh et al., 2022] and RAPPPID [Szymborski and Emad, 2022] predict protein-protein interaction (PPI) solely from widely-available primary sequence and are fast enough to run at genome scale. Recent work by Burke et al. [2023] has begun to close the gap between whole-genome interaction prediction and complex structure modeling, potentially unifying genome-scale

PPI prediction with complex structure prediction. The Human Reference Interactome (HuRI) [Luck et al., 2020] remains the most complete experimentally-verified human protein interaction network.

5 Discussion

In this report, we compare full fine-tuning of protein language models with parameter-efficient fine-tuning and classification on frozen representations. This work represents an exploration of the computation/performance trade-off on PPI prediction. The first significant takeaway is that even with simple models, using embeddings from ESM2 [Lin et al., 2023] improves upon the state-of-the-art—this suggests that combining more informative embeddings with more advanced models for PPI prediction could yield further improvement. Secondly, we show that parameter-efficient fine-tuning with LoRA achieves better performance than full fine-tuning with a reduced memory footprint. This reduced memory allows for tuning of deep transformer layers, likely resulting in improved learning. Full fine-tuning may also suffer due to catastrophic forgetting [McCloskey and Cohen, 1989]. Finally, learning on frozen embeddings remains a compelling option—it is computationally more efficient than either form of tuning and achieves competitive performance.

Acknowledgments and Disclosure of Funding

Thanks to Thomas Roca, Zhongqi Miao, and Zalan Fabian for helpful discussions regarding parameter-efficient fine-tuning for large language models. Thanks to Fabian Salamo for technical assistance. Thanks to Bowen Jing, Lena Erlach, and Shuvom Sadhuka for feedback on the manuscript.

S.S. was supported by the NSF Graduate Research Fellowship under Grant No. 2141064. B.B. was supported by the NIH grant R35GM141861. This research was conducted using computational resources and services at Microsoft.

References

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, 2019.

Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell Systems*, 12(6):654–669, 2021.

Judith Bernett, David B Blumenthal, and Markus List. Cracking the black box of deep sequence-based protein-protein interaction prediction. *bioRxiv*, pages 2023–01, 2023.

David F Burke, Patrick Bryant, Inigo Barrio-Hernandez, Danish Memon, Gabriele Pozzati, Aditi Shenoy, Wensi Zhu, Alistair S Dunham, Pascal Albanese, Andrew Keller, et al. Towards a structurally resolved human protein interaction network. *Nature Structural & Molecular Biology*, 30(2):216–225, 2023.

Bo Chen, Xingyi Cheng, Yangli-ao Geng, Shen Li, Xin Zeng, Boyan Wang, Jing Gong, Chiming Liu, Aohan Zeng, Yuxiao Dong, et al. xtrimopglm: Unified 100b-scale pre-trained transformer for deciphering the language of protein. *bioRxiv*, pages 2023–07, 2023.

Muhao Chen, Chelsea J-T Ju, Guangyu Zhou, Xuelu Chen, Tianran Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. Multifaceted protein–protein interaction prediction based on siamese residual RCNN. *Bioinformatics*, 35(14):i305–i314, 2019.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scGPT: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, pages 2023–04, 2023.

Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, and Timothy Hospedales. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. *arXiv preprint arXiv:2305.08252*, 2023.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with AlphaFold-Multimer. *biorxiv*, pages 2021–10, 2021.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Baohao Liao, Shaomu Tan, and Christof Monz. Make your pre-trained model reversible: From parameter to memory efficient fine-tuning. *arXiv preprint arXiv:2306.00477*, 2023.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Katja Luck, Dae-Kyum Kim, Luke Lambourne, Kerstin Spirohn, Bridget E Begg, Wenting Bian, Ruth Brignall, Tiziana Cafarelli, Francisco J Campos-Laborie, Benoit Charloteaux, et al. A reference map of the human binary protein interactome. *Nature*, 580(7803):402–408, 2020.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, 9(12):1134–1136, 2012.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *arXiv preprint arXiv:2106.09553*, 2021.

Rohit Singh, Kapil Devkota, Samuel Sledzieski, Bonnie Berger, and Lenore Cowen. Topsy-Turvy: Integrating a global view into sequence-based ppi prediction. *Bioinformatics*, 38(Supplement_1): i264–i272, 2022.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, pages 1–9, 2023.

Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982, 2021.

Joseph Szymborski and Amin Emad. RAPPPID: towards generalizable protein interaction prediction with AWD-LSTM twin networks. *Bioinformatics*, 38(16):3958–3967, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wei Yang, Chun Liu, and Zheng Li. Lightweight fine-tuning a pretrained protein language model for protein secondary structure prediction. *bioRxiv*, pages 2023–03, 2023.

Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Li Yun, Hejie Cui, Zhang Xuchao, Tianjiao Zhao, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*, 2023.

Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics*, 39(7):btad424, 2023.

S1 650M vs. 3B ESM2 Model

Typically in language modeling, larger models yield better performance leading to increasingly large models being trained. However, when we compared a multilayer perceptron classifier (MLP) trained on frozen embeddings from the 650M parameter model to the 3B parameter model, the 650M parameter model actually performed slightly better (Table S1). This indicates that even with reduced compute capacity available, smaller models may be sufficient to achieve good performance on proteomics tasks. All results presented below use the 650M parameter version of ESM2.

Table S1: Test set performance of a MLP classifier trained on pooled embeddings from the 650 million and 3 billion parameter versions of ESM2 with frozen weights. While the two models are competitive, the 650M parameter version outperforms the larger 3B parameter version in accuracy, MCC, AUPR, precision, and specificity. The 3B parameter model achieves a higher F1 score and recall.

	Accuracy	**F1**	**MCC**	**AUPR**	**Precision**	**Recall**	**Specificity**
650M	**0.631**	0.632	**0.261**	**0.684**	**0.630**	0.633	**0.623**
3B	0.607	**0.650**	0.221	0.656	0.586	**0.730**	0.484

S2 Baseline MLP Model

As a baseline to compare with fine-tuning, we train an MLPClassifier model from scikit-learn using embeddings extracted from ESM2 (650M parameters). Parameters for the MLPClassifier were selected by cross-validation on macro average precision over a grid search. We searched over all combinations of

- $activation = [“logistic'', “relu'', “identity'']$
- $alpha = [0.0001, 0.001, 0.01]$
- $learning\_rate\_init = [0.001, 0.01]$
- $max\_iter = 1000, 2000$
- $hidden\_layer\_sizes = [(64,), (128,), (512,), (64, 64), (128, 128), (64, 64, 64)]$
- $tol = [1e-4, 1e-5]$

S3 PEFT and FT Training and Validation Curves

We show training and validation loss curves, as well as validation AUPR curves, over training in Figure S1. We show validation recall and specificity curves in Figure S2. In Figure S3, we show training and validation loss curves, as well as validation AUPR curves, for all different combinations of Q/K/V matrices tested in Table 3.

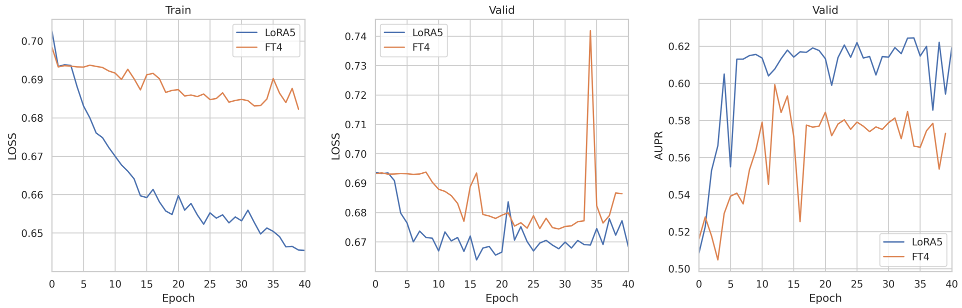

Figure S1: Training and validation loss curves, AUPR curves for FT (4 layers) and PEFT (5 layers) from Table 2. Note that all other training parameters were held constant between these runs.

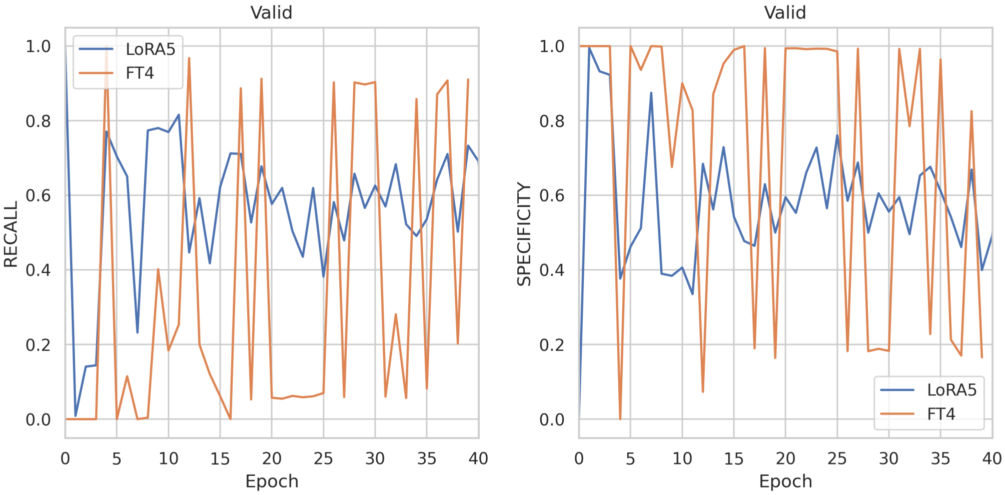

Figure S2: Recall and specificity fluctuate wildly throughout training with traditional fine-tuning, compared to the PEFT model training, which is much more stable. This explains why test set recall is so high, but specificity so low in Table 2—these are highly dependent on the specific epoch which was chosen based on validation AUPR.

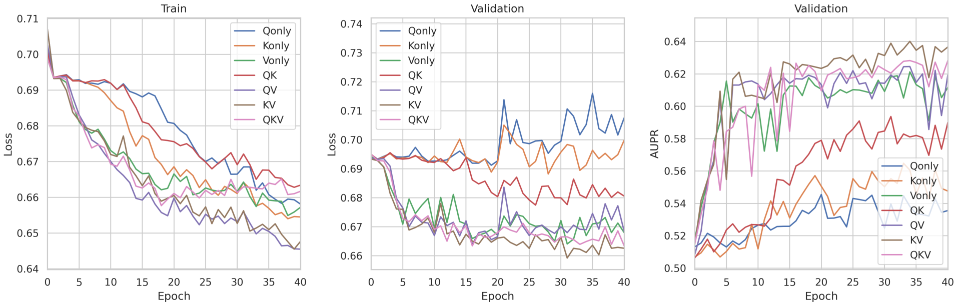

Figure S3: Training and validation loss curves, AUPR curves for models trained with LoRA adapters on Q, K, V, QK, QV, KV, QKV matrices from Table 3. Note that all other training parameters were held constant between these runs.