ML interview project

Jary Pomponi

27, October 2025

1 Problem and solution

The standard ESM2 architecture supports sequences up to 1020 residues. However, due to computational limitations [1], However, fine-tuning at this length was computationally infeasible, even when applying parameter-efficient fine-tuning (PEFT) methods such as LoRA, due to the quadratic scaling of attention with sequence length. The attention scales quadratically with the sequence length, and I wanted to mitigate this effect. Additionally, I was not fond of the idea of truncating testing proteins, and my solution also addresses this aspect.

For this purpose, I implemented a simple sampling strategy: instead of using a whole protein for each forward, I sub-sampled a portion of its chain. By using a subset of each protein, more iterations of the training dataset become possible with the same training tokens budget. Additionally, the forwarding of the model became less computationally demanding. I fixed the subset size to 200 for each experiment.

The model I selected is ESM2 8M UR50D. For the fine-tuning, I used LoRA [1] over the query and key weights of each attention layer. The LoRA parameters I used are: r=8, $\alpha = 8$, and dropout=0.05.

After reviewing the literature about the problem (e.g., [2, 3, 4, 5]), I decided to fine-tune using the contact head from ESM2. Later, I will also introduce how information about other proteins could be integrated.

To this end, I used Adam with a learning rate of 0.0001 and a budget of 5M training tokens for all the experiments. I also used a developer set to monitor the training, composed of 10% of training proteins.

1.1 Data pre-processing

Following the main ESM2 paper, I decided to use C_β contact distances as a binary target. I also followed the same distance calculation, based on [6].

In addition to the standard amino acids, I decided to keep also MODRES mapping from the PDB header, if present, and maps back hetero residues as

[1] While working on this project, a server I use was in maintenance and I had access to only my laptop. It has a GeForce RTX 4080 with 12 GB of VRAM, which was not enough to train the full model.

variants of standard amino acids, if possible. To do that, I implemented a custom PDB reader, which is an extension of the one present in Biopython API [7].

2 Results

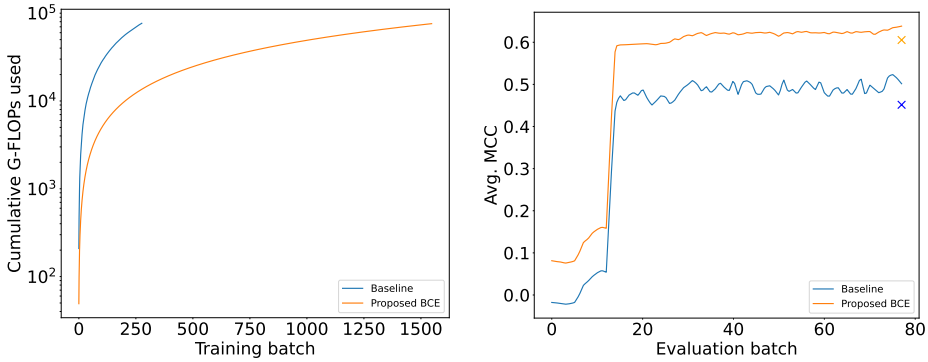

Figure 1: The FLOPS (left) and the results (right) for the two models. On the right, the X mask represents the test MCC, while the same metric on the development set.

In this section, we compare the results obtained from all the experiments. The metric used is protein-wise MCC averaged, as proposed in [8], calculated both on the testing set and the developer subset, and total training flops. As the baseline, I fine-tuned the EMS2 with the standard training approach, having 1020 tokens as input to the model.

As shown in Figure 1 (left), the standard fine-tuning approach quickly exhausts the available Tokens budget (5M) due to its high sequence length. In contrast, the proposed sub-sampling strategy extends the number of effective training iterations within the same budget. This longer training yields higher Matthews Correlation Coefficient (MCC) values for both the development and test sets (Figure 1, right).

3 Incorporating other proteins

Instead of incorporating other proteins in the process, my idea was to incorporate 3D structures (β-sheets and α-helix) of similar proteins. Given a pre-trained and fixed model and a training protein, a similar protein is extracted from an open and public database.

From the known protein, the overlapping sequences between it and the unknown one that contain such structures are extracted in the form of vectors, using the same model. It is done using the same model, but pooled based on

the position of each component in the structure (e.g., a helix token could be created by pooling tokens of residues belonging to it, plus an entity token for helices). Such vectors are then used in a cross-attention decoder model ending with a Contact Head, which is trained to predict the contact in the training protein.

Unfortunately, due to the computational limitations exposed before, implementing the whole pipeline was not feasible.

4 Estimated working time

Here, an estimation of the time needed to complete each part of the project:

- Literature overview and solution definition: 4-5 hours
- Defining scope, architecture, and design of the code: <1 hour
- writing and testing code: 4-8 hours
- Cleaning code and writing documentation: 1 hour
- Experiments and report writing: 2 hours

The complete code-base can be found in the following GitHub repository: `https://github.com/jaryP/ContactProject`.

5 Conclusion

This work proposed an efficient fine-tuning strategy for ESM2 in residue–residue contact prediction. By sub-sampling protein sequences, the approach reduced computational cost while maintaining predictive accuracy, achieving higher MCC scores than standard fine-tuning. Although limited by hardware, the results suggest that such sampling can effectively extend training within fixed budgets. Future work could explore integrating structural information from similar proteins to further enhance performance or integrating a custom positional embedding to cope with the subsampling strategy.

6 Appendices

A Alternative approach: contrastive learning

When this project was presented to me, I was following a PhD student doing work about self-supervised learning. So, I decided to give it a try also for this problem.

I implemented a contrastive loss and a sampling strategy, combined towards the goal of training the model to push closer pairs of residues with a contact in

the same protein, and apart from the negative ones. The core idea was to teach the model an underlying geometric structure.

The approach achieved around 30% MCC score, which was not comparable with the baseline. I suspect that more training time is needed for such operations, a bigger batch size (I implemented a gradient cumulation strategy to overcome this limit), and probably a careful tuning of other parameters (e.g., number of positive residues vs negatives). Said so, the overall approach was very efficient in the pipeline, and maybe worth exploring.

B Better positional embeddings

Since the proposed approach truncates the sequence, my idea was to add an offset to the rotary embedding to help preserve the distance across the training. However, my first implementation required some more calculations than the standard one, and since without it the model already performed well, I dropped the idea. The code is still present.

References

[1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[2] Yawen Sun, Rui Wang, Zeyu Luo, Lejia Tan, Junhao Liu, Ruimeng Li, Dongqing Wei, and Yu-Juan Zhang. Esm2_amp: an interpretable framework for protein–protein interactions prediction and biological mechanism discovery. *Briefings in Bioinformatics*.

[3] Jaspreet Singh, Thomas Litfin, Jaswinder Singh, Kuldip Paliwal, and Yaoqi Zhou. Spot-contact-lm: improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics*.

[4] J. Huang, J. Li, and Q. et al. Chen. Freeprotmap: waiting-free prediction method for protein distance map. *BMC Bioinformatics 25, 176 (2024)*.

[5] Zhidian Zhang, Hannah K. Wayment-Steele, Garyk Brixi, Haobo Wang, Dorothee Kern, and Sergey Ovchinnikov. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45):e2406285121, 2024.

[6] Rotamer libraries in the 21st century. *Current Opinion in Structural Biology*, 12(4):431–440, 2002.

[7] Biopython, at https://biopython.org/docs/1.75/api/index.html.

[8] Giuseppe Jurman Davide Chicco. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation.