Deep learning models for unbiased sequence-based PPI prediction plateau at an accuracy of 0.65

Timo Reim, Anne Hartebrodt, David B Blumenthal, Judith Bernett, Markus List

LifeLU reading group

presented by Özdeniz Dolu

11.09.2025

1. Introduction

- Protein-protein interactions (PPI for short) are crucial for understanding biological functions as at least 80% of them involve PPI’s. (Zhou et al. 2016)

- Experimental techniques are costly and low throughput compared to computational ones.

- The interactions are best understood in three-dimensional context. However, this study focuses on sequence-only methods.

1. Introduction

- Due to data leakage and improper data splitting, many sequence-only methods reported inflated accuracies (as high as 90%).
  - Very similar sequences appearing in both training and test datasets.
  - (Example) Training protein p_1 appears ONLY in positive set, therefore always predicted as interacting.
- Authors previously proposed a leakage-reduced gold standard PPI dataset in 2022. Since then, sequence-only methods using that dataset achieved around ~0.63 accuracy.
- Aim is to examine, test and discuss various aspects of deep learning approaches in this context.

2.1 Data

Dataset is divided into three groups:

Intra0 (val), Intra1 (train), and Intra2 (test), with 59260, 163192, and 52048 entries

All proteins belong to whole human proteome where splits are partitioned at %40 pairwise sequence similarity.

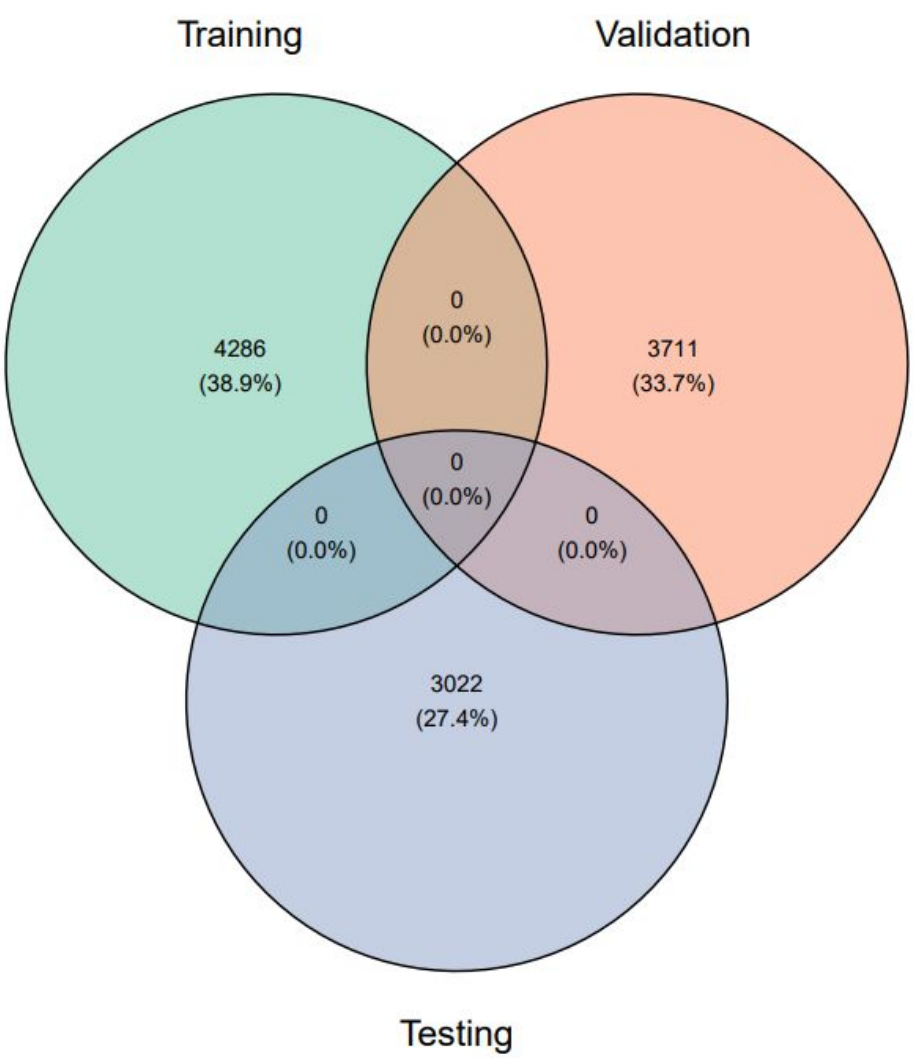
Positive samples are selected from HIPPIE v2.3 database.

Negative samples are generated randomly (matching the degree of the protein in the positive graph).

Negative and positive samples are balanced.

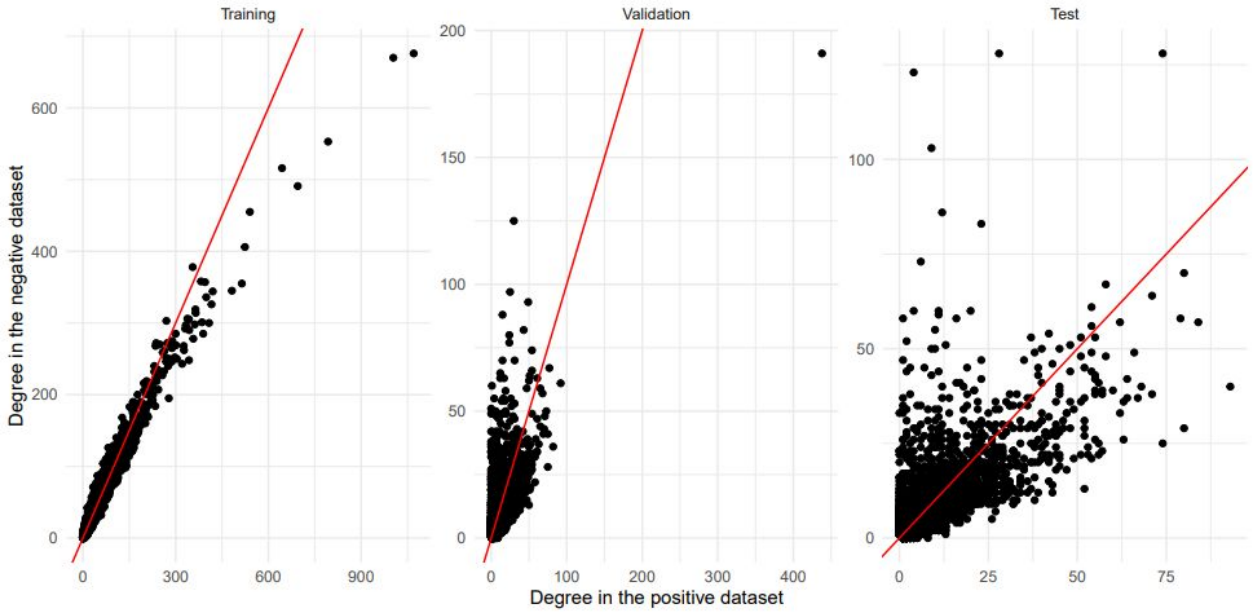
2.1 Data

Venn diagram for unique proteins in the dataset



2.1 Data

Positive vs negative node degree of each protein in each split (red line is optimal, y=x)



2.2 Embeddings

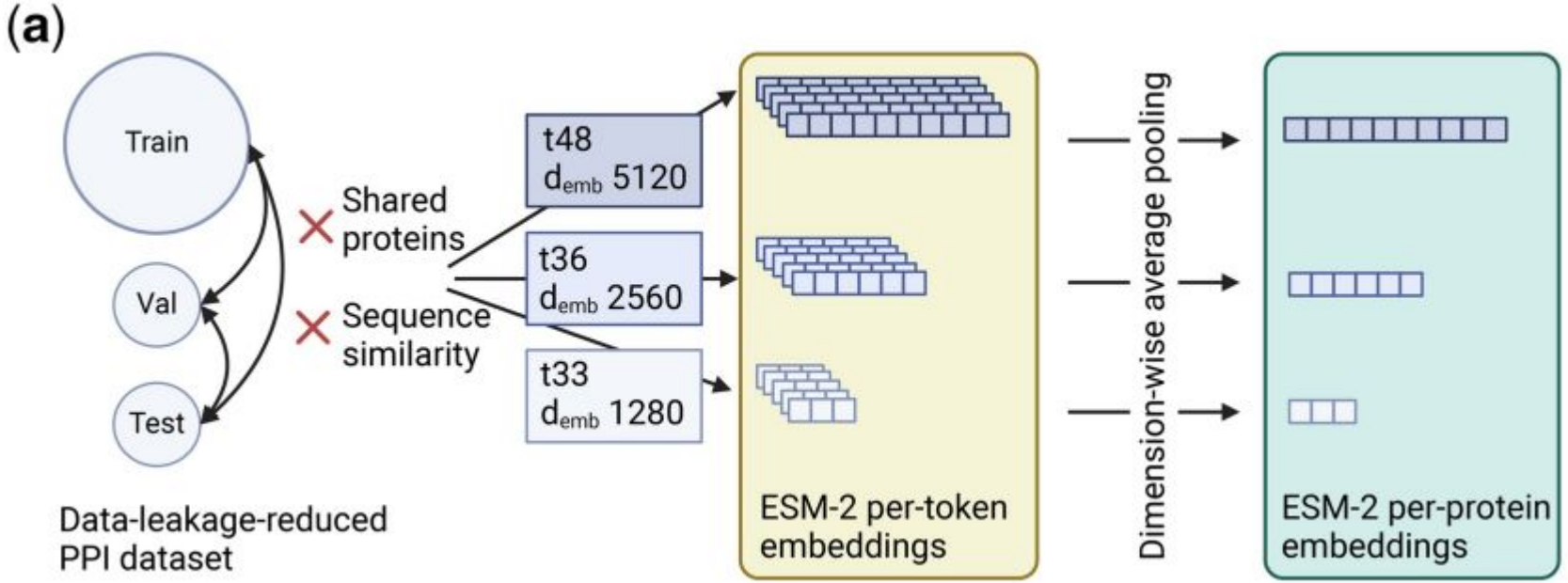
Embeddings from three largest ESM-2 models have been utilized.

- Models are referred to as t48, t36, and t33.
- Models have 15B, 3B, and 650M parameters respectively.
- Embedding dimensions are 5120, 2560 and 1280 respectively (d_emb).

Per-protein embeddings have been obtained by dimension-wise average pooling. (It’s a vector of size d_emb).

For models that use per-token embeddings, proteins containing more than 1000 residues are filtered out due to ESM-2 length limitation leaving 93719, 46421, and 41100 samples in the training, validation, and test sets.

2.2 Embeddings



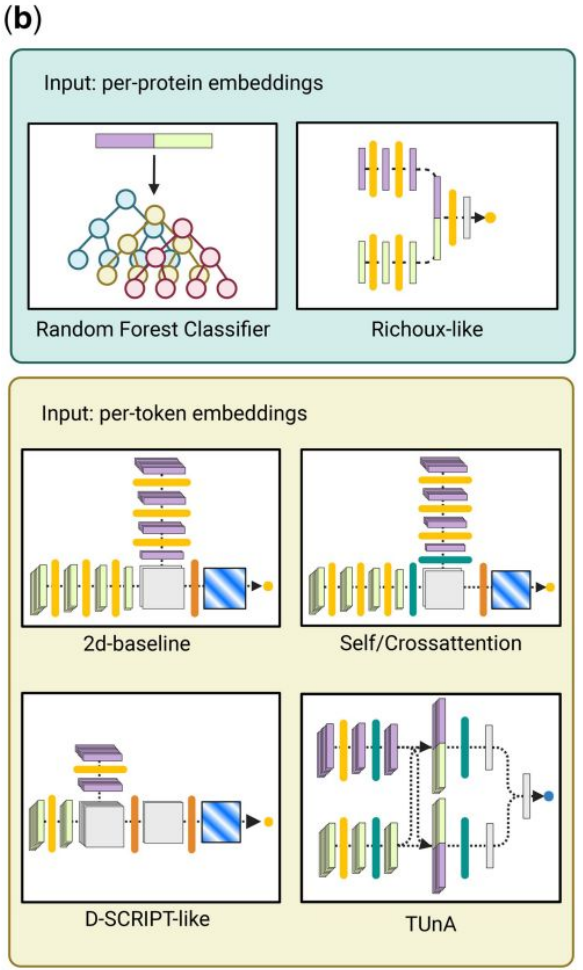
2.2 Embeddings

Model	ESM-2	Bepler & Berger	One-Hot	Embedding Type
RFC-40	0.577	–	–	PP
2d-baseline	**0.575**	0.569	0.501	PT
2d-Crossattention	**0.641**	0.613	0.502	PT
2d-Selfattention	**0.616**	0.581	0.501	PT
Richoux-ESM2	**0.633**	0.617	0.527	PP
D-SCRIPT-ESM-2	**0.628**	0.525	0.502	PT
TUnA	**0.645**	0.643	0.502	PT

2.3 Models

Models named D-SCRIPT, Richoux, and TUnA (from SOTA) have been re-implemented based on the descriptions given in their publications.

Other than that, random forest classifier and some simple architectures are also used as baselines.



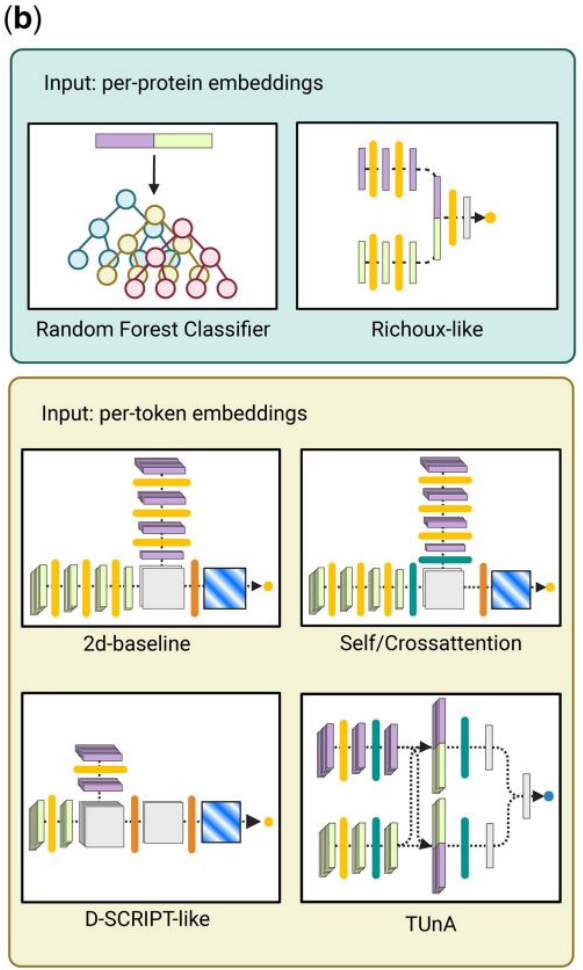
2.3.1 Baseline Models

For Random Forest Classifier (RFC for short):

RFC-40 refers to usage of vectors of size 20 for each protein, reduced via PCA.

RFC-400 is similar for 200.

RFC-mean is unreduced.

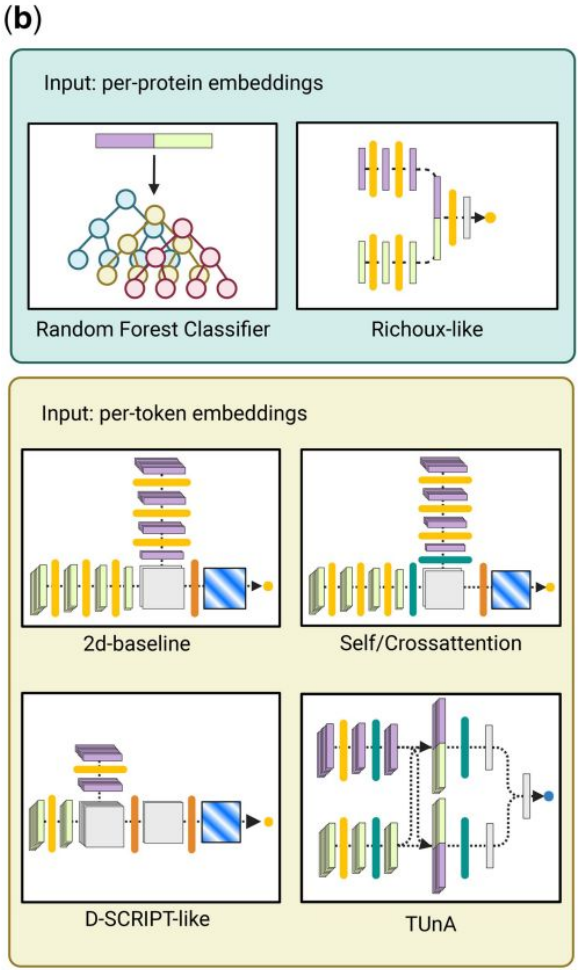


2.3.1 Baseline Models

For 2d-baseline:

2d refers to usage of per-token embeddings in input since it's two dimensional in that case.

2d-baseline model has a simple (compared to SOTA) three layer architecture involving FNNs and an outer product.



2.3.1 Baseline Models

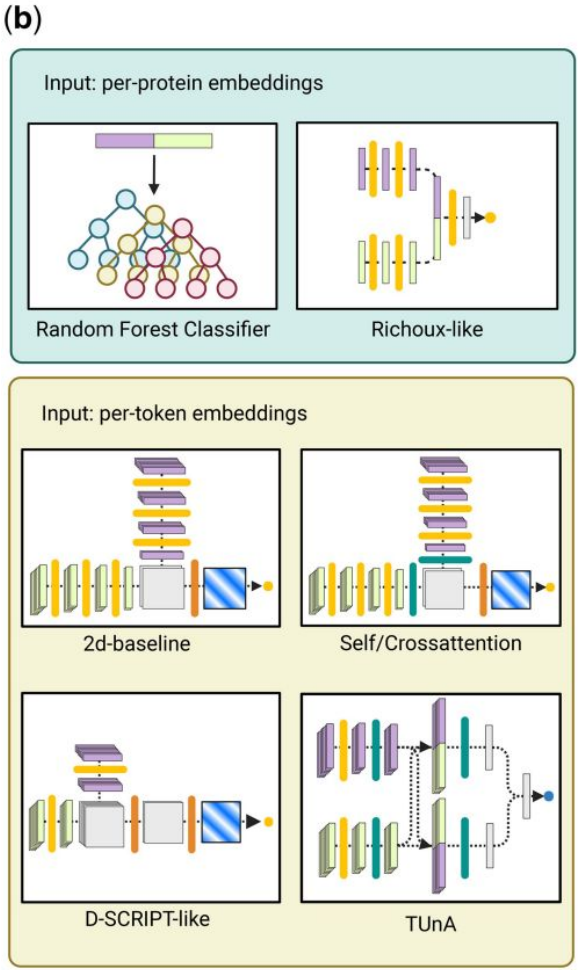
For Self/Cross-attention:

Modification upon 2d-baseline architecture.

Transformer encoder is added at the end of linear layers.

Spectral normalization is applied in transformer's linear layers.

Cross/self refers to inter/within protein attention.



3 Results

Table 1. Comparison of all tested models.

Model	Accuracy	Precision	Recall	F1	AUPR	Loss	Training time	Best epoch	Training time per epoch
Baseline models									
RFC-40	0.577	0.629	0.374	0.496	0.584	–	474	–	–
2d-baseline	0.575	0.630	0.373	0.468	0.635	0.738	30845	22	1234
Attention models									
2d-Crossattention	0.641	0.660	0.589	0.623	0.678	0.643	29411	6	2101
2d-Selfattention	0.616	0.611	0.553	0.591	0.641	0.670	41326	14	1878
Adaptions of published models									
Richoux-ESM-2	0.633	0.627	**0.654**	**0.640**	0.655	0.653	2483	8	**155**
D-SCRIPT-ESM-2	0.628	0.638	0.602	0.619	0.636	0.650	29849	7	1990
TUnA	**0.645**	**0.672**	0.580	0.622	**0.692**	**0.630**	30013	7	2001

3.1 Hyperparameter optimization did not yield better parameter combinations than defaults

In general, effect of hyperparameter optimization on model performance was not observed to be very high.

For models using attention (2d-self, 2d-cross, TUnA), learning rate had high negative correlation.

For 2d-self, max pooling and for 2d-cross, avg pooling was better.

3.2 ESM-2 embeddings boost test accuracies of all models to around 0.65

Generally, lower parameter size (and embedding size) observed to perform better.

Looking at main table, embedding table and fig 2, we see no model really have a good performance on this task, being barely above random baseline.

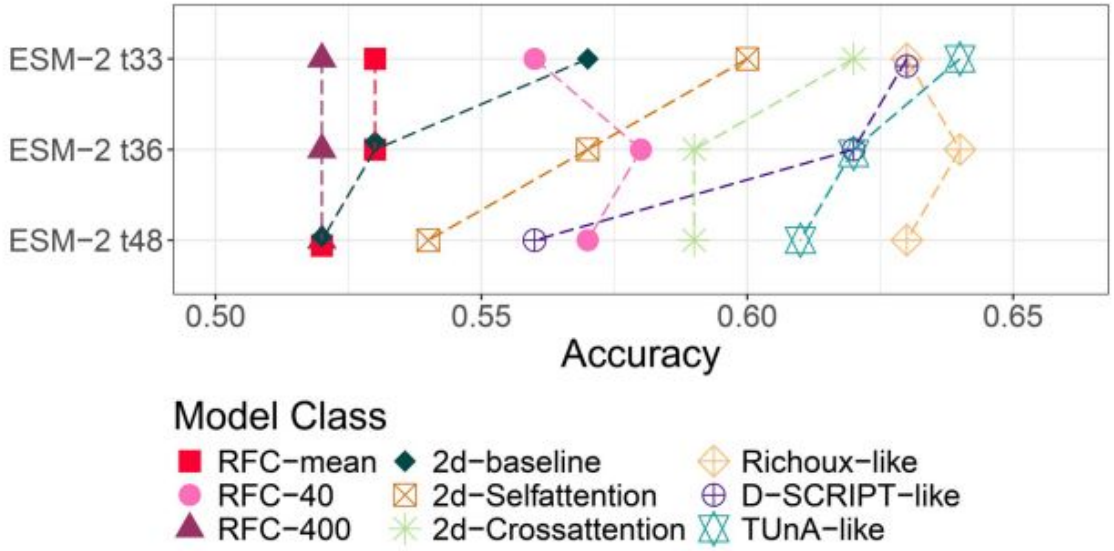


Figure 2. Validation performance with increasing ESM-2 embedding size. Most models perform best with the smaller t33 embedding.

3.3 Per-token embeddings do not yield better results than averaged per-protein embeddings

As seen on Supplementary Table S1 (too large to show here), models that use per-token embeddings do not have a large advantage over per-protein embeddings. However, the best performing models TUnA and 2d-Crossattention use per-token.

Models that use per-protein embeddings have less parameters and compute faster.

One would expect interaction specific information to be encoded more in token-level than in protein-level.

3.4 Models profit from smaller embeddings

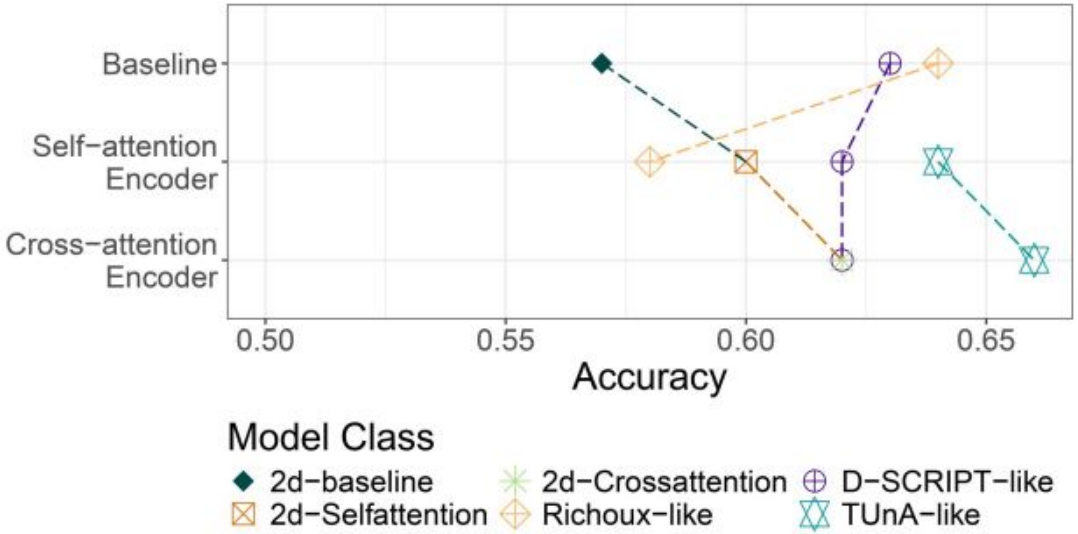
Lower embedding sizes have seen to be performing better on PPI task.

One thing to note here is that the performance on the training set is similar across various embedding sizes. This may be an indication of overfitting in larger embedding sizes.

3.5 Only the 2d-baseline profits from attention

When modified with encoders that use attention mechanism, SOTA models did not benefit. In some cases, drops in performance was observed.

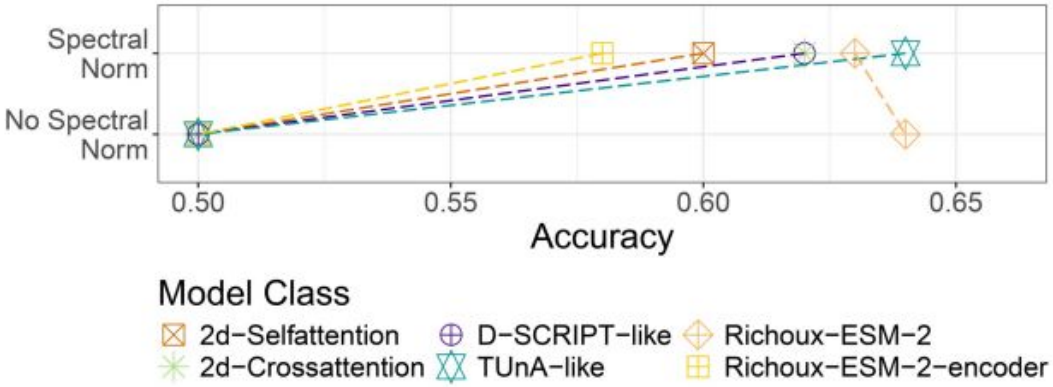
This may be indicative of a barrier of 0.65 accuracy for this dataset with ESM2 embeddings.



3.6 Attention-based models profit from spectral normalization

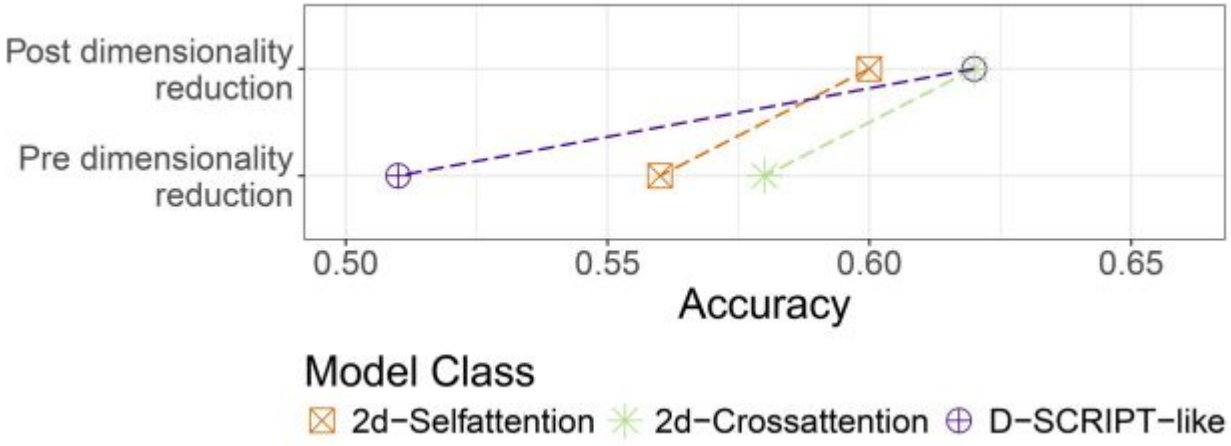
As seen in the figure, almost all attention-based models start behaving randomly when spectral normalization is removed from the model.

Models were observed to be either guessing all positive or all negative (without spectral normalization).



3.7 Attention should be applied after reducing embedding size

In all tested models, attention (encoder) layer positioning matters. It works best after reducing the dimensions. (with lower embedding sizes).



3.8 Distance maps cannot be predicted implicitly

The models D-SCRIPT-ESM-2, 2d-baseline, 2d-Selfattention and 2d-Crossattention internally generate a matrix of size len(p1) x len(p2).

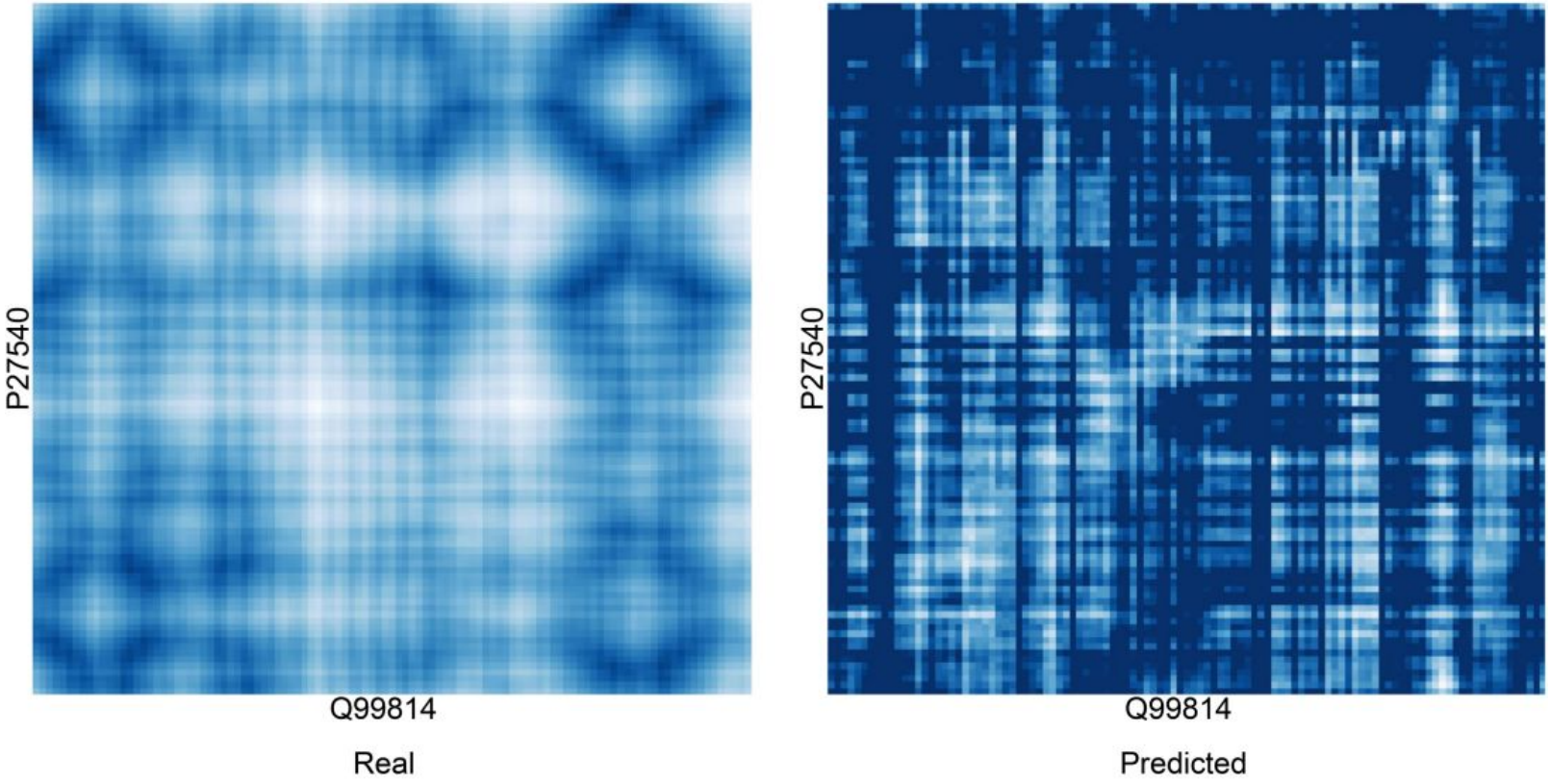
D-SCRIPT authors treats this as implicit contact map prediction.

To make a similar analysis, every interaction in the dataset that is predicted with a high confidence (>0.9) is considered. Out of 84 such interactions, 54 filtered out due to them having other ligands, homomers etc. in the experimental data. 19 more filtered out due to missing sequence data.

This left couple of cases to investigate. Although some important regions seem to be detected by the models, they don’t show high correlation (S16 to S18)

3.8 Distance maps cannot be predicted implicitly

Pearson Correlation: 0.05



4 Discussion

- All models encountered an accuracy barrier of 0.65.
- Authors associate this barrier to models being sequence-only and usage of ESM-2 embeddings. (? Can’t it also be due to information content of the dataset ?)

As for the limitations:

- Only ESM-2 embeddings are explored.
- The dataset
- PPI is generally more complex than a mere interaction of 2 chains
- Author’s resource limitations for testing and validation.

Thanks for listening!