Final Project Report for CS 184A/284A, Fall 2025

Project Title: Deepening Embedded Drug-Target Interactions with ConPLex
Project Number: 22

Student Names

Mohammadarshya Salehibakhs; 71713160; msalehib@uci.edu

Lucas Ueta; 16166139; uetal@hs.uci.edu

1. Introduction and Problem Statement

This project aims to reproduce and extend the ConPLex model (Singh et al., 2023), which predicts drug–target interactions (DTIs) using pretrained protein language models and contrastive learning. We will experiment with alternative neural architectures—adding deeper projection heads, nonlinearities, and potential cross-attention—to evaluate whether these modifications improve predictive accuracy over the original shallow model.

2. Related Work

This project is wholly based on the research paper, "Contrastive learning in protein language space predicts interactions between drugs and protein targets," by Singh et al. We are building upon their work in order to make some small, yet tangible improvements to their predictive model. The strength of their model/pipeline is the innovative idea of embedding both proteins and drugs within a latent space to predict their potential to bind/binding affinity. It is a conceptually engaging idea that potentially opens the doors to many other applications that are currently limited by pairwise computations. Such computations run in quadratic time, but they could potentially implement a similar embedding algorithm to speed up the screening process to linear time.

That being said, one shortcoming of the paper is the architecture used for the co-embedding: a single fully-connected layer. This perceptron architecture is not flawed in itself, but the lack of experimentation with different architectures was the inspiration for this project. We believe that there is another co-embedding model better suited for these datasets.

We will make use of the existing open-source code referenced in the research paper. Fortunately, all of the code is very well-maintened, commented, and iterable, enabling us to add our own architectures with ease. The GitHub repository with the code used during the writeup of the paper can be found here: https://github.com/samsledje/ConPLex_dev; while the updated repository with minor fixes can be found here: https://github.com/samsledje/ConPLex.

3. Data Sets

Table 4. Full specification of benchmark datasets

Dataset	Drugs	Targets	Median Coverage	# Training	# Validation	# Test
BIOSNAP	4,510	2,181	0.0023/0.0020	9,670/9,568	1,396/1,352	2,770/2,727
Unseen Drugs				9,535/9,616	1,383/1,353	2,918/2,675
Unseen Targets				9,876/9,499	1,382/1,386	2,578/2,762
BindingDB	7,165	1,254	0.0008/0.0010	6,334/6,334	927/5,717	1,905/11,384
DAVIS	68	379	0.3707/0.3676	1,043/1,043	160/2,846	303/5,708
TDC-DG	140,746	477	0.0021/0.0005	146,891	36,539	49,028
Phosphatase	165	218	1.0/1.0	5,054/27,286	—	370/3,260
Esterase	96	146	1.0/1.0	2,150/10,426	—	926/514
Glycosyltransferase	89	54	0.9259/0.9778	725/3,042	—	113/417
Halogenase	62	42	1.0/1.0	303/1,991	—	20/290
BKACE	17	161	1.0/1.0	255/2,193	—	19/270
DUD-E †				8,996/406,208	—	11,430/521,132
GPCR	99,671	5	18,563			
Kinase	315,399	26	15,409			
Protease	286,089	15	9,271			
Nuclear	151,133	11	16,257			

The original model used the following datasets:

- ***BIOSNAP*** *(Stanford Biomedical Network Dataset Collection)*

  "This is a drug-target interaction network that contains information on which genes (i.e., proteins encoded by genes) are targeted by drugs that are on the U.S. market. Drug targets are molecules that play a critical role in the transport, delivery or activation of the drug. Drug target information is widely used to facilitate computational drug target discovery, drug design, drug docking or screening, drug metabolism prediction, drug interaction prediction, and general pharmaceutical research."

- ***BindingDB***

  "BindingDB curates US Patents. We have scanned patents back to 2013 for suitable data and are currently up to date as of mid-2020… BindingDB curates a set of journals not covered by other public databases."

- ***DAVIS***

  "Dataset Description: The interaction of 72 kinase inhibitors with 442 kinases covering >80% of the human catalytic protein kinome. Task Description: Regression. Given the target amino acid sequence/compound SMILES string, predict their binding affinity. Dataset Statistics: 0.3.2 Update: 25,772 DTI pairs, 68 drugs, 379 proteins. Before: 27,621 DTI pairs, 68 drugs, 379 proteins."

- ***TDC-DG*** *(Therapeutics Data Commons - Domain Generalization)*

  "In this benchmark, we use DTIs in TDC.BindingDB that have patent information. Specifically, we formulate each domain consisting of DTIs that are patented in a specific year. We test various domain generalization methods to predict out-of-distribution DTIs in 2019-2021 after training on 2013-2018 DTIs, simulating the realistic scenario."

- ***DUD-E*** *(Database of Useful Decoys: Enhanced)*

"An enhanced and rebuilt version of DUD, a directory of useful decoys. DUD-E is designed to help benchmark molecular docking programs by providing challenging decoys."

- *Protein-family specific datasets:* ***Phosphatase, Esterase, Glycosyltrasferase, Glycosyltransferase, Halogenase, BKACE***

These datasets/benchmarks were split into three categories in the original paper, reflecting their structure and how they were used to train the pair of co-embedding models:

- ***Low coverage***

  Datasets with little coverage - defined as the mean ratio of proteins that have been experimentally tested against each drug, and vice versa. These datasets are composed (almost) entirely of positive drug-target interactions (DTIs), where the drug compound has been experimentally shown to bind to the protein's active site.

  To artificially generate negative DTIs, the original research team took a random unknown DTI and assumed it to be negative. Statistically, this is almost always the case. But we believe that training under these unchecked artificial data is an important flaw in this study that should be addressed in the future.

  Training with these datasets used cross-entropy error as the loss function, focusing on overall accuracy (and recall) in DTI predictions over precision.

  Includes: ***BioSNAP, BindingDB, DAVIS***

- ***Continuous***

  Dataset with binding affinity data, giving a continuous prediction target instead of a classifier response from the model.

  This dataset allows the model to adjust its predictions to create a more representative likelihood of binding based on experimental data, allowing even DTIs that were not present in this dataset, to have their binding affinity predicted.

  Training with this dataset used mean squared error as the loss function.

  Includes: ***TDC-DG***

- ***High coverage***

  Datasets with high coverage. These datasets conversely show many more samples of negative DTIs. Notably, these also include decoys - drug compounds similar to true binding compounds that don't actually bind.

  Training with these datasets employ contrastive learning to reduce rates of false positives, improving the models precision over accuracy.

Because some datasets (e.g., Surfaceome, TDC-DG) are large or require special processing, we propose using ***DAVIS****, due to its small size. Additionally, DAVIS poses an interesting challenge due to its many*

few-shot data points: the research paper noted particularly low AUPR scores and lacking recall for this dataset. We would like to improve upon it.

4. Description of Technical Approach

We will begin by reproducing the core ConPLex pipeline:

1. ***Protein embedding:*** *Pretrained protein language models (ProtBert, ESM2, or ProSE).*
2. ***Drug embedding:*** *Morgan fingerprints (baseline), plus experiments with learned embeddings (Mol2Vec or GNN-based embeddings).*
3. ***Projection into joint latent space:*** *Baseline single perceptron with a fully-connected layer.*

Our work begins by researching different neural network architectures that may show some improvements upon the baseline model. We settled on the following architectures: ***CNN, Cross Attention, Duo Layer Perceptron, Quituple Layer Perceptron, Residual***

We will start off with an initial screening of these models, training up to epochs on each model.

We will be using binary cross-entropy, since DAVIS is a low-coverage dataset. Contrastive learning would not be appropriate in our case, since DAVIS does not provide decoys. We will assess the quality of the model based on its AUROC and AUPR scores on a separate validation and testing datasets.

We found that the Duo Layer Perceptron model (with a hidden layer dimension of 1024) was the only one to surpass the baseline model. So we created a few variations of it:

- ***Small Duo*** *Layer Perceptron (hidden layer dimension: 512)*
- ***Large Duo*** *Layer Perceptron (hidden layer dimension: 2048)*
- ***Triple Layer*** Perceptron

After repeating the initial screening, we found the simple **Duo Layer Perceptron** to perform the best. So we measured its performance by training it till it reached a plateau in its performance (~30 epochs) against the baseline.

5. Software

We used several open-source libraries:

- ***PyTorch*** *(deep learning)*
- ***HuggingFace*** *Transformers (ProtBert, ESM2)*
- ***RDKit*** *(Morgan fingerprints)*
- ***scikit-learn*** *(evaluation)*
- ***PyTorch Metric Learning*** *(optional)*
- ***TDC*** *(data loading utilities)*
- ***Matplotlib*** *(data visuals)*
- ***Pandas*** *(dataset manipulation)*

… as well as the original ***ConPLex*** *model released on GitHub. Of note, the repository provided us the following helpful tools:*

- *Featurization models for the proteins and drug compounds*
- *A customisable training pipeline*
- *The original* ***baseline*** *model*

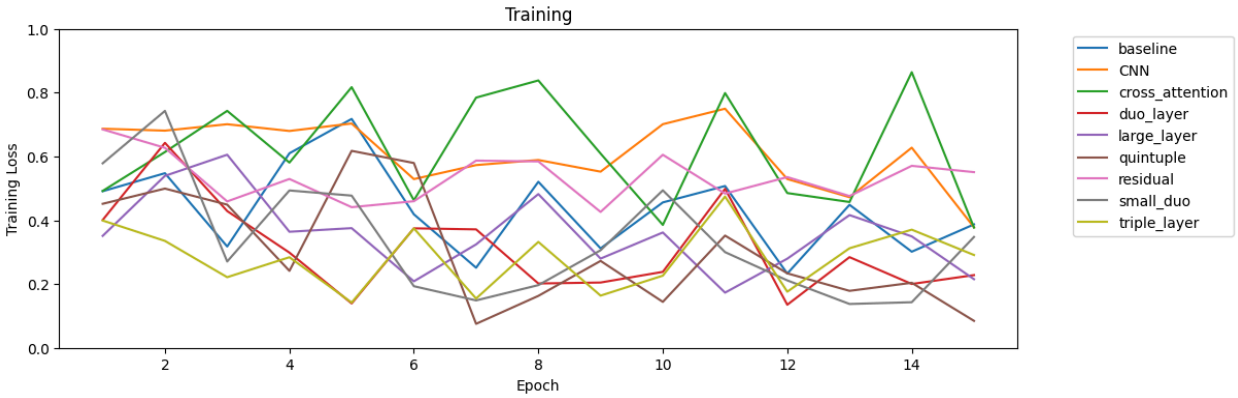
- *A well documented **CLI** for training models and downloading datasets*

As a group, we personally wrote:

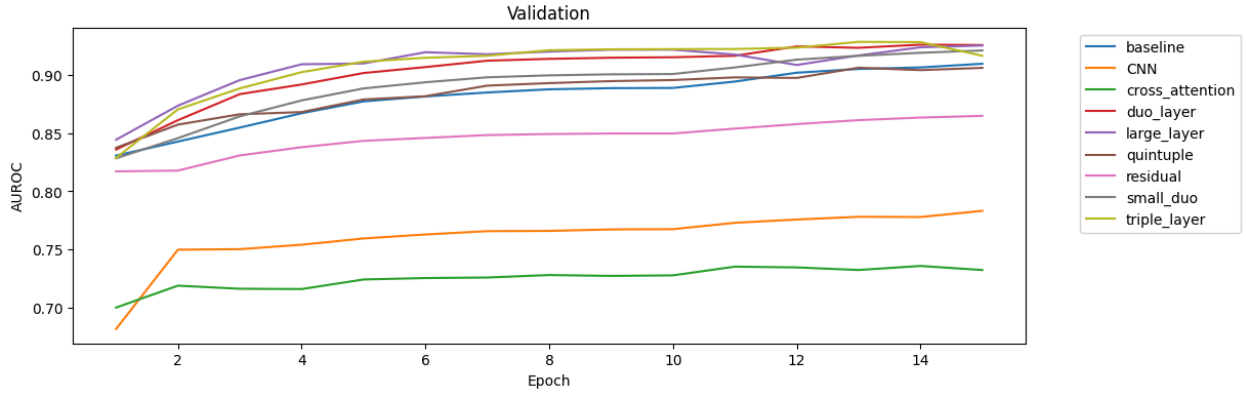
- *PyTorch modules as the varied neural network architectures (architectures.py)*
    - *ResidualCoembedding: Adds residual blocks with learnable alpha and dropout to the projection heads.*
    - *CrossAttentionCoembedding: Implements a cross-attention mechanism where drug embeddings attend to target protein embeddings.*
    - *DeepCoembedding: A deeper Multi-Layer Perceptron (MLP) projection head.*
    - *ModernBaseline: An improved baseline with LayerNorm, GELU activation, and learnable temperature scaling.*
    - *DuoLayerPerceptron: A MLP with a single hidden layer (size: 1024).*
    - *LargeDuoLayerPerceptron & SmallDuoLayerPerceptron: Similar, but with half and twice the size, respectively.*
    - *TripleLayerPectron: A MLP with 2 hidden layers - both of size: 1024*
    - *QuintupleLayerPerceptron: A MLP with 4 hidden layers of diminishing size.*
    - *CNN: A three-layer convolutional structure with a 5 x 5 kernel size that sequentially increases the number of feature maps from 32 to 64 to 128.*
- *Shell scripts to execute our custom training pipeline, derived partially by the default_config.yaml.*
- *A python notebook to rebuild trained models and evaluate them based on the testing dataset.*
- *The project notebook used to analyse the performance of our models through graphs and different performance metrics.*

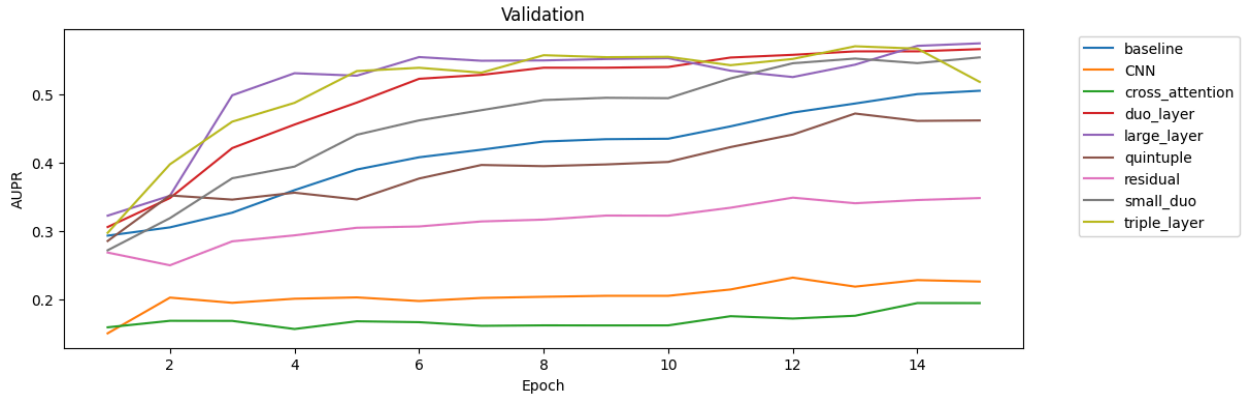
6. Experiments and Evaluation

*As we covered in the **Description of Technical Approach**, models were trained first with 15 epochs using the DAVIS dataset, without contrastive learning. We used the baseline's dynamic learning rate, which may be responsible for the hectic training loss performance of most models. To address this fluidity, we kept a "best model" - based on the validation dataset - for the final testing evaluation. Each training procedure took ~30 minutes.*



We also conducted some validation tests for every epoch of training. Allowing us to demonstrate the improvements made to the models during training, below.

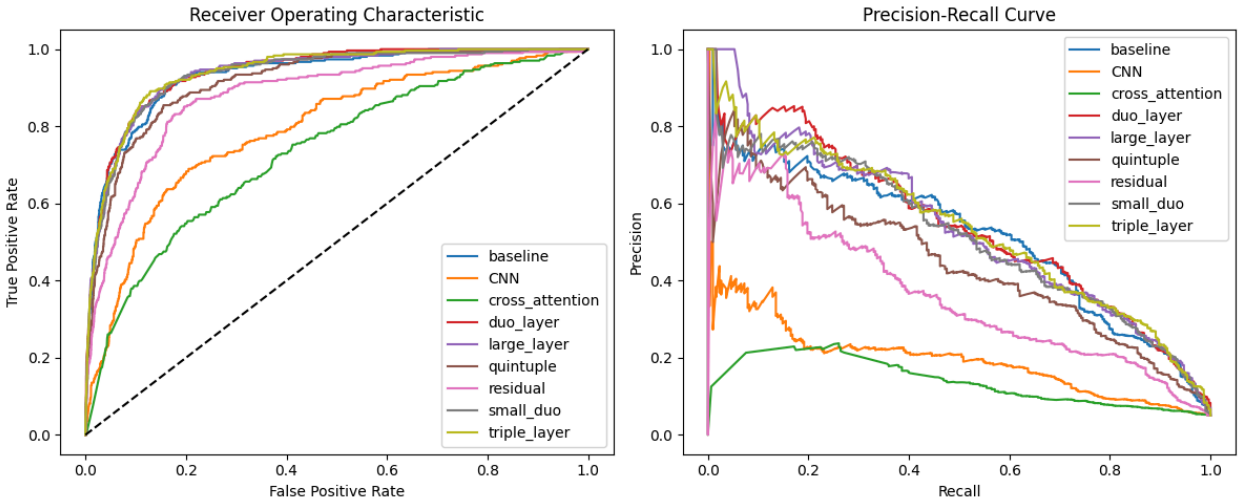




We decided to measure the AUROC and AUPR curves to gather a comprehensive assessment of the models' abilities to distinguish between positivity and negative drug-target interactions. In particular, we would like to focus on the AUPR because of the extremely small likelihood of a random drug-target interaction being positive. As we can see by the AUPR plot, all models continue to struggle with the recall rate.

The two plots above better demonstrate the quality of models, immediately making some more complex models, such as the Cross Attention model and the CNN, stand out for their low metrics. The MLP models seem to show the most potential as replacements for the baseline, however, it is visually difficult to distinguish between them.

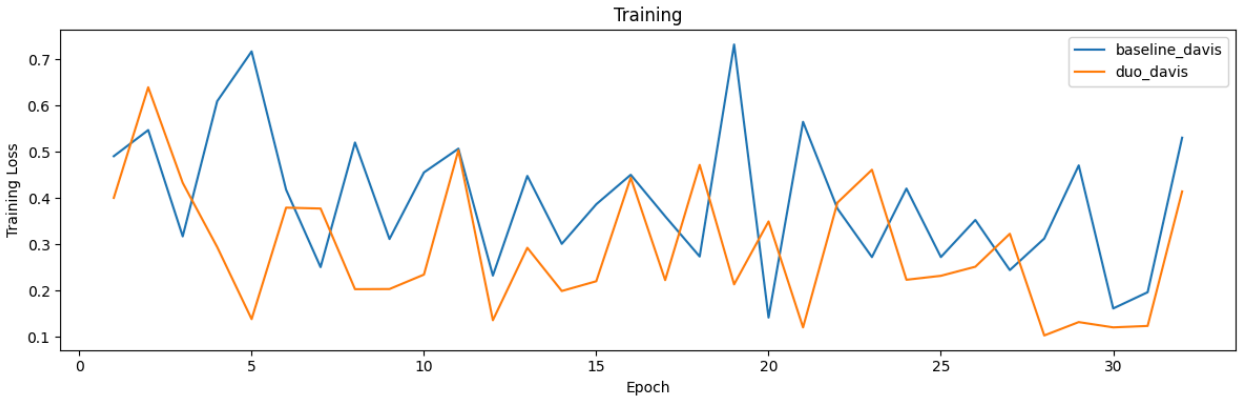
We continued on to evaluate the models on a separate sub-dataset of DAVIS, with many compounds to which the models had little exposure (as few-shot benchworks).

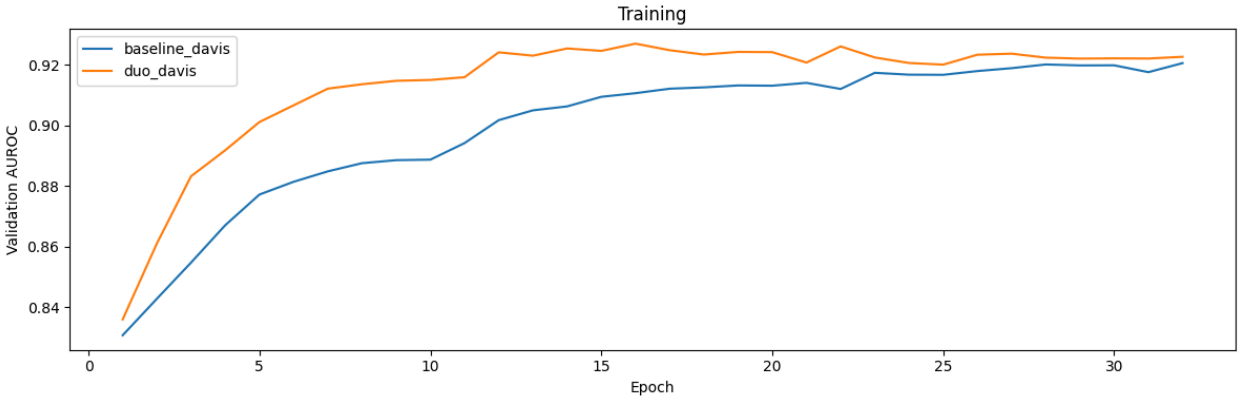


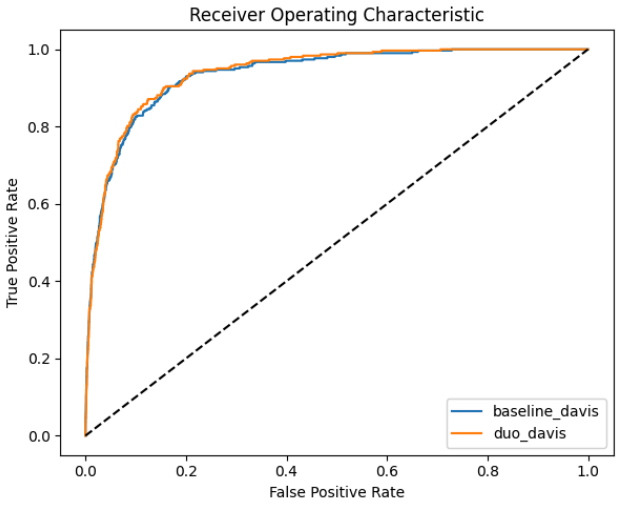
Again, it is hard to visually tell the models apart. So we outputted a table of values.

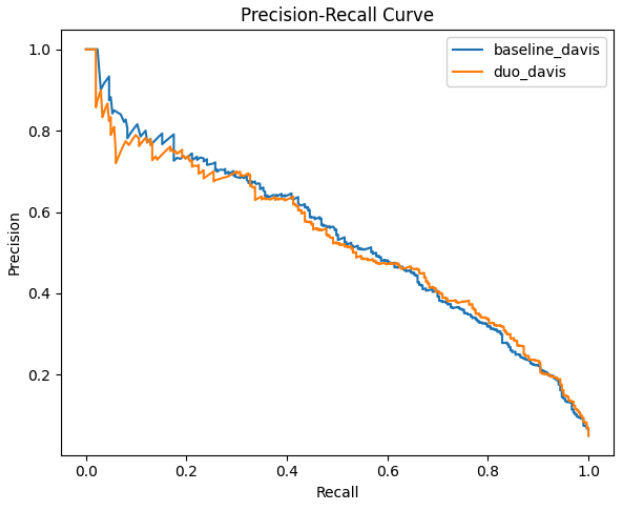
Model Architecture	***Screening AUROC***	***Screening AUPR***
duo_layer	*0.939993*	*0.5408*
large_layer	*0.937022*	*0.535805*
triple_layer	*0.93994*	*0.535398*
small_duo	*0.934399*	*0.517335*
baseline	*0.933034*	*0.516749*
quintuple	*0.916646*	*0.445599*
residual	*0.880454*	*0.368105*
CNN	*0.791915*	*0.19842*
cross_attention	*0.735064*	*0.142155*

Now, we can focus our attention to the Duo Layer Perceptron model in relation to the baseline. We were curious to see how much the model could improve if we let it train till it plateaued. So we repeated our experiment, this time with >30 epochs of training.









Model Architecture	***Testing AUROC***	***Testing AUPR***
baseline_davis	*0.935862*	*0.536214*
duo_davis	*0.939739*	*0.52909*

Despite some promising results with the training and validation datasets for our Duo Layer Perceptron model, the two showed near identical performances on the testing dataset. Our model showed marginally better AUROC and marginally worse AUPR. Both models showed very little improvement

despite over double the amount spent on training. Of note, the AUPR curve of the Duo Layer Perceptron model worsened under this training, indicating some potential overfitting.

7. Discussion and Conclusion

In the end, we were unable to provide strong evidence that our model is a significant improvement upon the embedding model within the ConPLex pipeline. However, our results suggest that a simple MLP may be further tuned into a better model, given sufficient computing resources for more experiments on the hidden layer size and activation functions.

During our project, we were limited by the time - less than a quarter - and computing resources available - personal laptops and a few lab desktops. For a more thorough study, our models should be tested on all the datasets provided by ConPLex, with less restrictions on the training lengths.

Regardless, the most significant insight from this project is that simplicity often outperforms complexity when working with high-quality pre-trained representations. The PLM embeddings used by ConPLex are already semantically rich, making simple transformative models effective at predicting certain behaviors such as binding. It appears that adding depth (through Deep MLP) or complex interaction mechanisms (with Cross-Attention) tends to disrupt this alignment rather than enhance it.

Therefore if we were to continue this project, we would be interested in exploring techniques like LoRA (Low-Rank Adaptation) to fine-tune the protein encoder, instead of using it as is. This would allow the model to learn task-specific features that might benefit from deeper projection heads.

8. *Individual Contributions*

Mohammadarshya Salehibakhs*: worked and trained on three different architectural models, Residual Coembedding, Deep Coembedding, and Cross Attention Coembedding and made graph visualizations for the project demo notebook. Also worked on improving the baseline model by tuning training parameters, despite results not being covered in the final report. Wrote early drafts of the proposal and report.*

Lucas Ueta*: provided the project idea from a research paper. Explored different avenues to access additional computing resources. Implemented original model. Trained CNN and variants of MLP models. Edited final proposal and report.*