Protein–Protein Interaction Prediction from Sequence Using Pretrained Protein Language Models

Abstract

Protein–protein interactions (PPIs) underpin essential cellular processes such as signaling, regulation, transport, and complex formation. Although experimental assays provide high-quality evidence, they are costly, time-consuming, and incomplete at proteome scale. This paper presents an end-to-end machine learning pipeline for predicting PPIs directly from amino-acid sequences. Each protein is encoded using a pretrained protein language model (ESM-2), producing fixed-length embeddings. Candidate protein pairs are represented using embedding concatenation and classified by a lightweight multilayer perceptron (MLP). To reduce information leakage and provide a rigorous estimate of generalization, we adopt a protein-disjoint split in which proteins in the test set do not appear in training. On a large-scale human PPI dataset derived from STRING (organism 9606), the model achieves test ROC-AUC = 0.8609 and PR-AUC = 0.8697, with F1 = 0.7645 (precision = 0.8235, recall = 0.7134) at a 0.5 threshold. We detail dataset construction, negative sampling, embedding extraction, training protocol (early stopping on validation PR-AUC), and evaluation. The complete source code, datasets, and instructions to reproduce all experiments are available at: https://github.com/ARUNSRINIVASAN12/protein-protein-interaction

Index Terms

protein–protein interaction, PPI prediction, protein language models, ESM-2, embeddings, machine learning, bioinformatics

I. Introduction

Protein–protein interactions are fundamental to cellular function. Complex formation enables enzymatic catalysis, scaffolding and signaling, transcriptional regulation, immune responses, and transport. A high-quality interactome supports disease mechanism discovery, pathway reconstruction, drug target identification, and systems biology modeling. However, experimental PPI discovery remains challenging: high-throughput assays can be noisy and biased, while targeted assays are expensive and slow. As a result, interaction databases remain incomplete and skewed toward well-studied proteins.

Computational prediction is valuable as a *prioritization layer*: it ranks candidate interactions so that laboratory resources can be focused on high-confidence hypotheses. Historically, PPI prediction relied on engineered features (sequence similarity, domains, co-expression, phylogenetic profiling, structural templates). More recently, transformer-based protein language models trained on massive unlabeled sequence corpora have enabled general-purpose representations that correlate with structure and function. This project builds a practical and reproducible PPI predictor that uses pretrained embeddings and a compact classifier, with a careful emphasis on evaluation methodology.

A. Problem Statement

Given two proteins A and B described by their amino-acid sequences, predict whether they interact. We frame this as supervised binary classification producing an interaction probability $\hat{y} \in [0, 1]$.

B. Contributions

- A complete sequence-only PPI prediction pipeline using pretrained ESM-2 embeddings and an MLP classifier.
- A protein-disjoint evaluation protocol to reduce information leakage across splits.
- Detailed experimental reporting with learning curves, PR/ROC curves, a confusion matrix, and discussion of operating points.

II. Related Work and Background

A. Classical PPI Prediction

Early methods leveraged sequence alignment, co-evolutionary couplings, gene neighborhood signals, and phylogenetic profiles. Feature-based machine learning introduced amino-acid composition, k-mer frequencies, and physicochemical encodings paired with SVMs or ensemble models. These approaches can work well on specific datasets but often require careful feature design and can struggle to generalize across organisms or data regimes.

B. Deep Learning and Protein Language Models

Deep learning approaches applied CNNs and RNNs to raw sequences, learning features end-to-end but typically requiring large labeled datasets. Protein language models (PLMs) pretrain on unlabeled sequences using self-supervised objectives, enabling transfer learning. The ESM family is widely used for embeddings that capture properties correlated with structure and function, providing a strong foundation for downstream prediction tasks including PPI inference.

TABLE I
DATASET SPLIT SUMMARY (PROTEIN-DISJOINT) FOR THE REPORTED RUN.

Split	#Pairs	#Positives	#Negatives
Train	97,647	48,682	48,965
Validation	4,439	—	—
Test	4,616	—	—

C. Evaluation Pitfalls

PPI benchmarks are particularly sensitive to split strategy. Pair-level splits allow the same protein to appear in training and test pairs, enabling models to learn protein-specific priors and inflating performance. Protein-disjoint splits are stricter and better reflect generalization to unseen proteins; we adopt this setting throughout.

III. Dataset Construction and Preprocessing

A. Data Source

Positive interactions are derived from the STRING database for *Homo sapiens* (organism 9606). We use high-confidence interactions as defined by our STRING extraction pipeline. Protein sequences are obtained from corresponding `FASTA` downloads and mapped to the proteins used in the interaction network.

B. Positive/Negative Pair Generation

Let $\mathcal{P}$ denote proteins with valid sequences and embeddings. Let $\mathcal{E}^+$ be the set of positive interaction edges after filtering. Negative edges $\mathcal{E}^-$ are sampled from $\mathcal{P} \times \mathcal{P}$ under constraints: (i) exclude self-pairs (A, A), and (ii) exclude any known positives in $\mathcal{E}^+$. We sample negatives to obtain a near-balanced training set for stable optimization.

C. Protein-Disjoint Splitting

To reduce leakage, we split proteins into train/validation/test sets and keep only pairs whose endpoints lie within the same split. This ensures that proteins in the test split do not appear in training or validation pairs, making the evaluation stricter and more realistic.

D. Dataset Summary

Table I summarizes the split sizes for the reported run.

E. Sequence Length Handling

Sequences are truncated to a maximum length of 1024 tokens during embedding computation to control compute and memory usage. This choice is common in PLM pipelines but may discard distal domains in long proteins; we discuss this limitation in Section VIII.

IV. Methodology

A. Pipeline Overview

Fig. 1 illustrates the full pipeline, from dataset construction and sequence embedding to pairwise classification and evaluation.

B. Protein Embeddings with ESM-2

Given a tokenized sequence, ESM-2 produces contextual token embeddings $\{\mathbf{h}_i\}_{i=1}^{L}$. We compute a fixed-length embedding by mean pooling:

$$\mathbf{e} = \frac{1}{L} \sum_{i=1}^{L} \mathbf{h}_i. \tag{1}$$

In our setup, $\mathbf{e} \in \mathbb{R}^{320}$ (mean-pooled embedding dimension).

C. Pairwise Feature Construction

For a candidate pair (A, B) with embeddings $\mathbf{e}_A$ and $\mathbf{e}_B$, we use concatenation:

$$\mathbf{x}_{A,B} = [\mathbf{e}_A; \mathbf{e}_B] \in \mathbb{R}^{640}. \tag{2}$$

Concatenation is a strong baseline with minimal assumptions. Common alternatives include absolute difference $|\mathbf{e}_A - \mathbf{e}_B|$ and Hadamard product $\mathbf{e}_A \odot \mathbf{e}_B$; we outline these as future work.

Algorithm 1 Training with early stopping on validation PR-AUC.

Compute and cache embeddings for all proteins using ESM-2.
Construct train/val/test pairs using a protein-disjoint split.
Initialize MLP parameters θ.
for epoch $= 1$ to $E_{\max}$ **do**
  Update θ using AdamW on mini-batches to minimize $\mathcal{L}$.
  Evaluate validation metrics; track validation PR-AUC.
  if validation PR-AUC improves **then**
    Save checkpoint $\theta^\star$; reset patience counter.
  else
    Increment patience counter.
  end if
  if patience counter exceeds limit **then**
    Stop training.
  end if
end for
Evaluate the best checkpoint $\theta^\star$ on the test set and report metrics.

TABLE II
TEST PERFORMANCE ON THE PROTEIN-DISJOINT SPLIT.

Metric	Value
ROC-AUC	0.8609
PR-AUC	0.8697
F1-score	0.7645
Precision	0.8235
Recall	0.7134

D. Classifier and Loss

We train an MLP that outputs a logit z and probability $\hat{y} = \sigma(z)$, where σ is the sigmoid. We optimize a weighted binary cross-entropy loss:

$$\mathcal{L} = -w^+ y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}), \tag{3}$$

with positive-class weight $w^+ \approx 1.006$ computed from the training distribution.

E. Training with Early Stopping

We select the best checkpoint based on validation PR-AUC (a prioritization-relevant metric), then report test performance.

V. EXPERIMENTAL SETUP

A. Implementation Details

Embeddings are extracted with ESM-2 using mean pooling and a maximum length of 1024 tokens. The classifier is trained with AdamW (learning rate 10^{-3}), dropout (as configured), and early stopping with patience 3 on validation PR-AUC. Embeddings are cached to accelerate experimentation.

B. Evaluation Metrics

We report ROC-AUC, PR-AUC, F1, precision, and recall. Since PPI datasets are often imbalanced in realistic settings, PR-AUC is emphasized for ranking and prioritization. We also report a confusion matrix at a decision threshold of 0.5 for interpretability.

C. Learning Curves

The best validation PR-AUC occurs at epoch 12 with val PR-AUC = 0.8538 and val ROC-AUC = 0.8551.

VI. RESULTS

A. Primary Test Metrics

Table II reports test-set performance under the protein-disjoint protocol.

At a 0.5 threshold, the confusion matrix counts are: TP=1670, FP=358, TN=1917, FN=671. Precision exceeds recall, which is desirable for prioritization scenarios where validating false positives is costly.

TABLE III
HYPERPARAMETERS USED IN THE REPORTED RUN (EDIT AS NEEDED).

Parameter	Value
Embedding model	ESM-2 (mean pooling)
Embedding dimension	320
Max sequence length	1024
Pair feature	Concatenation
Optimizer	AdamW
Learning rate	10^{-3}
Class weight (w^+)	1.006
Early stopping	patience = 3 on val PR-AUC
Decision threshold	0.5

B. ROC and Precision–Recall Curves

C. Confusion Matrix

D. Operating Point Discussion

The optimal decision threshold depends on the downstream objective. If the goal is to propose a small set of high-confidence interactions, a threshold that maximizes precision (or precision@K) may be preferable. If coverage matters more, a lower threshold can increase recall at the expense of more false positives.

VII. DISCUSSION

Pretrained embeddings likely encode signals related to evolutionary constraints and functional motifs. The strong test PR-AUC suggests the model effectively prioritizes true interactions above sampled negatives even under a protein-disjoint split. This is encouraging for realistic use cases where interactions must be predicted for unseen proteins.

VIII. LIMITATIONS

Random negatives may be "easy" and can inflate performance relative to harder negatives (e.g., degree-matched or pathway-matched sampling). Database labels can contain noise and may be incomplete, and some sampled negatives may correspond to undiscovered positives. Finally, sequence truncation to 1024 tokens may remove interaction-relevant domains for long proteins.

IX. FUTURE WORK

Ablate alternative pair representations (abs-diff, Hadamard, combined features) and evaluate with the same protein-disjoint split. Adopt harder negative sampling and report precision@K to better reflect prioritization settings. Optional extensions include probability calibration and incorporating structural priors.

APPENDIX A
REPRODUCIBILITY CHECKLIST

Include: dataset source (STRING organism 9606), confidence threshold, negative sampling procedure, split strategy and seed, embedding model/pooling/max length, classifier architecture, optimizer/hyperparameters, and all plots referenced in the text.

APPENDIX B
HYPERPARAMETERS

ACKNOWLEDGMENT

This report was prepared as a course project submission.

REFERENCES

[1] D. Szklarczyk *et al.*, "STRING: functional protein association networks," *Nucleic Acids Research*.
[2] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *PNAS*, 2021.

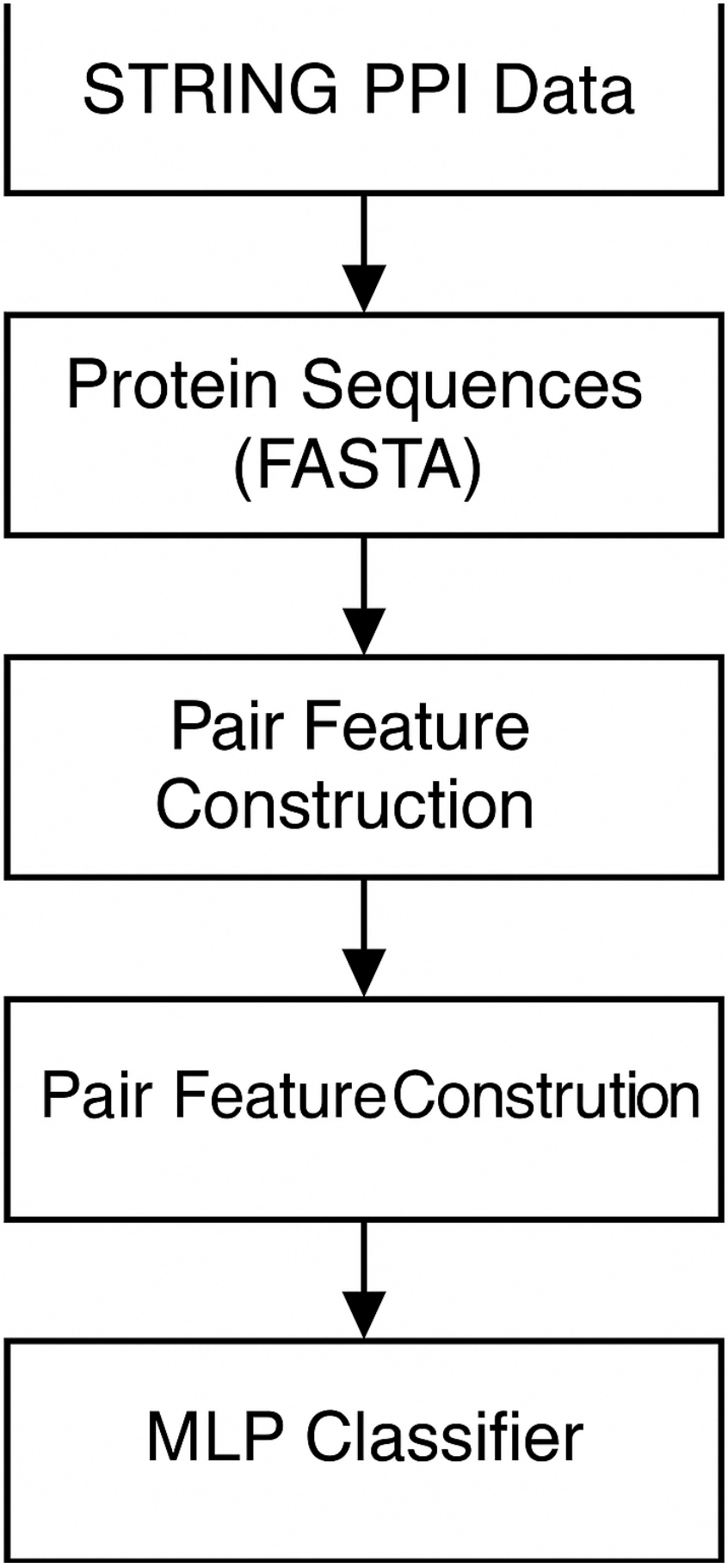

Fig. 1. End-to-end pipeline for protein–protein interaction prediction. Sequences are embedded using ESM-2; pairwise features are constructed and passed to an MLP classifier to predict interaction probability.

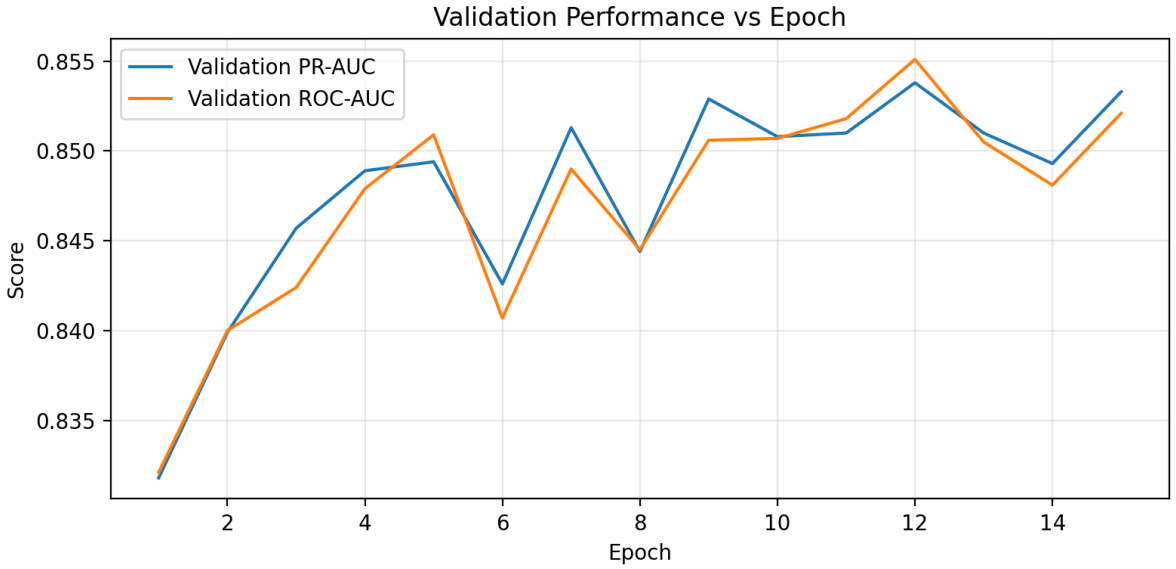

Fig. 2. Validation ROC-AUC and PR-AUC versus epoch for the reported run. Best validation PR-AUC occurs at epoch 12.

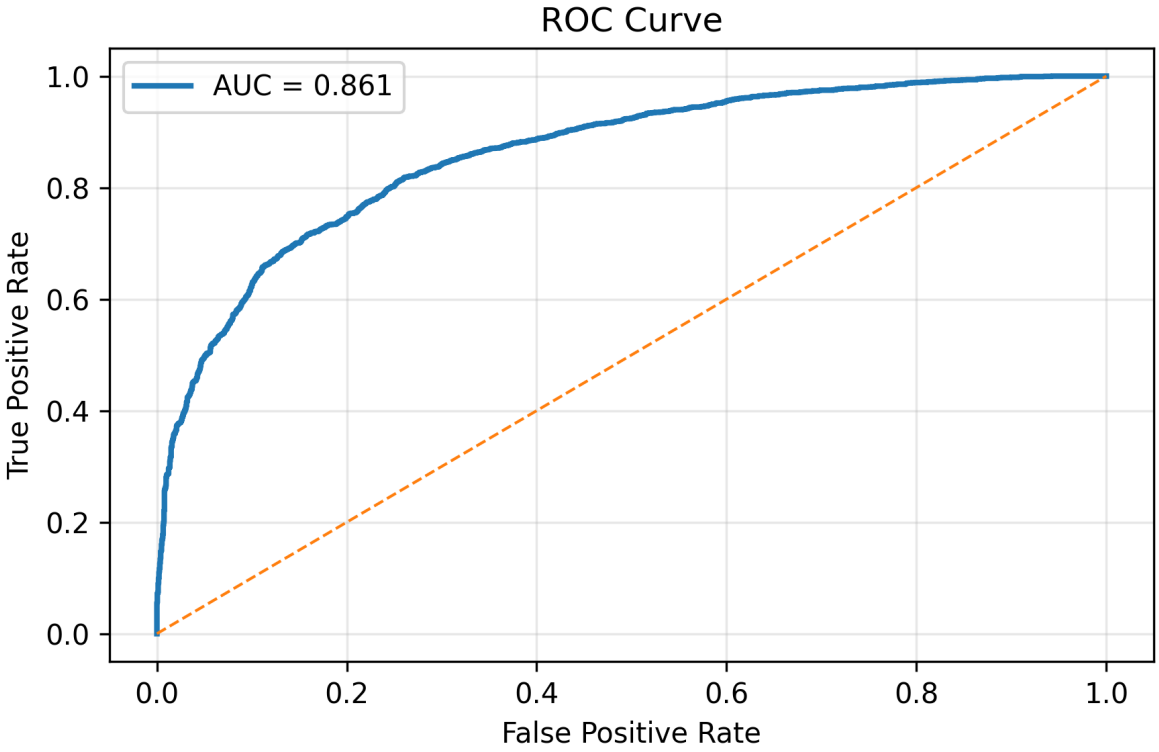

Fig. 3. ROC curve on the protein-disjoint test set.

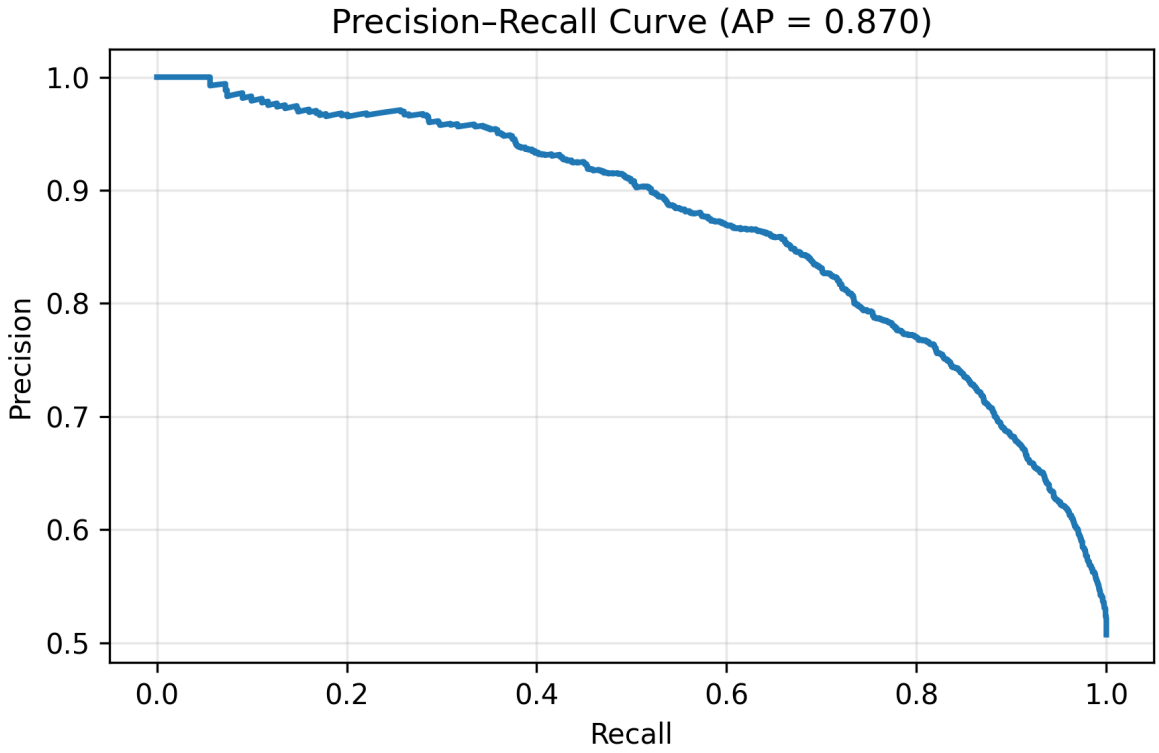

Fig. 4. Precision–Recall curve on the protein-disjoint test set.

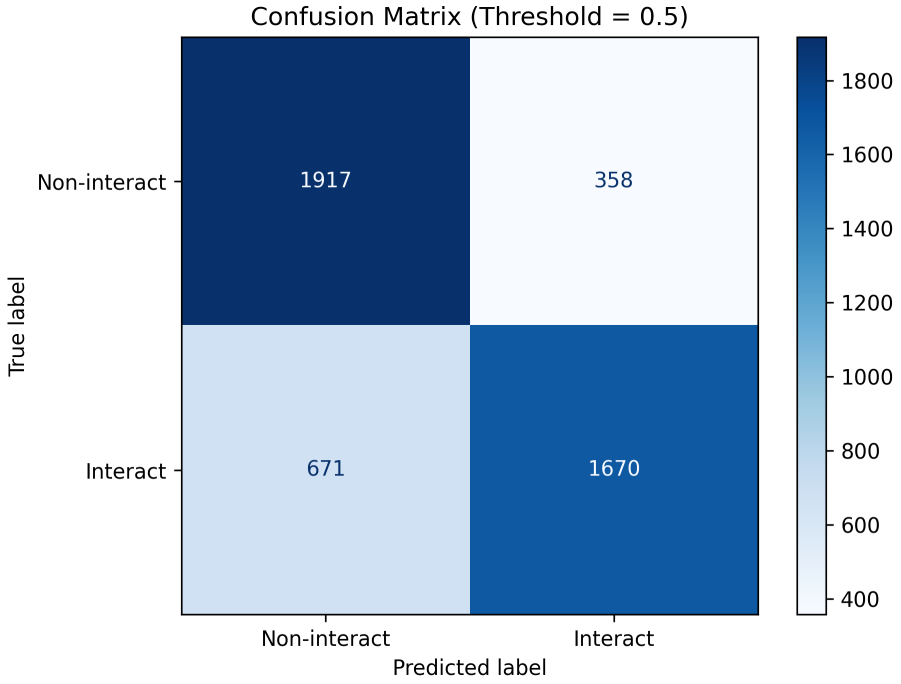

Fig. 5. Confusion matrix on the test set at decision threshold 0.5.