Improving PPI Prediction with ESM-derived Features

戴佳文 齐奕婷 王安瑞

1. Feature Extraction

- Utilized **ESMC** for protein sequence embedding
    - **ESM Cambrian** focuses on creating representations of the underlying biology of proteins.
    - Old methods use biological annotations or only protein sequences
- Technical Details:
    - Input: Protein sequences from the `b4ppi` dataset
    - Model: ESM-C pre-trained transformer
    - Output: Embeddings with shape $[L+2, 960]$, where L represents sequence length and 960 is the embedding dimension.(other 2 are [CLS], [EOS] token)

2. Data Processing

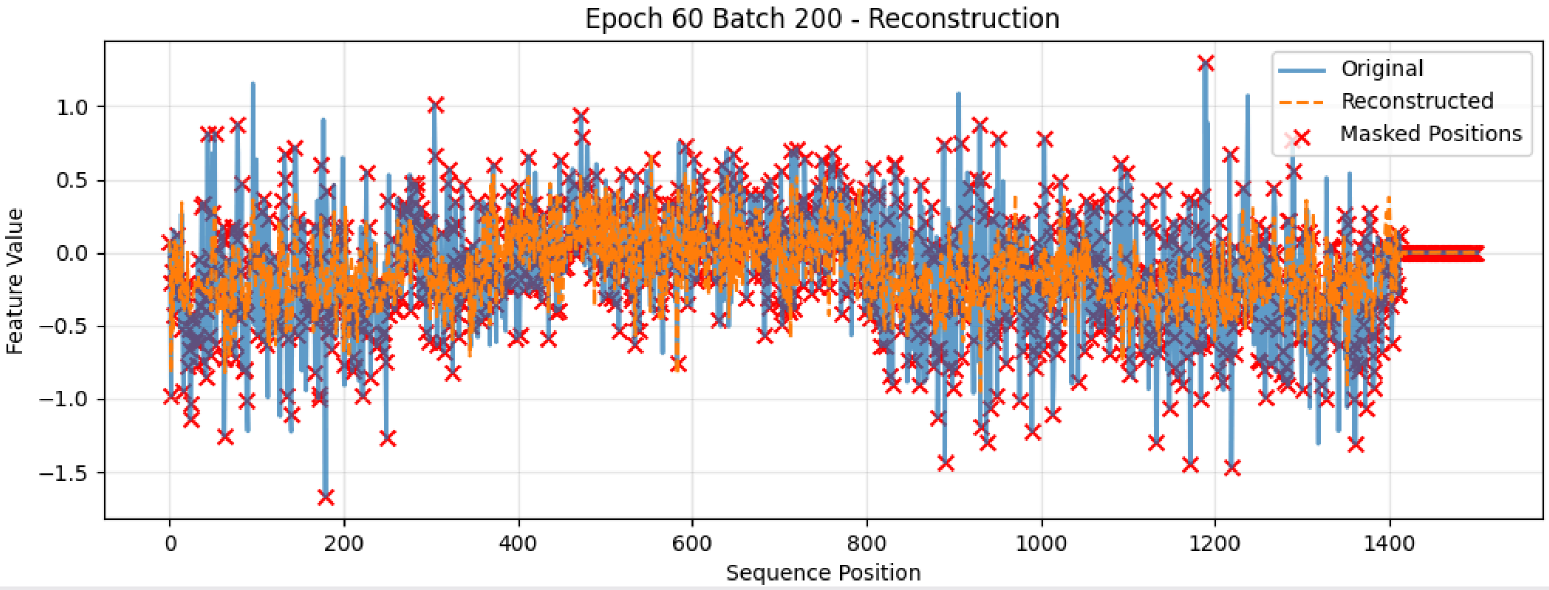
To enable downstream classification, we explored multiple pooling strategies to transform variable-length embeddings [L, 960] into fixed-size representations [960].

Initial Approach: average pooling/max pooling

- Applied **average pooling** along the sequence dimension `L+2` to:
  - Handle variable-length inputs uniformly
  - Reduce computational complexity (the embedding matrix is too big)
  - Preserve channel-wise information
- Resulting compressed embedding: `[1, 960]` per protein
- Lose important positional and structural information

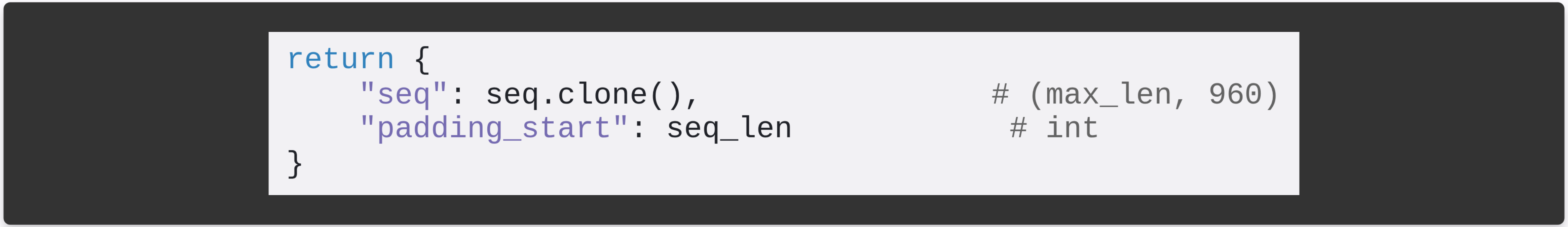
Improved Approach: Masked Autoencoder (MAE)

- Length Standardization = 1502
  - Pad shorter sequences with zeros (`[PAD] = 0`)
  - Truncate longer sequences
  - 75% mask
- Problems:
  - data with small variances 0.0002-0.001 and small loss ----> multiply a scale_factor
  - mask padding positions ----> calculate the loss only for non-padding positions


Epoch 60 Batch 200 - Reconstruction
Original
Reconstructed
Masked Positions
Feature Value
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
200
400
600
800
1000
1200
1400
Sequence Position

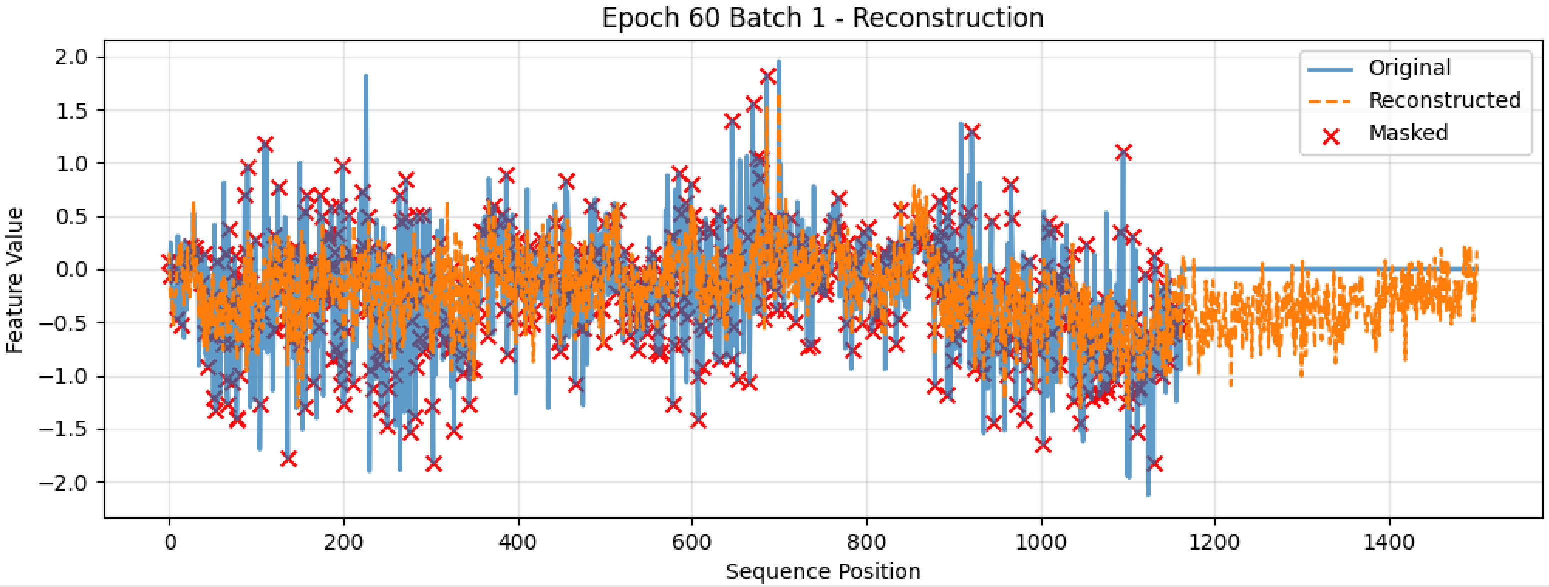
Key Improvement:

- Added **padding position markers** for each protein sequence



```
return {
    "seq": seq.clone(),                # (max_len, 960)
    "padding_start": seq_len            # int
}
```

- Modified masking to only operate on **non-padding regions**.
  - 50% mask


Epoch 60 Batch 1 - Reconstruction
Feature Value
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
200
400
600
800
1000
1200
1400
Sequence Position
Original
Reconstructed
Masked

other modifications:

- We first concatenated the two features and then applied compression; because the data volume was too large, we truncated the length of the concatenated protein embeddings to 2000.
- Use the CLS token added by ESMC as the compressed embedding.

Downstream Supervised Learning Models

We evaluated multiple classification algorithms to determine the optimal approach for PPI prediction.

- **Dataset Characteristics**:
  - Test1 dataset is a balanced dataset with 50% positive and 50% negative samples.
  - Test2 dataset presents a realistic challenge with 9.09% positive and 90.91% negative samples, closely mimicking real-world PPI prediction scenarios

L2-cos Method

Similarity	Dataset	Best AUROC
L2	Test1	**0.5547**
L2	Test2	**0.5718**
Cosine	Test1	0.5330
Cosine	Test2	0.5676

- High similarity between proteins does not necessarily imply that they have a PPI.

Logistic Regression

1. **As baseline**: Tests linear separability of features
2. **Computational efficiency**:
   - Fast training
   - Easy hyperparameter tuning (e.g., regularization strength `C`)
3. **Explainability**: Direct biological interpretation of feature contributions

Method	Dataset	AUROC
L2 Similarity	Test1	0.5547
L2 Similarity	Test2	0.5718
Logistic Regression	Test1	**0.6182**
Logistic Regression	Test2	**0.6500**

Support Vector Machine (SVM)

- Finds optimal decision boundary with regularization
- Use different kernels to capture non-linearities:

```
"svm__kernel" : ['linear', 'poly', 'rbf', 'sigmoid']
# Handles linear/nonlinear cases```
```

- Effective in feature-rich, sample-limited scenarios

Method	Dataset	AUROC
Logistic Regression	Test1	0.6182
Logistic Regression	Test2	**0.6500**
SVM	Test1	**0.6285**
SVM	Test2	0.6201

XGBoost

Nonlinear feature interactions:

- Discovers complex patterns in 960-D space

Method	Dataset	AUROC
Logistic Regression	Test1	0.6182
Logistic Regression	Test2	**0.6500**
XGBoost	Test1	**0.6441**
XGBoost	Test2	0.6480

Multi-Layer Perceptron (MLP)

1. The encoder of MAE is essentially a multi-layer fully connected structure, and using MLP as the downstream classification layer can naturally connect. The encoder and MLP can be merged into a large end-to-end model.
2. Captures higher-order interactions than XGBoost
3. A unified backpropagation pipeline can be useful to fine-tune both the MAE encoder and MLP

Reference: "Pitfalls of machine learning models for protein–protein interaction networks"

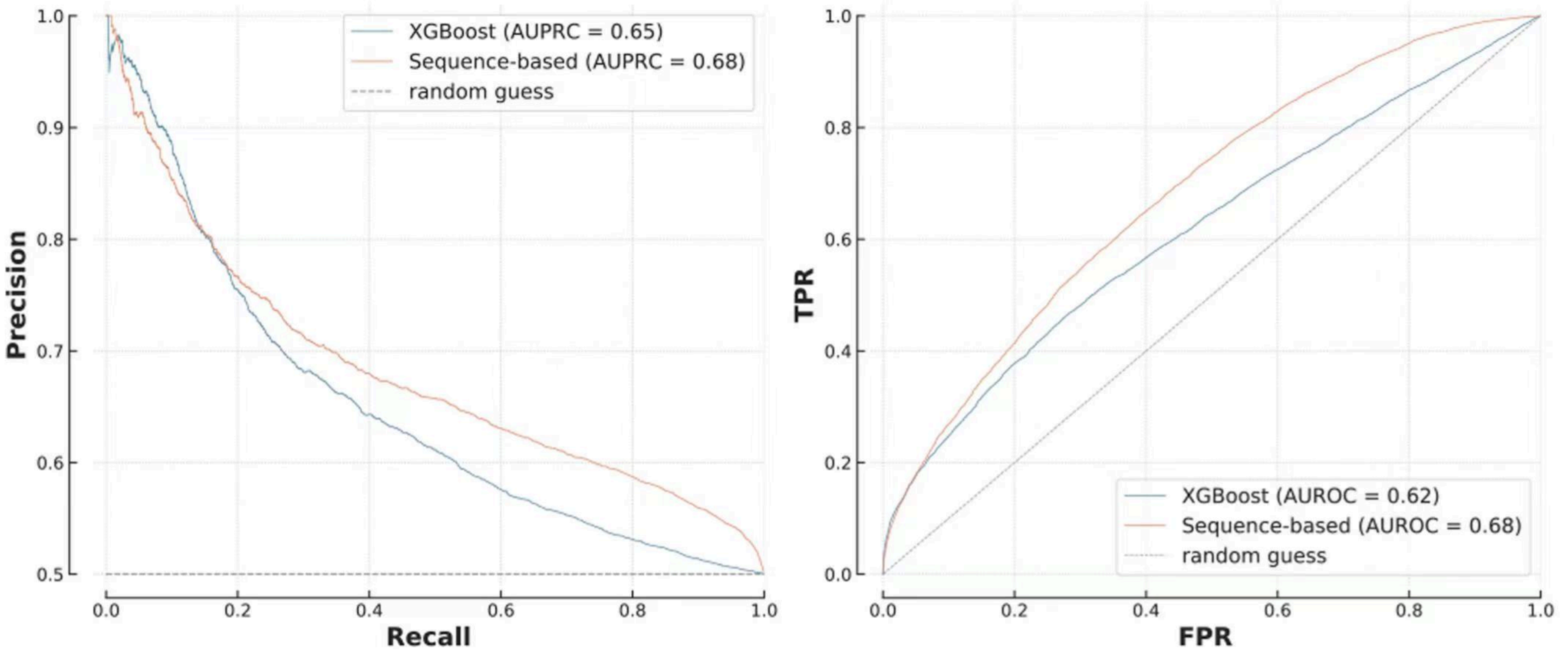
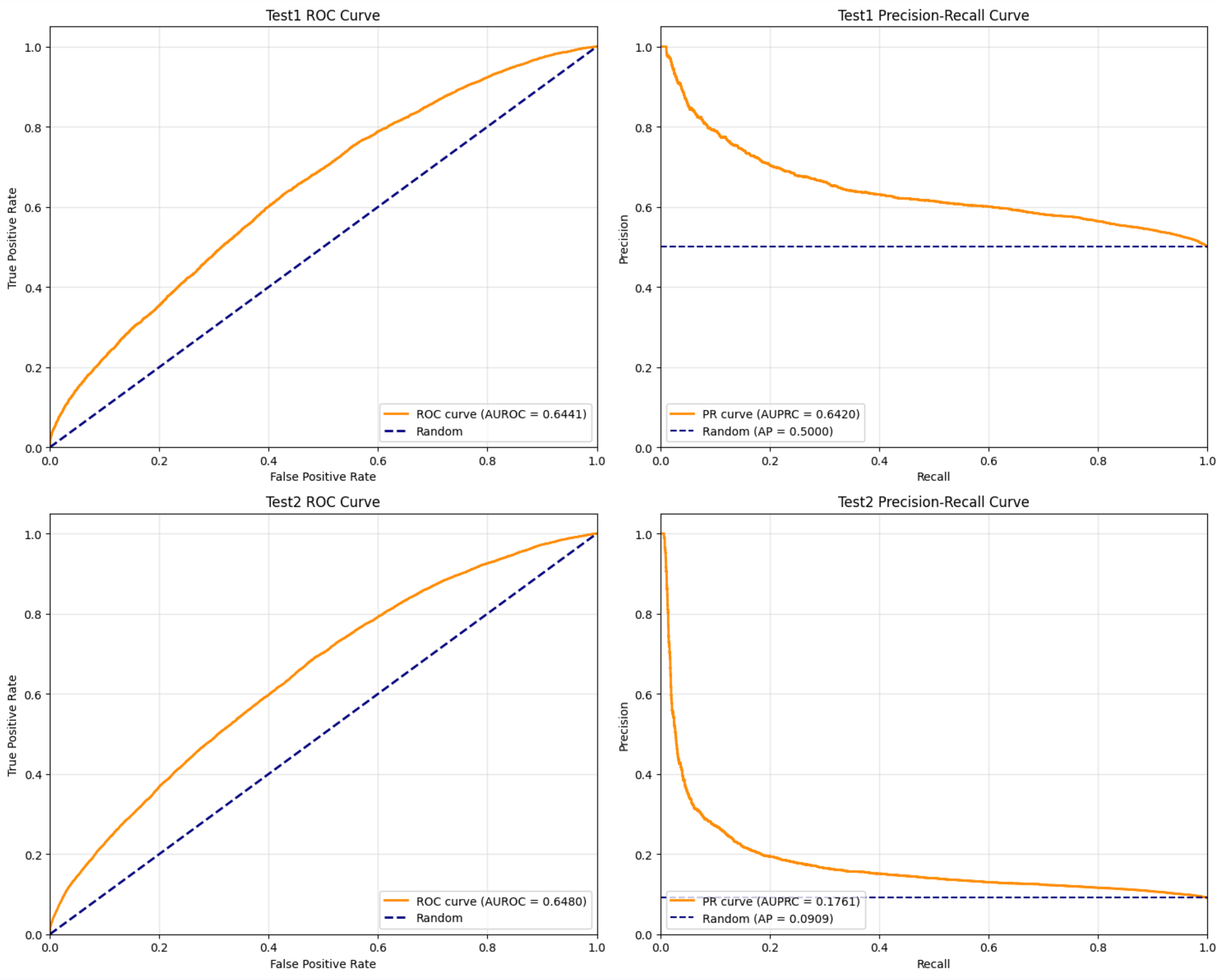


Figure 3. Siamese network versus XGBoost on *T1*. The difference between the ROC curves was statistically significant ($P = 9 \times 10^{-47}$).

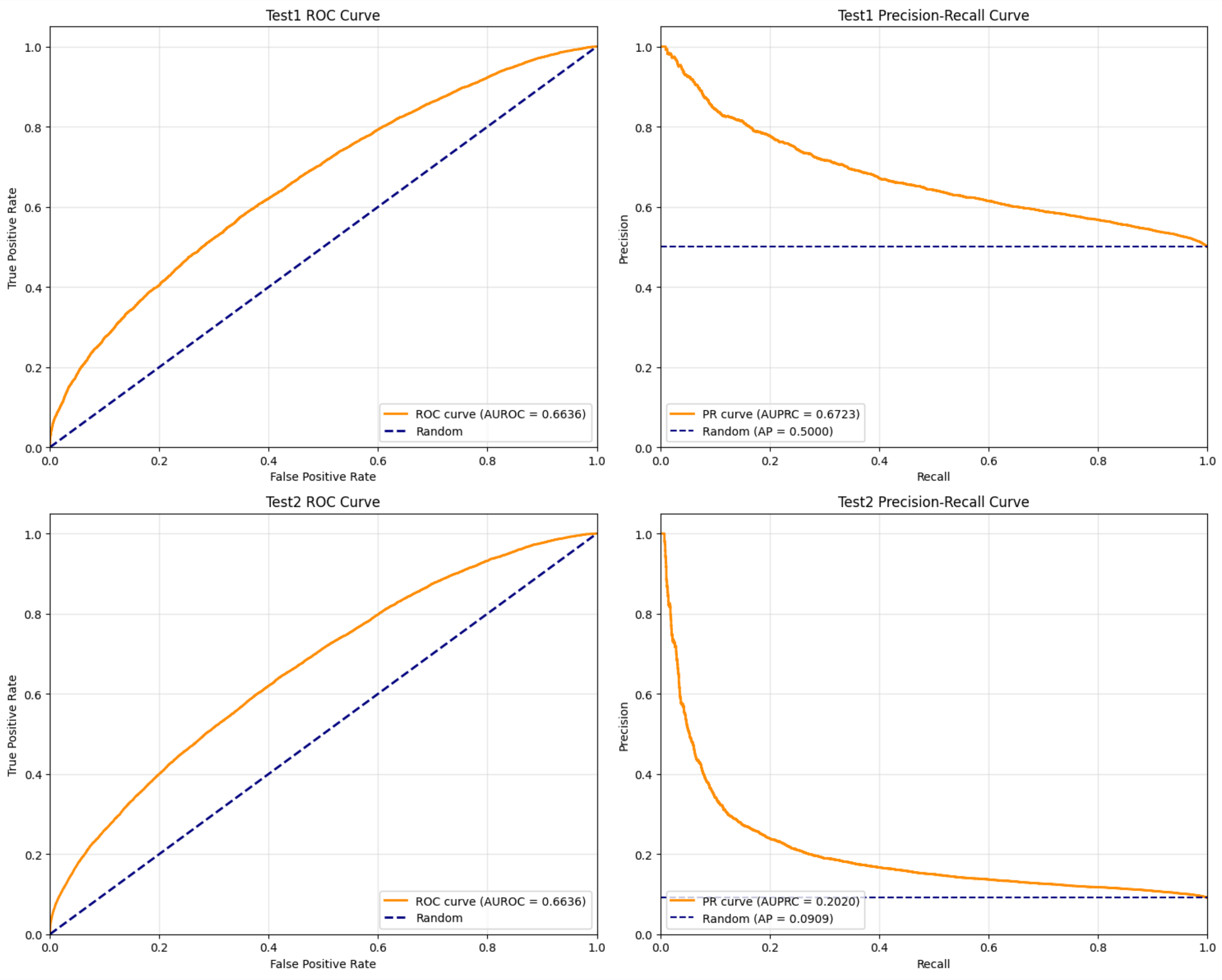


Test1 ROC Curve

Test1 Precision-Recall Curve

Test2 ROC Curve

Test2 Precision-Recall Curve


Test1 ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUROC = 0.6636)
Random
Test1 Precision-Recall Curve
Precision
Recall
PR curve (AUPRC = 0.6723)
Random (AP = 0.5000)
Test2 ROC Curve
True Positive Rate
False Positive Rate
ROC curve (AUROC = 0.6636)
Random
Test2 Precision-Recall Curve
Precision
Recall
PR curve (AUPRC = 0.2020)
Random (AP = 0.0909)

Final Comparison Table (Test1 Only)

Method	Test Set	AUROC	AUPRC
Sequence-based (RNN, from literature)	Test1	**0.68**	**0.68**
XGBoost (from literature)	Test1	0.62	0.65
XGBoost + AvgPooling	Test1	0.6441	0.6447
	Test2	0.6480	0.1761
XGBoost + Masked Autoencoder (MAE)	Test1	0.6466	**0.6771**
	Test2	**0.6636**	**0.2020**

Conclusion

- **Competitive Results**: Our XGBoost + MAE approach outperforms literature XGBoost baselines, but achieves slightly lower AUROC than sequence-based RNN models from literature.
- **XGBoost Superiority**: XGBoost demonstrates the best performance among traditional ML classifiers, effectively capturing nonlinear feature interactions

Future Directions

- End-to-end fine-tuning of MAE encoder with downstream classifiers(MLP)
- Exploration of more sophisticated attention mechanisms