Predicting ncRNA Interactions with Deep Learning

Valentina Debbia - 74280A

December 15, 2025

Abstract

The prediction of functional interactions between non-coding RNA molecules is a significant challenge in computational biology and genomics. This project develops a deep learning framework to predict whether specific long non-coding RNAs (lncRNAs) interact with pseudogene-derived RNAs (pseudoRNAs), using only their nucleotide sequences.The approach integrates RNA-FM, a foundation model for RNA sequence representation, with a feed-forward neural network for binary classification. This approach provides an effective computational tool for identifying candidate RNA-RNA interactions, guiding subsequent experimental validation in the study of gene regulatory networks.

1 Introduction

Understanding the role of RNA in the cellular context has undergone a radical transformation in recent decades. Although RNA was initially considered mainly an intermediary between DNA and proteins, it is now widely recognized that more than 90% of the genome is transcribed, whereas less than 2% encode proteins. [1] This indicates that most transcribed genes produce non-coding RNAs (ncRNAs).

The term ncRNA refers to those RNA molecules that, although they do not have the ability to encode proteins [2], are involved in complex regulatory mechanisms and fundamental cellular processes such as proliferation, differentiation, apoptosis, and maintenance of homeostasis. The malfunction of these molecules can contribute to the onset of diseases, making the study of their functions and interaction networks essential.

Current knowledge on the roles of cRNAs has been obtained primarily through experimental approaches. However, techniques such as RNA immunoprecipitation or crosslinking-based methods are costly, time-consuming, and low-throughput. In this context, computational biology represents a valuable complement to traditional experimentation: predictive methods can facilitate the identification of potential RNA-RNA or RNA-protein interactions, which can subsequently be validated through in vitro or in vivo experiments.

This integration gives rise to a process of mutual enrichment in which experimental data continuously inform the development of computational models, while predictions generate new hypotheses that can be experimentally tested.

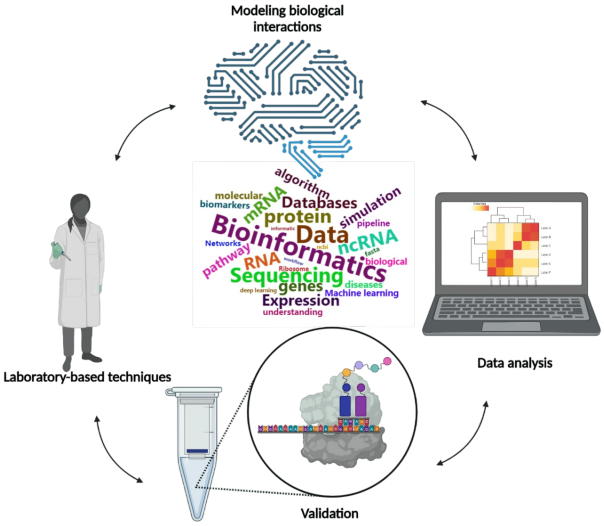


Figure 1: Iterative process diagram

In the field of non-coding RNA interaction prediction, several studies have developed computational frameworks to infer interactions between lncRNAs and other classes of non-coding RNAs, including microRNAs (miRNAs), circular RNAs (circRNAs), and other lncRNAs. These methods employ a variety of approaches, ranging from thermodynamic modeling of RNA secondary structures to machine-learning techniques that integrate sequence features, structural predictions, and evolutionary conservation signals.

For example, lncRNA-miRNA interactions have been characterized by showing how deep representations of sequences and their potential contact points can capture patterns relevant to the interaction propensity. Recent methods, such as those based on Transformer Encoders combined with CNNs [3], exploit these multichannel representations to identify informative signals without relying on handcrafted descriptors. Similarly [4], studies on lncRNA–protein interactions have highlighted the effectiveness of machine-learning and deep-learning models, facilitated both by the high cost of traditional experimental approaches and by the ability of these models to learn directly from sequence data.

This project aims to develop a deep learning-based binary classification model capable of predicting interactions between pairs of RNA sequences, specifically between lncRNA and pseudoRNA. The lncRNAs are transcripts longer than 200 nucleotides, lacking coding potential but involved in multilayer regulation of gene expression. PseudoRNAs, on the other hand, are transcribed from genomic loci that have lost their protein-coding ability due to mutations, while still retaining potential regulatory roles.

The approach combines two key components: RNA-FM (RNA Foundation Model), a large-scale pretrained transformer trained on more than 23 million non-coding RNA sequences, and a feed-forward neural network (FFNN) specifically trained for interaction classification. This combination enables the exploitation of rich and contextual representations of sequences, improving predictive performance compared to traditional methods based on manual features.

2 Methods

This section provides a comprehensive description of the experimental methodology employed in this study. Each component is detailed to fully ensure reproducibility of the results.

2.1 Dataset

The dataset consists of interaction pairs between long non-coding RNAs (lncRNAs) and pseudogene-derived RNAs (pseudoRNAs). The raw data are stored in a CSV file (`lncrna_pseudo.csv`) containing paired RNA sequences, where each row represents a potential interaction between two RNA molecules identified by the columns `RNA_sequence_x` and `RNA_sequence_y`, along with their respective categorical labels `Category_x` and `Category_y`.

In accordance with standard machine-learning practice, the dataset was subsequently partitioned into training and testing subsets using a 90/10 split. The training subset was further divided, allocating 80% for model training and 20% for validation during model development.

2.2 Data Augmentation

Given the limited size of experimentally validated interaction datasets a comprehensive data augmentation strategy was implemented to artificially expand the training set while preserving biological plausibility. The augmentation scheme generated four variants for each original interaction pair:

1. Original pair: the unmodified sequence pair (r_i, r_j).
2. Swapped pair: the sequences exchanged (r_j, r_i), addressing the symmetric nature of interactions where partner order is arbitrary.
3. Reversed pair: both sequences individually reversed (r_i^F, r_j^F) (where F denotes reversal), simulating potential directional reading variations.

4. Swapped and reversed pair: the combination of both operations (r_j^F, r_i^F).

This augmentation strategy resulted in a four-fold expansion of the dataset, increasing both training and test sets proportionally. Importantly, augmentation was applied after the train-test split to prevent data leakage, ensuring that augmented variants of test samples did not appear in the training set.

2.3 Negative Sampling

Binary classification requires both positive examples (confirmed interactions) and negative examples (non-interacting pairs). Since exhaustive experimental screening of all possible non-interacting pairs is infeasible, negative examples were computationally generated at a ratio of 1:20 to positive samples, reflecting the realistic sparsity of biological interaction networks. Negative pairs were sampled exclusively from sequences present within each data split, explicitly excluding true positives, their augmented variants and self-interactions to ensure valid negative examples.

The negative sampling process was parallelized across 10 worker threads and conducted independently for the training and test sets to prevent information leakage and ensure an unbiased evaluation of generalization performance.

2.4 RNA-FM Encoder

Rather than operating directly on raw nucleotide sequences or engineering features manually, this study employed RNA-FM (RNA Foundation Model), a large-scale pretrained transformer-based model specifically designed to capture evolutionary, structural, and functional patterns in RNA sequences. The RNA-FM architecture consists of 12 stacked bidirectional transformer attention layers, allowing information to flow in both directions along the sequence.

Each nucleotide is tokenized individually and mapped to a 640-dimensional embedding space. When processing an RNA sequence of length L nucleotides, RNA-FM produces a hidden state matrix $\mathbf{X} \in \mathbb{R}^{L\times 640}$, where each row corresponds to a contextualized representation of one nucleotide position, influenced by all other positions through the self-attention mechanism.

To convert these variable-length embeddings into fixed-dimensional representations suitable for downstream classification, a dual pooling strategy was employed that captures complementary aspects of the sequence representation. Average pooling $e_{\text{avg}} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{x}_i$ captures the overall global characteristics and contextual information of the sequence, while max pooling $e_{\max} = \max_i \mathbf{x}_i$, which preserves the most salient positional features

at any position. These two 640-dimensional vectors were concatenated to form a single 1280-dimensional embedding for each RNA sequence. This approach leverages the strengths of both pooling operations. For RNA-RNA interaction classification, pair representations were constructed by concatenating the embeddings of both partner RNA sequences: $\mathbf{x}_{\text{pair}} = [\mathrm{e}_i; \mathrm{e}_j] \in \mathbb{R}^{2560}$. This concatenated representation serves as the input to the feed-forward neural network classifier and implicitly assumes that relevant interaction features can be learned from the juxtaposition of the two individual representations.

2.5 FFNN

The core prediction model is a feed-forward neural network (FFNN) that maps the 2560-dimensional paired RNA embeddings to interaction probabilities.

A relatively deep architecture was designed to enable the learning of complex, hierarchical patterns in the embedding space, through which true interactions are distinguished from non-interactions. The baseline architecture consists of an input layer followed by four hidden layers of 1024 neurons each, with ReLU activation functions to enable learning of complex decision boundaries. To mitigate overfitting, dropout regularization (rate = 0.2) was applied after each hidden layer. The final output layer uses sigmoid activation to produce interaction probability scores between 0 and 1. In total, this configuration contains approximately 9.4 million trainable parameters. The model was trained using the Adam optimizer with binary cross-entropy loss, a popular adaptive learning rate method that has become standard in deep learning applications. The learning rate schedule employed cosine annealing with linear warmup, starting from an initial value and decaying to a minimum of 5×10^{-5} to enable larger updates early in training and finer adjustments near convergence. To address the severe class imbalance in the dataset, a balanced batch sampler was implemented to construct mini-batches containing 30% positive and 70% negative examples, thereby preventing bias toward the majority class. Early stopping with a patience of 10 epochs was employed based on validation loss to prevent overfitting, with the best-performing model weights being retained.

Before feeding the embeddings into the neural network, standardization was applied, which transforms each feature to have zero mean and unit variance. Critically, the scaling parameters were computed solely from the training data and subsequently applied to the validation and test sets. This procedure prevents information leakage and ensures that evaluation on the test set reflects true generalization performance.

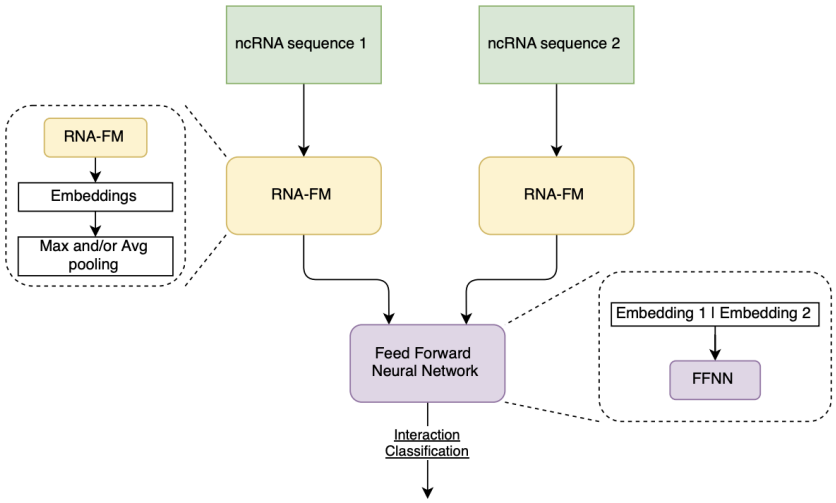


Figure 2: Architecture used

2.6 Hyperparameter Exploration

While the baseline configuration (learning rate = 0.0005, 4 hidden layers of 1024 neurons, dropout = 0.2, batch size = 512, 4-epoch warmup) described above represents the primary model, it is well known that the optimal architecture and training hyperparameters are not always immediately obvious. Therefore, a systematic exploration of the hyperparameter space was conducted to evaluate whether alternative configurations could improve performance.

The first alternative, which is called "`Deeper_HighDropout`", explored whether a deeper network with stronger regularization could capture more complex patterns. This configuration increased the number of hidden layers from 4 to 6 and raised the dropout rate to 0.3, while slightly reducing the learning rate to 0.0003 to account for the increased model capacity. The longer warmup period of 5 epochs was intended to help the larger model stabilize during initial training.

In the second configuration, "`Wider_LowLR`", the possibility that increasing the width of each layer rather than the depth might be beneficial was investigated. The hidden dimension was doubled from 1024 to 2048 neurons per layer while the original depth of 4 layers was maintained. To prevent instability associated with this much larger model, which contains significantly more parameters, the learning rate was reduced to 0.0001 and the warmup phase was extended to 6 epochs.

Conversely, the "`Smaller_HighLR`" configuration tested whether a more compact model could indeed generalize better, particularly if the RNA-FM embedding features are already highly informative. This configuration reduced the network to just three hidden layers with 512 neurons each and lowered the dropout rate to 0.15. With fewer parameters to

optimize, the learning rate was increased to 0.001 and the batch size was reduced to 256, hypothesizing that this lighter architecture could train faster and avoid overfitting.

Finally, the "`Balanced`" configuration represented a middle ground, with 5 layers of 1536 neurons each-intermediate between the baseline and the wider configuration. The learning rate of 0.0002, dropout of 0.25, and batch size of 384 were all chosen to balance between the extremes of the other configurations.

2.7 Metrics

By training and evaluating models with all five configurations, an empirical determination was made of which architectural choices and hyperparameter settings were best suited to this specific problem, rather than relying solely on default values or intuition.

Model performance was assessed using multiple complementary metrics:

- Accuracy: $\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$
- Balanced accuracy: $\text{BAcc} = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$
- Precision: $\text{Prec} = \frac{TP}{TP+FP}$
- Recall: $\text{Rec} = \frac{TP}{TP+FN}$
- F1 score: $\text{F1} = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$
- AUROC, which means Area Under the ROC Curve, measures discriminative ability across all thresholds
- AUPRC, which is Area Under the Precision-Recall Curve, particularly informative for unbalanced data sets.

Finally, normalized confusion matrices were calculated to visualize the true positive rate, true negative rate, false positive rate, and false negative rate.

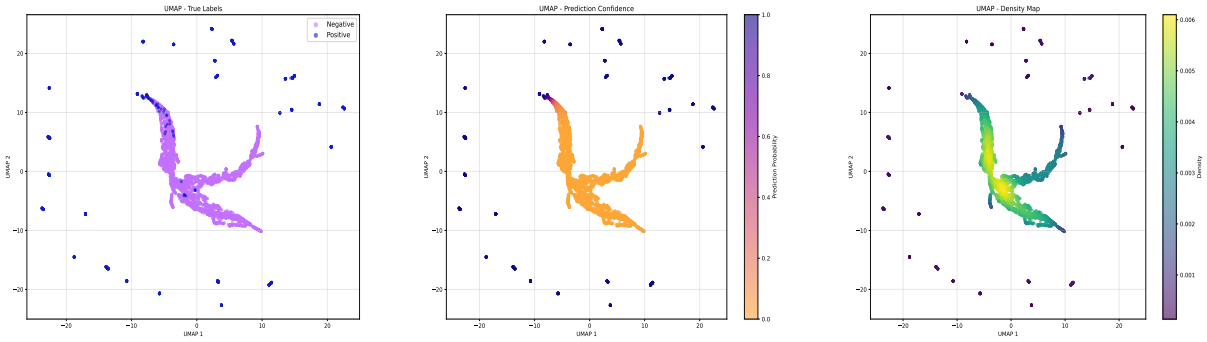
2.8 Embedding Visualization

To gain insight into the patterns learned by the neural network from the RNA-FM embeddings, dimensionality reduction and visualization of the learned representations were performed. Specifically, activations were extracted from the second-to-last layer of the trained network. This layer captures the high-level features used by the model to make its final classification decision.

Since direct visualization of a 1024-dimensional space is not feasible, two complementary dimensionality reduction techniques were applied to project these embeddings into two dimensions. The first method, UMAP (Uniform Manifold Approximation and Projection),

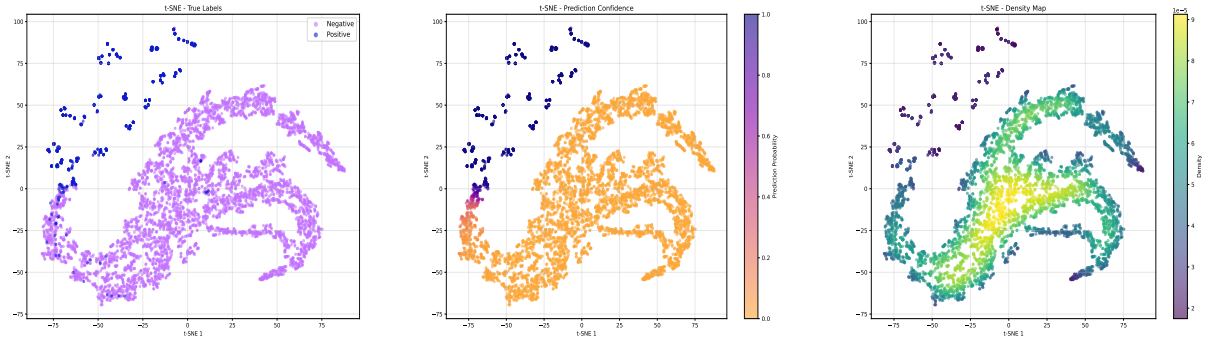
is a modern technique designed to preserve both local and global structure in the data. The second method, t-SNE (t-Distributed Stochastic Neighbor Embedding), is an older but well-established technique that prioritizes the preservation of local neighborhoods. These 2D projections allow the organization of RNA pairs within the model's internal representation space to be visualized.

Several visualizations were generated: scatter plots colored by true interaction labels to assess class separability, plots colored by prediction confidence (probability) to identify regions of high and low certainty, and density plots to reveal the distribution of training examples across the embedding space.



(a) (b) (c)

Figure 3: UMAP visualizations of the model's embedding space: (a) distribution of true interaction labels, (b) model prediction confidence scores, and (c) density of training examples across the embedding space.



(a) (b) (c)

Figure 4: t-SNE visualizations of the model's embedding space: (a) distribution of true interaction labels, (b) model prediction confidence scores, and (c) density of training examples across the embedding space.

The density visualization, computed using Gaussian kernel density estimation, highlights where the model has encountered many similar examples during training versus regions where it must extrapolate from sparse data.

By examining density alongside prediction confidence, it becomes possible to distinguish between well-supported predictions, which are marked by high confidence backed by many training examples, and potentially unreliable extrapolations, which occur when high confidence is expressed in low-density regions.

3 Results

An initial exploratory analysis of the dataset revealed a total of 1369 unique RNA sequences participating in 6967 interaction pairs. The sequence length distribution is heavily right-skewed, with the majority of sequences being relatively short (predominantly under 2000 - 3000 nucleotides), though a small number of substantially longer sequences were also observed.

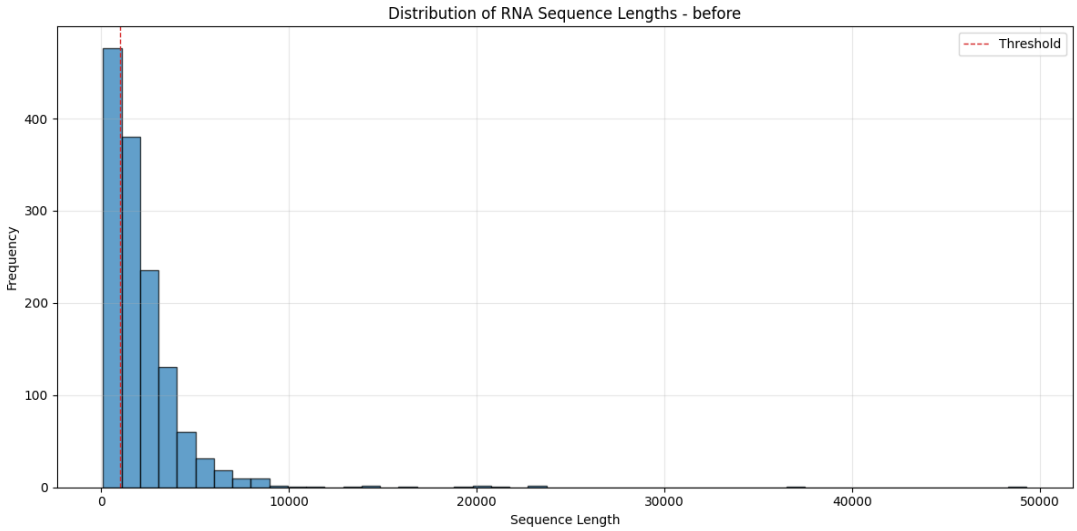


Figure 5: Dataset distribution before filtering

To ensure compatibility with the selected model and maintain computational efficiency, a sequence length filtering threshold of 1022 nucleotides was applied, removing sequence pairs where either partner exceeded this limit. This threshold reflects both the maximum input capacity of the RNA foundation model and the empirical length distribution, balancing data retention with architectural constraints.

After applying the sequence length threshold of 1022 nucleotides, the dataset was reduced to 131 unique RNA sequences. This filtering step significantly narrowed the dataset to sequences fully compatible with the model's input requirements.

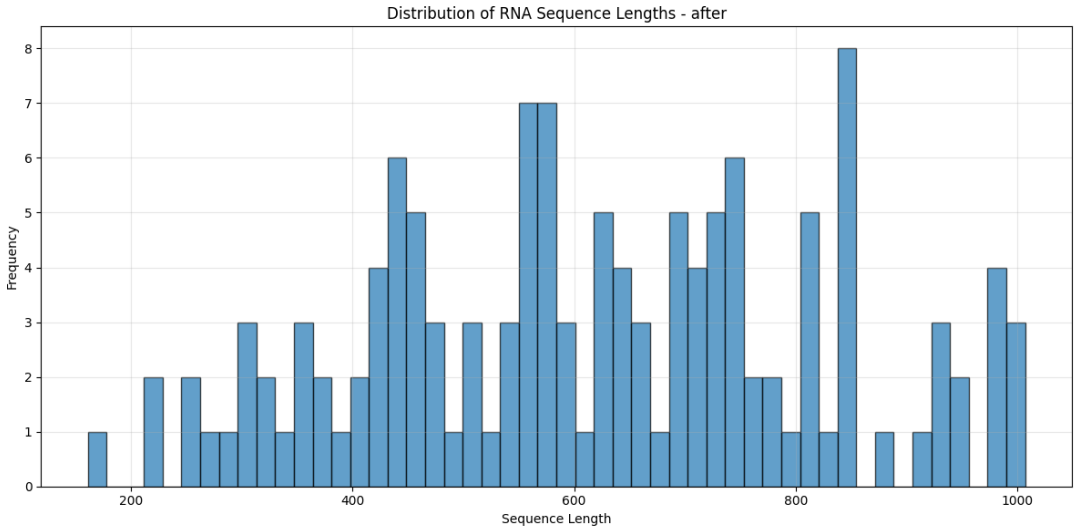


Figure 6: Dataset distribution after filtering

The post-filtering length distribution exhibits a more uniform profile among shorter sequences, with notable peaks around 600 and 850 nucleotides, indicating that the retained sequences fall well within the model's processing capacity.

Following filtration, the dataset connectivity was examined by constructing a graph-based representation where nodes correspond to unique RNA sequences and edges represent experimentally validated interactions. Network analysis revealed a highly fragmented topology comprising 55 disconnected components and exhibiting very sparse connectivity (density = 0.0089). The degree distribution was extremely skewed: 95% of nodes (125/131) had exactly one interaction partner, five nodes had between two and five partners, and a single hub node displayed 17 connections.

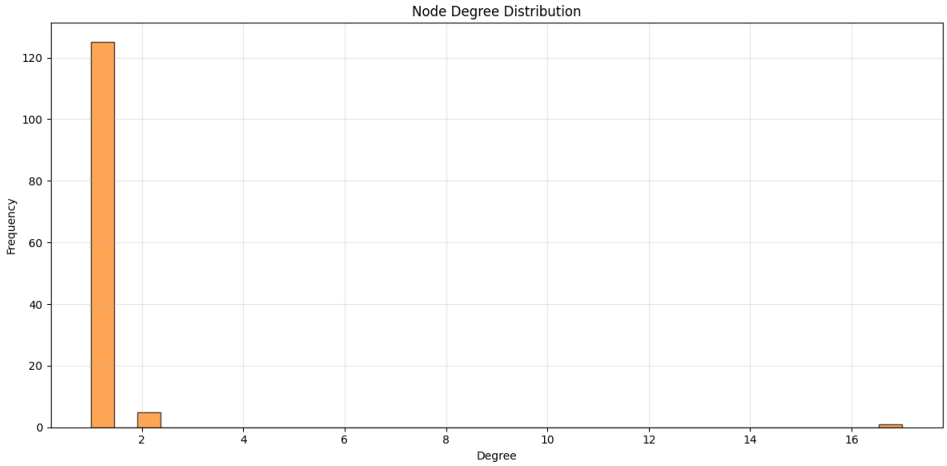


Figure 7: Node degree distribution

This star-like topology, characterized by one central hub and predominantly isolated edges, suggests one of the following scenarios: (1) the presence of a highly promiscuous RNA with broad binding capacity, (2) incomplete interaction data, or (3) a filtering effect that disproportionately retained interactions involving a specific sequence.

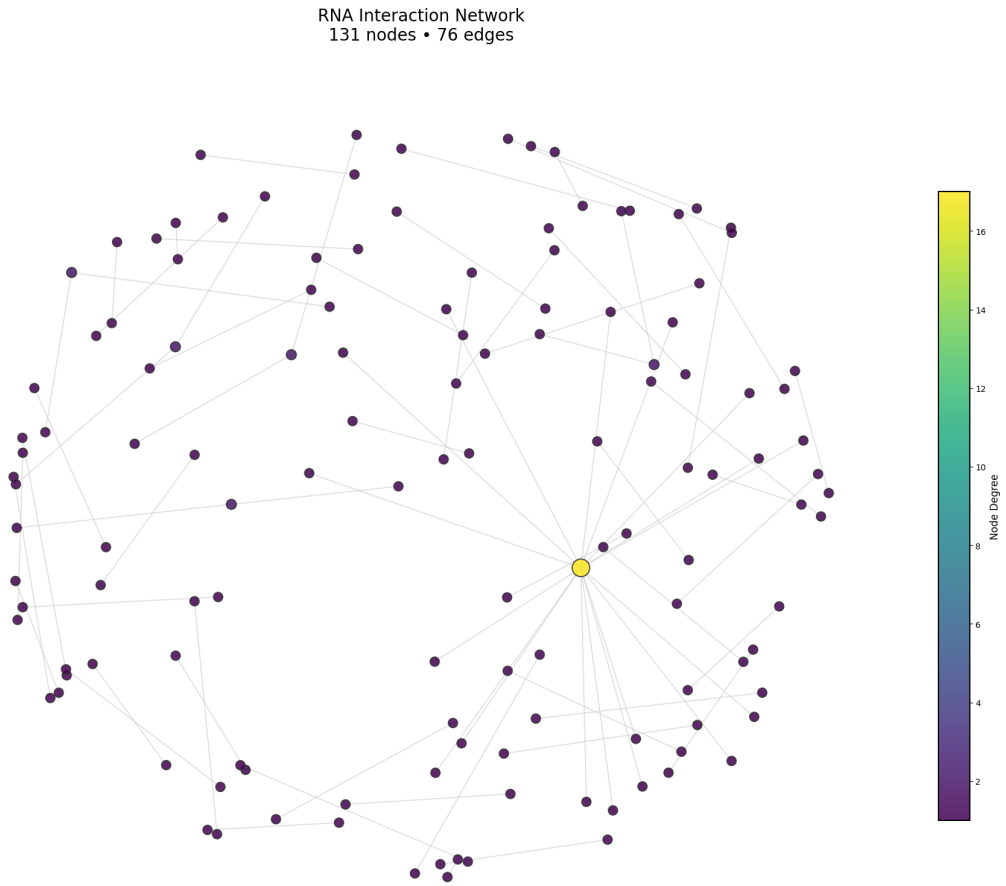


Figure 8: Interaction network

The fragmented structure indicates that the remained RNA interactions do not form a cohesive biological network but instead consist largely of isolated pairs with minimal cross-talk. Quantitative network metrics further confirms this sparse connectivity pattern, demonstrating exceptionally low density and high fragmentation. Table 1 summarizes these metrics, highlighting both the fundamental structural properties of the graph and the pronounced skew in degree distribution that characterizes this interaction landscape.

Category	Metric	Value
Basic Properties	Nodes	131
	Edges	76
	Graph density	0.0089
	Is connected	No
	Connected components	55
	Largest component	18
Degree Statistics	Average degree	1.16
	Median degree	1
	Max degree	17
	Min degree	1
Degree Distribution	Degree = 1	125
	Degree 2-5	5
	Degree > 10	1

Table 1: Graph Network Analysis Results

To identify the optimal neural network architecture for predicting lncRNA-pseudoRNA interactions, five distinct hyperparameter configurations were systematically evaluated, spanning a range of architectural choices. The baseline configuration served as the reference point, followed by four alternative strategies: increasing depth, expanding width, reducing complexity, and adopting an intermediate design.

Configuration	Layers	Width	Dropout	LR	Batch	Warmup
`Baseline`	4	1024	0.20	0.0005	512	4
`Deeper_HighDropout`	6	1024	0.30	0.0003	512	5
`Wider_LowLR`	4	2048	0.20	0.0001	512	6
`Smaller_HighLR`	3	512	0.15	0.0010	256	6
`Balanced`	5	1536	0.25	0.0002	384	5

Table 2: Hyperparameter configurations for model exploration

All models were trained on identical train-validation-test splits, each represented as 2560-dimensional concatenated RNA-FM embeddings. Training employed cosine annealing with warmup, balanced batch sampling to address class imbalance, and early stopping based on validation loss with a patience of 10 epochs. For each model, the optimal classification threshold was determined by maximizing the F1 score on the validation set rather than using the conventional 0.5 threshold, thereby accounting for the inherent class imbalance in the interaction data.

Training revealed distinct convergence patterns across configurations. The baseline model showed rapid early learning, with training loss dropping substantially during the four-epoch warmup period, ultimately reaching its minimum by epoch 14. However, validation loss plateaued after epoch 4 and subsequently fluctuated within a narrow range, indicating overfitting despite dropout regularization. Instead, the `Deeper_HighDropout` configuration converged more slowly due to increased capacity and stronger regularization, starting with noticeably higher initial losses. Despite this, it achieved its best validation performance at epoch 8 and trained the longest, stopping at epoch 16, suggesting that deeper architectures with aggressive regularization require extended training but ultimately capture more complex patterns. By contrast, the `Wider_LowLR` model exhibited the most conservative learning trajectory, with its extremely low learning rate producing gradual optimization over 18 epochs. Although it achieved very low training loss, its validation performance lagged behind other configurations, indicating overfitting despite cautious learning. In comparison, the `Smaller_HighLR` configuration converged most rapidly, reaching its best validation loss at epoch 4 - the earliest and strongest early performance among all models. However, its aggressive learning rate limited fine-tuning capability, resulting in early stopping at epoch 14. Finally, the `Balanced` configuration demonstrated moderate, steady improvement across 18 epochs, with its intermediate capacity appearing well-matched to the dataset complexity, avoiding both underfitting and excessive training requirements.

A comprehensive evaluation on the held-out test set revealed that all five configurations achieved remarkably strong performance, with accuracies ranging from 98.60% to 99.00%. However, more nuanced differences emerged when examining metrics specifically designed for imbalanced classification tasks.

The `Wider_LowLR` configuration emerged as the overall best performer, achieving the highest scores in four of seven evaluation metrics. Its combination of high accuracy, strong balanced accuracy, and the best F1 score indicates that it not only distinguishes interacting from non-interacting pairs effectively but also maintains an advantageous trade-off between precision and recall. The model's competitive AUROC and AUPRC further confirm that its discriminative ability remains robust across decision thresholds. Crucially, the success of this configuration highlights the value of increased representational capacity for modeling the complex, high-dimensional structure of RNA-FM embeddings. The wide hidden layers appear to capture subtle interaction patterns that narrower or shallower architectures miss, while the conservative learning rate and extended warmup stabilize training despite the large parameter count. In practical terms, this makes `Wider_LowLR` the most dependable choice for downstream biological applications, offering both sensitivity to true interactions and confidence in its predictions.

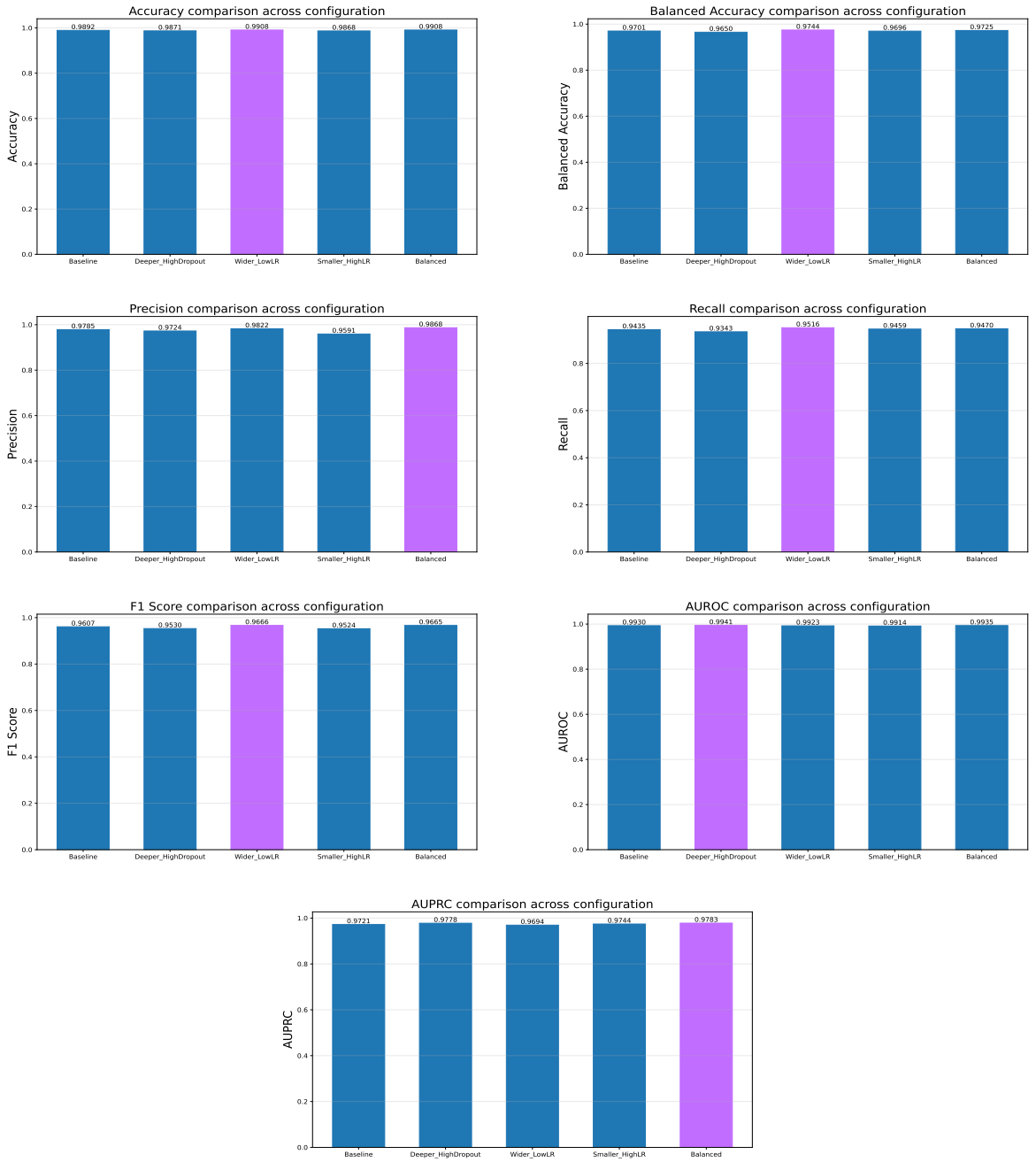
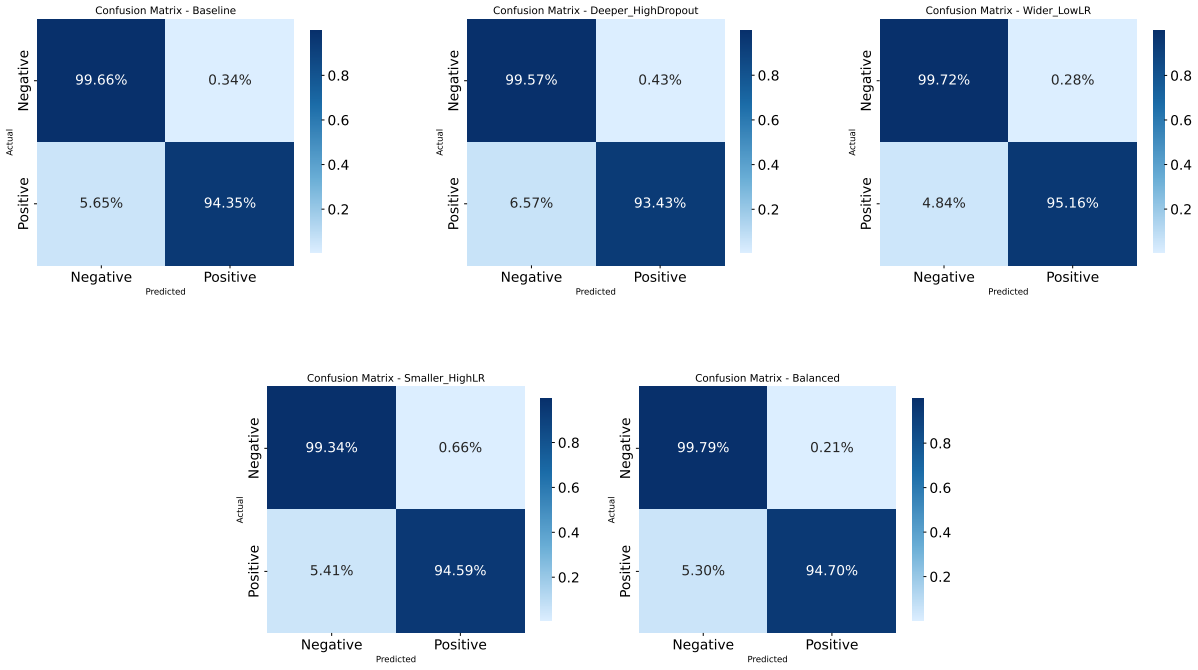


Figure 9: Metrics comparison across configuration

Instead, the `Deeper_HighDropout` configuration achieved the best threshold-independent performance, with the highest AUROC and AUPRC, indicating superior discriminative ability across all possible classification thresholds. Interestingly, the baseline configuration achieved the highest precision, meaning it made the fewest false positive predictions. However, its recall was the lowest among all but the `Smaller_HighLR` configuration, indicating a conservative prediction strategy that favored specificity over sensitivity.

The `Balanced` configuration performed solidly across all metrics without excelling in any particular one. The `Smaller_HighLR` configuration, despite its computational efficiency, exhibited the weakest overall performance. However, its AUPRC surprisingly competitive with larger models, indicates that when it does predict interactions, it generally assigns appropriate confidence scores, even if the absolute classification performance is diminished.



Normalized confusion matrices provided granular insight into classification error patterns. The `Wider_LowLR` configuration achieved 99.6% true negative rate and 95.2% true positive rate, demonstrating strong performance on both classes with false positive and false negative rates of approximately 0.3% and 4.8%, respectively. This highly asymmetric error distribution indicates the model exhibits substantially better specificity than sensitivity, though both remain strong. The `Deeper_HighDropout` model showed similar pattern but with slightly degraded performance on both classes. While, the `baseline` configuration demonstrated intermediate performance with 99.7% true negative rate and 94.4% true positive rate, its error pattern falls between the other two configurations.

All configurations demonstrated true positive rates exceeding 93% and true negative rates above 97%, confirming robust classification performance on both classes. The relatively small variation in confusion matrix elements across models suggests that all tested architectures are fundamentally capable of learning the interaction patterns, with performance differences reflecting fine-tuning rather than qualitative capability gaps.

References

[1] Rincón-Riveros, A.; Morales, D.; Rodríguez, J.A.; Villegas, V.E.; López-Kleine, L.; *Bioinformatic Tools for the Analysis and Prediction of ncRNA Interactions* Int. J. Mol. Sci. 2021, 22, 11397. https://doi.org/10.3390/ijms222111397

[2] CD Genomics Blog [July, 2025] *Role of non-coding RNAs in gene regulation.* Available at https://www.cd-genomics.com/blog/role-non-coding-rnas-gene-regulation/

[3] Tingpeng Yang; Yonghong He; Yu Wang; *Introducing TEC-LncMir for prediction of lncRNA-miRNA interactions through deep learning of RNA sequences*, Briefings in Bioinformatics, Volume 26, Issue 1, January 2025, bbaf046, https://doi.org/10.1093/bib/bbaf046

[4] Zhao Jingxuan *Research progress on predicting the interaction between long noncoding RNA and protein based on deep learning*, Vol. 12 No. 4, 2022. 10.12677/CSA.2022.124087