

RBPTool: A Deep Language Model Framework for Multi-Resolution RBP-RNA Binding Prediction and RNA Molecule Design

Jiyue Jiang^{♡*}, Yitao Xu^{♡*}, Zikang Wang[♣], Yihan Ye[♣], Yanruisheng Shao[♡],

Yuheng Shan[◊], Jiuming Wang[♡], Xiaodan Fan[♡], Jiao Yuan^{△★}, Yu Li[♡]

[♡] CUHK, HKSAR, [♣] PolyU, HKSAR, [♣] Tongji University, [◊] NUS, Singapore,

[△] Guangzhou National Laboratory, [★] Guangzhou Medical University

{jiangjy, yitaoxu, 1155231342, jmwang}@link.cuhk.edu.hk,

zikang.wang@connect.polyu.hk, 2250116@tongji.edu.cn, shan.yuheng@u.nus.edu,

xfan@cuhk.edu.hk, yuan_jiao@gzlab.ac.cn, liyu@cse.cuhk.edu.hk

Abstract

RNA-binding proteins (RBPs) play essential roles in post-transcriptional gene regulation via recognizing specific RNA molecules as well as modulating several key physiological processes *in cellulo*, represented by alternative splicing and RNA degradation. Despite extensive research, most existing approaches still rely on superficial sequence features or coarse structural representations, limiting their ability to capture the intricate nature of RBP-RNA interactions. The recent surge in large language models (LLMs), combined with advances in geometric deep learning for extracting three-dimensional representations, enables the integration of multi-modal, multi-scale biological data for precise modeling and biologically informed *de novo* RNA design. In this work, we curate and extend **RPI15223** into a multi-resolution, structure-level RBP-RNA dataset, and introduce **RBPTool**, a multi-task, multi-resolution framework that combines a geometric vector perception (GVP) module together with a deep language model encoder to fuse sequence and structural information. Our tool achieves state-of-the-art performance on public benchmarks and the RPI15223 dataset, while also supporting fine-grained level predictions and enabling *de novo* RNA design through a generative module conditioned on protein, cell-type, and specified species. RBPTool provides a fast and versatile platform for both fundamental RBP-RNA research and practical RNA drug design, delivering enhanced predictive accuracy and fine-grained structural insights.

1 Introduction

In eukaryotic cells, RNA-binding proteins play pivotal roles in gene expression and the execution of cellular functions by interacting with specific RNA molecules (Gerstberger et al., 2014; Sakakibara et al., 2002). Accurately characterizing the binding patterns between RBPs and RNAs not only

deepens our understanding of complex biological processes but also provides new ideas for RNA molecule design (Kim et al., 2013; Jiang et al., 2025a). However, most existing methods focus on shallow sequences or simple structural features and face limitations in capturing the true interaction patterns between RBPs and RNAs.

In recent years, the rapid rise of large language models endows them with powerful capabilities to encode extensive contextual information (Yu et al., 2024; ESM Team, 2024; Devlin et al., 2018; Jiang et al., 2025b; Wang et al., 2025); meanwhile, in structural biology, three-dimensional feature extraction techniques based on geometric deep learning continue to evolve (Jing et al., 2021; Huang et al., 2024; Batzner et al., 2022), supporting the integration of multi-modal and multi-scale biological data. Additional related work is provided in Appendix A.1. A major challenge remains in combining the contextual understanding offered by language models with three-dimensional structural perception to enhance fine-grained RBP-RNA interaction prediction and support RNA molecule design informed by biological function constraints.

To address these limitations, in addition to existing sequence-level data (Xu et al., 2023; Ray et al., 2009, 2017), we curate and expand a structural-level RBP-RNA dataset, RPI15223, and develop a multi-task, multi-resolution prediction and generation tool called RBPTool. Building on these data, we introduce a unified neural framework that integrates a geometric vector perception module with sequence encoders, forming a dual-channel architecture that jointly captures sequence and structural information. RBPTool achieves state-of-the-art (SOTA) classification accuracy at the sequence level on public benchmarks, while also supporting finer-grained level predictions, as demonstrated on the RPI15223 dataset. Furthermore, our generative module designs custom RNA molecules under specific protein, cell-type, and organism con-

*These authors are co-first authors.

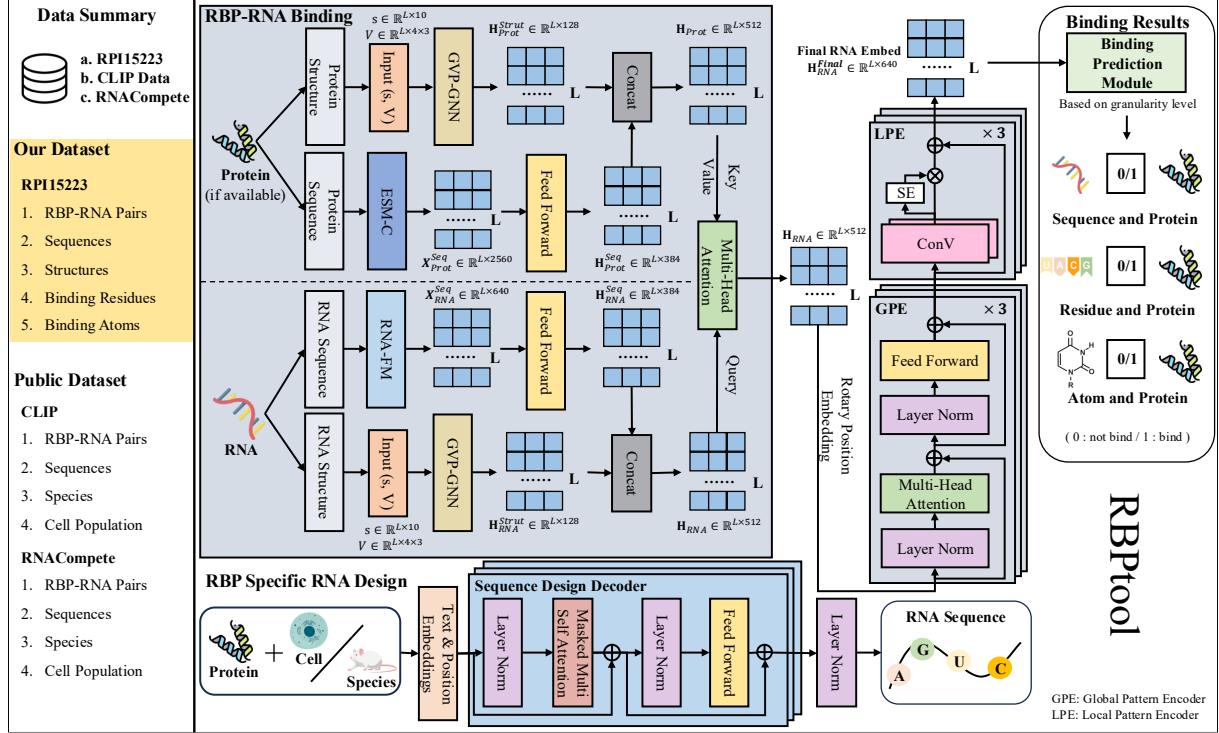


Figure 1: **Overview of the RBPtool’s architecture for Binding Prediction and RNA Design.** The binding prediction pipeline illustrated assumes that both RNA and protein sequences, along with their three-dimensional structures, are available as input. While the diagram shows all modalities being provided, the model is designed to flexibly omit any component except the RNA sequence, which is required.

ditions, providing functionally constrained RNA sequences.

The main contributions of this work are as follows: (1) We develop RBPtool, a unified neural framework powered by pretrained language models for modeling RBP-RNA interactions, which integrates sequence information with tertiary structural representation, enabling multi-granularity prediction. (2) We propose an RNA generation model that designs functional RNA sequences under real biological constraints (e.g., protein targets, cell types, and species). (3) We demonstrate through experiments on multiple benchmark datasets (e.g., CLIP, RNAcompete, RPI15223) that RBPtool achieves leading performance in both RBP-RNA binding prediction and RNA molecule design. (4) RBPtool serves as an integrated framework for RBP biology research and RNA-focused drug design, offering extensibility and general applicability to many biological and pharmaceutical contexts.

2 Proposed Method: RBPtool

We present **RBPtool**, a deep language model framework to perform various RBP tasks: (1) predicting RBP-RNA binding and (2) designing RNA se-

quences with specific RBP-binding properties. An overview of the framework is shown in Figure 1. For binding prediction, sections 2.1.1 and 2.1.2 describe how RNA and protein embeddings are extracted. Sections 2.1.4 and 2.1.5 introduce modules for capturing global and local RNA patterns. Section 2.1.6 details the prediction heads for binding at multiple resolutions. The RNA sequence design task is described in Section 2.2.

2.1 RBP-RNA Binding Prediction

2.1.1 Sequence Module

We employ the pre-trained language model RNA-FM (Chen et al., 2022) to encode RNA sequences into contextual embeddings $\mathbf{X}_{\text{RNA}}^{\text{seq}} \in \mathbb{R}^{L_r \times 640}$, where L_r denotes the length of the RNA sequence. To match the hidden dimension d_{seq} used in downstream modules, these embeddings are projected via a linear transformation and activation function:

$$\mathbf{H}_{\text{RNA}}^{\text{seq}} = \text{LayerNorm} (\text{GeLU} (\mathbf{X}_{\text{RNA}}^{\text{seq}} \mathbf{W} + \mathbf{b})) , \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{640 \times d_{\text{seq}}}$ and $\mathbf{b} \in \mathbb{R}^{d_{\text{seq}}}$. The resulting representation $\mathbf{H}_{\text{RNA}}^{\text{seq}} \in \mathbb{R}^{L_r \times d_{\text{seq}}}$ serves as the encoded RNA sequence embedding.

When protein sequences are available, we utilize

the ESM-C model (ESM Team, 2024) to generate contextual embeddings $\mathbf{X}_{\text{Prot}}^{\text{seq}} \in \mathbb{R}^{L_p \times 2560}$, where L_p is the protein sequence length. These embeddings are projected into the unified space of dimension d_{seq} using an independent projection layer.

2.1.2 Structure Module

We follow the GVP-GNN architecture (Jing et al., 2021) to extract SE(3)-equivariant structure features from the RNA backbone¹. The RNA is represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ corresponds to a nucleotide, and edges $(v_i, v_j) \in \mathcal{E}$ connect its $k = 10$ nearest neighbors based on distance between C1' atoms.

Node and Edge Features. Each node feature $\mathbf{h}_v^{(i)} = (\mathbf{s}_i, \mathbf{V}_i)$ includes scalar components $\mathbf{s}_i \in \mathbb{R}^{10}$, composed of dihedral angle encodings $\{\sin, \cos\} \circ \{\phi_i, \psi_i, \omega_i\}$ and an one-hot encoding of the nucleotide type², and vector components $\mathbf{V}_i \in \mathbb{R}^{4 \times 3}$, including local directions and orientation vectors (e.g., C1'-C4', C1'-N1/N9).

Each edge (i, j) carries a feature $\mathbf{h}_e^{(j \rightarrow i)} = (\mathbf{s}_{ij}, \mathbf{V}_{ij})$, where the scalar component $\mathbf{s}_{ij} \in \mathbb{R}^{32}$ is the concatenation of a Gaussian radial basis encoding of the Euclidean distance $\|\text{C1}'_i - \text{C1}'_j\|_2$ and a sinusoidal positional encoding of the sequence distance $|i - j|$ following Vaswani et al. (2017). The vector component $\mathbf{V}_{ij} \in \mathbb{R}^{1 \times 3}$ is the unit direction vector pointing from C1'_i to C1'_j.

Message Passing and Update. We apply three layers of GVP-GNN to iteratively update node features via message passing from neighboring nodes $\mathcal{N}(i)$. At each layer, messages are propagated as follows:

$$\begin{aligned} \mathbf{m}^{(j \rightarrow i)} &:= g(\mathbf{h}_v^{(j)} \parallel \mathbf{h}_e^{(j \rightarrow i)}), \\ \mathbf{h}_v^{(i)} &\leftarrow \text{LayerNorm}(\mathbf{h}_v^{(i)} + \frac{1}{k'} \sum_{j \in \mathcal{N}(i)} \mathbf{m}^{(j \rightarrow i)}) \end{aligned} \quad (2)$$

where g denotes a sequence of three GVPs, \parallel is concatenation operation, $\mathbf{m}^{(j \rightarrow i)}$ represents the passed messages and $k' = |\mathcal{N}(i)|$. Each GVP-GNN layer operates on hidden dimensions (192, 16) for nodes and (32, 1) for edges. We also update each node feature $\mathbf{h}_v^{(i)}$ between message passing layers via a pointwise feedforward module:

$$\mathbf{h}_v^{(i)} \leftarrow \text{LayerNorm}(\mathbf{h}_v^{(i)} + \text{Dropout}(g'(\mathbf{h}_v^{(i)}))) \quad (3)$$

¹The backbone atoms of RNA are C1', C4', and N1/N9.

² ϕ_i, ψ_i, ω_i are the backbone dihedral angles of residue *i*.

where g' denotes a sequence of two GVP layers, and Dropout is applied for regularization. After all layers, we extract the scalar component \mathbf{s}_i of each node from the scalar channel of final GVP output $\mathbf{h}_v^{(i)}$ and collect them into the final structural embedding $\mathbf{H}_{\text{RNA}}^{\text{str}} \in \mathbb{R}^{L_r \times 128}$. The final GVP uses an output dimension of (128, 0). If protein backbone structures are also provided³, we construct an analogous graph and apply the same GVP-GNN architecture (with independent parameters) to obtain the structure embedding $\mathbf{H}_{\text{Prot}}^{\text{str}} \in \mathbb{R}^{L_p \times 128}$.

2.1.3 Embeddings Integration

RNA input is required for all tasks. For each molecule (RNA or protein), if both sequence and structure are available, we concatenate their embeddings; otherwise, only the sequence embedding is used. When protein input is provided, we apply an 8-head multi-head attention layer with RNA as the query and protein as key and value to inject protein context. The final RNA representation is denoted as $\mathbf{H}_{\text{RNA}} \in \mathbb{R}^{L_r \times d_{\text{hidden}}}$.

2.1.4 Global Pattern Encoder Module

The embeddings from RNA-FM capture rich sequence features but are not optimized for RBP-specific patterns. To extract high-level contextual information aligned with RNA-protein interactions, we apply a modified transformer encoder, called the Global Pattern Encoder (GPE).

We enhance the transformer encoder with three main modifications. We first use rotary positional encoding (RoPE) (Su et al., 2024) to capture relative positional patterns better, since binding patterns in RNA often depend on relative rather than absolute positions. RoPE also generalizes well to variable-length sequences, which is common in transcriptome-wide settings. It encodes positions via complex-valued rotations such that, for a query-key pair at positions i and j , their inner product becomes:

$$\langle \text{RoPE}(q_i), \text{RoPE}(k_j) \rangle = \langle q_i, R^{i-j} k_j \rangle \quad (4)$$

where R^{i-j} is a relative rotation matrix depending on the offset $i - j$. Second, we adopt a pre-layer normalization scheme to stabilize training and improve convergence speed. For activation, we adopt Gated GeLU (GeGLU) (Shazeer and Stern, 2018), which enhances feedforward expressiveness by introducing multiplicative interactions. Given

³The protein backbone includes atoms N, C_α, and C.

an input token representation $x \in \mathbb{R}^d$,

$$\text{GeGLU}(x) = \text{GeLU}(xW + b) \odot (xV + c) \quad (5)$$

where $W, V \in \mathbb{R}^{d \times d}$, $b, c \in \mathbb{R}^d$, and \odot denotes element-wise multiplication. Stacking n such transformer layers yields the GPE module, which models global dependencies across the RNA sequence:

$$\mathbf{H}_{\text{RNA}}^{\text{refined}} = \text{GPE}^{(n)}(\mathbf{H}_{\text{RNA}}) \quad (6)$$

where $\mathbf{H}_{\text{RNA}}^{\text{refined}} \in \mathbb{R}^{L_r \times d_{\text{hidden}}}$, and n is the number of global pattern encoder layers.

2.1.5 Local Pattern Encoder Module

While the GPE module captures long-range dependencies, its global attention may overlook discriminative RNA motifs. To address this, we introduce the Local Pattern Encoder (LPE) module.

Each LPE layer begins with two consecutive 1D convolutions with kernel size k , followed by a squeeze-and-excitation (SE) block (Hu et al., 2018) that reweights channel-wise features. Formally, given input $\mathbf{X} \in \mathbb{R}^{B \times L \times C}$, the output is:

$$\text{LPE}(\mathbf{X}) = \mathbf{X} + (\text{SE}(\mathbf{U}) \odot \mathbf{U}) \quad (7)$$

where $\mathbf{U} = \text{Conv}^2(\mathbf{X})$, $\text{Conv}^2(\cdot)$ denotes two stacked 1D convolutions with kernel size 3, and \odot is element-wise multiplication. The SE block first applies global average pooling over the sequence length L to obtain $\mathbf{U}_{\text{avg}} \in \mathbb{R}^{B \times C \times 1}$, then uses two pointwise (1×1) convolutions with LeakyReLU (Maas et al., 2013):

$$\text{SE}(\mathbf{U}) = \text{Sigmoid}(\mathbf{U}_{\text{avg}} \mathbf{W}_1 \sigma(\cdot) \mathbf{W}_2),$$

where $\mathbf{W}_1 \in \mathbb{R}^{C \times C/32}$, $\mathbf{W}_2 \in \mathbb{R}^{C/32 \times C}$, and $\sigma(\cdot)$ denotes LeakyReLU. We stack m such LPE layers to obtain the final RNA embedding:

$$\mathbf{H}_{\text{RNA}}^{\text{final}} = \text{LPE}^{(m)}(\mathbf{H}_{\text{RNA}}^{\text{refined}}) \quad (8)$$

where $\mathbf{H}_{\text{RNA}}^{\text{final}} \in \mathbb{R}^{L_r \times d_{\text{hidden}}}$. The LPE module reinforces compact, motif-like patterns within the RNA sequence, complementing the global context captured by GPE⁴.

⁴Since the GPE module outputs are in (batch, sequence, hidden) format and the LPE expects (batch, channel, sequence), we apply a transpose before LPE.

2.1.6 Binding Prediction Module

We predict binding at three levels-sequence, residue, and atom-using a two-layer MLP classifier. This classifier produces logits and is defined as:

$$f(\mathbf{x}) = \mathbf{W}_{\text{out}} \cdot \text{SiLU}(\mathbf{W}_{\text{mid}} \mathbf{x} + \mathbf{b}_{\text{mid}}) + \mathbf{b}_{\text{out}}, \quad (9)$$

where $\mathbf{W}_{\text{mid}} \in \mathbb{R}^{d_{\text{mid}} \times d_{\text{hidden}}}$, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{mid}}}$, $\mathbf{b}_{\text{mid}} \in \mathbb{R}^{d_{\text{mid}}}$, and $\mathbf{b}_{\text{out}} \in \mathbb{R}^{d_{\text{out}}}$. We use $d_{\text{mid}} = d_{\text{hidden}}/2$, and d_{out} depends on the prediction level.

Sequence-Level Prediction. To obtain a fixed-size RNA representation, we apply a gated attention mechanism over the final embeddings $\mathbf{H}_{\text{RNA}}^{\text{final}} \in \mathbb{R}^{L_r \times d_{\text{hidden}}}$:

$$\begin{aligned} \boldsymbol{\alpha} &= \text{Softmax}(\mathbf{W}_q \sigma(\mathbf{H}_{\text{RNA}}^{\text{final}} \mathbf{W}_k)), \\ \mathbf{z} &= \sum_{i=1}^{L_r} \alpha_i \mathbf{H}_{\text{RNA}}^{\text{final}}[i] \end{aligned} \quad (10)$$

where $\mathbf{W}_k \in \mathbb{R}^{d_{\text{hidden}} \times d_a}$, $\mathbf{W}_q \in \mathbb{R}^{d_a \times 1}$, σ is LeakyReLU, and $\boldsymbol{\alpha} \in \mathbb{R}^{L_r}$. The aggregated vector $\mathbf{z} \in \mathbb{R}^{d_{\text{hidden}}}$ is then passed through $f(\cdot)$ with $d_{\text{out}} = 1$ to produce the binding logit $\hat{y} \in \mathbb{R}$.

Residue-Level Prediction. For each position i , the embedding $\mathbf{H}_{\text{RNA}}^{\text{final}}[i] \in \mathbb{R}^{d_{\text{hidden}}}$ is fed independently into $f(\cdot)$ with $d_{\text{out}} = 1$ to yield the residue-level binding logit $\hat{y}_i \in \mathbb{R}$.

Atom-Level Prediction. To predict atom-level binding, we again input each $\mathbf{H}_{\text{RNA}}^{\text{final}}[i]$ into the MLP, but set $d_{\text{out}} = 3$, producing a three-dimensional logit vector $\hat{\mathbf{y}}_i \in \mathbb{R}^3$ corresponding to the backbone atoms C1', C4', and N1/N9:

$$\hat{\mathbf{y}}_i = f(\mathbf{H}_{\text{RNA}}^{\text{final}}[i]). \quad (11)$$

For all three levels, the model is trained by minimizing a weighted binary cross-entropy loss:

$$\begin{aligned} \mathcal{L}_{\text{WCE}} &= -\frac{1}{N} \sum_{i=1}^N \left[w_{\text{pos}} \cdot y_i \cdot \log(\sigma(\hat{y}_i)) \right. \\ &\quad \left. + (1 - y_i) \cdot \log(1 - \sigma(\hat{y}_i)) \right], \end{aligned} \quad (12)$$

where $\sigma(\cdot)$ is the sigmoid function, w_{pos} is the positive class weight, and N is the number of instances. Some parameter settings vary based on the available input modalities; see the Appendix A.2 for detailed configurations.

2.2 RNA Sequence Design for Target RBP

The goal of this task is to generate RNA sequences that specifically bind to a target RBP, conditioned on contextual labels including the target RBP (P_{target}) and either the associated cell type (C_{cell}) or species (S_{species}).

First, we convert the input conditional label $L_{\text{cond}} = (P_{\text{target}}, C_{\text{cell}})$ or $L_{\text{cond}} = (P_{\text{target}}, S_{\text{species}})$ into a numerical embedding.

This is a standard conditional text generation task, and the model is optimized by negative log likelihood loss:

$$\mathcal{L} = - \sum_{i=1}^N \log p(y_i | y_{<i}, L_{\text{cond}}) \quad (13)$$

3 Experiments and Results

In this section, we comprehensively evaluate the effectiveness of our proposed RBPtool across two RBP tasks: (1) **RBP-RNA Binding** (Section 3.3), which is examined at three levels of granularity, **sequence** (Section 3.3.1), **residue** (Section 3.3.2), and **atom** (Section 3.3.3); and (2) **RBP-Specific RNA Design** (Section 3.4). A general overview of the experimental setup is provided in Section 3.1, while task-specific configurations and evaluation protocols are detailed in their respective sections.

3.1 Experiment Setting

We train all models using a batch size of 32 for up to 200 epochs, with early stopping based on a patience of 20 epochs. Optimization is performed using the Adam optimizer (Kingma and Ba, 2014) with a maximum learning rate of 1e-4, combined with a scheduler that applies 10% linear warm-up followed by cosine annealing with restarts (Loshchilov and Hutter, 2017). All experiments are conducted on four NVIDIA RTX 3090 (24GB) for binding tasks and eight A100 GPUs (80GB) for design. Default hyperparameters are used unless otherwise specified.

3.2 RPI15223 Dataset

We construct this dataset through the following pipeline: (1) **Retrieval:** We query EMDB and PDB using keywords related to RBP-RNA complexes and collect all matching structures from the Protein Data Bank (PDB). (2) **Pair Identification:** An RNA-protein pair is considered binding if any heavy atom in the RNA is within 3.5 Å of any heavy atom in the protein. (3) **Filtering:** We retain pairs

with structure resolution better than 4 Å, where the RNA is between 10 and 1,022 nucleotides long, and the protein is between 10 and 2,046 residues long. The resulting dataset consists of 15,223 unique RNA-protein binding pairs with corresponding sequence and structural information.

3.3 RBP-RNA Binding

In this task, we evaluate RBPtool on a series of RBP-RNA Binding tasks across three structural levels: sequence, residue, and atom. Each level captures different biological and structural aspects of RNA-protein interactions, enabling a comprehensive evaluation of the model under varying information conditions.

3.3.1 Sequence Level

At the sequence level, the primary objective is to predict whether a given RNA sequence can bind to a specific RBP under a defined cellular context. For each RBP, we train an independent binary classifier using RNA sequences as input. Additionally, we also include a supplementary experiment using both RNA and protein sequences to predict sequence-level binding, demonstrating the broad applicability of RBPtool.

Datasets. We use three datasets: **CLIP**, **RNAcompete**, and **RPI15223**. The **CLIP** dataset (Xu et al., 2023) includes 171 RBPs, each with 15,000 RNA sequences (101 nt), split into 5,000 positives and 10,000 negatives. The **RNAcompete** dataset (Ray et al., 2009, 2017) covers 162 RBPs with 1,520–16,265 sequences per RBP (30–41 nt), following a 1:2 positive-to-negative ratio. For **RPI15223**, we first removed duplicates from all RNA sequences to obtain the positive set and generated a negative set of twice the size by randomly sampling RNA sequences of lengths between 12 and 1022 nucleotides to construct the whole dataset. Dataset details and splits are in Section A.4.

Baselines. We compare our approach with three representative baselines: PrismNet, iDeepS, and HDRNet. We only use RNA sequence information without other information as input across all models for a fair comparison. As required by HDRNet, we generate sequence embeddings using its pre-trained BERT model⁵ described in its paper. All baselines are trained for each RBP individually

⁵<https://figshare.com/articles/dataset/24132423>

Models	CLIP			RNCompete		
	ACC	AUPR	AUROC	ACC	AUPR	AUROC
PrismNet (Sun et al., 2021)	0.632 ± 0.166	0.674 ± 0.111	0.801 ± 0.076	0.872 ± 0.072	0.883 ± 0.098	0.932 ± 0.074
iDeepS (Pan et al., 2025)	0.709 ± 0.115	0.664 ± 0.113	0.768 ± 0.104	0.863 ± 0.091	0.880 ± 0.091	0.928 ± 0.079
HDRNet (Zhu et al., 2023)	0.654 ± 0.174	0.646 ± 0.110	0.780 ± 0.075	0.748 ± 0.158	0.807 ± 0.110	0.890 ± 0.077
RBTool	0.773 ± 0.067	0.720 ± 0.094	0.824 ± 0.065	0.878 ± 0.066	0.884 ± 0.096	0.931 ± 0.073

Table 1: Mean \pm standard deviation of ACC, AUPR, and AUROC across all RBP-specific models on the CLIP and RNCompete datasets for the RBP binding task. Bold numbers indicate the highest average scores for each metric.

using their publicly available codes and default hyperparameter settings provided on GitHub.

Evaluation Metrics. We assess model performance on this task using three standard metrics: **accuracy (ACC)**, **area under the precision-recall curve (AUPR)**, and **area under the ROC curve (AUROC)**. For each RBP, an independent binary classifier is trained, yielding one set of evaluation scores per RBP. We calculate the mean and standard deviation of each metric across all RBP-specific classifiers for each dataset separately, 171 classifiers for CLIP and 162 for RNCompete, providing an aggregated view of model performance on both datasets to evaluate the overall performance and robustness of the methods.

Main Results. Table 1 reports the performance of different models on the RBP binding task in two datasets. **RBTool achieves the highest average scores in all three metrics on both datasets, with only a marginal drop in AUROC on the RNCompete dataset.** The advantage becomes more obvious on the CLIP dataset, where **RBTool outperforms all baselines by a clear margin**. This dataset is more challenging due to its more diverse RNA sequences, which introduce greater complexity in learning binding patterns. These results highlight the effectiveness of RBTool in modeling complex sequence contexts, due to the integration of GPE and pre-trained language models.

Moreover, **RBTool yields the lowest standard deviation across all metrics**, suggesting that the model remains robust despite its increased capacity. The added architectural flexibility appears to enhance generalization, enabling RBTool to better handle diverse RBP-RNA interactions.

Additionally, Table 13 shows that **RBTool performs strongly on the external RPI15223 dataset**, which involves predicting the binding potential of arbitrary RNA to unknown RBPs. Our model outperforms all baselines by a substantial margin

across all metrics, demonstrating its strong generalization capability and robustness to variable-length RNA inputs on sequence level prediction. Baseline models show noticeable limitations in this setting, since iDeepS and HDRNet both need to set a maximum length. This may be partly attributed to the Embedding Integration Module, which incorporates protein context through attention rather than simple concatenation. More results on this task can be found in Section A.6.

3.3.2 Residue Level

At the residue level, we focus on determining which nucleotides in an RNA molecule participate in direct interactions with an RBP.

Datasets. We constructed a refined benchmark from the RPI15223 dataset for residue-level RBP-RNA binding prediction. The final dataset contains 996 unique RNA sequences and 26,703 labeled nucleotides. We formulate this task using RNA input only here, since few existing baselines use both RNA and protein at this resolution. Details of full preprocessing steps are described in Appendix A.5.

Baselines. We evaluate two sequence-only models, FMbind and RNAPin, and three structure-based models, ZHmol, RLbind, and RNABind, on our dataset. FMbind is fine-tuned from the RNA foundation model RNA-FM for this task. Meanwhile, RNABind incorporates RNA language models to generate node embeddings for its GNN encoder; we assess its several variants using different pretrained models, including RNA-FM (Yu et al., 2024), RNA-MSM (Zhang et al., 2024), and ERNIE-RNA (Yin et al., 2024). FMbind and RNAPin are only trained on sequence information and all other methods are trained on both sequence and coordinates. All baselines are retrained on our dataset using their original codes and default hyperparameters under a unified evaluation pipeline.

Sequence Type	CLIP			RNACompete		
	RBTool	PrismNet	iDeepS	RBTool	PrismNet	iDeepS
Natural Sequence	73.40	75.02	74.30	87.50	75.02	87.36
Random	49.65	51.78	45.91	11.27	25.03	10.27
Genetic algorithm (random)	51.03	52.67	47.02	12.34	26.13	11.03
RBTool (design)	53.22	59.67	56.10	44.38	53.26	47.27

Table 2: Comparison of WSR (% , ↑) for different types of RNA sequences evaluated using RBTool, PrismNet and iDeepS on the CLIP and RNACompete datasets. The introduction of baselines is in Section A.7.

Sequence Type	CLIP			RNACompete		
	MSE	Pearson	Spearman	MSE	Pearson	Spearman
Random	0.052	0.247	0.211	0.297	0.043	0.004
RBTool (design)	0.015	0.882	0.859	0.092	0.646	0.488

Table 3: Pearson correlation (↑), Spearman correlation (↑) and MSE (↓) of AUPR between natural and synthetic sequences evaluated by Prismnet.

Evaluation Metrics. Similar to the task at the sequence level, performance is evaluated using **AUPR and AUROC**, computed globally across all nucleotides in the test set.

Main Results. As shown in Table 4, **RBTool achieves the best overall performance on the RBP binding prediction task, obtaining the highest AUPR and AUROC scores among all baselines.** The next best-performing group consists of five RNABind variants. This finding highlights the significant impact of pre-trained RNA representations on RNA-related tasks. Furthermore, RBTool consistently outperforms all RNABind variants by a notable margin of 4.3 to 9.2 points in AUPR. This demonstrates that integrating GPE modules into RBTool enhances its ability to capture complex sequence-structure patterns in RBP interactions, while the local motif information retained by the LPE may also contribute to this improvement.

3.3.3 Atom Level

At the atomic level, we further refine the task to predicting whether individual RNA backbone atoms bind to an RBP.

Datasets. We adopt the same binding-site definition as in Section 3.3.2 to label the binding backbone atoms in the **RPI15223** dataset and remove redundant pairs. The resulting set of non-redundant RNA-protein pairs, with atom-level annotations, is divided into training and test subsets using an 80/20 split, while ensuring that no RNA sequences overlap between sets to prevent data leakage.

Models	AUPR	AUROC
ZHmol (Zhuo et al., 2024)	0.576	0.532
FMbind (Yu et al., 2024)	0.638	0.621
RNAPin (Panwar and Raghava, 2015)	0.608	0.548
RLbind (Wang et al., 2023a)	0.665	0.628
RNABind_fm (Zhu et al., 2025)	0.634	0.606
RNABind_rnamsm	0.683	0.661
RNABind_rnaernie	0.668	0.619
RNABind_ernierina	0.680	0.663
RNABind_renalmo	0.665	0.642
RBTool	0.726	0.706

Table 4: AUPR and AUROC across all models on RBP binding sites prediction. Among all the baselines, RBTool achieves the highest AUPR and AUROC.

Baselines. There are no existing baselines specifically for atom-level RBP-RNA interaction prediction. Therefore, we compare the performance of RBTool at both residue and atom levels to assess whether the increased resolution leads to performance improvements or trade-offs in accuracy.

Evaluation Metrics. Due to a strong class imbalance between binding and non-binding atoms, we evaluate using metrics that are more robust in imbalanced settings: **F1 score**, **Matthews Correlation Coefficient (MCC)** and **AUPR**. The classification threshold is chosen to maximize the F1 score on the training set.

Main Results. As shown in Table 14, **RBTool achieves comparable performance on atom-level binding prediction, with only a moderate de-**

Models	CLIP			RNAcompete		
	ACC	AUPR	AUROC	ACC	AUPR	AUROC
RBTool	0.773 ± 0.067	0.720 ± 0.094	0.824 ± 0.065	0.878 ± 0.066	0.884 ± 0.096	0.931 ± 0.073
-w/o fm	0.694 ± 0.103	0.636 ± 0.122	0.759 ± 0.090	0.844 ± 0.124	0.857 ± 0.126	0.914 ± 0.098
-w/o gpe	0.708 ± 0.101	0.653 ± 0.118	0.784 ± 0.081	0.854 ± 0.094	0.839 ± 0.106	0.909 ± 0.080
-w/o lpe	0.728 ± 0.097	0.681 ± 0.115	0.798 ± 0.083	0.859 ± 0.121	0.868 ± 0.124	0.917 ± 0.111

Table 5: Ablation study of RBP binding task on CLIP and RNAcompete datasets.

Models	AUPR	AUROC
RBTool	0.726	0.706
-w/o fm	0.704	0.687
-w/o gpe	0.695	0.678
-w/o gvp	0.624	0.647

Table 6: Ablation study of RBP binding sites task.

crease relative to the residue level. This suggests that our model maintains reasonably good accuracy even at the atomic resolution, supporting the feasibility of fine-grained RBP–RNA interaction modeling.

3.4 RBP Target RNA Design

This task aims to design RNA sequences that bind to a target RBP with high affinity.

Datasets. We reuse the CLIP and RNAcompete datasets from the RBP binding task. Each instance is augmented with the target RBP’s protein sequence, the corresponding cell population, and species annotation (Section A.5).

Evaluation Metrics. We assess generation quality using two custom metrics: Weighted Success Rate (WSR) and Metric Similarity. WSR measures the predicted binding success rate of generated sequences, while Metric Similarity quantifies how closely models behave on synthetic versus natural data, using correlation and mean squared error (MSE) as indicators. Higher WSR and correlation, and lower MSE, reflect better design quality. Detailed definitions are provided in Appendix A.7.

Baselines. With no established baselines for this task, we compare against two alternatives: Random and Genetic Algorithm, built using different generation strategies described in Appendix A.7.

Main Results. Table 2 and Table 3 shows that the RNA sequences designed by RBTool achieve significantly higher WSR and Metric Similarity

compared to randomly generated sequences on both datasets. This result indicates that RBTool is capable of designing RNA sequences that specifically bind to the target RBP. The complete results of Metric Similarity are provided in Appendix A.9. High WSRs calculated on natural sequences by all three models illustrate that our metrics can effectively estimate the success rate.

3.5 Ablation Study

Both the RNA-FM and GPE significantly enhance RBP binding, while LPE and GVP-GNN further improve the language module. We ablated four variants: (1) **RBTool-w/o-fm**: one-hot instead of RNA-FM. (2) **RBTool-w/o-gpe**: no GPE module. (3) **RBTool-w/o-lpe**: no LPE in sequence-level. (4) **RBTool-w/o-gvp**: no GVP in residue-level.

Results (Tables 5 and 6) show that removing FM or GPE severely weakens both sequence- and residue-level performance, while excluding GVP significantly affects residue-level accuracy. LPE’s impact is minor, likely because RNA-FM and GPE already cover both local and distant relationships. Though structural data is beneficial, it can be hard to obtain, underscoring the importance of RNA-FM and GPE for sequence-based learning. More analysis can be found in Section A.10.

4 Conclusion and Outlook

RBTool introduces a unified framework that integrates deep language models with structural information for multi-resolution RBP–RNA binding prediction and functional RNA molecule design. It achieves state-of-the-art performance, particularly on our curated RPI15223 dataset, demonstrating its strong potential to advance both fundamental research in RBP biology and the development of RNA-based therapeutics.

In the future, incorporating richer biological data and generative algorithms could further boost cross-species and pathological predictions

Limitations

This paper introduces an RNA sequence generation module tailored for specific RBPs, and employs model scoring to evaluate the design outcomes. However, model scoring cannot fully substitute for wet-lab experimental validation. Even sequences that score highly may not achieve their expected functions *in vitro* or *in vivo*, necessitating additional experimental methods to confirm the actual binding efficiency and biological functions of the designed RNA.

In addition, the experiments primarily focus on typical RNA-binding proteins, common species, and cell types, and have not yet been validated in the context of less common or structurally complex RBPs, viral RNAs, or rare species. The generalizability and robustness of the model in these extreme scenarios still require further testing and tuning.

In this paper, we merely use AI tools to refine the language of the paper.

Ethnics Statement

This paper does not involve issues related to ethics.

Acknowledgments

We want to thank our anonymous area chair and reviewers for their feedback. This work was supported by the Chinese University of Hong Kong (CUHK; award numbers 4937025, 4937026, 5501517, 5501329, 8601603, 8601663, and SHIAE BME-p1-24 to Y.L.); the Research Grants Council of the Hong Kong Special Administrative Region, China (Hong Kong SAR; project no. CUHK 24204023 and 14208525 to Y.L.); the Innovation and Technology Commission of the Hong Kong SAR, China (project numbers GHP/065/21SZ, ITS/247/23FP and PRP/033/24FX to Y.L.); the Major Project of Guangzhou National Laboratory (Grant No. GZNL2024A01003, GZNL2023A02007, GZNL2025C02028 to J.Y.), National Natural Science Foundation of China (Grant No. 32400547 to J.Y.), Pearl River Talent Recruitment Program (2023QN10Y296 to J.Y.), Guangzhou Young Top Talent Program, and National Key R&D Program of China (2023YFF1204701 to J.Y.); a General Research Fund (14306324) sponsored by the Research Grants Council of Hong Kong to X.F.; and a Strategic Seed Funding for Collaborative Research Scheme from The Chinese University of Hong Kong (3136017) to X.F.

References

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. 2022. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. [The Protein Data Bank](#). *Nucleic Acids Res.*, 28(1):235–242.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, and 1 others. 2022. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*.

Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinhartz, Yann Ponty, Jérôme Waldspühl, and Danny Barash. 2018. Design of rnas: Comparing programs for inverse rna folding. *Briefings in bioinformatics*, 19(2):350–358.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

ESM Team. 2024. [Esm cambrian: Revealing the mysteries of proteins with unsupervised learning](#).

Stefanie Gerstberger, Markus Hafner, and Thomas Tuschl. 2014. A census of human rna-binding proteins. *Nature Reviews Genetics*, 15(12):829–845.

Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, Peter Schuster, and 1 others. 1994. Fast folding and comparison of rna secondary structures. *Monatshefte fur chemie*, 125:167–167.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7132–7141.

Han Huang, Ziqian Lin, Dongchen He, Liang Hong, and Yu Li. 2024. Ribodiffusion: tertiary structure-based rna inverse folding with generative diffusion models. *Bioinformatics*, 40(Supplement_1):i347–i356.

Jiyue Jiang, Pengan Chen, Jiuming Wang, Dongchen He, Ziqin Wei, Liang Hong, Licheng Zong, Sheng Wang, Qinze Yu, Zixian Ma, and 1 others. 2025a. Benchmarking large language models on multiple tasks in bioinformatics nlp with prompting. *arXiv preprint arXiv:2503.04013*.

Jiyue Jiang, Zikang Wang, Yuheng Shan, Heyan Chai, Jiayi Li, Zixian Ma, Xinrui Zhang, and Yu Li. 2025b. Biological sequence with language model prompting: A survey. *arXiv preprint arXiv:2503.04135*.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. 2021. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*.

Chaitanya K Joshi, Arian R Jamasb, Ramon Viñas, Charles Harris, Simon V Mathis, Alex Morehead, Rishabh Anand, and Pietro Liò. 2024. grnade: Geometric deep learning for 3d rna inverse design. *bioRxiv*.

Hong Joo Kim, Nam Chul Kim, Yong-Dong Wang, Emily A Scarborough, Jennifer Moore, Zamia Diaz, Kyle S MacLea, Brian Freibaum, Songqing Li, Amandine Molliex, and 1 others. 2013. Mutations in prion-like domains in hnrrnpa2b1 and hnrrnpa1 cause multisystem proteinopathy and als. *Nature*, 495(7442):467–473.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yanting Li, Jiyue Jiang, Zikang Wang, Ziqian Lin, Dongchen He, Yuheng Shan, Yanruisheng Shao, Jiayi Li, Xiangyu Shi, Jiuming Wang, and 1 others. 2025. Ds-progen: A dual-structure deep language model for functional protein design. *arXiv preprint arXiv:2505.12511*.

Haoquan Liu, Yiren Jian, Chen Zeng, and Yunjie Zhao. 2025. Rna-protein interaction prediction using network-guided deep learning. *Communications Biology*, 8(1):247.

Ilya Loshchilov and Frank Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, and 1 others. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, page 3. Atlanta, GA.

Yaron Orenstein, Yuhao Wang, and Bonnie Berger. 2016. Rck: accurate and efficient inference of sequence- and structure-based protein–rna binding models from rnacompete data. *Bioinformatics*, 32(12):i351–i359.

Xiaoyong Pan, Yi Fang, Xianfeng Li, Yang Yang, and Hong-Bin Shen. 2020. Rbpsuite: Rna-protein binding sites prediction suite based on deep learning. *BMC genomics*, 21:1–8.

Xiaoyong Pan, Yi Fang, Xiaojian Liu, Xiaoyu Guo, and Hong-Bin Shen. 2025. RBPsuite 2.0: An updated RNA-protein binding site prediction suite with high coverage on species and proteins based on deep learning. *BMC Biology*, 23(1).

Bharat Panwar and Gajendra P.S. Raghava. 2015. Identification of protein-interacting nucleotides in a RNA sequence using composition profile of tri-nucleotides. *Genomics*, 105(4):197–203.

Cheng Peng, Siyu Han, Hui Zhang, and Ying Li. 2019. Rpter: a hierarchical deep learning framework for ncRNA–protein interaction prediction. *International Journal of Molecular Sciences*, 20(5):1070.

Debashish Ray, Kevin CH Ha, Kate Nie, Hong Zheng, Timothy R Hughes, and Quaid D Morris. 2017. Rnacompete methodology and application to determine sequence preferences of unconventional rna-binding proteins. *Methods*, 118:3–15.

Debashish Ray, Hilal Kazan, Esther T Chan, Lourdes Pena Castillo, Sidharth Chaudhry, Shaheynoor Talukder, Benjamin J Blencowe, Quaid Morris, and Timothy R Hughes. 2009. Rapid and systematic analysis of the rna recognition specificities of rna-binding proteins. *Nature biotechnology*, 27(7):667–670.

Shin-ichi Sakakibara, Yuki Nakamura, Tetsu Yoshida, Shinsuke Shibata, Masato Koike, Hiroshi Takano, Shuichi Ueda, Yasuo Uchiyama, Tetsuo Noda, and Hideyuki Okano. 2002. Rna-binding protein musashi family: roles for cns stem cells and a subpopulation of ependymal cells revealed by targeted disruption and antisense ablation. *Proceedings of the National Academy of Sciences*, 99(23):15194–15199.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Lei Sun, Kui Xu, Wenze Huang, Yucheng T Yang, Pan Li, Lei Tang, Tuanlin Xiong, and Qiangfeng Cliff Zhang. 2021. Predicting dynamic cellular protein–rna interactions by deep learning using in vivo rna structures. *Cell research*, 31(5):495–516.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Bozhen Hu, Siyuan Li, Zicheng Liu, and Stan Z Li. 2024. Rdesign: Hierarchical data-efficient representation learning for tertiary structure-based rna design. In *The Twelfth International Conference on Learning Representations*.

Akito Taneda. 2010. Modena: a multi-objective rna inverse folding. *Advances and Applications in Bioinformatics and Chemistry*, pages 1–12.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Kaili Wang, Renyi Zhou, Yifan Wu, and Min Li. 2023a. RLBInd: A deep learning method to predict RNA–ligand binding sites. *Briefings in Bioinformatics*, 24(1):bbac486.

Yifei Wang, Pengju Ding, Congjing Wang, Shiyue He, Xin Gao, and Bin Yu. 2024. Rpi-ggcn: prediction of rna–protein interaction based on interpretability gated graph convolution neural network and co-regularized variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*.

Yifei Wang, Xue Wang, Cheng Chen, Hongli Gao, Adil Salhi, Xin Gao, and Bin Yu. 2023b. Rpi-capsulegan: predicting rna-protein interactions through an interpretable generative adversarial capsule network. *Pattern Recognition*, 141:109626.

Zhenyu Wang, Zikang Wang, Jiyue Jiang, Pengan Chen, Xiangyu Shi, and Yu Li. 2025. [Large language models in bioinformatics: A survey](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3602–3615, Vienna, Austria. Association for Computational Linguistics.

Katherine Deigan Warner, Christine E. Hajdin, and Kevin M. Weeks. 2018. [Principles for targeting rna with drug-like small molecules](#). *Nature Reviews Drug Discovery*, 17(8):547–558.

Felix Wong, Dongchen He, Aarti Krishnan, Liang Hong, Alexander Z Wang, Jiuming Wang, Zhihang Hu, Sato-taka Omori, Alicia Li, Jiahua Rao, and 1 others. 2024. Deep generative design of rna aptamers using structural predictions. *Nature Computational Science*, pages 1–11.

Yiran Xu, Jianghui Zhu, Wenze Huang, Kui Xu, Rui Yang, Qiangfeng Cliff Zhang, and Lei Sun. 2023. Prismnet: predicting protein–rna interaction using in vivo rna structural information. *Nucleic Acids Research*, 51(W1):W468–W477.

Keisuke Yamada and Michiaki Hamada. 2022. [Prediction of rna–protein interactions using a nucleotide language model](#). *Bioinformatics Advances*, 2(1):vbac023.

Weijie Yin, Zhaoyu Zhang, Liang He, Rui Jiang, Shuo Zhang, Gan Liu, Xuegong Zhang, Tao Qin, and Zhen Xie. 2024. Ernie-rna: An rna language model with structure-enhanced representations. *bioRxiv*, pages 2024–03.

Haopeng Yu, Heng Yang, Wenqing Sun, Zongyun Yan, Xiaofei Yang, Huakun Zhang, Yiliang Ding, and Ke Li. 2024. An interpretable RNA foundation model for exploring functional RNA motifs in plants. *Nature Machine Intelligence*, 6(12):1616–1625.

Xiaoli Zhang and Shiyong Liu. 2017. [Rbppred: predicting rna-binding proteins from sequence using svm](#). *Bioinformatics*, 33(6):854–862.

Yikun Zhang, Mei Lang, JiuHong Jiang, Zhiqiang Gao, Fan Xu, Thomas Litfin, Ke Chen, Jaswinder Singh, Xiansong Huang, Guoli Song, and 1 others. 2024. Multiple sequence alignment-based rna language model and its application to structural inference. *Nucleic Acids Research*, 52(1):e3–e3.

Yichong Zhao, Kenta Oono, Hiroki Takizawa, and Masaaki Kotera. 2024. Generrna: A generative pre-trained language model for de novo rna design. *PLoS One*, 19(10):e0310814.

Haoran Zhu, Yuning Yang, Yunhe Wang, Fuzhou Wang, Yujian Huang, Yi Chang, Ka-chun Wong, and Xiangtao Li. 2023. Dynamic characterization and interpretation for protein-RNA interactions across diverse cellular conditions using HDRNet. *Nature Communications*, 14:6824.

Weimin Zhu, Xiaohan Ding, Hong-Bin Shen, and Xiaoyong Pan. 2025. Identifying RNA-small molecule binding sites using geometric deep learning with language models. *Journal of Molecular Biology*, 437(8):169010.

Chen Zhuo, Jiaming Gao, Anbang Li, Xuefeng Liu, and Yunjie Zhao. 2024. A machine learning method for RNA–small molecule binding preference prediction. *Journal of Chemical Information and Modeling*, 64(19):7386–7397.

A Appendix

A.1 Related Work

A.1.1 RBP Binding at Sequence Level

Sequence-level models classify whether an RNA binds a target RBP, often using: (i) **RNA-only models** relying solely on RNA features, and (ii) **RNA-protein joint models** using both RNA and protein.

RNA-only models. Early approaches like RBP-Pred ([Zhang and Liu, 2017](#)) and RCK ([Orenstein et al., 2016](#)) employed classic ML on sequence/structure features. DeepBind ([Alipanahi et al., 2015](#)) and iDeepS ([Pan et al., 2020](#)) then introduced CNN/RNN modules, exploiting residue dependencies. More recent work leverages LLM-based embeddings for contextual patterns ([Yamada and Hamada, 2022](#); [Zhu et al., 2023](#)), and PrismNet ([Xu et al., 2023](#)) further improves performance by incorporating cellular context into a ResNet.

RNA-protein joint models. These methods fuse RNA and protein information using diverse architectures. RPITER ([Peng et al., 2019](#)) stacks autoencoders and CNNs, while RPI-CapsuleGAN ([Wang et al., 2023b](#)) explores GAN-based designs. Graph neural networks and pretrained LMs have also been

applied, such as RPI-GGCN (Wang et al., 2024) and ZHMolGraph (Liu et al., 2025).

A.1.2 RBP Binding at Residue Level

Residue-level models pinpoint specific nucleotides binding to protein residues. RNAPin (Panwar and Raghava, 2015) and ZHmol (Zhuo et al., 2024) applied ML to biological features, while recent deep learning methods better capture context and structure. RLBIND (Wang et al., 2023a) uses CNNs, and RNABIND (Zhu et al., 2025) integrates language model embeddings with GNNs. Nonetheless, joint RNA–protein information and atom-level binding remain underexplored.

A.1.3 RNA Design

Molecular design seeks sequences that fold into specific structures or perform designated functions (Warner et al., 2018; Churkin et al., 2018; Li et al., 2025). Early RNA design tools like RNAinverse and MODENA (Hofacker et al., 1994; Taneda, 2010) used heuristic searches for secondary structure. Newer approaches employ GNNs for 3D-informed design (Joshi et al., 2024; Huang et al., 2024; Tan et al., 2024), or transformers to capture global dependencies (Zhao et al., 2024; Wong et al., 2024). However, most focus on structural fidelity, overlooking functional constraints such as RNA–protein interactions. Our work merges RBP–RNA prediction with function-aware RNA design for broader biological utility.

A.2 Model Configuration Details

There are four cases of Input Modalities: (A) RNA sequence only; (B) RNA sequence and structure; (C) Sequence of both RNA & protein; (D) Sequence and structure of both RNA & protein.

Config	Case A	Case B	Case C	Case D
d_{seq}	384	256	768	384
d_{strut}	-	128	-	128
#MP	-	1	-	1
#GPE	1	3	2	3
#Heads (GPE)	6	6	8	8
#LPE	3	2	2	3

Table 7: Model configurations under different input modality settings. MP, GPE, and LPE refer to the Message Passing, Global, and Local Pattern Encoder modules, respectively. If structure is not provided, d_{strut} and #MP are omitted.

The following is the selection of architectural parameters, for the sequence-level task (which only

takes RNA sequences as input), we have conducted extensive experiments on the CLIP-seq datasets to evaluate the impact of varying the number of layers in the Global Pattern Encoder (GPE) and Local Pattern Encoder (LPE), as well as the projection dimension d_{seq} . The results are summarized in Table 8, and it can be found that the configuration (#GPE=1, #LPE=3, $d_{\text{seq}} = 384$) achieves the best overall performance across metrics.

Model	#GPE	#LPE	d_{seq}	ACC	AUPR	AUROC
Variant 0	1	3	384	0.7727	0.7202	0.8242
Variant 1	1	3	768	0.7355	0.6978	0.8061
Variant 2	1	2	384	0.7587	0.6999	0.8130
Variant 3	2	2	384	0.7598	0.7000	0.8144
Variant 4	2	3	384	0.7472	0.6906	0.8087
Variant 5	3	2	384	0.7355	0.6978	0.8061

Table 8: Performance of sequence-level model under different architectural parameters.

Additionally, here is a comprehensive set of experiments on the residue-level model by varying the number of Global Pattern Encoder (GPE) layers and Message Passing (MP) layers. The results are summarized in Table 9. As shown, the configuration with 3 GPE layers and 1 MP layer (Variant 4) achieves the best performance in both AUPR and AUROC.

Model	#GPE	#MP	AUPR	AUROC
Variant 0	3	3	0.7263	0.7064
Variant 1	1	1	0.7267	0.7048
Variant 2	1	3	0.7049	0.7010
Variant 3	2	3	0.7214	0.7035
Variant 4	3	1	0.7282	0.7157
Variant 5	3	2	0.7134	0.7036

Table 9: Performance of residue-level model under different architectural parameters.

A.3 Training Details

There are four cases of Input Modalities: (A) RNA sequence only; (B) RNA sequence and structure; (C) Sequence of both RNA & protein; (D) Sequence and structure of both RNA & protein. The following table summarizes the training details for each input modality and prediction level. For the CLIP dataset, 171 RBP-specific models are trained, one for each RBP. Similarly, RNACOMPete requires 162 separate models. For all other cases, a single unified model is trained per setting.

Case	Level	Dataset	#Params	Time
Case A	Seq	CLIP	7,196,838	36h (5-7s)
Case A	Seq	RNAcompete	7,196,838	12h (3-4s)
Case B	Res	Refined RPI	14,299,345	7min
Case C	Seq	RPI15223	31,279,410	15h
Case D	Res	RPI15223	16,091,113	8h
Case D	Atom	RPI15223	17,349,768	9h

Table 10: Training details under different input types and prediction levels. Seq: sequence-level; Res: residue-level; Atom: atom-level. The time in parentheses is the average training time of RBPtool for one RBP per epoch in CLIPs and RNAcompete datasets.

Case	Level	Dataset	#Params	Time
Design	Seq	CLIP	110 million	28
Design	Seq	RNAcompete	110 million	14

Table 11: Training details of the Design model in CLIPs and RNAcompete datasets.

A.4 Datasets Information of Sequence-Level Binding Site Prediction

We use three datasets: **CLIP**, **RNAcompete**, and **RPI15223**.

The **CLIP** dataset, curated by [Xu et al. \(2023\)](#), includes 171 RBPs. For each RBP, 15,000 fixed-length RNA sequences (101 nucleotides) are provided, consisting of 5,000 positive and 10,000 negative samples. Each RBP is annotated with the specific cell type in which the CLIP experiment was performed, and includes the full protein sequence.

The **RNAcompete** dataset, derived from the *in vitro* RNAcompete assay ([Ray et al., 2009, 2017](#)), contains 162 RBPs. RNA sequences range from 30 to 41 nucleotides in length, with 1,520 to 16,265 sequences per RBP, and maintain a 1:2 positive-to-negative ratio consistent with the CLIP dataset. Each RBP includes its full protein sequence and is annotated with its species of origin.

The **RPI15223** dataset is derived from PDB ([Berman et al., 2000](#)), which contains 15,223 non-redundant RNA-protein binding pairs. We also generate 26,682 negative samples by randomly pairing RNA and protein from different complexes.

In total, we construct 333 RBP-specific datasets, each randomly divided into 80% training and 20% testing sets for the main experiment. Similarly, the external dataset RPI15223 is split with the same ratio for the supplementary experiment, ensuring no overlap of RNA sequences between training and test sets to prevent data leakage. Additional dataset details are provided in Appendix A.5.

A.5 Supplementary Information on Datasets

1. **CLIP.** Positive samples are RNA sequences with experimentally verified binding sites centered within the sequence. Negative samples are randomly selected from non-binding regions of the transcriptome.
2. **RNAcompete.** We label sequences with high binding scores as positive and randomly sample twice as many negatives from the remaining pool.
3. **RPI15223.** We construct this dataset through the following pipeline:
 - **Retrieval:** We query EMDB and PDB using keywords related to RBP-RNA complexes and collect all matching structures from the Protein Data Bank (PDB).
 - **Pair Identification:** An RNA-protein pair is considered binding if any heavy atom in the RNA is within 3.5 Å of any heavy atom in the protein.
 - **Filtering:** We retain pairs with structure resolution better than 4 Å, where the RNA is between 6 and 1,021 nucleotides long, and the protein is between 6 and 2,045 residues long.
 The resulting dataset consists of 15,223 unique RNA-protein binding pairs with corresponding sequence and structural information.
4. **Refined RPI15223.** We further process RPI15223 to support residue-level binding prediction:
 - **Filtering:** We retain RNA chains with a length between 12 and 512 nucleotides, resulting in 1,181 RNA structures after removing redundancy.
 - **Binding-site labeling:** A nucleotide is labeled as binding if any of its heavy atoms is within 3.5 Å of a heavy atom from another chain or ligand within the biological assembly. Labels are merged across PDB entries when the same RNA appears in multiple assemblies.
 - **Clustering and alignment:** RNA sequences are clustered at 95% identity using CD-HIT, aligned with MAFFT, and binding labels are propagated across aligned positions.

The final dataset contains 996 non-redundant RNA sequences with 26,703 labeled nucleotides.

A.6 External Binding Task Results

Table 12 is the one-sided t-test between the accuracy of RBPtool and baseline models in the sequence-level RBP binding task on RNACOMPete.

Baseline	t-statistic	p-value (one-sided)
PrismNet	1.7183	4.3830e-02
iDeepS	2.5230	6.2993e-03
HDRNet	23.4472	0e-04

Table 12: Statistical comparison against baselines.

Table 13 is the performance of the sequence-level RBP binding task on RPI15223.

Models	ACC	AUPR	AUROC
PrismNet	0.7648	0.7109	0.7782
iDeepS	0.7475	0.7193	0.7910
HDRNet	0.7451	0.7475	0.8319
RBPTool	0.8510	0.8551	0.8941

Table 13: Performance in sequence-level RBP binding task on RPI15223.

Table 14 is the performance of RBPTool on the residue- and atom-level RBP binding task on RPI15223.

Level	F1	MCC	AUPR
Residue	0.667	0.655	0.697
Atom	0.651	0.636	0.687

Table 14: Comparison of RBPTool performance on RBP Binding task at residue and atom levels on RPI15223.

A.7 RBP Target RNA Design

Evaluation Metrics. We use two metrics to evaluate the quality of the generated sequences: Weighted Success Rate (WSR) and Metric Similarity. They are defined as follows:

- **Weighted Success Rate:** Weighted Success Rate (WSR) estimates an overall rate that the synthetic RNA sequences are true binders of the target RBP. For each RBP, we generate a set of candidate sequences and use the pre-trained, RBP-specific classifier in Section 3.3.1 to estimate the proportion of sequences

predicted as positive. For both the CLIP and RNACOMPete datasets, we calculate the WSR independently by taking the weighted average of the positive rates estimated across all RBPs in each dataset. The weight assigned to each RBP is its classifier’s AUPR on the test set, reflecting the classifier’s reliability. A higher WSR indicates a higher binding likelihood of the generated sequences. The formula for this metric is provided in Appendix A.8.

- **Metrics Similarity:** This metric quantifies how closely the generated sequences resemble natural RBP target RNA by evaluating the similarity between model performance on generated sequences and on true positive examples. High similarity implies that the synthetic sequences effectively capture the underlying characteristics of real RBP binding sites. For each RBP, we generate synthetic positive sequences to replace the real positive examples in the test set. We then re-evaluate the trained classifier on this modified test set to obtain new performance metrics (e.g., ACC, AUPR, AUROC). For each metric, we compute the Pearson correlation, Spearman correlation, and mean squared error (MSE) between the scores obtained on the natural and synthetic sets. Higher Pearson or Spearman correlation, or lower MSE—indicating greater metric similarity—suggests that the generated sequences more faithfully capture the properties of true positive examples. To avoid potential bias, we evaluate the performance of PrismNet on generated sequences instead of RBPTool.

Baselines. Since there are no existing methods specifically tailored for RBP-targeted RNA sequence design, we construct two competitive baselines for comparison: **Random** and **Genetic Algorithm (Random)**.

- **Random.** For each RBP, we generate RNA sequences of the same length as our designed sequences. Each nucleotide is sampled independently and uniformly from the set {A, C, G, U}. This baseline reflects the performance of completely unstructured generation and serves as a lower bound for design quality.
- **Genetic Algorithm (Random).** To make the baseline more competitive, we introduce

a selection mechanism inspired by the genetic algorithm paradigm. For each RBP, we generate five independent sets of random sequences using the same random sampling strategy described above. We then estimate the WSR of each set using three high-performing classifiers—RBTool, PrismNet, and iDeepS—identified in Section 3.3.1. The set with the highest average predicted binding probability is retained as the final baseline for that RBP. This procedure ensures a stronger and more competitive stochastic baseline against which to evaluate our designed sequences.

We compare the evaluation metrics across four groups of sequences: natural RNA, sequences designed by RBTool, randomly generated sequences, and those selected through the Genetic Algorithm (Random) baseline.

A.8 Formula of Weighted Positive Rate (WSR)

For each RBP i in either the CLIP or RNACOMPete dataset, let:

- $r_i = \frac{1}{|G_i|} \sum_{x \in G_i} \mathbb{I}(\hat{y}_i(x) = 1)$, where G_i is the set of generated RNA sequences for RBP i , and $\hat{y}_i(x) \in \{0, 1\}$ is the predicted label given by the corresponding RBP-specific classifier.
- $w_i = \text{AUPR}_i$, the area under the precision-recall curve (AUPR) of the classifier for RBP i evaluated on the natural test set.

Then, the **Weighted Success Rate (WSR)** for a given dataset is computed as:

$$\text{WSR}_{\text{dataset}} = \frac{\sum_{i=1}^N w_i \cdot r_i}{\sum_{i=1}^N w_i}$$

where N denotes the number of RBPs in that dataset (CLIP or RNACOMPete).

A.9 Detailed Metric Similarity

To further evaluate the quality of the generated RNA sequences, we report the detailed results of *Metrics Similarity* (measured by MSE, Pearson correlation, and Spearman correlation) between predicted and actual evaluation scores (ACC, AUROC) on the CLIP and RNACOMPete datasets. We compare results for sequences generated by random sampling and our designed method.

Sequence Type	MSE	Pearson	Spearman
Random	0.034	0.073	0.169
RBTool (design)	0.026	0.386	0.505

Table 15: ACC evaluation on **CLIP**. Lower MSE and higher correlation indicate better similarity to natural sequences.

Sequence Type	MSE	Pearson	Spearman
Random	0.092	0.234	0.152
RBTool (design)	0.048	0.408	0.434

Table 16: ACC evaluation on **RNACOMPete**. Lower MSE and higher correlation indicate better similarity to natural sequences.

Sequence Type	MSE	Pearson	Spearman
Random	0.009	0.849	0.849
RBTool (design)	0.028	0.355	0.303

Table 17: AUROC evaluation on **CLIP**. Lower MSE and higher correlation indicate better similarity to natural sequences.

Sequence Type	MSE	Pearson	Spearman
Random	0.052	0.558	0.403
RBTool (design)	0.181	0.018	-0.059

Table 18: AUROC evaluation on **RNACOMPete**. Lower MSE and higher correlation indicate better similarity to natural sequences.

A.10 Ablation Study: How Does Each Component Work?

Both the RNA Foundation Model (FM) and the GPE facilitate RBP binding tasks. In addition, LPE and GVP-GNN can respectively further expand the assistance brought by the language module. We conduct ablation studies on both the RBP Binding tasks at the sequence level and the residue level to understand the influence of different components in each task. The models we compare are listed as follows: (1) RBTool-w/o-fm replaces the RNA-FM with one-hot encoding to extract RNA representations; (2) RBTool-w/o-gpe removes the BERT block in both tasks. (3) RBTool-w/o-lpe removes the ResNet module only in the sequence level task. (4) RBTool-w/o-gvp removes the GVP module merely in the residue level task.

The result in Table 5 and 6 indicates that replacing the RNA-FM with one-hot encoding and the removal of GPE Component both lead to a significant decline in performance of Binding Task at both the sequence level and the residue level. Meanwhile, the absence of the GVP module decreases the performance at residue level the most, and removing the LPE module slightly decreases the performance. These results indicate that RNA-FM and transformer-based GPE modules, as the basis of language models, play a crucial role in learning the information in the RBP binding task. Meanwhile, structural information can also be further and significantly enhanced through the GVP module for the learning of information. However, the LEP module is relatively less helpful for the task. We believe that this is because the GPE and FM modules have simultaneously obtained the information of both the distant and near distances in the RNA sequence, which weakens the help that the LPE module brings to the model by learning the motif information to some extent. It is worth noting that although structural information can bring significant help to the RBP task, since such data is relatively more difficult to obtain, the role of RNA-FM and GPE modules for sequences is more meaningful.