Deciphering RNA Secondary Structure Prediction: A Probabilistic K-Rook Matching Perspective

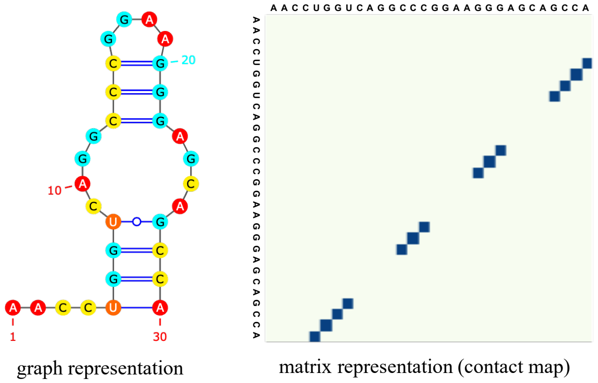
Cheng Tan[1 2 *] Zhangyang Gao[2 1 *] Hanqun Cao[3] Xingran Chen[4] Ge Wang[2] Lirong Wu[2] Jun Xia[2]
Jiangbin Zheng[2] Stan Z. Li[† 2]

Abstract

The secondary structure of ribonucleic acid (RNA) is more stable and accessible in the cell than its tertiary structure, making it essential for functional prediction. Although deep learning has shown promising results in this field, current methods suffer from poor generalization and high complexity. In this work, we reformulate the RNA secondary structure prediction as a K-Rook problem, thereby simplifying the prediction process into probabilistic matching within a finite solution space. Building on this innovative perspective, we introduce RFold, a simple yet effective method that learns to predict the most matching K-Rook solution from the given sequence. RFold employs a bi-dimensional optimization strategy that decomposes the probabilistic matching problem into row-wise and column-wise components to reduce the matching complexity, simplifying the solving process while guaranteeing the validity of the output. Extensive experiments demonstrate that RFold achieves competitive performance and about eight times faster inference efficiency than the state-of-the-art approaches. The code is available at github.com/A4Bio/RFold.

1. Introduction

The functions of RNA molecules are determined by their structure (Sloma & Mathews, 2016). The secondary structure, which contains the nucleotide base pairing information, as shown in Figure 1, is crucial for the correct functions of RNA molecules (Fallmann et al., 2017). Although experimental assays such as X-ray crystallography (Cheong et al., 2004), nuclear magnetic resonance (Fürtig et al., 2003), and cryogenic electron microscopy (Fica & Nagai, 2017) can be implemented to determine RNA secondary structure, they suffer from low throughput and expensive cost.



graph representation
matrix representation (contact map)

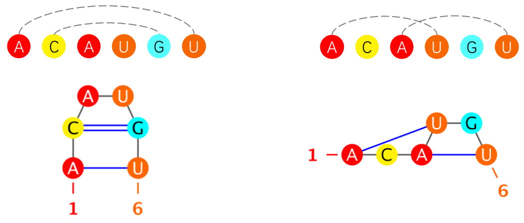
Figure 1. The graph and matrix representation of an RNA secondary structure example.

Computational RNA secondary structure prediction methods have been favored for their high efficiency in recent years (Iorns et al., 2007). Currently, mainstream methods can be broadly classified into two categories (Rivas, 2013; Szikszai et al., 2022): (i) comparative sequence analysis and (ii) single sequence folding algorithm. Comparative sequence analysis determines the secondary structure conserved among homologous sequences but the limited known RNA families hinder its development (Gutell et al., 2002; Griffiths-Jones et al., 2003; Gardner et al., 2009; Nawrocki et al., 2015). Researchers thus resort to single RNA sequence folding algorithms that do not need multiple sequence alignment information. A classical category of computational RNA folding algorithms is to use dynamic programming (DP) that assumes the secondary structure is a result of energy minimization (Bellaousov et al., 2013; Nicholas & Zuker, 2008; Lorenz et al., 2011; Zuker, 2003; Mathews & Turner, 2006; Do et al., 2006). However, energy-based approaches usually require the base pairs have a nested structure while ignoring some valid yet biologically essential structures such as pseudoknots, i.e., non-nested base pairs (Chen et al., 2019; Seetin & Mathews, 2012; Xu & Chen, 2015), as shown in Figure 2. Since predicting secondary structures with pseudoknots under the energy minimization framework has shown to be hard and

[*] Equal contribution [1] Zhejiang University [2] Westlake University [3] The Chinese University of Hong Kong [4] University of Michigan. Correspondence to: Stan Z. Li <Stan.ZQ.Li@westlake.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

NP-complete (Wang & Tian, 2011; Fu et al., 2022), deep learning techniques are introduced as an alternative.



nested structure   non-nested structure

Figure 2. Examples of nested and non-nested secondary structures.

Attempts to overcome the limitations of energy-based methods have motivated deep learning methods that predict RNA secondary structures in the absence of DP. SPOT-RNA (Singh et al., 2019) is a seminal work that ensembles ResNet (He et al., 2016) and LSTM (Hochreiter & Schmidhuber, 1997) and applies transfer learning to identify molecular recognition features. SPOT-RNA does not constrain the output space into valid RNA secondary structures, which degrades its generalization ability on new datasets (Jung et al.). E2Efold (Chen et al., 2019) employs an unrolled algorithm for constrained programming that post-processes the network output to satisfy the constraints. E2Efold introduces a convex relaxation to make the constrained optimization tractable, leading to possible structural constraint violations and poor generalization ability (Sato et al., 2021; Fu et al., 2022; Franke et al., 2023; 2022). RTfold (Jung et al.) utilizes the Fenchel-Young loss (Berthet et al., 2020) to enable differentiable discrete optimization with perturbations, but the approximation cannot guarantee the satisfaction of constraints. Developing an appropriate optimization that enforces the output to be valid becomes a crucial concern.

Since deep learning-based approaches cannot directly output valid RNA secondary structures, existing approaches usually formulate the problem into a constrained optimization problem and optimize the output of the model to fulfill specific constraints as closely as possible. However, these methods typically necessitate iterative optimization, leading to reduced efficiency. Moreover, the extensive optimization space involved does not ensure the complete satisfaction of these constraints. In this study, we introduce a novel perspective for predicting RNA secondary structures by reframing the challenge as a K-Rook problem. Recognizing the alignment between the solution spaces of the K-Rook problem and RNA secondary structure prediction, our objective is to identify the most compatible K-Rook solution for each RNA sequence. This is achieved by training the deep learning model on prior data to learn matching patterns.

Considering the high complexity of the solution space in the symmetric K-Rook problem, we introduced RFold, an innovative approach. This method utilizes a bi-dimensional optimization strategy, effectively decomposing the problem into separate row-wise and column-wise components. This decomposition significantly reduces the matching complexity, thereby simplifying the solving process while guaranteeing the validity of the output. We conduct extensive experiments to compare RFold with state-of-the-art methods on several benchmark datasets and show the superior performance of our proposed method. Moreover, RFold has faster inference efficiency than those methods due to its simplicity.

2. Related work

Comparative Sequence Analysis Comparative sequence analysis determines base pairs conserved among homologous sequences (Gardner & Giegerich, 2004; Knudsen & Hein, 2003; Gorodkin et al., 2001). ILM (Ruan et al., 2004) combines thermodynamic and mutual information content scores. Sankoff (Hofacker et al., 2004) merges the sequence alignment and maximal-pairing folding methods (Nussinov et al., 1978). Dynalign (Mathews & Turner, 2002) and Carnac (Touzet & Perriquet, 2004) are the subsequent variants of Sankoff algorithms. RNA forester (Hochsmann et al., 2003) introduces a tree alignment model for global and local alignments. However, the limited number of known RNA families (Nawrocki et al., 2015) impedes the development.

Energy-based Folding Algorithms When the structures consist only of nested base pairing, dynamic programming can predict the structure by minimizing energy. Early works include Vienna RNAfold (Lorenz et al., 2011), Mfold (Zuker, 2003), RNAstructure (Mathews & Turner, 2006), and CONTRAfold (Do et al., 2006). Faster implementations that speed up dynamic programming have been proposed, such as Vienna RNAplfold (Bernhart et al., 2006), LocalFold (Lange et al., 2012), and LinearFold (Huang et al., 2019). However, they cannot accurately predict structures with pseudoknots, as predicting the lowest free energy structures with pseudoknots is NP-complete (Lyngsø & Pedersen, 2000), making it difficult to improve performance.

Learning-based Folding Algorithms Deep learning methods have inspired approaches in bioengineering applications (Wu et al., 2024a;b; Lin et al., 2022; 2023; Tan et al., 2024; 2023). SPOT-RNA (Singh et al., 2019) is a seminal work that employs deep learning for RNA secondary structure prediction. SPOT-RNA2 (Singh et al., 2021) improves its predecessor by using evolution-derived sequence profiles and mutational coupling. Inspired by Raptor-X (Wang et al., 2017) and SPOT-Contact (Hanson et al., 2018), SPOT-RNA uses ResNet and bidirectional LSTM with a sigmoid function. MXfold (Akiyama et al., 2018) combines support vector machines and thermodynamic models. CDPfold (Zhang et al., 2019), DMFold (Wang et al., 2019), and MXFold2 (Sato et al., 2021) integrate deep learning techniques with energy-based methods. E2Efold (Chen et al., 2019) constrains the output to be valid by learning unrolled algorithms. However, its relaxation for making the optimization tractable may violate the constraints. UFold (Fu et al., 2022) introduces U-Net model to improve performance.

3. Background

3.1. Preliminaries

The primary structure of RNA is a sequence of nucleotide bases A, U, C, and G, arranged in order and represented as $\boldsymbol{X} = (x_1, ..., x_L)$, where each x_i denotes one of these bases. The secondary structure is the set of base pairings within the sequence, modeled as a sparse matrix $\boldsymbol{M} \in \{0,1\}^{L\times L}$, where $\boldsymbol{M}_{ij} = 1$ indicates a bond between bases i and j. The key challenges include (i) designing a model, characterized by parameters Θ, that captures the complex transformations from the sequence $\boldsymbol{X}$ to the pairing matrix $\boldsymbol{M}$ and (ii) correctly identifying the sparsity of the secondary structure, which is determined by the nature of RNA. Thus, the transformation $\mathcal{F}_\Theta : \boldsymbol{X} \mapsto \boldsymbol{M}$ is usually decomposed into two stages for capturing the sequence-to-structure relationship and optimizing the sparsity of the target matrix:

$$\mathcal{F}_\Theta := \mathcal{G}_{\theta_g} \circ \mathcal{H}_{\theta_h}, \tag{1}$$

where $\mathcal{H}_{\theta_h} : \boldsymbol{X} \mapsto \boldsymbol{H}$ represents the initial processing step, transforming the RNA sequence into an intermediate, unconstrained representation $\boldsymbol{H} \in \mathbb{R}^{L\times L}$. Subsequently, $\mathcal{G}_{\theta_g} : \boldsymbol{H} \mapsto \boldsymbol{M}$ parameterizes the optimization stage for the intermediate distribution into the final sparse matrix $\boldsymbol{M}$.

3.2. Constrained Optimization-based Approaches

The core problem of secondary structure prediction lies in sparsity identification. Numerous studies regard this task as a constrained optimization problem, seeking the optimal refinement mappings by gradient descent. Besides, keeping the hard constraints on RNA secondary structures is also essential, which ensures valid biological functions (Steeg, 1993). These constraints can be formally described as:

- (a) Only three types of nucleotide combinations can form base pairs: $\mathcal{B} := \{\mathrm{AU}, \mathrm{UA}\} \cup \{\mathrm{GC}, \mathrm{CG}\} \cup \{\mathrm{GU}, \mathrm{UG}\}$. For any base pair $x_i x_j$ where $x_i x_j \notin \mathcal{B}$, $\boldsymbol{M}_{ij} = 0$.
- (b) No sharp loop within three bases. For any adjacent bases within a distance of three nucleotides, they cannot form pairs with each other. For all $|i-j| < 4$, $\boldsymbol{M}_{ij} = 0$.
- (c) There can be at most one pair for each base. For all i and j, $\sum_{j=1}^{L} \boldsymbol{M}_{ij} \leqslant 1, \sum_{i=1}^{L} \boldsymbol{M}_{ij} \leqslant 1$.

The search for valid secondary structures is thus a quest for *symmetric* sparse matrices $\in \{0,1\}^{L\times L}$ that adhere to the constraints above. The first two constraints can be satisfied by defining a constraint matrix $\overline{\boldsymbol{M}}$ as: $\overline{\boldsymbol{M}}_{ij} := 1$ if $x_i x_j \in \mathcal{B}$ and $|i-j| \geqslant 4$, and $\overline{\boldsymbol{M}}_{ij} := 0$ otherwise. Addressing the third constraint is critical as it necessitates employing sparse optimization techniques. Therefore, our primary objective is to devise an effective sparse optimization strategy. This strategy is based on the symmetric inherent distribution $\boldsymbol{H}$ and $\boldsymbol{M}$, which support constraints (a) and (b), and additionally addresses constraint (c).

SPOT-RNA subtly enforces the principles of sparsity. It streamlines the pathway from the raw neural network output $\boldsymbol{H}$ by harnessing the $\mathrm{Sigmoid}$ function to distill a sparse pattern. The transformation applies a threshold to yield a binary sparse matrix. This process can be represented as:

$$\mathcal{G}(\boldsymbol{H}) = \mathbb{1}_{[\mathrm{Sigmoid}(\boldsymbol{H}) > s]}. \tag{2}$$

In this approach, a fixed threshold s of 0.5 is applied, typical for inducing sparsity. It omits complex constraints or extra parameters θ_g, simply using this cutoff to achieve sparse structure representations.

E2Efold introduces a non-linear transformation to the intermediate value $\widehat{\boldsymbol{M}} \in \mathbb{R}^{L\times L}$ and an additional regularization term $\|\widehat{\boldsymbol{M}}\|_1$.

$$\frac{1}{2}\left\langle \boldsymbol{H} - s, \mathcal{T}(\widehat{\boldsymbol{M}}) \right\rangle - \rho\|\widehat{\boldsymbol{M}}\|_1, \tag{3}$$

where $\mathcal{T}(\widehat{\boldsymbol{M}}) = \frac{1}{2}(\widehat{\boldsymbol{M}} \odot \widehat{\boldsymbol{M}} + (\widehat{\boldsymbol{M}} \odot \widehat{\boldsymbol{M}})^T) \odot \overline{\boldsymbol{M}}$ ensures symmetry and adherence to RNA base-pairing constraints (a) and (b), s is the log-ratio bias term set to $\log(9.0)$, and the ℓ_1 penalty $\rho\|\widehat{\boldsymbol{M}}\|_1$ promotes sparsity. To fulfill constraint (c), the objective is combined with conditions $\mathcal{T}(\widehat{\boldsymbol{M}})\mathbb{1} \leqslant \mathbb{1}$. Denote $\boldsymbol{\lambda} \in \mathbb{R}_+^L$ as the Lagrange multiplier, the formulation for the sparse optimization is expressed as:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \geqslant \mathbf{0}} \max_{\widehat{\boldsymbol{M}}} \; & \frac{1}{2}\left\langle \boldsymbol{H} - s, \mathcal{T}(\widehat{\boldsymbol{M}}) \right\rangle - \rho\|\widehat{\boldsymbol{M}}\|_1 \\ & - \left\langle \boldsymbol{\lambda}, \mathrm{ReLU}(\mathcal{T}(\widehat{\boldsymbol{M}})\mathbb{1} - \mathbb{1}) \right\rangle, \end{aligned} \tag{4}$$

In the training stage, the optimization objective is the output of score function $\mathcal{S}$ dependent on $\widehat{\boldsymbol{M}}$ and $\boldsymbol{H}$. It can be regarded as an optimization function $\mathcal{G}$ parameterized by θ_g:

$$\mathcal{G}_{\theta_g}(\boldsymbol{H}) = \mathcal{T}(\arg\max_{\widehat{\boldsymbol{M}} \in \mathbb{R}^{L\times L}} \mathcal{S}(\widehat{\boldsymbol{M}}, \boldsymbol{H})). \tag{5}$$

Although the complicated design to the constraints is explicitly formulated, the iterative updates may fall into suboptimal or invalid solutions. Besides, it requires additional parameters θ_g, making the model training complicated.

RTfold introduces a differentiable function that incorporates an additional Gaussian perturbation $\boldsymbol{W}$. The objective function is expressed as:

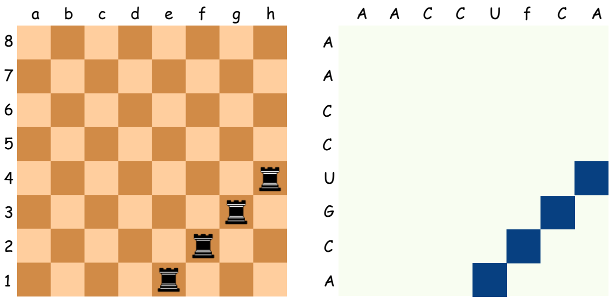
$$\min_{\widehat{\boldsymbol{M}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{T}\left(\boldsymbol{H} + \epsilon \boldsymbol{W}^{(i)}\right) - \widehat{\boldsymbol{M}} \tag{6}$$

where $\mathcal{T}$ denotes the non-linear transformation to constrain the initial output $\boldsymbol{H}$, and N is the number of random samples. The random perturbation $\boldsymbol{W}^{(i)}$ adjusts the distribution by leveraging the gradient during the optimization process. While RTFold designs an efficient differential objective function, the constraints imposed by the non-linear transformation on a noisy hidden distribution may lead to biologically implausible structures.

4. RFold

4.1. Probabilistic K-Rook Matching

The symmetric K-Rook arrangement (Riordan, 2014; Elkies & Stanley, 2011) is a classic combinatorial problem involving the placement of $K (K \leqslant L)$ non-attacking Rooks on an $L \times L$ chessboard, where the goal is to arrange the Rooks such that they form a symmetric pattern. The term 'non-attacking' means that no two Rooks are positioned in the same row or column. An interesting parallel can be drawn between this combinatorial scenario and the domain of RNA secondary structure prediction, as illustrated in Figure 3. This analogy stems from the conceptual similarity in the arrangement patterns required in both cases. The RNA sequence can be regarded as a chessboard of size L and the base pairs are the Rooks. The core problem is to determine an optimal arrangement of these base pairs.



(a) symmetric Rooks arrangement

(b) RNA secondary structure

Figure 3. The analogy between the symmetric K-Rook arrangement and the RNA secondary structure prediction.

Given a finite solution space $\mathcal{M}$ defined by the symmetric K-Rook arrangement, we reformulate our objective as a probabilistic matching problem. The goal is to find the most matching solution $\boldsymbol{M}^* \in \mathcal{M}$ for the given sequence $\boldsymbol{X}$. The optimal solution $\boldsymbol{M}^*$ is defined as:

$$\boldsymbol{M}^* = \arg \max_{\boldsymbol{M} \in \mathcal{M}} \mathcal{P}(\boldsymbol{M}|\boldsymbol{X}). \tag{7}$$

According to Bayes' theorem, the posterior probability can be represented as $\mathcal{P}(\boldsymbol{M}|\boldsymbol{X}) = \frac{\mathcal{P}(\boldsymbol{X}|\boldsymbol{M})\mathcal{P}(\boldsymbol{M})}{\mathcal{P}(\boldsymbol{X})}$. Since the denominator $P(\boldsymbol{X})$ is constant for all $\boldsymbol{M}$, and assuming that the solution space is finite and each solution within it is equally likely, we can adopt a uniform prior $\mathcal{P}(\boldsymbol{M})$ in this context. Therefore, maximizing the posterior probability is equivalent to maximizing the likelihood $\mathcal{P}(\boldsymbol{X}|\boldsymbol{M})$. This leads to the following equation:

$$\boldsymbol{M}^* = \arg \max_{\boldsymbol{M} \in \mathcal{M}} \mathcal{P}(\boldsymbol{X}|\boldsymbol{M}). \tag{8}$$

Therefore, our primary task becomes computing the likelihood $\mathcal{P}(\boldsymbol{X}|\boldsymbol{M})$ for the given sequence $\boldsymbol{X}$ under each possible solution $\boldsymbol{M}$.

4.2. Bi-dimensional Optimization

Computing the likelihood $\mathcal{P}(\boldsymbol{X}|\boldsymbol{M})$ directly poses significant challenges. To address this, we propose a bi-dimensional optimization strategy that simplifies the problem by decomposing it into row-wise and column-wise components. This approach is mathematically represented as:

$$\mathcal{P}(\boldsymbol{X}|\boldsymbol{M}) = \mathcal{P}(\boldsymbol{X}|\boldsymbol{R})\mathcal{P}(\boldsymbol{X}|\boldsymbol{C}), \tag{9}$$

where $\boldsymbol{M}$ is the product of the row-wise component $\boldsymbol{R} \in \mathbb{R}^{L \times L}$ and the column-wise component $\boldsymbol{C} \in \mathbb{R}^{L \times L}$, i.e., $\boldsymbol{M} = \boldsymbol{R} \odot \boldsymbol{C}$. Each component represents the optimal solution for the row-wise and column-wise matching problems, respectively. Importantly, the row-wise and column-wise components are independent, and the comprehensive solution for the entire problem is derived from the product of the optimal solutions for these two sub-problems.

Applying Bayes' theorem, for the row-wise component, we have $\mathcal{P}(\boldsymbol{R}|\boldsymbol{X}) = \frac{\mathcal{P}(\boldsymbol{X}|\boldsymbol{R})\mathcal{P}(\boldsymbol{R})}{\mathcal{P}(\boldsymbol{X})}$. Given that the solution space of $\boldsymbol{R}$ is both finite and valid, we can regard it as a uniform distribution. The analysis for the column-wise component, $\mathcal{P}(\boldsymbol{C}|\boldsymbol{X})$, follows a similar approach. Therefore, the optimal solution $\boldsymbol{M}^*$ can be represented as:

$$\begin{aligned} \boldsymbol{M}^* &= \arg \max_{\boldsymbol{R}, \boldsymbol{C}} \mathcal{P}(\boldsymbol{R}|\boldsymbol{X})\mathcal{P}(\boldsymbol{C}|\boldsymbol{X}) \\ &= \arg \max_{\boldsymbol{R}} \mathcal{P}(\boldsymbol{R}|\boldsymbol{X}) \arg \max_{\boldsymbol{C}} \mathcal{P}(\boldsymbol{C}|\boldsymbol{X}) \end{aligned} \tag{10}$$

The next phase involves establishing proxies for $\mathcal{P}(\boldsymbol{R}|\boldsymbol{X})$ and $\mathcal{P}(\boldsymbol{C}|\boldsymbol{X})$. To this end, we introduce the basic symmetric hidden distribution, $\widehat{\boldsymbol{H}} = (\boldsymbol{H} \odot \boldsymbol{H}^T) \odot \overline{\boldsymbol{M}}$. The row-wise and column-wise components are then derived by applying Softmax functions to $\widehat{\boldsymbol{H}}$, resulting in their respective probability distributions:

$$\mathcal{R}(\widehat{\boldsymbol{H}}) = \frac{\exp(\widehat{\boldsymbol{H}}_{ij})}{\sum_{k=1}^{L} \exp(\widehat{\boldsymbol{H}}_{ik})}, \mathcal{C}(\widehat{\boldsymbol{H}}) = \frac{\exp(\widehat{\boldsymbol{H}}_{ij})}{\sum_{k=1}^{L} \exp(\widehat{\boldsymbol{H}}_{kj})}. \tag{11}$$

The final output is the element-wise product of the row-wise component $\mathcal{R}(\widehat{\boldsymbol{H}})$ and the column-wise component $\mathcal{C}(\widehat{\boldsymbol{H}})$. This operation integrates the individual insights from both dimensions, leading to the optimized matrix $\boldsymbol{M}^*$:

$$\boldsymbol{M}^* = \arg \max \mathcal{R}(\widehat{\boldsymbol{H}}) \odot \arg \max \mathcal{C}(\widehat{\boldsymbol{H}}). \tag{12}$$

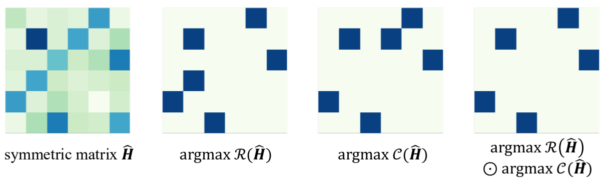

symmetric matrix $\widehat{\boldsymbol{H}}$
argmax $\mathcal{R}(\widehat{\boldsymbol{H}})$
argmax $\mathcal{C}(\widehat{\boldsymbol{H}})$
argmax $\mathcal{R}(\widehat{\boldsymbol{H}})$ $\odot$ argmax $\mathcal{C}(\widehat{\boldsymbol{H}})$

Figure 4. The visualization of $\arg \max \mathcal{R}(\widehat{\boldsymbol{H}}) \odot \arg \max \mathcal{C}(\widehat{\boldsymbol{H}})$.

As illustrated in Figure 4, we consider a random symmetric 6×6 matrix as an example. For simplicity, we disregard the constraints (a-b) from $\overline{\boldsymbol{M}}$. This example demonstrates the outputs of $\mathcal{R}(\cdot)$, $\mathcal{C}(\cdot)$, and their element-wise product $\mathcal{R}(\cdot) \odot \mathcal{C}(\cdot)$. The row-wise and column-wise components jointly select the value that has the maximum in both its row and column while keeping the output matrix symmetric.

Given the definition of $\widehat{\boldsymbol{H}} = (\boldsymbol{H} \odot \boldsymbol{H}^T) \odot \overline{\boldsymbol{M}}$, it is evident that $\widehat{\boldsymbol{H}}$ inherently forms a symmetric and non-negative matrix. Regarding optimization, the operation $\mathcal{R}(\widehat{\boldsymbol{H}}) \odot \mathcal{C}(\widehat{\boldsymbol{H}})$ can be equivalently simplified to optimizing $\frac{1}{2}(\mathcal{R}(\widehat{\boldsymbol{H}}) + \mathcal{C}(\widehat{\boldsymbol{H}}))$. This is because both approaches fundamentally aim to maximize the congruence between the row-wise and column-wise components of $\widehat{\boldsymbol{H}}$. The underlying reason for this equivalence is that both optimizing the Hadamard product and the arithmetic mean of $\mathcal{R}(\widehat{\boldsymbol{H}})$ and $\mathcal{C}(\widehat{\boldsymbol{H}})$ focus on reinforcing the alignment and coherence across the various dimensions of the matrix.

Moreover, examining the gradients of these operations sheds light on their computational efficiencies. The gradient of $\mathcal{R}(\widehat{\boldsymbol{H}}) \odot \mathcal{C}(\widehat{\boldsymbol{H}})$ entails a blend of partial derivatives interconnected via element-wise multiplication. It can be formally expressed as follows:

$$\begin{aligned} &\frac{\partial(\mathcal{R}(\widehat{\boldsymbol{H}}) \odot \mathcal{C}(\widehat{\boldsymbol{H}}))_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}} \\ =&\mathcal{C}(\widehat{\boldsymbol{H}})_{ij} \cdot \frac{\partial \mathcal{R}(\widehat{\boldsymbol{H}})_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}} + \mathcal{R}(\widehat{\boldsymbol{H}})_{ij} \cdot \frac{\partial \mathcal{C}(\widehat{\boldsymbol{H}})_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}}. \end{aligned} \tag{13}$$

In contrast, the gradient of $\frac{1}{2}(\mathcal{R}(\widehat{\boldsymbol{H}}) + \mathcal{C}(\widehat{\boldsymbol{H}}))$ is characterized by a straightforward sum of partial derivatives:

$$\frac{\partial(\mathcal{R}(\widehat{\boldsymbol{H}}) + \mathcal{C}(\widehat{\boldsymbol{H}}))_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}} = \frac{\partial \mathcal{R}(\widehat{\boldsymbol{H}})_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}} + \frac{\partial \mathcal{C}(\widehat{\boldsymbol{H}})_{ij}}{\partial \widehat{\boldsymbol{H}}_{ij}}. \tag{14}$$

Element-wise addition, as used in the latter, tends to be numerically more stable and less susceptible to issues like floating-point precision errors, which are more common in element-wise multiplication operations. This stability is particularly beneficial when dealing with large-scale matrices or when the gradients involve extreme values, where numerical instability can pose significant challenges.

The proposed simplification not only maintains the mathematical integrity of the optimization problem but also provides computational advantages, making it a desirable strategy in practical scenarios involving large and intricate datasets. Consequently, we define the overall loss function as the mean square error (MSE) between the averaged row-wise and column-wise components of $\widehat{\boldsymbol{H}}$ and the ground truth secondary structure $\boldsymbol{M}$:

$$\mathcal{L}(\boldsymbol{M}^*, \boldsymbol{M}) = \frac{1}{L^2} \left\| \frac{1}{2}(\mathcal{R}(\widehat{\boldsymbol{H}}) + \mathcal{C}(\widehat{\boldsymbol{H}})) - \boldsymbol{M} \right\|^2. \tag{15}$$

4.3. Practical Implementation

We identify the problem of predicting $\boldsymbol{H} \in \mathbb{R}^{L \times L}$ from the given sequence attention map $\widehat{\boldsymbol{Z}} \in \mathbb{R}^{L \times L}$ as an image-to-image segmentation problem and apply the U-Net model to extract pair-wise information, as shown in Figure 5.

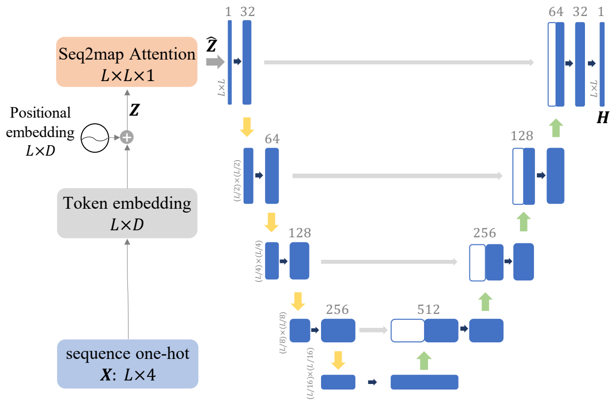


Figure 5. The overview model architecture of RFold.

To automatically learn informative representations from sequences, we propose a Seq2map attention module. Given a sequence in one-hot form $\boldsymbol{X} \in \mathbb{R}^{L \times 4}$, we first obtain the sum of the token embedding and positional embedding as the input of the Seq2map attention. We denote the input as $\boldsymbol{Z} \in \mathbb{R}^{L \times D}$ for convenience, where D is the hidden layer size of the token and positional embeddings.

Motivated by the recent progress in attention mechanisms (Vaswani et al., 2017; Choromanski et al., 2020; Katharopoulos et al., 2020; Hua et al., 2022), we aim to develop a highly effective sequence-to-map transformation based on pair-wise attention. We obtain the query $\boldsymbol{Q} \in \mathbb{R}^{L \times D}$ and key $\boldsymbol{K} \in \mathbb{R}^{L \times D}$ by applying per-dim scalars and offsets to $\boldsymbol{Z}$:

$$\begin{aligned} \boldsymbol{Q} &= \gamma_Q \boldsymbol{Z} + \beta_Q, \\ \boldsymbol{K} &= \gamma_K \boldsymbol{Z} + \beta_K, \end{aligned} \tag{16}$$

where $\gamma_Q, \gamma_K, \beta_Q, \beta_K \in \mathbb{R}^{L \times D}$ are learnable parameters.

Then, the pair-wise attention map is obtained by:

$$\bar{\boldsymbol{Z}} = \mathrm{ReLU}^2(\boldsymbol{Q}\boldsymbol{K}^T / L), \tag{17}$$

where ReLU^2 is an activation function that can be recognized as a simplified Softmax function in vanilla Transformers (So et al., 2021). The output of Seq2map is the gated representation of $\bar{\boldsymbol{Z}}$:

$$\widehat{\boldsymbol{Z}} = \bar{\boldsymbol{Z}} \odot \sigma(\bar{\boldsymbol{Z}}), \tag{18}$$

where $\sigma(\cdot)$ is the Sigmoid function that performs as a gate.

5. Experiments

We conduct experiments to compare our proposed RFold with state-of-the-art and commonly used approaches. Multiple experimental settings are taken into account, including standard structure prediction, generalization evaluation, large-scale benchmark evaluation, cross-family evaluation, pseudoknot prediction and inference time comparison. Detailed experimental setups can be found in the Appendix B.

5.1. Standard RNA Secondary Structure Prediction

Following (Chen et al., 2019), we split the RNAStralign dataset into train, validation, and test sets by stratified sampling. We report the results in Table 1. Energy-based methods achieve relatively weak F1 scores ranging from 0.420 to 0.633. Learning-based folding algorithms like E2Efold and UFold significantly improve performance by large margins, while RFold obtains even better performance among all the metrics. Moreover, RFold obtains about 8% higher precision than the state-of-the-art method. This suggests that our optimization strategy is strict to satisfy all the hard constraints for predicting valid structures.

Table 1. Results on RNAStralign test set. Results in bold and underlined are the top-1 and top-2 performances, respectively.

Method	Precision	Recall	F1
Mfold	0.450	0.398	0.420
RNAfold	0.516	0.568	0.540
RNAstructure	0.537	0.568	0.550
CONTRAfold	0.608	0.663	0.633
LinearFold	0.620	0.606	0.609
CDPfold	0.633	0.597	0.614
E2Efold	0.866	0.788	0.821
UFold	0.905	0.927	0.915
RFold	**0.981**	**0.973**	**0.977**

5.2. Generalization Evaluation

To verify the generalization ability of our proposed RFold, we directly evaluate the performance on another benchmark dataset ArchiveII using the pre-trained model on the RNAStralign training dataset. Following (Chen et al., 2019), we exclude RNA sequences in ArchiveII that have overlapping RNA types with the RNAStralign dataset for a fair comparison. The results are reported in Table 2.

It can be seen that traditional methods achieve F1 scores in the range of 0.545 to 0.842. A recent learning-based method, MXfold2, obtains an F1 score of 0.768, which is even lower than some energy-based methods. Another state-of-the-art learning-based method improves the performance to the F1 score of 0.905. RFold further improves the F1 score to 0.921, even higher than UFold. It is worth noting that RFold has a relatively lower result in the recall metric and a significantly higher result in the precision metric. The reason for this phenomenon may be the strict constraints of RFold. While none of the current learning-based methods can satisfy all the constraints we introduced in Sec. 3.2, the predictions of RFold are guaranteed to be valid. Thus, RFold may cover fewer pair-wise interactions, leading to a lower recall metric. However, the highest F1 score still suggests the great generalization ability of RFold.

Table 2. Results on ArchiveII dataset. Results in bold and underlined are the top-1 and top-2 performances, respectively.

Method	Precision	Recall	F1
Mfold	0.668	0.590	0.621
CDPfold	0.557	0.535	0.545
RNAfold	0.663	0.613	0.631
RNAstructure	0.664	0.606	0.628
CONTRAfold	0.696	0.651	0.665
LinearFold	0.724	0.605	0.647
RNAsoft	0.665	0.594	0.622
Eternafold	0.667	0.622	0.636
E2Efold	0.734	0.660	0.686
SPOT-RNA	0.743	0.726	0.711
MXfold2	0.788	0.760	0.768
Contextfold	0.873	0.821	0.842
RTfold	0.891	0.789	0.814
UFold	0.887	**0.928**	0.905
RFold	**0.938**	0.910	**0.921**

5.3. Large-scale Benchmark Evaluation

The bpRNA dataset is a large-scale benchmark, comprises fixed training (TR0), evaluation (VL0), and testing (TS0) sets. Following previous works (Singh et al., 2019; Sato et al., 2021; Fu et al., 2022), we train the model in bpRNA-TR0 and evaluate the performance on bpRNA-TS0 by using the best model learned from bpRNA-VL0. The detailed results can be found in Table 3.

Table 3. Results on bpRNA-TS0 set.

Method	Precision	Recall	F1
E2Efold	0.140	0.129	0.130
RNAstructure	0.494	0.622	0.533
RNAsoft	0.497	0.626	0.535
RNAfold	0.494	0.631	0.536
Mfold	0.501	0.627	0.538
Contextfold	0.529	0.607	0.546
LinearFold	0.561	0.581	0.550
MXfold2	0.519	0.646	0.558
Externafold	0.516	0.666	0.563
CONTRAfold	0.528	0.655	0.567
SPOT-RNA	0.594	0.693	0.619
UFold	0.521	0.588	0.553
RFold	**0.692**	0.635	**0.644**

RFold outperforms the prior state-of-the-art method, SPOT-RNA, by a notable 4.0% in terms of the F1 score. This improvement in the F1 score can be attributed to the consistently superior performance of RFold in the precision metric when compared to baseline models. However, it is important to note that the recall metric remains constrained, likely due to stringent constraints applied during prediction.

Table 4. Results on long-range bpRNA-TS0 set. Results in bold and underlined are the top-1 and top-2 performances, respectively.

Method	Precision	Recall	F1
Mfold	0.315	0.450	0.356
RNAfold	0.304	0.448	0.350
RNAstructure	0.299	0.428	0.339
CONTRAfold	0.306	0.439	0.349
LinearFold	0.281	0.355	0.305
RNAsoft	0.310	0.448	0.353
Externafold	0.308	0.458	0.355
SPOT-RNA	0.361	0.492	0.403
MXfold2	0.318	0.450	0.360
Contextfold	0.332	0.432	0.363
UFold	0.543	0.631	0.584
RFold	**0.803**	**0.765**	**0.701**

Following (Fu et al., 2022), we conduct an experiment on long-range interactions. Given a sequence of length L, the long-range base pairing is defined as the paired and unpaired bases with intervals longer than $L/2$. As shown in Table 4, RFold performs unexpectedly well on these long-range base pairing predictions and improves UFold in all metrics by large margins, demonstrating its strong predictive ability.

5.4. Cross-family Evaluation

Table 5. Results on bpRNA-new. Results in bold and underlined are the top-1 and top-2 performances, respectively.

Method	Precision	Recall	F1
E2Efold	0.047	0.031	0.036
SPOT-RNA	**0.635**	0.641	0.620
Contrafold	0.620	0.736	**0.661**
UFold	0.500	0.736	0.583
UFold + aug	0.570	**0.742**	0.636
RFold	0.614	0.619	0.616
RFold + aug	0.618	0.687	0.651

The bpRNA-new dataset is a cross-family benchmark dataset comprising 1,500 RNA families, presenting a significant challenge for pure deep learning approaches. UFold, for instance, relies on the thermodynamic method Contrafold for data augmentation to achieve satisfactory results. As shown in Table 5, the standard UFold achieves an F1 score of 0.583, while RFold reaches 0.616. When the same data augmentation technique is applied, UFold's performance increases to 0.636, whereas RFold yields a score of 0.651. This places RFold second only to the thermodynamics-based method, Contrafold, in terms of F1 score.

5.5. Predict with Pseudoknots

Following E2Efold (Chen et al., 2019), we consider a sequence to be a true positive if it is correctly identified as containing a pseudoknot. For this analysis, we extracted all sequences featuring pseudoknots from the RNAStralign test dataset and assessed their predictive accuracy. The results of this analysis are summarized in the following table:

Table 6. Results on RNA structures with pseudoknots.

Method	Precision	Recall	F1 Score
RNAstructure	0.778	0.761	0.769
SPOT-RNA	0.677	0.978	0.800
E2Efold	0.844	0.990	0.911
UFold	0.962	0.990	0.976
RFold	**0.971**	**0.993**	**0.982**

RFold demonstrates superior performance compared to its counterparts across all evaluated metrics, i.e., precision, recall, and F1 score. This consistent outperformance across multiple dimensions of accuracy underscores the efficacy and robustness of the RFold approach in predicting RNA structures with pseudoknots.

5.6. Inference Time Comparison

We compared the running time of various methods for predicting RNA secondary structures using the RNAStralign testing set with the same experimental setting and the hardware environment as in (Fu et al., 2022). The results are presented in Table 7, which shows the average inference time per sequence. The fastest energy-based method, LinearFold, takes about 0.43s for each sequence. The learning-based baseline, UFold, takes about 0.16s. RFold has the highest inference speed, costing only about 0.02s per sequence. In particular, RFold is about eight times faster than UFold and sixteen times faster than MXfold2.

Table 7. Inference time on the RNAStralign test set. Results in bold and underlined are the top-1 and top-2 performances, respectively.

Method	Time
CDPfold (Tensorflow)	300.11 s
RNAstructure (C)	142.02 s
CONTRAfold (C++)	30.58 s
Mfold (C)	7.65 s
Eternafold (C++)	6.42 s
RNAsoft (C++)	4.58 s
RNAfold (C)	0.55 s
LinearFold (C++)	0.43 s
SPOT-RNA(Pytorch)	77.80 s (GPU)
E2Efold (Pytorch)	0.40 s (GPU)
MXfold2 (Pytorch)	0.31 s (GPU)
UFold (Pytorch)	0.16 s (GPU)
RFold (Pytorch)	**0.02 s** (GPU)

5.7. Ablation Study

Bi-dimensional Optimization To validate the effectiveness of our proposed bi-dimensional optimization strategy, we conduct an experiment that replaces them with other optimization methods. The results are summarized in Table 8, where RFold-E and RFold-S denote our model with the optimization strategies of E2Efold and SPOT-RNA, respectively. While precision, recall, and F1 score are evaluated at base-level, we report the validity which is a sample-level metric evaluating whether the predicted structure satisfies all the constraints. It can be seen that though RFold-E has comparable performance in the first three metrics with ours, many of its predicted structures are invalid. The optimization strategy of SPOT-RNA has incorporated no constraint that results in its low validity. Moreover, its strategy seems to not fit our model well, which may be caused by the simplicity of our proposed RFold model.

Table 8. Ablation study on different optimization strategies (RNAStralign testing set).

Method	Precision	Recall	F1	Validity
RFold	**0.981**	**0.973**	**0.977**	**100.00%**
RFold-E	0.888	0.906	0.896	50.31%
RFold-S	0.223	0.988	0.353	0.00%

Seq2map Attention We also conduct an experiment to evaluate the proposed Seq2map attention. We replace the Seq2map attention with the hand-crafted features from UFold and the outer concatenation from SPOT-RNA, which are denoted as RFold-U and RFold-SS, respectively. In addition to performance metrics, we also report the average inference time for each RNA sequence to evaluate the model complexity. We summarize the result in Table 9. It can be seen that RFold-U takes much more inference time than our RFold and RFold-SS due to the heavy computational cost when loading and learning from hand-crafted features. Moreover, it is surprising to find that RFold-SS has a little better performance than RFold-U, with the least inference time for its simple outer concatenation operation. However, neither RFold-U nor RFold-SS can provide informative representations like our proposed Seq2map attention. With comparable inference time with the simplest RFold-SS, our RFold outperforms baselines by large margins.

Table 9. Ablation study on different pre-processing strategies (RNAStralign testing set).

Method	Precision	Recall	F1	Time
RFold	**0.981**	**0.973**	**0.977**	0.0167
RFold-U	0.875	0.941	0.906	0.0507
RFold-SS	0.886	0.945	0.913	**0.0158**

Row-wise and Column-wise Componenets We conducted comprehensive ablation studies on the row-wise and column-wise components of our proposed model, RFold, by modifying the inference mechanism using pre-trained checkpoints. These studies were meticulously designed to isolate and understand the individual contributions of these components to our model's performance in RNA secondary structure prediction. The results, presented across three datasets—RNAStralign (Table 10), ArchiveII (Table 11), and bpRNA-TS0 (Table 12)—highlight two key findings: (i) Removing both the row-wise and column-wise components leads to a substantial drop in the model's performance, underscoring their pivotal role within our model's architecture. This dramatic reduction in effectiveness clearly demonstrates that both components are integral to achieving high accuracy. The significant decline in performance when these components are omitted highlights their essential function in capturing the complex dependencies within RNA sequences. (ii) The performance metrics when isolating either the row-wise or column-wise components are remarkably similar across all datasets. This uniformity suggests that the training process, which incorporates row-wise and column-wise softmax functions, likely yields symmetric outputs. Consequently, this symmetry implies that each component contributes in an almost equal measure to the model's overall predictive capacity.

Table 10. Ablation study on row-wise and column-wise components (RNAStralign testing set).

Method	Precision	Recall	F1	Validity
RFold	0.981	0.973	0.977	100.00%
RFold w/o C	0.972	0.975	0.973	75.99%
RFold w/o R	0.972	0.975	0.973	75.99%
RFold w/o R,C	0.016	0.031	0.995	0.00%

Table 11. Ablation study on row-wise and column-wise components (ArchiveII).

Method	Precision	Recall	F1	Validity
RFold	0.938	0.910	0.921	100.00%
RFold w/o C	0.919	0.914	0.914	49.14%
RFold w/o R	0.919	0.914	0.914	49.14%
RFold w/o R,C	0.013	0.997	0.025	0.00%

Table 12. Ablation study on row-wise and column-wise components (bpRNA-TS0).

Method	Precision	Recall	F1	Validity
RFold	0.693	0.635	0.644	100.00%
RFold w/o C	0.652	0.651	0.637	12.97%
RFold w/o R	0.652	0.651	0.637	12.97%
RFold w/o R,C	0.021	0.995	0.040	0.00%

5.8. Visualization

We visualize two examples predicted by RFold and UFold in Figure 6. The corresponding F1 scores are denoted at the bottom right of each plot. The first row of secondary structures is a simple example of a nested structure. It can be seen that UFold may fail in such a case. The second row of secondary structures is much more difficult that contains over 300 bases of the non-nested structure. While UFold fails in such a complex case, RFold can predict the structure accurately. Due to the limited space, we provide more visualization comparisons in Appendix D.

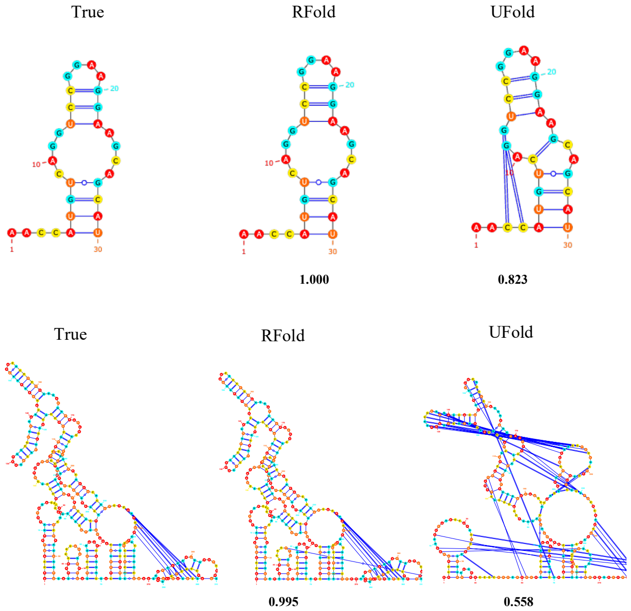


Figure 6. Visualization of the true and predicted structures.

6. Conclusion

In this study, we reformulate RNA secondary structure prediction as a K-Rook problem, thus transforming the prediction process into probabilistic matching. Subsequently, we introduce RFold, an efficient learning-based model, which utilizes a bidimensional optimization strategy to decompose the probabilistic matching into row-wise and column-wise components, simplifying the solving process while guaranteeing the validity of the output. Comprehensive experiments demonstrate that RFold achieves competitive performance with faster inference speed.

The limitations of RFold primarily revolve around its stringent constraints. This strictness in constraints implies that RFold is cautious in predicting interactions, leading to higher precision but possibly at the cost of missing some true interactions. Though we have provided a naive solution in Appendix C, it needs further studies to obtain a better strategy that leads to more balanced precision-recall trade-offs and more comprehensive structural predictions.

Acknowledgements

This work was supported by the Science & Technology Innovation 2030 Major Program Project No. 2021ZD0150100, National Natural Science Foundation of China Project No. U21A20427, Project No. WU2022A009 from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University, and Project No. WU2023C019 from the Westlake University Industries of the Future Research. Finally, we thank the Westlake University HPC Center for providing part of the computational resources.

Impact Statement

RFold is the first learning-based method that guarantees the validity of predicted RNA secondary structures. Its capability to ensure accurate predictions. It can be a valuable tool for biologists to study the structure and function of RNA molecules. Additionally, RFold stands out for its speed, significantly surpassing previous methods, marking it as a promising avenue for future developments in this field. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Akiyama, M., Sato, K., and Sakakibara, Y. A max-margin training of rna secondary structure prediction integrated with the thermodynamic model. *Journal of bioinformatics and computational biology*, 16(06):1840025, 2018.

Andronescu, M., Aguirre-Hernandez, R., Condon, A., and Hoos, H. H. Rnasoft: a suite of rna secondary structure prediction and design software tools. *Nucleic acids research*, 31(13):3416–3422, 2003.

Bellaousov, S., Reuter, J. S., Seetin, M. G., and Mathews, D. H. Rnastructure: web servers for rna secondary structure prediction and analysis. *Nucleic acids research*, 41 (W1):W471–W474, 2013.

Bernhart, S. H., Hofacker, I. L., and Stadler, P. F. Local rna base pairing probabilities in large sequences. *Bioinformatics*, 22(5):614–615, 2006.

Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., and Bach, F. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

Chen, X., Li, Y., Umarov, R., Gao, X., and Song, L. Rna secondary structure prediction by learning unrolled algorithms. In *International Conference on Learning Representations*, 2019.

Cheong, H.-K., Hwang, E., Lee, C., Choi, B.-S., and Cheong, C. Rapid preparation of rna samples for nmr spectroscopy and x-ray crystallography. *Nucleic acids research*, 32(10):e84–e84, 2004.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2020.

Do, C. B., Woods, D. A., and Batzoglou, S. Contrafold: Rna secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, 2006.

Elkies, N. and Stanley, R. P. Chess and mathematics. *Recuperado el*, 11, 2011.

Fallmann, J., Will, S., Engelhardt, J., Grüning, B., Backofen, R., and Stadler, P. F. Recent advances in rna folding. *Journal of biotechnology*, 261:97–104, 2017.

Fica, S. M. and Nagai, K. Cryo-electron microscopy snapshots of the spliceosome: structural insights into a dynamic ribonucleoprotein machine. *Nature structural & molecular biology*, 24(10):791–799, 2017.

Franke, J., Runge, F., and Hutter, F. Probabilistic transformer: Modelling ambiguities and distributions for rna folding and molecule design. *Advances in Neural Information Processing Systems*, 35:26856–26873, 2022.

Franke, J. K., Runge, F., and Hutter, F. Scalable deep learning for rna secondary structure prediction. *arXiv preprint arXiv:2307.10073*, 2023.

Fu, L., Cao, Y., Wu, J., Peng, Q., Nie, Q., and Xie, X. Ufold: fast and accurate rna secondary structure prediction with deep learning. *Nucleic acids research*, 50(3):e14–e14, 2022.

Fürtig, B., Richter, C., Wöhnert, J., and Schwalbe, H. Nmr spectroscopy of rna. *ChemBioChem*, 4(10):936–962, 2003.

Gardner, P. P. and Giegerich, R. A comprehensive comparison of comparative rna structure prediction approaches. *BMC bioinformatics*, 5(1):1–18, 2004.

Gardner, P. P., Daub, J., Tate, J. G., Nawrocki, E. P., Kolbe, D. L., Lindgreen, S., Wilkinson, A. C., Finn, R. D., Griffiths-Jones, S., Eddy, S. R., et al. Rfam: updates to the rna families database. *Nucleic acids research*, 37 (suppl_1):D136–D140, 2009.

Gorodkin, J., Stricklin, S. L., and Stormo, G. D. Discovering common stem–loop motifs in unaligned rna sequences. *Nucleic Acids Research*, 29(10):2135–2144, 2001.

Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S. R. Rfam: an rna family database. *Nucleic acids research*, 31(1):439–441, 2003.

Gutell, R. R., Lee, J. C., and Cannone, J. J. The accuracy of ribosomal rna comparative structure models. *Current opinion in structural biology*, 12(3):301–310, 2002.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y., and Zhou, Y. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 34(23):4039–4045, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hochsmann, M., Toller, T., Giegerich, R., and Kurtz, S. Local similarity in rna secondary structures. In *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, pp. 159–168. IEEE, 2003.

Hofacker, I. L., Bernhart, S. H., and Stadler, P. F. Alignment of rna base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227, 2004.

Hua, W., Dai, Z., Liu, H., and Le, Q. Transformer quality in linear time. In *International Conference on Machine Learning*, pp. 9099–9117. PMLR, 2022.

Huang, L., Zhang, H., Deng, D., Zhao, K., Liu, K., Hendrix, D. A., and Mathews, D. H. Linearfold: linear-time approximate rna folding by 5'-to-3'dynamic programming and beam search. *Bioinformatics*, 35(14):i295–i304, 2019.

Iorns, E., Lord, C. J., Turner, N., and Ashworth, A. Utilizing rna interference to enhance cancer drug discovery. *Nature reviews Drug discovery*, 6(7):556–568, 2007.

Jung, A. J., Lee, L. J., Gao, A. J., and Frey, B. J. Rtfold: Rna secondary structure prediction using deep learning with domain inductive bias.

Kalvari, I., Nawrocki, E. P., Ontiveros-Palacios, N., Argasinska, J., Lamkiewicz, K., Marz, M., Griffiths-Jones, S., Toffano-Nioche, C., Gautheret, D., Weinberg, Z., et al. Rfam 14: expanded coverage of metagenomic, viral and microrna families. *Nucleic Acids Research*, 49(D1):D192–D200, 2021.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

Knudsen, B. and Hein, J. Pfold: Rna secondary structure prediction using stochastic context-free grammars. *Nucleic acids research*, 31(13):3423–3428, 2003.

Lange, S. J., Maticzka, D., Möhl, M., Gagnon, J. N., Brown, C. M., and Backofen, R. Global or local? predicting secondary structure and accessibility in mrnas. *Nucleic acids research*, 40(12):5215–5226, 2012.

Lin, H., Huang, Y., Liu, M., Li, X. C., Ji, S., and Li, S. Z. Diffbp: Generative diffusion of 3d molecules for target protein binding. *ArXiv*, abs/2211.11214, 2022. URL https://api.semanticscholar.org/CorpusID:253734621.

Lin, H., Huang, Y., Zhang, H., Wu, L., Li, S., Chen, Z., and Li, S. Z. Functional-group-based diffusion for pocket-specific molecule generation and elaboration. *ArXiv*, abs/2306.13769, 2023. URL https://api.semanticscholar.org/CorpusID:259251644.

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. Viennarna package 2.0. *Algorithms for molecular biology*, 6(1):1–14, 2011.

Lyngsø, R. B. and Pedersen, C. N. Rna pseudoknot prediction in energy-based models. *Journal of computational biology*, 7(3-4):409–427, 2000.

Mathews, D. H. and Turner, D. H. Dynalign: an algorithm for finding the secondary structure common to two rna sequences. *Journal of molecular biology*, 317(2):191–203, 2002.

Mathews, D. H. and Turner, D. H. Prediction of rna secondary structure by free energy minimization. *Current opinion in structural biology*, 16(3):270–278, 2006.

Nawrocki, E. P., Burge, S. W., Bateman, A., Daub, J., Eberhardt, R. Y., Eddy, S. R., Floden, E. W., Gardner, P. P., Jones, T. A., Tate, J., et al. Rfam 12.0: updates to the rna families database. *Nucleic acids research*, 43(D1): D130–D137, 2015.

Nicholas, R. and Zuker, M. Unafold: Software for nucleic acid folding and hybridization. *Bioinformatics*, 453:3–31, 2008.

Nussinov, R., Pieczenik, G., Griggs, J. R., and Kleitman, D. J. Algorithms for loop matchings. *SIAM Journal on Applied mathematics*, 35(1):68–82, 1978.

Riordan, J. An introduction to combinatorial analysis. 2014.

Rivas, E. The four ingredients of single-sequence rna secondary structure prediction. a unifying perspective. *RNA biology*, 10(7):1185–1196, 2013.

Ruan, J., Stormo, G. D., and Zhang, W. An iterated loop matching approach to the prediction of rna secondary structures with pseudoknots. *Bioinformatics*, 20(1):58–66, 2004.

Sato, K., Akiyama, M., and Sakakibara, Y. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications*, 12(1):1–9, 2021.

Seetin, M. G. and Mathews, D. H. Rna structure prediction: an overview of methods. *Bacterial regulatory RNA*, pp. 99–122, 2012.

Singh, J., Hanson, J., Paliwal, K., and Zhou, Y. Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature communications*, 10(1):1–13, 2019.

Singh, J., Paliwal, K., Zhang, T., Singh, J., Litfin, T., and Zhou, Y. Improved rna secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, 37(17):2589–2600, 2021.

Sloma, M. F. and Mathews, D. H. Exact calculation of loop formation probability identifies folding motifs in rna secondary structures. *RNA*, 22(12):1808–1818, 2016.

So, D., Mańke, W., Liu, H., Dai, Z., Shazeer, N., and Le, Q. V. Searching for efficient transformers for language modeling. *Advances in Neural Information Processing Systems*, 34:6010–6022, 2021.

Steeg, E. W. Neural networks, adaptive optimization, and rna secondary structure prediction. *Artificial intelligence and molecular biology*, pp. 121–160, 1993.

Szikszai, M., Wise, M. J., Datta, A., Ward, M., and Mathews, D. Deep learning models for rna secondary structure prediction (probably) do not generalise across families. *bioRxiv*, 2022.

Tan, C., Zhang, Y., Gao, Z., Hu, B., Li, S., Liu, Z., and Li, S. Z. Hierarchical data-efficient representation learning for tertiary structure-based rna design. In *The Twelfth International Conference on Learning Representations*, 2023.

Tan, C., Gao, Z., Wu, L., Xia, J., Zheng, J., Yang, X., Liu, Y., Hu, B., and Li, S. Z. Cross-gate mlp with protein complex invariant embedding is a one-shot antibody designer.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15222–15230, 2024.

Tan, Z., Fu, Y., Sharma, G., and Mathews, D. H. Turbofold ii: Rna structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic acids research*, 45(20):11570–11581, 2017.

Touzet, H. and Perriquet, O. Carnac: folding families of related rnas. *Nucleic acids research*, 32(suppl_2):W142–W145, 2004.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, L., Liu, Y., Zhong, X., Liu, H., Lu, C., Li, C., and Zhang, H. Dmfold: A novel method to predict rna secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers in genetics*, 10:143, 2019.

Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1): e1005324, 2017.

Wang, X. and Tian, J. Dynamic programming for np-hard problems. *Procedia Engineering*, 15:3396–3400, 2011.

Wayment-Steele, H. K., Kladwang, W., Strom, A. I., Lee, J., Treuille, A., Participants, E., and Das, R. Rna secondary structure packages evaluated and improved by high-throughput experiments. *BioRxiv*, pp. 2020–05, 2021.

Wu, L., Huang, Y., Tan, C., Gao, Z., Hu, B., Lin, H., Liu, Z., and Li, S. Z. Psc-cpi: Multi-scale protein sequence-structure contrasting for efficient and generalizable compound-protein interaction prediction. *arXiv preprint arXiv:2402.08198*, 2024a.

Wu, L., Tian, Y., Huang, Y., Li, S., Lin, H., Chawla, N. V., and Li, S. Z. Mape-ppi: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. *arXiv preprint arXiv:2402.14391*, 2024b.

Xu, X. and Chen, S.-J. Physics-based rna structure prediction. *Biophysics reports*, 1(1):2–13, 2015.

Zakov, S., Goldberg, Y., Elhadad, M., and Ziv-Ukelson, M. Rich parameterization improves rna structure prediction. *Journal of Computational Biology*, 18(11):1525–1542, 2011.

Zhang, H., Zhang, C., Li, Z., Li, C., Wei, X., Zhang, B., and Liu, Y. A new method of rna secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in genetics*, 10:467, 2019.

Zuker, M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13): 3406–3415, 2003.

A. Comparison of mainstream RNA secondary structure prediction methods

We compare our proposed method RFold with several other leading RNA secondary structure prediction methods and summarize the results in Table 13. RFold satisfies all three constraints (a)-(c) for valid RNA secondary structures, while the other methods do not fully meet some of the constraints. RFold utilizes a sequence-to-map attention mechanism to capture long-range dependencies, whereas SPOT-RNA simply concatenates pairwise sequence information and E2Efold/UFold uses hand-crafted features. In terms of prediction accuracy on the RNAStralign benchmark test set, RFold achieves the best F1 score of 0.977, outperforming SPOT-RNA, E2Efold and UFold by a large margin. Regarding the average inference time, RFold is much more efficient and requires only 0.02 seconds to fold the RNAStralign test sequences. In summary, RFold demonstrates superior performance over previous methods for RNA secondary structure prediction in both accuracy and speed.

B. Experimental Details

Datasets We use three benchmark datasets: (i) RNAStralign (Tan et al., 2017), one of the most comprehensive collections of RNA structures, is composed of 37,149 structures from 8 RNA types; (ii) ArchiveII (Sloma & Mathews, 2016), a widely used benchmark dataset in classical RNA folding methods, containing 3,975 RNA structures from 10 RNA types; (iii) bpRNA (Singh et al., 2019), is a large scale benchmark dataset, containing 102,318 structures from 2,588 RNA types. (iv) bpRNA-new (Sato et al., 2021), derived from Rfam 14.2 (Kalvari et al., 2021), containing sequences from 1500 new RNA families.

Baselines We compare our proposed RFold with baselines including energy-based folding methods such as Mfold (Zuker, 2003), RNAsoft (Andronescu et al., 2003), RNAfold (Lorenz et al., 2011), RNAstructure (Mathews & Turner, 2006), CONTRAfold (Do et al., 2006), Contextfold (Zakov et al., 2011), and LinearFold (Huang et al., 2019); learning-based folding methods such as SPOT-RNA (Singh et al., 2019), Externafold (Wayment-Steele et al., 2021), E2Efold (Chen et al., 2019), MXfold2 (Sato et al., 2021), and UFold (Fu et al., 2022).

Metrics We evaluate the performance by precision, recall, and F1 score, which are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{F1} = 2\,\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{19}$$

where TP, FP, and FN denote true positive, false positive and false negative, respectively.

Implementation details Following the same experimental setting as (Fu et al., 2022), we train the model for 100 epochs with the Adam optimizer. The learning rate is 0.001, and the batch size is 1 for sequences with different lengths.

C. Discussion on Abnormal Samples

Although we have illustrated three hard constraints in 3.2, there exist some abnormal samples that do not satisfy these constraints in practice. We have analyzed the datasets used in this paper and found that there are some abnormal samples in the testing set that do not meet these constraints. The ratio of valid samples in each dataset is summarized in the table below:

As shown in Table 8, RFold forces the validity to be 100.00%, while other methods like E2Efold only achieve about 50.31%. RFold is more accurate than other methods in reflecting the real situation.

Nevertheless, we provide a soft version of RFold to relax the strict constraints. A possible solution to relax the rigid procedure is to add a checking mechanism before the Argmax function in the inference. Specifically, if the confidence given by the Softmax is low, we do not perform Argmax and assign more base pairs. It can be implemented as the following pseudo-code:

```
y_pred = row_col_softmax(y)
int_one = row_col_argmax(y_pred)

# get the confidence for each position
conf = y_pred * int_one
all_pos = conf > 0.0

# select reliable position
conf_pos = conf > thr1

# select unreliable position with the full row and column
uncf_pos = get_unreliable_pos(all_pos, conf_pos)

# assign "1" for the positions with the confidence higher than thr2
# note that thr2 < thr1
y_pred[uncf_pos] = (y_pred[uncf_pos] > thr2).float()
int_one[uncf_pos] = y_pred[uncf_pos]
```

We conduct experiments to compare the soft-RFold and the original version of RFold in the RNAStralign dataset. The results are summarized in the Table 15. It can be seen that soft-RFold improves the recall metric by a small margin. The minor improvement may be because the number of abnormal samples is small. We then select those samples that do not obey the three constraints to further analyze the performance. The total number of such samples is 179. It can be seen that soft-RFold can deal with abnormal samples well. The improvement of the recall metric is more obvious.

Table 13. Comparison between RNA secondary structure prediction methods and RFold.

Method	SPOT-RNA	E2Efold	UFold	RFold
pre-processing	pairwise concat	pairwise concat	hand-crafted	seq2map attention
optimization approach	×	unrolled algorithm	unrolled algorithm	bi-dimensional optimization
constraint (a)	×	✓	✓	✓
constraint (b)	×	✓	✓	✓
constraint (c)	×	×	×	✓
F1 score	0.711	0.821	0.915	**0.977**
Inference time	77.80 s	0.40 s	0.16 s	**0.02** s

Table 14. The ratio of valid samples in the datasets.

Dataset	RNAStralign	ArchiveII	bpRNA
Validity	93.05%	96.03%	96.51%

Table 15. The results of soft-RFold and RFold on the RNAStralign.

Method	Precision	Recall	F1
RFold	0.981	0.973	0.977
soft-RFold	0.978	0.974	0.976

Table 16. The results of soft-RFold and RFold on the abnormal samples on the RNAStralign.

Method	Precision	Recall	F1
RFold	0.956	0.860	0.905
soft-RFold	0.949	0.889	0.918

D. Visualization

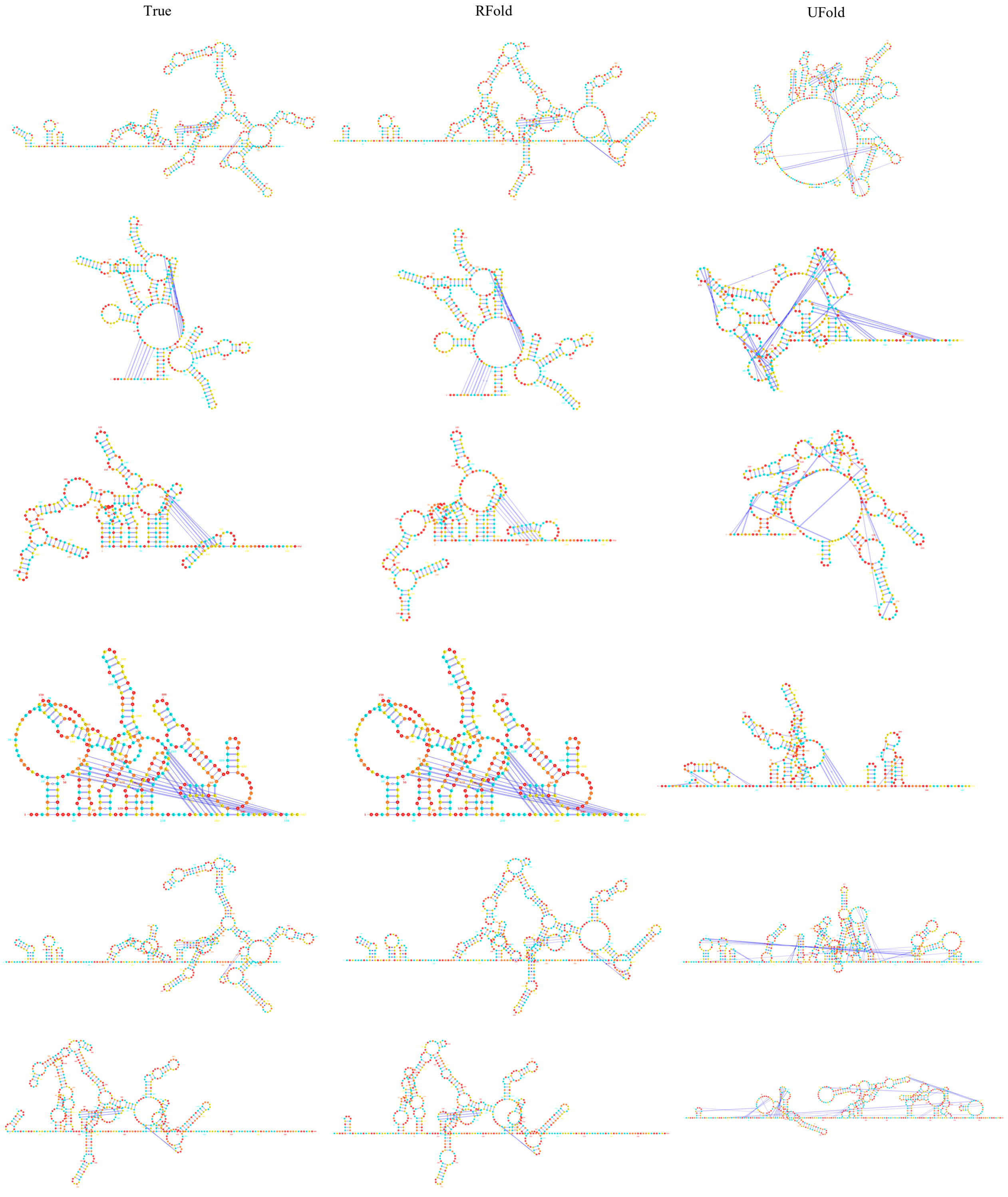


Figure 7. Visualization of the true and predicted structures.