TOWARDS GENERATIVE RNA DESIGN WITH TERTIARY INTERACTIONS

Sharat Patil*
Department of Computer Science
University of Freiburg
patilsh@cs.uni-freiburg.de

Frederic Runge
Department of Computer Science
University of Freiburg
runget@cs.uni-freiburg.de

Jörg K.H. Franke
Department of Computer Science
University of Freiburg
frankej@cs.uni-freiburg.de

Frank Hutter
Department of Computer Science
University of Freiburg
fh@cs.uni-freiburg.de

ABSTRACT

The design of RNAs that fulfill desired functions is one of the major challenges in computational biology. The function of an RNA molecule depends on its structure and a strong structure-to-function relationship is already achieved on the secondary structure level of RNA. Therefore, computational RNA design is often interpreted as the inversion of a folding algorithm: Given a target secondary structure, find an RNA sequence that folds into the desired structure. However, existing RNA design approaches cannot invert state-of-the-art folding algorithms because they can only predict a limited set of base interactions. In this work, we propose *RNAinformer*, a novel generative transformer based approach to the inverse RNA folding problem. Leveraging axial attention, we are able to process secondary structures represented as adjacency matrices, which allows us to invert state-of-the-art folding algorithms. Consequently, RNAinformer is the first model capable of designing RNAs from secondary structures without base pair restrictions. We demonstrate RNAinformer's strong performance across different RNA design benchmarks and showcase its novelty by inverting a state-of-the-art deep learning based secondary structure prediction algorithm.

1 INTRODUCTION

Ribonucleic acid (RNA) is one of the major regulatory molecules inside the cells of living organisms with key roles during differentiation and development (Morris & Mattick, 2014). RNAs fold hierarchically (Tinoco Jr & Bustamante, 1999) and the structure is key to their function: Base interactions via hydrogen bonds result in a fast formation of a *secondary structure*, with tertiary interactions stabilizing the formation of the final 3D shape (Vicens & Kieft, 2022). A strong structure-to-function relationship is already achieved on a secondary structure level (Hammer et al., 2019), and therefore, RNA secondary structure prediction recently got into the focus of the deep learning community, achieving state-of-the-art results (Singh et al., 2019; Fu et al., 2022; Chen et al., 2022; Franke et al., 2022; 2024). Compared to more traditional methods, these algorithms predict an $L \times L$ adjacency matrix representation of the secondary structure instead of the commonly used but less expressive dot-bracket string notation (Hofacker et al., 1994). This has the advantage that they are not limited to the prediction of specific kinds of base pairs but can predict non-Watson-Crick interactions, pseudoknots (Staple & Butcher, 2005), as well as base multiplets (nucleotides that pair with more than one other nucleotide) (Singh et al., 2019), which all play significant roles for RNA structures and functions (Reyes et al., 2009; Vicens & Kieft, 2022).

Structure-based RNA design considers the inverse problem: Given a target structure, find an RNA primary sequence that folds into the desired structure. It is thus intricately tied to RNA folding.

*Corresponding author.

However, there is currently no structure-based RNA design algorithm available that can invert state-of-the-art deep learning-based secondary structure prediction algorithms, which could clearly lead to better designs.

In this work, we propose *RNAinformer*, the first inverse RNA folding algorithm that is capable of designing RNAs while considering all kinds of base interactions. We show that a vanilla transformer architecture, enhanced with axial attention inspired by the RNAformer (Franke et al., 2024), can reliably design RNAs in different settings, including RNA design with non-canonical interactions, pseudoknots, and base multiplets. We see our main contributions as follows:

- We propose *RNAinformer*, a novel generative transformer model for the inverse RNA folding problem. Using axial attention, our model is the first RNA design algorithm that can design RNAs from secondary structures with all types of base interactions.
- We show that our model outperforms existing algorithms on nested and pseudoknotted structures, while further being capable of designing sequences that form base multiplets.

2 Related Work

Traditional Methods The problem of computational RNA design was first introduced as the inverse RNA folding problem by Hofacker et al. (1994). Since then, different methods were proposed for solving the problem using approaches like local search (Hofacker et al., 1994; Andronescu et al., 2004), constraint programming (Garcia-Martin et al., 2013; 2015; Minuesa et al., 2021), or evolutionary methods (Esmaili-Taheri et al., 2014; Esmaili-Taheri & Ganjtabesh, 2015). However, in contrast to our approach, these methods are limited to the design of nested structures, typically considering canonical base pairs only.

Learning Based Approaches More recently, RNA design was also approached with learning based methods. One line of research use human priors to design RNAs based on player strategies obtained from the online gaming platform Eterna (Shi et al., 2018; Koodli et al., 2019). However, these models incorporate human strategies that might not be available for all designs and consider nested structures only. The other, more general approach seeks to learn RNA design purely from data. Eastman et al. (2018) propose to use reinforcement learning (RL) to adjust an initial input sequence by replacing nucleotides based on structural information. In contrast, Runge et al. (2019) and Riley et al. (2023) use a generative approach to the problem. Runge et al. (2019) employs a joint architecture and hyperparameter search approach (Bansal et al., 2022) via automated reinforcement learning (AutoRL) (Parker-Holder et al., 2022) to derive an RL system that is capable of generatively designing RNAs that fold into a desired target structure. Riley et al. (2023) uses a GAN (Goodfellow et al., 2020) approach specifically for the design of toehold switches (Green et al., 2014). However, all learning-based approaches so far consider RNA design for nested structures only, ignoring pseudoknots and base multiplets, while often being limited to the design of canonical base interactions.

Pseudoknotted Structures Pseudoknots are an important type of base pairs that influence the function of an RNA (Staple & Butcher, 2005). Therefore, some approaches tried to design RNAs from pseudoknotted structures (Taneda, 2012; Kleinkauf et al., 2015; Merleau & Smerlak, 2022). However, these algorithms work on a string notation in dot-bracket format (Hofacker et al., 1994), and thus, they cannot express base multiplets.

Overall none of the existing algorithms can design RNAs including non-canonical base pairs, pseudoknots, and base multiplets.

3 Methods

RNA secondary structures can be represented in different ways, including the common dot-bracket string notation (Hofacker et al., 1994) or adjacency matrices. We show different representations in Figure 1. One advantage of an adjacency matrix representation is that it can model all types of base interactions, especially if a nucleotide interacts with more than one other, a situation prevalent for most experimentally solved structures (Singh et al., 2019). In the following, we detail our generative approach to design RNAs from secondary structures using matrix representations.

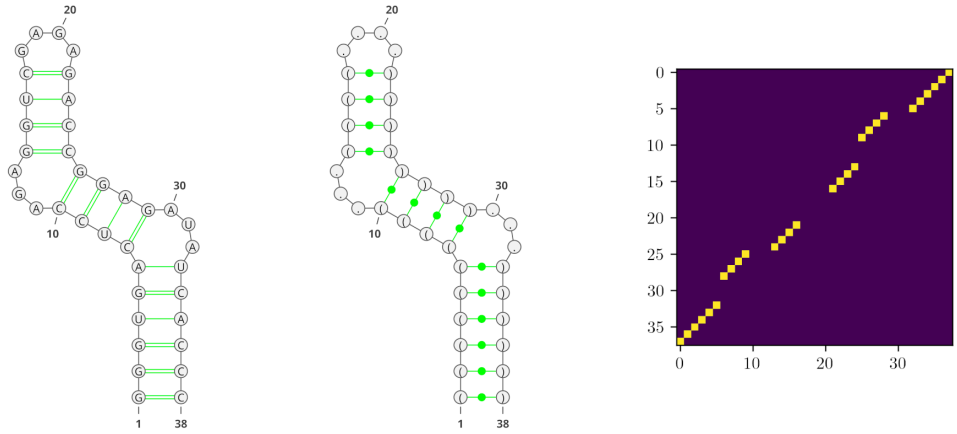

Figure 1: Representations of RNA secondary structures. (Left) Common graph representation of the RNA. (Middle) Dot-bracket notation in the graph structure. A pair of nucleotides is indicated by a pair of matching brackets, unpaired nucleotides are indicated by a dot. (Right) Matrix representation of the RNA. The matrix is a binary $L \times L$ square matrix, where L is the sequence length of the RNA. Pairing nucleotides are shown in yellow.

Loss The problem of RNA design is often addressed by defining a structural loss function $L_\omega = d(\omega, \mathcal{F}(\phi))$ that quantifies the difference between the target structure ω and the folding, $\mathcal{F}(\cdot)$, of the designed candidate sequence ϕ (Runge et al., 2019). However, a folding engine might not be differentiable, which makes it hard to employ this strategy to deep learning based design approaches. Therefore, we train a model to maximize the sequence recovery by minimizing the mean Cross Entropy Loss $\mathcal{L}_\psi$ over a nucleotide sequence. For a designed candidate $\phi \in \{A, C, G, U\}^l$ of length l and a given target nucleotide sequence $\psi \in \{A, C, G, U\}^l$ of the same length, this loss is defined as:

$$\mathcal{L}_\psi = \frac{1}{l} \sum_{i=1}^{l} L_{CE}(\psi_i, \phi_i) \qquad , \tag{1}$$

where $L_{CE}(\psi_i, \phi_i)$ is the cross entropy loss between the target sequence and the designed sequence at position i.

Model Our model is a vanilla auto-regressive encoder-decoder transformer model (Vaswani et al., 2017) with a next token prediction objective. The encoder embeds the structure information, while the decoder auto-regressively generates RNA nucleotide sequences by sampling from the softmax distribution. For *RNAinformer* we use axial attention in the first encoder block to process the matrix input similar to the RNAformer (Franke et al., 2024). However, instead of working on a 2D latent space, we use pooling to reduce the 2D representation to a 1D vector that is then passed through the encoder and the decoder to generate candidate sequences. Figure 3 and Figure 4 in Appendix A give an overview over our model.

Training Details Due to hardware limitations we set the maximum sequence length to 100 for all our experiments. We train our model with 6 encoder blocks and 6 decoder blocks with an embedding dimension of 256. The model is trained using cosine annealing learning rate schedule with warm-up and AdamW (Loshchilov & Hutter, 2019). The hyperparameters used for training our model are described in Table 2 in Appendix B.

4 EXPERIMENTS

We evaluate the RNAinformer in three settings with increasing complexity: We design RNAs for nested structures (Section 4.1), pseudoknotted structures (Section 4.2), and for experimentally val-

Table 1: Performance on the Rfam and bpRNA datasets for nested and pseudoknotted structures, respectively.

Model	Rfam	bpRNA	
	Solved	Solved	Solved PK
RNAinformer	**98.7%**	**31.7%**	9.7%
LEARNA (Runge et al., 2019)	64.8%	✗	✗
Meta-LEARNA (Runge et al., 2019)	64.3%	✗	✗
Meta-LEARNA-Adapt (Runge et al., 2019)	64.1%	✗	✗
aRNAque (Merleau & Smerlak, 2022)	✗	✗	9.7%

idated structures obtained from the Protein Data Bank (PDB) (Berman et al., 2000), including all kinds of base interactions (Section 4.3). For each of the experiments we train a separate model on the different datasets. We use two different folding algorithms: RNAfold (Lorenz et al., 2011) and RNAformer (Franke et al., 2024). While the former is the most widely used folding algorithm, the latter is the current state-of-the-art deep learning based approach, capable of predicting RNA structures with all kinds of base pairs. We use RNAfold for our experiments on nested structures and the RNAformer for all other experiments, since RNAfold can provide solutions for nested structures only. During evaluation, we generate between 20 and 100 candidate sequences for each task. The first sequence is generated using a greedy strategy and the rest of the sequences are generated using multinomial sampling. All datasets used for our experiments are detailed in Appendix C.

Metrics The ultimate goal of structure-based RNA design is to generate sequences that fold back into the target structure. Following the common convention in the field of RNA design, we report the number of solved tasks for a given benchmark dataset. However, we provide a more comprehensive analysis of all experiments with different performance measures in Appendix D.

4.1 RNA Design for Nested Structures

We compare the performance of RNAinformer against one of the currently best performing set of algorithms, LEARNA, Meta-LEARNA and Meta-LEARNA-Adapt (Runge et al., 2019). For each task, we generate 20 sequences with each algorithm and report the percentage of solved tasks. A task thus counts as solved if one of the 20 sequences folds into the desired target structure.

Data We use the Rfam dataset provided by Franke et al. (2024). While the set was originally built for learning a simplified biophysical model of RNA folding, it serves exactly our needs: Homologies between the training- and test set have been removed using RNA family annotations from the Rfam database (Griffiths-Jones et al., 2003), it contains a large amount of training data, and all sequences have been folded with RNAfold to obtain secondary structures. Hence, the dataset contains only canonical base pair interactions.

Results The results on the Rfam dataset are shown in Table 1 (left). We observe that RNAinformer clearly outperforms the other methods, solving 98.7% of the tasks. Notably, these are $\sim$35% more solved tasks compared to the next best competitor, LEARNA (64.8% solved tasks). Furthermore, RNAinformer generates multiple, highly diverse solutions for each task, indicated by a high diversity score of 0.713 as depicted in Table 6 in Appendix E.

4.2 RNA Design with Pseudoknots

In this section, we assess the performance of RNAinformer when designing RNAs for pseudoknotted input structures. We compare against aRNAque (Merleau & Smerlak, 2022) a recently proposed Lévy flight mutation based design algorithm that supports pseduoknotted structures. However, the evaluation of aRNAque is computationally expensive with rather high runtimes (an evaluation for 31 pseudoknotted structures nearly took 24 hours on two CPUs using 50 generations). We, therefore,

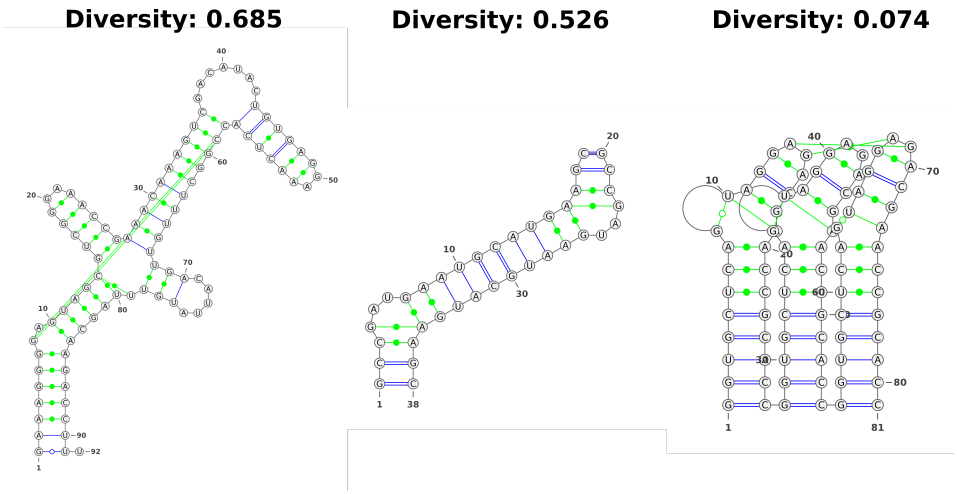

Figure 2: Example design predictions of solved structures including base multiplets and pseudoknot interactions.

evaluate aRNAque only on the pseudoknotted samples for our comparison. We again generate 20 candidate sequences per task for the evaluation of RNAinformer.

Data We use the bpRNA dataset provided by Franke et al. (2024) which uses VL0 and TS0 provided by Singh et al. (2019) for validation and testing, respectively. The datasets include non-canonical base interactions and pseudoknotted structures. As the dataset was originally created for structure prediction based on sequence similarity, it contains test structures that are also present in the training set. To ensure that there is no overlap between training- and test set, we remove pseudoknot-free test samples with identical structures in the training set, and pseudoknotted training samples with identical structures in the test set, since there are only a few pseudoknotted test structures available.

Results We report results in Table 1 (right). The RNAinformer solves nearly a third of the tasks (31.7% solved tasks) with over half (52.5%) of the designed sequences that solve a task having non-canonical base pairs (see Table 7). We solve the same number of pseudoknotted structures as aRNAque, but are able to predict more pseudoknot base pairs correctly on average as shown in Table 8 in Appendix E. Remarkably, all of the solutions generated by the RNAinformer for the pseudoknotted structures also contain non-canonical base pairs (Table 7).

4.3 RNA Design with all Kinds of Base Interactions

In this section, we investigate the ability of RNAinformer to design RNAs from structure data that contains all kinds of base pairs. To account for the difficulty of the task, we sample 100 sequences instead of only 20 sequences.

Data For our evaluations, we use the inter-family dataset provided by RnaBench (Runge et al., 2024). The dataset was prepared to ensure no data homologies between train- and test data based on sequence and structure similarity. The underlying test sets, TS1, TS2, TS3, and TS_hard, are derived from experimental structures of the Protein Data Bank (PDB) and were originally provided by Singh et al. (2021). All datasets contain structures with both pseudoknots and base multiplets.

Results We observe that RNAinformer cannot solve structures for the different test sets, indicating that designing sequences for structures with all kinds of base pairs seems to be much more challenging than for nested structures or structures with pseudoknots only. However, for all samples with base multiplets, we predict more than two of the multiplets present in the structures correctly on average, reported in Table 9 in Appendix E. Furthermore, Figure 2 shows examples of the training predictions that solve structures that contain base multiplets as well as pseudoknots. We conclude

that RNAinformer is generally capable of designing RNA sequences from structures that contain all kinds of base interactions. Nevertheless, we admit that further improvements in performance might require adjustments to our model like scaling in terms of model size or applying a finetuning strategy.

5 CONCLUSION

In this work, we propose RNAinformer, the first RNA design algorithm capable of designing RNA sequences for structures that contain all kinds of base interactions, including non-canonical base pairs, pseudoknots, and base multiplets. We demonstrate the strong performance of RNAinformer on tasks with nested structures only, tasks that contain pseudoknots, as well as on experimentally derived structures with all kinds of base interactions. We think that RNAinformer is a useful basis for future approaches to RNA design and expect it to be of great value for the RNA design community. For the future, we plan to further condition our model on different properties of RNA, to e.g. design RNAs with desired G and C nucleotide ratios.

6 ACKNOWLEDGEMENTS

We acknowledge funding by the European Union (via ERC Consolidator Grant DeepLearning 2.0, grant no. 101045765). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

The authors further acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG (bwForCluster NEMO) as well as funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under SFB 1597 (SmallData), grant number 499552394.

REFERENCES

Mirela Andronescu, Anthony P Fejes, Frank Hutter, Holger H Hoos, and Anne Condon. A new algorithm for rna secondary structure design. *Journal of molecular biology*, 336(3):607–624, 2004.

Archit Bansal, Danny Stoll, Maciej Janowski, Arber Zela, and Frank Hutter. Jahs-bench-201: A foundation for research on joint architecture and hyperparameter search. *Advances in Neural Information Processing Systems*, 35:38788–38802, 2022.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

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

Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

Peter Eastman, Jade Shi, Bharath Ramsundar, and Vijay S Pande. Solving the rna design problem with reinforcement learning. *PLoS computational biology*, 14(6):e1006176, 2018.

Ali Esmaili-Taheri and Mohammad Ganjtabesh. Erd: a fast and reliable tool for rna design including constraints. *BMC bioinformatics*, 16(1):20, 2015.

Ali Esmaili-Taheri, Mohammad Ganjtabesh, and Morteza Mohammad-Noori. Evolutionary solution for the rna design problem. *Bioinformatics*, 30(9):1250–1258, 2014.

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

Jörg K.H. Franke, Frederic Runge, Ryan Köksal, Rolf Backofen, and Frank Hutter. Rnaformer: A simple yet effective deep learning model for rna secondary structure prediction. *bioRxiv*, 2024. doi: 10.1101/2024.02.12.579881.

Laiyi Fu, Yingxin Cao, Jie Wu, Qinke Peng, Qing Nie, and Xiaohui Xie. Ufold: fast and accurate rna secondary structure prediction with deep learning. *Nucleic acids research*, 50(3):e14–e14, 2022.

Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. Rnaifold: a constraint programming algorithm for rna inverse folding and molecular design. *Journal of Bioinformatics and Computational Biology*, 11(02):1350001, 2013.

Juan Antonio Garcia-Martin, Ivan Dotu, and Peter Clote. Rnaifold 2.0: a web server and software to design custom and rfam-based rna molecules. *Nucleic Acids Research*, 43(W1):W513–W521, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Alexander A Green, Pamela A Silver, James J Collins, and Peng Yin. Toehold switches: de-novo-designed regulators of gene expression. *Cell*, 159(4):925–939, 2014.

Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, and Sean R. Eddy. Rfam: an rna family database. *Nucleic acids research*, 31 1:439–41, 2003.

Stefan Hammer, Christian Günzel, Mario Mörl, and Sven Findeiß. Evolving methods for rational de novo design of functional rna molecules. *Methods*, 161:54 – 63, 2019. ISSN 1046-2023. Development and engineering of artificial RNAs.

Ivo Hofacker, Walter Fontana, Peter Stadler, Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte fuer Chemie/Chemical Monthly*, 125:167–188, 02 1994.

Robert Kleinkauf, Torsten Houwaart, Rolf Backofen, and Martin Mann. antaRNA–Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. *BMC bioinformatics*, 16(1): 389, 2015.

Rohan V Koodli, Benjamin Keep, Katherine R Coppess, Fernando Portela, Eterna participants, and Rhiju Das. Eternabrain: Automated rna design through move sets and strategies from an internet-scale rna videogame. *PLoS computational biology*, 15(6):e1007059, 2019.

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, Nov 2011. ISSN 1748-7188.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Nono S. C. Merleau and Matteo Smerlak. arnaque: an evolutionary algorithm for inverse pseudoknotted rna folding inspired by lévy flights. *BMC Bioinformatics*, 23, 2022.

Gerard Minuesa, Cristina Alsina, Juan Antonio Garcia-Martin, Juan Carlos Oliveros, and Ivan Dotu. Moirnaifold: a novel tool for complex in silico rna design. *Nucleic acids research*, 49(9):4934–4943, 2021.

Kevin V Morris and John S Mattick. The rise of regulatory rna. *Nature Reviews Genetics*, 15(6): 423–437, 2014.

Jack Parker-Holder, Raghu Rajan, Xingyou Song, André Biedenkapp, Yingjie Miao, Theresa Eimer, Baohe Zhang, Vu Nguyen, Roberto Calandra, Aleksandra Faust, Frank Hutter, and Marius Lindauer. Automated reinforcement learning (autorl): A survey and open problems. *Journal of Artificial Intelligence Research (JAIR)*, 74:517–568, 2022.

Francis E Reyes, Andrew D Garst, and Robert T Batey. Strategies in rna crystallography. *Methods in enzymology*, 469:119–139, 2009.

Aidan T. Riley, James M. Robson, and Alexander A. Green. Generative and predictive neural networks for the design of functional rna molecules. *bioRxiv*, 2023.

Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In *International Conference on Learning Representations*, 2019.

Frederic Runge, Karim Farid, Jorg K.H. Franke, and Frank Hutter. Rnabench: A comprehensive library for in silico rna modelling. *bioRxiv*, 2024.

Jade Shi, Rhiju Das, and Vijay S Pande. Sentrna: Improving computational rna design by incorporating a prior of human design strategies. *arXiv preprint arXiv:1803.03146*, 2018.

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

Jaswinder Singh, Kuldip Paliwal, Tongchuan Zhang, Jaspreet Singh, Thomas Litfin, and Yaoqi Zhou. Improved rna secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, 37, 2021.

David W Staple and Samuel E Butcher. Pseudoknots: Rna structures with diverse functions. *PLoS biology*, 3(6):e213, 2005.

Akito Taneda. Multi-objective genetic algorithm for pseudoknotted rna sequence design. *Frontiers in Genetics*, 3:36, 2012.

Ignacio Tinoco Jr and Carlos Bustamante. How rna folds. *Journal of molecular biology*, 293(2): 271–281, 1999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Quentin Vicens and Jeffrey S Kieft. Thoughts on how to think (and talk) about RNA structure. *Proceedings of the National Academy of Sciences*, 119(17):e2112677119, 2022.

A MODEL DETAILS

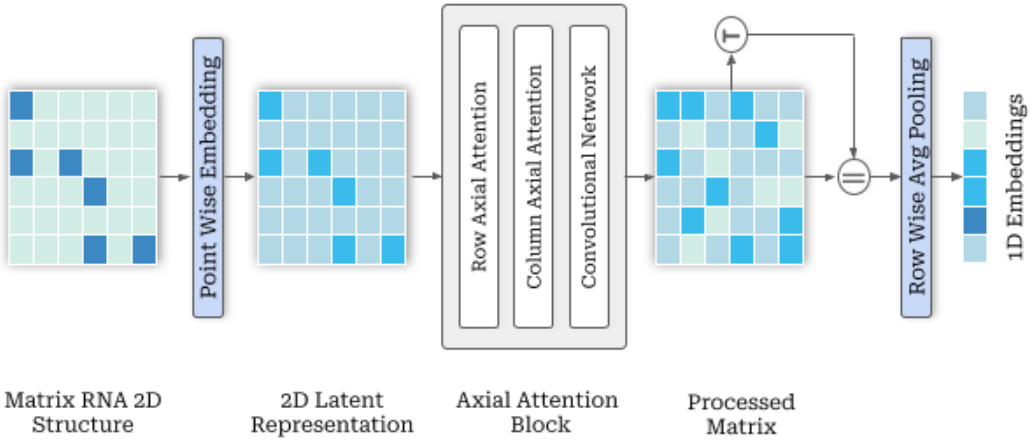

Figure 3: Overview of matrix input processing in RNAinformer.

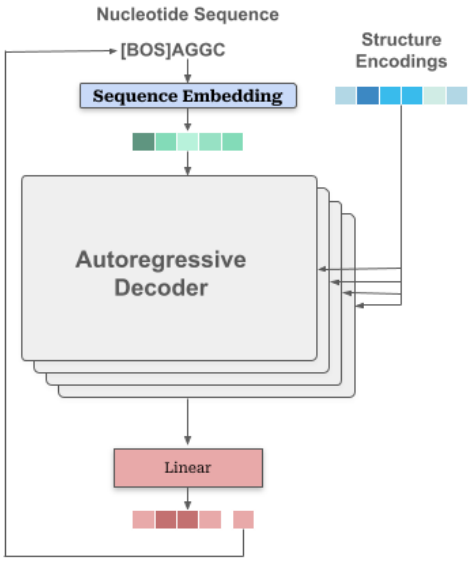

Figure 4: Overview of nucleotide sequence generation.

B TRAINING DETAILS

Table 2: Hyperparmeters for RNAinformer training.

Group	Parameter	Value
Optimizer	lr	0.001/0.0003
	weight decay	0.1
	betas	0.9,0.98
	Warmup steps	1000
	LR schedule	cosine annealing
	LR decay factor	0.1
Model	Model dim	256
	Layers	6
	Num head	4
	FeedForward factor	4
	FeedForward kernel	3
	Dropout	0.1
Training	Batch size	128/64
	Max steps	50k/100k

C DATASETS

Table 3: Overview of the Rfam dataset.

Set	#Samples	Avg Length	Pseudoknots	Multiplets
Rfam-Train	276242	75	0(0.00%)	0(0.00%)
Rfam-Valid	2291	73	0(0.00%)	0(0.00%)
Rfam-Test	2979	71	0(0.00%)	0(0.00%)

Table 4: Overview of the bpRNA dataset.

Set	#Samples	Avg Length	Pseudoknots	Non-Canonical BP	Multiplets
TR0 (Train)	25309	77	907(3.58%)	12802(50.55%)	0(0.00%)
VL0 (Valid)	603	78	30(4.98%)	373(61.86%)	0(0.00%)
TS0 (Test)	462	77	31(6.7%)	282(61.0%)	0(0.00%)

Table 5: Overview of the Inter-family Dataset.

Set	#Samples	Avg Length	Pseudoknots	Non-Canonical BP	Multiplets
Train	19540	73	2047(10.47%)	11114(56.70%)	1330(6.80%)
Valid	494	77	12(2.43%)	287(57.86%)	13(2.63%)
TS1	54	61	43(79.62%)	49(90.74%)	40(74.07%)
TS2	36	45	23(63.88%)	35(97.22%)	26(72.22%)
TS3	16	67	15(93.75%)	15(93.75%)	15(93.75%)
TS-Hard	25	55	17(68.00%)	21(84.0%)	18(72.00%)

D METRICS

Valid Sequences We refer any candidate sequence that solves a task as a valid sequence. We measure the efficiency of the generative process by the number of valid sequences that are produced for each task.

$$ValidSequences = \frac{\#ValidSequences}{\#CandidateSequences} \tag{2}$$

Diversity To measure the diversity of the valid sequences generated for a target structure, we use the pairwise Hamming distance. For N valid sequences of length l the diversity is defined as,

$$Diversity = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{N}\frac{1}{l}\sum_{k=1}^{l}H(S_{ik}, S_{jk}) \qquad , \tag{3}$$

where $H(S_{ik}, S_{jk})$ describes the positional Hamming distance:

$$H(S_{ik}, S_{jk}) = \begin{cases} 0 & \text{if } S_{ik} = S_{jk} \\ 1 & \text{else} \end{cases} \qquad . \tag{4}$$

NC To measure the models ability to design with non-canonical base pair interactions we report the number of valid sequences containing non-canonical base pairs.

F1 Score The F1 Score is a commonly used performance measure to assess the quality of secondary structure prediction algorithms. It is based on the confusion matrix, which describes the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) when comparing a predicted structure to the ground truth. The F1 score is the harmonic mean of precision and sensitivity, defined as:

$$F1 = \frac{2 \cdot TP}{(2 \cdot TP + FP + FN)} \tag{5}$$

Matthews Correlation Coefficient Compared to the F1 score that emphasizes on positives, the MCC is a more balanced measure (Chicco & Jurman, 2020). The MCC can be calculated as follows.

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{6}$$

For each task in a test set we take the maximum F1 and MCC scores achieved by a candidate sequence, and report the average over these values across three random seeds.

E ADDITIONAL RESULTS

Table 6: Results for the design of RNAs for nested structures of the Rfam dataset.

Model	Test Set	Solved	Valid Sequences	Diversity	F1	MCC
RNAinformer	Rfam	**0.987±0.007**	**0.409±0.008**	**0.713±0.001**	**0.999±0.001**	**0.999±0.001**
LEARNA	Rfam	0.648	0.222	0.547	0.965	0.966
Meta-LEARNA	Rfam	0.643	0.223	0.547	0.965	0.966
Meta-LEARNA-Adapt	Rfam	0.641	0.222	0.542	0.963	0.964

Table 7: Results for the design of RNAs including pseudoknots using the bpRNA dataset.

Model	Test Set	Solved	Valid Sequences	Diversity	F1	MCC	NC
RNAinformer	BpRNA	0.317±0.013	0.512±0.012	0.347±0.001	0.589±0.008	0.601±0.008	0.525±0.023
	pK	**0.097±0.032**	0.136±0.042	0.080±0.027	0.467±0.030	0.481±0.030	1.0±0.0
aRNAque	pK	**0.097**	**1.000**	**0.222**	**0.824**	**0.831**	-

Table 8: Comparison between RNAinformer and aRNAque on pseudoknotted structures from the bpRNA dataset.

Model	**Test Set**	**Pseudoknot hits**	**Total Pseudoknots**
RNAinformer	pK	**4.56±0.46**	9.81
aRNAque	pK	3.35	9.81

Table 9: Results for RNA design for experimentally validated structures with all kinds of base interactions.

Test Set	**F1**	**MCC**	**Multiplet Hits**	**Total Multiplets**
TS1	0.388±0.018	0.426±0.018	2.35	13.00
TS2	0.498 ± 0.006	0.524±0.009	2.31	9.92
TS3	0.297 ± 0.015	0.333±0.019	2.58	13.87
TS-Hard	0.363± 0.025	0.391±0.026	2.19	12.67

F FOLDING ALGORITHMS

Table 10: Tasks and folding algorithms.

Task	**Folding Algorithm**
Biophysical Model Inversion	RNAfold
Pseudoknot Design	RNAformer
Multiplet Design	RNAformer