Improving secondary structure predictions with secondary structure "fixing" task

Juneki Hong
School of Electrical Engineering and Computer Science, Oregon State University
Current affiliation: TikTok, San Jose, CA
junekihong@gmail.com

Dezhong Deng
School of Electrical Engineering and Computer Science, Oregon State University
Current affiliation: Two Sigma, New York, NY
dengdezhong816@gmail.com

David A Hendrix
School of Electrical Engineering and Computer Science, Oregon State University
Department of Biochemistry and Biophysics, Oregon State University
david.hendrix@oregonstate.edu

Abstract

The structure of RNA can determine the function, and improvements in RNA secondary structure prediction can help in understanding the functions of RNA. Nucleotides in RNA sequences form base-pairing interactions in context-specific preferential behavior to help determine the secondary structure. Structure prediction algorithms have been developed to predict the secondary structure, including dynamic programming, and machine learning approaches. One of the central challenges in the prediction of secondary structure with deep learning is that these architectures are not good at bracketed structure prediction. To overcome this challenge, we present a deep learning approach for predicting secondary structure that uses an input predicted structure to provide a scaffolding for the structure prediction. We find that architectures using LSTM and self-attention-based transformer layers predict a strong baseline in the prediction of base pairs ($F1 = 53.73$), but significantly improves ($F1 = 59.52$) when predictions from dynamic programming methods are provided as input. Model interpretation shows that patterns of attention for different layers of the network are enriched for specific paired regions or regions that should be paired. Analysis of neural network models like this can shed light on possible missed interactions, and what other positions contribute most to output fixed positions.

Introduction

Ribonucleic acids (RNA) are a ubiquitous and essential molecule for life, and the structures of RNA molecules are widely understood to inform their function. Much research is devoted to the fascinating aspect of their structure preserving properties, in that RNA sequences can vary widely in a myriad of different examples but can still all fold into similar structures. The task of predicting the secondary structure of RNA sequences were traditionally approached with thermodynamics-based dynamic programming algorithms[8] and machine learning-derived parameters[4, 10]. However,

Preprint. Under review.

recent advances in the application of deep learning have demonstrated progress in RNA structure prediction performances in accuracy while still maintaining tractability[5, 18, 16].

There has been interest in utilizing recent advanced deep learning models, such as using the attention-based transformer model [19], which was original designed for translation between languages. This network has been applied to a wide variety of tasks within biological domains including protein structure [1], drug-target interaction [9], DNA enhancer prediction [12], and DNA sequence classification [7]. Transformers utilize self-attention, which describe relationships between elements of the input, and can be integrated to compute the elements of an output. This combination of information is not necessarily uniform in distribution, and is instead weighted differently for each part of the input sequence depending its importance deemed by the neural network. Thus the term "self-attention" comes from representing a element from the input sequence in relation to all elements including itself and the neural network's model weights.

While thermodynamic models of secondary structure prediction have readily interpretable parameters, deep learning models generally require additional tools for interpretation [17, 14, 21]. Attention weights when applied to RNA secondary structure suggest an attractive means of network interpretation because one can visualize the relevant parts of the input nucleotide sequence that inform the network's decision on forming a base pair.

However, a notable theoretical limitation of self-attention based neural networks is the inability to pair an arbitrary number of brackets[6], an important property of the RNA secondary structure prediction task. Indeed, common representations of RNA secondary structures include "dotbracket" sequences, which represent base pairs as a pair of matched parentheses. To partially alleviate a part of what transformers may find difficult about RNA folding, we use the output of the dynamic programming methods as part of the input, which already have matching base pair brackets. Allowing this additional input to form a "scaffolding", we aid the network in providing an already well-formed base pair bracketing such that during training its learning efforts would focus on other aspects in the secondary structure prediction (RNA folding) task. We call this task *base pair fixing*, where a previous model's predictions (e.g. using the output of Linearfold or RNAfold) are input in addition to the nucleotide sequence and the output dotbracket positions that differ from the input are considered *fixes* to the input. We call our approach "bpRNA-fix" because it fixes base pairs in RNA, and was trained on data from our meta-database, bpRNA-1m.

Base pair fixing offers an additional way towards neural network explanation and visualization, compared to sequence-only prediction approaches. Self-attention weights have an inherent directionality to them, and one can specifically focus on and visualize the attention to positions that are corrected in the output, and even categorized across different types of fixes to be analyzed. We explore whether our model's attention weights are directly interpretable in this way. An advantage of this approach is that the usage of another structure prediction program indirectly makes use of their previously designed thermodynamic models, standing on their proverbial shoulders in order to simplify our task and expedite training of machine learning models.

Methods

Transformer Networks

Transformer networks are a deep learning architecture that utilize self-attention to model sequential data [19, 3] or otherwise structured data, such as images or graphs [15, 20]. In biological sequence data such as RNA, each nucleotide is represented by a multi-dimensional vector, and to apply a "layer" of a transformer network would be to update each nucleotide vector representation using a series of matrix multiplications involving all nucleotides in the sequence. The resulting output amounts to updating every x_i to an enriched version x_i', which incorporates information from every position. These attention units are typically stacked in multiple "layers", and the final sequence representation is utilized for a downstream task, for example such as site-wide classifications of whether each nucleotide is paired or unpaired, or sequence-wide predictions such as for an existence of a pseudoknot. Transformers additionally have *encoder* and *decoder* components, that both utilize the self-attention mechanism detailed above.

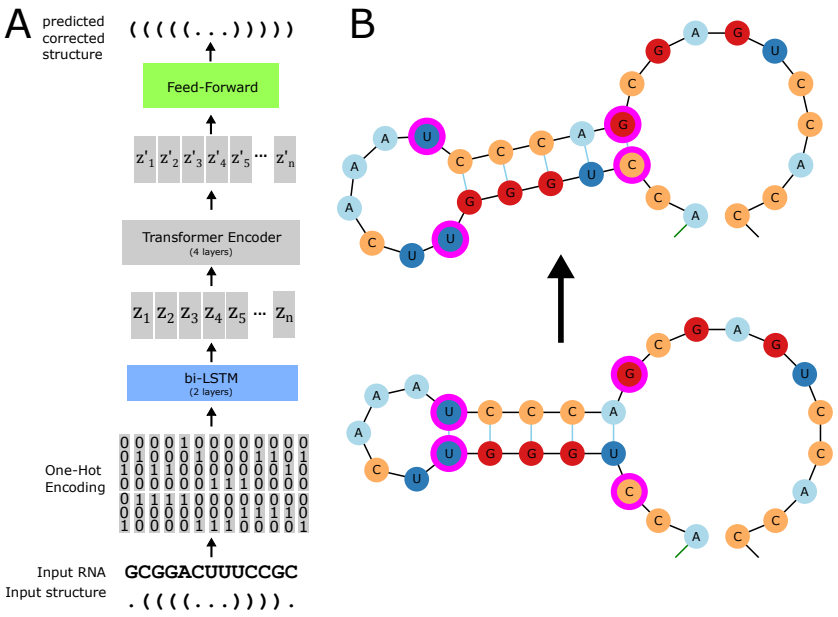

Figure 1: A: Architecture Diagram for the Basepair Fixing task. B: A simple example of an input RNA sequence (bottom) becoming "fixed" by our model (top), with the fixed positions highlighted in magenta. In this case the added base-pairs shift the pairings of the entire stem.

Architecture and Training

Reading in the input nucleotide and secondary structure sequences as a one-hot encoding, the input is richly encoded with the use of a two-layer bidirectional LSTM (bi-LSTM), followed by four layers of transformer self-attention encoders (Fig 1A). We use these bi-LSTM layers instead of the positional encoding that is commonly used for attention networks.

The final output is then predicted with a simple feed forward network, such that each position of the sequence independently has three possible outputs (left-bracket/pair, right-bracket/pair, dot/unpaired). This decoding scheme is unconstrained and it is entirely possible to output nonsensical, mismatching, or unbalanced base pairs. From this simple decoding, we still demonstrate strong accuracy performance seen in Figure 2A, and so we instead focus more on discussing the model's explainability, where each position of the output can be tied directly to the attention weights of the model uncomplicated from any additional post-processing, lending itself to be subsequently visualized. Notably, bpRNA-fix shows fewer unpaired brackets than AttentionFold, and predictions from bpRNA-fix show greater parity, such that brackets always come in pairs (Figure 2B-D).

Our deep learning model was coded using the Pytorch library, with a cross-entropy training loss and the Adam optimizer [11]. The Adam hyperparameter used were all default values from the Pytorch library other than the learning rate (0.001), betas values (0.9, 0.98), and epsilon value (1e-09). Since the network can predict every position of a sequence as well as every sequence in a minibatch in parallel, the training loss can all be computed with vector notation within a GPU for efficiency.

Data set and Preparation

Starting with the RNA sequences and associated secondary structures in the bpRNA-1m dataset[2], sequences with lengths between 20 and 200 were kept. The remaining sequences were further filtered for diversity using CD-HIT-EST [13] to have less than 90% sequence similarity. This collection of low-similarity data was split randomly using an 80-10-10 training, validation, and testing. This consideration addresses over-fitting, for the potential issue of training examples to be very similar to those found in the validation and test, and separates RNA sequence types and families within the data inasmuch as they are less than 90% similar. To add more training data, sequences that were

part of the CD-HIT-EST clusters with training examples were also added back into the training data, thereby there might be similar sequences in the training, but not to the test set. Using the EMBOSS Needleman-Wunsch global alignment program (`needle`), we further required that the selected sequences have less than 90% similarity with the validation and testing sequences. The final process resulted in a split consisting of 42,859, 2,344, and 2,345 training, validation, and testing sequence examples respectively. The data set includes both canonical and non-canonical (non-Watson-Crick) base pair interactions, so the possibility exists for the model to learn to predict non-canonical interactions from an canonical base pair scaffolding.

0.1 Attention Weight Information Content

For a sequence of length n, the self-attention can be described by an $n \times n$ weight matrix W such that for $i, j \in [0, n]$ the weight W_{ij} indicates the weight of the attention from indices j to i. The attention weights to a given index are softmax-normalized from the transformer network, thus $\sum_j w_{ij} = 1$. To quantify how focused the attention was, we computed information content relative to a baseline entropy H_0. We define information content of a given index i as the reduction in Shannon entropy, $-\Delta H$, when going from a uniform distribution to the distribution defined by the self-attention to that position.

$$IC_i = -\Delta H_i = H_u - H_i$$

The expected entropy for a uniform baseline at $w_{ij} = 1/n$ is given by $H_0 = -\sum_{j=0}^{n} \left(\frac{1}{n} \cdot \log_2 \frac{1}{n}\right) = \log_2 n$, and the observed entropy at position i given by $\hat{H}_i = -\sum_{j=0}^{n} w_{ij} \cdot \log_2(w_{ij})$. We therefore have $IC_i = \log_2 n + \sum_{j=0}^{n} w_{ij} \cdot \log_2(w_{ij})$. The information content for attention weights describes attention going to each index i, and we also can calculate the information content for attention weights coming from a given index. To compute the outgoing information content, we must normalize the outgoing weights with a normalizing constant like $Z_j = \sum_{i=0}^{n} w_{ij}$, the sum of the attention weights to the position j, which can be found plotted in Figure 3A. Our observed outgoing entropy becomes $\hat{H}_j = -\sum_{i=0}^{n} \frac{w_{ij}}{Z_j} \cdot \log_2(\frac{w_{ij}}{Z_j})$, and thus $IC_j = \log_2 n + \sum_{i=0}^{n} \frac{w_{ij}}{Z_j} \cdot \log_2(\frac{w_{ij}}{Z_j})$.

Results

Positional Encodings vs LSTM

A notable challenge in transformer network architectures is that attentions are computed without natural notions of position. Thus all x'_i output embeddings incorporate information from inputs x_j from all positions j without even considering whether $i < j$ for example. This is typically addressed with the addition of a "positional encoding", usually with trigonometric functions of different periods, to encode every position index. The positional encoding is then combined with the input embedding, allowing the model to differentiate between nucleotides occurring at different positions. Instead of using a positional encoding, we used another common neural network architecture in the Bidirectional Long-Short Term Memory (bi-LSTM), designed to model sequential information, because we found that a bi-LSTM encoding leads to an overall improvement in final evaluation performance compared to positional encodings, shown in Figure 2A. Using an LSTM as the first encoding layer of our network, we forgo the usage of positional encodings, and instead allow the network to model relative sequential information as a part of its learning.

F1-Score Evaluation of Secondary Structure

In order to compare the performances of RNAfold and Linearfold to our bpRNA-fix methods, the first-pass approach for evaluation would be to calculate the site-wise confusion matrix and accuracy. Indeed, Table 1 shows a robust overall accuracy greater than 91%. However, due to the unconstrained nature of our dotbracket predictions, it is still possible to do well on this metric while still having inconsistent on nonsensical base pairings, as seen in site accuracy of Table 2.

The more standard approach is to more directly take RNA secondary structure base pairings into account with an evaluation metric based on the precision and recall of the (i, j) base pairs in the labeled structure. This evaluation is much more strict in that a single missed or added base pair can

Symbol	(	)	.	Label
(	46521	461	5553	52535
)	385	46616	5534	52535
.	4371	4304	125706	134381
Predicted	51277	51381	136793	239451
Precision	90.7%	90.7%	91.9%	91.4%

Table 1: Confusion matrix summarizing the performative accuracy of our Linearfold+bpRNA-fix model on our held-out test set, with our overall site-wise accuracy shown in the bottom right-hand corner.

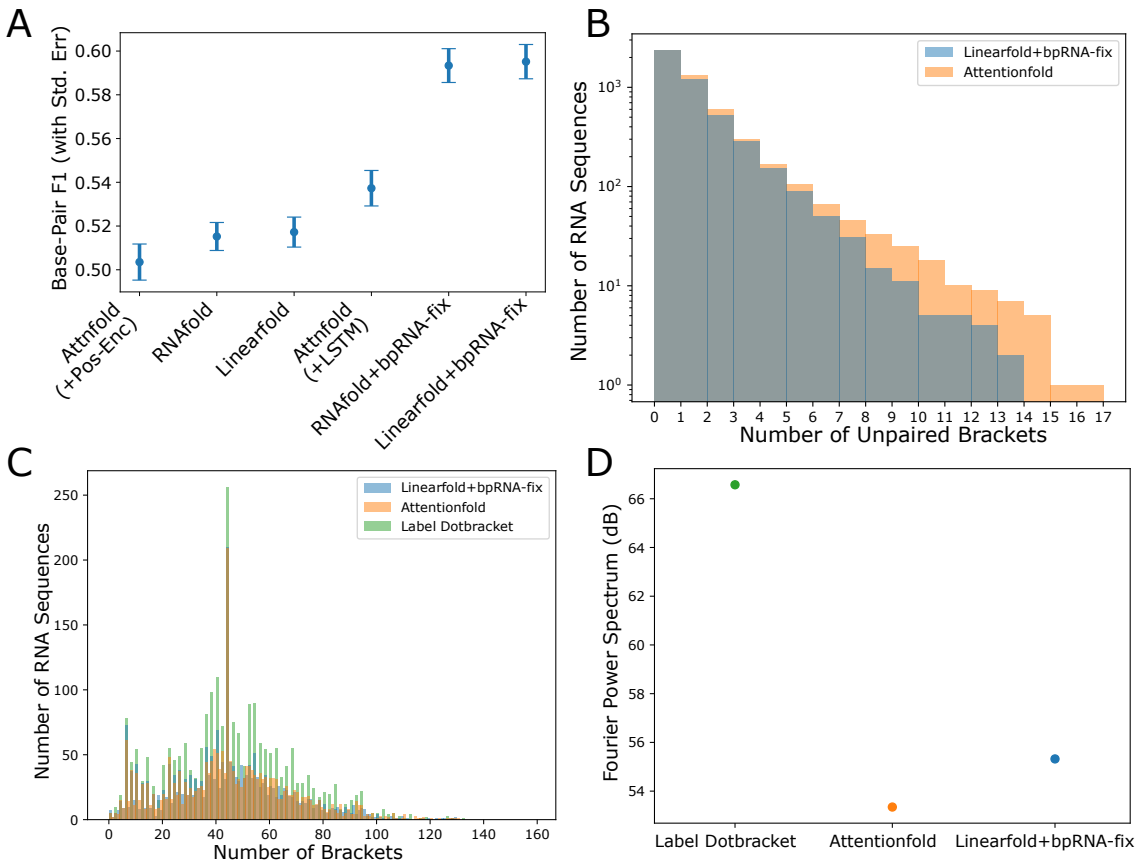

Figure 2: **A.** Sequence-level F1 score performances with standard error bars for each of our models evaluated on the test set. **B** Reverse-CDF of the number of unpaired brackets produced by bpRNA-fix vs Attentionfold on the test set. Further quantifying the performance difference between the two models, Attentionfold tends to produce more unpaired brackets in its output than bpRNA-fix. **C.** Histogram binning the number of brackets predicted by our models compared to the labeled dataset. bpRNA-fix tends to predict more even and less odd numbers of basepairs compared to Attentionfold. **D.** Treating the Figure 2C histogram as time-series data, we apply a fast-Fourier transform to measure the parity of the model output predictions. The resulting peaks at frequency 0.5 (corresponding to an even parity of base pair predictions) are compared in this plot, showing that bpRNA-fix predicts base pairs with higher even parity than Attentionfold.

Model	Site Accuracy	Test F1 Score (Seq Avg)
RNAfold	68.11	51.52
Linearfold	72.61	51.72
Attentionfold (+Pos Enc)	90.55	50.35
Attentionfold (+LSTM)	91.74	53.73
bpRNA-fix (+RNAfold)	91.26	59.34
bpRNA-fix (+Linearfold)	91.39	**59.52**

Table 2: Final F1 Score for our models, averaged across our test dataset sequences. Figure 2A plots these with standard error bars.

mismatch an entire set of index pairings, and thus potentially ruin the subsequent evaluated F1-score. For our unconstrained model, we predict each position independently potentially making invalid or unbalanced bracket outputs possible, and making this pair-centric F1 very stringent. For each test sequence we calculate the resulting F1-score, computed as the harmonic mean of the precision and recall, and we show the average and standard error of this evaluation in Figure 2A.

To compute the base pairings, we used a simple program that reads a dotbracket from left to right using a stack. Left brackets are pushed on to the stack and right brackets pop from the stack, forming a series of (i, j) pairs as a result. In the case of invalid or mismatched brackets, right brackets that try to pop from an empty stack are ignored as well as remaining left brackets on the stack after the entire dotbracket has been processed. In this way, we can extract the corresponding series of base pairs from the predicted output even if the prediction is unbalanced, and use this to compare with the labeled structure.

We compare bpRNA-fix to a version of our model relying only on the underlying neural network model without additional input predictions, that we call “AttentionFold", as well as to the input programs RNAfold and Linearfold. We observe a surprisingly strong performance in the trained Attentionfold baseline model with LSTM+transformer network layers, and it outperforms the thermodynamic models. However, there is an even greater improvement for bpRNA-fix, which incorporates the thermodynamic model outputs (Table 2).

Interpretation of Attention Weights

In addition to better folding performances, the task of secondary structure fixing has an added benefit of having more focused model interpretation via inspection of attention weights at positions where the thermodynamic inputs differ from the predicted outputs (fixed positions). The analysis of attention weights are an opportunity to interpret the computation that results in the fixed base pairs, and what other positions contribute the most to that output.

Attention weights in neural networks are typically normalized, allowing for a distributional interpretation across the input sequence. Thus, attention weights from all positions towards a particular position sum to one. A first approach towards attention weight interpretation simply reversing this summation, summing the attention weights emanating *from* each position, $Z_j = \sum_i W_{ij}$. An example of this can be seen in Figure 3A where we see the summed attention weights become more organized around paired regions in subsequent layers of the network. In this exapmle, the summed attention is greatest over paired positions, suggesting that these positions have the greatest contribution to the attention going to other positions in the sequence.

To focus on the strongest attention, we assessed statistical significance via a z-score can be computed from the distribution of attentions from all other positions. Z-scores are computed by taking the mean μ and standard deviation σ from all the attention weights w_{ij} across all positions i. Looking at the attentions to a fixed position, we visualize arcs for attention weights with a significant z-score. Figure 3B, shows arcs to fixed positions that have z-score greater than one. The arc plot shows that significant attentions are coming from structured regions in this example, and that the significant attentions behave differently from layer to layer. The higher layer, layer 4, shows more of the significant attention is coming from paired positions.

The fixed positions also show an even parity, suggesting when they are added, then tend to be in pairs. The histogram in Figure 3C shows that many sequences tend to have an even number of fixes,

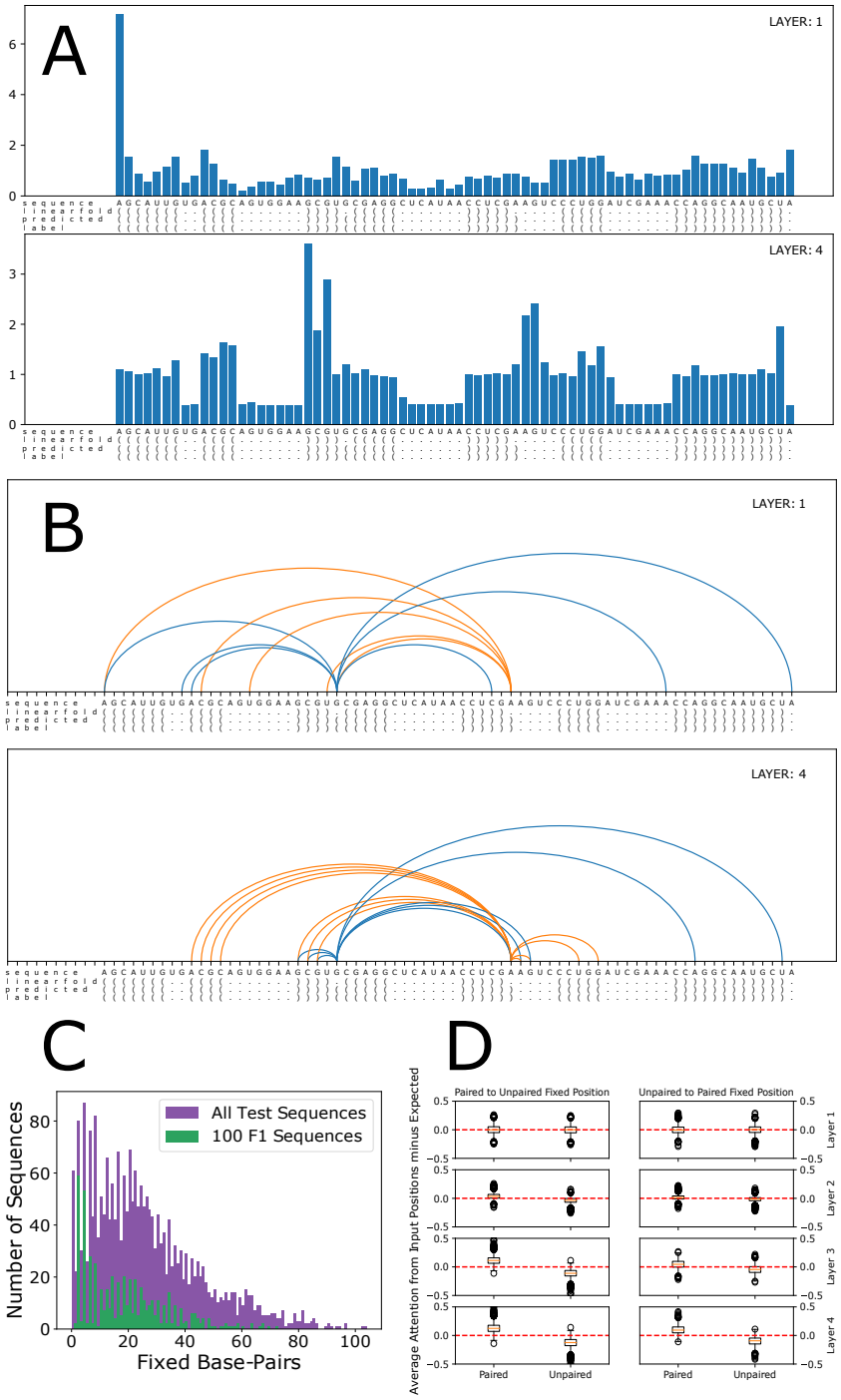

Figure 3: **A.** Summed attention weights for bpRNA-fix for an example structure (bpRNA_CRW_43036). The summed attention weights outgoing from each position are computed, with the first and last layers shown. While the initial attention weights in layer one look to be more uniform in nature, the weights become more organized by the fourth layer, concentrating around regions with structure. **B.** Arc plot of the same structure as panel A, showing the statistically significant attention values to "fixed" positions, of the first and last self-attention layers of the model. **C.** Number of sequences of the test data set binned by the number of fixed base-pairs for all sequences and for 100 F1 sequences. **D.** Boxplots show attention weights to fixed positions across the test set, subdivided by whether the output fixed position is a paired or unpaired fix (left or right column) and whether the attentions are coming from paired or unpaired input positions (left or right boxplot). Only sequences that our model get correct (100 F1) are considered.

even though the model is not constrained to do so. For comparison, we also show the subset of test sequences where our model perfectly predicted the dotbracket structure.

Focusing on fixed positions in the output, we show that attention weights are stronger coming from paired regions of the input compared to a uniform baseline, as seen in Figure 3D. The summed attention weights emanating from each position as well as strong individual attentions going towards a fixed position are visualized, we see overall stronger attention weights coming from input paired positions. We see that both paired and unpaired fixes, attention weights from paired positions across the input are higher than a uniform expected baseline, especially in subsequent layers of the network.

In order to further quantify which positions attract strong attention weights from the network, we examined non-uniformity in attention weights. We describe the non-uniformity of the distribution of attention using information content compared to a uniform distribution of weights, and measured the information content for different types of sequence positions across each layer of our transformer model, summarized in Figure 4A. From this heatmap of fix cases we see the strongest information content associated with fixed positions that flip the direction of a base pair in the lower layers. We see in higher layers that there is a slight enrichment in fixes that go from paired to unpaired as well, but with overall lower non-uniformity. Overall, non-uniformity such as more focused attention associated with removing the left brackets compared to right brackets suggests some asymmetry in the focus of the attention weights for particular types of fixes. Figure 4B highlights an example with lower incoming attention information content for the fixed right brackets for lower layers, but higher layers show much greater information for the added base pairs upon fixing as well as for other paired positions. Similarly, Figure 4C shows a much greater outgoing information content for removed base pairs, consistent with the trends observed in Figure 4 A.

bpRNA Structure Labels

bpRNA is a tool for automated annotation of RNA secondary structures into different loop types (e.g. H=hairpin, M=multiloop etc) and stems, providing more context than the dotbracket notation [2]. To further extend our own information content-based analysis, we incorporate the use of these structure types, labeling both the Linearfold input and bpRNA-fix output predictions.

Computing the incoming information content at each position across the test data and binning them according to Linearfold input and bpRNA-fix output structures, gives us the heatmap at Figure 5A. Here we see what types of fixes cause the attention weights of our model to be the most non-uniform. Some of the most significant averaged attention are associated with E$\rightarrow$H fixes (from an end to a hairpin loop). We visualize the structure of a Linearfold input in Figure 5B and the corresponding model output in Figure 5C. Arguably, modifying an "end" position to a hairpin loop like this would be among the most extreme changes that could happen, so it is an interesting correlation with the greatest information content. In addition, we color-coded the fixed position in magenta, with other positions heatmap-weighted according to attention to the fixed position with red corresponding to the highest attention. The attention weights to the fixed position in the last layer 4 of our model were found to be enriched around the model output base-paired regions, the two stems seen in Figure 5 C.

Discussion

We have shown that the introduction of a well-formed secondary structure prediction from one program helps improve the performance of our a transformer self-attention based network, adding parity information that serves as a "scaffolding" for the transformer predictions. This results in bpRNA-fix making predictions that are more balanced than a baseline transformer model without this additional input (Attentionfold), and better overall performance than Linearfold and RNAfold, which provided our secondary structure inputs. The RNA fixing task allows us to focus our analysis and visualization efforts on fixed positions, where our model output differs from the input dotbracket, rather than making sense of the structure in its entirety. Having a small number of fixed positions, and examining weights to that fixed position, makes interpretation more tractable.

Analysis of attention weights from bpRNA-fix using information content shows differentiation of roles between subsequent layers of the network, going from being more uniform to organizing around paired regions. We further show that stronger attentions at fixed positions are paid towards paired output regions. In this way, the distribution of attention weights shed light on greater focus on other

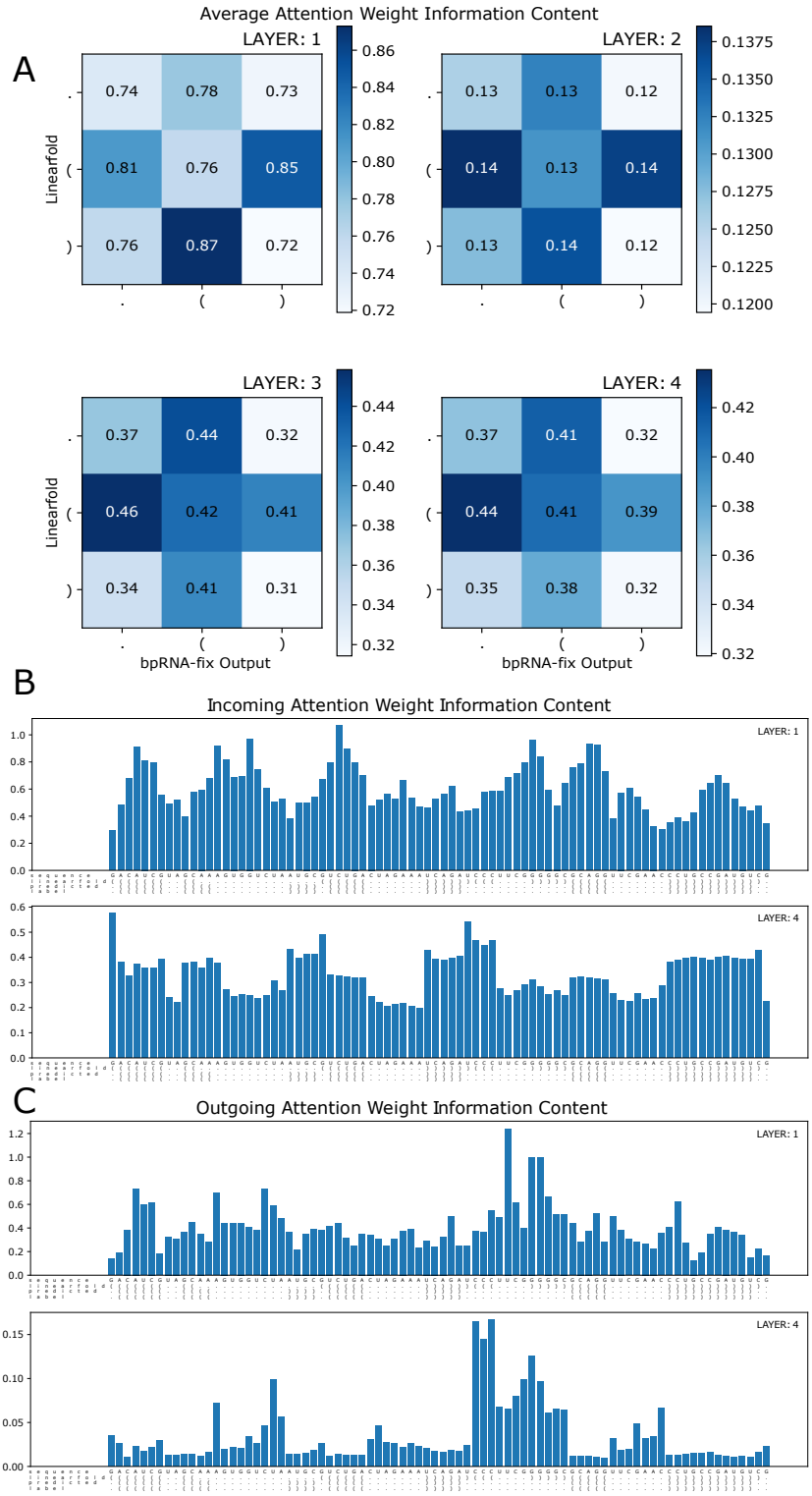

Figure 4: **A.** Average incoming attention weight information content for each modification made by bpRNA-fix on Linearfold input for the test set. **B.** Incoming information content of an example fixed sequence (bpRNA_RFAM_1939), for the Attention Weights to each index. Note that below each plot in B and C, the dotbracekt and sequence are shown below. We see the incoming information content at layer 4, is higher in paired and fixed regions. **C.** Outgoing information content of the same sequence, for the attention weights leaving from each index. This example features prominent larger outgoing information content at a fix (removed hairpin).

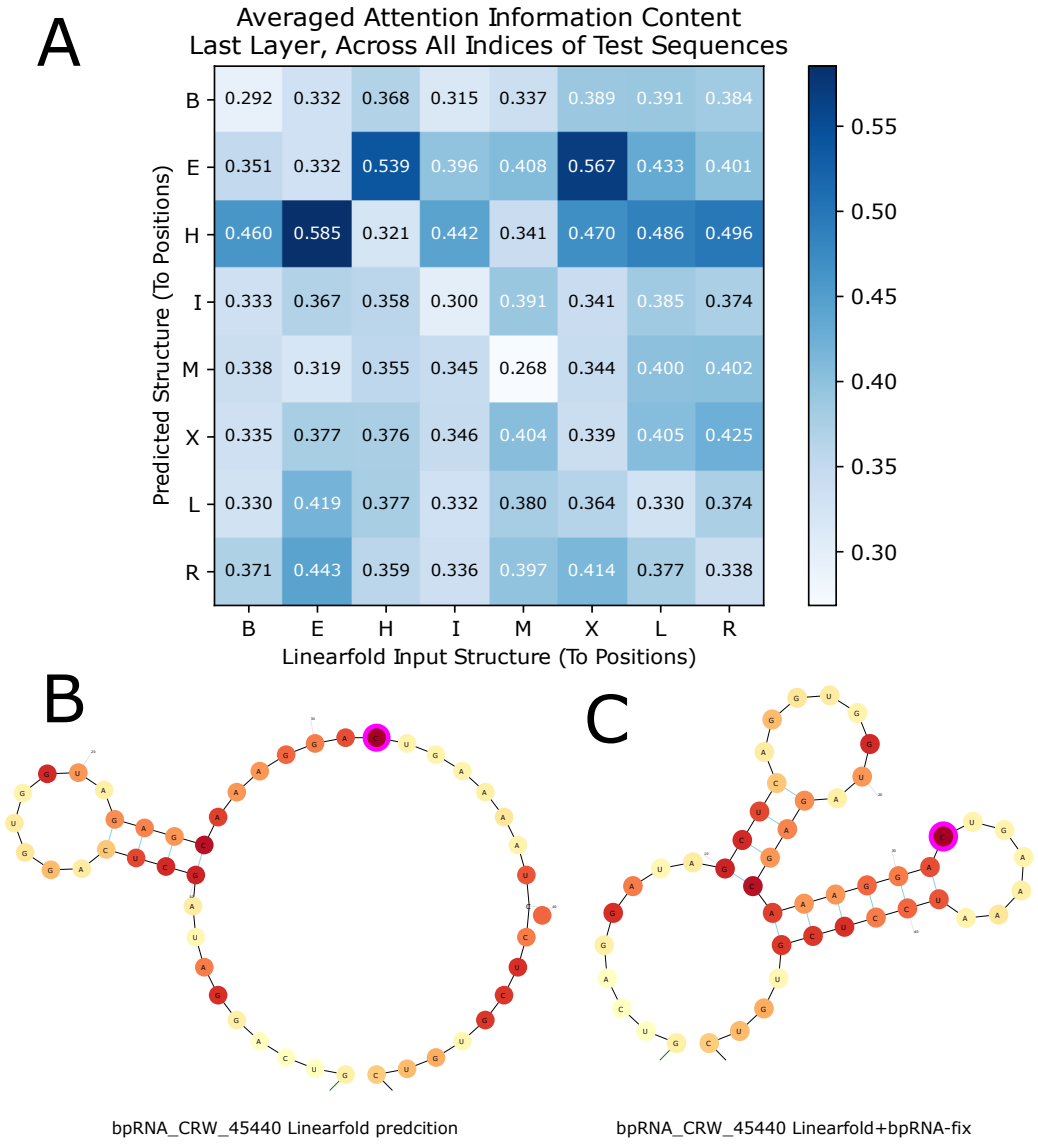

Figure 5: **A**: Average incoming information content in our model's last layer for each type of bpRNA structure label fix, going from Linearfold's input to bpRNA-fix's predicted output. Fixes consisting of an end (E) to a hairpin (H) have high average information Content. **B**: Example test sequence as paired by Linearfold with the fix position (E to H) highlighted in magenta, and with the attention weights to this fix position heatmapped in red. **C**: The resulting predicted output sequence from our model, with the same fix position highlighted and the same attention weights to this position, showing that attention across the linearfold input dotbracket correspond to structures in the predicted output. Attentions between the hairpins can also be seen, suggesting the modeling of pseudoknot structures.

positions that contribute to a particular fixed position. These can highlight regions that should be paired, or other potential positions that could interaction with the fixed position. We also more broadly demonstrate differentiation of roles across layers, as the level of information content for each type of fix changes from layer to layer. Future work could more precisely define the contribution of each layer in the computations that result in fixed positions.

While the input secondary structure prediction programs were designed for more general RNA folding task use cases, a major limitation of our current approach is that it relies on a source database, which will contain biases. Most available RNA secondary data sets have biases towards certain particular types or families (e.g. tRNAs), and any systematic errors or biases in the training set would of course be learned and affect the fixed base pairs. A limitation of our comparison to RNAfold and Linearfold is that those programs only predict canonical base pairs, and our data set does include non-canonical base pairs. Therefore, this gives bpRNA-fix an advantage when evaluated with bpRNA-1m over these programs because it was trained on examples with non-canonical base pairs. On the other hand, this provides more flexibility to bpRNA-fix's output compared to the thermodynamic models. Another limitation of our approach is the use of 90% similarity as the condition to split training, testing and validation data, which does not rule out the possibility of examples from the same RNA family occurring in both the training and testing data sets.

More generally, our approach of using one program's output as input to a transformer model can be readily applicable to other problem domains, especially those that utilize transformer models on paired structure prediction tasks. Going forward, our approach could be used to impute non-canonical interactions and pseudoknots using a scaffolding of canonical base pairs. Future data sets that are not limited to well-studied RNA families may ultimately provide more generalized prediction of RNA secondary structure when used as training data for similar approaches to bpRNA-fix. Finally, we have demonstrated that attention information content can be an effective way of interpreting the fixed positions of this kind of network, which could give new insight and help the development of improved thermodynamic models for traditional dynamic programming approaches. Code for this project is available at: `https://github.com/junekihong/bpRNA-fix`.

References

[1] Patrick Cramer. Alphafold2 and the future of structural biology. *Nature Structural & Molecular Biology*, 28(9):704–705, 2021.

[2] Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bprna: large-scale automated annotation and analysis of rna secondary structure. *Nucleic acids research*, 46(11):5381–5394, 2018.

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

[4] Chuong B. Do, Daniel A. Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, 07 2006.

[5] 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, 11 2021.

[6] Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

[7] Shujun He, Baizhen Gao, Rushant Sabnis, and Qing Sun. Nucleic transformer: Deep learning on nucleic acids with self-attention and convolutions. *bioRxiv*, 2021.

[8] Ivo L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, 07 2003.

[9] Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. MolTrans: Molecular Interaction Transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 10 2020.

[10] Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics*, 35(14):i295–i304, 07 2019.

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

[12] Nguyen Quoc Khanh Le, Quang-Thai Ho, Trinh-Trung-Duong Nguyen, and Yu-Yen Ou. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. *Briefings in Bioinformatics*, 22(5), 02 2021. bbab005.

[13] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[14] J. Padarian, A. B. McBratney, and B. Minasny. Game theory interpretation of digital soil mapping convolutional neural networks. *SOIL*, 6(2):389–397, 2020.

[15] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[16] Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications*, 12(1):1–9, 2021.

[17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *CoRR*, abs/1704.02685, 2017.

[18] Raphael J. L. Townshend, Stephan Eismann, Andrew M. Watkins, Ramya Rangan, Maria Karelina, Rhiju Das, and Ron O. Dror. Geometric deep learning of rna structure. *Science*, 373(6558):1047–1051, 2021.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[21] Zhongheng Zhang, Marcus W Beck, David A Winkler, Bin Huang, Wilbert Sibanda, Hemant Goyal, et al. Opening the black box of neural networks: methods for interpreting neural network models in clinical applications. *Annals of translational medicine*, 6(11), 2018.