RNAFLOW: RNA STRUCTURE & SEQUENCE CO-DESIGN VIA INVERSE FOLDING-BASED FLOW MATCHING

Divya Nori
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
divnor80@mit.edu

Wengong Jin
Broad Institute of MIT and Harvard
Cambridge, MA 02142, USA
wjin@broadinstitute.org

ABSTRACT

The growing significance of RNA engineering in diverse biological applications has spurred interest in AI methods for structure-based RNA design. Adapting existing biomolecular structure design methods for RNA presents challenges due to RNA's conformational flexibility and the computational cost of fine-tuning large structure prediction models. We propose RNAFlow, a flow matching model for protein-conditioned RNA sequence-structure co-design. Its denoising network integrates an RNA inverse folding model and a pre-trained RosettaFold2NA network for generation of RNA sequences and structures. We enhance the inverse folding model by conditioning it on inferred conformational ensembles to model the dynamic nature of RNAs. Evaluation on structure and sequence generation tasks demonstrates RNAFlow's advantage over existing RNA design methods.

1 INTRODUCTION

In recent years, RNA molecules have been engineered for versatile and controllable functions in biological systems (Dykstra et al., 2022). However, experimental methods for high-throughput RNA selection like SELEX (Gold, 2015) remain time-consuming and labor-intensive. To unlock the full potential of RNA therapeutics, there is a pressing need for developing deep learning models to automatically design RNAs that bind with a protein of interest (Sanchez de Groot et al., 2019).

State-of-the-art protein structure design method, RFDiffusion (Watson et al., 2023), fine-tuned RoseTTAFold (Baek et al., 2021) to generate protein structures that meet complex constraints, including conditional generation of protein binders. While we can adopt a similar approach for RNA by fine-tuning a protein-nucleic acid structure prediction network like RoseTTAFold2NA (RF2NA) (Baek et al., 2023), there are two major caveats. First, RNAs exhibit a high degree of conformational flexibility, which is often the key to their biological function (Ganser et al., 2019). A generative model that outputs a single structure may bottleneck the downstream sequence generation process. Second, fine-tuning a large structure prediction network is computationally expensive.

In this paper, we propose RNAFlow, a flow matching model for RNA sequence-structure co-design. The denoising network in RNAFlow is composed of an RNA inverse folding model and a pre-trained RF2NA network. In each iteration, RNAFlow first generates a RNA sequence given a noisy protein-RNA complex and then uses RF2NA to fold into a denoised RNA structure. For computational efficiency, we train the inverse folding model to minimize the flow matching objective while keeping RF2NA fixed. RNAFlow offers three advantages over previous methods. First, RNAFlow generates an RNA sequence and its structure simultaneously. Second, it is much easier to train because we do not fine-tune a large structure prediction network. Third, our framework enables us to model the dynamic nature of RNA structures for inverse folding. Specifically, we utilize the final few structure predictions from inference trajectories as an effective approximation of an RNA conformational ensemble and enhance our inverse folding model to condition on dynamic RNA conformations.

2 Methods

Problem Formulation. RNAFlow receives the protein backbone atom structure $\vec{P} \in \mathbb{R}^{L_p \times 3 \times 3}$, where L_p is the number of residues, and each residue contains backbone atoms N, C_α, and C. The protein sequence is also given as input where a single token is p_i, $i \in \{0, 1, 2, \ldots, L_p - 1\}$. The model is trained to predict RNA sequence with tokens $r_i, i \in \{0, 1, 2, \ldots, L_r - 1\}$. The model also predicts RNA backbone structure $R \in \mathbb{R}^{L_r \times 3 \times 3}$, where L_r is the number of nucleotides, and each nucleotide contains backbone atoms P, C_4', and N_1/N_9 (pyrimidine/purine) (Wadley et al., 2007).

Background: Conditional Flow-Matching. By the flow matching framework (Lipman et al., 2022), consider data distribution p_1 of RNA backbone structures and a prior distribution $p_0(\vec{R}|\vec{R}_1)$, where $\vec{R}_1$ is a sample from p_1. A *flow* transforms p_0 to p_1, and this flow can be parameterized by a sequence of time-dependent conditional probability paths $p_t(\vec{R}_t|\vec{R}_1)$, $t \in [0, 1]$. A sample from p_t can be computed by linear interpolation, as given by Equation 1, where $\vec{R}_0$ is a sample from the prior distribution.

$$\vec{R}_t|\vec{R}_1 = (1 - t) * \vec{R}_0 + t * \vec{R}_1 \tag{1}$$

To generate new data with these conditional flows, we define the time-dependent conditional vector field $v_t(\vec{R}|\vec{R}_1)$. v_t gives the dynamics of the flow at each point along the conditional probability path, and this vector field can be integrated from time 0 to time 1 to generate samples from p_1 given a noise sample from p_0. We aim to learn this vector field with a neural network trained with the reparameterized conditional flow matching objective. Particularly, as with diffusion, the neural network can conveniently be trained to predict samples from the data distribution.

We choose to employ flow matching rather than the related diffusion framework (Ho et al., 2020), primarily due to computational efficiency. Diffusion inference often requires thousands of forward passes to generate quality samples. Since our score model involves a large structure prediction network, RNAFlow inference time would be much slower in a diffusion setting (Yim et al., 2023). In contrast, quality samples can be generated by flow matching with a much fewer number of passes.

2.1 RNAFlow: Overview

We generate RNA sequences and structures using a conditional flow matching model. The flow matching score predictor is an inverse folding denoiser (**Noise-to-Seq**) and pre-trained folding network. Noise-to-Seq is a geometric graph neural network conditioned on protein structure and sequence, pre-trained on the RNA inverse folding task and subsequently fine-tuned on the flow matching objective. It follows the RNA inverse folding architecture presented by Joshi et al. (2023).

Initialization. As our prior distribution, we choose a unit Gaussian on $\mathbb{R}^3$ centered at zero, from which we sample $\vec{R}_0$. We also translate the true protein-RNA complex from our target data distribution $[\vec{P}_1, \vec{R}_1]$ such that the center of mass of $\vec{R}_1$ is zero. By this approach, we train with the ground-truth pose of the protein relative to the RNA, though this is not leaked at inference time.

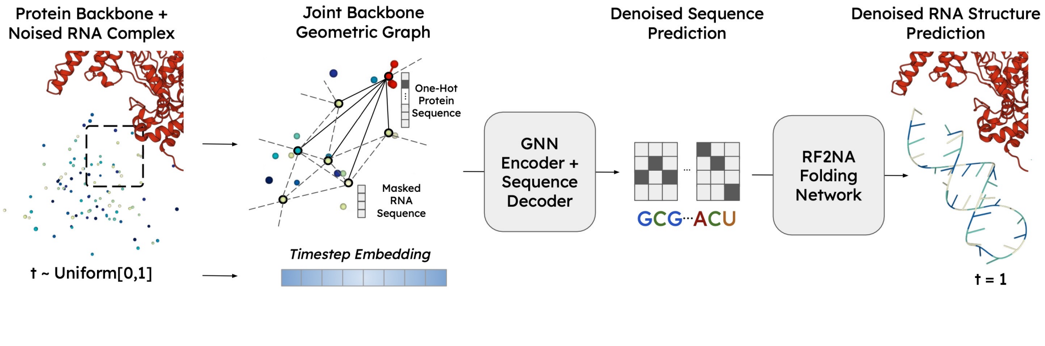

Figure 1: One forward pass in RNAFlow training.

Training. As shown in Figure 1, during training, we sample a timestep t and interpolate the true RNA backbone with the sampled prior to arrive at noised backbone $\vec{R}_t$. The true protein backbone, noised RNA backbone, and protein sequence are given as input to Noise-to-Seq which predicts a denoised RNA sequence $\{\hat{r}_i\}_{\forall i}$. RF2NA folds $\{\hat{r}_i\}_{\forall i}$ into predicted structure $\hat{R}_1$. The reparameterized flow matching objective gives that the score model should be trained with a standard L_2 loss. Therefore, as given in Equation 2, we compute the MSE between all predicted and true RNA backbone coordinates. Before MSE computation, we Kabsch align the two RNA backbones. Additionally, we supervise the sequence with cross entropy between predicted and true nucleotide types.

$$\mathcal{L} = \text{MSE}(\hat{R}_1, \vec{R}_1) + \sum_i \text{CE}(\hat{r}_i, r_i) \tag{2}$$

The sequence output from Noise-to-Seq is used as input to RF2NA, and the resulting structure is supervised. Since gradients must propagate between Noise-to-Seq and RF2NA, we apply the Gumbel-Softmax estimator to differentiably sample from the predicted logits (Jang et al., 2016).

Inference: RNAFlow-Base. To avoid leaking the ground-truth pose of the RNA relative to the protein, we generate a “pose guess” by folding a mock RNA sequence of all adenines and true protein MSA with RF2NA. Prior $\vec{R}_0$ is drawn from the prior distribution, and $\vec{P}_0$ is the true protein backbone aligned to the centered pose guess. Noise-to-Seq then predicts denoised sequence $\{\hat{r}_i\}_{\forall i}$, which is folded into $\hat{R}_1$. $\hat{R}_1$ is linearly interpolated with the predicted backbone from the previous timestep to re-noise the structure. Thus, RNAFlow iteratively predicts the true complex.

Inference: RNAFlow-Traj. In standard RNAFlow inference, Noise-to-Seq predicts a sequence based on a single RNA structure. However, an RNA backbone structure can adopt multiple conformations, and it is important that the final RNA sequence reflects this conformational diversity (Ganser et al., 2019). Indeed, Joshi et al. (2023) demonstrates that inverse folding on a set of RNA conformations improves the native sequence recovery rate compared to single-structure model. To this end, we propose to generate a final RNA sequence by conditioning on multiple RNA structures generated over the course of flow matching inference. This inverse folding model, referred to as **Trajectory-to-Seq (Traj-to-Seq)**, is a multi-graph neural network which can handle multiple RNA conformation inputs. Its architecture is detailed in section 2.3.

Output Rescoring Model. Since RNAFlow can be sampled many times to obtain several RNA designs, we train an output rescoring model to select from amongst the samples based on predicted recovery rate. Details are given in A.6.

2.2 Noise-to-Seq Module

Graph Representation. We represent each backbone 3D point cloud $[\vec{P}, \vec{R}]$ as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each amino acid i is assigned a node at C_α coordinate $\vec{x_i} \in \vec{P}$, and each nucleotide j is assigned a node at C_4' coordinate $\vec{x_j} \in \vec{R}$. We construct a nearest neighbors graph where node features are denoted as $\mathcal{V} = \{v_1, \ldots, v_{L_p+L_r}\}$ and edge features are denoted as $\mathcal{E} = \{e_{i,j}\}_{i \neq j}$. Graph construction and featurization details are given in A.4.

Model. Noise-to-Seq is a graph-based inverse folding model where node and edge features are encoded by message-passing GVP-GNN layers (Jing et al., 2020). The architecture is given in A.5. We then incorporate protein and RNA sequence information with an embedding layer and concatenate these features onto the edge features. By the autoregressive scheme, we mask out sequence features for edges where the source node’s position is greater than the destination node’s position. Our decoder’s architecture is identical to the encoder. Finally, we apply a softmax to predict probabilities of the 4 nucleotide classes at each position, which is supervised by a cross entropy loss.

2.3 Traj-to-Seq Module

Motivation. As shown in Figure 2, flow matching inference generates a trajectory of RNA structures over refinement steps. Traj-to-Seq is a graph-based inverse folding model that predicts RNA sequences based on a flow matching output trajectory. At the final stages of the trajectory, the structures have a high degree of secondary structure similarity while displaying variation in tertiary structure, resembling a conformational ensemble (Fornili et al., 2013). Therefore, we can leverage the outputs of a flow matching inference pass as a conformational ensemble approximation.

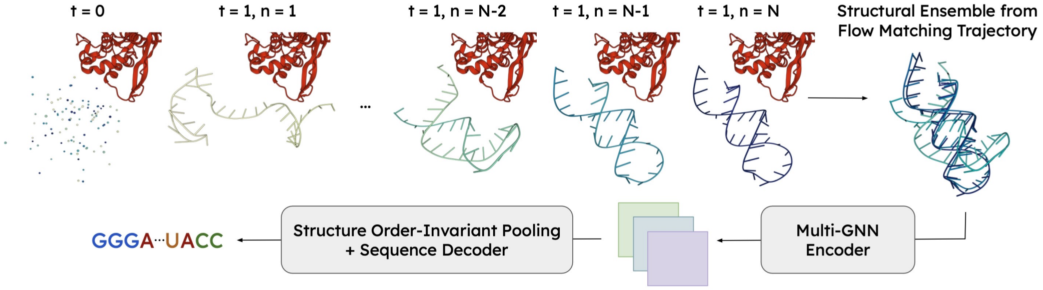
t = 0
t = 1, n = 1
t = 1, n = N-2
t = 1, n = N-1
t = 1, n = N
Structural Ensemble from Flow Matching Trajectory
GGGA···UACC
Structure Order-Invariant Pooling + Sequence Decoder
Multi-GNN Encoder

Figure 2: One forward pass during Traj-to-Seq inference.

Model Inputs. We represent a trajectory of RNA backbones as a set of independent graphs $\{\mathcal{G}^{(1)}, ..., \mathcal{G}^{(k)}\}$ where k is the number of conformers. Each node in $\mathcal{G}^{(n)}$ corresponds to a nucleotide j at C_4' coordinate $\vec{x_j} \in \vec{R}^{(n)}$. Nearest neighbor edges are drawn between nodes of the same conformer graph before constructing a "multi-graph." As introduced by Joshi et al. (2023), a multi-graph is constructed by stacking each RNA graph's scalar and vector features and building a joint adjacency matrix, computed as the union of each individual graph's adjacency matrix.

Architecture. Traj-to-Seq employs the same encoder-decoder architecture as Noise-to-Seq. The GVP encoder processes each structure independently, such that $h_{v_i} \in \mathbb{R}^{k \times f}$ where f is the feature dimension. A merged representation is computed by a structure order-invariant mean. Finally, a decoder identical to Noise-to-Seq is employed to predict RNA sequence.

3 Experiments

We evaluate RNAFlow in two experimental settings. First, we evaluate RNA sequence and structure prediction error with respect to existing protein-RNA complexes. Second, we evaluate whether RNAFlow can design RNA aptamers for the GRK2 protein, given a known 4-nucleotide RNA binding sequence motif (Mayer et al., 2008; Lennarz, 2015).

3.1 Structure & Sequence Generation

As mentioned, we evaluate two variants of our method: *RNAFlow-Base* which is based on Noise-to-Seq and *RNAFlow-Trajectory* which additionally applies Traj-to-Seq. For both methods, we begin by generating 10 structure-sequence pairs per protein in the test set. From amongst the 10 designs, we additionally report the performance when 1 sample is selected by the output rescoring model. All samples are generated with 5 integration steps.

Dataset. Protein-RNA complexes from the PDBBind dataset (2020 version) were used for training and evaluation (Liu et al., 2017). Dataset preparation details are described in A.8. We perform all experiments on two splits. The first accounts for RF2NA pre-training – all examples from complexes in the RF2NA validation or test sets were assigned to the test split, and remaining examples were randomly split into training and validation in a 9:1 ratio. We also evaluate on an RNA sequence similarity split. All RNA chains were clustered using CD-HIT (Fu et al., 2012) where sequences in the same cluster share $\geq 80\%$ sequence identity, following the clustering protocol of Joshi et al. (2023). The clusters were randomly split into train, validation, and test in an 8:1:1 ratio.

Traj-to-Seq was trained separately on RNASolo (Adamczyk et al., 2022), a dataset of RNA sequences and structures where many of the sequences are associated with multiple structure conformers. At inference time, Traj-to-Seq is run on the final 3 structures in the flow matching trajectory.

Baselines. For structure generation, we compare against MMDiff (Morehead et al., 2023), an $SE(3)$-discrete diffusion model that designs nucleic acid sequences and structures. We sample 10 RNA designs of the ground-truth RNA's length per protein-RNA complex in our test set, con-

Method	*RF2NA Pre-Training Split*		*Sequence Similarity Split*	
	RMSD	lDDT	RMSD	lDDT
Conditional MMDiff	14.82 ± 1.01	0.34 ± 0.02	17.42 ± 0.86	0.38 ± 0.01
RNAFlow-Base	12.85 ± 0.63	0.51 ± 0.01	14.77 ± 0.34	**0.57 ± 0.01**
RNAFlow-Traj	13.12 ± 0.64	0.52 ± 0.01	15.11 ± 0.33	0.57 ± 0.00
RNAFlow-Base + Rescore	**10.61 ± 1.73**	**0.53 ± 0.03**	**14.60 ± 1.05**	0.56 ± 0.02
RNAFlow-Traj + Rescore	15.30 ± 1.89	0.52 ± 0.03	15.31 ± 0.93	0.56 ± 0.02
RF2NA [Upper Bound]	4.67 ± 1.29	0.76 ± 0.04	9.83 ± 1.69	0.79 ± 0.02

Table 1: Structure generation results. We report Mean ± SEM for RMSD and lDDT. RMSD is computed after Kabsch alignment with the true RNA. lDDT is computed on C_α atoms.

Method	*RF2NA Pre-Training Split*	*Sequence Similarity Split*
	Recovery	Recovery
Random	0.25 ± 0.00	0.25 ± 0.00
LSTM	0.27 ± 0.01	0.24 ± 0.01
Conditional MMDiff	0.24 ± 0.02	0.22 ± 0.02
RNAFlow-Base	0.30 ± 0.02	0.30 ± 0.01
RNAFlow-Traj	0.31 ± 0.01	0.28 ± 0.01
RNAFlow-Base + Rescoring	0.33 ± 0.02	**0.32 ± 0.03**
RNAFlow-Traj + Rescoring	**0.37 ± 0.05**	0.29 ± 0.02
Inverse Folding [Upper Bound]	0.46 ± 0.01	0.35 ± 0.01

Table 2: Sequence generation results. We report Mean ± SEM for native sequence recovery.

ditioned on the protein backbone structure and sequence. As an upper bound, we compare against RNA structures folded by RF2NA, given the ground-truth RNA sequence.

For sequence generation, we compare to a random baseline where RNA sequences of specified length are generated by randomly selecting a nucleotide for each position. Next, we compare against sequences designed by MMDiff. Our third baseline is a standard sequence-only model, consisting of an LSTM layer to encode protein sequence and an autoregressive LSTM decoder layer to predict RNA sequence. We establish an upper bound by comparing against results from our pre-trained inverse folding model, which takes the ground-truth protein-RNA backbone complex as input.

Results. As shown in Table 1, RNAFlow outperforms the baseline on the structure generation task. On the RF2NA and sequence similarity splits respectively, the best RNAFlow model gives a 28% and 16% reduction in RMSD relative to MMDiff. Without rescoring, RNAFlow gives a 15% reduction in RMSD on both splits. Ablations and visualizations of RNA designs are given in A.7 and A.10.

As shown in Table 2, RNAFlow outperforms all baselines on the sequence generation task. On the RF2NA split, the best RNAFlow model gives a 48% improvement over random generation and a 54% improvement over MMDiff. RNAFlow-Trajectory outperforms RNAFlow-Base, suggesting that flow matching trajectories provide useful RNA dynamics information that is not present in a single structure. Likewise, on the sequence similarity split, the best RNAFlow model gives a 28% improvement over random generation and a 45% improvement over MMDiff. RNAFlow-Trajectory does not improve recovery, indicating that the generated trajectories may not approximate conformational ensembles when trying to generalize to biologically dissimilar RNAs.

3.2 Motif-Scaffolded Design of RNAs for GRK2

Design Procedure. For this task, we train RNAFlow on all protein-RNA examples except the ground-truth GRK2-C28 protein-aptamer complex. We evaluate the structure and sequence accuracy of GRK2-conditioned RNA design given a 4-nucleotide binding site sequence motif. For all methods, we sample 10 RNA designs and select 1 top design by the output rescoring model.

Results. As shown in Table 3, RNAFlow outperforms the baselines in terms of both RMSD and recovery. Compared to MMDiff, RNAFlow-Trajectory gives an 22.85% improvement in RMSD and a 68.75% improvement in recovery. Crucially, we observe that the sequence predicted by RNAFlow-Trajectory recovers nucleotides on either end of the aptamer, far away from the given binding motif. As noted by Tesmer et al. (2012), these stem regions are critical for the aptamer's high affinity

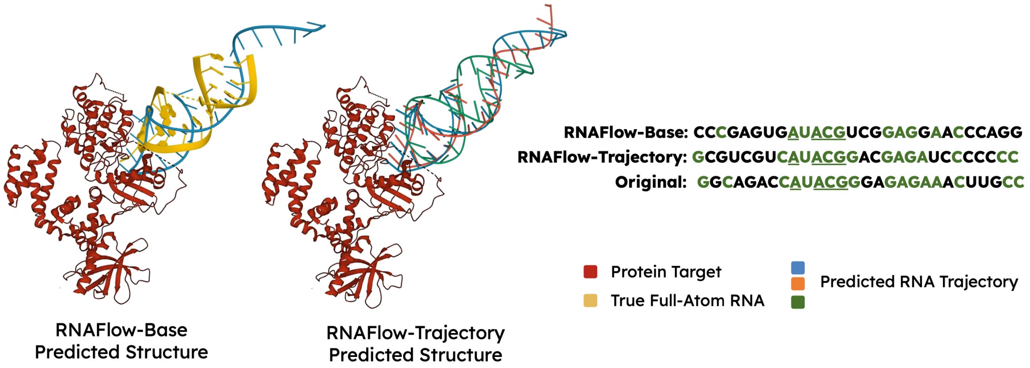
RNAFlow-Base: CCCGAGUGAUACGUCGGAGGAACCCAGG
RNAFlow-Trajectory: GCGUCGUCAUACGGACGAGAUCCCCCCC
Original: GGCAGACCAUACGGGAGAGAAACUUGCC
🟥 Protein Target
🟨 True Full-Atom RNA
🟦 🟧 🟩 Predicted RNA Trajectory
RNAFlow-Base Predicted Structure
RNAFlow-Trajectory Predicted Structure

Figure 3: Designed RNAs for GRK2 binding. Green-colored characters show nucleotides that are correctly recovered from the ground-truth. Underlined nucleotides are part of the given motif.

	RMSD	Recovery
LSTM	-	0.29
MMDiff	9.19	0.32
RNAFlow-Base	8.44	0.43
RNAFlow-Trajectory	**7.09**	**0.54**

Table 3: GRK2 binder metrics.

because they “limit the number of possible conformations for the selected RNA.” Thus, we argue that having conformational information is important to design these regions.

4 Conclusion

In this paper, we present RNAFlow, the first protein-conditioned generative model for RNA structure and sequence co-design. We show that an inverse folding model is an effective score prediction network within the flow matching framework, enabling the design of RNAs that outperform the baselines in terms of structure and sequence accuracy. Additionally, we show that by inverse folding over an inferred conformational ensemble, we can design plausible aptamers for GRK2 binding.

References

Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. Rnasolo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 38(14):3668–3670, 2022.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature Methods*, pp. 1–5, 2023.

Avishek Joey Bose, Tara Akhound-Sadegh, Kilian Fatras, Guillaume Huguet, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Andrey A Buglak, Alexey V Samokhvalov, Anatoly V Zherdev, and Boris B Dzantiev. Methods and applications of in silico aptamer design and modeling. *International Journal of Molecular Sciences*, 21(22):8420, 2020.

Peter B Dykstra, Matias Kaplan, and Christina D Smolke. Engineering synthetic rna devices for cell control. *Nature Reviews Genetics*, 23(4):215–228, 2022.

Arianna Fornili, Alessandro Pandini, Hui-Chun Lu, and Franca Fraternali. Specialized dynamical properties of promiscuous residues revealed by simulated conformational ensembles. *Journal of chemical theory and computation*, 9(11):5127–5147, 2013.

Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

Laura R Ganser, Megan L Kelly, Daniel Herschlag, and Hashim M Al-Hashimi. The roles of structural dynamics in the cellular functions of rnas. *Nature reviews Molecular cell biology*, 20(8): 474–489, 2019.

Larry Gold. Selex: How it happened and where it will go. *Journal of Molecular Evolution*, 81(5-6): 140–143, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jinho Im, Byungkyu Park, and Kyungsook Han. A generative model for constructing nucleic acid sequences binding to a protein. *BMC genomics*, 20(13):1–13, 2019.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

Natsuki Iwano, Tatsuo Adachi, Kazuteru Aoki, Yoshikazu Nakamura, and Michiaki Hamada. Generative aptamer discovery using raptgen. *Nature Computational Science*, 2(6):378–386, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021.

Chaitanya K. Joshi, Arian R. Jamasb, Ramon Viñas, Charles Harris, Simon Mathis, and Pietro Liò. Multi-state rna design with geometric multi-graph neural networks. In *ICML 2023 Workshop on Computation Biology*, 2023.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32 (5):922–923, 1976.

Namhee Kim, Joseph A Izzo, Shereef Elmetwaly, Hin Hark Gan, and Tamar Schlick. Computational generation and screening of rna motifs in large nucleotide sequence pools. *Nucleic acids research*, 38(13):e139–e139, 2010.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *arXiv preprint arXiv:2306.15030*, 2023.

Dennis M Krüger, Saskia Neubacher, and Tom N Grossmann. Protein–rna interactions: structural characteristics and hotspot amino acids. *Rna*, 24(11):1457–1465, 2018.

Sabine Michaela Irmgard Lennarz. *RNA aptamers as selective protein kinase inhibitors*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2015.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50 (2):302–309, 2017.

Günter Mayer, Bernhard Wulffen, Christian Huber, Jörg Brockmann, Birgit Flicke, Lars Neumann, Doris Hafenbradl, Bert M Klebl, Martin J Lohse, Cornelius Krasel, et al. An rna molecule that specifically inhibits g-protein-coupled receptor kinase 2 in vitro. *Rna*, 14(3):524–534, 2008.

Alex Morehead, Jeffrey Ruffolo, Aadyot Bhatnagar, and Ali Madani. Towards joint sequence-structure generation of nucleic acid and protein complexes with se (3)-discrete diffusion. *arXiv preprint arXiv:2401.06151*, 2023.

Natalia Sanchez de Groot, Alexandros Armaos, Ricardo Graña-Montes, Marion Alriquet, Giulia Calloni, R Martin Vabulas, and Gian Gaetano Tartaglia. Rna structure drives interaction with proteins. *Nature communications*, 10(1):3246, 2019.

Hannes Stark, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Harmonic prior self-conditioned flow matching for multi-ligand docking and binding site design. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Hanqun Cao, and Stan Z. Li. Hierarchical data-efficient representation learning for tertiary structure-based rna design, 2023.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

Valerie M Tesmer, Sabine Lennarz, Günter Mayer, and John JG Tesmer. Molecular mechanism for inhibition of g protein-coupled receptor kinase 2 by a selective rna aptamer. *Structure*, 20(8): 1300–1309, 2012.

Leven M Wadley, Kevin S Keating, Carlos M Duarte, and Anna Marie Pyle. Evaluating and learning from rna pseudotorsional space: quantitative validation of a reduced representation for rna structure. *Journal of molecular biology*, 372(4):942–957, 2007.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Joseph D Yesselman, Daniel Eiler, Erik D Carlson, Michael R Gotrik, Anne E d'Aquino, Alexandra N Ooms, Wipapat Kladwang, Paul D Carlson, Xuesong Shi, David A Costantino, et al. Computational design of three-dimensional rna structure and function. *Nature nanotechnology*, 14(9): 866–873, 2019.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Qingtong Zhou, Xiaole Xia, Zhaofeng Luo, Haojun Liang, and Eugene Shakhnovich. Searching the sequence space for potent aptamers using selex in silico. *Journal of chemical theory and computation*, 11(12):5939–5946, 2015.

A Appendix

A.1 Related Work

Protein-Conditioned RNA Design. Early methods for computational design of protein-binding RNAs involved generating a large number of RNA sequences and selecting by desired secondary structure motifs Kim et al. (2010) or molecular dynamics Zhou et al. (2015); Buglak et al. (2020). These approaches are computationally expensive and require specifying design constraints a priori, which are often unknown for new protein targets. More recently, generative sequence-based approaches based on LSTMs and VAEs have been trained on SELEX data Im et al. (2019); Iwano et al. (2022). However, these methods can only be trained for one protein at a time and cannot be applied to proteins where SELEX data does not exist.

Unconditional RNA Design. Computational methods have been developed for RNA structure design, including classical algorithmic approaches Yesselman et al. (2019) and generative modeling methods. In particular, Morehead et al. (2023) recently proposed an $SE(3)$-discrete diffusion model (MMDiff) for joint generation of nucleic acid sequences and structures. While MMDiff can generate short micro-RNA molecules, it has trouble conditionally generating protein-binding RNAs and samples of longer sequence length.

Deep learning methods for inverse folding generally apply graph-based encoders for a single RNA structure Tan et al. (2023) or several conformers Joshi et al. (2023). We adapt an inverse folding model to accept protein information as a condition and train on a denoising objective.

Protein Structure Design. Deep generative methods for protein design have demonstrated that realistic protein backbones can be generated efficiently via flow matching Yim et al. (2023); Bose et al. (2023). Prior work has also shown that diffusion models can be conditioned on a target protein to generate functional protein binders Ingraham et al. (2023); Watson et al. (2023). We apply a similar approach within the flow matching framework. Our work is also related to sequence-structure co-design methods where the outputs are iteratively refined Jin et al. (2021); Stark et al. (2023).

A.2 RNAFlow Training and Inference Algorithms

Training. As mentioned, we choose a unit Gaussian on $\mathbb{R}^3$ centered at zero as our prior distribution, from which we sample $\vec{R}_0$. We also translate the true protein-RNA complex $[\vec{P}_1, \vec{R}_1]$ such that the center of mass of $\vec{R}_1$ is zero. As shown in Algorithm 1, we Kabsch align Kabsch (1976) the sampled noise $\vec{R}_0$ with the true RNA $\vec{R}_1$ to eliminate global rotations, leading to faster training convergence Klein et al. (2023).

Inference. As shown in Algorithm 2, we begin by generating a "pose guess" using RF2NA. Specifically, we fold the protein MSA with a mock RNA sequence consisting of all adenines, of the same length as the true RNA sequence. The predicted complex is used as an initial guess of protein pose with respect to the RNA centroid. We translate the pose guess such that the RNA centroid is zero. $\vec{P}_0$ is the true protein backbone Kabsch-aligned onto the predicted protein pose, and $\vec{R}_0$ is drawn from the prior distribution. After $\hat{R}_1$ prediction, $\hat{R}_1$ is interpolated with the prior to re-noise the structure for the subsequent inference iteration.

Algorithm 1 RNAFlow: Train

```
Require: $\{p_i\}_{\forall i}$, $\{r_i\}_{\forall i}$, $[\vec{P}_1, \vec{R}_1]$
  Sample prior $\vec{R}_0 \sim \mathcal{N}(0, I_3)^{L_r}$
  Kabsch align noise $\vec{R}_0$ with true backbone $\vec{R}_1$
  Sample timestep $t \sim$ Uniform$[0, 1]$
  Interpolate $\vec{R}_t \leftarrow t * \vec{R}_1 + (1 - t) * \vec{R}_0$
  Predict $\{\hat{r}_i\}_{\forall i} \leftarrow$ Noise-to-Seq$\{[\vec{P}_1, \vec{R}_t], \{p_i\}_{\forall i}, t\}$
  Predict $\hat{R}_1 \leftarrow$ RF2NA$\{\{\hat{r}_i\}_{\forall i}\}$
```

Algorithm 2 RNAFlow-Traj: Inference

```
Require: $\{p_i\}_{\forall i}$, $\vec{P}_1$
  Position $\vec{P}_0$ by aligning $\vec{P}_1$ with RF2NA pose guess
  Sample prior $\vec{R}_0 \sim \mathcal{N}(0, I_3)^{L_r}$
  Initialize $traj \leftarrow []$
  for $n \leftarrow 1$ to $N$ do
      Let $t_2 \leftarrow n/N$ and $t_1 \leftarrow (n-1)/N$
      $\{\hat{r}_i\}_{\forall i} \leftarrow$ Noise-to-Seq$\{[\vec{P}_{t_1}, \vec{R}_{t_1}], \{p_i\}_{\forall i}, t_1\}$
      Predict $[\hat{P}_1, \hat{R}_1] \leftarrow$ RF2NA$\{\{\hat{r}_i\}_{\forall i}$, protein MSA$\}$
      Append $\hat{R}_1$ to $traj$
      Interpolate $\vec{R}_{t_2} \leftarrow \vec{R}_{t_1} + \frac{(\hat{R}_1 - \vec{R}_{t_1})}{(1-t_1)} * (t_2 - t_1)$
      Align $\vec{P}_{t_2} \leftarrow$ Kabsch$(\vec{P}_1, \hat{P}_1)$
  end for
  Predict $\{\hat{r}_i\}_{\forall i} \leftarrow$ Traj-to-Seq$\{traj\}$
  Outputs: $\{\hat{r}_i\}_{\forall i}$, $traj[-1]$
```

A.3 Noise-to-Seq Graph Construction

Recall that we represent each backbone 3D point cloud [$\vec{P}$, $\vec{R}$] as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each amino acid i is assigned a node at C_α coordinate $\vec{x_i} \in \vec{P}$, and each nucleotide j is assigned a node at C'_4 coordinate $\vec{x_j} \in \vec{R}$. For all nodes $\vec{x_i} \in \vec{P}$, we draw edges to 10 nearest neighbor nodes whose 3D coordinates are also in $\vec{P}$. Likewise for all nodes $\vec{x_i} \in \vec{R}$, we draw edges to 10 nearest neighbor nodes whose 3D coordinates are also in $\vec{R}$. For all nodes $\vec{x_i} \in \vec{P}$, we additionally draw edges to 5 nearest neighbor nodes $\vec{x_j} \in \vec{R}$. Nearest neighbors are computed by $\|\vec{x_i} - \vec{x_j}\|_2$.

We compute node features and edge features as follows. Node vector features include unit vectors to neighboring nodes on the backbone chain and unit vectors to the remaining two atoms of the nucleotide (P and N_1/N_9) or amino acid (N and C). Node scalar features include magnitudes of all vector features, encoded by radial basis functions, and a one hot encoding of residue identity for the protein chain. Edge vector features include a unit vector from the source node to the destination node, and its magnitude encoded by radial basis functions is a scalar feature. Edge scalar features also include distance along the backbone between the source and destination, encoded with a sinusoidal embedding. If the edge connects nodes in two different chains, the encoded number is equal to $L_p + L_r$.

A.4 Noise-to-Seq Architecture

Here, we give the Noise-to-Seq encoder architecture. First, input node features v_i and edge features $\{e_{i,j}\}_{i \neq j}$ are encoded by GVPs.

$$h_{v_i} = g_v(\text{LayerNorm}(v_i))$$
$$h_{e_{i,j}} = g_e(\text{LayerNorm}(e_{i,j}))$$

Node embedding GVP g_v and edge embedding GVP g_e apply the vector gating strategy proposed by Jing et al. (2021). If the input structure is noised, we add a timestep embedding to the node features, encoded by random Fourier features of size 256 Tancik et al. (2020) and concatenated onto the scalar component of h_{v_i}. For every position i, h_{v_i} is residually updated by a sequence of three message-passing layers where each layer has the following architecture:

$$m_{v_i} = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} g_{MSG}(h_{v_i}, h_{v_j}, h_{e_{i,j}})$$
$$h^{'}_{v_i} = \text{LayerNorm}(h_{v_i} + \text{Dropout}(m_{v_i}))$$

g_{MSG} is a sequence of three GVPs with vector gating and ReLU activation on the scalar features. After each message-passing layer, we also apply pointwise feedforward updates to the node embedding.

$$h_{v_i}^{'} = \text{LayerNorm}(h_{v_i}^{'} + \text{Dropout}(g_{FF}(h_{v_i}^{'})))$$

g_{FF} is a sequence of two GVPs with vector gating and the ReLU activation on the scalar features.

For both splits, the Noise-to-Seq encoder and decoder GVP layers use a node scalar feature dimension of 128, node vector feature dimension of 16, edge scalar feature dimension of 32, and edge vector feature dimension of 1. On both splits, the model was pre-trained for 100 epochs using an Adam optimizer with a learning rate of 0.001, which takes a few hours on an NVIDIA A5000-24GB GPU.

A.5 Output Rescoring Model

To train the output rescoring model, we generate 6 mutated RNA sequences for each ground-truth protein-RNA complex. Binary labels are generated based on whether or not a generated RNA sequence has a recovery rate $\geq 30\%$. At test time, we evaluate many RNAFlow output samples for a single protein and make selections based on the highest predicted probability of a positive outcome, prioritizing designs with the greatest likelihood of achieving a recovery rate of at least 30%.

The inputs to the output scoring model are protein sequence, protein structure, RNA sequence, and RF2NA-folded RNA structure. The protein and RNA are each encoded by a GVP model with the same architecture as the Noise-to-Seq encoder, and the resulting node-level representations are averaged and processed by a feed-forward network. The output is supervised by a binary cross entropy loss.

Following Noise-to-Seq, the encoder and decoder GVP layers use a node scalar feature dimension of 128, node vector feature dimension of 16, edge scalar feature dimension of 32, and edge vector feature dimension of 1. The node output feature dimension is 256, and the final feedforward network consists of three fully connected layers with ReLU activation. The model was trained for 50 epochs using an Adam optimizer with a learning rate of 0.01.

A.6 Ablation Study

We conduct an ablation study to determine which components of our model contribute to its performance. As shown in Figure 4, we report recovery rate and RMSD on the sequence similarity split under various conditions, since the sequence split is more general to a design setting. First, we fine-tune Noise-to-Seq on sequence cross-entropy only, removing the structure MSE loss. When this denoiser is used in the standard five-step flow matching loop, the RMSD increases by 1.25 compared to RNAFlow-Base, and recovery rate decreases by 1 percentage point. Next, we remove protein conditioning and find that RMSD increases by 2.52, and recovery drops by 7 percentage points, showing that providing protein information is important.

A.7 Dataset Details

A.7.1 PDBBind

We filter PDBBind to complexes where at least one protein C_α atom and RNA C_4' atom were within 7 Å, a threshold that has been used to perform alanine scans for protein-RNA interaction site analysis Krüger et al. (2018). For complexes containing many protein-RNA interaction sites, we use the interaction with least distance between the protein C_α atom and RNA C_4' atom. We filter to RNA chains of length ≥ 6 and ≤ 96, and protein chains are contiguously cropped to length 50. As shown in Figure 5, while there are many examples of short RNAs, the model is also exposed to several longer RNA examples.

In the sequence similarity split, there are 1015 complexes in train, 105 in validation, and 72 in test. The RF2NA split has 1059 complexes in train, 117 in validation, and 16 in test.

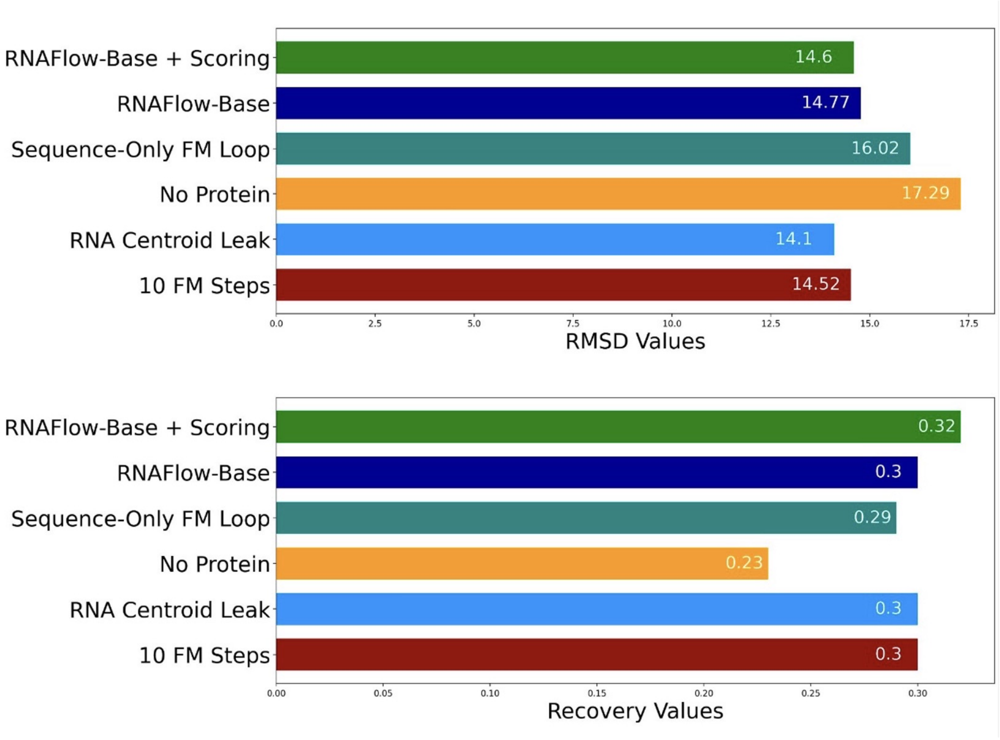

Figure 4: Ablation study of RNAFlow components. We report RMSD and sequence recovery on the sequence similarity split.

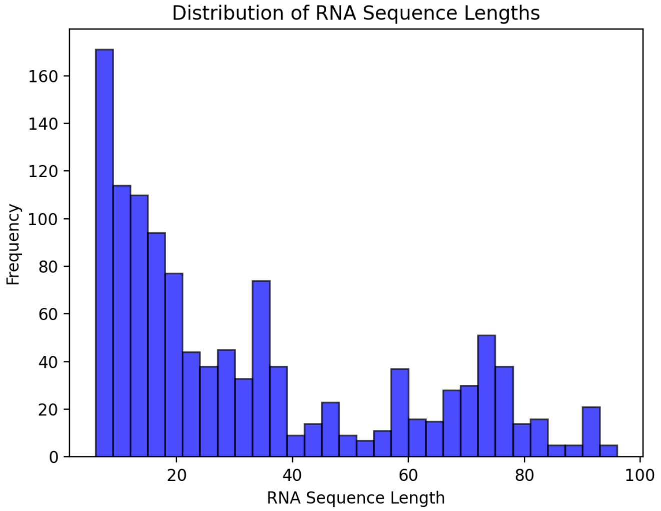

Figure 5: Distribution of RNA lengths in processed PDBBind dataset.

A.7.2 RNASolo

As mentioned, we use the RNASolo dataset for Traj-to-Seq training. The dataset consists of RNAs from apo RNA structures, protein-RNA complexes, and RNA-DNA complexes. On average, the dataset contains 3 structures per sequence. Following Joshi et al. (2023), we filtered to structures with resolution ≤ 3 Å. For the RF2NA split, we curate the Traj-to-Seq training set such that RNAs in our validation and test sets are not used for training. For the sequence similarity split, we remove RNAs that have $\geq 80\%$ similarity with data points in validation/test.

As shown in Figure 6, the distribution of RNA lengths in the dataset is somewhat bimodal, with many sequences around length 10 and around length 75. In the sequence similarity split, there are 2314 distinct RNA sequences in train, 106 in validation, and 78 in test. In the RF2NA split, there are 2370 distinct RNA sequences in train, 110 in validation, and 16 in test.

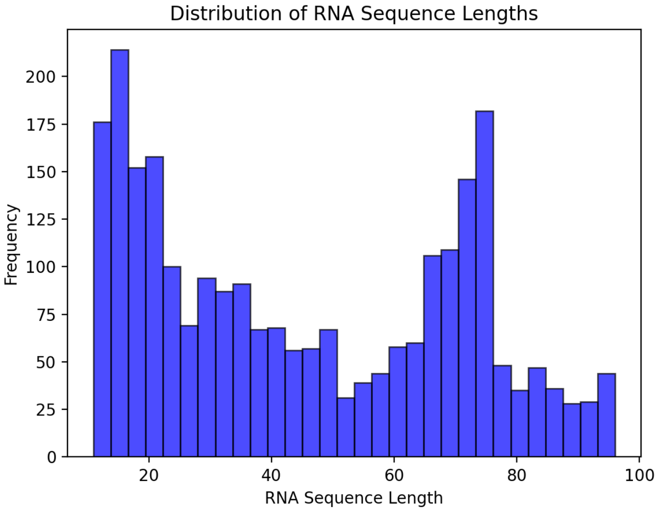

Figure 6: Distribution of RNA lengths in processed RNASolo dataset.

A.8 Training and Architecture Details

A.8.1 Traj-to-Seq

For both splits, the Traj-to-Seq encoder and decoder GVP layers use a node scalar feature dimension of 128, node vector feature dimension of 16, edge scalar feature dimension of 64, and edge vector feature dimension of 4. The model was trained for 100 epochs by an Adam optimizer with a learning rate of 0.001.

A.8.2 RNAFlow

We fine-tune RNAFlow for 100 epochs which takes one day on an NVIDIA A5000-24GB GPU, and we use the Adam optimizer with a learning rate of 0.001.

A.8.3 LSTM

For both splits, the encoder's hidden dimension is 128 which was selected by a hyperparameter sweep, and the model was trained by an Adam optimizer with a learning rate of 0.001.

A.9 Visualizations of RNA Designs

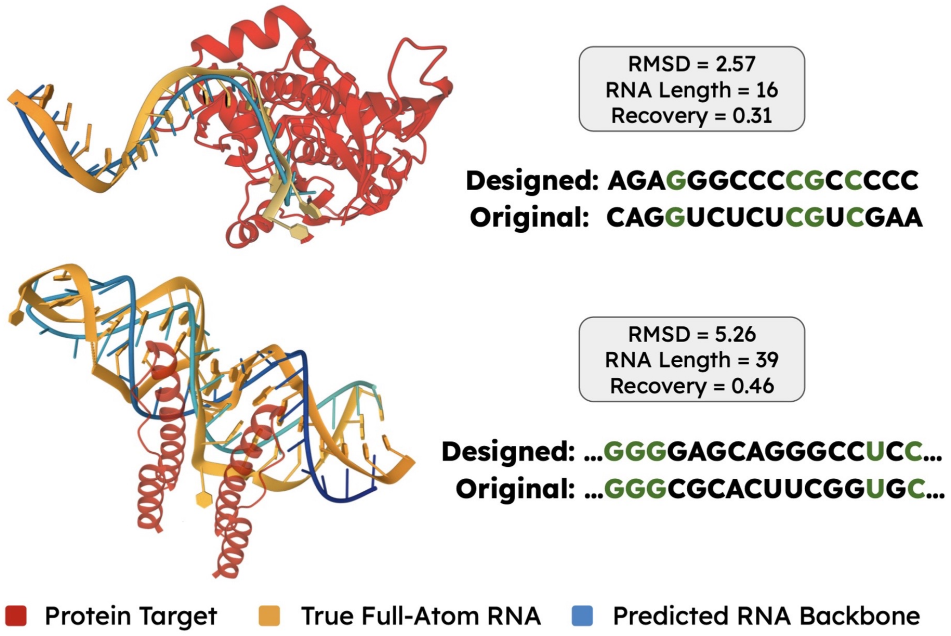

Figure 7: Top: Structure and sequence design of RNA for interaction with a viral RNA-dependent RNA polymerase (PDB ID: 4K4X). Bottom: Design of RNA for interaction with HIV-1 Rev protein (PDB ID: 4PMI).