RNAgrail: graph neural network and diffusion model for RNA 3D structure prediction

Marek Justyna[1*] mjustyna@cs.put.poznan.pl
Craig L. Zirbel[2] zirbel@bgsu.edu
Maciej Antczak[1,3] mantczak@cs.put.poznan.pl
Marta Szachniuk[1,3] mszachniuk@cs.put.poznan.pl

[1] **Institute of Computing Science, Poznan University of Technology, Poland**
[2] **Department of Mathematics and Statistics, Bowling Green State University, USA**
[3] **Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poland**

Abstract

The function of RNA is intrinsically tied to its 3D structure traditionally explored by X-ray crystallography, NMR, and Cryo-EM. However, these experiments often lack atomic-level resolution, creating the need for accurate *in silico* RNA structure prediction tools. This need has driven advances in artificial intelligence (AI), which has already revolutionized protein structure prediction. Unfortunately, similar breakthroughs in the RNA field remain limited due to sparse and unbalanced structural data. Here, we introduce RNAgrail, a novel RNA 3D structure prediction method that focuses on RNA substructures using a denoising diffusion probabilistic model (DDPM). Unlike AlphaFold 3 (AF3), considered by many to be an oracle, RNAgrail allows expert users to define base pair constraints, offering superior flexibility and precision. Our method outperformed AF3 by 12% in terms of mean RMSD and by 24% in terms of mean eRMSD. Additionally, it perfectly reproduced the canonical secondary structure outperforming Af3 by 40% in terms of interaction network fidelity (INF). RNAgrail demonstrated robustness across diverse RNA motifs and families. Despite being trained exclusively on rRNA and tRNA, it effectively generalizes to new RNA families, thus, addressing one of the major challenges in RNA 3D structure prediction. These results underscore the potential of focusing on small RNA components and integrating user-defined constraints to significantly enhance RNA 3D structure prediction, setting a new standard in RNA modeling.

Introduction

Ribonucleic acid (RNA) is fundamental to a wide range of biological processes across all living organisms. It plays a critical role in gene transcription regulation, protein synthesis, and many other cellular functions. RNA also constitutes the genetic material of some pandemic-causing viruses, including HIV and SARS-CoV-2. In medicine, this molecule serves as a valuable biomarker for cancer detection and a target in cancer therapeutics [36, 38]. Understanding the full spectrum of RNA functions is based heavily on structural studies, with a particular focus on deciphering the three-dimensional shape of this molecule.

*Corresponding author

Machine Learning for Structural Biology Workshop, NeurIPS 2024.

For decades, researchers have studied 3D structures of RNA, initially relying on experiments such as X-ray crystallography and NMR spectroscopy. However, accurate resolving of RNA structures through these techniques has proven challenging. Cryo-EM, which has recently gained popularity, also produces data with suboptimal resolutions, typically between 4-10 Å [25]. Consequently, there is a pressing need for accurate *in silico* methods to predict 3D RNA structures. In the era of artificial intelligence (AI), the latter is expected to surpass traditional approaches, as seen in the protein domain, where AlphaFold [30, 19] outperforms all competitors [4, 18]. However, the RNA field lags behind, as shown in CASP15 [13], where the top prediction groups relied on cryo-EM data, heuristics, or molecular dynamics rather than AI. Despite several deep learning (DL) models for RNA structure prediction [9], none have yet surpassed non-machine learning methods in reliability and performance. Even attempts to adapt AlphaFold-like models to RNA[31, 7, 35] have not yielded competitive results. A key limitation is the reliance on multiple sequence alignment (MSA) and structural templates. The Protein Data Bank (PDB) [8] contains over 200,000 protein structures, compared to just 7,956 RNA structures (as of July 2024). Filtering for non-redundant high-resolution (<3.5 Å) RNA structures further reduces the dataset to 2,416 entries. Additional challenges include significant data imbalances, with tRNA and rRNA making up 26% and 61% of known RNA structures, respectively [29].

The scarcity and imbalance of RNA structure data pose significant challenges for deep learning (DL) approaches. Overcoming these obstacles, we introduce a novel strategy that predicts RNA substructures, known as local descriptors [5], rather than whole RNA molecules. By exclusively training on rRNA and tRNA, and testing on other RNA families, this approach addresses one of the major challenges in RNA 3D structure prediction. It allows for the construction of smaller structural bricks using input-defined constraints, such as base pairing information, enabling generalization across diverse RNA families. Implemented in the RNAgrail system, our method employs a denoising diffusion probabilistic model (DDPM) to predict atomic positions, inspired by generative models in image synthesis that create high-quality images from textual prompts. In RNAgrail, the "prompt" is a set of user-defined constraints, including strand ids, sequences, and Watson-Crick-Franklin (WCF) interactions. Our model offers a scalable, context-aware approach. Unlike AlphaFold3, RNAgrail is open-source and offers a comprehensive toolkit for training and processing user data.

Results

Neural Network Model

RNAgrail (GRAph neural network and diffusIon modeL for RNA 3D structure prediction) comprises three major components: an RNA language model, a graph neural network (GNN), and a transformer[34] (Figure 1). The model takes a dot-bracket-like input provided by the user, defining the RNA sequence and the secondary structure. RNAgrail then predicts the 3D structure while ensuring that all specified 2D contacts are preserved.

Inspired by a flow-matching approach[23, 15] to protein design that incorporates a protein language model[22], we applied a similar concept for RNA. We used RiNALMo[27], an RNA language model pre-trained on all RNA sequences from RNAcentral[32], which contains a strong inductive bias and significantly enhances the performance of RNAgrail.

As the second component, we chose a graph neural network, currently considered a state-of-the-art approach in molecular modeling[16, 17, 15, 33]. We wanted it to consider close contacts (interactions between atoms 0–5 Å apart) and long-range interactions (up to 16 Å) that are critical for predicting RNA 3D structure. Thus, we used PAMNet[37] and adjusted it to accommodate both types of interactions, with each of its two layers responsible for handling a specific interaction type. Another modification enabled PAMNet to return a vector containing the coordinates and types of atoms, and the types of residues, instead of a single value. This was achieved by removing the global pooling operation from the final layer. The input dimensions were adjusted to include features such as input coordinates, one-hot encoded atom types, residue embeddings, and denoising time steps. PAMNet uses a global message-passing mechanism in which information about each atom (node) is passed to its neighbor atoms in a single run. The transformer encoder architecture was inspired by Invariant Point Attention (IPA), originally implemented in AlphaFold[19]. It consists of six layers with eight attention heads. Incorporating this component significantly improved the quality of the RNA structures compared to using the GNN alone. GNN efficiently feeds the transformer with structural embeddings.

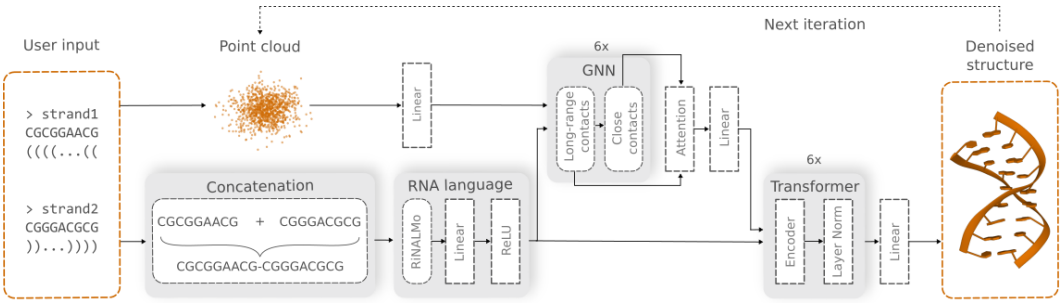

Figure 1: Components of the RNAgrail model.

Training

Due to memory constraints and the need for computational efficiency, we adopted a coarse-grained 5-bead representation of the RNA structure. Each residue was represented by two backbone atoms (P and C4′) and three nucleobase atoms (N1-C2-C4 for pyrimidines, N9-C2-C6 for purines). During data preprocessing, we transformed each full-atom structure into a coarse-grained model and constructed the corresponding graph of interatomic interactions (Figure 2a). The graph was supplemented with edges representing canonical base-pair interactions, added based on the extracted secondary structure (Figure 2b). This approach ensures that two residues are kept close in the RNA structure, even when their atoms are spatially distant in the early stages of the prediction process. To maintain consistency across all structures, we centered them around the point (0,0,0) by computing and subtracting the mean of the original coordinates. Finally, we divided the coordinate values by 10 for numerical stability. Each RNA structure in the set was gradually corrupted through a diffusion process using the standard denoising diffusion probabilistic model (DDPM) framework[28]. It involved adding Gaussian noise to the atomic coordinates in multiple steps while preserving atomic properties (such as atom names) and the connectivity graph. The neural network model of RNAgrail was trained to reverse this process by predicting a less noisy (previous) state at each step. This approach enabled the model to learn to reconstruct the original 3D structure from noisy input data.

During inference, we first create a graph representing the RNA structure to be predicted. This graph includes all molecular features (e.g., atom types, residue embeddings) and the topology derived from the input sequence and 2D interactions. The missing information — atomic coordinates — is initialized as random noise. The model then performs denoising steps on these coordinates, gradually refining them with each iteration. At the end of the denoising process, the final structure is determined.

Evaluation

The performance of the RNAgrail model was tested in two experiments. In the first, it was evaluated on 1,613 randomly selected samples from the test set. The 3D structures predicted by RNAgrail were compared with ground truth data and assessed using three common metrics: RMSD[20], eRMSD[12], and INF[26]. The model achieved a median RMSD of 7.05 Å and a median eRMSD of 1.33. For INF, which checks whether the predicted RNA structure faithfully reproduces user-provided 2D structure constraints, the mean value obtained was 0.98, and the median reached 1.0.

In the second experiment, RNAgrail was benchmarked against AlphaFold 3[1]. AF3 was chosen for comparison because, like RNAgrail, it is a generative model that allows users to define multiple strands and predict interactions between them. However, key differences exist between the two methods. Unlike AF3, RNAgrail enables users to precisely define expected interactions, such as Watson-Crick-Franklin (WCF) base pairs, between residues. Both tools were used to predict the 3D structures of 40 randomly selected RNA descriptors, which were then compared to ground truth data (see Figure 4 in Appendix). RNAgrail outperformed AF3 by 12% in terms of mean RMSD, 24% in terms of mean eRMSD, and 40% in terms of mean INF (Table 1). Our model preserved 100% of the WCF interactions specified in the input (see Figure 3). A few 3-segment RNA descriptors posed more challenges for RNAgrail, leading to significantly higher RMSD values. The difficulty arises because our model does not account for the spatial arrangement of the strands, resulting in poor structural alignment during superimposition and consequently higher RMSD.

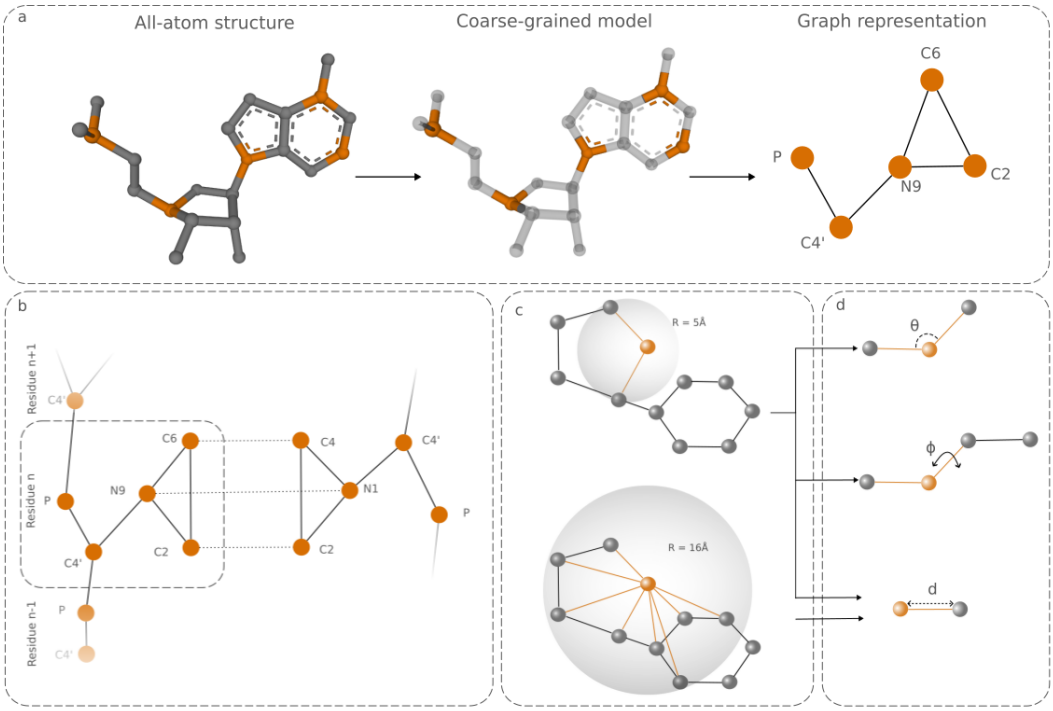

Figure 2: Graph construction. **(a)** Transformation from all-atom via coarse-grained to graph representation illustrated for purine. Orange represents coarse-grained atoms, while light gray (middle panel) denotes elements excluded from the coarse-grained model. **(b)** Each node in the graph represents an atom. Solid edges denote covalent bonds between atoms, while dotted edges represent interactions derived from user-provided 2D structure constraints. This dual-edge system enables the model to capture both the structural connectivity and the additional interactions specified in the input. **(c)** Close (top) and long-range (bottom) interactions between neighboring atoms. **(d)** Feature computation between nodes. For close interaction, we compute the distance d and two types of angles – θ between the node and the neighbors in one hop and ϕ between the node and neighbors in two hops. For a long-range interaction, we only compute the distances.

Table 1: RMSD, eRMSD, and INF computed for 40 randomly selected descriptors predicted by RNAgrail and AF3.

Method	Mean RMSD	Median RMSD	Mean eRMSD	Median eRMSD	Mean INF	Median INF
RNAgrail	6.50 Å $\pm$3.7	5.75 Å	1.31 $\pm$0.3	1.29	0.96 $\pm$0.09	1.0
AlphaFold 3	7.39 Å $\pm$4.67	6.63 Å	1.63 $\pm$0.42	1.73	0.67 $\pm$0.26	0.73

Discussion

A key strength of RNAgrail is its emphasis on local 3D RNA descriptors, enabling more flexible predictions that account for interactions between RNA residues and external elements beyond the immediate structure. This approach improves the system's ability to capture structural variability, resulting in more accurate predictions across diverse RNA environments. Notably, RNAgrail outperforms AF3 in predicting localized substructures. AF3, likely due to its training on full-length RNA structures where helices dominate, tends to over-predict helical conformations.

However, the training set poses a limitation for our model. While we filtered out single-stranded examples lacking interactions, some outliers might remain, potentially disrupting the training process. Improving the dataset by removing these problematic examples would likely lead to more reliable predictions. Additionally, expanding the dataset to include descriptors with more complex and diverse interactions could provide the model with richer training examples, enhancing its ability to generalize

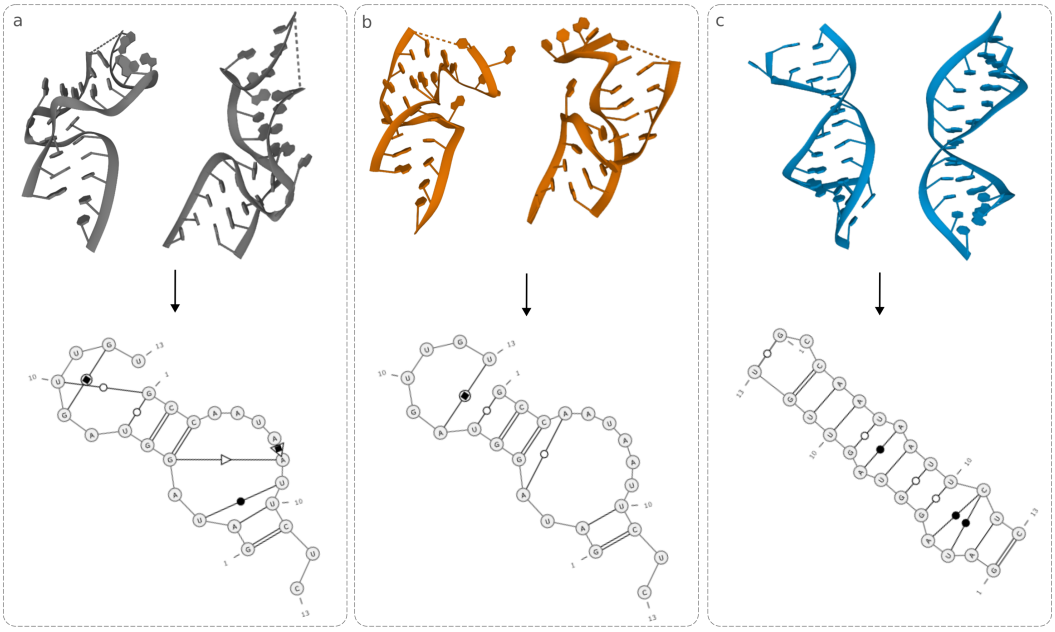

Figure 3: Example 2-segment descriptor in (a) ground-truth structure (6YW5), and predictions by (b) RNAgrail and (c) AF3.

across different RNA families. Another challenge is predicting small RNA fragments without the broader structural context of the entire molecule. Isolated segment predictions are inherently difficult, as the absence of external constraints diminishes model accuracy. Our results indicate that incorporating additional context significantly enhances the predictions, allowing the model to utilize more structural constraints. Future efforts should focus on integrating broader contextual information into the prediction process, which could improve performance and address this limitation.

Future research should address current limitations by expanding the descriptor set to include larger fragments with four or more segments, which would offer a more comprehensive context for predictions and enable RNAgrail to handle complex structures more effectively. Implementing a masking mechanism could enhance the model's accuracy by allowing it to focus on refining specific regions within a larger context. Long-term, efforts should aim at predicting complete RNA 3D structures by assembling high-accuracy predictions of smaller segments. Additionally, transitioning from a diffusion model to a flow-matching model, which predicts translation vectors instead of direct atomic coordinates, could significantly improve performance and accuracy for larger structures.

Finally, it is important to note that beyond the prediction of RNA structure, the descriptor-based approach of RNAgrail has broader implications for structural biology. The model's adaptability extends to RNA design, RNA-ligand and RNA-protein interactions, and docking studies. Its efficient prediction of smaller substructures also offers significant computational advantages, making RNAgrail accessible to a wide range of users without extensive resources. Additionally, this approach addresses critical issues in the RNA field, such as the scarcity of high-quality data and the overrepresentation of specific RNA types, like rRNA and tRNA, in structural databases.

Data and Code Availability

The training and test datasets used in this work, along with the pre-trained model weights, are available for download from Zenodo. The source code of RNAgrail is publicly available at GitHub.

Acknowledgements

This project was supported by the National Science Centre, Poland [grant 2020/39/O/ST6/01488 to MJ and MS].

References

[1] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

[2] B. Adamczyk, M. Antczak, and M. Szachniuk. RNAsolo: a repository of cleaned PDB-derived RNA 3D structures. *Bioinformatics*, 38(14):3668–3670, 2022.

[3] A. F. Agarap. Deep learning using rectified linear units (relu), 2019.

[4] M. AlQuraishi. AlphaFold at CASP13. *Bioinformatics*, 35(22):4862–4865, 2019.

[5] M. Antczak, M. Kasprzak, P. Lukasiak, and J. Blazewicz. Structural alignment of protein descriptors - a combinatorial model. *BMC Bioinformatics*, 17(1):383, 2016.

[6] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.

[7] M. Baek, R. McHugh, I. Anishchenko, H. Jiang, D. Baker, and F. DiMaio. Accurate prediction of protein-nucleic acid complexes using RoseTTAFoldNA. *Nat Methods*, 21(1):117–121, 2024.

[8] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1):235–242, 2000.

[9] C. Bernard, G. Postic, S. Ghannay, and F. Tahi. State-of-the-RNArt: benchmarking current methods for RNA 3D structure prediction. *NAR Genom Bioinform*, 6(2):lqae048, 2024.

[10] M. J. Boniecki, G. Lach, W. K. Dawson, K. Tomala, P. Lukasz, T. Soltysinski, K. M. Rother, and J. M. Bujnicki. SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. *Nucleic Acids Res*, 44(7):e63–e63, 2015.

[11] S. Bottaro, G. Bussi, G. Pinamonti, S. Reißer, W. Boomsma, and K. Lindorff-Larsen. Barnaba: software for analysis of nucleic acid structures and trajectories. *RNA*, 25(2):219–231, Feb. 2019.

[12] S. Bottaro, F. Di Palma, and G. Bussi. The role of nucleobase interactions in RNA structure and dynamics. *Nucleic Acids Res.*, 42(21):13306–13314, Dec. 2014.

[13] R. Das, R. C. Kretsch, A. J. Simpkin, T. Mulvaney, P. Pham, R. Rangan, F. Bu, R. M. Keegan, M. Topf, D. J. Rigden, Z. Miao, and E. Westhof. Assessment of three-dimensional RNA structure prediction in CASP15. *Proteins*, 91(1):1747–1770, 2023.

[14] J. Gasteiger, J. Groß, and S. Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

[15] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, A. Tong, and A. J. Bose. Sequence-augmented se(3)-flow matching for conditional protein backbone generation, 2024.

[16] I. Igashov, H. Stärk, C. Vignac, A. Schneuing, V. Satorras, P. Frossard, M. Welling, M. Bronstein, and B. Correia. Equivariant 3D-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 6:1–11, 2024.

[17] C. K. Joshi, A. R. Jamasb, R. Viñas, C. Harris, S. V. Mathis, A. Morehead, R. Anand, and P. Liò. gRNAde: Geometric Deep Learning for 3D RNA inverse design, 2024.

[18] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Applying and improving AlphaFold at CASP14. *Proteins*, 89(12):1711–1721, 2021.

[19] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583 – 589, 2021.

[20] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976.

[21] I. Kalvari, E. P. Nawrocki, N. Ontiveros-Palacios, J. Argasinska, K. Lamkiewicz, M. Marz, S. Griffiths-Jones, C. Toffano-Nioche, D. Gautheret, Z. Weinberg, E. Rivas, S. R. Eddy, R. D. Finn, A. Bateman, and A. I. Petrov. Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Res*, 49(D1):D192–D200, 2021.

[22] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. d. Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[23] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling, 2023.

[24] S. M. Rnapolis: computational platform for rna structure analysis. *Foundations of Computing and Decision Sciences*, 44:241–257, 2019.

[25] H. Ma, X. Jia, K. Zhang, and Z. Su. Cryo-EM advances in RNA structure determination. *Signal Transduct Target Ther*, 7(1):58, 2022.

[26] M. Parisien, J. A. Cruz, E. Westhof, and F. Major. New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA*, 15(10):1875–1885, Oct. 2009.

[27] R. J. Penić, T. Vlašić, R. G. Huber, Y. Wan, and M. Šikić. RiNALMo: General-purpose rna language models can generalize well on structure prediction tasks, 2024.

[28] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] B. Schneider, B. A. Sweeney, A. Bateman, J. Cerny, T. Zok, and M. Szachniuk. When will RNA get its AlphaFold moment? *Nucleic Acids Res*, 51(18):9522–9532, 2023.

[30] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[31] T. Shen, Z. Hu, Z. Peng, J. Chen, P. Xiong, L. Hong, L. Zheng, Y. Wang, I. King, S. Wang, S. Sun, and Y. Li. E2efold-3d: End-to-end deep learning method for accurate de novo rna 3d structure prediction, 2022.

[32] The RNAcentral Consortium, A. I. Petrov, S. J. E. Kay, I. Kalvari, K. L. Howe, K. A. Gray, E. A. Bruford, P. J. Kersey, G. Cochrane, R. D. Finn, A. Bateman, A. Kozomara, S. Griffiths-Jones, A. Frankish, C. W. Zwieb, B. Y. Lau, K. P. Williams, P. P. Chan, T. M. Lowe, J. J. Cannone, R. Gutell, M. A. Machnicka, J. M. Bujnicki, M. Yoshihama, N. Kenmochi, B. Chai, J. R. Cole, M. Szymanski, W. M. Karlowski, V. Wood, E. Huala, T. Z. Berardini, Y. Zhao, R. Chen, W. Zhu, M. D. Paraskevopoulou, I. S. Vlachos, A. G. Hatzigeorgiou, L. Ma, Z. Zhang, J. Puetz, P. F.

Stadler, D. McDonald, S. Basu, P. Fey, S. R. Engel, J. M. Cherry, P.-J. Volders, P. Mestdagh, J. Wower, M. B. Clark, X. C. Quek, and M. E. Dinger. RNAcentral: a comprehensive database of non-coding RNA sequences. *Nucleic Acids Res*, 45(D1):D128–D134, 2017.

[33] R. J. L. Townshend, S. Eismann, A. M. Watkins, R. Rangan, M. Karelina, R. Das, and R. O. Dror. Geometric deep learning of RNA structure. *Science*, 373(6558):1047–1051, 2021.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need, 2023.

[35] W. Wang, C. Feng, R. Han, Z. Wang, L. Ye, Z. Du, H. Wei, F. Zhang, Z. Peng, and J. Yang. trRosettaRNA: automated prediction of RNA 3D structure with transformer network. *Nat Commun*, 14(1):7266, 2023.

[36] X. Xi, T. Li, Y. Huang, J. Sun, Y. Zhu, Y. Yang, and Z. J. Lu. RNA biomarkers: Frontier of precision medicine for cancer. *Noncoding RNA*, 3(1), 2017.

[37] S. Zhang, Y. Liu, and L. Xie. A universal framework for accurate and efficient geometric deep learning of molecular systems. *Scientific Reports*, 13(1), 2023.

[38] Y. Zhu, L. Zhu, X. Wang, and H. Jin. RNA-based therapeutics: an overview and prospectus. *Cell Death Dis*, 13(7):644, 2022.

A Appendix / supplemental material

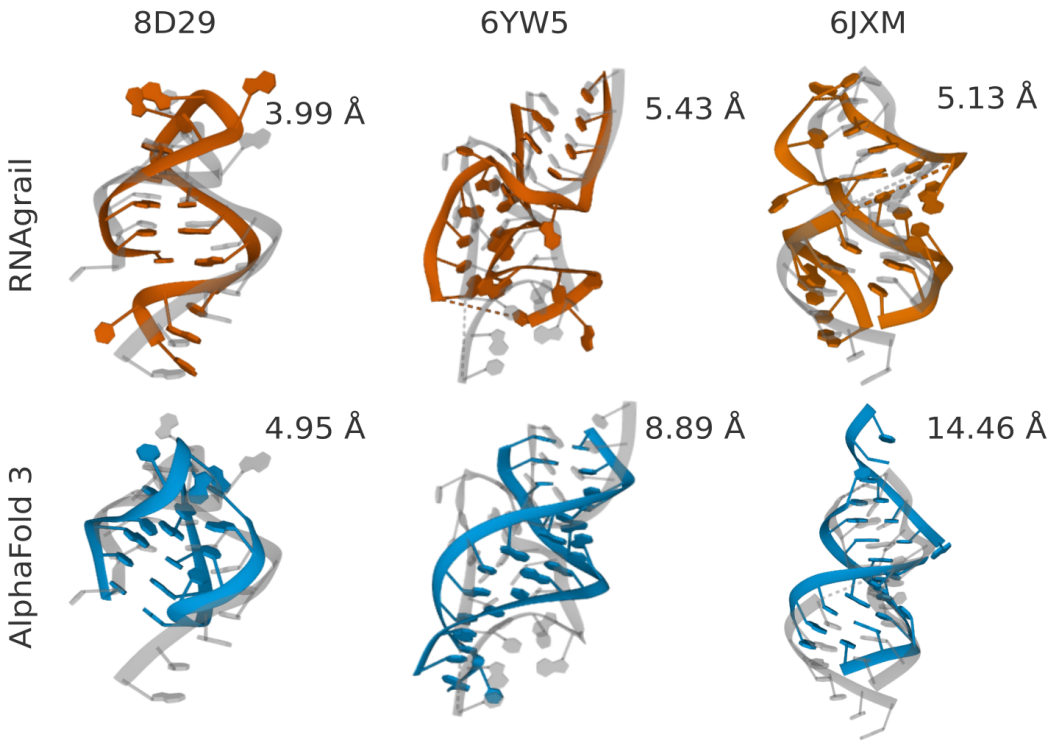

Figure 4: Example descriptors extracted from ground truth structures (8D29, 6YW5, and 6JXM) in gray aligned with their predictions by RNAgrail (orange) and AlphaFold 3 (blue).

Local 3D RNA descriptors

Local 3D RNA descriptors [5] are substructures extracted from the original RNA molecule, designed to capture the spatial neighborhood of each residue. The process of generating these descriptors

involves *N-4* iterations, with *N* corresponding to the number of residues in the RNA analyzed (Figure 5). The algorithm processes successive residues – starting with the 3rd and ending with N-2 residue – and performs 3 operations for each. In the first operation, the central atom of the current residue, in this case C5′, is selected. Next, all in-contact residues within a specified distance threshold *T* are identified and included in the descriptor; the distances are computed between the C5′ atoms. The descriptors must be large enough to capture relevant residue-residue interactions; however, it is important to note that increasing the threshold also increases the complexity of the descriptors, which in turn requires more computational power for model training. In this study, the threshold *T* was set at 16 Å. The descriptor may consist of one or more discontinous strands, referred to as segments. These segments are extended in the third operation by considering two additional residues for every central and in-contact residue in the descriptor. This ensures that the segments are sufficiently large, preventing isolated nucleotides from appearing in the descriptor. Additional residues are not required to be within the *T* threshold.

A notable feature of local 3D RNA descriptors is their recurrence in non-homologous structures, indicating that similar local environments can be found across diverse RNA molecules. This unique characteristic presents an opportunity to design models capable of generalizing to new RNA families. Depending on the size of the parent structure, the 3D arrangement of its strands, and the overall compactness, a single RNA molecule can generate multiple descriptors of varying sizes.

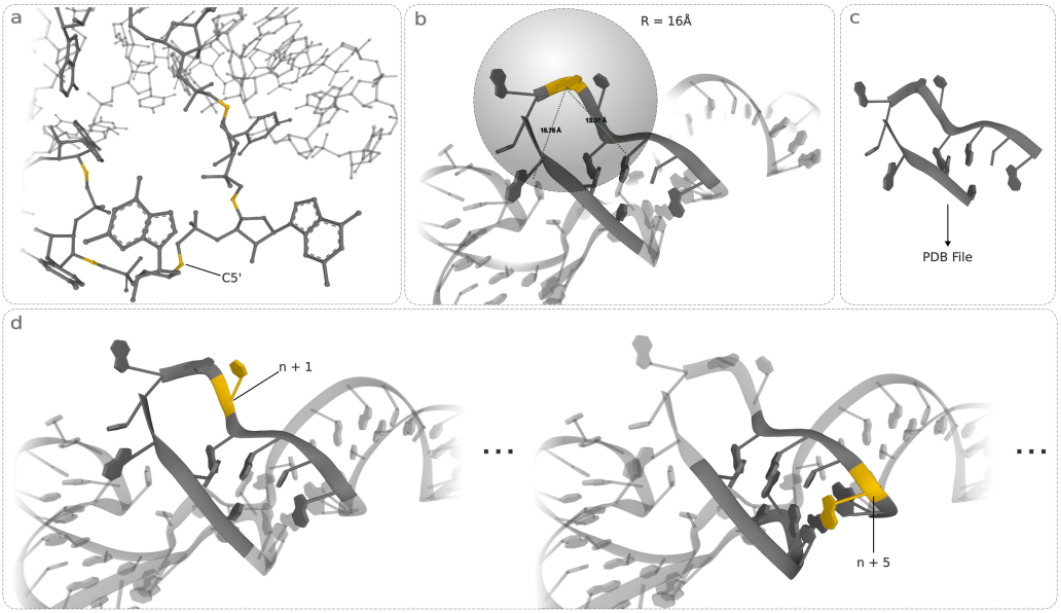

Figure 5: Extraction of local 3D RNA descriptors. (a) The C5′ atom is selected for each residue as the central reference point. (b) Distances are calculated between the central residue and all other residues; all residues within a 16 Å radius form a descriptor. (c) The descriptor is extended by adding two additional residues on each side of the both central and in-contact residue to ensure robustness; its final structure is saved in the PDB file. (d) Steps (a)-(c) are repeated iteratively for each residue in the RNA molecule considered.

Training and testing sets

The training and test sets were constructed from local 3D RNA descriptors. We began by downloading non-redundant RNA representatives from RNAsolo[2], focusing on structures with a maximum resolution of 3.5 Å. The data set, time-stamped March 2023, comprised 1,564 experimental structures. From these structures, we generated 177,629 descriptors, ranging from 1 to 18 segments, using the *descs-standalone* package [5]. We then performed data cleaning to address redundancy and repetition among similar substructures, excluding single-stranded fragments and all descriptors with more than 80% sequence identity or more than three segments. This reduced the set to 76,067 descriptors. Spatial alignment was performed using descs-standalone to eliminate structural repetitiveness, resulting

in the removal of 31,966 redundant descriptors. The final dataset comprised 3,636 one-segment descriptors, 16,849 two-segment descriptors, and 23,616 three-segment descriptors. To reduce bias, we divided the collection into training and test sets based on their Rfam family assignments[21]. Descriptors associated with rRNA or tRNA (33,518 descriptors) were assigned to the training set, while all others (10,583 structures) were placed in the test set.

Model Implementation and Training

RNAgrail was implemented in PyTorch (version 2.3.0), with graph components built using PyTorch Geometric (PyG, version 2.5.3). The connectivity graph was constructed in COO format, with nodes representing atoms in the coarse-grained model. Node features were one-hot encoded to capture atom type (carbon, phosphorus, or nitrogen), a C4′ flag, and residue type (A, G, C, or U). Additionally, nodes stored vector-based information for sequence embeddings from the RNA language model and time embeddings for denoising steps. Due to memory constraints and the need for numerical stability, we limited RNAgrail to six blocks of PAMNet. Atomic coordinates were processed through a linear layer with layer normalization[6] and sigmoid activation, encoding them as 256-dimensional vectors. The graph's nodes were connected in a stepwise process, beginning with intraresidue contacts and then expanding to interresidue interactions (see Figure 2). Interaction types were encoded as edge features. Edge attributes included information about angles (for edges representing close contacts) and distances (for edges representing long-range interactions). In the case of close contacts, features were computed based on angles, including both one-hop and two-hop neighbors (the latter similar to torsion angles), as shown in Figure 2. The angle information was processed using basis functions [14] and then passed through a trainable linear layer to generate embeddings. Moreover, new edges are dynamically added to the graph based on the defined distance thresholds: 5Å for close contacts and 16Å for long-range interactions. To enhance computational efficiency, the number of new connections is limited to the 20 nearest neighbors.

For the RNA language component, we used the giga-v1 model architecture and weights [27]. During training, we froze this model and added a linear layer with 1,280 neurons, using the ReLU activation[3] function. In cases involving multiple RNA segments, their sequences were merged into a single string to maintain sequential context and improve computational efficiency, rather than processing each segment independently. The RNA language model generated meaningful nucleotide embeddings, which were subsequently used as features in the GNN.

During training, we employed 5,000 denoising steps with a linear noise scheduler. The training process ran for 800 epochs with a batch size of 128, taking approximately 40 hours. The model was trained in a distributed manner using 8 Nvidia A100 GPUs. To enhance numerical stability and prevent exploding gradients, we applied gradient clipping with a threshold of 2. Additionally, a step learning rate scheduler was used, adjusting the initial learning rate of 0.003 by a factor of 0.9 every 30 epochs, and the Adam optimizer was used for optimization.

Model Evaluation

The RNA 3D structure predictions were evaluated using three common metrics: RMSD (Root Mean Square Deviation)[20], eRMSD (epsilon RMSD)[12], and INF (Interaction Network Fidelity)[26]. RMSD is a general distance measure used to assess the structural accuracy of biological molecules, whereas eRMSD (a distance measure) and INF (a similarity measure) are specifically tailored for RNA structure evaluation.

RMSD was calculated using the open-source PyMOL tool (version 3.0.0). Since RMSD operates on full-atom structures, the predicted coarse-grained models had to be converted into full-atom representations, which was done using SimRNA tools[10]. RMSD was then calculated across the entire structure using formula 1, where N is the total number of atoms, r^{pred} represents the predicted atomic coordinates, and r^{true} denotes the corresponding coordinates in the ground truth structure.

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{r}_i^{\text{pred}} - \mathbf{r}_i^{\text{true}}\right)^2} \tag{1}$$

eRMSD, which also requires a full-atom structure, was calculated using Barnaba[11]. This metric evaluates the relative arrangement of nucleobases and their distances, making it particularly sensitive to base pairing. eRMSD is continuous and symmetric, with typical values interpreted as follows: values below 0.7 indicate native-like stems and loops, values between 0.7 and 1.3 correspond to structures with native stems but non-native loops, and values greater than 1.3 are indicative of significant deviations from the native state. Its computation follows formula 2, where $\tilde{r}_{ij}$ are the rescaled position vectors for all pairs of bases, and *G* is a nonlinear function of $\tilde{r}$.

$$\epsilon RMSD = \sqrt{\frac{1}{N}\sum_{i,j}\left(G(\tilde{\mathbf{r}}_{ij}^{pred}) - G(\tilde{\mathbf{r}}_{ij}^{true})\right)^2} \tag{2}$$

INF evaluates discrepancies between the predicted and reference secondary structures, providing a normalized score between 0 and 1. To compute it, we first extracted base pair information from both the predicted and reference structures using the RNApolis package[24]. INF is then calculated using formula 3[26]. True positives (TP) represent interactions that are present in both the predicted and ground truth structures. False positives (FP) refer to interactions predicted by the model but absent in the ground truth, while false negatives (FN) are interactions present in the ground truth but missing from the predicted structure.

$$INF = \sqrt{\frac{|TP|}{|TP| + |FP|} \times \frac{|TP|}{|TP| + |FN|}} \tag{3}$$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we highlight the novelty of using local RNA descriptors for 3D structure prediction, which is supported by the experimental results. The model's ability to generalize across RNA families, despite being trained exclusively on rRNA and tRNA, is a major contribution, and this is discussed thoroughly in the results and conclusion sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Discussion section addresses limitations such as problematic dataset examples and the difficulty of predicting small RNA fragments without broader context.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: No theoretical results included.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We described all model details, such as components used, size of layers, hyperparameters. Moreover, we deliver code, scripts, configs, and dataset that will allow other researchers to fully reproduce our work.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code, model weights and dataset are fully open access. Link for download are in section "Data and Code Availability".

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the hyperparameters, optimizer type, and other settings. Additionally, we include a dataset with the exact train-test split.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide Standard Deviation for mean values presented in the table.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the appendix we provide the detailed information about infrastructure we used for training.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper fully conforms with the NeurIPS Code of Ethics. No ethical concerns or violations are present in the methods, data usage, or findings of this work

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work presented in this paper focuses on technical advancements in RNA 3D structure prediction and does not have direct societal implications. As such, no potential positive or negative societal impacts are anticipated.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no risk in using the released data and model.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets, including code, data, and models used in the paper, are credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we provide the code and data under CC-BY 4.0 license. All of that is explicitly mentioned in the repositories.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.