Improving RNA 3D Structure Prediction via Language Model-Augmented AlphaFold 3

Shuxian Zou[1*], **Jiayou Zhang**[1], **Bingkang Zhao**[4], **Hui Li**[1], **Eric P. Xing**[1,2,3], **Le Song**[1,3*]

[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Carnegie Mellon University
[3]GenBio AI
[4]Zhejiang University

Abstract

Predicting RNA 3D structure from sequence remains challenging due to the structural flexibility of RNA molecules and the scarcity of experimentally resolved structures. We ask how self-supervised RNA language models (LMs), trained on millions of RNA sequences, can best enhance AlphaFold 3 (AF3) for RNA structure prediction. Using an open-source AF3 reproduction, we run controlled experiments that fix data and hyperparameters while varying fusion position and method. We find large performance variations between fusion strategies, and without Multiple Sequence Alignment (MSA), they are generally not effective. When incorporating MSA, the most effective approach is additive fusion applied at the late stage of the conditional network, refining AF3's single representations with RNA LM embeddings. On RecentPDB-RNA (47 newly released targets), our best model achieves an average TM-score of 0.438 and a success rate of 30% (TM-score $\geq$ 0.6), significantly outperforming all baseline models. On 11 CASP16-RNA targets, it matches the best automated system trRosettaRNA. These results show that properly fused RNA LM features substantially advance RNA 3D structure prediction.

1 Introduction

Accurately predicting RNA 3D structures from primary sequences is essential for understanding RNA function and for enabling RNA-based therapeutics, such as mRNA vaccines, ASOs, and aptamers [1, 2]. While AlphaFold has transformed protein structure prediction with near-experimental accuracy [3, 4], RNA modeling remains more difficult due to the scarcity of experimentally solved structures (only a few thousand in the Protein Data Bank (PDB) [5]). This makes RNA 3D structure prediction a small-data, high-dimensional machine learning problem. Recent CASP16 results show that all top-performing RNA prediction groups are human expert predictors [6].

Computational approaches to RNA structure modeling have been developed for two decades, evolving from energy-based and template-driven methods to deep learning–based strategies inspired by AF (see Appendix Section A.5 for related work). Recent methods such as trRosettaRNA [7], RhoFold+ [8], and NuFold [9] adapt AF's design for RNA, while AF3 extends predictions to multiple biomolecule types, including RNA. Benchmarking studies show AF3 is competitive and often outperforms earlier methods for RNA 3D prediction [10]. Despite this progress, RNA-specific modeling remains underdeveloped compared to proteins.

[*]Corresponding authors: shuxian.zou@mbzuai.ac.ae, le.song@mbzuai.ac.ae

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Machine Learning for Structural Biology (MLSB).

In parallel with advances in RNA structure modeling, progress in RNA sequence modeling has driven the development of increasingly powerful RNA language models. Through self-supervised learning on tens of millions of RNA sequences, RNA LMs capture evolutionary and structural information, achieving impressive performance across diverse RNA function and structure prediction tasks [8, 11, 12]. A natural question is: can representations learned from massive RNA sequences by RNA LMs be leveraged to enhance AF3's performance on RNA 3D structure prediction? The motivation is that, although AF3 is jointly trained on protein, RNA, and DNA structural data, proteins dominate the training set. As a result, RNA-specific representations may be underdeveloped and could benefit from the richer features provided by RNA LMs.

To answer this question, we are facing a multimodal fusion problem. The technical challenges are: 1) representations from RNA LM and AF3 are not in the same feature space; and 2) it is difficult to build models that exploit supplementary and not only complementary information [13]. The high complexity of AF3's architecture further complicates the problem. We locate five positions in AF3 that can be good candidates for feature fusion. For each position, we can use multiple fusion methods. It remains unclear where and how to best incorporate RNA LM representations into AF3.

To investigate this, we design a series of controlled experiments on fusing RNA LM's representation into AF3 [2], keeping the training data and hyperparameters fixed while varying only the fusion positions and methods. We evaluate these models on RecentPDB-RNA, a curated test set comprising 47 RNA targets from the PDB, each released after the training data temporal cutoff and filtered to ensure a maximum sequence similarity of 0.8 to the training set. We find large performance variations between fusion strategies, and without MSA, they are generally not effective. When incorporating MSA, the most effective approach is additive fusion applied at the late stage of the conditional network, refining AF3's single representations with RNA LM embeddings. On 47 RecentPDB-RNA targets, our best fusion model achieves a state-of-the-art in RNA 3D structure prediction, with an average TM-score of 0.438 and a success rate of 30%. On 11 CASP16-RNA targets released in 2025, it surpasses most of the baselines, reaching the performance of the best automated method in the CASP16 competition. These results demonstrate that the representations learned from RNA LMs are informative for RNA 3D structure prediction when incorporated using the right strategy.

2 Methodology

2.1 Preliminary: AF3-style architecture

AF3 is a diffusion-based generative model that, conditioned on primary sequences and optional inputs such as MSAs, predicts all-atom 3D coordinates of biomolecules. As shown in Figure 1, the overall architecture of AF3 consists of three main components: **1) Input Embedder**: A small Transformer embeds the tokens in the primary sequence of length N_{token} into *single* representations $\mathbf{s}^{\text{inputs}} \in \mathbb{R}^{(N_{\text{token}}, c_{s\_\text{inputs}})}$, and produces an initial *single* representation $\mathbf{s}^{\text{init}} \in \mathbb{R}^{(N_{\text{token}}, c_s)}$ by a linear projection and a *pair* representation $\mathbf{z}^{\text{init}} \in \mathbb{R}^{(N_{\text{token}}, N_{\text{token}}, c_z)}$ by the outer sum of the single representation as inputs for the following Pairformer blocks. **2) Pairformer**: A large trunk jointly updates the single representation $\mathbf{s}$ and pair representation $\mathbf{z}$ using attention with geometric interactions. The trunk stacks 48 blocks that exchange information between $\mathbf{s}$ and $\mathbf{z}$ and injects structural priors (e.g. triangular inequality), producing conditioning features $\mathbf{s}^{\text{trunk}} \in \mathbb{R}^{(N_{\text{token}}, c_s)}$ and $\mathbf{z}^{\text{trunk}} \in \mathbb{R}^{(N_{\text{token}}, N_{\text{token}}, c_z)}$ tailored for coordinate generation. **3) Diffusion Module**: A conditional, non-equivariant generative model operates on the point cloud of atoms. Conditioned on the trunk outputs $\mathbf{s}^{\text{trunk}}$ and $\mathbf{z}^{\text{trunk}}$ and the Input Embedder output $\mathbf{s}^{\text{inputs}}$, a denoising diffusion module iteratively refines noisy atomic coordinates to a final structure (distribution). It adopts a two-level design, alternating between atom-level, token-level, and back to atom-level operations to produce atomic coordinate predictions.

2.2 Our method

We extend the AF3-style architecture to investigate how the LM representations learned from millions of RNA sequences can enhance RNA 3D structure prediction. To this end, we incorporate RNA LM embeddings into an AF3-like architecture, systematically exploring different fusion positions and methods. The basic idea is to refine AF3's single or pair representations using RNA LM embeddings. The detailed algorithms are in Appendix Section A.4.

[2] Due to the prohibited use policy of AF3, we used Protenix [14] in our experiments.

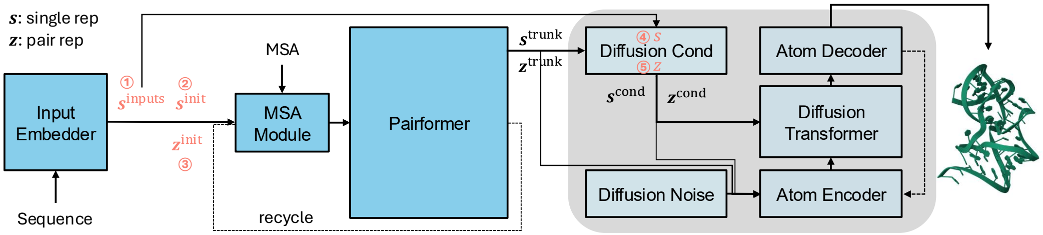

Figure 1: **Overview of the AF3-style model architecture**, showing the information flow from sequence to single and pair representation, and finally to atomic structure. The grey block denotes the Diffusion Module, and the salmon-colored regions indicate candidate positions for feature fusion.

Feature extraction from RNA LM Given an RNA sequence of N_{token} nucleotides, let $\mathbf{s}^{\text{rnalm}} \in \mathbb{R}^{(N_{\text{token}}, c_{\text{rnalm}})}$ denote the final-layer hidden states from the RNA LM, where c_{rnalm} is the embedding dimension. We lift these single-token embeddings to pair space by forming $\mathbf{z}^{\text{rnalm}} \in \mathbb{R}^{(N_{\text{token}}, N_{\text{token}}, c_z)}$ by projecting $\mathbf{s}^{\text{rnalm}}$ twice to c_z channels and computing an outer sum between the two projected matrices, where c_z is the pair representation embedding dimension in AF3. In specific,

$$\mathbf{z}_{ij}^{\text{rnalm}} = \mathbf{s}_i^{\text{rnalm}} W_1 + \mathbf{s}_j^{\text{rnalm}} W_2$$

where $W_1, W_2 \in \mathbb{R}^{(c_{\text{rnalm}}, c_z)}$ are trainable parameters, and i, j denote positions in the sequence. The outer-sum construction yields a symmetric pair representation, i.e., $\mathbf{z}_{ij}^{\text{rnalm}} = \mathbf{z}_{ji}^{\text{rnalm}}$, when the two projections are tied ($W_1 = W_2$).

Fusion positions Examining the AF3 architecture, we identify five candidate positions for feature fusion: 1) The input single representation $\underline{\mathbf{s}^{\text{inputs}}}$; 2) The initial single representation $\underline{\mathbf{s}^{\text{init}}}$; 3) The initial pair representation $\underline{\mathbf{z}^{\text{init}}}$; 4) The single conditioning representation $\underline{\mathbf{s}}$ in diffusion module; 5) The pair conditioning representation $\underline{\mathbf{z}}$ in diffusion module. As shown in Figure 1, among the five candidate fusion positions, the first three lie upstream of the Pairformer; features fused at these locations are subsequently processed by the Pairformer. The remaining two positions are downstream of the Pairformer; features injected there bypass it and are used directly to condition the Diffusion Module. To avoid redundancy, we fuse at a single position per model variant rather than at multiple positions simultaneously.

Fusion methods We consider three commonly used fusion methods: 1) **Add Fusion**: add RNA LM embedding (or its outer-sum projection) to the target representation; 2) **Concat Fusion**: concatenate RNA LM embedding with the target representation along the feature dimension; and 3) **Cross-attention Fusion**: use the target representation as query and RNA LM embedding as key and value in a multi-head cross-attention mechanism [15]. Note that Concat Fusion changes the dimensionality of the target representation, whereas Add Fusion and Cross-attention Fusion preserve it.

Fusion strategies: combinations of fusion positions and methods For single representation $\mathbf{s}_{af} \in \mathbb{R}^{(N_{\text{token}}, c)}$, the updated single representation after feature fusion is

$$\mathbf{s}_{af} = \begin{cases} \sigma\big(\mathbf{s}^{\text{rnalm}} W_2\big) \odot \mathbf{s}_{af} + \mathbf{s}^{\text{rnalm}} W_1, & \text{if Add Fusion,} \\ [\mathbf{s}_{af}; \mathbf{s}^{\text{rnalm}}], & \text{if Concat Fusion,} \\ \text{CrossAttention}(q = \mathbf{s}_{af},\ kv = \mathbf{s}^{\text{rnalm}}), & \text{if Cross-attention Fusion.} \end{cases} \tag{1}$$

where $W_1, W_2 \in \mathbb{R}^{(c_{\text{rnalm}}, c)}$, $\sigma(.)$ is a sigmoid function, $\odot$ is element-wise multiplication. When the fusion happens in the Diffusion Conditioning Module, we do not use the gate function for Add Fusion, so it becomes: $\mathbf{s}_{af} = \mathbf{s}_{af} + \mathbf{s}^{\text{rnalm}} W_1$.

For pair representation $\mathbf{z}_{af} \in \mathbb{R}^{(N_{\text{token}}, N_{\text{token}}, c_z)}$, the updated pair representation after feature fusion is

$$\mathbf{z}_{af} = \begin{cases} \mathbf{z}_{af} + \mathbf{z}^{\text{rnalm}}, & \text{if Add Fusion,} \\ [\mathbf{z}_{af}; \mathbf{z}^{\text{rnalm}}], & \text{if Concat Fusion.} \end{cases} \tag{2}$$

Table 1: **RNA 3D structure prediction performance of RNA LM fusion strategies in an AF3-like architecture (Protenix) on RecentPDB-RNA.** Bold indicates the best result and underline indicates the second best results.

	Fusion position	Fusion method	Use MSA	TM-score ↑	#Success ↑
Original Protenix [14]			✘	0.325	3
Finetuned Protenix	none	none	✘	0.415	11
		none	✔	0.399	11
RLM-aug Protenix	inputs $\mathbf{s}^{\text{inputs}}$	add	✘	0.382	11
		cross attention	✘	0.376	8
	init single rep $\mathbf{s}^{\text{init}}$	add	✘	0.419	11
		add	✔	0.409	10
		concat	✘	0.406	10
		cross attention	✘	0.412	11
		cross attention	✔	0.409	11
	init pair rep $\mathbf{z}^{\text{init}}$	add	✘	0.370	9
	single conditioning $\mathbf{s}$	add	✘	0.397	12
		add	✔	**0.438**	**14**
		concat	✘	0.360	10
	pair conditioning $\mathbf{z}$	add	✘	0.402	10
		concat	✘	0.393	10

3 Experiments

Effect of fusion strategies Table 1 summarizes the performance of models trained with different fusion strategies. We observe substantial variation across strategies, and without MSA, most approaches provide little to no benefit. When MSA is incorporated, the RLM-aug Protenix (single conditioning, add) stands out as the most effective: it achieves a TM-score of 0.438, a 10% relative improvement over Finetuned Protenix with MSA, and raises the success rate from 23% to 30%. It also significantly outperforms the other RLM-aug Protenix variants (one-sided paired t-test, $\alpha = 0.05$).

Interaction between RNA LM and MSA We conducted a 2×2 ablation study examining the effects of MSA and RNA LM embeddings (fused through single conditioning, add) during training. As shown in Figure 2, finetuning Protenix with either LM or MSA alone does not yield noticeable improvements, whereas combining both leads to a significant increase in TM-score, which indicates a complementary relationship between RNA LM and MSA. Furthermore, we found that integrating the RNA LM embedding enhances the model's capacity to exploit evolutionary information from MSAs during inference (Appendix Figure 1), suggesting that the RNA LM and MSA act synergistically.

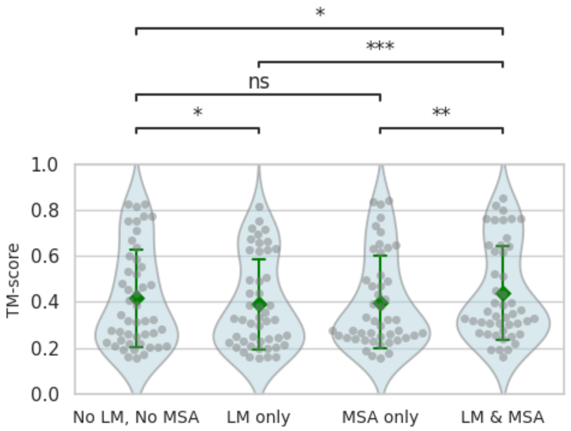

Figure 2: **Effect of RNA LM and MSA in training on RecentPDB-RNA.** Gray dots represent individual TM-scores, and green lines indicate ±1 standard deviations from the mean. All four models are variants of Protenix finetuned on the same dataset, differing only in whether RNA LM or MSA is used during training. For models incorporating the RNA LM, the (single conditioning, add) fusion strategy is used. Two-sided paired t-tests were performed between models (ns: $p > 0.05$, *: $0.01 < p \leq 0.05$, **: $0.001 < p \leq 0.01$, ***: $p \leq 0.001$).

Table 2: **TM-score on RecentPDB-RNA by maximum sequence similarity to the training set.**

	All	0.7-0.8	0.6-0.7	0.5-0.6	<0.5
[1] Protenix	0.325	0.266	0.458	0.333	0.225
[2] Finetuned Protenix w/ MSA	0.399	0.539	0.592	0.356	0.242
[3] RLM-aug Protenix w/ MSA (single cond., add)	0.438	0.609	0.619	0.377	0.303
Contribution of Data & MSA ([2]-[1])/([3]-[1])	**66%**	**80%**	**83%**	**52%**	22%
Contribution of RNA LM ([3]-[2])/([3]-[1])	34%	20%	17%	**48%**	**78%**

Table 3: **RNA 3D structure prediction results on RecentPDB-RNA and CASP16-RNA.** For the Vfold Pipeline, 14 targets on RecentPBD-RNA and 5 targets on CASP16-RNA failed to return predicted 3D structures. TM-score averages were taken on those with predicted structures. The trRosettaRNA server generates five structural decoys but outputs only the one with minimal free energy. ** denotes RLM-aug Protenix w/ MSA (single conditioning, add) is significantly better than the corresponding baseline model (one-sided pair t-test on RecentPDB-RNA, one-sided Wilcoxon signed-rank test on CASP16-RNA, p-value < 0.05) while "ns" denotes not significant.

	RecentPDB-RNA (47)		**CASP16-RNA (11)**	
	TM-score ↑	Success rate ↑	TM-score ↑	Success rate ↑
Vfold (human expert) from CASP16 website			0.486	36%
Vfold Pipeline* [16]	0.279	0%	0.289	0%
NuFold [9]	0.282 **	2%	0.243 **	0%
RhoFold+ [8]	0.309 **	9%	0.277 **	0%
DRfold2 [17]	0.316 **	9%	0.313 **	18%
trRosettaRNA* [7]	0.332 **	11%	0.412 (ns)	**27%**
AlphaFold 3 [4]	0.358 **	9%	0.371 (ns)	9%
Protenix [14]	0.325 **	6%	0.340 **	18%
Finetuned Protenix w/ MSA	0.399 **	23%	0.421 (ns)	**27%**
RLM-aug Protenix w/ MSA (single conditioning, add)	**0.438**	**30%**	**0.422**	**27%**

Contribution of RNA LM To quantify the contribution of the RNA LM in RLM-aug Protenix w/ MSA (single conditioning, add), we grouped the RecentPDB-RNA targets by their maximum sequence identity to the training set. As shown in Table 2, the improvement from Protenix to Finetuned Protenix w/ MSA reflects the effect of data and MSA, while the improvement from Finetuned Protenix w/ MSA to RLM-aug Protenix w/ MSA reflects the added benefit of the RNA LM. Overall, data + MSA contribute 66% of the total gain and the RNA LM contributes 34%. Stratifying by sequence similarity reveals a clear pattern: 1) for high-similarity targets, gains come mostly from data + MSA; 2) for low-similarity targets, the RNA LM contribution becomes dominant.

Comparison with existing models Table 3 compares our best fusion model with existing RNA 3D structure predictors. On 47 targets from RecentPDB-RNA, our method achieves the top performance, with a TM-score of 0.438 and a 30% success rate, significantly outperforming all baselines (one-sided paired t-test). On 11 targets from CASP16-RNA, it reaches a TM-score of 0.422 and a 27% success rate, comparable to trRosettaRNA (0.412, 27%) and AF3 (0.371, 9%) while significantly exceeding NuFold, RhoFold+, and DRfold2. Although human expert modeling (Vfold) remains strongest (TM-score 0.486, 36%), our RNA LM–augmented Protenix meaningfully narrows the gap. Additionally, the model maintains top performance on long RNAs (nt > 400) (Appendix Figure 2). Example prediction visualizations are shown in Appendix Figure 3.

4 Conclusion

In this work, we systematically explored strategies for integrating RNA LM representations into an AF3-like architecture to improve RNA 3D structure prediction. We found that the most effective approach is to add LM representations to the single representation within the Diffusion Conditioning Module and finetune Protenix with MSAs, which enables the model to better exploit evolutionary information and substantially improves performance on low-similarity sequences and long RNAs.

References

[1] Yiran Zhu, Liyuan Zhu, Xian Wang, and Hongchuan Jin. Rna-based therapeutics: an overview and prospectus. *Cell Death & Disease*, 13(7):644, 2022.

[2] John R Androsavich. Frameworks for transformational breakthroughs in rna-based medicines. *Nature Reviews Drug Discovery*, 23(6):421–444, 2024.

[3] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[4] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

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

[6] Rachael C Kretsch, Alissa M Hummer, Shujun He, Rongqing Yuan, Jing Zhang, Thomas Karagianes, Qian Cong, Andriy Kryshtafovych, and Rhiju Das. Assessment of nucleic acid structure prediction in casp16. *BioRxiv*, pages 2025–05, 2025.

[7] Wenkai Wang, Chenjie Feng, Renmin Han, Ziyi Wang, Lisha Ye, Zongyang Du, Hong Wei, Fa Zhang, Zhenling Peng, and Jianyi Yang. trrosettarna: automated prediction of rna 3d structure with transformer network. *Nature Communications*, 14(1):7266, 2023.

[8] Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, Yixuan Wang, Liang Hong, Jin Xiao, et al. Accurate rna 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, pages 1–12, 2024.

[9] Yuki Kagaya, Zicong Zhang, Nabil Ibtehaz, Xiao Wang, Tsukasa Nakamura, Pranav Deep Punuru, and Daisuke Kihara. Nufold: end-to-end approach for rna tertiary structure prediction with flexible nucleobase center representation. *Nature Communications*, 16(1):881, 2025.

[10] Clément Bernard, Guillaume Postic, Sahar Ghannay, and Fariza Tahi. Has alphafold3 achieved success for rna? *Biological Crystallography*, 81(2), 2025.

[11] Rafael Josip Penić, Tin Vlašić, Roland G Huber, Yue Wan, and Mile Šikić. Rinalmo: General-purpose rna language models can generalize well on structure prediction tasks. *Nature Communications*, 16(1):5671, 2025.

[12] Shuxian Zou, Tianhua Tao, Sazan Mahbub, Caleb N Ellington, Robin Algayres, Dian Li, Yonghao Zhuang, Hongyi Wang, Le Song, and Eric P Xing. A large-scale foundation model for rna function and structure prediction. *BioRxiv*, pages 2024–11, 2024.

[13] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.

[14] ByteDance AML AI4Science Team, Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pages 2025–01, 2025.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] Jun Li, Sicheng Zhang, Dong Zhang, and Shi-Jie Chen. Vfold-pipeline: a web server for rna 3d structure prediction from sequences. *Bioinformatics*, 38(16):4042–4043, 2022.

[17] Yang Li, Chenjie Feng, Xi Zhang, and Yang Zhang. Ab initio rna structure prediction with composite language model and denoised end-to-end learning. *BioRxiv*, pages 2025–03, 2025.

[18] Marcell Szikszai, Marcin Magnus, Siddhant Sanghi, Sachin Kadyan, Nazim Bouatta, and Elena Rivas. Rna3db: A structurally-dissimilar dataset split for training and benchmarking deep learning models for rna structure prediction. *Journal of Molecular Biology*, 436(17):168552, 2024.

[19] Shujun He, CASP16 organizers, CASP16 RNA experimentalists, RNA-Puzzles consortium, VFOLD team, Rachael Kretsch, Alissa Hummer, Andrew Favor, Walter Reade, Maggie Demkin, Rhiju Das, et al. Stanford rna 3d folding. `https://kaggle.com/competitions/stanford-rna-3d-folding`, 2025. Kaggle.

[20] Chengxin Zhang, Yang Zhang, and Anna Marie Pyle. rmsa: a sequence search and alignment algorithm to improve rna structure modeling. *Journal of Molecular Biology*, 435(14):167904, 2023.

[21] Caleb N Ellington, Dian Li, Shuxian Zou, Elijah Cole, Ning Sun, Sohan Addagudi, Le Song, and Eric P Xing. Rapid and reproducible multimodal biological foundation model development with aido. modelgenerator. *BioRxiv*, pages 2025–06, 2025.

[22] Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. Us-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature Methods*, 19(9):1109–1115, 2022.

[23] Clément Bernard, Guillaume Postic, Sahar Ghannay, and Fariza Tahi. State-of-the-rnart: benchmarking current methods for rna 3d structure prediction. *NAR Genomics and Bioinformatics*, 6(2):lqae048, 2024.

[24] Michal J Boniecki, Grzegorz Lach, Wayne K Dawson, Konrad Tomala, Pawel Lukasz, Tomasz Soltysinski, Kristian M Rother, and Janusz M Bujnicki. Simrna: a coarse-grained method for rna folding simulations and 3d structure prediction. *Nucleic Acids Research*, 44(7):e63–e63, 2016.

[25] Dong Zhang, Jun Li, and Shi-Jie Chen. Isrna1: de novo prediction and blind screening of rna 3d structures. *Journal of Chemical Theory and Computation*, 17(3):1842–1857, 2021.

[26] Jun Li and Shi-Jie Chen. Rnajp: enhanced rna 3d structure predictions with non-canonical interactions and global topology sampling. *Nucleic Acids Research*, 51(7):3341–3356, 2023.

[27] Song Cao and Shi-Jie Chen. Physics-based de novo prediction of rna 3d structures. *The Journal of Physical Chemistry B*, 115(14):4216–4226, 2011.

[28] Mariusz Popenda, Marta Szachniuk, Maciej Antczak, Katarzyna J Purzycka, Piotr Lukasiak, Natalia Bartol, Jacek Blazewicz, and Ryszard W Adamiak. Automated 3d structure composition for large rnas. *Nucleic Acids Research*, 40(14):e112–e112, 2012.

[29] Robin Pearce, Gilbert S Omenn, and Yang Zhang. De novo rna tertiary structure prediction at atomic resolution using geometric potentials from deep learning. *BioRxiv*, pages 2022–05, 2022.

[30] Yang Li, Chengxin Zhang, Chenjie Feng, Robin Pearce, P Lydia Freddolino, and Yang Zhang. Integrating end-to-end learning with deep geometrical potentials for ab initio rna structure prediction. *Nature Communications*, 14(1):5745, 2023.

[31] Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature Methods*, 21(1):117–121, 2024.

[32] Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

[33] Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, et al. Boltz-1: Democratizing biomolecular interaction modeling. *BioRxiv*, pages 2024–11, 2024.

[34] Chai Discovery team, Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhonikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life. *BioRxiv*, pages 2024–10, 2024.

[35] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[36] Xiaomin Fang, Fan Wang, Lihang Liu, Jingzhou He, Dayong Lin, Yingfei Xiang, Kunrui Zhu, Xiaonan Zhang, Hua Wu, Hui Li, et al. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nature Machine Intelligence*, 5(10):1087–1096, 2023.

[37] Yong He, Pan Fang, Yongtao Shan, Yuanfei Pan, Yanhong Wei, Yichang Chen, Yihao Chen, Yi Liu, Zhenyu Zeng, Zhan Zhou, et al. Generalized biological foundation model with unified nucleic acid and protein language. *Nature Machine Intelligence*, pages 1–12, 2025.

A Appendix

A.1 Data

A.1.1 Training data

For training data, we use RNA3DB, a curated collection of structured RNAs derived from Protein Data Bank [18]. The following chains were excluded: 1) shorter than 32 residues; 2) with structural resolution higher than 9Å; 3) a single nucleotide makes up more than 80% of residues; and 4) more than 30% of the residues are “unknown”. The remaining RNA chains were clustered at 99% sequence identity. RNA3DB preserves all the chains in the cluster since they are associated with different experimentally determined structures. While the chain is the same, it is possible that the presence of different interacting partners in the actual crystal structures may result in different structural conformations. RNA3DB preserves the full extent of the structural diversity present in PDB. We use the 2024-12-04 release of RNA3DB, comprising 12,892 samples spanning 2,687 unique RNA chains with approximately 5 structures per chain. The average sequence length for the unique sequences is 742, and 30% of the data have a sequence length over 384. For multiple sequence alignments (MSAs), we retrieve them from `MSA_v2` data from Stanford RNA 3D Folding [19] searched by rMSA [20], which covers 39% of the training sequences.

A.1.2 Evaluation data

RecentPDB-RNA evaluation set We collected RNA structures from the PDB released between December 4, 2024 and April 28, 2025, selecting entries with a resolution better than 4Å and RNA sequence lengths between 30 and 1,000 nucleotides. Each complex contains no more than 20 RNA chains, yielding 67 unique sequences. After filtering out samples with sequence similarity ≥80% to the training set, 54 sequences remained. We then cross-checked these PDB entries against the latest RNA3DB release (2025-10-01-incremental-release) and successfully retrieved 150 corresponding structures for 47 of the sequences, with each sequence corresponding to an average of 3 structures. The average sequence length is 242, with a minimum length of 36 and a maximum length of 814. We search MSA for these targets using rMSA, an automated pipeline that searches and aligns homologs from RNAcentral, Rfam, and nt databases (see Appendix Table 2 for database versions) for a target RNA. The distributions of sequence length, sequence similarity between the test and training sets, and the number of effective sequences (Neff) in the MSAs are shown in Appendix Table 1.

CASP16-RNA evaluation set We collected the CASP 16 RNA targets with experimental structures released in PDB in 2025, containing 11 targets in total. The target ids are: R1205, R1209, R1211, R1242, R1263v1, R1264v1, R1286, R1251, R1283v1, R1296, R1285. The MSA retrieval is the same as described in the above section. The average sequence length is 288, with a minimum length of 59

Table 1: Data distribution of sequence length, max sequence similarity to training set, and number of effective sequences in MSA for RecentPDB-RNA and CASP16-RNA.

		RecentPDB-RNA		CASP16-RNA	
		count	ratio	count	ratio
Seq len	≤400	35	74%	7	64%
	>400	12	26%	4	36%
Max seq sim to train	<0.5	13	28%	4	36%
	0.5-0.6	18	38%	3	27%
	0.6-0.7	11	23%	1	9%
	0.7-0.8	5	11%	0	0%
	≥0.8	0	0%	3	27%
Neff of MSA	0-10	8	17%	1	9%
	10-100	23	49%	5	45%
	100-1000	14	30%	5	45%
	1000-10000	2	4%	0	0%

Table 2: rMSA databases used in RNA MSA search.

Database	Temporal cutoffs
RNAcentral v20.0	2022/3/28
Rfam v14.7	2021/12/9
NCBI NT	2022/10/3

and a maximum length of 833. For distributions of the sequence length, sequence similarity to the training set, and Neff of MSA, please refer to Appendix Table 1.

A.2 Experimental setting

A.2.1 Training setting

Due to AF3's license restrictions, we cannot use it directly; instead, we adopt the open-source reproduction Protenix [14] as our backbone. Specifically, we use the Protenix released checkpoint `model_v0.2.0.pt` for all of our experiments. For the RNA LM, we use AIDO.RNA [12], a strong transformer-based encoder-only language model pretrained on 42 million non-coding RNA sequences from RNAcentral. Specifically, we use `AIDO.RNA-650M` through `AIDO.ModelGenerator` [21]. We train all the models on the RNA3DB dataset with AIDO.RNA frozen whenever it is used. For models with different fusion strategies, we use the same training settings for fair comparisons. We adopt a two-stage training approach, with the first stage warming up the adapters while keeping Protenix's weights frozen. For the cross-attention fusion adapter, we use a learning rate of 0.01; otherwise, we use a learning rate of 0.1. We apply an exponential moving average (EMA) to the model weights with a decay rate of 0.999. We freeze the confidence head and increase the diffusion trunk size to accelerate the training process. We also train two baseline models (with and without MSAs) without any feature fusion using the same setting to understand how the training data contributes to the performance. The detailed training hyperparameters are listed in Appendix Table 3. The global batch size is set to 16, with a micro batch size of 1 and gradient accumulation steps of 4.

Table 3: Training hyperparameters. ${rnalm_fusion_position}, ${rnalm_fusion_method}, ${use_msa} are variables subject to the experiments.

	Training stage 1	Training stage 2
seed	42	42
data.train_sets	rna3db_all	rna3db_all
data.msa.enable_rna_msa	${use_msa}	${use_msa}
dtype	bf16	bf16
diffusion_batch_size	48	48
diffusion_chunk_size	12	12
iters_to_accumulate	4	4
train_crop_size	384	384
max_steps	400	4000
warmup_steps	1	100
learning_rate	0.1/0.01	1e-3
ema_decay	/	0.999
augment.fast_training	True	True
augment.freeze_backbone	True	False
augment.use_rnalm	True	True
augment.rnalm_name	aido_rna_650m	aido_rna_650m
augment.rnalm_fusion_position	${rnalm_fusion_position}	${rnalm_fusion_position}
augment.rnalm_fusion_method	${rnalm_fusion_method}	${rnalm_fusion_method}

A.2.2 Inference setting

Protenix-based models We use the default inference setting in Protenix, with the last EMA checkpoint for each experiment. Note that the Protenix checkpoint was not trained with RNA MSAs.

For models trained without RNA MSAs, we do not use MSAs in inference. For models trained with RNA MSAs, MSAs are utilized during inference. The detailed inference hyperparameters are listed in Appendix Table 4.

Table 4: Inference hyperparameters. ${rnalm_fusion_position}, ${rnalm_fusion_method}, ${use_msa} are variables subject to the experiments.

	Description	Value
seeds	random seeds	101
model.N_cycle	number of recycles in Pairformer	10
use_msa	whether to use MSA or not	$use_msa
sample_diffusion.N_sample	number of structures for each target	5
sample_diffusion.N_step	number of diffusion steps	200
augment.use_rnalm	whether to use RNA LM or not	True
augment.rnalm_name	the RNA LM used	aido_rna_650m
augment.rnalm_fusion_position	RNA LM fusion position	${rnalm_fusion_position}
augment.rnalm_fusion_method	RNA LM fusion method	${rnalm_fusion_method}

Baseline models The model versions and MSA used for the baseline models are shown in Appendix Table 5.

Table 5: Model versions used for baseline model inference.

Model	Version	Training data cutoff	MSA used
Vfold Pipeline	VfoldPipeline-standalone-v2.0 (Download link requested in 2025/9/18)	unknown	No MSA needed
trRosettaRNA	Online server version updated in 2024/11/01, https://yanglab.qd.sdu.edu.cn/trRosettaRNA/	unkown but in 2022/01-2024/11/01	Same as our model
NuFold	Github version http://kiharalab.org/nufold/global_step145245.pt	2022/2/28	Same as our model
RhoFold+	Github version, model_20221010_params.pt, no structure refinement RNA-FM trained on RNAcentral ≤v20.0	2022/4/13 ≤2022/3/28 (RNAcentral)	Same as our model
DRfold2	Github version, model 0,1,2,8,9 in cfg_97, no structure refinement RCLM: epoch_67000 trained on RNAcentral v22.0	2023/12/31 2023/2/8 (RNAcentral)	No MSA needed
AlphaFold 3	Online server https://alphafoldserver.com/	2021/9/30	Searched by AlphaFold 3 Server
Protenix	Github version model_v0.2.0.pt	2021/9/30	Same as our model

A.2.3 Evaluation metrics

Following common practice, we use TM-score (Template Modeling Score) as our major evaluation metric, which is used to assess the structural similarity between the predicted structure and the ground truth structure. It ranges from 0.0 to 1.0, with a higher value indicating a better prediction. A prediction is considered successful if its TM-score is ≥ 0.6. For each target in the test set, we generate 5 predictions. The final score is the average of best-of-5 TM-scores of all targets. The TM-score is computed on the C1' atom using the following USalign [22] script: `USalign {pred_pdb} {true_pdb} -atom " C1'" -m - -mol RNA -TMscore 1`.

A.3 Analysis

Effect of RNA LM and MSA in inference To assess how RNA MSAs influence model performance during inference, we grouped the RecentPDB-RNA test sequences into four bins based on their effective number of sequences (Neff) in the MSA: 0–10, 10–100, 100–1000, and 1000–10000, containing 8, 23, 14, and 2 samples, respectively. We evaluated AlphaFold 3 (as a reference), Protenix, Finetuned Protenix with MSA, and three RLM-aug Protenix variants with MSA. For each model, we plotted the TM-score improvement obtained by inference with MSA compared to inference without MSA (Appendix Figure 1). For the two Protenix models without RNA LM, incorporating MSA generally did not improve performance, especially for low-Neff sequences. In contrast, the three models trained with RNA LM + MSA showed a consistent positive trend: as Neff increased,

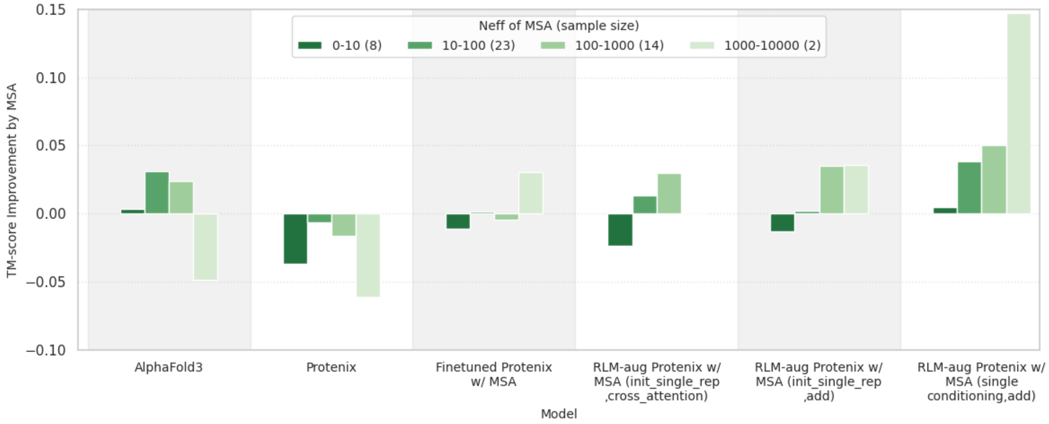

Figure 1: **Effect of MSA in inference on RecentPDB-RNA.** The y-axis denotes the difference of TM-score between using MSAs and not using MSAs during inference for the same model.

the performance gain from MSA became more pronounced. Notably, the RLM-aug Protenix (single-conditioning, add) variant exhibited the strongest ability to leverage MSA information, achieving the largest improvements with higher Neff. These results demonstrate that integrating an RNA LM enhances the model's capacity to exploit evolutionary information from MSAs during inference, suggesting that the RNA LM and MSA act synergistically—the LM provides contextual priors that enable the network to make better use of evolutionary features otherwise underutilized in baseline architectures.

Effect of sequence length To assess the effect of sequence length, we divided the RecentPDB-RNA test set into short ($\leq$ 400 nucleotides, 35 samples) and long ($>$ 400 nucleotides, 12 samples) targets. As shown in Appendix Figure 2, for short sequences, most models perform reasonably well, with RLM-aug Protenix w/ MSA (single-conditioning, add) achieving the highest TM-score of 0.489. For long sequences, performance drops substantially across all models. The Vfold Pipeline fails to generate valid predictions for these targets, while RhoFold+, DRfold2, trRosettaRNA, and NuFold reach TM-scores of 0.132, 0.172, 0.187, and 0.195, respectively, highlighting the significant difficulty of large RNA structure prediction. AlphaFold 3 attains a TM-score of 0.287, demonstrating the strongest robustness to sequence length, likely due to its effective use of RNA MSA and multi-stage training on long sequences. In comparison, RLM-aug Protenix w/ MSA (single-conditioning, add) achieves a TM-score of 0.289, matching AlphaFold 3. When comparing the fine-tuned Protenix variants, we observe that incorporating either MSA or RNA LM embeddings during training improves

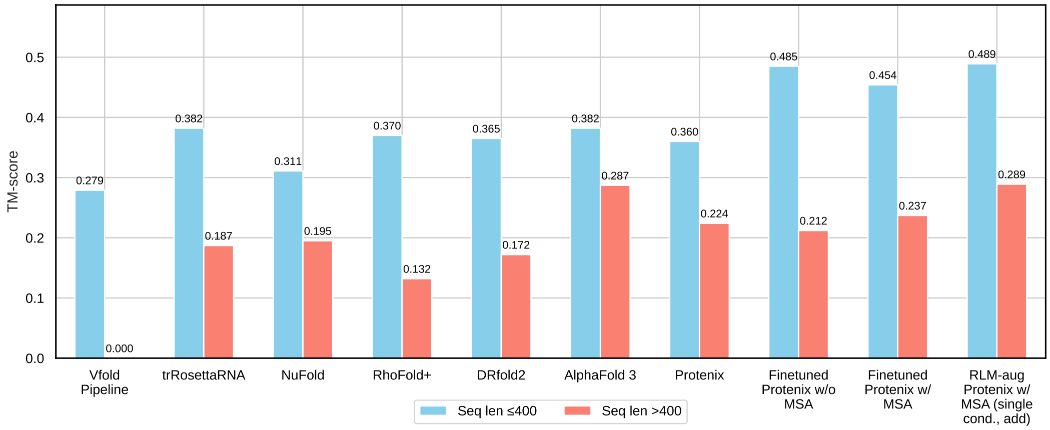

Figure 2: Performance comparison on RecentPDB-RNA by RNA sequence length. The test set is divided into two groups: sequence length $\leq$ 400 (35 samples) and $>$ 400 (12 samples).

performance on long RNA sequences, indicating that evolutionary information plays an important role in modeling large and complex RNA structures.

Case study For illustration, we visualize the predicted structures of our model alongside other methods on the test target 8SYK_A in Figure 3, where the ground truth structure from the PDB is shown in green.

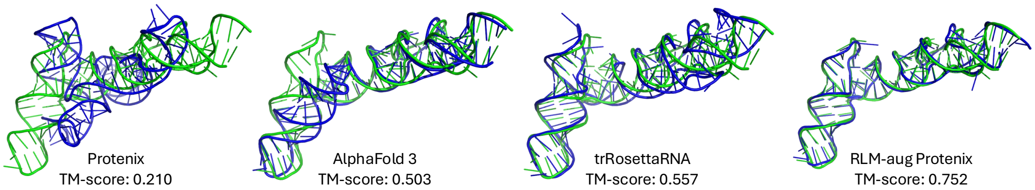

Figure 3: Visualization of PDB structure: 8SYK_A from RecentPDB-RNA. It is a synthetic RNA with 107 nucleotides, where the maximum sequence identity to the training set is 0.50. Green denotes ground truth structure, blue denotes predicted structure of the corresponding model.

A.4 Algorithms

In this section, we present the major algorithms we modified (highlighted in yellow) in AlphaFold 3 [4]. For the meaning of notations, please refer to the AlphaFold 3 paper.

Algorithm 1 Main Inference Loop (Algorithm 1 in AlphaFold 3)

```
def MainInferenceLoop($\{\mathbf{f}^*\}$, rnalm, fusion_position, fusion, $N_{\text{cycle}} = 4, c_s = 384, c_z = 128$):
$\{s_i^{\text{inputs}}\} \leftarrow \text{InputFeatureEmbedder}(\{\mathbf{f}^*\})$
$s_i^{\text{rnalm}} = \text{GetRNAEmbeddings}(\text{rnalm}, \mathbf{f}^*)$
# Fusion position 1
if fusion_position == s_inputs then
    $s_i^{\text{inputs}} \leftarrow \text{fusion}(s_i^{\text{inputs}}, s_i^{\text{rnalm}})$
end if
$s_i^{\text{init}} \leftarrow \text{LinearNoBias}(s_i^{\text{inputs}})$
# Fusion position 2
if fusion_position == s_init then
    $s_i^{\text{init}} \leftarrow \text{fusion}(s_i^{\text{init}}, s_i^{\text{rnalm}})$
end if
$z_{ij}^{\text{init}} \leftarrow \text{LinearNoBias}(s_i^{\text{inputs}}) + \text{LinearNoBias}(s_j^{\text{inputs}})$
# Fusion position 3
if fusion_position == z_init then
    $z_{ij}^{\text{rnalm}} = \text{LinearNoBias}(s_i^{\text{rnalm}}) + \text{LinearNoBias}(s_j^{\text{rnalm}})$
    $z_{ij}^{\text{init}} \leftarrow \text{fusion}(z_{ij}^{\text{init}}, z_{ij}^{\text{rnalm}})$
end if
$z_{ij}^{\text{init}} \mathrel{+}= \text{RelativePositionEncoding}(\{\mathbf{f}^*\})$
$z_{ij}^{\text{init}} \mathrel{+}= \text{LinearNoBias}(f_{ij}^{\text{token\_bonds}})$
$\{\hat{z}_{ij}\}, \{\hat{s}_i\} \leftarrow \mathbf{0}, \mathbf{0}$
for $c \in [1, \ldots, N_{\text{cycle}}]$ do
    $z_{ij} = z_{ij}^{\text{init}} + \text{LinearNoBias}(\text{LayerNorm}(\hat{z}_{ij}))$
    $\{z_{ij}\} = \text{MsaModule}(\{s_i^{\text{msa}}\}, \{z_{ij}\}, \{s_i^{\text{inputs}}\})$
    $s_i = s_i^{\text{init}} + \text{LinearNoBias}(\text{LayerNorm}(\hat{s}_i))$
    $\{s_i\}, \{z_{ij}\} \leftarrow \text{PairformerStack}(\{s_i\}, \{z_{ij}\})$
    $\{\hat{s}_i\}, \{\hat{z}_{ij}\} \leftarrow \{s_i\}, \{z_{ij}\}$
end for
$\{\vec{x}_i^{\text{pred}}\} \leftarrow \text{SampleDiffusion}(\{\mathbf{f}^*\}, \{s_i^{\text{inputs}}\}, \{s_i\}, \{z_{ij}\})$
$p_{ij}^{\text{distogram}} \leftarrow \text{DistogramHead}(z_{ij})$
return $\{\vec{x}_i^{\text{pred}}, p_{ij}^{\text{distogram}}\}$
```

Algorithm 2 Diffusion Conditioning (Algorithm 21 in AlphaFold 3)

```
def DiffusionConditioning( $\hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\}, \{\mathbf{s}_i^{\text{rnalm}}\}, \{\mathbf{z}_{ij}^{\text{rnalm}}\}$, fusion_position,
fusion_method, $\sigma_{\text{data}}, c_z = 128, c_s = 384$) :
    # Pair conditioning, fusion position 5
    if fusion_position == z then
        if fusion_method == add then
            $\mathbf{z}_{ij} = \text{concat}\left([\mathbf{z}_{ij}^{\text{trunk}} + \mathbf{z}_{ij}^{\text{rnalm}}, \text{RelativePositionEncoding}(\{\mathbf{f}^*\})]\right)$
        else
            $\mathbf{z}_{ij} = \text{concat}\left([\mathbf{z}_{ij}^{\text{trunk}}, \text{RelativePositionEncoding}(\{\mathbf{f}^*\}), \mathbf{z}_{ij}^{\text{rnalm}}]\right)$
        end if
    else
        $\mathbf{z}_{ij} = \text{concat}\left([\mathbf{z}_{ij}^{\text{trunk}}, \text{RelativePositionEncoding}(\{\mathbf{f}^*\})]\right)$
    end if                                                              ▷ $\mathbf{z}_{ij} \in \mathbb{R}^{c_z}$
    $\mathbf{z}_{ij} \leftarrow \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$
    for $b \in [1, 2]$ do
        $\mathbf{z}_{ij} \mathrel{+}= \text{Transition}(\mathbf{z}_{ij}, n = 2)$
    end for
    # Single conditioning, fusion position 4
    if fusion_position == s then
        if fusion_method == add then
            $\mathbf{s}_i = \text{concat}\left([\mathbf{s}_i^{\text{trunk}} + \mathbf{s}_i^{\text{rnalm}}, \mathbf{s}_i^{\text{inputs}}]\right)$
        else
            $\mathbf{s}_i = \text{concat}\left([\mathbf{s}_i^{\text{trunk}}, \mathbf{s}_i^{\text{inputs}}, \mathbf{s}_i^{\text{rnalm}}]\right)$
        end if
    else
        $\mathbf{s}_i = \text{concat}\left([\mathbf{s}_i^{\text{trunk}}, \mathbf{s}_i^{\text{inputs}}]\right)$
    end if                                                              ▷ $\mathbf{s}_i \in \mathbb{R}^{c_s}$
    $\mathbf{s}_i \leftarrow \text{LinearNoBias}(\text{LayerNorm}(\mathbf{s}_i))$
    $\mathbf{n} = \text{FourierEmbedding}\left(\frac{1}{4}\log(\hat{t}/\sigma_{\text{data}}), 256\right)$
    $\mathbf{s}_i \mathrel{+}= \text{LinearNoBias}(\text{LayerNorm}(\mathbf{n}))$
    for $b \in [1, 2]$ do
        $\mathbf{s}_i \mathrel{+}= \text{Transition}(\mathbf{s}_i, n{=}2)$
    end for
    return $\{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}$
```

Algorithm 3 Diffusion Module (Algorithm 20 in AlphaFold 3)

```
def DiffusionModule($\{\vec{\mathbf{x}}_l^{\text{noisy}}\}, \hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\}, \{\mathbf{s}_i^{\text{rnalm}}\}, \{\mathbf{z}_{ij}^{\text{rnalm}}\}$, fusion_position, fusion_method, $\sigma_{\text{data}} = 16, c_{\text{atom}} = 128, c_{\text{atompair}} = 16, c_{\text{token}} = 768$) :
    # Conditioning
    $\{\mathbf{s}_i\},\ \{\mathbf{z}_{ij}\} = \text{DiffusionConditioning}\left(\hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\}, \{\mathbf{s}_i^{\text{rnalm}}\}, \{\mathbf{z}_{ij}^{\text{rnalm}}\},\right.$
            $\left.\text{fusion\_position}, \text{fusion\_method}, \sigma_{\text{data}}\right)$
    # Scale positions to dimensionless vectors with approximately unit variance.
    $\mathbf{r}_l^{\text{noisy}} = \vec{\mathbf{x}}_l^{\text{noisy}} / \sqrt{\hat{t}^2 + \sigma_{\text{data}}^2}$    $\triangleright\ \mathbf{r}_l^{\text{noisy}} \in \mathbb{R}^3$
    # Sequence-local Atom Attention and aggregation to coarse-grained tokens
    $\{\mathbf{a}_i\}, \{\mathbf{q}_l^{\text{skip}}\}, \{\mathbf{c}_l^{\text{skip}}\}, \{\mathbf{p}_{lm}^{\text{skip}}\} = \text{AtomAttentionEncoder}\left(\{\mathbf{f}^*\}, \{\mathbf{r}_l^{\text{noisy}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}\}, c_{\text{atom}},\right.$
            $\left.c_{\text{atompair}}, c_{\text{token}}\right)$
                                                        $\triangleright\ \mathbf{a}_i \in \mathbb{R}^{c_{\text{token}}}$
    # Full self-attention on token level.
    $\mathbf{a}_i \mathrel{+}= \text{LinearNoBias}(\text{LayerNorm}(\mathbf{s}_i))$
    $\{\mathbf{a}_i\} \leftarrow \text{DiffusionTransformer}\left(\{\mathbf{a}_i\}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, \beta_{ij} = 0, N_{\text{block}} = 24,\ N_{\text{head}} = 16\right)$
    $\mathbf{a}_i \leftarrow \text{LayerNorm}(\mathbf{a}_i)$
    # Broadcast token activations to atoms and run Sequence-local Atom Attention
    $\{\mathbf{r}_l^{\text{update}}\} = \text{AtomAttentionDecoder}\left(\{\mathbf{a}_i\}, \{\mathbf{q}_l^{\text{skip}}\}, \{\mathbf{c}_l^{\text{skip}}\},\ \{\mathbf{p}_{lm}^{\text{skip}}\}\right)$
    # Rescale updates to positions and combine with input positions
    $\vec{\mathbf{x}}_l^{\text{out}} = \sigma_{\text{data}}^2 / (\sigma_{\text{data}}^2 + \hat{t}^2) \cdot \vec{\mathbf{x}}_l^{\text{noisy}}\ +\ \sigma_{\text{data}} \cdot \hat{t} / \sqrt{\sigma_{\text{data}}^2 + \hat{t}^2} \cdot \mathbf{r}_l^{\text{update}}$
    return $\{\vec{\mathbf{x}}_l^{\text{out}}\}$
```

A.5 Related work

A.5.1 RNA 3D structure prediction methods

Computational modeling of RNA 3D structures, which seeks to predict the atomic positions of nucleotides, has been studied for over two decades. Existing approaches can be broadly categorized into three groups: *ab initio*, template-based, and deep learning–based methods [23].

Ab initio methods simulate the underlying physics of RNA folding by optimizing energy functions through sampling [24, 25, 26]. While physically motivated, these approaches face two key limitations: (1) the simulations are computationally expensive, particularly for large RNAs, and (2) inaccuracies in the energy function can bias sampling and yield incorrect predictions.

Template-based methods leverage the principle that evolutionarily related molecules often adopt similar structures. They construct models using global and local structural information from experimentally solved homologous RNAs [27, 28, 16]. When suitable templates are available, these methods can be highly accurate. However, they are constrained by template availability, which is often lacking for designed or novel RNA sequences.

Deep learning–based methods have recently emerged as powerful alternatives. These approaches train neural networks to predict RNA 3D structures from sequences and/or multiple sequence alignments (MSAs). Based on scope, they can be divided into RNA-specific and general methods. RNA-specific models include DeepFoldRNA [29], trRosettaRNA [7], DRfold [30], RhoFold+ [8], NuFold [9], and DRfold2 [17]. Among these, the first three employ hybrid strategies, combining deep learning for feature learning with energy minimization for final refinement, while the latter three adopt end-to-end architectures inspired by AlphaFold 2 [3]. General-purpose approaches include RoseTTAFoldNA [31], RoseTTAFold All-Atom [32], AF3 [4], and its reproductions such as Protenix [14], Boltz-1 [33], and Chai-1 [34]. Among them, AF3 currently delivers SOTA performance across diverse macromolecular assemblies, but its usage is strictly limited by its license.

A.5.2 Incorporating language models into structure prediction

The integration of pretrained LMs into structure prediction has gained significant attention in recent years due to the huge success of large language models. In the protein domain, ESMFold [35] and HelixFold-single [36] demonstrate that large-scale pretrained protein LMs can substitute for MSAs, achieving performance close to AlphaFold 2 while providing substantially faster inference.

In RNA, recent studies have begun to explore similar directions, not to eliminate MSAs but to improve structural accuracy. RhoFold+ [8] and DRfold2 [17] both incorporate pretrained RNA LMs and report strong improvements in RNA 3D structure prediction. Notably, RhoFold+ retains both the RNA LM and MSA modules, representing a hybrid approach rather than a full replacement.

In this work, we extend AF3's RNA structure prediction capability by incorporating RNA LM representations, emphasizing the enhancement of RNA representation quality in AF3 or AF3-like architectures through effective multimodal fusion.

A.6 Limitations

Despite these advances, our approach has several limitations: (1) it is specialized for RNA structure prediction, leaving its applicability to proteins and DNA uncertain; (2) the confidence prediction head was not fine-tuned, making it an unreliable reference beyond the Protenix version; (3) as a data-driven method, performance strongly depends on the quantity and diversity of training data and the generalization ability to out-of-distribution targets is limited; and (4) the absolute accuracy for large RNA structures remains suboptimal. A natural direction to address the first limitation is to extend our framework by replacing the RNA LM with multimodal biological language models, such as LucaOne [37], thereby enabling all-atom structure prediction across proteins, RNA, and DNA. We leave this exploration for future work.

A.7 Data availability

For the training data, it is publicly available in https://github.com/marcellszi/rna3db/releases/tag/2024-12-04-full-release. The MSAs for the training sequences are publicly available at folder /MSA_v2 in https://www.kaggle.com/competitions/stanford-rna-3d-folding/data.

For the detailed target list and the MSAs of RecentPDB-RNA and CASP16-RNA test sets, we will share them through our Github repository.

A.8 Code availability

Our code is largely based on Protenix https://github.com/bytedance/Protenix and AIDO.ModelGenerator https://github.com/genbio-ai/ModelGenerator. We will share our code and trained models on our GitHub repository.