ACCURATE RNA 3D STRUCTURE PREDICTION VIA LANGUAGE MODEL-AUGMENTED ALPHAFOLD 3

Anonymous authors
Paper under double-blind review

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 variation: the strongest gains come from additive fusion applied at middle or late stages of the conditional network, refining AF3's single representations with RNA LM embeddings. On RecentPDB-RNA (67 newly released structures), our best model achieves a new state of the art with an average TM-score of 0.472 (+21% over AF3) and a 33% success rate (TM-score $\geq$ 0.6), more than doubling AF3's 15%. 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. We will release the data, code, and model weights to support open science, reproducibility, and the development of automated RNA structure prediction models.

1 INTRODUCTION

Accurately predicting 3D structures of RNA molecules from primary sequences is a remaining grand challenge in biology. It is an important step towards understanding the diverse functions of RNA molecules. It also holds great promise for developing RNA-related therapeutics, such as mRNA vaccines, anti-sense oligonucleotide (ASO) and aptamers (1; 2). In recent years, AlphaFold has transformed computational protein structure modeling, achieving predictions with near-experimental accuracy (3; 4). By contrast, RNA structure prediction remains far more challenging. RNA molecules, composed of only four nucleotides, are inherently more dynamic and flexible than proteins, making experimental determination substantially harder. As of July 8, 2025, the Protein Data Bank (PDB) contains only a few thousand RNA structures, the number of which is less than 5% of the number of deposited protein structures (5). Due to the scarcity of experimentally determined RNA structures, RNA 3D structure prediction becomes a small data high-dimensional machine learning problem. As measured in the Critical Assessment of protein Structure Prediction (CASP) 16 blind competition, all the top-performing groups for RNA structure prediction are human expert predictors (6). The reliance on manual expertise in modeling each RNA structure, however, significantly limits the prediction speed and application scope, particularly in the scenario of drug candidate screening.

In this work, we seek an automated model for accurate RNA 3D structure prediction. For a given RNA nucleotide sequence, an automated model can directly output the 3D coordinate prediction for each atom in the RNA molecule without being further processed by human experts. Computational methods have been developed for more than two decades. Early approaches to RNA structure prediction primarily rely on physics-inspired energy functions to simulate molecular folding. Template-based methods, which resemble retrieval strategies, are later introduced to leverage homologous RNA structural information. More recently, with the rise of deep learning and particularly following the success of AlphaFold, deep learning–driven approaches have attracted increasing attention and are rapidly reshaping the field. Methods such as trRosettaRNA (7), RhoFold+ (8) and NuFold (9) are very much inspired by AlphaFold 2, while recently AlphaFold 3 extends its predictions to different molecules including RNA. Systematic benchmarking shows that AF3 is a competitive method that outperforms most of the existing solutions for RNA 3D structure prediction (10).

In parallel with advances in RNA structure modeling, progress in RNA sequence modeling has driven the development of increasingly powerful RNA language models (LMs). 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, integrating information from multiple modalities with the goal of predicting an RNA structure. The technical challenges for multimodal fusion 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. There are five positions in AF3, lying in upstream and downstream of the network, that can be good candidates for feature fusion. For each position, there are several fusion methods, such as add fusion and attention-based fusion, that can be used. It is unclear how to best incorporate RNA LM's representation into AF3.

To investigate this, we design a series of controlled experiments on fusing RNA LM's representation into AF3 [1], keeping the training data and hyperparameters fixed while varying only the fusion positions and methods. We evaluate these models on RecentPDB-RNA, a carefully curated test set consisting of 67 RNA structures from PDB with release dates after the training data temporal cutoff. Experiment results reveal notable variation across fusion strategies: the most effective positions lie in the middle or late stages of the conditional network, and the most effective method is additive fusion. When comparing to existing RNA structure prediction methods, our model with the best fusion strategy achieves a new SOTA in RNA 3D structure prediction on RecentPDB-RNA, with an average TM-score of 0.472, outperforming AF3 by +21%. Moreover, it reaches a success (TM-score ≥ 0.6) rate of 33%, doubling the success rate of AF3 (15%). On 11 CASP16-RNA targets released in 2025, our model 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 Related work

2.1 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 (15).

Ab initio methods simulate the underlying physics of RNA folding by optimizing energy functions through sampling (16; 17; 18). 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 (19; 20; 21). 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 (22), trRosettaRNA (7), DRfold (23), RhoFold+ (8), NuFold (9), and DRfold2 (24). 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

[1] Due to the license of AF3, we use a fully open-sourced reproduction called Protenix (14) in our experiments.

architectures inspired by AlphaFold 2 (3). General-purpose approaches include RoseTTAFoldNA (25), RoseTTAFold All-Atom (26), AF3 (4), and its reproductions such as Protenix (14), Boltz-1 (27), and Chai-1 (28). Among them, AF3 currently delivers SOTA performance across diverse macromolecular assemblies, but its usage is strictly limited by its license.

2.2 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 (29) and HelixFold-single (30) 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 (24) 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.

3 Preliminary

AF3 is a diffusion-based generative model that, conditioned on primary sequences and optional inputs such as multiple sequence alignments (MSAs), predicts all-atom 3D coordinates of biomolecules. For example, for an RNA primary sequence, given noisy coordinates of all the atoms in the RNA molecule, it iteratively denoises them into physically plausible conformations by conditioning on the sequence. Most computation resides in the conditioning network, which takes the primary sequence as input and produces rich single- and pair-wise features that guide the diffusion sampler. As shown in Figure 1, the overall architecture of AF3 contains:

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 outer concatenation of the single representation as inputs for the following Pairformer blocks.

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.

Diffusion Module: A conditional, non-equivariant generative model works 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 is a two-level architecture, operating first on atom-level, then on token-level, and then on atom-level to produce atom-level coordinate predictions.

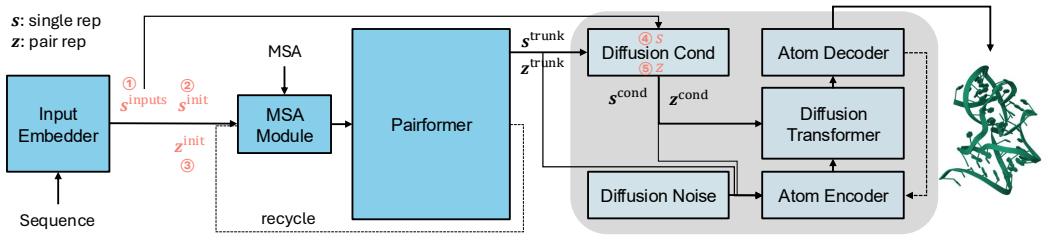

Figure 1: Overview of AF3-style model architecture and the information flow from sequence to single representation and pair representation and to final atomic structure. The grey block denotes the Diffusion Module. The salmon color indicates candidate positions for feature fusion.

4 METHODOLOGY

We extend AF3 to investigate whether self-supervised representations learned from millions of RNA sequences can enhance RNA 3D structure prediction. To this end, we incorporate RNA LM embeddings into AF3, systematically exploring different fusion positions and methods. The basic idea is to refine AF3's single or pair representations using RNA LM embeddings.

4.1 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 $\mathbf{z}^{\text{rnalm}} \in \mathbb{R}^{(N_{\text{token}}, N_{\text{token}}, c_{\text{z}})}$ 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 ($W_1 = W_2$).

4.2 MULTIMODAL FUSION STRATEGIES

Fusion positions By diving into the architecture of AF3, we locate 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 For fusion methods, we adopt commonly used methods as candidates:

1. Add Fusion: add RNA FM's embedding (or its outer concatenation) to the targeted representation;
2. Concat Fusion: concatenate RNA FM's embedding with the targeted representation along the feature dimension;
3. Cross-attention Fusion: treat the targeted representation as a query and RNA FM's embedding as key and value, and use the multi-head cross attention mechanism (31) to extract information from the RNA FM's embedding to the targeted representation.

Note that Concat Fusion will change the targeted representation's dimension, while Add Fusion and Cross-attention Fusion do not change the targeted representation's dimension.

Fusion strategies 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. 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

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

For the detailed algorithms, please refer to the Appendix Section A.2.

5 Experiments

5.1 Training data

For training data, we use RNA3DB, a curated collection of structured RNAs derived from Protein Data Bank (32). 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. 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 (33), which covers 39% of the training sequences.

5.2 Evaluation data

RecentPDB-RNA evaluation set We collected RNA structures from PDB released between December 4, 2024 and April 28, 2025, with resolution better than 4Å and RNA sequence lengths between 30 and 1000 nucleotides. Each complex contains no more than 20 chains. Duplicate sequences were removed within the dataset, as well as any sequences overlapping with the RNA3DB train and test sets. The final dataset comprises 67 unique samples. The average sequence length is 213, with a minimum length of 36 and a maximum length of 814 (Appendix Table 1). 14 sequences have a length over 400 nucleotides. The distribution of sequence similarity between the test set and the training set is shown in Appendix Table 2. We search MSAs for these targets using rMSA (34), which is an automated pipeline to search and align homologs from RNAcentral, Rfam, and nt databases for a target RNA (See Appendix Table 3 for detailed versions and temporal cutoffs).

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. For detailed statistics about the sequence length, MSA depth, and sequence identity with the training set, please refer to Appendix Table 1 and 2.

5.3 Experimental setting

Training setting Due to the license of AlphaFold 3, we cannot use it and instead, we use a successful and fully open-sourced reproduction called Protenix [2] as the backbone for our experiment (14). In specific, we use the Protenix released checkpoint `model_v0.2.0.pt` for all of our experiments. For the RNA foundation model, we use AIDO.RNA, a strong transformer-based encoder-only language model pretrained on 42 million non-coding RNA sequences from RNAcentral (12). In specific, we use `AIDO.RNA-650M` through `AIDO.ModelGenerator` (35). We train all the models on the

[2] https://github.com/bytedance/Protenix

RNA3DB dataset with AIDO.RNA frozen whenever it is used. For models with different fusion strategies, we use the same training setting for fair comparisons. We use a two-stage training, with Stage 1 to warm up the newly initialized weights (adapters) while keeping Protenix frozen and Stage 2 to jointly train Protenix and the adapters. 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 4. The global batch size is set to 16, with a micro batch size of 1 and gradient accumulation steps of 4. For each experiment, training was performed on 4 NVIDIA A100-80GB GPUs with distributed data parallel. The training time for each experiment is about 2.5 days.

Inference setting 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 that trained without RNA MSAs, we do not use MSAs in inference. For models that trained with RNA MSAs, MSAs are utilized during inference. The detailed inference hyperparameters are listed in Appendix Table 5.

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 (36) script: `USalign {pred_pdb} {true_pdb} -atom " C1'" -m - -mol RNA -TMscore 1.`

6 Results and analysis

6.1 Effect of RNA LM feature fusion strategies within AF3-like architecture

We conducted controlled experiments to evaluate different fusion strategies for incorporating RNA LM representations into Protenix, varying only the position and method of feature fusion. Since the original Protenix model was not trained with RNA MSAs, we adopted the same setting and first

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)			✘	0.365	8
Finetuned Protenix	none	none	✘	0.450	18
		none	✔	0.441	19
RLM-aug Protenix	inputs $\mathbf{s}^{\text{inputs}}$	add	✘	0.423	19
		cross attention	✘	0.416	17
	init single rep $\mathbf{s}^{\text{init}}$	add	✘	<u>0.453</u>	<u>21</u>
		add	✔	0.449	21
		concat	✘	0.446	17
		cross attention	✘	<u>0.453</u>	<u>21</u>
		cross attention	✔	0.451	19
	init pair rep $\mathbf{z}^{\text{init}}$	add	✘	0.417	19
	single conditioning $\mathbf{s}$	add	✘	0.431	20
		add	✔	**0.472**	**22**
		concat	✘	0.407	18
	pair conditioning $\mathbf{z}$	add	✘	0.445	18
		concat	✘	0.434	16

trained 11 models without MSA input, including a baseline without any feature fusion for reference. As shown in Table 1, the original Protenix achieves a TM-score of 0.365 with 8 successful predictions out of 67 targets on RecentPDB-RNA. Finetuning Protenix on RNA3DB substantially improves performance, increasing the TM-score to 0.450 and the number of successful predictions to 18. In comparison, RLM-aug Protenix models with feature fusion show diverse performances, highlighting that different fusion strategies have different effects on RNA 3D structure prediction.

Regarding the fusion position, incorporating RNA LM features at the initial single representation $\mathbf{s}^{\text{init}}$ or single conditioning $\mathbf{s}$ is effective when combined with appropriate fusion methods. By contrast, fusing RNA LM representations at the input single representation $\mathbf{s}^{\text{inputs}}$ or pair representations generally degrades the performance. For $\mathbf{s}^{\text{inputs}}$, a plausible explanation is the difficulty of aligning $\mathbf{s}^{\text{inputs}}$ — which is derived from the concatenation of atom-level aggregations, one-hot token embeddings, and additional information — with RNA LM embeddings, which are distributed "semantic" features. The degradation observed when fusing at pair representations may stem from the limited structural information captured in the RNA LM-derived pair representations $\mathbf{z}^{\text{rnalm}}$. Regarding the fusion method, Add Fusion is generally effective, followed by Cross-attention Fusion, while Concat Fusion tends to impair performance, likely due to disruptions in Protenix's information flow caused by altering some of the original weight dimensions.

Given the established effectiveness of MSAs in protein structure prediction and the fact that AF3 was trained with RNA MSAs, we further trained four models using RNA MSAs: three using the top-performing fusion strategies identified above without MSAs — namely (initial single representation, add), (initial single representation, cross-attention), and (single conditioning, add) — and one baseline model without feature fusion as a reference. As shown in Table 1, RLM-aug Protenix (single conditioning, add) trained with RNA MSAs achieves a TM-score of 0.472, surpassing Protenix finetuned with RNA MSAs by 7%. It also improves the success rate by 16% compared to the Protenix finetuned baseline. Although it is hard to disentangle the individual contribution of RNA MSAs and the RNA LM, the results clearly demonstrate that incorporating RNA language models into AF3-like architectures benefits RNA 3D structure prediction.

6.2 Benchmarking against existing RNA structure prediction models

In this section, we benchmark our best fusion model RLM-aug Protenix (single conditioning, add fusion) trained with RNA MSAs against existing automated methods, including AlphaFold 3 [3].

[3] In the following section, unless otherwise specified, RLM-aug Protenix denotes our model trained with (single conditioning, add) fusion strategy and RNA MSAs.

Table 2: **RNA 3D structure prediction results on RecentPDB-RNA and CASP16-RNA test sets.** For the Vfold Pipeline, only 46/67 targets on RecentPBD-RNA and 6/11 targets on CASP16-RNA have predicted 3D structures. We take the averages on those with predicted structures for TM-score. AlphaFold 3 and trRosettaRNA results are obtained from their servers. Vfold (human expert) results are taken from the CASP16 website.

	RecentPDB-RNA (67)		**CASP16-RNA (11)**	
	TM-score ↑	Success rate ↑	TM-score ↑	Success rate ↑
Vfold (human expert)			0.486	36%
Vfold Pipeline* (21)	0.272*	1%	0.289*	0%
trRosettaRNA (7)	0.386	19%	**0.422**	**27%**
NuFold (9)	0.330	3%	0.282	0%
RhoFold+ (8)	0.352	13%	0.306	0%
DRfold2 (24)	0.382	12%	0.342	18%
AlphaFold 3 (4)	0.389	15%	0.402	9%
Protenix (14)	0.365	12%	0.341	18%
RLM-aug Protenix (ours)	**0.472**	**33%**	<u>0.421</u>	**27%**

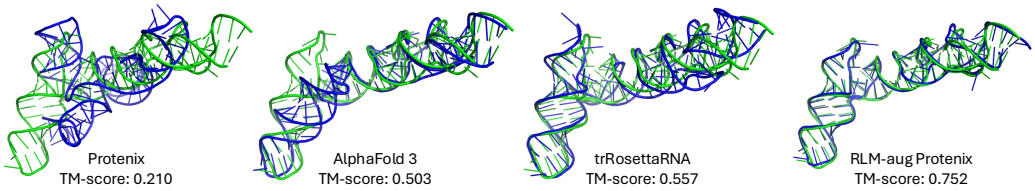

Figure 2: Visualization of PDB structure: 8SYK_A from RecentPDB-RNA. It is a synthetic RNA with 107 nucleotides, where the maximum sequence identity to the training set is 0.50. Green denotes ground truth structure, blue denotes predicted structure of the corresponding model.

Results on RecentPDB-RNA As shown in Table 2, Protenix attains a TM-score of 0.365 on 67 RecentPDB-RNA targets, with a success rate of 12%. The LM-based models RhoFold+ and DRfold2 achieve comparable success rates. By contrast, AlphaFold 3 achieves a slightly higher TM-score of 0.389 and a success rate of 15%, while trRosettaRNA delivers competitive performance, matching AlphaFold 3 in TM-score but surpassing it with a 19% success rate. Our RLM-aug Protenix model sets a new state of the art, achieving a TM-score of 0.472 and a 33% success rate. This represents an improvement of 120% over AlphaFold 3 and 69% over trRosettaRNA in success rate, demonstrating substantial performance gains in RNA 3D structure prediction. For illustration, we visualize the predicted structures of our model alongside other methods on the test target 8SYK_A in Figure 2, where the ground truth structure from the PDB is shown in green.

Results on CASP16-RNA We further evaluate our model on 11 CASP16-RNA targets. On this benchmark, RLM-aug Protenix achieves a TM-score of 0.421 and a success rate of 27%, outperforming most baselines including AlphaFold 3, RhoFold+, and DRfold2, as shown in Table 2. Its performance is also competitive with trRosettaRNA [4], which ranked first in the Server groups in the CASP16 RNA prediction experiment (team name: "Yang-Server") . These results highlight the generalization ability of our RNA LM–augmented Protenix. To gauge the current state of the field, we reference the top-performing group (Vfold with human expert input) from the CASP16 website and observe that a significant gap still separates automated methods, indicating that RNA 3D structure prediction remains a considerable challenge.

6.3 Analysis

Effect of sequence length To assess the impact of sequence length, we divided the RecentPDB-RNA test set into short ($\leq$ 400 nucleotides, 53 samples) and long ($>$ 400 nucleotides, 14 samples) targets. As shown in Figure 3, for short sequences, most models perform reasonably well, with RLM-aug Protenix achieving the highest TM-score (0.522). For long sequences, performance drops substantially across all models, underscoring the challenge of modeling larger RNAs. Among baseline models, AlphaFold 3 achieves the best performance (0.286), while RhoFold+ (0.133) performs poorly. RLM-aug Protenix remains the top performer (0.283), matching AlphaFold 3. Furthermore, RLM-aug Protenix improves upon finetuned Protenix by +5% on short sequences and +25% on long sequences, suggesting that RNA LM representations provide substantial benefits for structure prediction, particularly in the long-sequence regime.

Effect of sequence identity To evaluate how sequence identity between training and test data affects model performance, we grouped the RecentPDB-RNA targets into four categories based on their maximum sequence identity to the training sequences: High identity ($>$ 0.8, 13 samples), Moderate-high identity ((0.6,0.8], 16 samples), Moderate identity ((0.5,0.6], 23 samples), and Low identity ($\leq$ 0.5, 15 samples). As shown in Table 3, the gap between Finetuned Protenix and the original Protenix reflects the contribution of training data, while the gap between RLM-aug Protenix and Finetuned Protenix reflects the added benefit of RNA LM representations. Finetuning directly on RNA3DB substantially improves prediction accuracy for targets with high and moderate-high sequence identity, but yields only marginal gains for low-identity cases. In contrast, integrating RNA

[4]The result for trRosettaRNA is obtained from their server, which is updated on 11/01/2024 after the original publication.

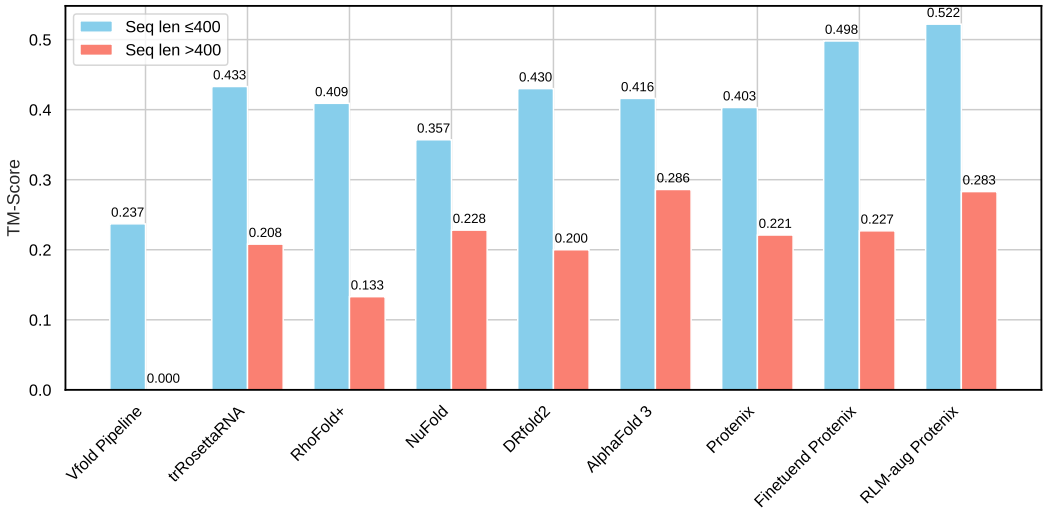

Figure 3: Performance comparison on RecentPDB-RNA by RNA sequence length. The test set is divided into two groups: sequences of length ≤ 400 (53 samples) and those of length > 400 (14 samples).

Table 3: Effect of sequence identity on RecentPDB-RNA. TM-score is reported.

	High identity (13)	Moderate-high identity (16)	Moderate identity (23)	Low identity (15)
[1] Protenix (14)	0.566	0.392	0.326	0.221
[2] Finetuned Protenix	0.677	0.570	0.355	0.230
[3] RLM-aug Protenix (ours)	0.678	0.605	0.377	0.296
Δ ([2] - [1])	**+0.111**	**+0.178**	**+0.029**	+0.009
Δ ([3] - [2])	+0.001	+0.035	+0.022	**+0.065**

LM representations enhances performance across all similarity levels, with a large gain observed for low-identity targets. These findings indicate that RNA LM features improve generalization beyond training-like examples.

7 Conclusions and Future Work

In this work, we systematically explore strategies for integrating RNA LM representations into an AF3-like architecture to improve RNA 3D structure prediction. Our results show that injecting the LM representation into the single representation of the Diffusion Conditioning Module yields the most effective performance, achieving SOTA or near-SOTA performance on two test sets. Additional analyses further suggest that RNA LMs are particularly beneficial for predicting large RNA structures.

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.

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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[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 EVALUATION DATA

Table 1: Data distribution of sequence length and MSA depth for RecentPDB-RNA and CASP16-RNA.

	Seq length					MSA depth		
	min	max	mean	#(≤ 400)	#(> 400)	min	max	mean
RecentPDB-RNA	36	814	213	53	14	3	22312	2368
CASP16-RNA	59	833	288	7	4	17	2893	1391

Table 2: Distribution of sequence identity between test and training sets. For each test sequence, we compute its maximum sequence similarity between the training sequences as its sequence identity to the training data.

Seq ID	RecentPDB-RNA		CASP16-RNA	
	Count	Ratio	Count	Ratio
$\leq$0.5	15	22%	4	36%
(0.5,0.6]	23	34%	3	27%
(0.6,0.8]	16	24%	1	9%
$>$0.8	13	19%	3	27%
Total	67	100%	11	100%

Table 3: 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

A.2 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**
  $\mathbf{z}_{ij}$ += $\text{Transition}(\mathbf{z}_{ij}, n = 2)$
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 ▷ $\mathbf{s}_i \in \mathbb{R}^{c_s}$
$\mathbf{s}_i \leftarrow \text{LinearNoBias}(\text{LayerNorm}(\mathbf{s}_i))$
$\mathbf{n} = \text{FourierEmbedding}\left(\frac{1}{4}\log(\hat{t}/\sigma_{\text{data}}), 256\right)$
$\mathbf{s}_i$ += $\text{LinearNoBias}(\text{LayerNorm}(\mathbf{n}))$
for $b \in [1, 2]$ **do**
  $\mathbf{s}_i$ += $\text{Transition}(\mathbf{s}_i, n{=}2)$
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

$\{\mathbf{s}_i\},\ \{\mathbf{z}_{ij}\} = \text{DiffusionConditioning}\left(\hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\}, \{\mathbf{s}_i^{\text{rnalm}}\}, \{\mathbf{z}_{ij}^{\text{rnalm}}\}, \text{fusion\_position}, \text{fusion\_method}, \sigma_{\text{data}}\right)$

 # Scale positions to dimensionless vectors with approximately unit variance.

$\mathbf{r}_l^{\text{noisy}} = \vec{\mathbf{x}}_l^{\text{noisy}} / \sqrt{\hat{t}^2 + \sigma_{\text{data}}^2}$ ▷ $\mathbf{r}_l^{\text{noisy}} \in \mathbb{R}^3$

 # Sequence-local Atom Attention and aggregation to coarse-grained tokens

$\{\mathbf{a}_i\}, \{\mathbf{q}_l^{\text{skip}}\}, \{\mathbf{c}_l^{\text{skip}}\}, \{\mathbf{p}_{lm}^{\text{skip}}\} = \text{AtomAttentionEncoder}\left(\{\mathbf{f}^*\}, \{\mathbf{r}_l^{\text{noisy}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}\}, c_{\text{atom}}, c_{\text{atompair}}, c_{\text{token}}\right)$ ▷ $\mathbf{a}_i \in \mathbb{R}^{c_{\text{token}}}$

 # 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.3 Models

A.3.1 Training hyperparameters

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.

Table 4: 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.3.2 INFERENCE HYPERPARAMETERS

Table 5: 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}

A.4 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.5 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.