RiboC2F: Pose-Concentrated Coarse-to-Fine Flow Matching for Protein-Conditioned RNA Co-Design

Dengdeng Huang
School of Computer Science
Shanghai Jiao Tong University
Shanghai, China
erika_dd@sjtu.edu.cn

Shikui Tu*
School of Computer Science
Shanghai Jiao Tong University
Shanghai, China
tushikui@sjtu.edu.cn

Abstract

Computational design of ribonucleic acid (RNA) is essential for therapeutic discovery and synthetic biology. Designing RNAs that bind specified protein targets is bottlenecked by a scarcity of paired-complex 3D structural data. Existing approaches focus on importing structural priors from external biomolecular datasets, leaving where to localize the complex signal itself unresolved. We propose RiboC2F, a coarse-to-fine flow-matching framework that concentrates this signal primarily on the global binding pose. RiboC2F runs a two-stage cascade over a shared trunk, where a coarse stage proposes the binding pose and a fine stage generates the full RNA design. The coarse pose enters as a conditioning signal rather than a fixed initialization, avoiding cross-stage error accumulation. On PRI30k, RiboC2F delivers the strongest docking among published baselines, with competitive intrinsic validity and multi-sample diversity. Probing further confirms that pose orientation is the operative component of the coarse signal. Ablations show that the two-stage cascade makes a key contribution to the framework's performance gains. Code is available at https://github.com/CMACH508/RiboC2F.

1 Introduction

Ribonucleic acid (RNA) is a functionally versatile biomolecule central to gene regulation and catalysis. Many RNAs act through specific interactions with proteins, making protein-conditioned RNAs important design targets [1]. This has motivated growing interest in computational RNA modeling, spanning both structural understanding [2–4] and generative design [5, 6]. Notably, unlike small-molecule ligands, RNAs vary widely in length, and a long RNA typically contacts the protein over only part of its length [7]. The bound conformation therefore reflects two determinants: the protein-conditioned placement and interface contacts, and the protein-independent folding rules intrinsic to the RNA [8]. However, RNA–protein complexes with experimentally resolved 3D structures are scarce, an order of magnitude fewer than the protein-only corpora used for structure-based design. This scarcity leaves protein-conditioned generation with insufficient supervision to acquire both determinants from scratch.

To compensate, recent work has focused its design effort on structural priors from larger related biomolecular datasets. RiboFlow [9] initializes from a pretrained RNA-only generative model and finetunes with distance-aware refinement. RiboDiff [10] and RNAFlow [11] build on RF2NA [3] as their structural prior. However, a key allocation question of the complex data itself remains unresolved: *where* should the scarce complex signal be concentrated? Across these approaches, complex signal allocation is overlooked: pretrain-then-condition pipelines fold the RNA before introducing protein supervision, and joint finetuning distributes supervision uniformly across all

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

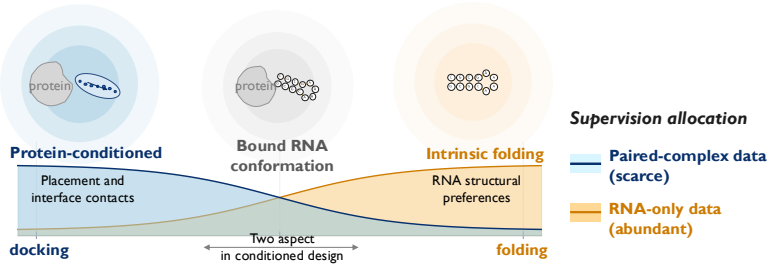


Figure 1: Supervision allocation in protein-conditioned RNA co-design. The bound RNA conformation reflects protein-conditioned placement and interface contacts as well as intrinsic folding preferences. Scarce paired-complex data supervise the former, while abundant RNA-only data provide structural priors for the latter.

design aspects. This omission can weaken conditional generation, as the scarce complex signal may be reduced to a post-hoc correction, or spread across capabilities the prior already covers. Figure 1 illustrates such a supervision allocation.

To fill this gap, we adopt a signal-concentrating cascade for conditional RNA design. The core idea is to single out the RNA's rigid-body placement relative to the protein, which we define as the ***pose***. Based on this definition, the model encodes where the signal should be concentrated: primarily on the pose. That is, the cascade treats generation as a coarse-to-fine progression in granularity, in which the pose serves both as a coarse description of the RNA and as a critical feature for its fine co-design.

Following this strategy, we propose RiboC2F, a pose-concentrated flow-matching framework organized as a two-stage cascade. To realize this, we design a shared trunk that serves both stages, distinguished by stage-specific time variables and trained in alternation. Both stages start from noise, generating the pose at the coarse stage and the full RNA structure and sequence at the fine stage. The coarse pose serves as an additional constraint rather than a fixed initialization for the fine stage, avoiding cross-stage error accumulation. Meanwhile, by featurizing protein and RNA on an equal footing, the protein condition becomes a direct source of supervision rather than an auxiliary signal.

RiboC2F has two distinct advantages: First, concentrating complex supervision on the binding pose makes more efficient use of the scarce data. Second, coupling the two stages through feature-level conditioning allows the pose to benefit the co-design without accumulating error across stages. To evaluate RiboC2F, we compare it against several published conditioned RNA generators. Our evaluation covers complex docking, RNA intrinsic validity, and multi-sample diversity. Results show RiboC2F achieves the strongest docking among published baselines, with competitive intrinsic RNA quality. Ablations isolate the two-stage cascade as the source of this gain, and further probing shows that the coarse signal carries strong directional information that is what the fine stage needs for the conditional generation.

2 Related Work

RNA structure and sequence modeling. A first line of computational RNA work treats the molecule on its own, covering subtasks such as structure prediction, inverse folding, and unconditional generation. Folding models such as trRosettaRNA [12] and RoseTTAFold2NA [3] predict 3D structure from sequence, providing strong single-molecule structural priors. Inverse-folding methods such as gRNAde [13] design sequences compatible with a given target fold, enabling fold-conditioned sequence design. Unconditional generative models such as RNA-FrameFlow [6] sample new RNAs under a light prior, introducing SE(3)-equivariant flow-matching to the RNA setting. Collectively these works shape and search RNA structure–sequence space, yet leave open how to generate RNAs conditioned on a specified protein partner.

Conditioned biomolecular design. Conditioned biomolecular design generates molecules under a specified target context, such as a binding partner, a pocket, or a structural motif. Existing methods can be grouped by how the conditioning is integrated into the generative model. Factorized methods use a frozen pre-trained component for part of the task, with LigandMPNN [14] designing protein

sequences against a fixed backbone and ligand pocket and RNAFlow [11] delegating the folding half of RNA design to a frozen RF2NA [3]. End-to-end conditional generation jointly models the target and the molecule in a single trainable network, and has become the dominant paradigm across biomolecular modalities including protein, RNA, and small-molecule design [5, 15–17]. Pretrain-and-finetune methods start from unconditional pretraining and add conditioning during fine-tuning, as in RiboFlow [9] for protein-conditioned RNA design.

3 Problem Formulation

The protein-conditioned RNA co-design problem seeks to simultaneously generate an RNA sequence and its full-atom three-dimensional structure that binds a given target protein.

An RNA molecule has a backbone made of alternating phosphate groups (P, $OP1$, $OP2$, $O5'$) and the sugar ribose ($C1'$–$C5'$, $O2'$, $O3'$, $O4'$). A nitrogen atom at the base attachment site ($N9$ for purines or $N1$ for pyrimidines) is also typically included in modeling. Utilizing recent RNA parameterization methods [6], the N_a heavy atoms of each nucleotide are described compactly through a local rigid frame, a set of torsion angles, and idealized bond lengths and angles. The frame for residue i, denoted $\mathbf{F}_i = (\mathbf{R}_i, \mathbf{x}_i) \in \mathrm{SE}(3)$, is built from its $C3'$, $C4'$, and $O4'$ atoms via Gram–Schmidt, with translation component $\mathbf{x}_i \in \mathbb{R}^3$ and rotation component $\mathbf{R}_i \in \mathrm{SO}(3)$. The torsion configuration $\boldsymbol{\chi}_i \in \mathbb{R}^{8\times 2}$ uses a sin/cos parameterization. The residue type is denoted $s_i \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$. A length-$L_R$ RNA is then represented as the tuple $(\mathbf{x}_R, \mathbf{R}_R, \boldsymbol{\chi}_R, \mathbf{s}_R) := (\{\mathbf{x}_i\}_{i=1}^{L_R}, \{\mathbf{R}_i\}_{i=1}^{L_R}, \{\boldsymbol{\chi}_i\}_{i=1}^{L_R}, \{s_i\}_{i=1}^{L_R})$.

The conditioning context $\mathcal{C} = (\mathbf{s}_P, \mathbf{X}_P, \mathcal{L})$ comprises the target protein's amino-acid sequence $\mathbf{s}_P \in \{\mathrm{A}, \ldots, \mathrm{Y}\}^{L_P}$ and full-atom structure $\mathbf{X}_P \in \mathbb{R}^{L_P \times N_a^P \times 3}$ (with N_a^P heavy atoms per residue), both held fixed throughout generation, plus an optional bound ligand $\mathcal{L} = \{(\mathbf{a}_j, e_j)\}_{j=1}^{M}$, a small-molecule point cloud of M heavy atoms with coordinates $\mathbf{a}_j \in \mathbb{R}^3$ and element types e_j. Our goal is to learn the conditional joint distribution:

$$p_{\boldsymbol{\theta}}\big(\mathbf{x}_R,\, \mathbf{R}_R,\, \boldsymbol{\chi}_R,\, \mathbf{s}_R \,\big|\, \mathcal{C}\big). \tag{1}$$

4 Method

In this section, we present the concrete implementation of the RiboC2F framework, in two parts: Sec. 4.1 specifies how each of the RNA modalities is defined under flow matching, and Sec. 4.2 describes how we single out the binding pose to realize the coarse-to-fine cascade. Figure 2 gives the overall architecture, with full architectural details and training hyperparameters in Appendix A and F.

4.1 Per-modality flow matching

Frame. The frame of RNA residue i consists of a translation $\mathbf{x}_i \in \mathbb{R}^3$ (the $C4'$ coordinate) and a rotation $\mathbf{R}_i \in \mathrm{SO}(3)$. We model both with conditional flow matching (CFM) [18]: translations on $\mathbb{R}^3$ follow the linear interpolant of Tong et al. [19], and rotations on $\mathrm{SO}(3)$ follow the unit-speed geodesic interpolant of Yim et al. [20, 21]. Learned velocity fields $v_{\boldsymbol{\theta}}^{\mathrm{trans}}$ and $v_{\boldsymbol{\theta}}^{\mathrm{rot}}$ regress against the conditional vector field of each interpolant:

$$\mathcal{L}_{\mathrm{trans}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\mathbf{x}_1} \sum_i \big\| v_{\boldsymbol{\theta}}^{\mathrm{trans}}(\mathbf{x}_t^{(i)}, t) - (\mathbf{x}_1^{(i)} - \mathbf{x}_0^{(i)}) \big\|_2^2, \tag{2}$$

$$\mathcal{L}_{\mathrm{rot}} = \mathbb{E}_{t,\,\mathbf{R}_0,\,\mathbf{R}_1} \sum_i \left\| v_{\boldsymbol{\theta}}^{\mathrm{rot}}(\mathbf{R}_t^{(i)}, t) - \frac{\log_{\mathbf{R}_t^{(i)}} \mathbf{R}_1^{(i)}}{1-t} \right\|_{\mathrm{SO}(3)}^2, \tag{3}$$

where $t \sim p_T$ and $(\mathbf{x}_t, \mathbf{R}_t)$ are drawn along the conditional paths of the two interpolants. Prior choices, sampling procedures, and the explicit conditional path expressions are in Appendix C.

Torsion. The eight per-nucleotide torsion angles $\boldsymbol{\chi}_i \in \mathbb{R}^{8\times 2}$ (sin/cos form) determine the sugar and base side-group conformation within each frame. Defining a flow-matching path on the 8-torus is technically nontrivial, requiring geodesic interpolation that respects the torus periodicity, and no prior RNA flow-matching framework adopts a torsion flow. We thus simply follow Ma et al. [9] and use

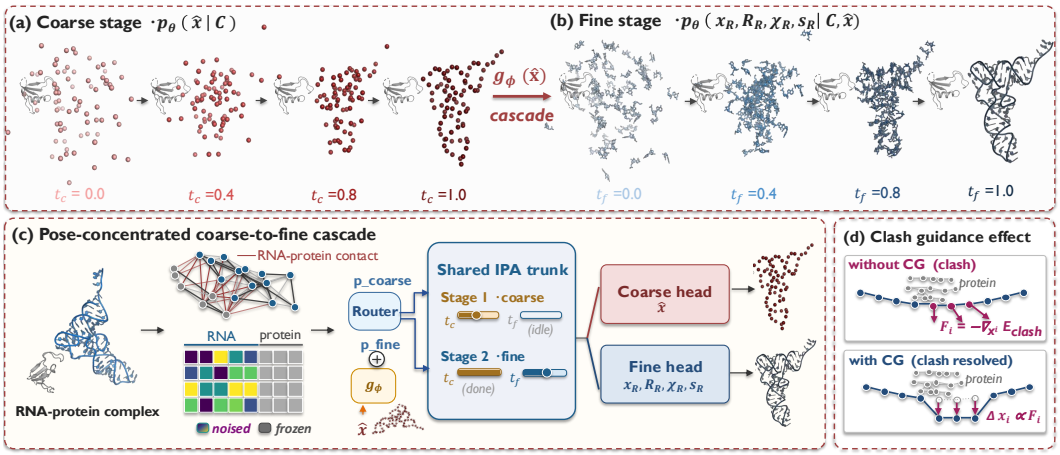


Figure 2: (a) Coarse stage trajectory along t_c: translations denoise into the binding pose. (b) Fine stage trajectory along t_f: full codesign unfolds under the proposed coarse pose. (c) Architecture of the pose-concentrated coarse-to-fine cascade. (d) Clash guidance resolves residual RNA–protein atom-level overlaps at sampling time.

the AngleResnet of Anand et al. [6] for direct torsion prediction, supervised by sin/cos mean squared error (MSE):

$$\mathcal{L}_{\text{tors}} = \frac{1}{8\,L_R} \sum_{i=1}^{L_R} \sum_{k=1}^{8} \|\hat{\boldsymbol{\chi}}_{1,k}^{(i)} - \boldsymbol{\chi}_{1,k}^{(i)}\|_2^2. \tag{4}$$

Residue type. We adopt the discrete masking flow of Campbell et al. [22] for the categorical nucleotide types. Each residue type is encoded as a one-hot probability vector $\mathbf{s}^{(i)} \in \Delta^5$ over the extended alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}, [\mathrm{MASK}]\}$, with $\mathbf{s}_1^{(i)} = \mathrm{onehot}(c^{(i)})$. The prior concentrates all mass on $[\mathrm{MASK}]$ and the conditional probability path linearly interpolates between $[\mathrm{MASK}]$ and the data,

$$p_t\big(\mathbf{s}_t^{(i)} \mid \mathbf{s}_1^{(i)}\big) = t\,\mathbf{s}_1^{(i)} + (1-t)\,\delta_{[\mathrm{MASK}]}. \tag{5}$$

The model parameterizes a softmax distribution $\hat{\mathbf{p}}_{\boldsymbol{\theta}}(\mathbf{s}_t^{(i)}, t)$ over the alphabet, supervised by per-position cross-entropy:

$$\mathcal{L}_{\text{type}} = \mathbb{E}_{t,\,\mathbf{s}_1} \sum_i \mathrm{CE}\big(\hat{\mathbf{p}}_{\boldsymbol{\theta}}(\mathbf{s}_t^{(i)}, t),\, \mathbf{s}_1^{(i)}\big). \tag{6}$$

4.2 Coarse-to-fine architecture

To concentrate the scarce paired-complex supervision on the binding pose, we introduce an auxiliary pose variable $\hat{\mathbf{x}} \in \mathbb{R}^{L_R \times 3}$ as a coarse-stage estimate of $\mathbf{x}_R$ and factorize the generative model via the chain rule:

$$p_{\boldsymbol{\theta}}\big(\hat{\mathbf{x}}, \mathbf{x}_R, \mathbf{R}_R, \boldsymbol{\chi}_R, \mathbf{s}_R \,\big|\, \mathcal{C}\big) = \underbrace{p_{\boldsymbol{\theta}}\big(\hat{\mathbf{x}} \,\big|\, \mathcal{C}\big)}_{\text{coarse}} \cdot \underbrace{p_{\boldsymbol{\theta}}\big(\mathbf{x}_R, \mathbf{R}_R, \boldsymbol{\chi}_R, \mathbf{s}_R \,\big|\, \mathcal{C}, \hat{\mathbf{x}}\big)}_{\text{fine}}. \tag{7}$$

Through this cascade, the coarse factor $p_{\boldsymbol{\theta}}(\hat{\mathbf{x}} \mid \mathcal{C})$ carries the bulk of the paired-complex supervision (Fig. 2(a)), while $\hat{\mathbf{x}}$ enters the fine factor as a conditioning signal alongside $\mathcal{C}$ (Fig. 2(b)). We treat protein and RNA on equal footing as two classes to enable direct use of the protein condition, with each represented as residue-level features and interaction edges that jointly serve as input to the Invariant Point Attention (IPA) [23] trunk. Full node and edge feature compositions are deferred to Appendix B.

Dual-time shared trunk. Since both stages supervise quantities in the same residue-level coordinate space, we share a single IPA trunk $f_{\boldsymbol{\theta}}$ across them for parameter efficiency, with two stage-specific output heads. A dual-time schedule informs the trunk of the active stage by conditioning $f_{\boldsymbol{\theta}}$ on two

independent time variables $(t_c, t_f) \in [0,1]^2$ added to every node feature. A coarse forward pass sets $t_f = 0$ and runs the translation flow along t_c, producing the coarse-grained binding pose from noise. A fine forward pass instead sets $t_c = 1$, indicating that the coarse-grained pose has been obtained, and runs all four modality flows along t_f. At each training step, a Bernoulli stage selector (the *router* in Fig. 2(c)) picks the active stage and dispatches the trunk's output to the matching head.

Coarse stage. The coarse stage generates the RNA's binding pose, operating only on per-residue translations. As a coarse-grained signal, this stage targets a rough geometric profile rather than precise residue coordinates, including the direction along which the RNA extends and the overall geometric envelope (e.g., elongated or compact). Accordingly, we design three lightweight auxiliary losses on top of the base translation regression $\mathcal{L}_{\text{trans}}$: $\mathcal{L}_{\text{iface}}$ up-weights RNA–protein distance errors for residues within 10 Å of any protein heavy atom, $\mathcal{L}_{\text{ori}}$ takes a Frobenius residual between the predicted and ground-truth RNA principal-axis frames, and $\mathcal{L}_{\text{pd}}$ matches pairwise $C4'$-$C4'$ distance matrices in ℓ_2. The coarse objective is their weighted sum:

$$\mathcal{L}_{\text{coarse}} = \mathcal{L}_{\text{trans}} + \lambda_{\text{ori}}\,\mathcal{L}_{\text{ori}} + \lambda_{\text{if}}\,\mathcal{L}_{\text{iface}} + \lambda_{\text{pd}}\,\mathcal{L}_{\text{pd}}. \tag{8}$$

In the auxiliary training loss $\mathcal{L}_{\text{ori}}$, *orientation* is represented by the molecule's principal-axis frame, computed from the translations alone. It is distinct from the per-residue frame rotations $\mathbf{R}_i$ of Sec. 4.1, which are left to the fine stage. Exact definitions and weight choices are deferred to Appendix D.

Coarse-to-fine coupling. We inject the coarse pose $\hat{\mathbf{x}}$ into the fine stage as a conditioning signal via a learned residue-wise embedding $g_{\boldsymbol{\phi}} : \mathbb{R}^3 \to \mathbb{R}^{d_s}$ added to every RNA node feature:

$$\mathbf{h}_i^R \leftarrow \mathbf{h}_i^R + g_{\boldsymbol{\phi}}(\hat{\mathbf{x}}_i). \tag{9}$$

$g_{\boldsymbol{\phi}}$ injects $\hat{\mathbf{x}}$ as a feature rather than as the fine stage's initial state, preventing error accumulation along the fine denoising trajectory. This is also the natural choice within the flow-matching framework, since the flow draws its sample stochasticity from the noise prior and would collapse to a one-to-one map with a deterministic start. Additionally, $g_{\boldsymbol{\phi}}$ operates in the protein-anchored reference frame defined by the fixed backbone, which absorbs the global SE(3) symmetry for the shared IPA trunk.

Fine stage. The fine stage takes on the full codesign-from-noise task under the guidance of the pose embedding. By default, we initialize the IPA from the unconditional RNA generative prior [6]. This responds to the two-aspect view of RNA, with the coarse pose supporting the local interface reshaping and the pretrain initialization supporting the intrinsic folding rules. During training, the model does not repeat the coarse generation process, instead assuming the coarse stage is approximately converged. We thus obtain the surrogate $\tilde{\mathbf{x}} = (1-\beta)\mathbf{x}_1 + \beta\boldsymbol{\varepsilon}$ directly by adding noise to the ground-truth translations, with $\beta \sim \mathcal{U}(0, 0.25)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I})$. The fine objective is the weighted sum of the four per-modality losses:

$$\mathcal{L}_{\text{fine}} = \lambda_{\text{trans}}\,\mathcal{L}_{\text{trans}} + \lambda_{\text{rot}}\,\mathcal{L}_{\text{rot}} + \lambda_{\text{tors}}\,\mathcal{L}_{\text{tors}} + \lambda_{\text{type}}\,\mathcal{L}_{\text{type}}. \tag{10}$$

Weight choices are deferred to Appendix E.

Training-time vs. inference-time ordering. The cascade allocation translates into asymmetric schedules at training and inference. Training updates only one stage per step under an alternating Bernoulli, while inference runs the two stages sequentially from coarse to fine. Concretely, each training step draws the active stage with $p_{\text{coarse}} = 0.6$ and $p_{\text{fine}} = 0.4$, and the active stage's time variable is sampled from a logit-normal distribution clipped to $[\epsilon, 1-\epsilon]$. At inference, coarse first produces a clean pose $\hat{\mathbf{x}}$, and fine then consumes it via $g_{\boldsymbol{\phi}}$ in place of the training-time surrogate $\tilde{\mathbf{x}}$, Euler-integrating from its noise prior to deliver the full bound RNA. Time-distribution parameters are given in Appendix F.

Clash guidance. Hard atom-level overlaps, physically unrealizable and damaging to interface-geometry scores, are only implicitly avoided by coordinate regression toward clash-free ground truths. We therefore design Clash Guidance (CG), a training-free, parameter-free sampling-time module that, at each fine Euler step, applies a translation-only gradient on a violation-pair quadratic-hinge potential under a linear time schedule. With $\mathbf{z}_t \in \mathbb{R}^{L_R \times 3}$ the current $C4'$ translations and $\mathbf{x}^P \in \mathbb{R}^{L_P \times 3}$ the protein C_α coordinates (held as a stop-gradient constant), the violation energy and update at sampling

step t are

$$E(\mathbf{z}_t) = \sum_{i,j} m_{ij} \left[\mathrm{ReLU}\big(\tau - \|\mathbf{z}_{t,i} - \mathbf{x}_j^P\|_2\big)\right]^2, \tag{11}$$

$$\mathbf{z}_t \leftarrow \mathbf{z}_t - \lambda(t)\, \nabla_{\mathbf{z}_t} E(\mathbf{z}_t), \qquad \lambda(t) = \lambda_0\, t, \tag{12}$$

with threshold $\tau = 5.0\,\text{Å}$, base scale $\lambda_0 = 1.0$, and pair mask m_{ij} that excludes padding residues. Algorithmic details and complexity are in Appendix G.

5 Experiments

5.1 Experimental setup

Datasets. We build training and test sets from PRI30k [24], currently the most comprehensive non-redundant collection of protein–RNA interaction complexes. CoPRA [24] constructs PRI30k from the $5{,}909$ protein–RNA complexes in the PDB by reducing the ~ 150k raw interacting chain pairs annotated in BioLiP2 [25] to a non-redundant set via a maximum-connected-subgraph algorithm. On top of PRI30k, we further restrict to compute-feasible complexes with $5 \le L_R \le 200$ and $L_P \le 200$, and additionally drop entries whose RNA backbone is split into multiple disconnected segments. We partition the resulting set in two steps. First, we cluster RNAs using MMseqs2 at 80% sequence identity, assign whole clusters to either the train or the test side without splitting any cluster, and cap each test cluster at 20 samples to prevent over-representation of any single cluster. Second, to control for protein-side overlap with training, we remove from the test set any complex whose RNA or protein sequence exactly matches a training entry. The final partition contains $9{,}106$ training samples and 324 leakage-controlled test samples. Filtering details are in Appendix K.

Baselines and RiboC2F variants. We compare against three published RNA design methods. **(i) MMDiff** [15] is an SE(3) joint diffusion model for unconditional protein–RNA complex co-design. We hold its protein chain fixed at inference to obtain protein-conditioned RNA samples. **(ii) RNAFlow** [11] factorizes conditional RNA design into an inverse-folding network conditioned on the protein context, whose RNA-sequence prediction is re-folded into structure by a frozen RF2NA [3]. **(iii) RiboFlow** [9] generates conditioned RNA by refining its unconditional iterates through a distance-aware module. Originally a *ligand*-conditioned RNA design method, it learns an RNA-only prior before conditional finetuning. We finetune this prior on our protein–RNA dataset. MMDiff and RNAFlow are evaluated with their original published checkpoints; only RiboFlow is retrained on our PRI30k split, for parity with our protein-conditioned setup. We further report RiboC2F under two binary design knobs: trunk initialization from the RNA-FrameFlow checkpoint (PRE) and inference-time clash guidance (CG), each either on or off. The default RiboC2F has both enabled, and *w/o* rows in Table 1 mark the knob being switched off. For the cascade structure, we consider two single-stage reductions: **Fine-only**, which removes the coarse stage and reduces the model to a single-stage flow-matching denoiser, and **Coarse-only**, which trains only the translation flow and generates no sequence. Training and inference settings are in Appendix L.

5.2 Protein-conditioned RNA co-design

This subsection evaluates the co-design task, where the model receives a protein target and jointly outputs an RNA sequence and a full-atom 3D structure that bind it. The reported metrics jointly capture docking accuracy, intrinsic RNA quality, and the diversity-novelty profile across multiple samples. All single-sample metrics are reported on one independently drawn sample per target, while the multi-sample diversity and novelty metrics use $k = 10$ samples per target.

Metrics. *Intrinsic quality* captures whether the generated RNA looks like a valid RNA structure and whether its sequence is consistent with that structure. **Val%**, the fraction of samples passing three intra-RNA structural checks: chain-break, atomic overlap, and radius-of-gyration deviation (Appendix H). **scTM%**, the fraction of complexes whose generated sequence, re-folded by Boltz-2 [26], attains TM-score [27] ≥ 0.45 to our generated structure, tests whether the sequence and structure are internally consistent under an independent structure predictor, with an extended discussion including oracle reliability bands given in Appendix I. **lDDT** [28], the local Distance Difference Test on $C4'$ atoms,

Table 1: Comparison on protein-conditioned RNA co-design. **Bold**: best per column. Underlined: second-best.

	Intrinsic quality					Docking			Multi-sample	
Method	Val%↑	scTM%↑	lDDT↑	SR%↑	Clash↓	dG%↑	Fnat↑	Acc.%↑	Div↑	Nov%↑
MMDiff	27.86	0.31	0.290	18.82	0.61	1.60	0.016	0.62	0.291	**7.3**
RNAFlow	73.63	0.00	0.422	26.45	0.60	—	0.002	0.00	**0.749**	0.8
RiboFlow	45.06	0.31	0.483	26.84	1.31	11.54	0.054	4.32	0.430	1.4
RiboC2F	81.48	**14.55**	**0.499**	30.09	**0.38**	**19.23**	0.160	21.60	0.419	2.2
w/o CG	80.86	13.00	0.497	**30.86**	1.31	14.42	0.161	21.60	0.416	2.1
w/o PRE	80.86	0.31	0.490	26.89	0.42	12.82	**0.191**	27.16	0.487	4.1
w/o PRE & CG	**82.10**	0.62	0.496	26.66	1.23	10.90	0.176	**27.78**	0.481	3.8

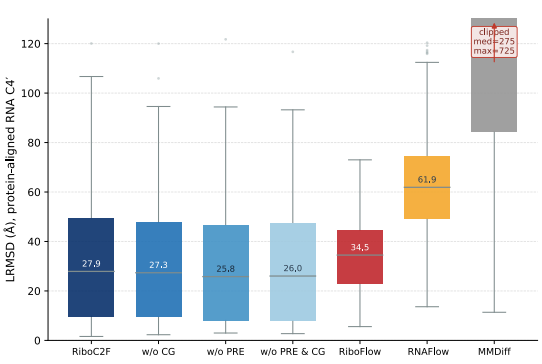


Figure 3: Per-target LRMSD distribution after protein alignment. Medians are annotated.

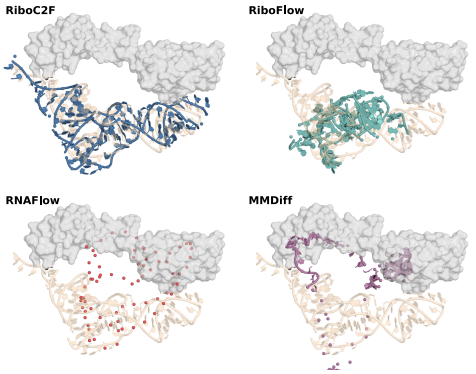


Figure 4: Overlay on PDB 7OJ0. GT protein surface is light grey and GT RNA is beige.

quantifies local geometric fidelity. **SR%**, sequence recovery, the per-residue fraction of nucleotides that match the ground truth. **Clash**, mean count of RNA $C4'$ to protein C_α pairs within 3 Å, counts how often the RNA interpenetrates the protein.

Docking captures whether the RNA binds the protein in the right place and forms an energetically reasonable interface. **LRMSD**, the RNA $C4'$ RMSD against the ground-truth complex, quantifies overall placement error (per-target distribution in Fig. 3). **dG%**, the fraction of complexes whose Rosetta [29] interface ΔG is more favorable than the ground truth's, indicates whether the predicted interface is thermodynamically reasonable (not reported for RNAFlow, which does not output a heavy-atom interface). **Fnat**, the fraction of native heavy-atom contacts recovered within 5 Å, measures how much of the true interface is preserved. **Acc.%**, the rate of complexes meeting the Critical Assessment of PRedicted Interactions (CAPRI) [30] acceptable threshold (DockQ [31] ≥ 0.23), gives the fraction of complexes that are at least usefully docked.

Multi-sample diversity and novelty captures whether stochastic sampling produces a useful design ensemble across $k = 10$ samples per target. **Div**, the per-target product of the validity rate and the qTMclust [32] cluster fraction at TM ≥ 0.45 over valid samples, gives the expected number of distinct folds per sampling attempt. **Nov%**, the per-target fraction of samples that are simultaneously valid and structurally divergent from the training pool (Appendix K), gives the percentage of sampling attempts yielding a valid novel design.

Result: strongest docking with competitive intrinsic quality. We summarize Table 1 along three findings. (i) *The cascade is the active ingredient on the protein-conditioned objective.* Compared with the published baselines, RiboC2F achieves a 5× higher acceptable docking rate and ∼1.7× stronger interface energetics relative to RiboFlow, while maintaining competitive intrinsic RNA quality. The comparison with RiboFlow, which also uses RNA-only pretraining, supports the explicit focus on pose and the coarse-to-fine cascade for addressing the where-to-place problem in protein-conditioned generation. (ii) *RNA-only structure and conditioned structure trade off.* Among the RiboC2F variants, starting from pretrained initialization preserves more of the RNA-only prior in the generated outputs, yielding stronger intrinsic quality; training from scratch sees only the paired-complex data and prefers

complex-bound RNA conformations, yielding stronger docking. CG resolves residual atomic clashes while keeping other metrics essentially unchanged, providing atomistic feasibility at near-zero cost. (iii) *Multi-sample metrics expose the validity–exploration tension.* RiboC2F finds a working point where validity and exploration coexist, exhibiting diversity that does not come from structural collapse and novelty that is not training-fold memorization in disguise.

5.3 Coarse analysis

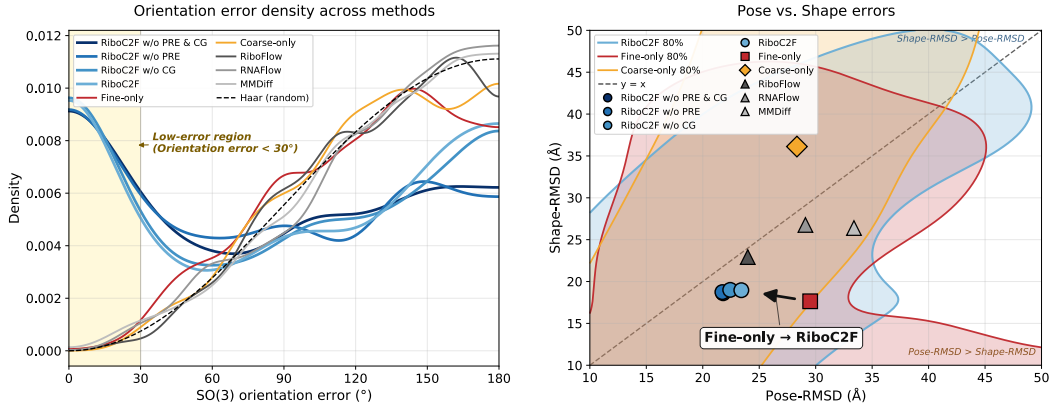
In this subsection we evaluate the pose accuracy of the joint cascade and characterize how the fine stage uses the coarse signal through controlled perturbation experiments.

Table 2: Binding-pose accuracy. RiboC2F can orient the RNA against its protein target.

Method	Ori-Acc % ↑	Ori-Score % ↑	BS-Hit % ↑
MMDiff	2.0	0.1	3.0
RNAFlow	0.8	0.0	3.7
RiboFlow	0.7	0.3	26.4
RiboC2F	24.4	25.8	35.3
w/o CG	**25.3**	27.3	36.2
w/o PRE	<u>25.1</u>	**31.7**	**44.2**
w/o PRE & CG	24.7	30.9	40.4
Random (Haar)	*0.8*	*0.0*	—

Table 3: Pose orientation is the key information in the coarse signal.

Aspect	Perturbation	Coarse RMSD	Fine RMSD
—	GT coarse	0.00	8.47
Noise	Gaussian $\sigma = 10$	17.20	13.38
Sparsity	drop 25% (linear interp.)	14.85	10.06
Sparsity	mask 50%	21.60	16.82
Order	residue shuffle	42.61	38.37
Orientation	reorient 30°	13.19	16.31
Orientation	reorient 90°	35.26	39.22
Orientation	reorient 180°	46.97	51.98



(a) Orientation error density.

(b) Pose vs. Shape errors.

Figure 5: (a) Kabsch orientation-error densities from RNA $C4'$ coordinates, with a Haar-random reference (dashed). The RiboC2F variants assign substantially more probability to near-correct orientations than the Haar baseline. (b) Per-method mean Pose-RMSD and Shape-RMSD, with 80% density contours for RiboC2F, Fine-only, and Coarse-only (definitions in Appendix M). Fold fidelity alone does not ensure accurate protein-relative placement.

Pose evaluation. The cascade's central claim is to redirect paired-complex supervision toward the binding pose, so we evaluate the RNA's overall *orientation* relative to its protein target. We define the orientation error as the SO(3) angle of the optimal Kabsch rotation aligning the separately centred predicted and ground-truth RNA $C4'$ point clouds, and use a Haar-uniform random baseline to anchor no-information performance (Table 2). **Ori-Acc%** reports the percentage of complexes whose orientation error is below 30°, indicating how often the method gets the orientation roughly right. **Ori-Score%** reports the percentage of the mean orientation-error gap closed between the Haar-random baseline (0%) and the ground truth (100%), with scores clipped at zero. Formulas and the distinction from principal-axis-frame training supervision are given in Appendix M. **BS-Hit%** reports the percentage of complexes whose binding-site recall is at least 0.5 at an 8 Å contact threshold, indicating whether the predicted RNA contacts the right binding-pocket residues.

Coarse decomposition. The fine stage receives the coarse stage's predicted positions through a learned projection, and we ask which aspect of those positions it actually uses. We use the RiboC2F

w/o CG configuration on a fixed subset of $N = 20$ test complexes drawn by stratified sampling across five RNA-length buckets ($\leq$ 10, 11–30, 31–60, 61–100, $>$ 100 nt) with per-bucket counts proportional to the full-set distribution and deduplicated by 4-character PDB ID. We apply two kinds of controlled perturbations to the GT coarse signal. Pointwise corruptions test whether the fine stage relies on exact per-residue coordinates: *Gaussian noise* adds isotropic noise of standard deviation σ to every $C4'$ coordinate, *sparse masking* zeroes a random 50% of residue coordinates, and *residue shuffle* permutes the order of residue coordinates along the chain. Rigid reorientation tests whether the fine stage follows the proposed orientation verbatim. **Coarse RMSD** measures the magnitude of the input perturbation, and **Fine RMSD** measures how much of that perturbation reaches the fine stage's output.

Results. (i) *RiboC2F can orient the RNA against its protein target.* Table 2 shows improved orientation accuracy and binding-site recovery across RiboC2F configurations. Fig. 5a further shows substantially greater probability of errors below 30° for RiboC2F than for either single-stage variant. (ii) *RNA shape agreement does not guarantee successful protein conditioning.* Similar LRMSD values can conceal different contributions from RNA shape mismatch and protein-relative orientation and position errors. RiboC2F's 80% contour spreads around the diagonal (Fig. 5b), suggesting a more balanced distribution of pose and shape errors. (iii) *Pose orientation is the key information in the coarse signal.* Table 3 shows that the fine stage can partially compensate for coordinate corruption and residue shuffling, but has limited ability to correct global orientation errors. This supports prioritizing orientation accuracy in the coarse signal while leaving moderate internal geometric errors for the fine stage to resolve.

5.4 Ablation analysis

Each variant in this subsection is matched to RiboC2F on the trunk, the pretrained initialization, the objectives, the router and the evaluation protocol, and changes exactly one design choice; with the two single-stage reductions of Sec. 5.1 as reference points, they separate three questions about the cascade (Table 4). **LATENT** changes what the coarse stage regresses, replacing the RNA translations with a 128-dim gRNAde [13] embedding that the fine stage receives through the same learned projection, and so asks whether the target must be the geometric pose. **Async-Schedule** changes where the coarse–fine split lives, replacing the two-stage factorization with a single-stage flow whose per-modality schedule is offset so that translations denoise ahead of the rest, and so asks whether an ordering in time can substitute for a structural split. **Unshared-Trunk** changes whether the split shares parameters, giving each stage its own trunk so that the fine loss no longer reaches the coarse parameters; it carries roughly twice the trunk parameters at the same per-stage update count, making it a generous rather than a restricted control. All three are trained with PRE on and evaluated with CG off on the full test set, and are therefore compared against RiboC2F (w/o CG). We report lDDT and SR% on the intrinsic-quality side, Fnat and Acc.% on the docking side, and Ori-Acc% and BS-Hit% from Sec. 5.3 on the pose side. For Coarse-only, full-atom coordinates are reconstructed from the predicted translations using identity residue rotations ($\mathbf{R}_i = \mathbf{I}_3$) to compute Fnat and Acc.%. It does not generate a sequence and is therefore omitted from SR%.

Table 4: The two-stage cascade makes a key contribution to the framework's performance gains. **Bold**: best per column.

Configuration	lDDT↑	SR%↑	Fnat↑	Acc.%↑	Ori-Acc%↑	BS-Hit%↑
RiboC2F (w/o CG)	0.497	30.86	**0.161**	21.60	**25.3**	36.2
Unshared-Trunk	0.555	30.1	0.158	**24.4**	21.6	**44.1**
LATENT	0.531	**32.7**	0.096	14.1	1.2	29.7
Async-Schedule	0.513	30.0	0.060	5.9	0.3	26.0
Fine-only	**0.563**	31.85	0.068	6.17	0.4	29.5
Coarse-only	0.219	—	0.049	2.52	0.7	22.5

Results. (i) *The two-stage structure contributes to docking.* LATENT's docking advantage over Fine-only supports a benefit from the cascade even without a geometric coarse target. Async-Schedule fails to recover this benefit, indicating that changing the denoising schedule cannot substitute for the structural split. (ii) *The geometric coarse target enables accurate orientation.* LATENT retains docking benefits but loses orientation accuracy, revealing the additional contribution of explicit

geometric supervision. Coarse-only's weak orientation further indicates that this supervision is effective within the complete cascade. (iii) *Trunk sharing realizes the cascade's gains with fewer parameters.* Compared with Unshared-Trunk, our model maintains strong docking and orientation performance under matched per-stage update counts, demonstrating that these benefits can be achieved without increasing the trunk parameter budget.

6 Conclusion and Limitations

We propose RiboC2F, a supervision-efficient flow-matching framework for protein-conditioned RNA co-design. The core idea is to concentrate scarce paired-complex supervision on the global binding pose through a coarse-to-fine cascade, allocating the conditional signal where it matters most. The coarse stage proposes the RNA's binding pose against the protein, and the fine stage generates the bound RNA from its own noise prior, conditioned on an embedding of the proposed pose. Both stages share a single IPA trunk with stage-specific time variables and output heads. Comprehensive experiments across docking accuracy, intrinsic RNA quality, and multi-sample diversity show that RiboC2F delivers the strongest docking with competitive intrinsic quality. Probing confirms that pose orientation is the operative component of the coarse signal, while ablations show that the two-stage cascade makes a key contribution to the framework's performance gains.

The current work is limited to single-RNA single-protein complexes within a 200-residue length cap; longer RNAs, multi-chain assemblies, ligand co-conditioning, and broader ribonucleoprotein settings are natural extensions of the pose-concentrated allocation, and we expect the pose-concentrated principle to generalize to other conditioned biomolecular design tasks where paired-complex supervision is scarce.

Acknowledgments and Disclosure of Funding

This work was supported by the Science and Technology Commission of Shanghai Municipality (grants No. 24510714300), and the Fundamental Research Funds for the Central Universities (project number YG2026QNB01), and the Shanghai Municipal Science and Technology Major Project, China (Grant No. 2021SHZDZX0102).

The authors declare no competing interests.

References

[1] Jiehua Zhou and John Rossi. Aptamers as targeted therapeutics: current potential and challenges. *Nature reviews Drug discovery*, 16(3):181–202, 2017.

[2] 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.

[3] 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.

[4] 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*, 21(12):2287–2298, 2024.

[5] Joseph L. Watson, David Juergens, Nicholas R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Wang, Koen Horst, Regina Barzilay, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620:1089–1100, 2023.

[6] Rishabh Anand, Chaitanya K. Joshi, Alex Morehead, Arian Rokkum Jamasb, Charles Harris, Simon V Mathis, Kieran Didi, Rex Ying, Bryan Hooi, and Pietro Lio. RNA-FrameFlow: Flow matching for de novo 3D RNA backbone design. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=wOc1Yx5s09`.

[7] Ranjit Prasad Bahadur, Martin Zacharias, and Joël Janin. Dissecting protein–RNA recognition sites. *Nucleic Acids Research*, 36(8):2705–2716, 2008.

[8] James R. Williamson. Induced fit in RNA–protein recognition. *Nature Structural Biology*, 7 (10):834–837, 2000.

[9] Runze Ma, Zhongyue Zhang, Zichen Wang, Chenqing Hua, Jiahua Rao, Zhuomin Zhou, and Shuangjia Zheng. RiboFlow: Conditional de novo RNA co-design via synergistic flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[10] Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, and Shuiwang Ji. A joint diffusion model with pre-trained priors for RNA sequence-structure co-design. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[11] Divya Nori and Wengong Jin. RNAFlow: RNA structure & sequence design via inverse folding-based flow matching. In *International Conference on Machine Learning*, pages 38395–38408. PMLR, 2024.

[12] 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.

[13] Chaitanya K Joshi and Pietro Lio. gRNAde: A geometric deep learning pipeline for 3D RNA inverse design. In *RNA Design: Methods and Protocols*, pages 121–135. Springer, 2024.

[14] Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using LigandMPNN. *Nature Methods*, 22(4):717–723, 2025.

[15] Alex Morehead, Aadyot Bhatnagar, Jeffrey A. Ruffolo, and Ali Madani. Towards joint sequence-structure generation of nucleic acid and protein complexes with SE(3)-discrete diffusion. In *NeurIPS Machine Learning in Structural Biology Workshop*, 2023.

[16] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

[17] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[19] Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research (TMLR)*, 2024.

[20] Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se(3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning (ICML)*, 2023.

[21] Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noe. Fast protein backbone generation with SE(3) flow matching. In *International Conference on Machine Learning (ICML)*, 2024.

[22] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.04997.

[23] John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021.

[24] Rong Han, Xiaohong Liu, Tong Pan, Jing Xu, Xiaoyu Wang, Wuyang Lan, Zhenyu Li, Zixuan Wang, Jiangning Song, Guangyu Wang, et al. CoPRA: Bridging cross-domain pretrained sequence models with complex structures for protein–RNA binding affinity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 246–254, 2025.

[25] Chengxin Zhang, Xi Zhang, Peter L. Freddolino, and Yang Zhang. BioLiP2: An updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, 52(D1):D404–D412, 2024.

[26] Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025.

[27] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[28] Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lDDT: A local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

[29] Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, Jason W. Labonte, Michael S. Pacella, Richard Bonneau, Philip Bradley, Roland L. Dunbrack, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017.

[30] Raúl Méndez, Raphaël Leplae, Leonardo De Maria, and Shoshana J. Wodak. Assessment of blind predictions of protein-protein interactions: Current status of docking methods. *Proteins: Structure, Function, and Bioinformatics*, 52(1):51–67, 2003.

[31] Sankar Basu and Björn Wallner. DockQ: A quality measure for protein-protein docking models. *PLOS ONE*, 11(8):e0161879, 2016.

[32] Yang Zhang and Jeffrey Skolnick. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

[33] Zeming Lin et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379:1123–1130, 2023.

[34] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. ColabFold: Making protein folding accessible to all. *Nature Methods*, 19: 679–682, 2022.

[35] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35:1026–1028, 2017.

[36] Chai Discovery. Chai-1: Decoding the molecular interactions of life. *bioRxiv*, 2024. bioRxiv 2024.10.10.617227.

[37] Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. US-align: Universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature Methods*, 19:1109–1115, 2022.

A Full architecture

Trunk. RiboC2F's trunk is a stack of $B = 6$ Invariant Point Attention [23] blocks with 256-dimensional node embeddings, 128-dimensional edge embeddings, 8 attention heads, and 8 query / 12 value points. Each block applies an IPA layer, a node MLP, and an edge update; we use the FrameFlow trunk design [21] without modification.

Heads. The fine stage carries four output heads operating on the final RNA node embeddings $\mathbf{h}_i^R$: a translation head (linear, $\mathbb{R}^{256} \to \mathbb{R}^3$), a rotation head (linear, $\mathbb{R}^{256} \to \mathbb{R}^3$, interpreted as a rotation-vector update to the current frame), the AngleResnet of Anand et al. [6] for the eight torsion angles, and a linear classifier over the five-element extended alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}, [\mathrm{MASK}]\}$. The coarse stage uses only the translation head.

B Featurization details

At every training and sampling step, we form a single node set of size $L_R + L_P$ by concatenating RNA and protein along the sequence dimension. A flow mask $m_i \in \{0, 1\}$ identifies which nodes are being denoised. *RNA nodes* ($m_i = 1$) carry per-residue frames $\mathbf{F}_i^R = (\mathbf{R}_i, \mathbf{x}_i)$ that are noised by the flow, and node features built from a nucleotide embedding (replaced by a $[\mathrm{MASK}]$ token when s_i is noised), a sinusoidal positional embedding of the residue index, and the dual timestep embeddings. *Protein nodes* ($m_i = 0$) carry ground-truth frames held fixed throughout generation; their node features are a linear projection of frozen ESM2-t33-650M [33] embeddings from 1280 to $d_s = 256$ dimensions. Edge features are computed over the full $(L_R + L_P)^2$ graph as two sets of radial basis function (RBF) distance features: a short-range set on $[0, 20]$ Å targeting local base pairing and stacking, and a long-range set on $[0, 200]$ Å targeting overall complex layout. Ligand atoms, when present, are appended to the same node set with a 21-class atom-type embedding and an RBF of RNA–ligand distances on $[2, 25]$ Å. Velocity heads read out only from nodes with $m_i = 1$.

C Per-modality flow details

This section gives the explicit interpolant equations, prior choices, and sampling procedures for the modalities of RiboC2F that are modeled by flow matching, summarized in Section 4.1. The training-time distribution over t, denoted p_T, is specified in Appendix F, and loss equations are given in the main text (Eqs. (2), (3), (6)).

Translation. The CFM linear interpolant on $\mathbb{R}^3$ has the form

$$\mathbf{x}_t^{(i)} = (1-t)\,\mathbf{x}_0^{(i)} + t\,\mathbf{x}_1^{(i)}, \qquad u_t^{\mathrm{trans}}\big(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_1^{(i)}, \mathbf{x}_0^{(i)}\big) = \mathbf{x}_1^{(i)} - \mathbf{x}_0^{(i)}, \tag{13}$$

with prior $\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbf{I}_3)$ at $\sigma_x = 25$ Å, chosen to match the spatial extent of RNA. To stabilize training, we follow Tong et al. [19] in pairing $(\mathbf{x}_0, \mathbf{x}_1)$ within each mini-batch by Kabsch-aligned optimal transport.

Rotation. The CFM unit-speed geodesic interpolant on $\mathrm{SO}(3)$ has the form

$$\mathbf{R}_t^{(i)} = \exp_{\mathbf{R}_0^{(i)}}\!\big(t \cdot \log_{\mathbf{R}_0^{(i)}} \mathbf{R}_1^{(i)}\big), \qquad u_t^{\mathrm{rot}}\big(\mathbf{R}_t^{(i)} \,|\, \mathbf{R}_1^{(i)}, \mathbf{R}_0^{(i)}\big) = \frac{\log_{\mathbf{R}_t^{(i)}} \mathbf{R}_1^{(i)}}{1-t}, \tag{14}$$

where $\exp_R$ and $\log_R$ are the SO(3) Riemannian exponential and logarithm at R, with body-frame representation $\log_{\mathbf{R}_t} \mathbf{R}_1 = \log(\mathbf{R}_t^\top \mathbf{R}_1)$. The prior $\mathbf{R}_0^{(i)}$ is drawn from $\mathrm{IG}_{\mathrm{SO}(3)}(\sigma = 1.5)$, the isotropic Gaussian on $\mathrm{SO}(3)$ at large scale that is effectively uniform.

Residue type. At inference, residue types are sampled by iterative continuous-time Markov-chain unmasking with rates derived from $\hat{\mathbf{p}}_{\boldsymbol{\theta}}$, following Campbell et al. [22].

D Auxiliary coarse losses

The three auxiliary terms in $\mathcal{L}_{\mathrm{coarse}}$ (Eq. (8)) are defined as follows. Let $\hat{\mathbf{Z}} \in \mathbb{R}^{L_R \times 3}$ denote the predicted RNA $C4'$ point cloud, $\mathbf{Z}_1$ the ground truth, both restricted to residues in the loss mask, and $\mathbf{X}_P^a \in \mathbb{R}^{N_P^a \times 3}$ the protein heavy atoms (with $N_P^a = L_P \cdot N_a^P$ total atoms).

Orientation $\mathcal{L}_{\mathrm{ori}}$. Let $R(\mathbf{Z}) \in \mathrm{SO}(3)$ be the orthonormal frame whose columns are the principal axes of the centred point cloud $\mathbf{Z}$, computed from the eigen-decomposition of its 3×3 covariance matrix. Because eigenvectors are unique only up to a sign and an overall reflection, we resolve the ambiguity deterministically by (i) ordering the columns by descending eigenvalue, (ii) flipping each column so that its largest-magnitude component is positive, and (iii) flipping the last column whenever $\det R(\mathbf{Z}) < 0$ to enforce a right-handed frame. Both $R(\hat{\mathbf{Z}})$ and $R(\mathbf{Z}_1)$ pass through this normalization before the residual is computed, and the loss is only evaluated when the masked point cloud has at least three valid residues:

$$\mathcal{L}_{\mathrm{ori}} = \big\| R(\hat{\mathbf{Z}})^\top R(\mathbf{Z}_1) - \mathbf{I}_3 \big\|_F^2. \tag{15}$$

This principal-axis-frame residual is an auxiliary training loss. The reported orientation metrics instead use the Kabsch alignment angle defined in Appendix M. When the leading eigenvalues are nearly degenerate (near-spherical point clouds), the deterministic axis assignment can become unstable; we found this regime to be rare on the RNA backbones in our training data and accept the residual sensitivity in this auxiliary training objective.

Interface $\mathcal{L}_{\text{iface}}$. For each RNA residue i, let $d^{\text{gt}}_{i,j} = \|\mathbf{Z}_{1,i} - \mathbf{X}^a_{P,j}\|$ and define the interface indicator $\mathbb{1}[\min_j d^{\text{gt}}_{i,j} < 10\,\text{Å}]$, i.e. the $C4'$ of residue i lies within 10 Å of any protein heavy atom. Let $w_i = 1 + (\kappa_{\text{if}} - 1) \cdot \mathbb{1}[\cdots]$ with $\kappa_{\text{if}} = 10$, so interface residues are weighted by 10 and the rest by 1. The interface loss is the weighted mean over residues of the squared error in RNA–protein pairwise distances,

$$\mathcal{L}_{\text{iface}} = \frac{\sum_i w_i\, e_i}{\sum_i w_i}, \qquad e_i = \frac{1}{N^a_P} \sum_{j=1}^{N^a_P} \big(\hat{d}_{i,j} - d^{\text{gt}}_{i,j}\big)^2, \tag{16}$$

with $\hat{d}_{i,j} = \|\hat{\mathbf{Z}}_i - \mathbf{X}^a_{P,j}\|$.

Pairwise distance $\mathcal{L}_{\text{pd}}$. The pairwise-distance loss is the squared error between predicted and ground-truth RNA–RNA pairwise distances, averaged over all valid residue pairs,

$$\mathcal{L}_{\text{pd}} = \frac{1}{|S|^2} \sum_{i,j \in S} \big(\|\hat{\mathbf{Z}}_i - \hat{\mathbf{Z}}_j\| - \|\mathbf{Z}_{1,i} - \mathbf{Z}_{1,j}\|\big)^2, \tag{17}$$

where S is the set of valid residue indices.

Weights. We use $(\lambda_{\text{ori}}, \lambda_{\text{if}}, \lambda_{\text{pd}}) = (10^{-2}, 2 \times 10^{-3}, 5 \times 10^{-4})$ in all reported runs.

E Fine-stage loss weights

The fine objective $\mathcal{L}_{\text{fine}}$ in Eq. (10) uses the per-modality weights $\lambda_{\text{trans}} = 2.0$, $\lambda_{\text{rot}} = 1.0$, $\lambda_{\text{tors}} = 1.0$, and $\lambda_{\text{type}} = 2.0$ in all reported runs. The continuous translation and rotation losses use endpoint prediction with the clipped time-normalization factor $[1 - \min(t, t_{\text{clip}})]^{-2}$, where $t_{\text{clip}} = 0.9$. The torsion loss is unnormalized, and the type loss has no time reweighting.

F Training schedule and inference hyperparameters

We list the values used to instantiate the training-time time distribution p_T, the cascade alternation, and the inference Euler integrator:

Quantity	Value
Time clip ϵ	0.01 (so $t \in [0.01, 0.99]$)
Coarse time $t_{\text{c}} \sim$ LogitNormal	$\mu = -1.0$, $\sigma = 1.0$
Fine time $t_{\text{f}} \sim$ LogitNormal	$\mu = 0.0$, $\sigma = 1.0$
Fine extra low-t mass	10% probability replaced by $\text{Uniform}(0.01, 0.5)$
Loss-normalization clip t_{clip}	0.9 (trans/rot denominator $[1 - \min(t, t_{\text{clip}})]^2$)
Bernoulli stage probability p_{coarse}	0.6 (training); $p_{\text{fine}} = 0.4$
Surrogate-pose noise β	$\sim \text{Uniform}(0, 0.25)$
Inference NFE (Number of Function Evaluations)	50 Euler steps shared between the two stages
Inference coarse fraction	0.5 of the integration budget
Translation noise scale σ_x	25 Å (matched to the spatial extent of the training RNAs)
Rotation prior	$\text{IG}_{\text{SO}(3)}(\sigma = 1.5)$; sampling exponential with rate 2
Type sampler temperature	0.4 (CTMC sampling, with noise = 10)

G Clash guidance: algorithm and design notes

Algorithm. Per fine sampling step, with current $C4'$ translations $\mathbf{z}_t \in \mathbb{R}^{B \times L_R \times 3}$ and protein C_α coordinates $\mathbf{x}^P \in \mathbb{R}^{B \times L_P \times 3}$:

1. compute $\lambda \leftarrow \lambda_0\, t$ and skip if λ is below a numerical floor;
2. enable autograd on $\mathbf{z}_t$;
3. form the pairwise distance matrix $D = \mathrm{cdist}(\mathbf{z}_t, \mathbf{x}^P) \in \mathbb{R}^{B\times L_R\times L_P}$;
4. compute the violation energy E as in Eq. (11);
5. obtain the gradient $\mathbf{g} = \partial E/\partial \mathbf{z}_t$ by autograd;
6. apply the update of Eq. (12).

The same step is applied after the final clean prediction at $t=1$ to remove residual overlaps from the integrated trajectory.

Design notes. CG only updates RNA translations, leaving rotations, torsions, and sequence logits intact, so the geometric predictor's learned SE(3) consistency is preserved. The protein is held as a stop-gradient constant since it is given as the conditional input rather than a quantity to be redesigned. The mask m ensures that flow-masked padding RNA residues are not pushed away from spurious distances, and that protein-side padding does not contribute repulsion. The linear time schedule $\lambda(t) = \lambda_0 t$ encodes the design intent that early in sampling the structure is too noisy to apply rigid repulsion meaningfully, while late in sampling repulsion focuses purely on clearing local hard overlaps.

Complexity. CG introduces no trainable parameters. Each step adds an $O(L_R L_P)$ pairwise distance computation and one corresponding backward pass; relative to a single IPA forward pass through the trunk, the additional cost is negligible.

Choice of the $3\,$Å clash threshold. The Clash metric in §5.2 uses a $C4'$-to-$C\alpha$ heavy-atom threshold of $\tau = 3.0\,$Å. We chose this value empirically from the training set: more than 99% of ground-truth complexes contain no $C4'$–$C\alpha$ pair below $3\,$Å, so any predicted pair below this distance is a configuration unseen in the data and can be safely flagged. We deliberately do not use a stricter $2.5\,$Å van-der-Waals cutoff since hydrogen-bond and salt-bridge contacts already populate the 2.7–$3.0\,$Å range; our threshold is calibrated to flag overlaps that exceed even the tightest physiological contacts in the data.

H RNA structural validity (Val)

We evaluate RNA structural validity using a composite metric that combines three independent geometric checks. A sample passes the Val gate if and only if it passes all three.

(a) Backbone integrity (Break check). For an L_R-residue RNA, let $d_i = \|\mathbf{x}_{i+1}^{C4'} - \mathbf{x}_i^{C4'}\|$ for $i = 1, \ldots, L_R - 1$, and define the per-sample break fraction

$$\text{break_frac} \;=\; \frac{|\{\, i : d_i > 8\,\text{Å}\,\}|}{L_R - 1}. \tag{18}$$

A sample passes the Break check if $\text{break_frac} \le \tau_b$ with $\tau_b = 0.34$.

(b) Steric integrity (Clash check). Let $\delta = \min_{|i-j|\ge 2} \|\mathbf{x}_i^{C4'} - \mathbf{x}_j^{C4'}\|$ be the minimum non-adjacent $C4'$–$C4'$ distance. A sample passes if $\delta \ge 2.0\,$Å.

(c) Compactness (Rg check). Let R_g^{pred} and R_g^{GT} be the radii of gyration over predicted and ground-truth $C4'$ coordinates. A sample passes if $0.7 \le R_g^{\text{pred}}/R_g^{\text{GT}} \le 1.4$.

Why a fraction-based break check. The canonical adjacent $C4'$–$C4'$ distance in folded RNA is $\sim 5.9\,$Å, and a strict "any $d_i > 8\,$Å" criterion would fail 23.4% of ground-truth crystal structures because of natural long-bond outliers in bulges, multi-way junctions, and partially disordered loops. The per-sample fraction is robust to such single-bond outliers while still detecting catastrophic chain rupture.

Threshold calibration on ground truth. All three thresholds are set so that 100% of the ground-truth crystal structures pass, removing threshold-engineering risk. Both Clash ($\delta \geq 2.0$ Å) and Rg ($R_g^{\text{pred}}/R_g^{\text{GT}} \in [0.7, 1.4]$) hold trivially on GT (Rg is identically 1 for GT).

Component-wise breakdown. Table 5 reports the per-component pass rates. All non-trivial models pass the Break check at $\geq 99.6\%$, confirming catastrophic chain rupture is rare for any trained generator on this task; the discriminative power of Val comes from Clash and Rg. RiboFlow's predominant failure mode is collapse: among samples that fail the Rg check, 26.5% of RiboFlow samples have $R_g^{\text{pred}}/R_g^{\text{GT}} < 0.7$ versus 3.7–8.3% for RiboC2F variants. MMDiff fails in both directions due to a near-total loss of structural prior.

Table 5: Component-wise Val breakdown. Break, Clash, Rg are pass rates (%); Val is the conjunction.

Method	Break %	Clash %	Rg %	Val %
GT (reference)	100.00	100.00	100.00	100.00
RiboC2F	100.00	91.98	87.96	81.48
w/o CG	100.00	91.67	87.65	80.86
w/o PRE	100.00	90.12	88.89	80.86
w/o PRE & CG	100.00	90.43	90.12	82.10
Fine-only	100.00	90.12	88.89	79.32
RiboFlow	100.00	59.57	70.37	45.06
RNAFlow	100.00	93.49	79.79	73.63
MMDiff	99.69	54.49	63.16	27.86

Threshold sensitivity. Sweeping $\tau_b \in \{0.05, 0.10, 0.15, 0.20, 0.34\}$ leaves the model ranking on Val unchanged; varying τ_b produces only a uniform 1–3 pp shift of the Break component, with a 5% drop in GT pass rate at the strictest setting.

I Self-consistency oracle: pipeline and oracle choice

Pipeline. For each test complex, we (i) build a Boltz-2 input YAML containing the GT protein chain and our generated RNA sequence; (ii) reuse a pre-fetched protein multiple sequence alignment (MSA) generated by ColabFold [34] on top of MMseqs2 [35] from a local cache (avoiding inference-time rate-limit stalls); (iii) run Boltz-2 with $\texttt{recycling_steps}=3$, $\texttt{sampling_steps}=200$, $\texttt{diffusion_samples}=1$, on 3 GPUs in parallel ($\sim$ 30–60 s/sample with MSA cache, $\sim$ 1 min/sample on cold MSA); (iv) extract the RNA chain from the Boltz-2 output PDB, align it to the generated RNA chain via `USalign A.pdb B.pdb -mol RNA`, and report scTM as the maximum of the two length-normalized TM-scores together with the aligned $C4'$ scRMSD reported by US-align.

Why Boltz-2 over AF3 and RF2NA. We ruled out RoseTTAFold2NA [3] after a paired sanity check on $n=5$ representative complexes: RF2NA fails to even reproduce the GT protein backbone, returning a mean protein-chain $C\alpha$ RMSD of $\sim$16 Å versus $\sim$2 Å for Boltz-2 on the same inputs, so it cannot serve as a reliable oracle in a protein-conditioned setting. Between Boltz-2 and AlphaFold3 (AF3) [2], we ran a head-to-head best-of-5 panel on $n=7$ representative complexes spanning $\sim$20–120 nt RNA; the table below reports the per-complex oracle (max across the 5 predictions) averaged across the panel:

Oracle	best-TM (mean)	best-iPTM	wall-clock per sample
Boltz-2	$\mathbf{0.249}$	reliable	$\sim$60 s (MSA cached)
AF3 (5 seeds)	0.239	$\leq$0.07 on $5/7$ samples	$\sim$63 min

The two oracles return structurally comparable best-of-5 TM, but AF3's iPTM collapses to near-zero on $5/7$ complexes, indicating that its chain-level confidence is unreliable in the small-RNA / weak-MSA regime that dominates our test set. AF3 is also roughly two orders of magnitude slower per sample (the bulk in the MSA pipeline). Boltz-2 is therefore selected as the SC oracle: comparable structural fidelity to AF3, more reliable confidence signal, and tractable cost. We also considered Chai-1 [36] (similar runtime to AF3 with no obvious gain on RNA–protein) and ruled it out.

Reliability bands and a broader caveat. We calibrate the self-consistency oracle with a GT-fold control on the test set: the ground-truth RNA sequence and protein chain are supplied to Boltz-2, and the predicted RNA is compared with the ground-truth structure using US-align, following the same pipeline and TM-score threshold as the scTM evaluation.

RNA length	N	GT-fold TM (mean)	Median	TM ≥ 0.45 (%)
≤ 40 nt	106	0.351	0.288	29.2
41–80 nt	74	0.438	0.306	39.2
>80 nt	144	0.611	0.638	63.9
All		0.486	0.414	46.9

The ground-truth sequence reaches TM-score ≥ 0.45 on only 46.9% of test complexes, which we denote as the `oracle_ok` subset. The remaining targets form `oracle_fail`. Absolute scTM rates should therefore be interpreted against this GT-fold reference success rate and its dependence on RNA length. A generated sample passing the scTM threshold on an `oracle_fail` target is not oracle-validated. Using the same oracle for every method does not by itself establish that cross-method rankings are unaffected by oracle failures.

J Statistical significance

Bootstrap confidence intervals. For each $(method, metric)$ cell of Table 1, we resample the test complexes $1{,}000$ times with replacement, paired across methods so that all methods see the same resamples (`numpy` default RNG, seed 42). Continuous metrics aggregate as the per-target mean and rate metrics as the per-target proportion, identical to the main table. We report the 2.5% / 97.5% percentile bootstrap interval as the 95% confidence interval. Cells where the metric is undefined for a method (e.g. RNAFlow's ΔG and Clash) are marked "—".

Table 6: Bootstrap 95% CIs for the intrinsic-quality columns of Table 1. Format: mean [lower, upper]. RNAFlow's Clash is reported in Table 1 but its RNA is re-folded by RF2NA in RNAFlow's own predicted protein frame rather than placed against the GT receptor used by all other methods, so its Clash is not on a comparable coordinate reference and we mark it "—" here to avoid implying frame-aware comparability.

Method	Val%↑	scTM%↑	lDDT↑	SR%↑	Clash↓
MMDiff	27.9 [22.7, 32.8]	0.3 [0.0, 0.9]	0.290 [0.278, 0.301]	18.8 [17.5, 20.2]	0.61 [0.42, 0.83]
RNAFlow	73.6 [68.2, 78.5]	0.0 [0.0, 0.0]	0.422 [0.403, 0.442]	26.5 [25.2, 27.8]	—
RiboFlow	45.1 [39.5, 50.3]	0.3 [0.0, 0.9]	0.483 [0.466, 0.501]	26.8 [25.7, 28.1]	1.31 [1.04, 1.58]
RiboC2F	81.5 [77.2, 85.8]	14.6 [10.8, 18.3]	0.499 [0.480, 0.519]	30.1 [28.5, 31.7]	0.38 [0.27, 0.51]
w/o CG	80.9 [76.2, 84.9]	13.0 [9.6, 17.0]	0.497 [0.478, 0.516]	30.9 [29.3, 32.4]	1.31 [1.09, 1.55]
w/o PRE	80.9 [76.5, 84.9]	0.3 [0.0, 0.9]	0.490 [0.474, 0.507]	26.9 [25.6, 28.3]	0.42 [0.31, 0.56]
w/o PRE & CG	82.1 [77.8, 86.1]	0.6 [0.0, 1.5]	0.496 [0.480, 0.512]	26.7 [25.4, 28.0]	1.23 [1.01, 1.45]
Fine-only	79.3 [74.7, 83.6]	25.1 [20.7, 29.7]	0.563 [0.543, 0.583]	31.8 [30.1, 33.8]	1.35 [1.08, 1.66]

Table 7: Bootstrap 95% CIs for the docking columns of Table 1 together with LRMSD. Format: mean [lower, upper].

Method	dG%↑	Fnat↑	DockQ↑	Acc.%↑	LRMSD↓
MMDiff	1.6 [0.3, 3.2]	0.016 [0.009, 0.023]	0.014 [0.010, 0.019]	0.6 [0.0, 1.5]	266.2 [248.8, 282.9]
RNAFlow	—	0.002 [0.001, 0.003]	0.015 [0.013, 0.017]	0.0 [0.0, 0.0]	63.2 [60.7, 65.7]
RiboFlow	11.5 [8.4, 15.2]	0.054 [0.041, 0.068]	0.051 [0.044, 0.059]	4.3 [2.2, 6.5]	35.5 [34.0, 37.1]
RiboC2F	19.2 [15.0, 23.2]	0.160 [0.136, 0.187]	0.125 [0.109, 0.142]	21.6 [17.3, 25.9]	33.8 [31.2, 36.4]
w/o CG	14.4 [10.6, 18.6]	0.161 [0.135, 0.187]	0.127 [0.110, 0.143]	21.6 [17.3, 26.2]	32.8 [30.3, 35.2]
w/o PRE	12.8 [9.2, 16.4]	0.191 [0.162, 0.218]	0.142 [0.123, 0.161]	27.2 [22.5, 32.1]	31.9 [29.7, 34.3]
w/o PRE & CG	10.9 [7.7, 14.5]	0.176 [0.149, 0.203]	0.137 [0.119, 0.155]	27.8 [23.1, 32.4]	31.7 [29.6, 34.1]
Fine-only	10.9 [7.7, 14.3]	0.068 [0.052, 0.085]	0.057 [0.048, 0.068]	6.2 [3.7, 9.0]	38.5 [36.6, 40.5]

Pairwise tests against RiboC2F. For each metric, we test the default RiboC2F against every baseline and ablation. Continuous metrics use the paired Wilcoxon signed-rank test; binary outcomes (Val pass, scTM ≥ 0.45, DockQ ≥ 0.23, and "ΔG more favorable than GT") use McNemar's test. Pairing is on shared targets after dropping rows where either side is undefined. We report the per-target mean delta $\Delta =$ RiboC2F $-$ baseline and two-sided p-values, without multiple-comparison correction. Cases where pairing is impossible are marked "n/a". Specifically, RNAFlow's RNA is re-folded by RF2NA in its own predicted protein frame rather than placed against the GT receptor used by all other methods, so its Clash and $\Delta G\%$ values are not on the same coordinate reference as the rest of the table and we exclude them from paired comparison.

Table 8: Paired tests of the default RiboC2F against each baseline / ablation on intrinsic-quality metrics. Continuous metrics use the paired Wilcoxon signed-rank test, and binary outcomes (Val pass, scTM ≥ 0.45) use McNemar's test.

Metric	Baseline	Δ	p	Test
Val%	MMDiff	$+53.9$	$< 10^{-4}$	McNemar
	RNAFlow	$+6.5$	0.073	McNemar
	RiboFlow	$+36.4$	$< 10^{-4}$	McNemar
	w/o CG	$+0.6$	0.874	McNemar
	w/o PRE	$+0.6$	0.902	McNemar
	w/o PRE & CG	-0.6	0.899	McNemar
	Fine-only	$+2.2$	0.476	McNemar
scTM%	MMDiff	$+14.3$	$< 10^{-4}$	McNemar
	RNAFlow	$+15.5$	$< 10^{-4}$	McNemar
	RiboFlow	$+14.2$	$< 10^{-4}$	McNemar
	w/o CG	$+1.5$	0.486	McNemar
	w/o PRE	$+14.2$	$< 10^{-4}$	McNemar
	w/o PRE & CG	$+13.9$	$< 10^{-4}$	McNemar
	Fine-only	-10.5	$< 10^{-4}$	McNemar
lDDT	MMDiff	$+0.208$	$< 10^{-4}$	Wilcoxon
	RNAFlow	$+0.040$	0.042	Wilcoxon
	RiboFlow	$+0.016$	0.78	Wilcoxon
	w/o CG	$+0.002$	0.85	Wilcoxon
	w/o PRE	$+0.009$	0.098	Wilcoxon
	w/o PRE & CG	$+0.003$	0.84	Wilcoxon
	Fine-only	-0.064	$< 10^{-4}$	Wilcoxon
SR%	MMDiff	$+11.2$	$< 10^{-4}$	Wilcoxon
	RNAFlow	$+2.1$	0.059	Wilcoxon
	RiboFlow	$+3.3$	$< 10^{-4}$	Wilcoxon
	w/o CG	-0.8	0.16	Wilcoxon
	w/o PRE	$+3.2$	$< 10^{-4}$	Wilcoxon
	w/o PRE & CG	$+3.4$	$< 10^{-4}$	Wilcoxon
	Fine-only	-1.8	0.077	Wilcoxon
Clash	MMDiff	-0.23	0.25	Wilcoxon
	RNAFlow	—	—	n/a
	RiboFlow	-0.93	$< 10^{-4}$	Wilcoxon
	w/o CG	-0.94	$< 10^{-4}$	Wilcoxon
	w/o PRE	-0.04	0.62	Wilcoxon
	w/o PRE & CG	-0.85	$< 10^{-4}$	Wilcoxon
	Fine-only	-0.98	$< 10^{-4}$	Wilcoxon

K Dataset details

Filtering pipeline. Our experimental partition is derived from the PRI30k release as follows:

- PRI30k raw release: $34{,}404$ structures.
- Successful parse to RNA-only structures: $12{,}938$ (an intermediate parsing artifact only).

Table 9: Paired tests of the default RiboC2F against each baseline / ablation on docking metrics, following the same protocol as Table 8. Binary outcomes here are DockQ ≥ 0.23 and "ΔG more favorable than GT".

Metric	Baseline	Δ	p	Test
Fnat	MMDiff	$+0.145$	$< 10^{-4}$	Wilcoxon
	RNAFlow	$+0.145$	$< 10^{-4}$	Wilcoxon
	RiboFlow	$+0.106$	$< 10^{-4}$	Wilcoxon
	w/o CG	-0.001	0.83	Wilcoxon
	w/o PRE	-0.030	0.013	Wilcoxon
	w/o PRE & CG	-0.015	0.17	Wilcoxon
	Fine-only	$+0.092$	$< 10^{-4}$	Wilcoxon
dG%	MMDiff	$+17.6$	$< 10^{-4}$	McNemar
	RNAFlow	—	—	n/a
	RiboFlow	$+7.7$	4.0×10^{-3}	McNemar
	w/o CG	$+4.8$	0.050	McNemar
	w/o PRE	$+6.4$	0.025	McNemar
	w/o PRE & CG	$+8.3$	1.5×10^{-3}	McNemar
	Fine-only	$+8.3$	1.8×10^{-3}	McNemar
DockQ	MMDiff	$+0.111$	$< 10^{-4}$	Wilcoxon
	RNAFlow	$+0.101$	$< 10^{-4}$	Wilcoxon
	RiboFlow	$+0.074$	$< 10^{-4}$	Wilcoxon
	w/o CG	-0.001	0.20	Wilcoxon
	w/o PRE	-0.017	0.011	Wilcoxon
	w/o PRE & CG	-0.012	0.93	Wilcoxon
	Fine-only	$+0.068$	$< 10^{-4}$	Wilcoxon
Acc.%	MMDiff	$+21.1$	$< 10^{-4}$	McNemar
	RNAFlow	$+20.2$	$< 10^{-4}$	McNemar
	RiboFlow	$+17.3$	$< 10^{-4}$	McNemar
	w/o CG	0.0	1.00	McNemar
	w/o PRE	-5.6	2.7×10^{-3}	McNemar
	w/o PRE & CG	-6.2	5.2×10^{-3}	McNemar
	Fine-only	$+15.4$	$< 10^{-4}$	McNemar
LRMSD	MMDiff	-232	$< 10^{-4}$	Wilcoxon
	RNAFlow	-28.6	$< 10^{-4}$	Wilcoxon
	RiboFlow	-1.7	0.26	Wilcoxon
	w/o CG	$+1.0$	0.40	Wilcoxon
	w/o PRE	$+1.9$	5.3×10^{-3}	Wilcoxon
	w/o PRE & CG	$+2.1$	1.2×10^{-3}	Wilcoxon
	Fine-only	-4.7	$< 10^{-4}$	Wilcoxon

- Pairing into RNA–protein complexes: 19,344.
- Length filter $L_R, L_P \leq 200$: 11,838.
- Affinity label available: 9,567.
- Sequence-based partitioning and leakage filtering follow Sec. 5.1.

The unconditional RNA pretraining initialization (PRE) used by RiboC2F is taken from the public RNA-FrameFlow checkpoint, which is trained on RNAsolo and is independent of the PRI30k-derived pool above.

Length filtering. The hard filter $L_R \leq 200$ and $L_P \leq 200$ is applied at the config level (`max_len: 200`, `max_protein_len: 200`); all entries used downstream satisfy it. In the broader 19,344-sample complex pool, $\sim 89\%$ of complexes satisfy $L_R \leq 200$ and $\sim 61\%$ satisfy $L_P \leq 200$.

Nov computation and reference pool. Nov% is the per-target fraction of $k = 10$ generated samples that simultaneously (i) pass the Val gate and (ii) have maximum symmetric TM-score $\max(\mathrm{TM}_1, \mathrm{TM}_2) \leq \tau = 0.7$ against every entry of a training reference pool, macro-averaged across

targets. The reference pool consists of $N = 1{,}269$ training RNAs obtained by clustering the $9{,}106$-sample training set with qTMclust at TM ≥ 0.5 and keeping one representative per cluster, so that the pool covers the training fold space without over-counting any dominant fold. Alignments use US-align (`-mol RNA -outfmt 2`) on RNA-only chains with the protein removed. The threshold $\tau = 0.7$ is chosen because the protein convention $\tau = 0.5$ yields near-zero novelty for all generators in the 50–100 nt RNA range, where TM-score is dominated by the limited fold-space diversity rather than method-level differences. Because the pool is fold-deduplicated rather than the full $9{,}106$-sample training set, the reported Nov% is an upper bound on Nov% computed against every individual training RNA.

L Training and inference setting

Setting. Training uses 2× NVIDIA A100-PCIe 40 GB under Distributed Data Parallel (DDP) with length-bucketed batching (`max_batch_size` $= 2$, `max_num_res_squared` $= 80{,}000$, effective batch $\sim$4–8). Each RiboC2F variant trains for ~ 130 epochs on the $9{,}106$-sample `aff_seq` train set, requiring approximately 72 GPU hours under DDP. Inference uses a single GPU, batch size 1, and 50 Euler steps per sample; the test set completes in ~ 1.5 GPU hours at $\sim$10–15 s per sample.

Software. PyTorch with PyTorch-Lightning, PyG, Biopython, ESM-2 (`esm2_t33_650M`), barnaba, and ViennaRNA; evaluation also relies on PyRosetta and PyMOL.

M Pose evaluation metrics

Kabsch alignment and orientation error. Let $\hat{\mathbf{Z}}, \mathbf{Z} \in \mathbb{R}^{n\times 3}$ contain the n predicted and ground-truth RNA $C4'$ coordinate pairs used in evaluation. Orientation and pose errors relative to the protein require both inputs to share a protein reference frame. Subtract each RNA's own centroid to obtain $\hat{\mathbf{Z}}_{\mathrm{c}}$ and $\mathbf{Z}_{\mathrm{c}}$. The optimal proper rotation, computed by Kabsch alignment with determinant correction, is

$$\mathbf{R}_{\mathrm{K}} = \underset{\mathbf{R}\in \mathrm{SO}(3)}{\arg\min} \left\| \hat{\mathbf{Z}}_{\mathrm{c}}\mathbf{R} - \mathbf{Z}_{\mathrm{c}} \right\|_F^2. \tag{19}$$

The orientation error in degrees is

$$\theta = \frac{180}{\pi} \arccos\left[\mathrm{clip}\left(\frac{\mathrm{tr}(\mathbf{R}_{\mathrm{K}}) - 1}{2}, -1, 1 \right) \right], \qquad 0 \leq \theta \leq 180. \tag{20}$$

This angle is computed from RNA coordinates for every method, without using a learned rotation head or applying axis-sign or symmetry minimization. It differs from the principal-axis-frame residual used for training in Appendix D.

Haar reference and aggregate metrics. For a Haar-uniform rotation, the angle density per degree is

$$p_{\mathrm{H}}(\alpha) = \frac{1 - \cos(\pi\alpha/180)}{180}, \qquad 0 \leq \alpha \leq 180. \tag{21}$$

Its mean is approximately $126.5°$, and its probability of an error below $30°$ is approximately 0.75% (reported as 0.8%). With $\bar{\theta}$ denoting the mean error over evaluable targets, the implementation computes

$$\text{Ori-Acc}\% = 100 \ \mathrm{mean}\big[\mathbb{1}(\theta < 30)\big], \qquad \text{Ori-Score}\% = 100 \max\left(0, 1 - \frac{\bar{\theta}}{126.5}\right). \tag{22}$$

In Fig. 5a, empirical angle densities are estimated by Gaussian KDE with reflection boundary correction at $0°$ and $180°$. The Haar reference is the theoretical density without kernel smoothing.

Raw-RMSD, Shape-RMSD, and Pose-RMSD. Using the same coordinate pairs and Kabsch rotation, we compute

$$
\begin{aligned}
d_{\mathrm{raw}} &= \frac{\|\hat{\mathbf{Z}} - \mathbf{Z}\|_F}{\sqrt{n}}, \\
d_{\mathrm{shape}} &= \frac{\|\hat{\mathbf{Z}}_{\mathrm{c}}\mathbf{R}_{\mathrm{K}} - \mathbf{Z}_{\mathrm{c}}\|_F}{\sqrt{n}}, \\
d_{\mathrm{pose}} &= \sqrt{\max\left(0, d_{\mathrm{raw}}^2 - d_{\mathrm{shape}}^2\right)}.
\end{aligned}
\tag{23}
$$

All three quantities are in Å. Raw-RMSD uses the input coordinates without RNA alignment. Shape-RMSD is the residual after removing the optimal rigid translation and rotation. Pose-RMSD is the square root of the reduction in squared RMSD obtained by that alignment. Its magnitude depends on the RNA's size and shape, so it is not a shape-independent distance between rigid poses; the decomposition does not imply statistical independence between pose and shape errors.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 state the pose-concentrated coarse-to-fine cascade and supervision-allocation strategy as the contribution; the experimental results in Section 5 quantitatively support these claims, with strongest docking and competitive intrinsic quality among published baselines.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 discusses the single-RNA single-protein scope and the 200-residue length cap, and Appendix I additionally documents the Boltz-2 oracle's reliability bands at short RNA lengths.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not introduce formal theorems requiring proofs. All claims are empirical and supported by the experiments in Section 5.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The full method is described in Section 4. The architecture, featurization, per-modality flow specifications, auxiliary coarse losses, fine-stage loss weights, training schedule and inference hyperparameters, and the clash-guidance algorithm are documented in Appendices A, B, C, D, E, F, and G, respectively.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: An anonymized snapshot of the training and inference code together with the PRI30k-derived split protocol is provided at `https://anonymous.4open.science/r/rnac2f-anon-submission-4A81/` for review; a non-anonymous public release will follow paper publication. The full architecture, featurization, per-modality flow specifications, auxiliary coarse losses, fine-stage loss weights, training schedule, inference hyperparameters, and clash-guidance algorithm are additionally documented in Appendices A, B, C, D, E, F, G, and L to enable independent reimplementation, and the PRI30k base data is publicly available from Han et al. [24].

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5.1 specifies the data splits and baselines; Appendix K gives the filtering pipeline and length distributions; Appendix L reports the training and inference hardware setting; Appendix F lists all training-time and inference-time hyperparameters.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: 95% bootstrap confidence intervals for every cell of Table 1 (1,000 paired resamples of the test set, seed 42) and pairwise tests of RiboC2F against every baseline and ablation (Wilcoxon signed-rank for continuous metrics, McNemar for binary outcomes) are reported in Appendix J (Tables 6, 7, 8, 9). Per-target distributions for LRMSD and the Kabsch-based SO(3) orientation error defined in Appendix M are additionally visualized in Figures 3 and 5a, and per-component Val pass rates are in Appendix H (Table 5).

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Appendix L lists the GPU type (2× NVIDIA A100-PCIe 40 GB), training duration ($\sim$72 GPU hours per RiboC2F variant), and inference cost ($\sim$1.5 GPU hours over the test set, $\sim$10–15 s per sample).

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics. The work uses publicly available structural data and pretrained models, does not involve human subjects, and the experimental setup preserves anonymity for double-blind submission.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Computational protein-conditioned RNA co-design is positioned in the introduction as a step toward therapeutic discovery and synthetic biology. The 200-residue length cap and the restriction to publicly available structural data limit the scope of generated RNAs to short, well-characterized targets, reducing dual-use concerns relative to large-scale biological-agent design.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The released model is restricted to $\leq$200-residue protein-conditioned RNA co-design and provides no capability beyond what is achievable by combining publicly available components (RNA-FrameFlow, ESM-2, US-align, Boltz-2). It poses no high-risk dual-use concern requiring additional safeguards beyond standard release norms.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All third-party assets are cited at first use, including PRI30k [24], RNA-FrameFlow [6], ESM-2 [33], IPA [23], Boltz-2 [26], US-align [37], and DockQ [31], and each is used under the original publication's terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assets are released with this submission. The PRI30k-derived split protocol used in our experiments is fully described in Appendix K.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing or research with human subjects.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: The work does not involve human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as components of the core method. ESM-2 is a protein language model used for fixed feature extraction only and is cited at first use.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.