RIBOWEAVE: INTEGRATING SECONDARY AND TERTIARY STRUCTURE KNOWLEDGE INTO SEQUENCE REPRESENTATIONS FOR RNA APTAMER DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA aptamers are single-stranded nucleic acid molecules that bind targets with high affinity and specificity, with applications in diagnostics, targeted therapeutics, and drug delivery. High-throughput screening yields aptamer candidates, but validating screened sequences is time-consuming and labor-intensive, motivating computational models to accelerate aptamer discovery. Screening data provide a natural source of supervision, yet typically contain only sequences and enrichment measurements without matched structural information. Moreover, modeling RNA-protein interactions requires integrating information across modalities and granularities, as sequence, secondary structure, and tertiary structure contribute at both local and whole-molecule levels. We present RiboWeave, a framework that transfers structural knowledge into sequence representations through nucleotide-level pairing and geometry prediction and molecule-level contrastive alignment with structural representations. Its Hybrid version combines the resulting representations through task-specific fusion, and all RiboWeave models require only RNA sequences at inference. Cross-modal evaluation reveals both shared and modality-specific structural information, while joint supervision balances the two structural modalities. For downstream activity prediction, RiboWeave Hybrid achieves the highest macro R^2 across CD3 and cMYC under cluster-disjoint splits, outperforming both single-modality cross-granularity variants and single-granularity capacity controls. In a common generation framework, RiboWeave Hybrid achieves the highest novel-sequence fractions while maintaining strong predicted activity and high sequence diversity across CD3 and cMYC. These results support the utility of multimodal and multigranular structural supervision for sequence-only aptamer activity prediction and computational design.

1 INTRODUCTION

RNA aptamers are single-stranded nucleic acid molecules that bind targets with high affinity and specificity, holding broad potential for diagnostics, targeted therapeutics, and drug delivery (Zhou & Rossi, 2017). High-throughput screening can identify aptamers with target-binding activity or specific biological functions (Ellington & Szostak, 1990; Tuerk & Gold, 1990; Su et al., 2025), but validating screened candidates remains time-consuming and labor-intensive. This motivates the development of computational models to accelerate aptamer discovery. Screening-derived data provide a natural source of supervision, with enrichment signals serving as pseudo-activity labels for model training (Zhang et al., 2026).

However, using these screening data to guide aptamer activity prediction and design presents two challenges. First, aptamer function depends on both sequence and structure, yet typical high-throughput screening primarily provides sequences and enrichment measurements without matched experimental structures (Ishida et al., 2020; Zhang et al., 2026). Second, RNA–protein interactions involve multiple modalities and granularities, spanning sequence, secondary-structure, and tertiary-structure information at local and whole-molecule levels. These factors jointly shape recognition, but their relative contributions may vary across RNA–protein interactions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

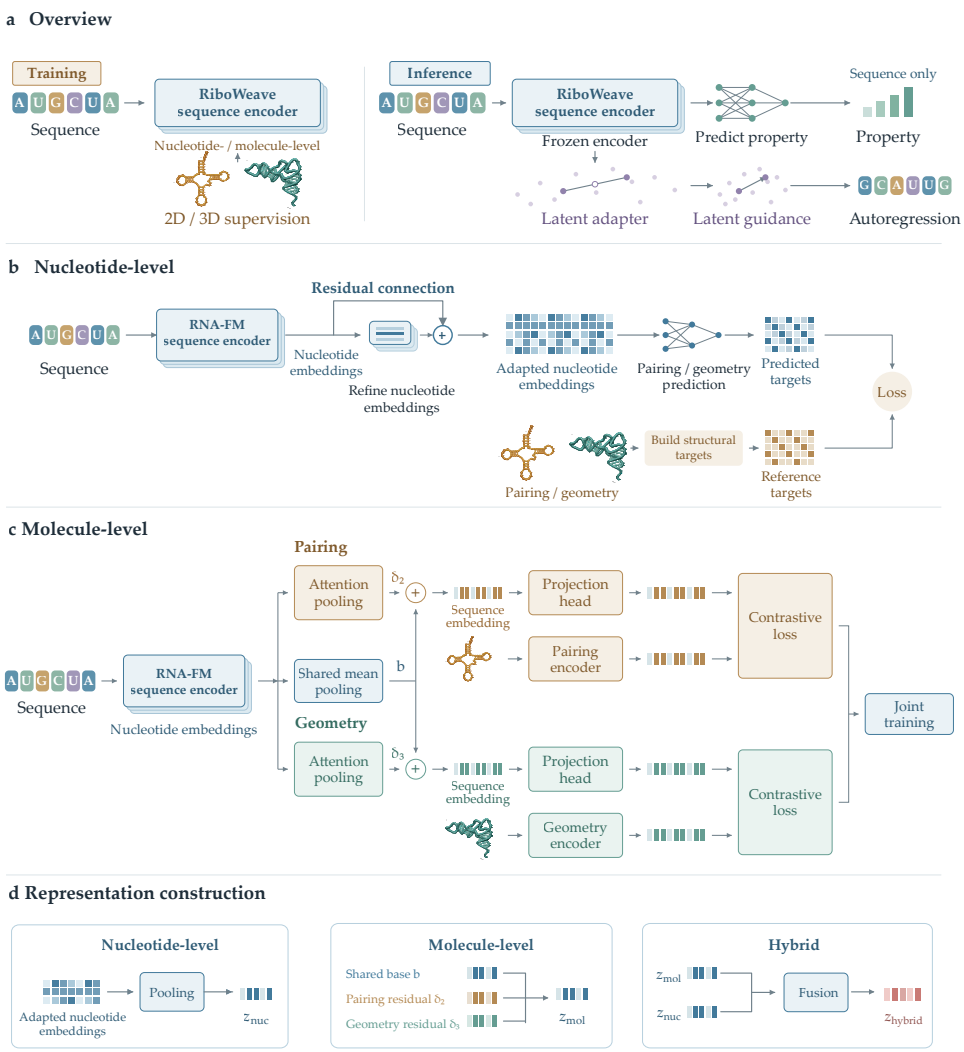

Figure 1: Overview of RiboWeave. (a) Structural knowledge is incorporated during upstream training; downstream prediction and generation use sequence-derived representations with frozen encoders. (b) Nucleotide-level learning supervises RNA-FM and residual adapters through pairing and geometric prediction objectives. (c) Molecule-level learning supervises RNA-FM by aligning secondary- and tertiary-structure encoder embeddings with sequence-derived representations augmented by modality-specific attention residuals. (d) Downstream representation construction.

We introduce RiboWeave, a framework that incorporates secondary- and tertiary-structure knowledge into sequence representations through nucleotide- and molecule-level supervision (Figure 1). Because experimentally resolved RNA structures are far fewer than RNA sequences, we build on RNA-FM (Chen et al., 2022), whose large-scale sequence pretraining yields transferable sequence representations for structural adaptation. Nucleotide-level models learn position-specific structural information through pairing and geometric prediction objectives, while molecule-level models align whole-molecule sequence representations with structural-encoder representations. These routes impose different learning constraints, motivating an empirical assessment of their combination for downstream tasks. At each granularity, secondary- and tertiary-structure supervision can be used separately or jointly, without requiring both annotations for the same RNA. After upstream training, the encoders are frozen, downstream prediction and generation require only RNA sequences. Its Hybrid version combines the nucleotide- and molecule-level representations obtained through joint structural supervision using a lightweight, learnable fusion module fitted to each task.

We evaluate RiboWeave in terms of structural information, activity prediction, and computational design. First, cross-structural evaluation at both granularities shows cross-modal transfer gains, while models trained with the matching structural modality retain an advantage over those transferring from the other modality. These results support shared and modality-specific aspects of the information captured by the two supervision sources. Jointly supervised models retain information useful for both structural modalities, with trade-offs on individual tasks. Next, on cluster-disjoint activity prediction, RiboWeave Hybrid achieves the highest macro R^2 across CD3 and cMYC among the evaluated configurations, outperforming both single-modality cross-granularity variants and single-granularity capacity controls, supporting complementary contributions from structural modality and supervision granularity. Furthermore, permuting structural supervision reduces downstream prediction performance, and predicted-structure-input controls provide comparisons with the direct downstream use of predicted structures. Finally, under evaluation separate from generation guidance, RiboWeave Hybrid achieves the highest novel-sequence fractions while maintaining strong predicted activity and high sequence diversity across CD3 and cMYC within a common generation framework. Together, these results show that integrating structural knowledge across modalities and granularities into sequence representations enables sequence-only aptamer prediction and design, making RiboWeave compatible with sequence–activity datasets without matched structural information.

2 Related Work

2.1 RNA Representation Learning

RNA language models learn transferable representations from large sequence collections. RNA-FM and RiNALMo support structural and functional prediction through sequence pretraining (Chen et al., 2022; Penić et al., 2025), while RNAErnie introduces motif-aware masking and type-guided fine-tuning (Wang et al., 2024a). Structure-aware models further incorporate structural priors during training: RNABERT uses structural alignment learning (Akiyama & Sakakibara, 2022), ERNIE-RNA introduces pairing-informed attention (Yin et al., 2025), and structRFM guides masked pretraining with secondary structure (Zhu et al., 2025). RiboWeave extends this direction by jointly examining secondary- and tertiary-structure supervision at nucleotide and molecule levels, and their effects on downstream activity prediction and design.

Structure-based pretraining has also been explored in representation learning. For example, GearNet learns protein representations directly from three-dimensional structure graphs (Zhang et al., 2023), and its encoder requires structural inputs for downstream tasks. RiboWeave instead uses structural supervision upstream to shape an RNA sequence encoder, so that inference requires only sequences. This enables structure-informed RNA modeling in the absence of structural data.

2.2 Cross-Modal Molecular Representation Learning

Molecular systems can be represented in multiple modalities, and cross-modal learning has been used to connect these complementary views. In protein representation learning, S-PLM aligns amino-acid sequences with three-dimensional structures through contrastive learning (Wang et al., 2025). In small-molecule representation learning, GraphMVP jointly models two-dimensional molecular topology and three-dimensional geometry (Liu et al., 2022). Related approaches have also been explored for RNA. For example, MM-ncRNAFP aligns ncRNA sequence representations with secondary-structure graph representations and subsequently fuses the two modalities for family prediction (Xu et al., 2024). RiboWeave extends this cross-modal setting by using RNA sequence as a common representation space for multiple structural modalities. At the molecule level, RiboWeave aligns sequence representations with secondary- and tertiary-structure representations through separate branches, which jointly supervise a shared sequence encoder in the multimodal setting. RiboWeave further introduces nucleotide-level structural prediction as another supervision granularity, enabling the joint study of structural modality and supervision granularity.

2.3 Function Prediction and Candidate Scoring

Computational approaches have been developed for RNA functional prediction and aptamer candidate evaluation. PrismNet combines RNA sequences with experimentally measured in vivo struc-

tural information to predict RNA-protein interactions (Sun et al., 2021). In aptamer screening, RaptRanker ranks candidates using local sequence and predicted secondary-structure information (Ishida et al., 2020), while RaptScore uses language-model-based representations for aptamer evaluation (Kimura-Yamazaki et al., 2026). RiboWeave instead evaluates how structure-informed sequence representations benefit aptamer activity prediction.

2.4 Computational Aptamer Design

Generative aptamer design spans screening-data-driven and structure-conditioned approaches. RaptGen learns latent representations of HT-SELEX sequences and supports activity-guided latent optimization (Iwano et al., 2022). AptaDiff uses diffusion for de novo aptamer generation (Wang et al., 2024b). GRAPE-LM combines language models, a conditional autoencoder, and screening-derived pseudo-activity scores (Zhang et al., 2026). Structure-conditioned methods such as gRNAde and RhoDesign instead formulate RNA design as inverse folding to generate sequences for specified structures (Joshi et al., 2025; Wong et al., 2024). RiboWeave connects these settings by learning structure-informed sequence representations upstream and using screening-derived activity signals for generation, without requiring a target structure at inference.

3 Method: RiboWeave

3.1 Problem Setup and Overview

Let $x = (x_1, \ldots, x_L)$ denote an RNA sequence. Starting from pretrained RNA-FM, RiboWeave learns structure-informed representations at nucleotide and molecule levels using secondary- and tertiary-structure datasets $\mathcal{D}_2$ and $\mathcal{D}_3$ (Figure 1). Each granularity includes secondary, tertiary, and joint supervision (Table 1). Joint supervision does not require both annotations for the same sequence. After upstream training, the encoders are frozen. For each target t, downstream fusion and adapter modules are trained on screening data $\mathcal{D}_t = \{(x, y_t)\}$ without matched structures.

Table 1: Configurations along two dimensions: supervision granularity and structural modality.

Granularity	2D supervision	3D supervision	Joint supervision
Nucleotide	Nuc-pair	Nuc-geo	Nuc-joint
Molecule	Mol-pair	Mol-geo	Mol-joint

3.2 Nucleotide-Level Structural Supervision

Nucleotide-level models incorporate pairing and geometric information through position-level structural prediction. Nuc-pair, Nuc-geo, and Nuc-joint share an architecture with gated residual adapters on top of RNA-FM, jointly updating the adapters and upper backbone layers. Let z_i denote the output representation at position i.

Nuc-pair predicts the pairing probability of positions i, j from z_i, z_j, using binary cross-entropy on annotated pairs and sampled negatives as its objective $\mathcal{L}_{\mathrm{Pair}}$. In our experiments, annotated positive pairs have target 1 and sampled negatives have target 0.

Nuc-geo predicts anchor-distance bins and four local targets: adjacent-residue geometry, backbone torsion, sugar-ring torsion proxy, and sugar-base orientation proxy. Its objective is

$$\mathcal{L}_{\mathrm{Geo}} = \mathcal{L}_{\mathrm{dist}} + \lambda_{\mathrm{local}} \mathcal{L}_{\mathrm{local}}, \tag{1}$$

where $\mathcal{L}_{\mathrm{dist}}$ is the cross-entropy loss for distance-bin prediction and $\mathcal{L}_{\mathrm{local}}$ averages the available local geometric classification losses.

Nuc-joint applies both pairing and geometric objectives to the same RNA-FM backbone and adapters, supervising a shared representation with both structural modalities.

For downstream applications, nucleotide-level representation z_{nuc} is obtained using a weighted combination of attention and mean pooling, followed by LayerNorm. Further model architecture, supervision targets, and training procedure are detailed in Appendix B.1.

3.3 Molecule-Level Structural Supervision

Molecule-level models incorporate whole-sequence structural information into the RNA-FM representation through sequence-structure alignment. Let u_i denote the nucleotide state produced by fine-tuned RNA-FM. A normalized mean defines a shared sequence representation b, while a modality-specific learned query aggregates the nucleotide states into a residual δ_m:

$$b = \mathrm{LN}\left(\frac{1}{L}\sum_{i=1}^{L} u_i\right), \qquad r_m = \mathrm{LN}_m\big(b + \delta_m(\{u_i\}_{i=1}^{L})\big), \quad m \in \{2,3\}. \tag{2}$$

Here $m = 2, 3$ denote secondary- and tertiary-structure supervision, respectively. Thus, each r_m combines a shared whole-sequence representation with a modality-specific attention residual.

Secondary structure is encoded with a Graph Isomorphism Network with Edge features (GINE) (Hu et al., 2020), while tertiary structure is encoded with a rotation- and translation-invariant GNN. For each modality, r_m is then projected to s_m, and v_m denotes the normalized representation produced by the corresponding structure encoder. The sequence and structure representations are aligned using bidirectional InfoNCE together with an auxiliary masked-language-model objective:

$$\mathcal{L}_{\mathrm{mol}}^{(m)} = \mathcal{L}_{\mathrm{InfoNCE}}(s_m, v_m) + \lambda_{\mathrm{MLM}}\mathcal{L}_{\mathrm{MLM}}^{(m)}. \tag{3}$$

Mol-pair and Mol-geo use one structural modality each. Mol-joint shares the RNA-FM backbone across both modalities, while retaining modality-specific query pooling and projection modules. During training, matched sequence-structure pairs form positives while the remaining minibatch samples provide negatives, with highly similar pairs masked to reduce false negatives.

For downstream applications, Mol-pair and Mol-geo directly use their corresponding single-modality sequence representations, r_2 and r_3, respectively. Mol-joint instead exports the shared base b together with the modality-specific residuals δ_2 and δ_3. These components are independently projected, and the two structural residuals are weighted by task-specific sigmoid gates before being added to the projected base and normalized to obtain z_{mol}. Further model architecture, supervision targets and training details are provided in Appendix B.2.

3.4 Cross-Granularity Fusion

RiboWeave Hybrid combines z_{nuc} from Nuc-joint with z_{mol} from Mol-joint:

$$h(x) = \mathrm{LN}(\mathrm{Linear}\left([\mathrm{LN}(z_{\mathrm{mol}}); \mathrm{LN}(z_{\mathrm{nuc}})]\right). \tag{4}$$

The upstream encoders remain frozen, while the cross-granularity fusion module and the downstream prediction head are trained jointly for each task.

3.5 Downstream Prediction and Generation

Activity prediction. For each target and representation configuration, a multilayer perceptron is trained on the sequence representations with mean squared error, using a cluster-disjoint train/validation/test split.

Computational generation. For each target, sequences are partitioned by cluster into a development set, a final-evaluator training set, and a natural holdout set. The development set is used to train and select the generator, the final-evaluator set is used to train predictors for scoring generated candidates, and the natural holdout set is reserved for assessing evaluator reliability.

The generative model consists of a latent adapter and an autoregressive decoder, with activity guidance provided by predictors trained on the development set. The adapter maps sequence representations into a 128-dimensional latent space, while the decoder reconstructs sequences conditioned on the latent codes. Training combines representation and sequence reconstruction with activity-guided objectives and latent-space regularization. Candidate latent codes are initialized from high-activity development sequences, shifted toward higher predicted activity while remaining close to the learned latent manifold, and decoded autoregressively. All representation configurations use the same generation architecture, with separately trained adapters and decoders. Additional generation details are provided in Appendix B.5.

4 EXPERIMENTAL SETUP

Upstream structure-informed representation learning. Secondary- and tertiary-structure datasets are preprocessed and split before upstream training. To avoid overlap with the cross-structural evaluation sets, candidate training sequences are filtered by removing exact matches and homologous sequences. Dataset sources, preprocessing and partition statistics are provided in Appendix A.1 and A.2.

Structural cross-modal evaluation. Nucleotide- and molecule-level evaluations use the same secondary- and tertiary-structure datasets. At the nucleotide level, frozen representations are evaluated on pairing and contact prediction using learned readout heads, with performance measured by the area under the precision–recall curve (AUPRC). At the molecule level, we assess how well sequence representations recover the similarity relationships defined by 2D and 3D structural reference spaces, using kNN@10, linear centered kernel alignment (CKA), and pairwise Spearman correlation. Full evaluation protocols are provided in Appendix C.1.

Activity prediction. We compare RNA-FM, six structure-supervised configurations, RiboWeave Hybrid, three additional RNA language models, and other controls on the CD3, RBD, and cMYC tasks from CRISmer (Zhang et al., 2026). For each target, sequences are clustered at 90% identity, and entire clusters are assigned to disjoint training, validation, and test partitions. Results are summarized across five downstream seeds. Macro R^2 in the main analysis is the mean across CD3 and cMYC within each seed, while the three-target average is retained with the complete results in Appendix D.3. Data splits and training details are provided in Appendices A.2 and B.4.

Computational design. For each target, sequences are clustered at 90% identity and assigned to disjoint development, final-evaluator training, and natural holdout partitions. We compare all representation configurations within the same generation framework and generate 1,000 candidates per target, configuration, and seed. Generated sequences are scored by a common panel of predictors trained on the disjoint final-evaluator partition. We report predicted activity, novelty relative to the development set, and within-set sequence diversity. The main analysis focuses on CD3 and cMYC, while RBD is treated as exploratory because of limited evaluator reliability (Appendix D.6). Data partitions, evaluator configurations, and metric definitions are provided in Appendices A.2 and C.3.

5 RESULTS

5.1 CROSS-MODAL EVALUATION

5.1.1 NUCLEOTIDE-LEVEL CROSS-MODAL EVALUATION

We evaluate pairing and contact prediction using the same decoder-training budget across representations within each task. Contacts are defined by C1′–C1′ distances of at most 8 Å, distinguishing close spatial contacts from canonical pairing. We report 2D AUPRC for pairing prediction and 3D AUPRC for contact prediction (Figure 2). Protocol details appear in Appendix C.1.1, and additional results in Appendix D.1.1.

Both single-modality experts improve prediction in the other modality relative to RNA-FM, supporting bidirectional transfer. However, transfer remains incomplete, as Nuc-pair achieves higher 2D AUPRC while Nuc-geo achieves higher 3D AUPRC, suggesting that secondary- and tertiary-structure supervision captures both shared and modality-specific information. Moreover, permutation controls further support the benefit of correctly matched structural supervision.

Nuc-joint achieves the highest mean 3D AUPRC, but its 2D AUPRC remains below Nuc-pair. Joint supervision thus retains both types of structural information while potentially trading off performance on individual tasks.

5.1.2 MOLECULE-LEVEL CROSS-MODAL EVALUATION

At molecule level, we assess agreement between model representations and fixed 2D and 3D structural references in molecular similarity relationships. We use kNN@10, CKA and pairwise Spear-

man correlation to measure agreement in local neighborhoods, overall representation geometry, and pairwise similarity rankings, respectively. Higher values indicate stronger agreement. Evaluation details appear in Appendix C.1.2.

Figure 2 shows cross-modal gains over RNA-FM for both experts, while the expert trained with the matching modality remains strongest. Mol-joint preserves molecular similarity relationships defined by both 2D and 3D structural references, but remains below the corresponding single-modality expert. Together, the results at both granularities support shared and modality-specific components of the structural information captured by 2D and 3D supervision. Jointly supervised models exhibit balanced profiles across the two structural modalities.

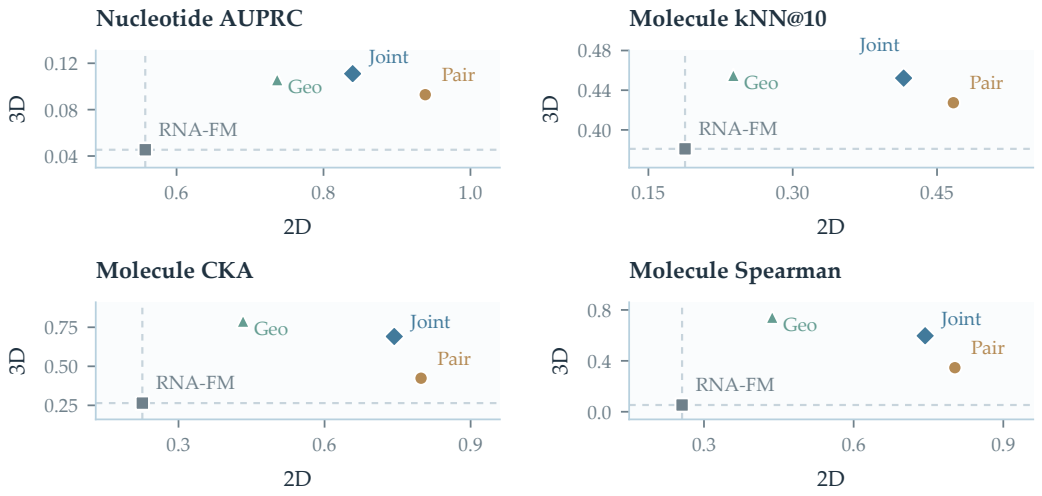

Figure 2: Cross-modal evaluation. Dashed lines mark RNA-FM. Full values: Tables 6 and 8.

5.2 Activity Prediction and Configuration Analysis

Figure 3a compares eleven configurations on cluster-disjoint activity prediction for CD3 and cMYC. Among the language-model baselines, RNA-FM achieves the highest two-target macro average. Five of the six structure-supervised single-route models outperform RNA-FM on both targets, indicating that structural supervision generally provides predictive information beyond pretrained sequence representations. Mol-geo is the exception, indicating that molecule-level geometric supervision alone does not yield consistent gains in these activity-prediction tasks. In contrast, Mol-joint, which combines molecule-level pairing and geometric information, outperforms either single-modality molecular representation, supporting their complementarity. RiboWeave Hybrid achieves the highest mean R^2 on CD3, cMYC, and their macro average, further showing that information learned at different granularities can be jointly exploited. Complete prediction results for all three targets are reported in Appendix D.3.

Figure 3b examines cross-granularity and cross-modal integration. Pair-hybrid and Geo-hybrid combine nucleotide- and molecule-level representations within a single structural modality and achieve higher mean R^2 than their corresponding single-granularity counterparts across CD3, cMYC, and the macro average. The full Hybrid further integrates pairing and geometric information across both granularities and achieves the highest mean performance on CD3, cMYC, and their macro average. It also outperforms both single-granularity capacity controls across all five downstream seeds. Together, these results support complementary benefits from cross-granularity fusion and multimodal integration, with their combination yielding the strongest overall activity-prediction performance under the evaluated setting.

Appendix D.5 further compares upstream supervision permutations, folding summaries, and direct graph inputs to assess the contribution of correctly aligned structural supervision and distinguish upstream structure-informed representation learning from the direct downstream use of predicted structures.

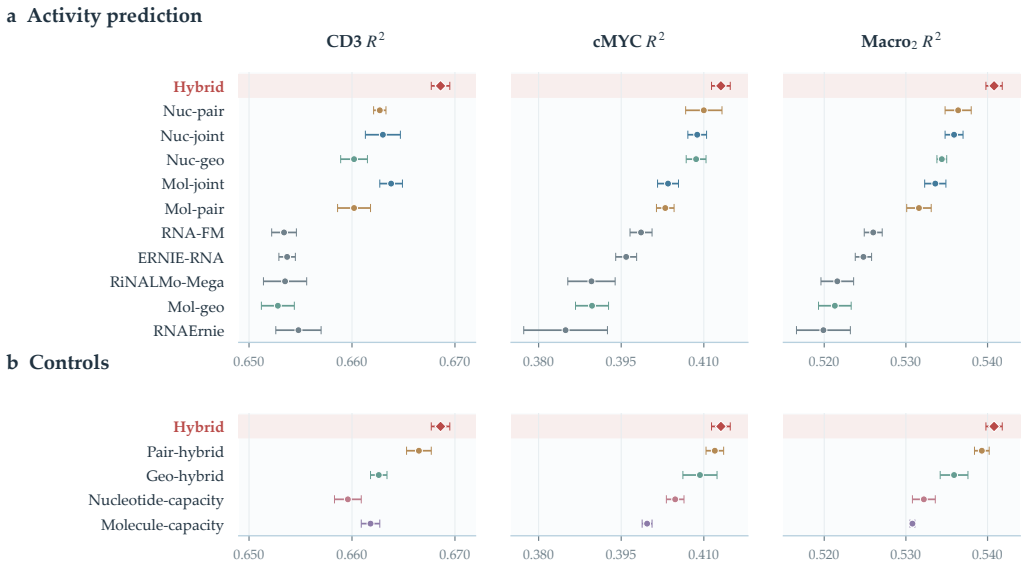

Figure 3: Activity prediction and controls under cluster-disjoint splits. (a) Eleven sequence-derived representations. (b) Hybrid, two single-modality hybrid variants, and two single-granularity capacity controls. Markers and horizontal error bars show mean test $R^2 \pm$ sample SD across five downstream seeds. Macro$_2$ averages CD3 and cMYC within each seed. Axis ranges are shared between (a) and (b) within each metric. Full numerical results are provided in Tables 12 and 13.

5.3 Computational Design

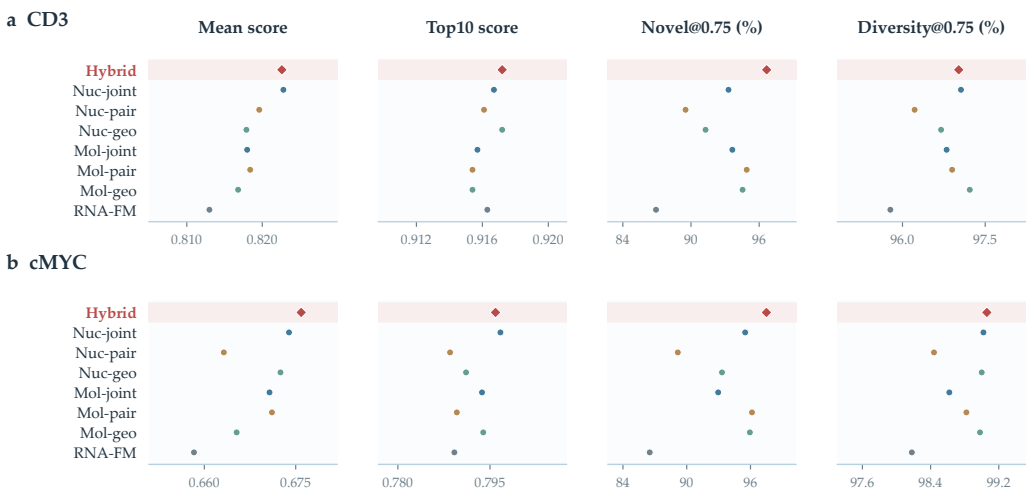

Figure 4: Generation metrics on (a) CD3 and (b) cMYC. Markers show means across five generation seeds. Scores are predictions from the common evaluator panel. Novelty is the fraction of generated sequences whose maximum identity to development-set sequence is at most 75%; diversity is the fraction of within-set sequence clusters defined at 75% identity. Axis ranges differ across panels. Full numerical results, including sample standard deviations, appear in Table 18.

Figure 4 compares eight representation configurations within a common generation framework. All six structure-supervised configurations and RiboWeave Hybrid achieve higher mean evaluator scores, novel-candidate fractions (Novel@0.75), and within-set diversity (Diversity@0.75) than RNA-FM on both CD3 and cMYC. These gains occur in both nucleotide- and molecule-level mod-

els, supporting the utility of structural supervision at either granularity for activity-guided generation. Among these configurations, Hybrid achieves the highest novel-candidate fraction on both targets while maintaining leading or near-leading mean evaluator scores.

Figure 5 examines the trade-off between predicted activity and sequence novelty by progressively restricting similarity to the development set. At matched identity thresholds, RiboWeave Hybrid retains a larger pool of candidates while maintaining competitive predicted activity compared with RNA-FM on both CD3 and cMYC.

Overall, the RiboWeave family improves the combination of predicted activity, sequence novelty, and diversity relative to RNA-FM within the evaluated framework. Exploratory RBD generation results are provided in Appendix D.8.

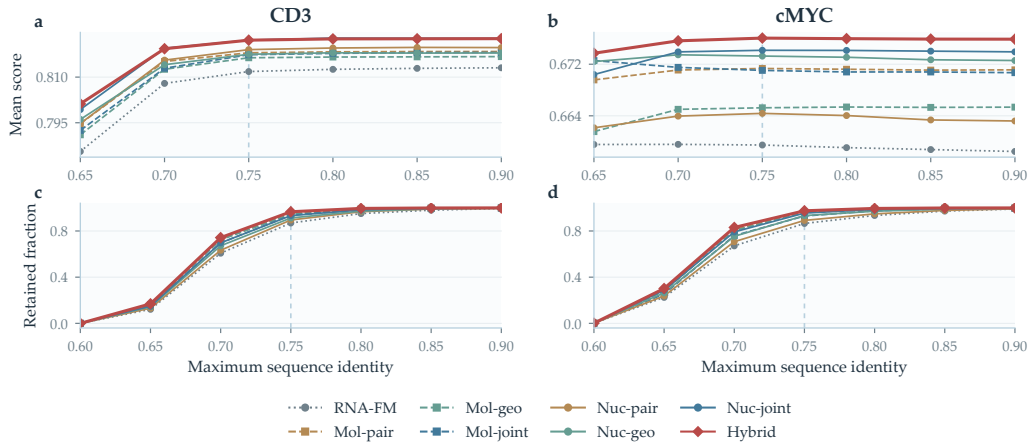

Figure 5: Predicted activity and candidate retention under varying novelty requirements. Candidates are retained when their maximum sequence identity to the development set is at most the threshold; lower thresholds impose stricter novelty requirements. Panels (a,b) show mean evaluator scores among retained candidates, and (c,d) show the retained fractions, for CD3 and cMYC, respectively. Curves are averaged over five generation seeds, and the dashed line marks the 0.75 threshold used for Novel@0.75. Scores are computational predictions, with candidate subsets determined separately for each configuration.

6 Conclusion

RiboWeave integrates secondary- and tertiary-structure information at nucleotide and molecule levels for sequence-only aptamer modeling. Task-specific fusion of these multimodal and multigranular representations improves activity prediction and supports computational design, suggesting broader potential for structure-aware RNA modeling in diagnostics and therapeutics.

Our current evaluation is limited to three target-specific screening tasks, so whether the framework generalizes to additional targets remains to be tested. In addition, the generated candidates have not yet been experimentally validated. Addressing these limitations requires two immediate steps: evaluating the framework on a broader set of targets and experimentally characterizing the generated candidates. Beyond these validations, incorporating RNA–protein interaction information could improve cross-target generalization, while feeding experimental measurements back into model training would ultimately support a Design–Build–Test–Learn (DBTL) pipeline (Carbonell et al., 2018) for iterative aptamer generation and optimization.

Reproducibility Statement

To support reproducibility, Appendix A describes the data sources, preprocessing, filtering, and dataset splits. The model architectures, training objectives, optimization settings, and checkpoint-selection procedures are detailed in Appendix B. Evaluation metrics, comparison protocols, and control experiments are specified in Appendix C, with additional numerical results in Appendix D. Random seeds and aggregation procedures are reported alongside the corresponding training and evaluation protocols.

Ethics Statement

This study presents computational analyses using existing datasets. The reported predictions do not establish experimental activity, and the generated candidates have not been experimentally validated. Practical applications would require further validation and assessment appropriate to the intended use.

AI Use Statement

We used generative AI tools to assist with code development and method implementation, debugging experimental pipelines, discussing and interpreting experimental results, organizing related literature, drafting and translating manuscript text, and revising references and figures. The authors reviewed and revised AI-assisted content and checked the implementation and experimental outputs. The authors take responsibility for the final manuscript, code, analyses, and reported results.

References

Manato Akiyama and Yasubumi Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics*, 4 (1):lqac012, 2022. doi: 10.1093/nargab/lqac012.

Pablo Carbonell, Adrian J Jervis, Christopher J Robinson, Cunyu Yan, Mark Dunstan, Neil Swainston, Maria Vinaixa, Katherine A Hollywood, Andrew Currin, Nicholas JW Rattray, et al. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology*, 1(1):66, 2018.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *BioRxiv*, pp. 2022–08, 2022.

Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Research*, 46(11):5381–5394, 2018. doi: 10.1093/nar/gky285.

Andrew D. Ellington and Jack W. Szostak. In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346(6287):818–822, 1990. doi: 10.1038/346818a0.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.12265.

Ryoga Ishida, Tatsuo Adachi, Aya Yokota, Hidehito Yoshihara, Kazuteru Aoki, Yoshikazu Nakamura, and Michiaki Hamada. RaptRanker: in silico RNA aptamer selection from HT-SELEX experiment based on local sequence and structure information. *Nucleic Acids Research*, 48(14): e82, 2020. doi: 10.1093/nar/gkaa484.

Natsuki Iwano, Tatsuo Adachi, Kazuteru Aoki, Yoshikazu Nakamura, and Michiaki Hamada. Generative aptamer discovery using RaptGen. *Nature Computational Science*, 2(6):378–386, 2022. doi: 10.1038/s43588-022-00249-6.

Chaitanya K. Joshi, Arian R. Jamasb, Ramon Viñas, Charles Harris, Simon V. Mathis, Alex Morehead, Rishabh Anand, and Pietro Liò. gRNAde: Geometric deep learning for 3D RNA inverse design. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lvw3UgeVxS.

Senthilkumar Kailasam, Dhananjay Bhattacharyya, and Manju Bansal. Sequence dependent variations in RNA duplex are related to non-canonical hydrogen bond interactions in dinucleotide steps. *BMC Research Notes*, 7:83, 2014. doi: 10.1186/1756-0500-7-83.

Akira Kimura-Yamazaki, Tatsuo Adachi, Shigetaka Nakamura, Yoshikazu Nakamura, and Michiaki Hamada. RaptScore: a large language model-based algorithm for versatile aptamer evaluation. *Nucleic Acids Research*, 54(2):gkaf1480, 2026. doi: 10.1093/nar/gkaf1480.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3D geometry. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xQUe1pOKPam.

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011. doi: 10.1186/1748-7188-6-26.

Rafael Josip Penić, Tin Vlašić, Roland G. Huber, Yue Wan, and Mile Šikić. RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks. *Nature Communications*, 16(1):5671, 2025. doi: 10.1038/s41467-025-60872-5.

Michael F. Sloma and David H. Mathews. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA*, 22(12):1808–1818, 2016. doi: 10.1261/rna.053694.115.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988.

Huang Su, Yifan Chen, Xuyang Zhao, Zengyi Lu, Tongxuan Wei, Qinguo Liu, Xiyang Liu, Qinhao Zhang, Siqi Bian, Qianwei Qiu, Panzhu Yao, Wenlang Liu, Zheng Zheng, Da Xu, and Liqin Zhang. Systematic evolution of functional oligonucleotides for targeted protein degradation. *Chem*, 11(5):102408, 2025. doi: 10.1016/j.chempr.2024.102408.

Lei Sun, Kui Xu, Wenze Huang, Yucheng T. Yang, Pan Li, Lei Tang, Tuanlin Xiong, and Qiangfeng Cliff Zhang. Predicting dynamic cellular protein–RNA interactions by deep learning using in vivo RNA structures. *Cell Research*, 31(5):495–516, 2021. doi: 10.1038/s41422-021-00476-y.

Marcell Szikszai, Marcin Magnus, Siddhant Sanghi, Sachin Kadyan, Nazim Bouatta, and Elena Rivas. RNA3DB: A structurally-dissimilar dataset split for training and benchmarking deep learning models for RNA structure prediction. *Journal of Molecular Biology*, 436(17):168552, 2024. doi: 10.1016/j.jmb.2024.168552.

Zhen Tan, Yinghan Fu, Gaurav Sharma, and David H. Mathews. TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research*, 45(20):11570–11581, 2017. doi: 10.1093/nar/gkx815.

Craig Tuerk and Larry Gold. Systematic Evolution of Ligands by Exponential Enrichment: RNA Ligands to Bacteriophage T4 DNA Polymerase. *Science*, 249(4968):505–510, 1990. doi: 10.1126/science.2200121.

Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, and Dong Xu. S-PLM: Structure-Aware Protein Language Model via Contrastive Learning Between Sequence and Structure. *Advanced Science*, 12(5):2404212, 2025. doi: 10.1002/advs.202404212.

Ning Wang, Jiang Bian, Yuchen Li, Xuhong Li, Shahid Mumtaz, Linghe Kong, and Haoyi Xiong. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nature Machine Intelligence*, 6(5):548–557, 2024a. doi: 10.1038/s42256-024-00836-4.

Zhen Wang, Ziqi Liu, Wei Zhang, Yanjun Li, Yizhen Feng, Shaokang Lv, Han Diao, Zhaofeng Luo, Pengju Yan, Min He, and Xiaolin Li. AptaDiff: de novo design and optimization of aptamers based on diffusion models. *Briefings in Bioinformatics*, 25(6):bbae517, 2024b. doi: 10.1093/bib/bbae517.

Felix Wong, Dongchen He, Aarti Krishnan, Liang Hong, Alexander Z. Wang, Jiuming Wang, Zhihang Hu, Satotaka Omori, Alicia Li, Jiahua Rao, Qinze Yu, Wengong Jin, Tianqing Zhang, Katherine Ilia, Jack X. Chen, Shuangjia Zheng, Irwin King, Yu Li, and James J. Collins. Deep generative design of RNA aptamers using structural predictions. *Nature Computational Science*, 4(11):829–839, 2024. doi: 10.1038/s43588-024-00720-6.

Ruiting Xu, Dan Li, Wen Yang, Guohua Wang, and Yang Li. Improving ncRNA family prediction using multi-modal contrastive learning of sequence and structure. *Bioinformatics*, 40(11):btae640, 2024.

Weijie Yin, Zhaoyu Zhang, Shuo Zhang, Liang He, Ruiyang Zhang, Rui Jiang, Gan Liu, Jingyi Wang, Xuegong Zhang, Tao Qin, and Zhen Xie. ERNIE-RNA: an RNA language model with structure-enhanced representations. *Nature Communications*, 16(1):10076, 2025. doi: 10.1038/s41467-025-64972-0.

Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature Methods*, 19(9):1109–1115, 2022. doi: 10.1038/s41592-022-01585-1.

Jun Zhang, Ju Zhang, Shaoxuan Tang, Chuancheng Liu, Yushan Cai, Hao Zeng, Xiangjie Meng, Bei Liu, Yang Zhang, and Yu Wang. Single-round evolution of RNA aptamers with GRAPE-LM. *Nature Biotechnology*, 2026. doi: 10.1038/s41587-026-03007-5.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=to3qCB3tOh9`.

Jiehua Zhou and John Rossi. Aptamers as targeted therapeutics: current potential and challenges. *Nature Reviews Drug Discovery*, 16(3):181–202, 2017. doi: 10.1038/nrd.2016.199.

Heqin Zhu, Ruifeng Li, Ao Chang, Haobin Chen, Feng Zhang, Fenghe Tang, Tong Ye, Xin Li, Yunjie Gu, Peng Xiong, and S. Kevin Zhou. A fully open structure-guided RNA foundation model for robust structural and functional inference. *bioRxiv*, 2025. doi: 10.1101/2025.08.06.668731. Preprint.

A DATA

A.1 DATASETS AND PREPROCESSING

Secondary-structure data. We merged the secondary-structure data from RNAStralign (Tan et al., 2017), ArchiveII (Sloma & Mathews, 2016), and bpRNA-1m (Danaee et al., 2018), removed exact duplicates and redundant contained sequences, and replaced unrecognized nucleotide symbols with `N`. Records without annotated base pairs were excluded. Sequences longer than 1,022 nt were split at boundaries not crossed by any annotated base pair, so that paired nucleotides remained within the same fragment. After preprocessing, the retained dataset contained 136,293 fragments. We reserved 1,123 records (1,040 5S and 83 16S rRNAs) for testing, leaving 135,170 candidate fragments for subsequent filtering and training–validation splitting.

Tertiary-structure data. We kept the original RNA3DB (Szikszai et al., 2024) training–test split, with each fragment inheriting the assignment of its source structure. Unresolved terminal regions were trimmed, and large internal gaps served as splitting boundaries. Short internal gaps were filled with `N` and excluded from the 3D graph. Sequences longer than 1,022 nt were split further by counting the spatial interactions crossing each fragment boundary. Cut positions were chosen to minimize disrupted interactions while staying within the length limit. Preprocessing yielded 27,309 fragments from the RNA3DB training partition and 2,115 from its test partition.

High-throughput screening data. Downstream experiments use the CD3, RBD, and cMYC screening datasets associated with GRAPE-LM (Zhang et al., 2026). These datasets pair RNA sequences with pseudo-activity scores derived from screening enrichment, and do not provide matched structures. We retain the original definition of pseudo-activity scores and use separate protocols for activity prediction and generation.

A.2 DATA FILTERING AND SPLITTING

Structural frozen-test filtering. The 2D and 3D test datasets are frozen and used to filter the candidate training pool by sequences. Exact matches to test sequences are removed, together with homologous candidates identified by MMseqs2 (Steinegger & Söding, 2017) at $\geq$80% sequence identity and $\geq$80% coverage of either the query or the target.

Structural training–validation data split. The retained training candidates are grouped using connected components of the 80%-identity homology graph. Each component is assigned entirely to training or validation. For RNA3DB, we split only the original training partition. Table 2 summarizes the candidate records after filtering and splitting.

Table 2: Structural dataset statistics after preprocessing, filtering and splitting.

Partition	Secondary structure	Tertiary structure	Total
Initial preprocessing training datasets	135,170	27,309	162,479
After frozen-test filtering training datasets	90,409	17,998	108,407
Final training datasets	84,579	17,098	101,677
Final validation datasets	5,830	900	6,730
Final test datasets	1,123	2,115	3,238

Aptamer sequence-function data splits. For each target, we pool the supplied functional records and remove the fixed 9-nt 5′ and 14-nt 3′ flanks for clustering. Variable regions are grouped into connected components at 90% sequence identity using Hamming distance. Entire clusters are assigned to partitions with activity stratification, so no cluster appeared in more than one partition.

For activity prediction, we use a nominal 70/15/15 training-validation-test split. For generation, we use a separate nominal 70/15/15 split into a development set, a final evaluator training set, and a natural holdout set. The development set is further divided for generator training and validation. Final evaluators are trained and selected within the final evaluator training set and are used to evaluate newly generated sequences, while the natural holdout set is reserved for assessing the reliability of the final evaluators.

A.3 Cross-modal Evaluation Samples

The nucleotide- and molecule-level evaluations draw on the same underlying 2D and 3D structural datasets described above. The molecule-level 2D evaluation starts from the 1,123-record 5S/16S test set. To reduce redundancy, these sequences are clustered with MMseqs2 at 80% sequence identity and 80% bidirectional coverage, yielding 261 clusters. One deterministic representative is retained per cluster, giving 261 RNAs for evaluation. For the 3D test set, task-specific validity filtering is applied before evaluation: starting from the 2,115 preprocessed records, 2,114 RNAs are retained for the molecule-level evaluation, while 2,105 records with valid evaluable contact labels are retained for the nucleotide-level contact evaluation.

B Models and Training Details

RNA-FM is the principal baseline. The structure-supervised models are Nuc-pair, Nuc-geo, Nuc-joint, Mol-pair, Mol-geo, and Mol-joint, with Hybrid integrating the two jointly supervised granularities. Activity prediction additionally includes ERNIE-RNA, RNAErnie, and RiNALMo-Mega, yielding eleven configurations. All representations are computed from RNA sequence alone.

B.1 Nucleotide-level Models

Encoder and adapter. Nuc-pair, Nuc-geo, and Nuc-joint map an input sequence x to nucleotide representations $Z = A_\phi(E_\theta(x)) \in \mathbb{R}^{L\times 640}$, where E_θ is RNA-FM and A_ϕ is a three-block residual adapter. RNA-FM states are normalized and projected before entering the adapter. Each block applies

$$u_i^{(\ell+1)} = u_i^{(\ell)} + \sigma(g_\ell)\,\mathrm{FFN}_\ell\Big(\mathrm{LN}(u_i^{(\ell)})\Big), \tag{5}$$

where g_ℓ is a learned scalar and each FFN has hidden width 1024 with GELU activation. After that, a final normalization produces z_i. Training updates the upper six backbone layers and the adapter. Nuc-pair and Nuc-geo are trained separately, whereas Nuc-joint shares one backbone and adapter across both objectives.

Pairing objective. The pairing head applies non-affine layer normalization, $\bar{z}_i = \mathrm{LN}(z_i)$, and predicts p_{ij} from the symmetric features

$$[\bar{z}_i + \bar{z}_j;\ |\bar{z}_i - \bar{z}_j|;\ \bar{z}_i \odot \bar{z}_j] \tag{6}$$

using a 256-unit GELU hidden layer and an output logit. Pairing supervision uses binary labels from individual BPSEQ records. Positive candidates are annotated AU, UA, GC, CG, GU, or UG pairs with sequence separation of at least four. Each candidate is matched to a negative using an epoch-dependent deterministic sampler. The sampler first seeks a non-positive pair of these types with the same sequence separation; if none is available, it uses another non-positive pair of these types with sequence separation at least four. After negative matching, retained positives form $\mathcal{P}$ and have target 1, while sampled negative instances form $\mathcal{N}$ and have target 0. The per-record loss is

$$\mathcal{L}_{\mathrm{Pair}} = \frac{1}{2|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \mathrm{BCE}(p_{ij}, 1) + \frac{1}{2|\mathcal{N}|}\sum_{(i,j)\in\mathcal{N}} \mathrm{BCE}(p_{ij}, 0). \tag{7}$$

Here $\mathcal{N}$ is counted with multiplicity. Record losses are averaged over the nominal batch.

Geometric objectives. Geometric supervision combines anchor-distance prediction with four local classification tasks (Table 3). The heads receive sequence representations using the pair and triplet features

$$u_{ij} = [z_i; z_j; |z_i - z_j|; z_i \odot z_j], \tag{8}$$

$$q_{ijk} = [z_i; z_j; z_k; |z_i - z_j|; |z_j - z_k|; z_i \odot z_k]. \tag{9}$$

Here $\odot$ denotes the Hadamard product. Pairwise and triplet heads contain two 512-unit SiLU hidden layers; single-position heads contain one 256-unit SiLU hidden layer.

Table 3: Geometric supervision targets. BASE denotes a base centroid.

Head	Coordinate-derived target	Input / classes
Anchor distances	P–P, C4′–C4′, C1′–C1′, BASE–BASE, BASE–C4′ distances	u_{ij} / 5×20
Adjacent geometry	Nearest reference center of nine standardized cross-distances between adjacent residues' P, C4′, and C1′ anchors	u_{ij} / 16
Backbone pseudo-torsion	Dihedral of $\mathrm{P}_i, \mathrm{C4}'_i, \mathrm{C4}'_j, \mathrm{C4}'_k$	q_{ijk} / 16
Sugar-ring torsion proxy	Within-residue dihedral of C1′, C2′, C3′, C4′	z_i / 8
Sugar–base orientation proxy	Cosine between C4′ →C1′ and C1′ →N	z_i / 8

Each of the five anchor distances is discretized into 20 bins. The bin boundaries are determined from the distribution of anchor distances sampled from RNA3DB structures and remain fixed during training. Training pairs are sampled across sequence-separation ranges.

For adjacent geometry, let d_{ij} contain the nine cross-distances. We compute per-dimension medians m and interquartile ranges s from adjacent-residue distance vectors sampled from RNA3DB structures, and obtain 16 reference centers c_k by K-means in the standardized space. These statistics and centers remain fixed during training. The target is

$$y_{ij}^{\mathrm{adj}} = \underset{k\in\{1,\dots,16\}}{\arg\min} \; \|(d_{ij} - m) \oslash s - c_k\|_2^2 \,. \tag{10}$$

Here $\oslash$ denotes elementwise division. Backbone and sugar-ring dihedrals are discretized uniformly over $[-\pi, \pi]$, and the orientation cosine is discretized into eight equal-width bins over $[-1, 1]$. Labels missing any required atom are excluded.

The geometric objective is $\mathcal{L}_{\mathrm{Geo}} = \mathcal{L}_{\mathrm{dist}} + 0.3\,\mathcal{L}_{\mathrm{local}}$, as defined in Eq. (1). Each geometric task first averages cross-entropy over valid labels within each record, then aggregates records with weights based on source frequency. Local classification tasks additionally use class-balancing weights to reduce the dominance of frequent classes. The local loss $\mathcal{L}_{\mathrm{local}}$ is the mean loss over local tasks with valid labels and is zero when none are available.

Optimization and joint training. All models are trained for 20 epochs with AdamW, constant learning rates, weight decay 10^{-5}, and gradient-norm clipping at 1. The learning rate is 10^{-5} for the upper backbone and 1.5×10^{-4} for the adapter. Pairing computations accumulate microbatch gradients before each nominal-batch update. Length-bucketed batches are shuffled, with a fixed number retained per epoch (Table 4). An epoch thus denotes a scheduled batch budget rather than necessarily a complete pass through the training set. Validation traverses all eligible records and retains incomplete batches.

Table 4: Training budgets and selected checkpoints. Joint training retains both task-specific batch budgets while combining updates at aligned events.

Setting	Nuc-pair	Nuc-geo	Nuc-joint
Pair batch size	48	–	48
Geo batch size	–	12	12
Pair batches per epoch	1760	–	1760
Geo batches per epoch	–	1408	1408
Optimizer updates per epoch	1760	1408	1760
Selected epoch	19	6	15

Joint training combines the pairing and geometric gradients with a fixed weight $\lambda \approx 0.093$. Before training updates, this weight is set to the ratio of the median geometric gradient norm to the median pairing gradient norm on the shared encoder over four paired training batches, and is held constant thereafter. The two batch lists are aligned in order to form 1408 paired events and 352 Pair-only events per epoch. At paired events, both gradients are evaluated at the same parameter state and combined before clipping and a single AdamW update (Algorithm 1).

Algorithm 1 Joint training of Nuc-joint

Require: Pair and Geo training/validation sets; one shared backbone and adapter
Initialize the shared encoder and task heads; $s^\star \leftarrow +\infty$.
Calibrate λ on four paired training batches without updating parameters.
for $e = 1, \ldots, 20$ **do**
  Construct and align 1760 Pair batches and 1408 Geo batches.
  for each aligned event **do**
    Compute $g_{\mathrm{Pair}} \leftarrow \nabla_\Theta \mathcal{L}_{\mathrm{Pair}}$ on the Pair batch.
    Compute $g_{\mathrm{Geo}} \leftarrow \nabla_\Theta \mathcal{L}_{\mathrm{Geo}}$ at the same state if assigned; otherwise $g_{\mathrm{Geo}} \leftarrow 0$.
    Clip $\lambda g_{\mathrm{Pair}} + g_{\mathrm{Geo}}$ to norm 1 and apply one AdamW update.
  end for
  Evaluate both validation losses; at $e = 1$, fix their normalization references.
  Compute $s_e = v^{(e)}_{\mathrm{Pair}}/r_{\mathrm{Pair}} + v^{(e)}_{\mathrm{Geo}}/r_{\mathrm{Geo}}$.
  if $s_e < s^\star$ **then**
    Save the shared encoder; $s^\star \leftarrow s_e$.
  end if
end for
return the selected shared encoder.

Here Θ denotes the optimized parameters.

Checkpoint selection. Nuc-pair and Nuc-geo select the checkpoint with the lowest corresponding validation loss across the 20 epochs. For Nuc-joint, the first completed epoch defines fixed references $r_{\mathrm{Pair}} = v^{(1)}_{\mathrm{Pair}}$ and $r_{\mathrm{Geo}} = v^{(1)}_{\mathrm{Geo}}$. The selected checkpoint minimizes the equally weighted normalized validation losses:

$$e^\star = \underset{e \in \{1,\ldots,20\}}{\arg\min} \left[\frac{v^{(e)}_{\mathrm{Pair}}}{r_{\mathrm{Pair}}} + \frac{v^{(e)}_{\mathrm{Geo}}}{r_{\mathrm{Geo}}} \right]. \tag{11}$$

Exact ties retain the earlier checkpoint.

B.2 Molecule-level Models

Secondary-structure encoder. Each RNA secondary structure is represented as a graph whose nodes correspond to nucleotides. Node features encode nucleotide identity, while edges represent backbone connectivity and annotated base-pair interactions. We use a two-layer Graph Isomorphism Network with Edge features (GINE) to perform edge-aware message passing. Global mean pooling followed by an MLP produces a normalized 640-dimensional secondary-structure representation.

Tertiary-structure encoder. For tertiary structure, we use a lightweight rotation- and translation-invariant GNN. A coordinate-based k-nearest-neighbor graph is constructed with $k = 16$, and nucleotide identities provide the initial node features. For each edge (i, j), the geometric features consist of a 16-dimensional radial basis function expansion of the Euclidean distance

$$d_{ij} = \|x_i - x_j\|_2, \tag{12}$$

together with the normalized relative sequence offset

$$\mathrm{clip}\left(\frac{j - i}{32}, -1, 1 \right), \tag{13}$$

which encodes their relative positions along the RNA sequence. Three message-passing layers aggregate these features, followed by global mean pooling and an MLP to obtain a normalized 640-dimensional representation. Coordinates enter the encoder only through pairwise distances used for neighborhood construction and edge features, making the representation invariant to global rigid rotations and translations.

Alignment objective. Each query-pooling module in Eq. (2) produces a 640-dimensional molecule-level representation. The bidirectional InfoNCE term in Eq. (3) is

$$\mathcal{L}_m^{\mathrm{NCE}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{\exp\left(a_{ii}^{(m)}\right)}{\sum_{j\in\mathcal{A}_i^{(m)}}\exp\left(a_{ij}^{(m)}\right)} + \log\frac{\exp\left(a_{ii}^{(m)}\right)}{\sum_{j\in\mathcal{A}_i^{(m)}}\exp\left(a_{ji}^{(m)}\right)}\right], \quad m\in\{2,3\}. \tag{14}$$

Here B is the batch size and $a_{ij}^{(m)}$ is the temperature-scaled cosine similarity between the projected sequence representation of record i and the structure representation of record j. The set $\mathcal{A}_i^{(m)}$ contains the samples retained after false-negative masking, including the matched positive.

False-negative masking. For 2D supervision, we compute length-normalized Levenshtein similarities for both the nucleotide sequence and the secondary-structure string. An off-diagonal pair (i, j) is excluded from the negative set only when

$$\mathrm{sim}_{\mathrm{seq}}(i,j)\geq 0.9 \quad \text{and} \quad \mathrm{sim}_{\mathrm{2D}}(i,j)\geq 0.9. \tag{15}$$

Pairs that are similar along only one of the two axes remain valid hard negatives.

For 3D supervision, false-negative masking is based on precomputed structure-equivalence clusters. RNA sequences are first grouped into sequence-homology clusters using MMseqs2. Within each sequence cluster, structures are further grouped using US-align (Zhang et al., 2022), with pairs considered structurally equivalent when their TM-score is at least 0.9. Each resulting 3D structure cluster is therefore nested within a sequence cluster. During contrastive training, samples assigned to the same structure cluster are excluded from one another's negative sets, whereas samples from the same sequence cluster but different structure clusters are retained as hard negatives. The matched diagonal sequence–structure pairs are always retained as positives.

Trainable parameters and objective. Mol-pair and Mol-geo learn from secondary- and tertiary-structure data, respectively, while Mol-joint shares the RNA-FM backbone across both modalities. Only the last two RNA-FM Transformer layers are unfrozen. The query-pooling modules, projection heads, structure encoders, and masked-language-modeling head are trained jointly with these layers. For each modality, training minimizes bidirectional InfoNCE with false-negative masking together with a masked-language-modeling objective weighted by 0.15. Canonical nucleotide positions are independently selected for masking with probability 0.15, and the sequence embeddings used for contrastive alignment are computed from the same masked input. Each modality has a learnable temperature, initialized to 0.5 and constrained to $[0.05, 1.0]$.

Joint optimization and batching. For Mol-joint, both structure encoders are initialized randomly. AdamW uses separate learning rates for the unfrozen RNA-FM layers, the remaining sequence-side modules, and the structure encoders (Table 5). Learning rates follow a cosine schedule over the allowed training horizon. Automatic mixed precision is used on CUDA, and gradients are unscaled before clipping their norm to 1.

Table 5: Optimization settings for Mol-joint.

Setting	Value
RNA-FM learning rate	1×10^{-5}
Sequence-side module learning rate	5×10^{-5}
Structure-encoder learning rate	1×10^{-4}
Weight decay	1×10^{-5}
2D / 3D batch size	48 / 12
2D / 3D updates per complete cycle	5 / 4
Maximum epochs / patience	100 / 12
Query-pooling attention heads	8
Query-pooling and projection dropout	0.1
Random seed	42

Training uses separate length-bucketed minibatch streams for 2D and 3D data. Each minibatch is processed independently and followed by one optimizer update. A cycle only specifies how many consecutive 2D and 3D updates are scheduled. The modality-specific batch sizes and update counts

per cycle are jointly chosen so that the effective sample exposure approximately follows the 5:1 size ratio of the two training sets. Batches are shuffled each epoch, and training proceeds only through complete minibatches and complete update cycles supported by both streams.

Validation and checkpoint selection. Validation traverses both validation sets while retaining incomplete batches. For each modality, the total loss is averaged over validation batches, and the joint selection score is the arithmetic mean of the 2D and 3D averages. Validation uses the same masking procedure and loss construction as training, with dropout disabled. The checkpoint is saved whenever the joint score strictly improves, and training stops after 12 consecutive epochs without improvement. Test-set and downstream performance are not used for checkpoint selection. Algorithm 2 summarizes the joint training procedure.

Algorithm 2 Joint molecule-level training

Require: 2D and 3D training/validation sets; pretrained RNA-FM; settings in Table 5
Freeze RNA-FM except its last two Transformer layers.
Initialize the query modules, projections, structure encoders, MLM head, and temperatures.
Initialize AdamW, the cosine scheduler, and mixed-precision scaling.
$v^\star \leftarrow +\infty$; $bad \leftarrow 0$.
for each training epoch **do**
    Shuffle the two length-bucketed training streams.
    for each complete update cycle supported by both streams **do**
        for each scheduled modality-specific minibatch update **do**
            Draw the next batch from the corresponding modality.
            Mask nucleotide tokens and encode the masked sequences.
            Encode the corresponding structures and apply false-negative masking.
            Compute bidirectional InfoNCE plus 0.15 times the MLM loss.
            Backpropagate, unscale, clip gradients to norm 1, and take one AdamW step.
        end for
    end for
    Evaluate each modality and average its validation batch losses.
    $v \leftarrow$ arithmetic mean of the two validation losses.
    Advance the cosine learning-rate scheduler.
    if $v < v^\star$ **then**
        Save the model; $v^\star \leftarrow v$; $bad \leftarrow 0$.
    else
        $bad \leftarrow bad + 1$.
    end if
    if $bad \geq 12$ **then**
        break
    end if
end for
return the checkpoint with the best joint validation score.

Molecule-level downstream representation. For downstream tasks, Mol-pair and Mol-geo use their corresponding sequence-derived molecule-level representations, r_2 and r_3, respectively. The upstream encoders remain frozen during downstream training. Mol-joint exports the shared sequence representation b together with the 2D- and 3D-specific sequence-derived residuals δ_2 and δ_3. These components are projected independently and combined through task-specific sigmoid gates:

$$z_{\mathrm{mol}} = \mathrm{LN}\left(\mathrm{Linear}_b(b) + \sum_{m\in\{2,3\}} \sigma(g_m)\,\mathrm{Linear}_m(\delta_m)\right). \tag{16}$$

The projections used here are distinct from the upstream contrastive projection heads. Each g_m is a learned scalar shared across samples within a fitted downstream task.

During downstream training of Mol-joint, the 3D contribution is independently dropped for each sample with probability 0.5 to prevent over-reliance on the 3D branch. When retained, it is rescaled by the inverse keep probability. The training objective combines the main prediction MSE with an auxiliary MSE computed without the 3D contribution and a regularization term that limits the relative magnitude of the 3D contribution, weighted by 0.5 and 10^{-3}, respectively. At validation and inference, all structural branches are retained without stochastic dropout.

B.3 MOLECULE-NUCLEOTIDE HYBRID

RiboWeave Hybrid combines z_{nuc} from Nuc-joint with z_{mol} from Mol-joint. For activity prediction, the fitted Nuc-joint and Mol-joint branches are kept frozen during Hybrid training. Only the cross-granularity fusion layers and the downstream prediction head are optimized jointly for each task.

B.4 ACTIVITY PREDICTION MODELS

Prediction heads and optimization. Each target and representation configuration has its own prediction head. The sequence representation is mapped to a 640-dimensional input and passed through a $640 \to 320 \to 160 \to 1$ multilayer perceptron. The hidden layers use LayerNorm, GELU, and dropout 0.1. Predictors are trained on enrichment-derived pseudo-activity scores using mean squared error. AdamW is used with a constant learning rate of 3×10^{-4} and weight decay 10^{-4}.

Training and checkpoint selection. All activity prediction models are trained for up to 200 epochs. Checkpoints are selected by validation MSE, with early stopping after 10 consecutive epochs without improvement and a minimum improvement threshold of 10^{-5}. Training and checkpoint selection use only the predefined training and validation partitions, while the test set is reserved for final evaluation. Downstream comparisons are repeated with seeds 13, 29, 43, 71, and 101.

B.5 GENERATION MODELS

Training stages and architecture. Generator training proceeds in two stages. We first fit a target-specific activity predictor on the development set. This stage learns any downstream representation mapping required by the corresponding configuration, including the 2D/3D modality fusion in Mol-joint and the additional nucleotide–molecule fusion in Hybrid. After fitting, the prediction head is no longer used for generation. The learned representation mapping is frozen and provides a fixed 640-dimensional, task-specific representation space for all subsequent generator training.

A trainable adapter then maps the frozen representation into a 128-dimensional latent code and contains embedding-reconstruction and latent activity heads. An autoregressive decoder is trained jointly with the adapter to reconstruct the variable-region sequence conditioned on the latent code. The decoder has hidden dimension 128, two Transformer layers, two attention heads, and one latent conditioning token, and is trained with teacher forcing. Each target, representation configuration, and generation seed has separately trained adapter and decoder weights.

Training objective. The overall objective combines representation and sequence reconstruction, latent-space regularization, and auxiliary guidance:

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{primary}} + \mathcal{L}_{\mathrm{guide}}, \tag{17}$$

$$\mathcal{L}_{\mathrm{primary}} = 2\mathcal{L}_{\mathrm{embrec}} + \mathcal{L}_{\mathrm{seq}} + 0.05\mathcal{L}_{\mathrm{var}} + 0.005\mathcal{L}_{\mathrm{cov}}, \tag{18}$$

$$\begin{aligned}\mathcal{L}_{\mathrm{guide}} = {} & \lambda_{\mathrm{reg}}(e)\mathcal{L}_{\mathrm{reg}} + \lambda_{\mathrm{rank}}(e)\mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{metric}}(e)\mathcal{L}_{\mathrm{metric}} \\ & + \lambda_{\mathrm{local}}(e)\mathcal{L}_{\mathrm{local}} + \lambda_{\mathrm{interp}}(e)\mathcal{L}_{\mathrm{interp}},\end{aligned} \tag{19}$$

where e denotes the training epoch. The guidance component combines pseudo-activity supervision with neighborhood and interpolation consistency.

For a minibatch of B sequences, let $\mathbf{u}_i \in \mathbb{R}^{640}$ denote the frozen task-specific representation of x_i, standardized using statistics from the generator-training split only. The adapter encoder produces $\mathbf{z}_i = E_\phi(\mathbf{u}_i) \in \mathbb{R}^{128}$. This latent code supports three outputs: representation reconstruction $\widehat{\mathbf{u}}_i = D_\psi(\mathbf{z}_i)$, pseudo-activity prediction $\widehat{y}_i = r_\omega(\mathbf{z}_i)$ through a linear head, and variable-region sequence reconstruction through an autoregressive decoder p_θ.

Representation reconstruction. We reconstruct the standardized representation using smooth-L_1 loss and cosine distance:

$$\mathcal{L}_{\mathrm{embrec}} = \frac{1}{B}\sum_{i=1}^{B}\left[\mathrm{SmoothL1}(\widehat{\mathbf{u}}_i, \mathbf{u}_i) + 1 - \cos(\widehat{\mathbf{u}}_i, \mathbf{u}_i)\right], \tag{20}$$

where $\mathrm{SmoothL1}$ uses transition parameter $\beta = 1$ and averages over representation dimensions.

Sequence reconstruction. The autoregressive decoder reconstructs the variable region conditioned on the same latent code:

$$\mathcal{L}_{\text{seq}} = -\frac{1}{BL_{\text{var}}} \sum_{i=1}^{B} \sum_{t=1}^{L_{\text{var}}} \log p_\theta(x_{i,\text{var},t} \mid x_{i,\text{var},<t}, \mathbf{z}_i)\,. \tag{21}$$

Here $L_{\text{var}} = 20$, and fixed flanking sequences are excluded from the reconstruction targets. Training uses teacher forcing, with a beginning-of-sequence token at the first position and the ground-truth prefix at subsequent positions.

Variance and covariance regularization. The variance loss discourages latent collapse by penalizing dimensions with minibatch standard deviations below one:

$$\mathcal{L}_{\text{var}} = \frac{1}{d_z} \sum_{k=1}^{d_z} \max(0, 1 - s_k), \qquad s_k = \sqrt{\frac{1}{B} \sum_{i=1}^{B} (z_{ik} - \bar{z}_k)^2 + 10^{-4}}, \tag{22}$$

where $\bar{z}_k = B^{-1} \sum_{i=1}^{B} z_{ik}$, B is the minibatch size, and $d_z = 128$.

The covariance loss reduces linear redundancy between latent dimensions by penalizing off-diagonal covariances:

$$\mathcal{L}_{\text{cov}} = \frac{1}{d_z} \sum_{k \neq l} C_{kl}^2, \qquad C_{kl} = \frac{\sum_{i=1}^{B} (z_{ik} - \bar{z}_k)(z_{il} - \bar{z}_l)}{\max(B - 1, 1)}. \tag{23}$$

Activity regression. We fit enrichment-derived pseudo-activity scores using smooth-L_1 loss with transition parameter $\beta = 1$:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B} \sum_{i=1}^{B} \text{SmoothL1}(\widehat{y}_i, y_i). \tag{24}$$

Activity ranking. The ranking loss encourages sequences with higher pseudo-activity labels to receive higher predicted scores. Let q_i denote the fraction of generator-training samples whose labels do not exceed y_i:

$$q_i = \frac{1}{N_{\text{fit}}} \sum_{j \in \mathcal{D}_{\text{fit}}} \mathbb{I}[y_j \leq y_i], \tag{25}$$

where $\mathcal{D}_{\text{fit}}$ contains N_{fit} generator-training samples. This training-set reference is also used to compute quantiles for validation samples.

Within each minibatch, we compare pairs whose activity quantiles differ by at least 0.2, corresponding to 20 percentile points:

$$\mathcal{P}_{\text{rank}} = \{(i, j) : i < j,\ |q_i - q_j| \geq 0.2\}.$$

The condition $i < j$ avoids counting each pair twice. We define

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}_{\text{rank}}|} \sum_{(i,j) \in \mathcal{P}_{\text{rank}}} \text{softplus}\left(-\frac{\text{sgn}(q_i - q_j)(\widehat{y}_i - \widehat{y}_j)}{\tau_{\text{rank}}} \right), \tag{26}$$

where $\tau_{\text{rank}} = 0.1$ scales the predicted score difference and $\text{softplus}(a) = \log(1 + \exp(a))$. The sign term specifies the desired ordering: when $q_i > q_j$, the loss decreases as $\widehat{y}_i - \widehat{y}_j$ increases; when $q_i < q_j$, the reverse ordering is encouraged.

Activity-aware metric learning. The metric loss directly organizes latent representations by pseudo-activity: it pulls together activity-similar pairs and encourages a minimum separation between activity-dissimilar pairs. Using the training-set activity quantiles q_i defined above, we construct two sets of minibatch pairs:

$$\mathcal{P}_+ = \{(i, j) : i < j,\ |q_i - q_j| \leq 0.05\},$$
$$\mathcal{P}_- = \{(i, j) : i < j,\ |q_i - q_j| \geq 0.5\}.$$

Thus, positive pairs differ by at most 5 percentile points, whereas negative pairs differ by at least 50 percentile points. Distances are measured between ℓ_2-normalized latent codes:

$$d_{ij} = \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \right\|_2.$$

With a separation margin $m = 0.8$, the loss is

$$\mathcal{L}_{\text{metric}} = \frac{1}{|\mathcal{P}_+|} \sum_{(i,j)\in\mathcal{P}_+} d_{ij}^2 + \frac{1}{|\mathcal{P}_-|} \sum_{(i,j)\in\mathcal{P}_-} \max(0, m - d_{ij})^2. \tag{27}$$

The first term reduces distances between positive pairs. The second penalizes negative pairs only when their distance is below m, with no further penalty once the margin is reached. Pairs outside both sets are excluded from this loss. Averages over empty pair sets are defined as zero in both the ranking and metric losses.

Local consistency. To preserve local geometry through the adapter, we construct neighborhood references using only the generator-training split. Standardized representations are projected into a PCA space with up to 50 components. For each training sample, one reference $n(i)$ is selected uniformly from its 20 nearest training neighbors; validation references are selected from the training set in the same way. The selected references remain fixed within each run.

We match the cosine distance between each sample and its reference before and after latent encoding:

$$\mathcal{L}_{\text{local}} = \frac{1}{B} \sum_{i=1}^{B} \text{SmoothL1}\left(1 - \cos(\mathbf{z}_i, \mathbf{z}_{n(i)}), 1 - \cos(\mathbf{u}_i, \mathbf{u}_{n(i)})\right), \tag{28}$$

where $\mathbf{z}_{n(i)} = E_\phi(\mathbf{u}_{n(i)})$.

Interpolation consistency. To encourage smooth and locally consistent mappings between the representation and latent spaces, we construct matched interpolations between each sample and its selected neighbor. For a standardized representation $\mathbf{u}_i$ and its neighbor $\mathbf{u}_{n(i)}$, their latent codes are $\mathbf{z}_i = E_\phi(\mathbf{u}_i)$ and $\mathbf{z}_{n(i)} = E_\phi(\mathbf{u}_{n(i)})$. For each pair, we independently sample $\alpha_i \sim \mathcal{U}(0, 1)$ and use the same interpolation coefficient in both spaces:

$$\begin{aligned} \mathbf{z}_i^\alpha &= (1 - \alpha_i)\mathbf{z}_i + \alpha_i \mathbf{z}_{n(i)}, \\ \mathbf{u}_i^\alpha &= (1 - \alpha_i)\mathbf{u}_i + \alpha_i \mathbf{u}_{n(i)}. \end{aligned} \tag{29}$$

Thus, $\mathbf{z}_i^\alpha$ and $\mathbf{u}_i^\alpha$ are matched interpolation points between the same sample–neighbor pair in the latent and representation spaces, respectively.

We decode the interpolated latent code as $\widehat{\mathbf{u}}_i^\alpha = D_\psi(\mathbf{z}_i^\alpha)$. The interpolation objective requires this decoded point to reconstruct the corresponding representation-space interpolation and, after re-encoding, to recover the original interpolated latent code:

$$\begin{aligned} \mathcal{L}_{\text{interp}} = \frac{1}{B} \sum_{i=1}^{B} \Big[&\text{SmoothL1}(\widehat{\mathbf{u}}_i^\alpha, \mathbf{u}_i^\alpha) + 1 - \cos(\widehat{\mathbf{u}}_i^\alpha, \mathbf{u}_i^\alpha) \\ &+ 1 - \cos(E_\phi(\widehat{\mathbf{u}}_i^\alpha), \mathbf{z}_i^\alpha) \Big]. \end{aligned} \tag{30}$$

The first two terms apply the same reconstruction criterion as $\mathcal{L}_{\text{embrec}}$ to interpolated points, encouraging linear interpolation in latent space to decode to the corresponding linear interpolation in representation space. The final term imposes decode–encode cycle consistency: the decoded interpolation $\widehat{\mathbf{u}}_i^\alpha$ is re-encoded by E_ϕ, and the resulting latent code is encouraged to remain aligned with the original interpolated code $\mathbf{z}_i^\alpha$.

Guidance schedule. With epochs indexed from one, define

$$\lambda_{\text{reg}}(e) = \mathbb{I}[e > 5], \qquad \gamma(e) = \min\left\{1, \max\left\{0, \frac{e - 10}{10}\right\}\right\}. \tag{31}$$

The ranking, metric, local, and interpolation weights are $0.10\gamma(e)$, $0.03\gamma(e)$, $0.10\gamma(e)$, and $0.10\gamma(e)$, respectively. Regression therefore starts at epoch 6, and the remaining guidance terms are ramped up over epochs 11–20. For the adapter encoder, primary and guidance gradients are computed separately and combined using the gradient-conflict procedure described below, rather than using the unmodified gradient of Eq. 17.

Optimization and gradient handling. The adapter and autoregressive decoder are jointly optimized with AdamW using a constant learning rate of 3×10^{-4}, weight decay 10^{-2}, and gradient-norm clipping at 1. Training runs for at most 200 epochs. For the adapter encoder, gradients from $\mathcal{L}_{\text{primary}}$ and $\mathcal{L}_{\text{guide}}$ are computed separately. When their inner product is negative, the component of the guidance gradient that conflicts with the primary gradient is removed. The adapter-encoder update then combines the primary gradient with 0.5 times the adjusted guidance gradient. This adjustment is applied only to the adapter encoder. The reconstruction head, latent activity head, and autoregressive decoder receive gradients from their corresponding objectives.

Checkpoint selection. Checkpoint selection uses only the validation split of the development set. To ensure that the learned latent space remains suitable for generation, a checkpoint is considered eligible only if it preserves the input representation, supports stable interpolation, and avoids latent collapse. Specifically, the mean embedding-reconstruction cosine similarity must be at least 0.90, the mean interpolation cycle-consistency cosine similarity must be at least 0.95, and at least 90% of latent dimensions must have validation standard deviation greater than 0.1. Cycle consistency is evaluated by decoding a latent midpoint back to the representation space and re-encoding it, then comparing the recovered latent code with the original midpoint.

Among eligible checkpoints, validation sequence cross-entropy is used to select the model with the best autoregressive reconstruction quality, with a minimum improvement of 10^{-4}. Only eligible epochs without sufficient improvement increment the patience counter, while ineligible epochs leave it unchanged. Training may stop once at least 20 epochs have been completed and the patience counter reaches 10, or when the maximum training horizon is reached.

Latent-space generation and decoding. Generation begins by selecting pairs of high-pseudo-activity parent sequences from the training split of the development set and using the midpoint of their latent codes as the initial proposal. The linear latent activity head defines a direction toward higher predicted pseudo-activity, and the proposal is moved along this direction with per-dimension scaling based on the training latent distribution.

To keep the optimized code close to regions supported by the training data, we impose a neighborhood constraint based on cosine distances to nearby training latent codes. If the proposed displacement moves the code too far from this latent manifold, the step size is progressively reduced by backtracking until the constraint is satisfied. If no acceptable proposal is found within the backtracking budget, the latent code of the higher-activity parent is used instead. The resulting code is then passed to the frozen autoregressive decoder to sample the variable-region sequence, after which the fixed flanking sequences are restored.

Algorithm 3 Adapter–decoder training for a representation configuration

Require: Frozen generator-side representation mapping; training and validation splits of the development set
Extract representations and fit standardization on the training split only.
Compute activity quantiles and neighborhood references from the training split.
Initialize the adapter, including its latent activity head, the autoregressive decoder, and AdamW.
$c^\star \leftarrow +\infty$; $bad \leftarrow 0$; selected checkpoint $\leftarrow$ none.
for $e = 1, \ldots, 200$ **do**
    for each minibatch from the training split **do**
        Encode representations into latent codes and reconstruct the inputs.
        Predict pseudo-activity and decode sequences with teacher forcing.
        Compute $\mathcal{L}_{\text{primary}}$ and $\mathcal{L}_{\text{guide}}$.
        Compute their gradients separately on the adapter encoder.
        Remove any component of the guidance gradient that opposes the primary gradient.
        Combine the primary gradient with 0.5 times the adjusted guidance gradient on the adapter encoder.
        Retain the corresponding gradients for the remaining parameters.
        Clip the combined gradient norm to 1 and take one AdamW step.
    end for
    Evaluate validation sequence cross-entropy c and the three latent-space eligibility criteria.
    if all eligibility criteria are satisfied **then**
        if $c < c^\star - 10^{-4}$ **then**
            Save the checkpoint; $c^\star \leftarrow c$; $bad \leftarrow 0$.
        else
            $bad \leftarrow bad + 1$.
        end if
    end if
    if $e \geq 20$ and $bad \geq 10$ **then**
        break
    end if
end for
if no checkpoint was selected **then**
    Reject the run.
else
    Reload the selected adapter and decoder for generation.
end if

C Evaluation and Control Protocols

C.1 Cross-modal Evaluation

The nucleotide- and molecule-level evaluations draw on the same structural datasets. Evaluation datasets are described in Appendix A.3. The descriptions below specify representations, readouts, and metrics.

C.1.1 Nucleotide-level Evaluation

Representations and readouts. We evaluate frozen 640-dimensional nucleotide representations from RNA-FM, Nuc-pair, Nuc-geo, and Nuc-joint, together with their upstream permutation controls. All representations use the same training, validation, and test partitions within each task.

Both readouts apply non-affine layer normalization to nucleotide representations. The pairing readout uses symmetric features $[z_i + z_j;\ |z_i - z_j|;\ z_i \odot z_j]$, followed by a 256-unit GELU hidden layer. The distance readout uses $[z_i; z_j;\ |z_i - z_j|;\ z_i \odot z_j]$ and two 512-unit GELU hidden layers. Within each task, all representations use the same readout architecture, training settings, and training budget. Both readouts use AdamW with batches of 16 records; learning rates are 3×10^{-4} and 1.5×10^{-4}, and weight decay is 10^{-4} and 10^{-5}, for pairing and distance prediction, respectively. Losses are averaged within records and then across each batch. We evaluate the final readout checkpoint from each run.

Targets and sampling. Pairing labels are obtained from BPSEQ annotations. Eligible pairs satisfy $j - i \geq 4$ and have an AU, UA, GC, CG, GU, or UG base combination. Training samples one negative per positive, preferentially matching the base combination and sequence-separation bucket.

Distance labels are computed from observed C1′ coordinates for pairs with $j - i \geq 4$. Training samples at most 512 pairs per record, allocating up to 40% to contacts and 20% to sequence separations of 4–16, with the remaining pairs sampled from unselected candidates. Evaluation scores all eligible pairs.

Contacts are defined by C1′–C1′ distances of at most 8 Å, and contact scores are the predicted probabilities of the first distance bin. This cutoff is below the approximately 10.5–10.6 Å mean distances reported for canonical Watson–Crick pairs, making contact prediction distinct from direct pairing recognition (Kailasam et al., 2014, Table 3).

Metrics and reporting. We compute average precision for each record, average within source groups, and then average equally across groups to obtain cluster-macro AUPRC. Groups are defined by the source `folder_id`. Results are reported as mean ± sample standard deviation across five readout seeds, with upstream checkpoints held fixed.

Upstream permutation controls follow Appendix C.4; their cross-modal results appear in Appendix D.1.1.

C.1.2 Molecule-level Evaluation

We evaluated whether sequence representations learned with one structural modality also preserve relationships defined by the other. For each modality, we used the same RNA records across Mol-pair, Mol-geo, Mol-joint, and RNA-FM. The 2D reference embeddings were obtained from the frozen structure encoder of Mol-pair, and the 3D reference embeddings from that of Mol-geo. Each reference was held fixed across all model comparisons. Mol-pair and Mol-geo were evaluated using their respective projected sequence embeddings. Mol-joint used its 2D or 3D projected track for the corresponding reference and RNA-FM used mean-pooled nucleotide representations.

For each RNA pair, we computed cosine similarity separately in the model-representation space and the structural-reference space, and compared the resulting similarity relationships among the same RNAs. This evaluation captures both cross-structural transfer and joint retention of structural information: Mol-pair is compared with the 3D reference and Mol-geo with the 2D reference to assess transfer across modalities, while Mol-joint is compared with both references to evaluate how well joint supervision preserves relationships associated with each structural modality. RNA-FM serves as the sequence-only baseline. We quantify agreement using the metrics described below.

Nearest-neighbor overlap. For each RNA, we identify its ten nearest neighbors by cosine similarity in the model-representation space and in the structural-reference space, excluding the RNA itself. KNN@10 is the fraction of neighbors shared between the two sets, averaged across all evaluated RNAs. It measures how well the model representation preserves local neighborhoods defined by the structural reference.

Linear CKA. Linear centered kernel alignment (CKA) measures global agreement between the two representation geometries. We construct linear Gram matrices from the model and structural-reference embeddings, center both matrices, and compute their normalized Frobenius inner product. Unlike KNN@10, CKA captures agreement over the overall representation geometry rather than only local neighborhoods.

Pairwise Spearman correlation. We compute cosine-similarity matrices in the model-representation and structural-reference spaces, extract the corresponding upper-triangular off-diagonal entries, and calculate Spearman correlation between the two vectors. This measures agreement in the ranking of pairwise RNA similarities without requiring their numerical values to match. Higher values indicate stronger agreement for all three metrics.

C.2 Activity Prediction and Controls

For each target and model, we trained a prediction head and any fusion modules using the training partition, while keeping the upstream encoders frozen. Training and checkpoint selection follow Appendix B.4. The labels were pseudo-activity scores. We reported test-set R^2 for each target.

Main-text macro R^2 is the unweighted mean across CD3 and cMYC and Appendix D.3 additionally reports the three-target macro average. Macro scores were calculated within each seed before reporting the mean and sample standard deviation across seeds.

We additionally evaluated two single-modality cross-granularity variants: Pair-hybrid combines Nuc-pair and Mol-pair, while Geo-hybrid combines Nuc-geo and Mol-geo. To control for added readout capacity, we duplicated either the Nuc-joint or Mol-joint representation while retaining the same concatenation–projection block and training protocol. All variants used the same data partitions and downstream seeds.

C.3 Generation Evaluation

Candidate scoring. RNA-FM, the six structure-supervised models, and Hybrid used the same generation framework, with adapters and decoders trained separately for each configuration. We generated 1,000 candidates per target, configuration, and seed, yielding 120,000 candidates across three targets, eight configurations, and five generation seeds. Candidates were locked before scoring. For each target, the same panel of eight evaluator families, each trained with five seeds, scored every generator's candidates. All evaluators were trained using the final evaluator training set. Panel scores are the unweighted mean of evaluator outputs, including the evaluator family matching the generator.

Mean and top-ten scores. For each target, configuration, and generation seed, we reported the mean panel score over all 1,000 candidates and the mean score of the ten highest-scoring candidates, ranked by that same panel. These summarize the average predicted pseudo-activity and the upper-scoring subset, respectively.

Sequence novelty. Each candidate was compared with the sequences in the development set using its variable region, excluding the fixed flanks. We recorded its maximum sequence identity to any development-set sequence. Novel@0.75 is the fraction of candidates whose maximum identity is at most 0.75.

Sequence diversity. We measure within-set sequence diversity using the normalized number of single-linkage clusters. For each generated set of $N = 1{,}000$ sequences, we construct an undirected graph in which two sequences are connected if their variable-region sequence identity is at least τ. Let K_τ denote the number of connected components. We define

$$\text{Diversity } @\tau = \frac{K_\tau}{N}. \tag{32}$$

We report Diversity@0.75 as a percentage, with higher values indicating more sequence clusters at the specified identity threshold. The metric is computed separately for each target, configuration, and generation seed, and summarized across five generation seeds.

Activity–novelty analysis. For each target, configuration, and generation seed, we filtered candidates by their maximum sequence identity to the development set, using thresholds from 0.60 to 0.90 in increments of 0.05. At each threshold, we calculated the fraction of the 1,000 candidates retained and their mean evaluator score. We report retained fractions over the full threshold range, but restrict activity comparisons to 0.65–0.90 because very few candidates satisfy thresholds below 0.65. Results were summarized as the mean and sample standard deviation across five generation seeds.

Evaluator reliability. We assessed evaluator predictions on the natural holdout set using R^2 and Spearman. Detailed results are reported in Appendix D.6. Candidate scores estimate pseudo-activity scores and are not experimental measurements of generated aptamer activity.

C.4 Structural Supervision and Predicted-Structure Controls

Control design. We evaluate three types of controls with distinct purposes. Upstream supervision permutations test the role of correspondence between structural targets and sequence positions.

Folding summaries test whether simple predicted-structure features predict activity scores on their own or complement RNA-FM. Direct graph inputs test the incremental benefit of adding predicted pairing edges to a backbone graph. Downstream comparisons use the same CD3 and cMYC training, validation, and test splits as the main activity-prediction experiment. This set of ablations focuses on these two targets because RBD prediction remains weak across models under the sequence-similarity-based split (Table 12). Results appear in Table 14, with the correctly supervised Nuc-pair and Nuc-geo results in Table 12.

Upstream supervision permutations. For Nuc-pair, pair permutation remaps pairing relationships among positions of the same nucleotide type. For Nuc-geo, target-row permutation independently shuffles valid labels within each target channel across the batch, while coordinate permutation reassigns complete coordinate blocks and their masks among observed positions within each record, without fixed points, and then recomputes geometric targets. These controls disrupt different aspects of supervision alignment. Each correctly supervised model and its corresponding controls share the initialization seed, data splits, batch-sampling schedule, optimizer settings, and a 20-epoch upstream training budget, updating the final six RNA-FM layers and the adapter. Permutations affect training supervision only and validation uses correctly aligned targets. Each encoder is selected by its lowest validation loss and then frozen for downstream prediction.

Folding summaries. We fold each 43-nucleotide sequence using ViennaRNA 2.7.2 (Lorenz et al., 2011) at 37°C with `dangles=2`. The resulting 17-dimensional summary covers four groups of features: energies (length-normalized minimum free energy and ensemble free energy); pairing statistics (mean, standard deviation, and maximum per-position pairing probability, mean unpaired probability, mean pairing entropy, and maximum pair probability); pair spans (probability-weighted normalized span and the probability-mass fraction at spans of at most 12 nucleotides); and structural contexts (the fraction of paired positions in the minimum-free-energy structure and proportions of six context categories). Features are standardized using statistics fitted on each task's training split. We evaluate the summary alone and concatenated with the frozen 640-dimensional RNA-FM representation; each input is projected to 640 dimensions and passed to the same prediction MLP.

Direct graph inputs. We compare two RNA-FM+GINE configurations that differ only in whether predicted pairing edges are included. Both use four-dimensional nucleotide one-hot node features and bidirectional backbone edges between adjacent positions; the second adds bidirectional pairing edges from the ViennaRNA minimum-free-energy structure. Edge attributes encode connection type and direction. The graph branch contains two GINE layers with hidden dimension 128, followed by global mean pooling, an output mapping, and L2 normalization to produce a 640-dimensional representation. Each branch is projected to 320 dimensions; the outputs are concatenated, normalized, and passed to the prediction MLP. RNA-FM remains frozen, while the randomly initialized GINE branch, fusion layers, and prediction MLP are trained on downstream activity labels. The two configurations use matched initial parameters and per-epoch sample orders for each seed.

Downstream training and reporting. All controls use the standard prediction MLP and AdamW settings in Appendix B.4. Checkpoint selection and early stopping follow the same validation-MSE rule. Test R^2 is reported as mean and sample standard deviation across the five downstream seeds specified there. Macro R^2 is the mean of the CD3 and cMYC scores within each seed, summarized across seeds in the same way.

D Additional Results and Ablations

D.1 Cross-modal Evaluation

D.1.1 Nucleotide-level Results

Both single-modality experts outperform RNA-FM on the other structural task. Among these two experts, Nuc-pair performs better on pairing prediction, whereas Nuc-geo performs better on contact prediction. Nuc-joint achieves the highest mean contact AUPRC, while its pairing AUPRC remains below Nuc-pair.

Correctly aligned supervision also outperforms the corresponding permutation controls in cross-modal evaluation. These results support a role for supervision alignment in cross-modal transfer.

Table 6: Nucleotide-level structural readout results. Values are cluster-macro AUPRC. Learned readouts are reported as mean ± sample standard deviation across five readout seeds, with upstream checkpoints fixed. Permutations are applied during upstream training only.

Representation	Pairing AUPRC	Contact AUPRC
Random scoring	0.0153 ± 0.0002	0.0197 ± 0.0005
RNA-FM	0.5574 ± 0.0223	0.0454 ± 0.0020
Nuc-pair	$\mathbf{0.9385} \pm 0.0022$	0.0929 ± 0.0050
Nuc-geo	0.7370 ± 0.0027	0.1059 ± 0.0036
Nuc-joint	0.8397 ± 0.0023	$\mathbf{0.1110} \pm 0.0027$
Nuc-pair, pair permutation	0.3249 ± 0.0152	0.0813 ± 0.0050
Nuc-geo, row permutation	0.0873 ± 0.0012	0.0305 ± 0.0010
Nuc-geo, coordinate permutation	0.2895 ± 0.0265	0.0304 ± 0.0012

The smaller 3D training set may partly explain the lower contact AUPRC. It contains 17,098 records, compared with 84,579 for secondary-structure supervision, providing fewer examples for learning sequence–geometry relationships.

Table 7: 2D AUPRC by cohort. Values use cluster-macro aggregation; learned readout values are means ± standard deviations over five readout seeds with upstream checkpoints fixed. Permutations are applied during upstream training only.

Representation	5S	16S
Random scoring	0.0162 ± 0.0002	0.0035 ± 0.0001
RNA-FM	0.5897 ± 0.0231	0.1530 ± 0.0180
Nuc-pair	0.9554 ± 0.0023	0.7279 ± 0.0027
Nuc-geo	0.7687 ± 0.0034	0.3403 ± 0.0101
Nuc-joint	0.8666 ± 0.0023	0.5031 ± 0.0038
Nuc-pair, pair permutation	0.3464 ± 0.0162	0.0547 ± 0.0022
Nuc-geo, row permutation	0.0934 ± 0.0013	0.0103 ± 0.0004
Nuc-geo, coordinate permutation	0.3098 ± 0.0287	0.0357 ± 0.0010

Table 6 reports full-test results, while Table 7 breaks down pairing AUPRC by the 5S and 16S cohorts.

D.1.2 Molecule-level Results

Table 8 reports agreement with the fixed 2D and 3D structural references under the protocol in Appendix C.1.2. Both single-modality models exceed RNA-FM on all three metrics against the opposite structural reference. Mol-joint exceeds RNA-FM against both references, while remaining below the corresponding single-modality specialist on each metric.

Table 8: Molecule-level cross-modal evaluation. Metrics quantify agreement between model representations and fixed structural references in molecular similarity relationships.

Representation	2D			3D		
	kNN@10	CKA	Spearman	kNN@10	CKA	Spearman
RNA-FM	0.1881	0.2257	0.2561	0.3809	0.2644	0.0531
RiboWeave Mol-pair	0.4670	0.7982	0.8029	0.4274	0.4237	0.3462
RiboWeave Mol-geo	0.2383	0.4326	0.4368	0.4550	0.7895	0.7429
RiboWeave Mol-joint	0.4153	0.7431	0.7433	0.4522	0.6910	0.5968

D.2 Score Distribution and Sequence-Neighborhood Diagnostics of the Activity Data

We characterize score distributions, training–test sequence neighborhoods, and high-score sequence-cluster coverage within the evaluation partitions used in this study.

Data partitions and score distributions. Sequences have length 43, with a fixed 9-position prefix and 14-position suffix, leaving 20 variable positions. Sequence clusters are connected components under at least 90% identity over these variable positions, with each cluster assigned entirely to training, validation, or test.

Table 9 summarizes sample counts and score distributions within the actual partitions. Training, validation, and test distributions are similar within each task, but RBD differs markedly from CD3 and cMYC: its median score is approximately 0.081, about 55% of records have scores at most 0.1, and fewer than 2.4% exceed 0.8.

Table 9: Sample counts and score distributions in the training, validation, and test partitions used in this paper. Percentages are computed within each partition.

Target	Partition	Records	Median	$y \leq 0.1$ (%)	$y > 0.5$ (%)	$y > 0.8$ (%)
CD3	Train	21,981	0.6733	0.09	78.04	24.79
CD3	Validation	4,700	0.6742	0.11	78.02	24.83
CD3	Test	4,700	0.6742	0.02	78.06	24.70
cMYC	Train	28,172	0.5896	0.01	72.52	5.73
cMYC	Validation	6,050	0.5898	0.02	72.46	5.77
cMYC	Test	6,039	0.5895	0.02	72.45	5.73
RBD	Train	27,093	0.0811	55.38	9.77	2.37
RBD	Validation	5,815	0.0812	55.51	9.58	1.65
RBD	Test	5,790	0.0808	55.23	9.34	1.90

Sequence neighborhoods under the evaluation partitions. For each test sequence, we identify its nearest neighbor in the corresponding training set by Hamming distance over the 20 variable positions. Ties are resolved by the first record in training-file order. The neighbor's score is used directly as the prediction, without fitting parameters.

Across all three tasks, every test sequence differs from its nearest training neighbor at at least three positions. The fractions differing at five or more positions are 98.23%, 98.31%, and 98.12% for CD3, cMYC, and RBD, respectively. Nearest-neighbor predictions have Spearman correlations of 0.2617, 0.1571, and 0.0429 on the three tasks, indicating weaker local score association for RBD under this simple prediction rule. To further characterize RBD, we group its test sequences by nearest-neighbor distances of 3–4 and at least 5 positions.

Table 10: Nearest-neighbor diagnostics on the RBD test partition used in this paper. Correlation, MAE, and R^2 compare each test score with its nearest training neighbor's score.

Statistic	All	Distance 3–4	Distance ≥ 5
Number of queries	5,790	109	5,681
Fraction of test set (%)	100.00	1.88	98.12
Median score	0.0808	0.1318	0.0802
Number with $y > 0.5$	541	16	525
Number with $y > 0.8$	110	1	109
Nearest-neighbor Spearman	0.0429	0.4110	0.0347
Nearest-neighbor MAE	0.1819	0.1128	0.1833
Nearest-neighbor R^2	-0.7311	-0.0460	-0.7476

RBD sequences with closer training neighbors show stronger score association, but account for only 1.88% of the test set. For the overwhelming majority with nearest-neighbor distance at least five, directly using the nearest training neighbor's score provides little ranking ability.

Sequence-cluster coverage of high-scoring samples. We also count the sequence clusters represented among high-scoring samples. In RBD, 641 training sequences with scores above 0.8 occupy 117 clusters; 96 validation sequences occupy 31 clusters; and 110 test sequences occupy only 25 clusters. The largest high-score test cluster contains 27 sequences, and the five largest together contain 75, or 68.18% of all high-scoring test sequences. The validation set shows similar concentration, with 65.63% of high-scoring sequences in its five largest clusters. By comparison, high-scoring CD3 and cMYC test sequences cover more distinct clusters (Table 11).

Table 11: High-scoring samples ($y > 0.8$) and their sequence-cluster distribution in the test partitions used in this paper. The last column is the percentage of high-scoring samples in the five clusters containing the most such samples.

Target	High-scoring samples	Clusters represented	Top-five share (%)
CD3	1,161	1,161	0.43
cMYC	346	345	1.73
RBD	110	25	68.18

Thus, high-scoring RBD test samples primarily come from a small number of similar sequence groups, with limited coverage of other high-scoring sequence types.

Interpretation. RBD scores are concentrated at low values, and high-scoring samples are few and concentrated in a small number of similar sequence groups. Under the current evaluation protocol, all compared models show weak predictive performance on RBD. We report RBD separately and retain the complete numerical results for comparison.

D.3 Activity Prediction Across All Targets

Table 12 reports activity prediction results for all three targets. We report both the two-target macro R^2 over CD3 and cMYC, used in the main analysis, and the three-target macro R^2 including RBD. Both macro scores are computed within each seed before summarizing across five downstream seeds.

Table 12: Sequence-only activity prediction. Values are mean ± sample standard deviation across five downstream seeds. Macro_2 averages CD3 and cMYC within each seed; Macro_3 averages all three targets. Rows are ordered by mean Macro_2. The RiboWeave prefix is omitted for its variants. Bold indicates the highest mean in each column.

Model	CD3 R^2	cMYC R^2	RBD R^2	Macro_2 R^2	Macro_3 R^2
Hybrid	**0.6686±0.0009**	**0.4131±0.0017**	0.0070±0.0014	**0.5408±0.0010**	**0.3629±0.0005**
Nuc-pair	0.6627±0.0006	0.4100±0.0033	0.0090±0.0016	0.5364±0.0016	0.3606±0.0010
Nuc-joint	0.6630±0.0017	0.4088±0.0017	0.0078±0.0027	0.5359±0.0011	0.3599±0.0004
Nuc-geo	0.6602±0.0013	0.4086±0.0018	0.0068±0.0017	0.5344±0.0006	0.3585±0.0005
Mol-joint	0.6638±0.0011	0.4035±0.0019	0.0112±0.0011	0.5336±0.0013	0.3595±0.0006
Mol-pair	0.6602±0.0016	0.4030±0.0016	**0.0122±0.0013**	0.5316±0.0015	0.3585±0.0007
RNA-FM	0.6534±0.0012	0.3986±0.0020	0.0066±0.0009	0.5260±0.0011	0.3529±0.0009
ERNIE-RNA	0.6537±0.0008	0.3959±0.0019	0.0059±0.0012	0.5248±0.0010	0.3518±0.0008
RiNALMo-Mega	0.6535±0.0021	0.3896±0.0043	0.0062±0.0014	0.5216±0.0020	0.3498±0.0017
Mol-geo	0.6528±0.0016	0.3897±0.0030	0.0063±0.0005	0.5213±0.0020	0.3496±0.0014
RNAErnie	0.6548±0.0022	0.3849±0.0076	0.0050±0.0021	0.5199±0.0033	0.3482±0.0022

D.4 Activity Prediction Controls

Table 13 reports the capacity-control results for all three targets. Hybrid achieves higher mean R^2 on CD3 and cMYC and a higher macro score than both single-route controls, supporting the benefit of combining the two supervision granularities. RBD prediction remains weak across all five configurations.

Table 13: Capacity controls and single-modality hybrid comparisons. Values are mean ± sample standard deviation across five downstream seeds. Pair-hybrid combines Mol-pair and Nuc-pair; Geo-hybrid combines Mol-geo and Nuc-geo. Macro_2 and Macro_3 follow Table 12. Bold indicates the highest mean in each column.

Representation	CD3 R^2	cMYC R^2	RBD R^2	Macro_2 R^2	Macro_3 R^2
Hybrid	**0.6686±0.0009**	**0.4131±0.0017**	**0.0070±0.0014**	**0.5408±0.0010**	**0.3629±0.0005**
Pair-hybrid	0.6665±0.0012	0.4120±0.0016	0.0066±0.0030	0.5393±0.0009	0.3617±0.0014
Geo-hybrid	0.6626±0.0008	0.4093±0.0031	0.0048±0.0011	0.5359±0.0017	0.3589±0.0013
Nucleotide-capacity	0.6596±0.0013	0.4048±0.0016	0.0063±0.0022	0.5322±0.0014	0.3569±0.0012
Molecule-capacity	0.6618±0.0009	0.3997±0.0009	0.0058±0.0022	0.5308±0.0003	0.3558±0.0007

D.5 Structural Supervision and Predicted-Structure Controls

Table 14: Structural supervision and predicted-structure controls on CD3 and cMYC. Values are test R^2 (mean $\pm$ sample standard deviation across five downstream seeds). Macro averages cover CD3 and cMYC. $\oplus$ denotes feature concatenation; Nuc-pair and Nuc-geo abbreviate the corresponding RiboWeave model names.

Method	CD3 R^2	cMYC R^2	Macro R^2
RNA-FM	0.6534 ± 0.0012	0.3986 ± 0.0020	0.5260 ± 0.0011
RiboWeave Hybrid	0.6686 ± 0.0009	0.4131 ± 0.0017	0.5408 ± 0.0010
Upstream permutation			
Nuc-pair (pair permutation)	0.6361 ± 0.0100	0.3674 ± 0.0058	0.5018 ± 0.0053
Nuc-geo (target-row permutation)	0.6090 ± 0.0166	0.2713 ± 0.0370	0.4402 ± 0.0256
Nuc-geo (coordinate permutation)	0.6430 ± 0.0047	0.3734 ± 0.0046	0.5082 ± 0.0042
Folding summaries			
Folding summaries	0.2127 ± 0.0038	0.0368 ± 0.0016	0.1248 ± 0.0026
RNA-FM $\oplus$ folding summaries	0.6529 ± 0.0014	0.3921 ± 0.0042	0.5225 ± 0.0027
Direct graph input			
RNA-FM + GINE (backbone edges)	0.6613 ± 0.0052	0.4032 ± 0.0047	0.5323 ± 0.0049
RNA-FM + GINE (backbone + predicted-pairing edges)	0.6625 ± 0.0021	0.4023 ± 0.0025	0.5324 ± 0.0021

All three upstream permutation controls have lower mean macro R^2 than RNA-FM. Folding summaries perform poorly alone and provide no improvement when concatenated with RNA-FM. Both RNA-FM+GINE configurations improve over RNA-FM but remain below RiboWeave Hybrid on both targets and their mean macro R^2, with nearly identical mean macro scores with and without predicted pairing edges. Control definitions and training settings appear in Appendix C.4.

D.6 Final Evaluator Reliability

Table 15 reports natural-holdout prediction performance for the common evaluator panel. CD3 and cMYC have positive mean R^2 across all evaluator families, whereas all RBD families have negative mean R^2. This motivates reporting RBD generation as exploratory.

Table 15: Final evaluator reliability on the natural holdout set. Values are mean $\pm$ sample standard deviation over five evaluator seeds.

Target	Evaluator family	R^2	Spearman
CD3	RNA-FM	0.6429 ± 0.0015	0.7739 ± 0.0010
	RiboWeave Nuc-pair	0.6508 ± 0.0025	0.7764 ± 0.0013
	RiboWeave Nuc-geo	0.6549 ± 0.0039	0.7790 ± 0.0028
	RiboWeave Nuc-joint	0.6508 ± 0.0035	0.7761 ± 0.0020
	RiboWeave Mol-pair	0.6578 ± 0.0011	0.7832 ± 0.0005
	RiboWeave Mol-geo	0.6336 ± 0.0026	0.7688 ± 0.0025
	RiboWeave Mol-joint	0.6484 ± 0.0043	0.7772 ± 0.0027
	RiboWeave Hybrid	0.6478 ± 0.0089	0.7753 ± 0.0057
cMYC	RNA-FM	0.3539 ± 0.0051	0.5531 ± 0.0050
	RiboWeave Nuc-pair	0.3819 ± 0.0047	0.5777 ± 0.0024
	RiboWeave Nuc-geo	0.3840 ± 0.0061	0.5782 ± 0.0043
	RiboWeave Nuc-joint	0.3768 ± 0.0093	0.5710 ± 0.0081
	RiboWeave Mol-pair	0.3808 ± 0.0014	0.5744 ± 0.0019
	RiboWeave Mol-geo	0.3517 ± 0.0044	0.5509 ± 0.0037
	RiboWeave Mol-joint	0.3736 ± 0.0069	0.5693 ± 0.0045
	RiboWeave Hybrid	0.3768 ± 0.0105	0.5716 ± 0.0079
RBD	RNA-FM	-0.1563 ± 0.0930	0.0448 ± 0.0150
	RiboWeave Nuc-pair	-0.1978 ± 0.0184	0.0339 ± 0.0159
	RiboWeave Nuc-geo	-0.2040 ± 0.0961	0.0307 ± 0.0254
	RiboWeave Nuc-joint	-0.2179 ± 0.0215	0.0281 ± 0.0119
	RiboWeave Mol-pair	-0.2462 ± 0.0358	0.0249 ± 0.0075
	RiboWeave Mol-geo	-0.2030 ± 0.0210	0.0286 ± 0.0117
	RiboWeave Mol-joint	-0.1917 ± 0.0257	0.0573 ± 0.0189
	RiboWeave Hybrid	-0.0579 ± 0.0276	0.0903 ± 0.0067

D.7 EVALUATOR-FAMILY SENSITIVITY OF GENERATION SCORES

The main generation analysis scores every locked candidate set using the same panel of eight evaluator families. To assess whether the reported generation performance depends on a particular evaluator family, we first score each generator separately with each evaluator family. We then repeat the panel aggregation after removing the evaluator family corresponding to the generator. These analyses test whether the generation comparisons are sensitive to evaluator choice or are preferentially favored by a matching evaluator.

Table 16: Generator-evaluator cross-scoring on CD3 and cMYC. Rows denote generator families and columns denote evaluator families. Entries are mean predicted pseudo-activity scores across five generation seeds. Underlining marks the evaluator corresponding to the generator, and bold indicates the highest mean within each evaluator column. Per-cell sample SDs are omitted for readability; variability of the common panel scores is reported with the main generation results.

Generator	RNA-FM	Mol-pair	Mol-geo	Mol-joint	Nuc-pair	Nuc-geo	Nuc-joint	Hybrid
CD3								
RNA-FM	0.8109	0.8171	0.8109	0.8120	0.8153	0.8143	0.8097	0.8140
Mol-pair	0.8160	0.8222	0.8159	0.8171	0.8213	0.8200	0.8152	0.8195
Mol-geo	0.8141	0.8200	0.8151	0.8155	0.8189	0.8181	0.8135	0.8187
Mol-joint	0.8150	0.8225	0.8155	0.8172	0.8207	0.8192	0.8145	0.8196
Nuc-pair	0.8172	0.8242	0.8167	0.8183	0.8225	0.8213	0.8162	0.8206
Nuc-geo	0.8154	0.8226	0.8147	0.8163	0.8207	0.8195	0.8149	0.8188
Nuc-joint	**0.8199**	0.8276	**0.8197**	0.8210	**0.8258**	**0.8246**	0.8196	0.8238
Hybrid	0.8193	**0.8280**	0.8196	**0.8211**	0.8253	0.8244	**0.8197**	**0.8238**
cMYC								
RNA-FM	0.6526	0.6577	0.6570	0.6619	0.6614	0.6588	0.6566	0.6605
Mol-pair	0.6654	0.6695	0.6701	0.6742	0.6746	0.6717	0.6704	0.6732
Mol-geo	0.6599	0.6640	0.6636	0.6677	0.6688	0.6669	0.6646	0.6669
Mol-joint	0.6650	0.6692	0.6691	0.6736	0.6743	0.6717	0.6698	0.6728
Nuc-pair	0.6575	0.6619	0.6616	0.6661	0.6671	0.6645	0.6622	0.6649
Nuc-geo	0.6662	0.6713	0.6712	0.6749	0.6758	0.6739	0.6719	0.6745
Nuc-joint	0.6673	0.6730	0.6714	0.6770	0.6772	0.6751	0.6734	0.6765
Hybrid	**0.6697**	**0.6747**	**0.6733**	**0.6784**	**0.6795**	**0.6776**	**0.6756**	**0.6782**

The cross-scoring results show substantial agreement across evaluator families. On cMYC, Hybrid is the highest-scoring generator under all eight individual evaluator families, indicating that its advantage is not specific to the evaluator derived from the Hybrid representation. On CD3, Hybrid and Nuc-joint consistently form the two strongest configurations: Hybrid has the highest mean score under four evaluator families and Nuc-joint under the other four. Thus, the fine-grained ordering between these two generators varies with evaluator family, whereas their advantage over the remaining configurations is broadly consistent.

To quantify this agreement, we compare the ranking of the eight generators induced by each evaluator family. The mean pairwise Spearman correlation between evaluator-family rankings is 0.933 on CD3 and 0.994 on cMYC, indicating that different evaluator families produce highly similar generator-level rankings.

We next remove the evaluator family corresponding to each generator and average the remaining seven scores. The resulting changes are small across both targets, and the overall ranking is largely preserved. In particular, Hybrid remains the strongest configuration on cMYC, while Hybrid and Nuc-joint remain closely matched on CD3. These results suggest that the generation comparisons are not driven by preferential scoring from matching evaluator families.

Together, the cross-scoring matrix, high agreement between evaluator rankings, and leave-matched-out analysis indicate that the main generation comparisons are not driven by a single evaluator family or by preferential self-evaluation. The CD3 ordering between Hybrid and Nuc-joint remains sensitive at the fine-grained level, whereas the cMYC advantage of Hybrid is consistent across evaluator families.

Table 17: Sensitivity of computational generation scores to exclusion of the evaluator family matching each generator. *Full panel* is the equally weighted mean score across all eight evaluator families, whereas *Matched family excluded* is the equally weighted mean across the remaining seven evaluator families after excluding the family corresponding to the generator. For each target, generator and generation seed, scores were first averaged over the 1,000 generated candidates. Values are mean ± s.d. across five generation seeds.

	CD3		cMYC	
Generator	Full panel	Matched family excluded	Full panel	Matched family excluded
Hybrid	0.8226 ± 0.0025	0.8225 ± 0.0026	0.6759 ± 0.0020	0.6756 ± 0.0019
Nuc-joint	0.8228 ± 0.0047	0.8232 ± 0.0046	0.6739 ± 0.0049	0.6739 ± 0.0049
Nuc-geo	0.8179 ± 0.0031	0.8176 ± 0.0031	0.6725 ± 0.0018	0.6723 ± 0.0018
Nuc-pair	0.8196 ± 0.0052	0.8192 ± 0.0051	0.6632 ± 0.0038	0.6627 ± 0.0038
Mol-joint	0.8180 ± 0.0079	0.8182 ± 0.0080	0.6707 ± 0.0039	0.6703 ± 0.0039
Mol-pair	0.8184 ± 0.0053	0.8179 ± 0.0053	0.6711 ± 0.0016	0.6713 ± 0.0016
Mol-geo	0.8168 ± 0.0036	0.8170 ± 0.0036	0.6653 ± 0.0031	0.6656 ± 0.0032
RNA-FM	0.8130 ± 0.0056	0.8133 ± 0.0057	0.6583 ± 0.0044	0.6591 ± 0.0044

D.8 Computational Design

Table 18 reports the complete numerical values underlying Figure 4 and exploratory RBD generation results.

Table 18: Generation results on CD3, cMYC and RBD. Mean and Top10 scores are model predictions. Novelty is measured relative to the development set; diversity is the fraction of within-set sequence clusters at 75% identity. Values are mean ± sample SD across five generation seeds.

Target	Generator	Mean score	Top10 score	Novel@0.75 (%)	Diversity@0.75 (%)
CD3	RNA-FM	0.8130±0.0056	0.9163±0.0027	86.92±2.63	95.78±0.49
CD3	RiboWeave Nuc-pair	0.8196±0.0052	0.9161±0.0034	89.52±1.75	96.22±1.07
CD3	RiboWeave Nuc-geo	0.8179±0.0031	**0.9172**±0.0020	91.28±1.55	96.70±0.69
CD3	RiboWeave Nuc-joint	**0.8228**±0.0047	0.9167±0.0026	93.30±0.97	97.06±0.27
CD3	RiboWeave Mol-pair	0.8184±0.0053	0.9154±0.0046	94.90±0.81	96.90±0.75
CD3	RiboWeave Mol-geo	0.8168±0.0036	0.9154±0.0039	94.54±0.90	**97.22**±0.33
CD3	RiboWeave Mol-joint	0.8180±0.0079	0.9157±0.0016	93.64±1.71	96.80±0.28
CD3	RiboWeave Hybrid	0.8226±0.0025	**0.9172**±0.0019	**96.66**±0.98	97.02±0.88
cMYC	RNA-FM	0.6583±0.0044	0.7892±0.0071	86.52±2.22	98.18±0.66
cMYC	RiboWeave Nuc-pair	0.6632±0.0038	0.7885±0.0040	89.16±1.19	98.44±0.15
cMYC	RiboWeave Nuc-geo	0.6725±0.0018	0.7911±0.0072	93.32±1.58	99.00±0.16
cMYC	RiboWeave Nuc-joint	0.6739±0.0049	**0.7967**±0.0078	95.50±0.71	99.02±0.28
cMYC	RiboWeave Mol-pair	0.6711±0.0016	0.7896±0.0098	96.14±0.83	98.82±0.22
cMYC	RiboWeave Mol-geo	0.6653±0.0031	0.7939±0.0089	95.94±0.68	98.98±0.23
cMYC	RiboWeave Mol-joint	0.6707±0.0039	0.7937±0.0088	92.96±0.64	98.62±0.33
cMYC	RiboWeave Hybrid	**0.6759**±0.0020	0.7959±0.0033	**97.50**±1.07	**99.06**±0.18
RBD	RNA-FM	0.1294±0.0010	0.3143±0.0138	79.50±2.24	93.06±1.76
RBD	RiboWeave Nuc-pair	0.1314±0.0011	0.3415±0.0133	83.34±3.99	94.82±2.10
RBD	RiboWeave Nuc-geo	**0.1342**±0.0009	**0.3601**±0.0177	86.30±2.24	95.96±1.11
RBD	RiboWeave Nuc-joint	0.1318±0.0024	0.3246±0.0145	91.44±2.40	97.70±1.19
RBD	RiboWeave Mol-pair	0.1322±0.0019	0.3494±0.0247	90.92±1.27	97.74±0.37
RBD	RiboWeave Mol-geo	0.1324±0.0028	0.3475±0.0197	86.36±1.90	96.18±0.74
RBD	RiboWeave Mol-joint	0.1309±0.0016	0.3223±0.0157	83.06±2.73	95.56±1.52
RBD	RiboWeave Hybrid	0.1321±0.0013	0.3397±0.0232	**95.92**±2.04	**98.98**±0.36

Figure 6 reports the RBD generation results, showing mean evaluator scores and retained candidate fractions across maximum-identity thresholds. Evaluator reliability is reported in Appendix D.6.

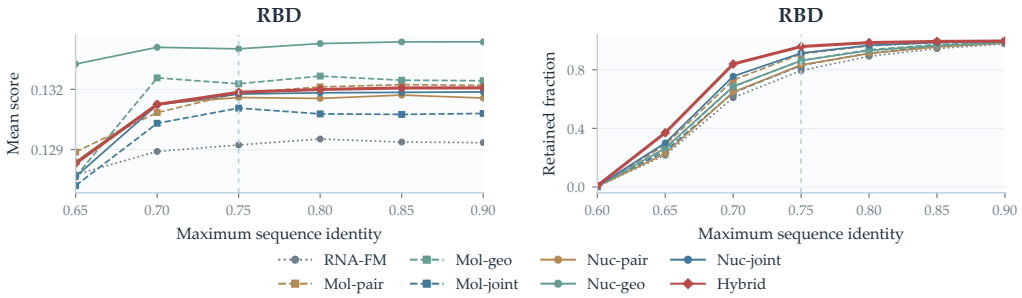

Figure 6: Exploratory RBD generation results: evaluator scores among candidates retained at each maximum-identity threshold (left), and the corresponding fractions retained (right).