RNACODER: TOKENIZING RNA STRUCTURE WITH LOCAL ALL-ATOM AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tokenizing protein structures, by reconciling atomic resolution and the efficiency of coarse-grained modeling, has led to notable successes in homology search, protein design and property prediction. Extending tokenization to RNA holds great potential to capture its structural richness despite data scarcity. Yet, existing approaches overlook two key features of RNA: the centrality of base pairs and dinucleotides as building blocks, and the prevalence of chemical modifications.

We introduce RNAcoder, a modification-aware, all-atom RNA structure tokenizer. Unlike previous approaches, RNAcoder tokenizes dinucleotides and base pairs alongside single nucleotides, each through a strictly local autoencoder. RNAcoder's learned tokens align with known Leontis-Westhof base pair families and the semantics of chemical modifications, although never trained on these labels. Integrated into coarse-grained graphs to represent whole RNA structures, RNAcoder tokens improve performance on small molecule binding site prediction, RNA-protein interface prediction, and ligand chemical class prediction. Our code is available anonymously at: `https://anonymous.4open.science/r/RNAcoder-D275`

1 INTRODUCTION

Structure tokenization consists of compressing the atomistic details of a structure into a vector representation. It has recently advanced protein representation learning, by accelerating similarity searches (Van Kempen et al., 2024), opening the door to sequence-structure multimodal foundation models (Hayes et al., 2025; Su et al., 2024; Li et al., 2024; Ouyang-Zhang et al., 2025) and latent diffusion models (Geffner et al., 2026), and facilitating property prediction (Yuan et al., 2025).

While these successes have focused on protein structures, ribonucleic acid (RNA) is increasingly recognized as a compelling therapeutic modality (Wang et al., 2025) and as a target for small-molecule drugs (Childs-Disney et al., 2022; Karroucha et al., 2026). Unlocking its full therapeutic potential will necessitate solving several computational tasks, including generative modeling for therapeutic RNA design, stability prediction for mRNA vaccine design, and binding affinity prediction for targeting RNA with small molecules.

Local structure tokenization holds great potential for RNA specifically, which has scarce structural data but displays strong local regularities (Leontis et al., 2006).

These regularities emerge not only at the nucleotide level but also at the scale of **base pairs** and pairs of consecutive nucleotides in sequence (denoted as **dinucleotides**) (Černỳ et al., 2020). Indeed, base stacking (which mostly occurs within dinucleotides) and base pairing (between nucleotides potentially far apart in sequence) are the two main driving forces of RNA folding, thus making base pairs and dinucleotides fundamental building blocks of RNA architecture. However, they have not been considered as potential tokens, as existing approaches focus on either atoms or single nucleotides.

In addition to these strong atomic-level geometric patterns, RNA nucleotides can undergo more than 180 distinct **chemical modifications** (Adams et al., 2025; Seelam et al., 2017) that alter their function, contribute to pathologies (Barbieri & Kouzarides, 2020), and are central to therapeutic RNA design (Wang et al., 2025). Yet existing RNA structure tokenizers do not account for chemical modifications.

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.

Main contributions **(i)** We propose RNAcoder, a strictly local, all-atom RNA structure tokenizer extending tokenization beyond single nucleotides to RNA base pairs and dinucleotides for the first time. Moreover, it explicitly accounts for nucleotide modifications, thus addressing two major gaps in current RNA tokenization. **(ii)** We further introduce the RNAtom37 nomenclature that unifies standard and modified nucleotides into a common matrix representation, which we release as a standalone resource for modification-aware RNA structure prediction and generation. **(iii)** We show, in Section 4, that the latent tokens display a biologically meaningful organization, especially regarding base pairing and nucleotide modifications. **(iv)** We integrate our local RNAcoder tokens into the node and edge features of a relational graph representing whole RNAs (Figure 1B). Replacing hand-crafted graph featurization, this novel representation bridges atomic and coarse-grained modeling. In Section 5, we show it consistently improves prediction performance across three tasks covering RNA-protein and RNA-small-molecule interactions.

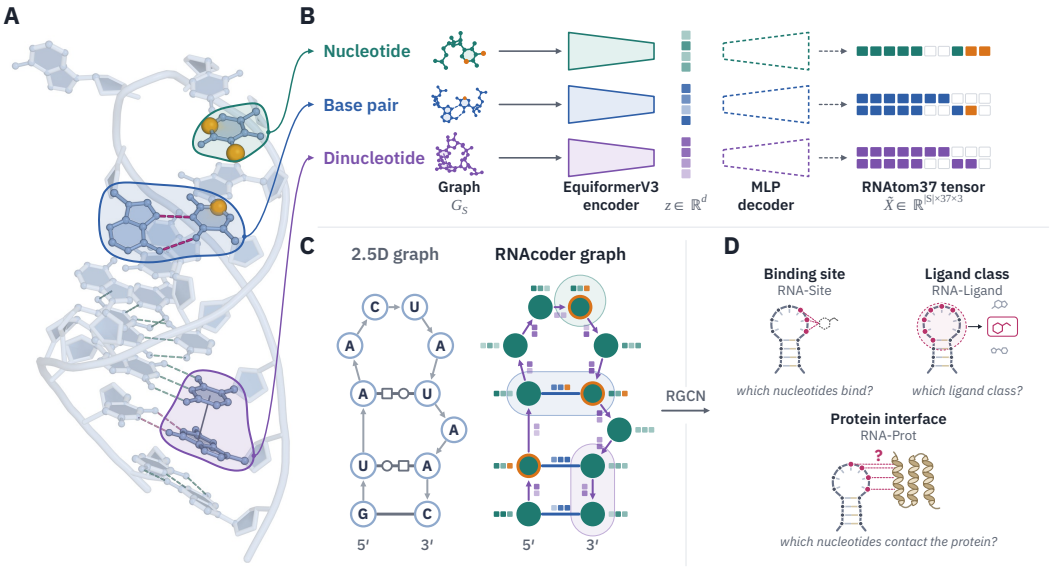

Figure 1: Overview of RNAcoder. **A.** 3D structure of the 69 helix of the E. coli 23S rRNA (PDB `7k00`), with one example of each tokenized subunit highlighted: a nucleotide, a base-pair and a stacking dinucleotide. Chemical modifications are shown in orange, and canonical base-pairs as green dashes. **B.** Three strictly local all-atom autoencoders each encode a subunit type (nucleotide, base pair, or dinucleotide). An SE(3)-equivariant EquiformerV3 encoder maps the atom graph G_S of a subunit to a token z, from which an MLP decoder reconstructs its RNAtom37 tensor $\tilde{X}$. The decoder is not used at inference time. Tokenizers are modification-aware. **C.** Downstream, the tokens serve as node and edge features of a coarse-grained graph. They replace the nucleotide identities and Leontis-Westhof edge types of the standard 2.5D graph. **D.** A relational graph convolutional network (RGCN) trained on this graph recovers long-range dependencies to predict RNA interactions. The tasks are prediction of small-molecule binding sites (RNA-Site) and protein interfaces (RNA-Prot) at the nucleotide level, and of the chemical class of the ligand of a binding pocket (RNA-Ligand).

2 Related Work

Protein structure tokenizers Structure tokenization has mostly been explored for proteins, where it was pioneered by Foldseek (Van Kempen et al., 2024), which learns a discrete 3D alphabet at the amino acid level. The term "token" has since been extended to continuous embeddings (Chen et al., 2026; Yuan et al., 2025), and numerous tokenizers have been proposed at the level of residues (Li et al., 2024; Hayes et al., 2025; Bojan et al., 2026; Ouyang-Zhang et al., 2025), whole proteins (Dilip et al., 2026), and conformational ensembles (Shi & Oliver, 2026).

Biomolecular structure tokenizers Beyond proteins, two cross-biomolecule tokenizers supporting RNA have been developed: Bio2token (Liu et al., 2024) and BiHiTo (Zheng et al., 2026). However,

both return embeddings at the atomic level and treat biomolecules as purely geometric point clouds of atoms, ignoring chemical identities. Only one RNA-specific tokenizer, RiboSphere (Zhang et al., 2026), has been developed to date. However, it restricts the RNA representation to backbone atoms and optionally the base-anchoring atom (N1/N9). It therefore cannot tokenize modified nucleotides and overlooks the internal geometry of the bases. Moreover, these three tokenizers are trained on a whole-RNA reconstruction task, which is unsuited to the scarcity of RNA structural data.

Base-pair representations Base-pair geometries underlie RNA's structure, function, and interactions (Leontis & Westhof, 2001). Their importance in machine learning is being increasingly emphasized: incorporating them as edge features in RNA graph representations (2.5D graphs) significantly improved prediction performance for drug affinity prediction (Oliver et al., 2020), and, more recently, across a broader benchmark of tasks (Wyss et al., 2025). Recent works proposed using base-pair geometries for RNA folding as an auxiliary training signal (Szikszai et al., 2025) or as a preliminary prediction objective (Zhu et al., 2026). However, all of these methods rely on the hand-crafted Leontis-Westhof discrete classification (Leontis & Westhof, 2001), which neither accounts for edge cases such as bifurcated base pairs nor captures the degree of geometric similarity between families, especially between isosteric families (Stombaugh et al., 2009).

RNA structure representations A parallel line of research develops models taking RNA structures as input, performing various tasks, from inverse folding and model quality assessment to binding affinity prediction. Most of these models represent RNAs as nucleotide-level graphs (Huang et al., 2024; Wang et al., 2024), and gRNAde (Joshi et al., 2025) extends this representation to a multi-graph to capture RNA conformational flexibility. Such coarse-grained approaches abstract away nucleotide modifications and fine-grained geometric variations. Alternatively, some approaches rely on all-atom representations of RNAs (Xu et al., 2025; 2026; Xia et al., 2025; Thakare et al., 2026), sometimes combined with hierarchical pooling (Li et al., 2025). However, all-atom graphs are more computationally expensive, and learning from such high-dimensional inputs typically requires more training data, which remains scarce for RNA structures.

3 Tokenizing RNA Structure with RNAcoder

3.1 All-atom local representation of RNA

Let $\mathcal{T}$ denote the set of named atom types present in RNA structures. The structure of an RNA residue $\mathcal{R}$ with $n_{\mathcal{R}}$ heavy atoms can be described by residue-level features $y_{\mathcal{R}} \in \mathbb{R}^k$ such as the residue type, and a set $V_{\mathcal{R}} = \{(t_j, x_j)\}_{j=1}^{n_{\mathcal{R}}} \subset \mathcal{T} \times \mathbb{R}^3$ of heavy atoms and their coordinates in a certain frame F. Several classic modeling challenges arise, as $n_{\mathcal{R}}$ varies across nucleotide types, atoms within a residue have no canonical ordering, and F is arbitrary. Our tokenizers are trained on a reconstruction task, in the auto-encoding framework, requiring inputs and outputs to be cleanly defined.

Graph representation One avenue to represent $\mathcal{R}$ is to use a radius graph $G_{\mathcal{R}} = (V_{\mathcal{R}}, E_{\mathcal{R}})$, where edges are formed between nodes closer than a threshold τ (5Å in our experiments). Node features encode *atomic elements* and coordinates. Graphs naturally accommodate variable sizes and permutation-invariance of the atoms. Furthermore, the output of an SE(3)-invariant neural network on a geometric graph is independent of the frame F in which its coordinates are expressed. As a consequence, graph representations are well-suited as input representations of our autoencoders. However, a graph does not constitute a convenient decoder target.

Tensor representation Conversely, tensor representations of atomic coordinates are well suited as reconstruction targets for the decoder. However, tensors do not intrinsically encode size-variability, permutation-invariance, and frame-invariance of RNA residues. We address these gaps by introducing a canonical nomenclature of atoms along with local frames.

RNAtom37 nomenclature To enable tensor representation of RNA subunits, we introduce a canonical, fixed-size mapping of RNA atoms, inspired by the atom37 nomenclature introduced for proteins by Jumper et al. (2021). We map all atoms to one of 37 potential heavy-atom slots such that structurally analogous atoms across nucleotide types are mapped to the same slot. The nomenclature

was constructed by manual inspection of the most frequent modified nucleotide types in experimental structures and is provided in full in Appendix A. Beyond RNAcoder, RNAtom37 could serve as a modification-aware target for RNA structure prediction and generation models.

Formally, let $\mathcal{T}_\mathcal{R} \subset \mathcal{T}$ denote the set of atom types present in residue $\mathcal{R}$. The RNAtom37 nomenclature defines an injective mapping $h_\mathcal{R} : \mathcal{T}_\mathcal{R} \to \{1, \dots, 37\}$ assigning indices to atoms, thereby ordering and stacking the atoms of $V_\mathcal{R}$ into canonical vectors $(A_\mathcal{R}, X_\mathcal{R}) \in \mathcal{T}^{n_\mathcal{R}} \times \mathbb{R}^{n_\mathcal{R} \times 3}$. This correspondence is encoded by a canonical selection matrix $\mathbf{S}_\mathcal{R} \in \{0,1\}^{n_\mathcal{R} \times 37}$, where entry (j,k) is 1 if $h_\mathcal{R}$ maps the j-th atom to slot k, and 0 otherwise. Using $\mathbf{S}_\mathcal{R}$, the variable-sized coordinate matrix $X_\mathcal{R}$ is lifted into a fixed-size target tensor $\tilde{X}_\mathcal{R} = \mathbf{S}_\mathcal{R}^\top X_\mathcal{R} \in \mathbb{R}^{37\times 3}$, which places atomic coordinates in their designated slots and zero-fills unoccupied ones. This fixed-dimensional representation $\tilde{X}_\mathcal{R}$ serves as a uniform target space for decoding, while left-multiplication of a predicted tensor $\hat{X}_\mathcal{R} \in \mathbb{R}^{37\times 3}$ by $\mathbf{S}_\mathcal{R}$ extracts the coordinates of the atoms physically present in $\mathcal{R}$.

Local frame construction and tensor representation To remove dependence on the arbitrary global frame F, we choose a canonical residue frame $F_\mathcal{R}$ centered at the phosphorus atom P_i. Its basis is generated via Gram-Schmidt orthogonalization: the primary axis follows the backbone direction $\overrightarrow{\mathrm{P}_i\mathrm{P}_{i+1}}$, the secondary axis points toward the base-anchoring glycosidic carbon ($\mathrm{C1'}$), and the third completes a right-handed system. By using RNAtom37 and expressing atom coordinates in $F_\mathcal{R}$, the residue geometry is unambiguously represented by its canonical target tensor $\tilde{X}_\mathcal{R} \in \mathbb{R}^{37\times 3}$. In addition, we introduce a base-centric frame $BF_\mathcal{R}$ centered at the base-anchoring atom (position 14 of RNAtom37), with axes normal to the base plane and along the Watson-Crick edge.

Representing pairs We naturally extend this construction to represent pairs of nucleotides (base pairs or dinucleotides). We start by choosing a pair ordering $(\mathcal{R}_1, \mathcal{R}_2)$. We systematically encode dinucleotides in the sequence order. Base pairs are encoded in a directed manner: each base pair can give rise to two distinct data samples, according to the residue chosen as the first one.

We then construct our graph representation on $V_{\mathcal{R}_1} \cup V_{\mathcal{R}_2}$. We add a binary nucleotide membership indicator as a node feature to distinguish the atoms of each nucleotide. For the tensor construction, we use the reference frame $F_{\mathcal{R}_1}$ for dinucleotides, and the base-centric frame $BF_{\mathcal{R}_1}$ in the case of base pairs. Details about the frame constructions are provided in Appendix B. Finally, the tensors of the two residues are stacked into a single tensor.

For notational convenience, we define an RNA subunit $\mathcal{S}$ to be a single nucleotide or a pair of nucleotides, and denote as $|\mathcal{S}|$ the number of nucleotides it is composed of.

3.2 Autoencoder architecture

The encoder f_θ is parameterized as an SE(3)-invariant graph neural network, mapping $G_\mathcal{S}$ to a latent representation $\mathbf{z} \in \mathbb{R}^d$ (we chose $d = 32$ for residues and $d = 64$ for base pairs and dinucleotides after an extensive hyperparameter search, as detailed in Appendix I). The decoder g_ϕ is a simple multi-layer perceptron, mapping the latent representation $\mathbf{z}$ to a tuple $(\hat{y}, \hat{X}, \hat{A}) \in \mathbb{R}^{|\mathcal{S}|\times k} \times \mathbb{R}^{|\mathcal{S}|\times 37\times 3} \times \mathbb{R}^{|\mathcal{S}|\times 37\times(|\mathcal{T}|+1)}$, corresponding to residue type prediction ($\hat{y}$) and reconstruction of atomic coordinates ($\hat{X}$) and atom types ($\hat{A}$). Full architectural details are provided in Appendix C.

We opt for a graph invariant encoder because encoding a graph and decoding a tensor encourages geometric reasoning instead of copying. Moreover, at inference time, this representation generalizes to nucleotides with any chemical modification or missing atom without requiring a fixed atom vocabulary or local frame alignment. We choose EquiformerV3 (Liao et al., 2026) for its strong performance across several molecular benchmarks. Moreover, its ability to model many-body interactions through spherical harmonics and SwiGLU-S^2 activations is well-suited to RNA, whose base stacking interactions strongly involve aromatic rings.

3.3 Training

Training loss The autoencoders are trained on a combination of three objectives. The **residue type** recovery objective consists of a hierarchical loss over the 23 residue-type classes. To compensate for the underrepresentation of modified nucleotides, we first predict a 5-way broad distribution P_{broad} over

$\{A, C, G, U, \text{modified}\}$, and conditionally a 19-way distribution $P_{\text{mod}}(j \mid \text{modified})$ over modified-residue subclasses, so that $P(r) = P_{\text{broad}}(r)$ for $r \in \{A, C, G, U\}$ and $P(r) = P_{\text{broad}}(\text{modified}) \cdot P_{\text{mod}}(r \mid \text{modified})$ if r is a modified nucleotide. The residue loss $\mathcal{L}_{\text{res}}$ is then a simple cross-entropy for non-modified nucleotides and a cascaded one for modified nucleotides.

The **coordinate reconstruction** objective is a mean squared error computed exclusively over present atoms by projecting the RNAtom37 predictions via the selection matrix $\mathbf{S}_{\mathcal{R}}$:

$$\mathcal{L}_{\text{coords}} = \frac{1}{\sum_{\mathcal{R} \in \mathcal{S}} n_{\mathcal{R}}} \sum_{\mathcal{R} \in \mathcal{S}} \|X_{\mathcal{R}} - \mathbf{S}_{\mathcal{R}} \hat{X}_{\mathcal{R}}\|_F^2. \tag{1}$$

To regularize atomic-level chemical representations in the latent space, we further include an **atom identity** reconstruction term across all 37 slots. While not strictly required for reconstruction since residue identity and atomic coordinates fully specify the chemical structure, it reinforces chemical awareness:

$$\mathcal{L}_{\text{atom}} = \frac{1}{|\mathcal{S}| \cdot 37} \sum_{\mathcal{R} \in \mathcal{S}} \text{CE}(\tilde{A}_{\mathcal{R}}, \hat{A}_{\mathcal{R}}), \tag{2}$$

where CE denotes the cross-entropy loss, $\tilde{A}_{\mathcal{R}} = \mathbf{S}_{\mathcal{R}}^{\top} A_{\mathcal{R}} \in (\mathcal{T} \cup \{\emptyset\})^{37}$ embeds the true atom types into their designated canonical slots and assigns a null token $\emptyset$ to unoccupied positions, and $\hat{A}_{\mathcal{R}} \in \mathbb{R}^{37 \times (|\mathcal{T}|+1)}$ denotes the predicted class logits. These losses are combined with weights $(w_{\text{res}}, w_{\text{coords}}, w_{\text{atom}}) = (0.1, 1.0, 0.1)$, into a reconstruction loss $\mathcal{L}_{\text{recon}}$:

$$\mathcal{L}_{\text{recon}} = w_{\text{res}}\, \mathcal{L}_{\text{res}} + w_{\text{coords}}\, \mathcal{L}_{\text{coords}} + w_{\text{atom}}\, \mathcal{L}_{\text{atom}}. \tag{3}$$

Training data We train on the non-redundant subset of RNA structures from the Protein Data Bank (PDB) introduced by Leontis & Zirbel (2012). In order to compensate for the overrepresentation of canonical base pairs in the dataset, we subsample base pairs belonging to frequent Leontis-Westhof families, drawing inspiration from the word2vec (Church, 2017) subsampling strategy for frequent words. Similarly, we subsample nucleotides and dinucleotides based on their secondary structure context (details in Appendix D.2). This results in 1,741,532 training samples each for nucleotides and dinucleotides, and 1,323,282 for base pairs. Details about training hyperparameters are provided in Appendix D.4.

3.4 Reconstruction results

On the held-out set with no RNA overlap, the autoencoders achieve strong performance on coordinate reconstruction (with global RMSD ranging from 0.145 Å down to 0.052 Å), atom type recovery, and residue type and modification subtype prediction, including on the rarest base-pair geometries and structural contexts (see Table 1 and E). Visualizations of reconstructed nucleotides, base pairs, and dinucleotides are provided in Figure 2 and, more extensively, in Appendix F.

Table 1: Test-set reconstruction metrics for the three autoencoders. Geometric reconstruction is evaluated using balanced RMSD (Å). Chemical reconstruction is reported as balanced accuracy (%).

Model	Geometric reconstruction	Atom type	Residue type	Broad residue type	Modification subtype
Residue (lat. 32)	0.153	97.7%	96.0%	99.97%	95.3%
Dinucleotide (lat. 64)	0.115	95.6%	97.6%	99.96%	91.8%
Base pair (lat. 64)	0.060	93.0%	95.4%	99.91%	83.6%

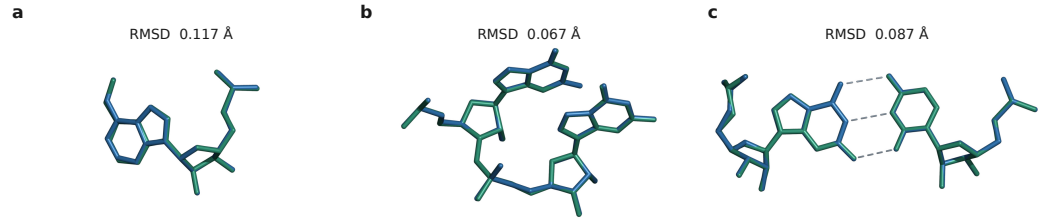

Figure 2: Comparisons between experimental structures (green) and RNAcoder reconstructions (blue) of a modified nucleotide, a base pair, and a dinucleotide.

4 Exploring Learned Token Spaces

4.1 Base pair tokens encode structural families and isostericity

To characterize what RNAcoder captures at the base-pair level, we first analyze how base pair tokens organize with respect to Leontis-Westhof families (Leontis & Westhof, 2001) without any supervision on these categories. A UMAP projection of the base pair embeddings (Figure 3A) shows that base pair tokens cluster by these families. We further demonstrate that Leontis-Westhof families can be recovered by a linear probe trained on RNAcoder tokens with 99.0% balanced accuracy (Figure 3B).

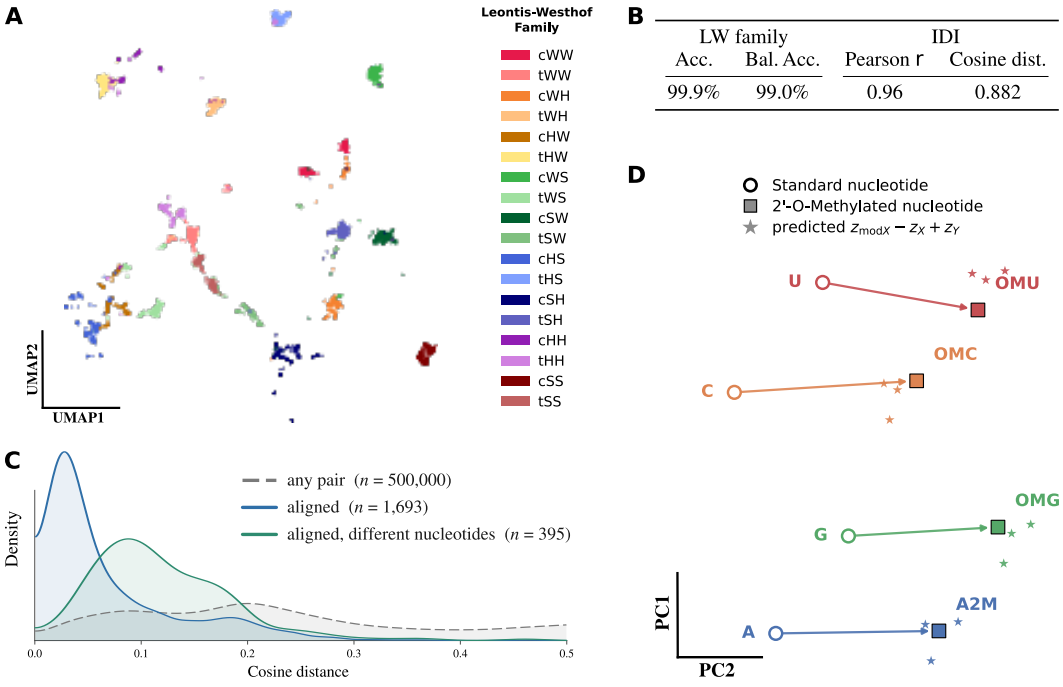

Figure 3: **A.** UMAP projection of base-pair embeddings, colored by Leontis-Westhof family. **B.** Linear-probe decodability of RNAcoder base-pair tokens. **C.** Distribution of cosine distances between pairs of RNAcoder tokens of base pairs that are either randomly picked (gray), evolutionarily aligned (blue), or evolutionarily aligned with different nucleotide types because of mutations (green). **D.** Two-dimensional PCA projection of RNAcoder centroids of the 4 standard nucleotides and the 2′-O-methylated nucleotides. Red stars denote the Emb(2′-O-methyl-X)-Emb(X)+Emb(U) combinations for all $X \in \{A, G, C\}$, which under the compositionality hypothesis result in Emb(2′-O-methyl-U). The same applies to orange (2′-O-methyl-C), green (2′-O-methyl-G), and blue (2′-O-methyl-A) stars.

Another key concept of RNA base pairs is isostericity: base pairs sharing specific geometric features are more likely to substitute for one another across evolution. A linear probe trained on the concatenation of RNAcoder tokens of two base pairs obtains a Pearson's correlation coefficient of

0.96 with the isodiscrepancy index (IDI) (Stombaugh et al., 2009) betweeen these two base pairs, that quantifies isostericity. As a direct measure of our ability to capture evolutionary invariants, we use an evolutionary alignment of base pairs coming from conserved ribosomal RNAs of two distinct bacteria: *Thermus thermophilus* and *Escherichia coli* (Stombaugh et al., 2009). Three types of pairs of base pairs are possible: (i) any-vs-any, (ii) base pairs that are evolutionarily aligned, and (iii) base pairs that are evolutionarily aligned but whose nucleotide identities have mutated. We compare the cosine distances between RNAcoder tokens for each scenario (Figure 3) and show that aligned base pairs have more similar tokens than randomly sampled pairs (AUROC = 0.90, 95% CI [0.89, 0.90], bootstrap over base pairs), including when their nucleotide identities differ (AUROC = 0.80, [0.79, 0.81]). This suggests that RNAcoder tokens could help machine learning models identify structure–function relationships that are conserved across mutations.

4.2 Residue tokens carry the semantics of chemical modifications

Learned embeddings were shown to have arithmetic properties in natural language processing (Mikolov et al., 2013a) and image generation (Radford et al., 2015). A well-known example is the word2vec arithmetic "King" - "Man" + "Woman" resulting in a vector close to the embedding of the word "Queen" (Mikolov et al., 2013b). Here, we propose a similar experiment for nucleotide chemical modifications in the residue token space.

For any nucleotide type X, let Emb(X) denote the centroid of the embeddings of nucleotides of type X. For any nucleotide types X and Y and chemical modification m such that X_m and Y_m belong to RNAtom37-supported modified nucleotides, we compare Emb(X_m)-Emb(X)+Emb(Y) to Emb(Z) for all standard and modified nucleotides Z. In all cases, we show that the nearest neighbor (in terms of cosine distance) of Emb(X_m)-Emb(X)+Emb(Y) is Emb(Y_m) (see Appendix G). We further evidence, in Figure 3, the ability of RNAcoder to learn implicitly the notion of 2′-O-methylation, whereas we did not provide any explicit supervision on this notion.

5 Integrating local RNAcoder tokens in RNA tasks

5.1 RNA tasks addressed

Having established the relevance of RNAcoder tokens at a local level, we now aim to assess their relevance as local features for tasks at the whole RNA level. The RNAglib package (Wyss et al., 2025) recently proposed a benchmark on experimental RNA structures. We use three tasks from this benchmark described shortly below and in more details in Appendix H.1.

Predicting the location of small-molecule binding sites on RNA structure is a key step in structure-based drug design. The **RNA-Site** task is a nucleotide-level binary classification task: for each nucleotide of an input RNA, the model predicts whether the nucleotide belongs to a small molecule binding site. This dataset holds 137 non redundant RNA structures. Once equipped with such pocket, the **RNA-Ligand** task is a multiclass classification task that tries to predict which chemical cluster is most likely to bind into an RNA pocket, with 359 examples. The last task focuses on RNA-protein interactions, which underpin numerous biological functions. RNAglib's **RNA-Prot** task aims at predicting whether each nucleotide of an input RNA belongs to a protein interface. 852 RNA compose this dataset.

5.2 RNA representations and models

RNA representations The aforementioned **2.5D graph** representation was shown to outperform 3D-aware methods (Wyss et al., 2025). The nodes of this coarse-grained graph are featurized with nucleotide identities and its directed edges encode base pairs and backbone connections. Edges are split into discrete edge types with 20 types corresponding to Leontis-Westhof families for base pair edges and two for each direction of the backbone. 2.5D graphs hold a strong biological and sparsity prior, particularly meaningful for low-data tasks like ours.

We benchmark against the only RNA nucleotide-level tokenizer to date, **RiboSphere** (Zhang et al., 2026) and against **BiHiTo** that is additionally trained on protein structures. Notably, both are trained

on a global reconstruction objective, contrary to us. We integrate their tokens as node features, replacing nucleotide identities with a geometric, pretrained alternative.

Finally, we replace the hand-crafted categories of 2.5D graphs with RNAcoder tokens, resulting in **RNAcoder graphs**. RNAcoder residue tokens are used as node features (as for RiboSphere and BiHiTo). The 20 base pair edge types are collapsed into one, with base-pair tokens as features. The two edge types corresponding to backbone directions are kept separate. Forward backbone edges $e_{ij}^{5'\to 3'}$ are featurized with dinucleotide tokens h_{ij}^{dn}. Since our encoder was trained only on the forward direction, we set $h_{ji}^{dn} = h_{ij}^{dn}$ and let the different edge types encode directionality.

Models To handle multiple edge types, 2.5D graphs are classically processed by an RGCN (Schlichtkrull et al., 2018). We used the same model to process the RiboSphere and RNAcoder graphs, adapted to accommodate edge features (Appendix H.2). We also tried considering edge types as one-hot encoded edge features and processing it with a simple edge-aware Graph Convolution Network (GCN) (Kipf & Welling, 2016).

5.3 Tokenized graphs are strong representations for downstream tasks

Results Compared to the hand-crafted features of 2.5D graphs, our local tokens result in a consistent and significant performance improvement across tasks (Table 2). RNAcoder also roughly halves the variance. Our rich, semantic tokens provide a well-conditioned inductive prior that stabilizes downstream learning.

Table 2: Benchmark results on downstream tasks: small molecule binding site prediction (RNA-Site), protein interface prediction (RNA-Prot), and pocket-aware ligand prediction (RNA-Ligand). Results are reported as follows: mean ± standard deviation across 5 seeds.

	RNA-Site		RNA-Prot		RNA-Ligand	
	Balanced accuracy (%)	AUC (%)	Balanced accuracy (%)	AUC (%)	Balanced accuracy (%)	AUC (%)
2.5D graph (GCN)	70.2 ± 0.8	75.5 ± 1.2	57.4 ± 0.7	59.6 ± 0.9	48.6 ± 7.9	68.2 ± 6.1
2.5D graph (RGCN)	70.1 ± 0.9	74.6 ± 2.0	60.4 ± 0.7	64.1 ± 0.6	33.5 ± 5.3	50.5 ± 7.6
Ribosphere	74.1 ± 1.2	80.8 ± 1.5	60.6 ± 0.7	65.2 ± 0.8	54.1 ± 3.6	73.4 ± 3.0
BiHiTo	$\mathbf{75.3 \pm 1.1}$	$\underline{81.4} \pm 0.5$	60.0 ± 0.7	63.9 ± 0.9	$\mathbf{69.2 \pm 3.9}$	$\mathbf{84.1 \pm 2.8}$
RNAcoder	71.0 ± 0.8	78.9 ± 1.0	$\mathbf{65.6 \pm 0.4}$	$\mathbf{70.5 \pm 0.3}$	54.0 ± 4.1	72.0 ± 2.6
BiHiTo + RNAcoder	$\underline{74.3 \pm 1.1}$	$\mathbf{83.3 \pm 1.3}$	$\underline{64.0 \pm 0.6}$	$\underline{69.0 \pm 0.4}$	$\underline{61.3 \pm 2.9}$	$\underline{78.6 \pm 3.0}$

Compared to tokens trained on full-structure, the picture is more nuanced. On ligand pocket understanding (RNA-Site and RNA-Ligand), pretrained global encoders have the upper hand. We postulate that this might stem from their stronger ability to integrate the complex coordinated geometry of a site. Notably, on RNA-Ligand, BiHiTo obtains a whooping performance, which might stem for successful transfer learning from protein structures. On RNA-Prot, RNAcoder obtains stronger results, which could be explained by the local structural motif signature of RNA-protein binding.

Motivated by these insights, we explore a hybrid graph, with BiHiTo tokens as node features and RNAcoder as edge features. The performance of this approach interpolates their advantages and results in the most versatile approach.

Ablation study We study the effect of model design choices on RNAcoder's tokens performance on downstream tasks. The use of base pair and backbone tokens in addition to single residue tokens, which lies at the core of RNAcoder's novelty, significantly contributes to its performance across the three tasks. We attribute this gap to the fact that base pairing and stacking patterns, which directly influence RNA interactions (Hermann & Westhof, 1999), are more straightforward to infer at the scale of base pairs and dinucleotides than at the scale of individual nucleotides. The ablation study also highlights the importance of the equivariant design of the encoder, which also provides a strict improvement over the non-equivariant transformer across the three tasks. This could be explained by the less trivial nature of the reconstruction task (the equivariant graph neural network reconstructs the coordinates from the atomic distances, not from the coordinates themselves).

Table 3: Ablation study. We compare to using only residue tokens, without base pair and dinucleotide tokens (*no bp/dn*) and with a non-equivariant transformer encoder (*no EquiformerV3*). We also investigate variational (VAE) and quantized (VQ-VAE, FSQ) autoencoders.

	RNA-Site		RNA-Prot		RNA-Ligand	
	Balanced accuracy (%)	AUC (%)	Balanced accuracy (%)	AUC (%)	Balanced accuracy (%)	AUC (%)
RNAcoder	71.0 ± 0.8	78.9 ± 1.0	$\mathbf{65.6 \pm 0.4}$	70.5 ± 0.3	$\mathbf{54.0 \pm 4.1}$	$\mathbf{72.0 \pm 2.6}$
no bp/dn	69.6 ± 1.2	74.8 ± 1.7	63.7 ± 0.6	68.8 ± 0.6	47.9 ± 5.4	65.8 ± 6.4
no EquiformerV3	70.3 ± 2.4	76.2 ± 1.9	64.5 ± 0.5	69.5 ± 0.7	53.3 ± 4.0	71.8 ± 1.4
VAE	66.9 ± 2.3	73.6 ± 2.1	65.3 ± 0.8	$\mathbf{71.0 \pm 0.7}$	42.8 ± 7.4	63.2 ± 7.9
VQ-VAE	$\mathbf{72.6 \pm 0.5}$	$\mathbf{79.3 \pm 0.9}$	64.8 ± 1.3	69.1 ± 1.3	51.0 ± 8.3	67.0 ± 5.0
FSQ	71.7 ± 2.1	78.6 ± 1.6	63.7 ± 0.5	68.9 ± 0.5	46.9 ± 2.4	64.9 ± 2.5

We also benchmarked the classical predictive autoencoder design against generative formulations: the Variational Autoencoder (VAE) and its discrete variants (VQ-VAE (Van Den Oord et al., 2017) and FSQ (Mentzer et al., 2024)). While VAE generally underperforms, the quantized tokenizers results are more contrasted: VQ-VAE outperforms the classical autoencoder on the RNA-Site task and is outperformed on RNA-Prot and RNA-Ligand. FSQ gives a poorer performance than VQ-VAE in our experiments.

6 Conclusion

We introduced RNAcoder, a tokenizer of RNA structure producing tokens for three subunits: nucleotides, base pairs, and dinucleotides. Along the way, we proposed an RNAtom37 nomenclature handling chemical modifications, frame alignments for each subunit and a dataset for future tokenizer developments.

RNAcoder produces tokens that display a meaningful organization regarding base pair geometry and chemical modifications. Moreover, RNAcoder tokens transfer effectively to downstream prediction tasks directly related to RNA interactions: RNA-small-molecule binding site prediction, RNA-protein interface prediction, and ligand chemical cluster prediction. The innovative strategy of RNAcoder to tokenize base pairs and backbone connections in addition to individual nucleotides significantly contributes to this performance gain. This opens the door to the integration of RNAcoder embeddings to a wider array of graph neural networks for RNA structure-based predictions.

Beyond property prediction, RNAcoder tokens could transfer more broadly, serving as descriptors or as regression targets in their own right. As base-pair geometry prediction is drawing renewed attention in the RNA folding community, RNAcoder could serve as auxiliary or pretraining regression target for RNA folding.

Furthermore, as ongoing initiatives (Adams et al., 2025) are establishing a comprehensive mapping of RNA modifications, unveiling unknown modifications and further investigating their effects, we anticipate the need for deep learning models capable of taking into account nucleotide modifications to grow. RNAcoder offers a ready-to-use solution to tokenize modified nucleotides and embed them in graph neural networks, which could help investigating the functional impact of these modifications. Beyond RNAcoder, we introduced the RNAtom37 nomenclature which could serve as a modification-aware reconstruction objective for RNA structure prediction or generation models.

AI use statement

In this work, we used generative AI tools to propose or refine hypotheses, provide feedback on research methodology or experiments, implement methods, clean and reformat dataset. We have not used generative AI tools for research ideas identification nor drafting of the main text. Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures

or images, create or edit software code. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors and the parts of the code which we know by experience to be the most sensitive, such as data splitting, were written manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

The development of deep learning models for RNA structure holds the potential to advance RNA therapeutics, be it RNA vaccines, RNA interference, or RNA-targeting small molecules, offering opportunities to improve human health. However, we must ensure that the released models are used in ways benefiting humanity and to prevent their potential harmful uses.

Reproducibility statement

To ensure full reproducibility of our work, we make our code available anonymously at: `https://anonymous.4open.science/r/RNAcoder-D275`. We use seeds, which are provided in our repository (generally 5 seeds per experiment) in order to ease reproducibility. We further make the hyperparameters retained for the experiments performed in our paper, especially in Appendices C, D.4, **??**, and H.2.

Key data processing steps are extensively detailed, such as the local frame construction (section 3 and appendix B) and the mapping of RNA atoms to a common nomenclature (appendix A).

References

Mark D. Adams, Etienne Boileau, Janusz M. Bujnicki, Vivian G. Cheung, Silvestro G. Conticello, Peter Dedon, Christoph Dieterich, Angela Gallo, Jonathan Göke, Mark Helm, et al. Unlocking the regulatory code of rna: launching the human rnome project. *Genome Biology*, 26(1):382, 2025. doi: 10.1186/s13059-025-03824-y. URL `https://link.springer.com/article/10.1186/s13059-025-03824-y`.

Isaia Barbieri and Tony Kouzarides. Role of rna modifications in cancer. *Nature Reviews Cancer*, 20 (6):303–322, 2020.

Meital Bojan, Sanketh Vedula, Sai Advaith Maddipatla, Nadav Bojan Sellam, Anar Rzayev, Federico Napoli, Paul Schanda, and Alexander Bronstein. Representing local protein environments with machine learning force fields. In *International Conference on Learning Representations*, volume 2026, pp. 100760–100799, 2026.

Jiří Černỳ, Paulína Božíková, Jakub Svoboda, and Bohdan Schneider. A unified dinucleotide alphabet describing both rna and dna structures. *Nucleic acids research*, 48(11):6367–6381, 2020.

Yilin Chen, Tianyu Lu, Cizhang Zhao, Hannah Wayment-Steele, and Po-Ssu Huang. SLAE: Strictly local all-atom environment for protein representation. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=4HDRkXxqNs`.

Jessica L Childs-Disney, Xueyi Yang, Quentin MR Gibaut, Yuquan Tong, Robert T Batey, and Matthew D Disney. Targeting rna structures with small molecules. *Nature Reviews Drug Discovery*, 21(10):736–762, 2022.

Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

Rohit Dilip, Evan Zhang, Ayush Varshney, and David Van Valen. Flow autoencoders are effective protein tokenizers. In *International Conference on Learning Representations*, volume 2026, pp. 145901–145926, 2026.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 67350–67395, 2026.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Thomas Hermann and Eric Westhof. Non-watson-crick base pairs in rna-protein recognition. *Chemistry & biology*, 6(12):R335–R343, 1999.

Zhijian Huang, Yucheng Wang, Song Chen, Yaw Sing Tan, Lei Deng, and Min Wu. Deeprsma: a cross-fusion-based deep learning method for rna–small molecule binding affinity prediction. *Bioinformatics*, 40(12):btae678, 2024.

Chaitanya Joshi, Arian Jamasb, Ramon Viñas, Charles Harris, Simon Mathis, Alex Morehead, Rishabh Anand, and Pietro Liò. grnade: Geometric deep learning for 3d rna inverse design. In *International Conference on Learning Representations*, volume 2025, pp. 10166–10191, 2025.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.

Wissam Karroucha, Carlos Oliver, Veronique Stoven, and Vincent Mallet. Machine learning for rna-targeting drug design. *Journal of Chemical Information and Modeling*, 2026.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of rna base pairs. *Rna*, 7(4):499–512, 2001.

Neocles B Leontis and Craig L Zirbel. Nonredundant 3d structure datasets for rna knowledge extraction and benchmarking. In *RNA 3D structure analysis and prediction*, pp. 281–298. Springer, 2012.

Neocles B Leontis, Aurelie Lescoute, and Eric Westhof. The building blocks and motifs of rna architecture. *Current opinion in structural biology*, 16(3):279–287, 2006.

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Pan Tan, and Liang Hong. Prosst: Protein language modeling with quantized structure and disentangled attention. *Advances in Neural Information Processing Systems*, 37:35700–35726, 2024.

Zongzhao Li, Jiacheng Cen, Wenbing Huang, Taifeng Wang, and Le Song. Size-generalizable rna structure evaluation by exploring hierarchical geometries. In *International Conference on Learning Representations*, volume 2025, pp. 42496–42521, 2025.

Yi-Lun Liao, Alexander J Hoffman, Sabrina C Shen, Alexandre Duval, Sam Walton Norwood, and Tess Smidt. Equiformerv3: Scaling efficient, expressive, and general se (3)-equivariant graph attention transformers. *arXiv preprint arXiv:2604.09130*, 2026.

Andrew Liu, Axel Elaldi, Nathan Russell, and Olivia Viessmann. Bio2token: All-atom tokenization of any biomolecular structure with mamba. *arXiv preprint arXiv:2410.19110*, 2024.

Vincent Mallet, Carlos Oliver, Jonathan Broadbent, William L Hamilton, and Jérôme Waldispühl. Rnaglib: a python package for rna 2.5 d graphs. *Bioinformatics*, 38(5):1458–1459, 2022.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013a.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 746–751, 2013b.

Carlos Oliver, Vincent Mallet, Roman Sarrazin Gendron, Vladimir Reinharz, William L Hamilton, Nicolas Moitessier, and Jérôme Waldispühl. Augmented base pairing networks encode RNA-small molecule binding preferences. *Nucleic Acids Res.*, 48(14):7690–7699, 2020. note: highlight.

Jeffrey Ouyang-Zhang, Chengyue Gong, Yue Zhao, Philipp Krähenbühl, Adam Klivans, and Daniel Diaz. Distilling structural representations into protein sequence models. In *International Conference on Learning Representations*, volume 2025, pp. 93715–93727, 2025.

Francesco P Panei, Rachel Torchet, Herve Menager, Paraskevi Gkeka, and Massimiliano Bonomi. Hariboss: a curated database of rna-small molecules structures to aid rational drug design. *Bioinformatics*, 38(17):4185–4193, 2022.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Michael Sarver, Craig L Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B Leontis. Fr3d: finding local and composite recurrent structural motifs in rna 3d structures. *Journal of mathematical biology*, 56(1):215–252, 2008.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Preethi P Seelam, Purshotam Sharma, and Abhijit Mitra. Structural landscape of base pairs containing post-transcriptional modifications in rna. *Rna*, 23(6):847–859, 2017.

Robert P Sheridan. Time-split cross-validation as a method for estimating the goodness of prospective prediction. *Journal of chemical information and modeling*, 53(4):783–790, 2013.

Kaiwen Shi and Carlos Oliver. ENSEMBITS: an alphabet of protein conformational ensembles, May 2026. URL http://arxiv.org/abs/2605.13789. arXiv:2605.13789 [cs.LG].

Jesse Stombaugh, Craig L Zirbel, Eric Westhof, and Neocles B Leontis. Frequency and isostericity of rna base pairs. *Nucleic acids research*, 37(7):2294–2312, 2009.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6MRm3G4NiU.

Marcell Szikszai, Marcin Magnus, Sachin Kadyan, and Elena Rivas. On inputs to deep learning for rna 3d structure prediction. *bioRxiv*, pp. 2025–02, 2025.

Riddhish H Thakare, Amirhossein Taghavi, Jielei Wang, Jessica L Childs-Disney, Chenglong Li, Matthew D Disney, and Yanjun Li. Geometric deep learning reveals ligandable and cryptic rna binding small molecule pockets (smartpocket). *bioRxiv*, 2026.

Bernhard C Thiel, Irene K Beckmann, Peter Kerpedjiev, and Ivo L Hofacker. 3d based on 2d: Calculating helix angles and stacking patterns using forgi 2.0, an rna python library centered on secondary structure elements. *F1000Research*, 8:ISCB–Comm, 2019.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

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

Junkai Wang, Lijun Quan, Zhi Jin, Hongjie Wu, Xuhao Ma, Xuejiao Wang, Jingxin Xie, Deng Pan, Taoning Chen, Tingfang Wu, and others. MultiModRLBP: A Deep Learning Approach for Multi-Modal RNA-Small Molecule Ligand Binding Sites Prediction. *IEEE J. Biomed. Health Inf.*, 2024.

Siyu Wang, Drew Weissman, and Yizhou Dong. Rna chemistry and therapeutics. *Nature Reviews Drug Discovery*, 24(11):828–851, 2025.

Luis Wyss, Vincent Mallet, Wissam Karroucha, Karsten Borgwardt, and Carlos Oliver. A Comprehensive Library for RNA Structure-Function Modeling. In *ICLR 2025 Workshop on AI for Nucleic Acids*, 2025. note: highlight.

Yunpeng Xia, Jiayi Li, Yi-Ting Chu, Jiahua Rao, Jing Chen, Chenqing Hua, Dong-Jun Yu, Xiu-Cai Chen, and Shuangjia Zheng. Deciphering RNA–ligand binding specificity with GerNA-Bind. *Nat. Mach. Intell.*, pp. 1–13, 2025.

Junjie Xu, Artem Moskalev, Tommaso Mansi, Mangal Prakash, and Rui Liao. HARMONY: A multi-representation framework for RNA property prediction. In *ICLR 2025 Workshop on Machine Learning for Genomics Explorations*, 2025. URL https://openreview.net/forum?id=U3Ejoy1BG2.

Junjie Xu, Jiahao Zhang, Mangal Prakash, Xiang Zhang, and Suhang Wang. Dualequi: A dual-space hierarchical equivariant network for large biomolecules. *Advances in Neural Information Processing Systems*, 38:53448–53483, 2026.

Xinyu Yuan, Zichen Wang, Marcus D. Collins, and Huzefa Rangwala. Protein Structure Tokenization: Benchmarking and New Recipe. In *Forty-third International Conference on Machine Learning*, June 2025. URL https://openreview.net/forum?id=o4ANDWaomX.

Zhou Zhang, Hanqun Cao, Cheng Tan, Fang Wu, Pheng Ann Heng, and Tianfan Fu. Ribosphere: Learning unified and efficient representations of rna structures. *arXiv preprint arXiv:2603.19636*, 2026.

Ruochong Zheng, Yutian Liu, Yian Zhao, Zhiwei Nie, Xuehan Hou, Chang Liu, Siwei Ma, Youdong Mao, and Jie Chen. Bihito: Biomolecular hierarchy-inspired tokenization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28848–28856, 2026.

Heqin Zhu, Ruifeng Li, Ao Chang, Mingqian Li, Hongyang Chen, Peng Xiong, and S Kevin Zhou. Nc-bench and ncfold: a benchmark and closed-loop framework for rna non-canonical base-pair prediction. In *International Conference on Learning Representations*, volume 2026, pp. 135816–135832, 2026.

A RNAtom37 Nomenclature

RNAtom37 represents each RNA residue as a fixed-length array of 37 atom slots (PDB heavy-atom name, or empty), analogous to ATOM37 for proteins. Slots 0–10 and 12 form the sugar-phosphate backbone and are identical across all 22 residue types; slot 11 holds a $2'$-O-methyl substituent when present, slot 13 the glycosidic-bond atom, and slots 14–36 the base and any modification-specific substituents, tabulated separately for purine-derived and pyrimidine-derived residues.

This nomenclature is motivated by the need to map RNA atomic coordinates and identities to tensors, epecially to make them suitable reconstruction targets.

The task is further complicated by the fact that atoms occupying structurally analogous positions (e.g., the N1 in pyrimidines, N9 in purines, and C5 in pseudouridine) have distinct names in the PDB. Conversely, from one modified nucleotide type to another, some atom could have the same name in the PDB despite occupying completely different positions in the nucleotide (e.g., CM2).

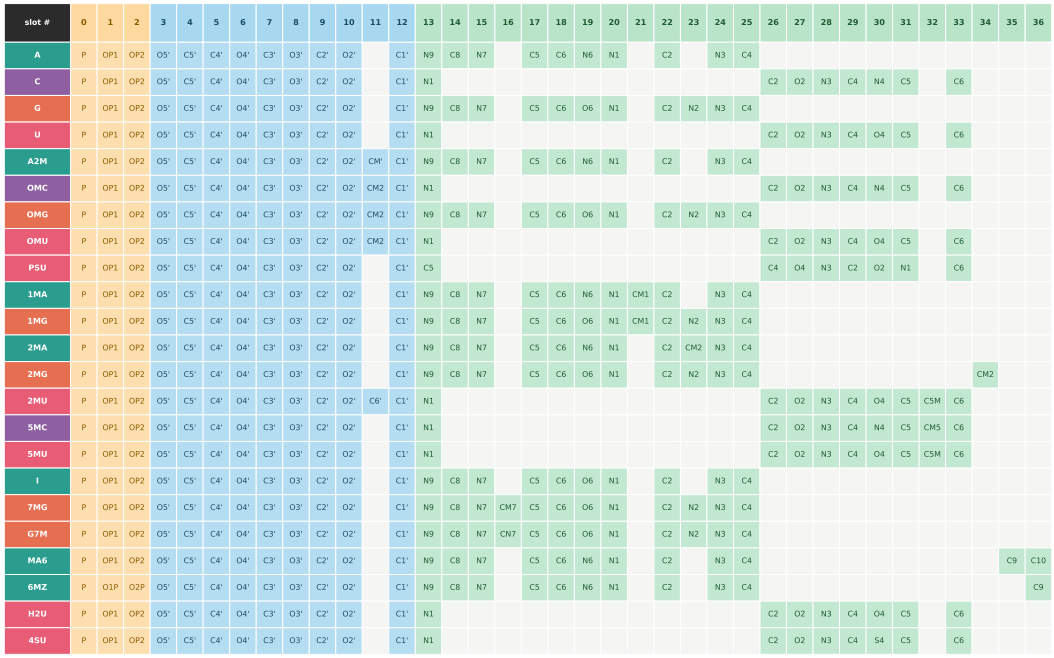

Figure 4: Full RNAtom37 nomenclature

B CONSTRUCTION OF LOCAL COORDINATE SYSTEMS

In the following, we denote the atom occupying the k-th position of RNAtom37 in residue i with A_i^k. Denoting atoms using this notation has the advantage of being robust to chemical modifications which might swap the atomic types among positions.

We define the local coordinate systems in the following way:

- **Nucleotides and backbone connections: backbone-centric frame.** For a residue i (or a dinucleotide between residues i and i+1), the origin of the frame is the phosphate atom (P_i).
  - We denote the vector following the direction of the backbone $\overrightarrow{P_iP_{i+1}}$ as $\mathbf{n_i}$
  - Then we denote the vector pointing towards the base as $\mathbf{t_i}$. In detail, we define $\mathbf{t_i} = \frac{\overrightarrow{P_iC'_{1i}}}{\|\overrightarrow{P_iC'_{1i}}\|}$.
  - Then we define $\mathbf{u_i} = \mathbf{z_i} \times \mathbf{t_i}$
  - Finally, we define $\mathbf{v_i} = \mathbf{z_i} \times \mathbf{u_i}$

  Finally, $\mathbf{u_i}$, $\mathbf{v_i}$, and $\mathbf{n_i}$ form the basis of residue i's local coordinate system.
- **Base pairs: base-centric frame.** For a base pair BP_{ij} between residues i and j, the origin of the frame is A_i^{14} (the base-anchoring atom, usually N1/N9 atom).
  - We denote the normal to the base plane of residue i pointing to the "top" direction of the face as $\mathbf{n_i}$. In detail, we compute the normal by singular value decomposition using the coordinates of al atoms that are part of the base, then we potentially flip the sign of the normal if it does not point to the top direction.
  - Then we denote the vector pointing in the direction of the Watson-Crick (WC) edge (Leontis & Westhof, 2001) of the base as $\mathbf{t_i}$. In detail, we define $\mathbf{t_i} = \frac{\overrightarrow{A_i^{13}A_i^{20}}}{\|\overrightarrow{A_i^{13}A_i^{20}}\|}$ for purines (usually equal to $\frac{\overrightarrow{N_{9i}N_{1i}}}{\|\overrightarrow{N_{9i}N_{1i}}\|}$) or $\mathbf{t_i} = \frac{\overrightarrow{A_i^{33}A_i^{28}}}{\|\overrightarrow{A_i^{33}A_i^{28}}\|}$ for pyrimidines (usually equal to $\frac{\overrightarrow{C_{6i}N_{3i}}}{\|\overrightarrow{C_{6i}N_{3i}}\|}$)

- Then we define $\mathbf{u_i} = \mathbf{z_i} \times \mathbf{t_i}$
- Finally, we define $\mathbf{v_i} = \mathbf{z_i} \times \mathbf{u_i}$ Finally, $\mathbf{u_i}$, $\mathbf{v_i}$, and $\mathbf{n_i}$ form the basis of residue i's local coordinate system.

C Details on the Autoencoders architecture

All three autoencoders (residue, base pair, backbone) share the same three-stage architecture

$$z_e = \mathcal{E}_\theta(\mathcal{S}), \qquad z = \mathcal{B}(z_e), \qquad \big(\hat{X}, \hat{P}^{\text{atom}}, \hat{P}^{\text{broad}}, \hat{P}^{\text{mod}}\big) = \mathcal{D}_\psi(z), \tag{4}$$

where $\mathcal{S}$ is the input sample (one residue, or a pair of residues) expressed in its local coordinate system (Appendix B), $\mathcal{E}_\theta$ is the encoder (EquiformerV3 or Transformer), $\mathcal{B}$ is the bottleneck ($\mathcal{B} = \mathrm{Id}$ for the plain autoencoder, see Appendix D.1 for VAE, VQ-VAE and FSQ), and $\mathcal{D}_\psi$ is an MLP decoder. We denote by $N \in \{1, 2\}$ the number of residues per sample ($N = 1$ for the residue autoencoder, $N = 2$ for the base-pair and backbone autoencoders), by $S = 37$ the number of RNATOM37 slots per residue, and by D the latent dimension ($D \in \{8, 16, 32, 64\}$; $D = 32$ for the retained residue encoder and $D = 64$ for the retained base-pair and backbone encoders). The encoder output $z_e \in \mathbb{R}^D$ is a single vector per sample, and the three autoencoders differ only by N, D and their training data. Table 4 summarizes all dimensions.

Input representation. Let $\mathcal{V}$ be the set of heavy atoms of the sample that are resolved in the structure. Each atom $v \in \mathcal{V}$ carries its local-frame position $\mathbf{x}_v \in \mathbb{R}^3$, its chemical element $Z_v \in \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{P}, \mathrm{S}\}$ and, for $N = 2$, a residue-membership index $r_v \in \{0, 1\}$ indicating whether it belongs to the first or second residue of the pair. Note that the encoders only receive element types, not the full atom names nor the residue identity (A/C/G/U/modified): the latter must be inferred from the geometry and are predicted by the decoder.

- *Graph input (EquiformerV3).* Atoms are the nodes of a graph $G = (\mathcal{V}, \mathcal{E})$ with $(u, v) \in \mathcal{E}$ iff $u \neq v$ and $d_{uv} = \|\mathbf{x}_u - \mathbf{x}_v\| < \tau$, with $\tau = 5$ Å for the retained models. Atoms outside of the RNATOM37 nomenclature are kept, so the graph size $|\mathcal{V}|$ is not bounded by NS.
- *Dense input (Transformer).* The sample is packed into the RNATOM37 layout: coordinates $X \in \mathbb{R}^{N \times S \times 3}$, element indices $t \in \{0, \dots, 5\}^{N \times S}$ (0 for an empty slot) and an occupancy mask $m \in \{0, 1\}^{N \times S}$.

EquiformerV3 encoder. We use the EquiformerV3 architecture (Liao et al., 2026), with all energy, force and stress heads removed. Each node carries an SO(3)-equivariant feature $\mathbf{h}_v \in \mathbb{R}^{(L+1)^2 \times C}$, i.e. C channels of real spherical-harmonic coefficients of degrees $\ell = 0, \dots, L$ (with $L = 2$, i.e. 9 coefficients per channel), and the SO(2) convolutions inside the attention are restricted to orders $|m| \leq M = 1$. We set the number of channels to the latent dimension, $C = D$, so that no final projection is needed.

Edge features. For each edge (u, v) the distance is expanded on $K = 64$ Gaussian radial basis functions with centers μ_k uniformly spaced on $[0, \tau]$,

$$\rho_k(d_{uv}) = \exp\left(-\frac{(d_{uv} - \mu_k)^2}{2\,(2\Delta\mu)^2}\right), \qquad \Delta\mu = \frac{\tau}{K - 1}, \quad k = 1, \dots, K, \tag{5}$$

and smoothly cut off by the polynomial envelope ($p = 5$, $s = d_{uv}/\tau$)

$$\varepsilon(d_{uv}) = \left(1 - \tfrac{(p+1)(p+2)}{2} s^p + p(p+2) s^{p+1} - \tfrac{p(p+1)}{2} s^{p+2}\right) \mathbf{1}[s < 1]. \tag{6}$$

The invariant edge feature is $\mathbf{e}_{uv} = \big[\boldsymbol{\rho}(d_{uv}) \,\|\, \mathbf{E}^{\text{src}}(Z_u) \,\|\, \mathbf{E}^{\text{tgt}}(Z_v)\big] \in \mathbb{R}^{64+2\times128}$, where $\mathbf{E}^{\text{src}}, \mathbf{E}^{\text{tgt}}$ are element embeddings of dimension 128. Every radial function ϕ below is an MLP $\mathbb{R}^{320} \to \mathbb{R}^{128} \to \mathbb{R}^{128} \to \mathbb{R}^{d_{\text{out}}}$ applied to $\mathbf{e}_{uv}$. Finally, $\mathbf{R}_{uv} \in SO(3)$ denotes a rotation aligning $\mathbf{x}_v - \mathbf{x}_u$ with the z-axis, and $\mathbf{W}(\mathbf{R}_{uv})$ the associated block-diagonal Wigner-D matrix acting on spherical-harmonic coefficients.

Node embedding. The initial features combine an element embedding and a residue-membership embedding, both injected in the invariant $\ell = 0$ component only (so that the membership signal does

not break equivariance), and an edge-degree embedding:

$$\mathbf{h}^{(0)}_{v,\ell=0} = \mathbf{E}^{Z}(Z_v) + \mathbf{E}^{r}(r_v) \in \mathbb{R}^{C}, \tag{7}$$

$$\mathbf{h}^{(0)}_{v} \mathrel{+}= \frac{1}{\bar{k}} \sum_{u \in \mathcal{N}(v)} \mathbf{W}(\mathbf{R}_{uv})^{-1} \Big[\varepsilon(d_{uv})\, \phi^{\text{deg}}(\mathbf{e}_{uv}) \Big]_{m=0}, \tag{8}$$

where $[\cdot]_{m=0}$ places the $(L+1) \times C$ outputs of ϕ^{deg} on the $m = 0$ coefficients of each degree (in the edge-aligned frame), and $\bar{k} \approx 23.4$ is a fixed average-degree normalization constant. For $N = 1$ the term $\mathbf{E}^{r}$ is omitted.

Transformer blocks. The features are then updated by $T = 3$ pre-norm EquiformerV3 blocks,

$$\mathbf{h}' = \mathbf{h}^{(t)} + \mathrm{GA}\big(\mathrm{LN}_1(\mathbf{h}^{(t)})\big), \tag{9}$$

$$\mathbf{h}^{(t+1)} = \mathbf{h}' + \mathrm{FFN}\big(\mathrm{LN}_2(\mathbf{h}')\big), \tag{10}$$

where LN is the equivariant (merged) layer normalization of EquiformerV3: the $\ell = 0$ component is re-centered across channels, then all components are divided by a single root-mean-square computed jointly over channels and degrees (each degree being weighted by $1/(2\ell+1)$), and finally rescaled by a learned per-degree, per-channel weight (with a learned bias on $\ell = 0$ only).

Equivariant graph attention (GA) uses $H = 2$ heads. For every edge $(u \to v)$, the features of the source and target nodes are concatenated, modulated per degree by a radial function, rotated into the edge frame and passed through two SO(2) linear layers separated by a gated SwiGLU activation σ:

$$\tilde{\mathbf{m}}_{uv} = \mathbf{W}(\mathbf{R}_{uv}) \Big(\big[\mathbf{h}_u \,\|\, \mathbf{h}_v \big] \odot \phi^{\text{att}}(\mathbf{e}_{uv}) \Big) \in \mathbb{R}^{(L+1)^2 \times 2C}, \tag{11}$$

$$\big(\mathbf{a}_{uv},\, \mathbf{g}_{uv}\big),\ \mathbf{m}^{1}_{uv} = \mathrm{SO2Lin}_1(\tilde{\mathbf{m}}_{uv}), \qquad \mathbf{m}^{1}_{uv} \in \mathbb{R}^{(L+1)^2 \times C_h}, \tag{12}$$

$$\mathbf{v}_{uv} = \mathrm{SO2Lin}_2\big(\sigma(\mathbf{m}^{1}_{uv}; \mathbf{g}_{uv})\big) \in \mathbb{R}^{(L+1)^2 \times HC_v}, \tag{13}$$

with $C_h = 32$ hidden channels and $C_v = 8$ value channels per head. The extra invariant outputs $\mathbf{a}_{uv} = (\mathbf{a}^{1}_{uv}, \ldots, \mathbf{a}^{H}_{uv})$, $\mathbf{a}^{h}_{uv} \in \mathbb{R}^{C_\alpha}$ with $C_\alpha = 16$, produced by the $m = 0$ part of $\mathrm{SO2Lin}_1$, give the attention logits, and $\mathbf{g}_{uv}$ are the gating scalars of σ. The attention weights are an envelope-weighted softmax over the neighbors of v,

$$s^{h}_{uv} = \boldsymbol{\omega}_h^{\top}\, \mathrm{SmoothLeakyReLU}\big(\mathrm{LN}(\mathbf{a}^{h}_{uv})\big), \qquad \alpha^{h}_{uv} = \varepsilon(d_{uv})\, \frac{\varepsilon(d_{uv}) \exp(s^{h}_{uv})}{\sum_{u' \in \mathcal{N}(v)} \varepsilon(d_{u'v}) \exp(s^{h}_{u'v})}, \tag{14}$$

with $\boldsymbol{\omega}_h \in \mathbb{R}^{C_\alpha}$ learned, so that contributions vanish smoothly at the cutoff. The messages of each head are weighted, rotated back to the global (local-frame) coordinate system, summed, and projected back to C channels by a degree-wise linear layer SO3Lin (a separate $C_{\text{in}} \times C_{\text{out}}$ matrix per degree ℓ, bias on $\ell = 0$ only):

$$\mathrm{GA}(\mathbf{h})_v = \mathrm{SO3Lin}\Big(\sum_{u \in \mathcal{N}(v)} \mathbf{W}(\mathbf{R}_{uv})^{-1} \big\|_{h=1}^{H} \alpha^{h}_{uv}\, \mathbf{v}^{h}_{uv} \Big) \in \mathbb{R}^{(L+1)^2 \times C}. \tag{15}$$

A dropout of rate 0.1 is applied to the attention weights α^{h}_{uv} during training (EquiformerV3 default); all other dropout and drop-path rates are 0.

Feed-forward network. The FFN acts node-wise, with hidden width $C_f = 64$:

$$\mathbf{u}_v = \mathrm{SO3Lin}_1(\mathbf{h}_v) \in \mathbb{R}^{(L+1)^2 \times C_f}, \tag{16}$$

$$\mathbf{u}'_v = \mathcal{F}^{-1}\Big(\mathrm{SwiGLU}_{\text{grid}}\big(\mathcal{F}(\mathbf{u}_v);\, \mathbf{h}_{v,\ell=0}\big) \Big) + \big[\mathrm{MLP}_{\text{scalar}}(\mathbf{h}_{v,\ell=0}) \big]_{\ell=0}, \tag{17}$$

$$\mathrm{FFN}(\mathbf{h})_v = \mathrm{SO3Lin}_2(\mathbf{u}'_v) \in \mathbb{R}^{(L+1)^2 \times C}, \tag{18}$$

where $\mathcal{F}$ (resp. $\mathcal{F}^{-1}$) projects spherical-harmonic coefficients onto (resp. back from) a discretized sphere, on which a gated SwiGLU MLP is applied pointwise, and the $\ell = 0$ component additionally receives the output of a scalar MLP.

Readout. After a final layer normalization, only the invariant $\ell = 0$ component is kept, and atoms are mean-pooled per sample:

$$z_e = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathrm{LN}\big(\mathbf{h}^{(T)}_{v}\big)_{\ell=0} \in \mathbb{R}^{D}. \tag{19}$$

Since only invariant features are pooled, z_e is invariant to rotations and translations of the input. In particular, z_e does not depend on the choice of local frame, whereas the decoder reconstructs coordinates expressed in the local frame. The encoder therefore has to capture the geometry relative to the local frame through the invariant features only, which makes its reconstruction task harder than that of the Transformer (see main text).

Transformer encoder. The non-equivariant encoder is a standard pre-norm Transformer (Vaswani et al., 2017) whose tokens are the NS RNATOM37 slots of the sample (37 tokens for residues, 74 for pairs). Each slot (n, a) is embedded by concatenating an element embedding, a position embedding and a one-hot encoding of its residue:

$$\mathbf{p}_{n,a} = \mathbf{W}_2\,\mathrm{SiLU}(\mathbf{W}_1\mathbf{x}_{n,a}) + \mathbf{b}_2 \in \mathbb{R}^{d_a}, \qquad \mathbf{W}_1 \in \mathbb{R}^{d_a\times 3}\ \text{(no bias)}, \tag{20}$$

$$\mathbf{h}^{(0)}_{n,a} = \mathbf{W}_{\text{in}}\big[\mathbf{E}^t(t_{n,a}) \,\|\, \mathbf{p}_{n,a} \,\|\, \mathbf{1}_n\big] + \mathbf{b}_{\text{in}} \in \mathbb{R}^d, \qquad \mathbf{W}_{\text{in}} \in \mathbb{R}^{d\times(2d_a+N)}, \tag{21}$$

with $d_a = 32$, model width $d = 128$, and $\mathbf{1}_n \in \{0,1\}^N$ the one-hot vector of residue n. No positional encoding of the slot index is added: apart from the residue one-hot, the encoder is invariant to permutations of the atoms, and the slot identity must be inferred from coordinates and element types. The token sequence is processed by $T = 3$ layers

$$\mathbf{H}' = \mathbf{H}^{(t)} + \mathrm{MHA}\big(\mathrm{LN}_1(\mathbf{H}^{(t)})\big), \tag{22}$$

$$\mathbf{H}^{(t+1)} = \mathbf{H}' + \mathbf{W}_2^f\,\mathrm{GELU}\big(\mathbf{W}_1^f\,\mathrm{LN}_2(\mathbf{H}') + \mathbf{b}_1^f\big) + \mathbf{b}_2^f, \qquad \mathbf{W}_1^f \in \mathbb{R}^{d_{\text{ff}}\times d}, \tag{23}$$

with $d_{\text{ff}} = 256$ and multi-head self-attention with $H = 4$ heads of dimension $d_k = d/H = 32$,

$$\mathrm{MHA}(\mathbf{H}) = \mathbf{W}^O\big\|_{h=1}^{H}\,\mathrm{softmax}\!\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}_h, \qquad M_{ij} = \begin{cases} 0 & \text{if } m_j = 1 \\ -\infty & \text{otherwise,} \end{cases} \tag{24}$$

where $\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h$ are linear projections of $\mathbf{H}$, and the key padding mask $\mathbf{M}$ prevents empty RNATOM37 slots from being attended to. After a final layer normalization, the occupied tokens are mean-pooled and linearly projected to the latent space:

$$z_e = \mathbf{W}_{\text{out}}\,\frac{\sum_{n,a} m_{n,a}\,\mathrm{LN}(\mathbf{h}^{(T)}_{n,a})}{\sum_{n,a} m_{n,a}} + \mathbf{b}_{\text{out}} \in \mathbb{R}^D, \qquad \mathbf{W}_{\text{out}} \in \mathbb{R}^{D\times d}. \tag{25}$$

Decoder. The decoder is shared by both encoder families. It consists of four independent MLP heads with hidden width $h = 64$ and GELU activations, whose outputs are reshaped per residue and per slot:

$$\hat{X} = \mathrm{reshape}_{N\times S\times 3}\Big(\mathbf{W}_3^c\,\mathrm{GELU}\big(\mathbf{W}_2^c\,\mathrm{GELU}(\mathbf{W}_1^c z)\big)\Big), \qquad \mathbf{W}_3^c \in \mathbb{R}^{3NS\times h}, \tag{26}$$

$$\hat{P}^{\text{atom}} = \mathrm{softmax}\Big(\mathrm{reshape}_{N\times S\times 41}\big(\mathbf{W}_2^a\,\mathrm{GELU}(\mathbf{W}_1^a z)\big)\Big), \qquad \mathbf{W}_2^a \in \mathbb{R}^{41NS\times h}, \tag{27}$$

$$\hat{P}^{\text{broad}} = \mathrm{softmax}\Big(\mathrm{reshape}_{N\times 5}\big(\mathbf{W}_2^b\,\mathrm{GELU}(\mathbf{W}_1^b z)\big)\Big), \qquad \mathbf{W}_2^b \in \mathbb{R}^{5N\times h}, \tag{28}$$

$$\hat{P}^{\text{mod}} = \mathrm{softmax}\Big(\mathrm{reshape}_{N\times 19}\big(\mathbf{W}_2^o\,\mathrm{GELU}(\mathbf{W}_1^o z)\big)\Big), \qquad \mathbf{W}_2^o \in \mathbb{R}^{19N\times h}, \tag{29}$$

with $\mathbf{W}_1^{\cdot} \in \mathbb{R}^{h\times D}$, $\mathbf{W}_2^c \in \mathbb{R}^{h\times h}$ and biases omitted for readability. $\hat{X}$ contains the local-frame coordinates of all 37 slots of each residue (only occupied slots enter $\mathcal{L}_{\text{coords}}$), $\hat{P}^{\text{atom}}$ the per-slot distribution over the 40 atom types and the "empty slot" class, and $\hat{P}^{\text{broad}}$, $\hat{P}^{\text{mod}}$ the hierarchical residue-type distributions of Appendix D.1. Since the decoder always predicts the full $N \times 37$ layout, the set of atoms present, their names and their positions are all reconstructed from z alone.

Bottlenecks. For the plain autoencoder, $z = z_e$. The generative variants (Appendix D.1) insert, between encoder and decoder, a linear layer $\mathbb{R}^D \to \mathbb{R}^{2D}$ producing $(\boldsymbol{\mu}, \log\boldsymbol{\sigma}^2)$ (VAE), or a pair of linear projections $\mathbb{R}^D \to \mathbb{R}^{D_q} \to \mathbb{R}^D$ around the quantization step (VQ-VAE, with D_q the dimension of the quantized space; FSQ, with D_q the number of quantized scalar channels). In all cases $z \in \mathbb{R}^D$, so the decoder is unchanged.

	EquiformerV3 encoder	Transformer encoder
Input tokens / nodes	resolved heavy atoms ($\lvert\mathcal{V}\rvert$)	$N \times 37$ RNATOM37 slots (masked)
Atom input features	element, residue membership	element, position, residue one-hot
Graph	$d_{uv} < 5$ Å	fully connected (attention)
Number of layers T	3	3
Width	$C = D$ channels, $L = 2$, $M = 1$	$d_a = 32$, $d = 128$
Attention heads	2 ($C_h = 32$, $C_\alpha = 16$, $C_v = 8$)	4 ($d_k = 32$)
FFN hidden width	$C_f = 64$ (gated SwiGLU on sphere grid)	$d_{\text{ff}} = 256$ (GELU)
Radial basis / edge channels	64 Gaussians / 128	–
Readout	mean pool of $\ell = 0$ features	masked mean pool + linear $d \to D$
Latent dimension D	8, 16, 32, 64 (retained: 32 residues, 64 base pairs and backbone)	
Encoder parameters ($N = 1$, $D = 32$)	0.89 M	0.41 M
Encoder parameters ($N = 2$, $D = 64$)	1.33 M	0.42 M
Decoder	4 MLP heads, hidden width $h = 64$, GELU; 0.12 M ($N = 1$, $D = 32$) / 0.24 M parameters ($N = 2$, $D = 64$)	

Table 4: Architecture dimensions of the autoencoders. N: number of residues per sample; D: latent dimension. Parameter counts are those of the trained checkpoints for the retained latent dimensions (plain autoencoder framework).

D Details on the Tokenizers Training

D.1 Training Losses

Reconstruction loss. Let $n \in \{1, \dots, N\}$ index the residue(s) of a sample ($N = 1$ for the single-residue autoencoder, $N = 2$ for the base-pair and backbone autoencoders), $a \in \{1, \dots, 37\}$ index the RNAtom37 atom slot within a residue, and $m_{n,a} \in \{0, 1\}$ be the ground-truth occupancy mask (1 if residue n has an atom at slot a).

Coordinate loss. $\mathcal{L}_{\text{coords}}$ is the mean squared error between predicted and ground-truth atomic coordinates, restricted to occupied slots:

$$\mathcal{L}_{\text{coords}} = \frac{\sum_{n,a,c} m_{n,a} \left(\hat{x}_{n,a,c} - x_{n,a,c}\right)^2}{3 \sum_{n,a} m_{n,a}}, \qquad c \in \{x, y, z\}. \tag{30}$$

Atom type loss. $\mathcal{L}_{\text{atom}}$ is the cross-entropy of the predicted atom type distribution against the ground-truth atom type class at every RNAtom37 slot of every residue, over a 41-way vocabulary that includes a dedicated "empty slot" class for positions unoccupied by that residue's type:

$$\mathcal{L}_{\text{atom}} = \frac{1}{N \cdot 37} \sum_{n,a} \ell\big(\hat{p}_{n,a}(\cdot),\, y^{\text{atom}}_{n,a}\big)\,. \tag{31}$$

Residue-type loss. $\mathcal{L}_{\text{res}}$ is a hierarchical negative log-likelihood over the 23 residue-type classes (the four canonical bases $\{A, C, G, U\}$ plus 19 modified-residue subclasses). The model predicts a 5-way broad distribution $P_{\text{broad}} \in \{A, C, G, U, \text{modified}\}$ and, conditionally, a 19-way distribution $P_{\text{mod}}(j \mid \text{modified})$ over modified-residue subclasses, so that

$$P(y = \text{A/C/G/U}) = P_{\text{broad}}(\text{A/C/G/U}), \qquad P(y = \text{mod}_j) = P_{\text{broad}}(\text{modified}) \cdot P_{\text{mod}}(j \mid \text{modified}), \tag{32}$$

$$\mathcal{L}_{\text{res}} = \frac{1}{N} \sum_{n} \ell(P(y_n = \cdot),\, y^{\text{res}}_n)\,. \tag{33}$$

Bottleneck regularization. The encoder output z_e is passed through one of four bottlenecks before decoding, each contributing its own $\mathcal{L}_{\text{bottleneck}}$ and weight schedule $\lambda(t)$:

- **Autoencoder.** z_e is passed through unchanged; $\mathcal{L}_{\text{bottleneck}} = 0$.
- **VAE.** z_e parameterizes a diagonal-Gaussian posterior $q(z \mid x) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$, sampled via the reparameterization trick during training. The bottleneck loss is the KL divergence to a standard normal prior, summed over the D latent dimensions and averaged over the batch:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{d=1}^{D} \left(\mu_d^2 + \sigma_d^2 - 1 - \log \sigma_d^2\right). \tag{34}$$

It enters the total loss at weight $\lambda(t) = w_{\text{KL}} \cdot \min\left(1, \frac{t+1}{T_{\text{warmup}}}\right)$, linearly warmed up from 0 to w_{KL} (default 10^{-4}) over T_{warmup} epochs (no warmup by default).

- **VQ-VAE.** z_e is projected to a quantization space, split into G groups (independent codebooks), and each group is replaced by its nearest entry z_q in a learned codebook. The bottleneck loss is the commitment loss, pulling the encoder toward its selected code,

$$\mathcal{L}_{\text{commit}} = \|z_e - \text{sg}[z_q]\|_2^2, \tag{35}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, weighted by a commitment cost β (default 0.25). When the codebook is updated by exponential moving average of its assignments (default), this is the entire bottleneck loss, entering the total at weight $\lambda(t) = 1$:

$$\mathcal{L}_{\text{VQ}} = \beta\, \mathcal{L}_{\text{commit}}. \tag{36}$$

When EMA updates are disabled, the codebook entries are instead learned parameters trained by an additional codebook loss $\mathcal{L}_{\text{codebook}} = \|z_q - \text{sg}[z_e]\|_2^2$, giving $\mathcal{L}_{\text{VQ}} = \beta\, \mathcal{L}_{\text{commit}} + \mathcal{L}_{\text{codebook}}$. Gradients flow from the decoder back to the encoder through the quantization step via a straight-through estimator.

- **FSQ.** z_e is projected to a small number of scalar channels, each independently bounded and rounded to a fixed integer grid (a straight-through estimator again carries the gradient through rounding). Because every grid point is reachable by construction, FSQ requires no auxiliary loss to shape the discrete space: $\mathcal{L}_{\text{bottleneck}} = 0$, so $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}}$ exactly.

D.2 Sampling strategy and connection with word2vec

Word2vec (Church, 2017) proposes an undersampling strategy for frequent words. We draw inspiration from it to design an undersampling strategy for frequent base pair geometric families and frequent secondary structure elements.

In word2vec, the probability of discarding an instance of a word c is given by $1 - \sqrt{\frac{t}{f_c}}$, where t is a hyperparameter defining the frequency threshold from which one wants to undersample and f_c is the frequency of the word c. Therefore, the probability for an instance of word c of being sampled should be $\sqrt{\frac{t}{f_c}}$. The total number of samples to draw from class c should then be $\text{round}\left(\sqrt{\frac{t}{f_c}} \cdot N_c\right)$.

In our case, we always keep all RNA subunits containing modified nucleotides and sample only the ones containing nonmodified nucleotides. Therefore, the number of nonmodified sampled subunits $\tilde{N}_c^{\text{non-mod}}$ should satisfy

$$N_c^{\text{modified}} + \tilde{N}_c^{\text{non-mod}} = \text{round}\left(\sqrt{\frac{t}{f_c}} \cdot N_c\right),$$

yielding:

$$\begin{aligned}
\tilde{N}_c^{\text{non-mod}} &= \text{round}\left(\sqrt{\frac{t}{f_c}} \cdot N_c\right) - N_c^{\text{modified}} \\
&= \text{round}\left(\sqrt{\frac{tN_{\text{total}}}{N_c}} \cdot N_c\right) - N_c^{\text{modified}} \\
&= \text{round}\left(\sqrt{tN_cN_{\text{total}}}\right) - N_c^{\text{modified}}.
\end{aligned}$$

We apply the same kind of subsampling for backbone and individual nucleotides, based on the secondary structure element of belonging instead of the base pair families. Since backbone connections and nucleotides located within a stem are more monotonous therefore easier to reconstruct, we apply to them a difficulty factor of 0.5, that is to say that the quota of bacbkone connections/base pairs part of a stem to be sampled is divided by 2 compared to the value it would get using the standard word2vec-style undersampling.

D.3 TRAINING DATASETS

We detail below the numbers of samples in our training datasets for residues, base pairs, and dinucleotides, along with their breakdown by Leontis-Westhof families for base pairs, and secondary structure context for nuclotides and dinucleotides. We used RNAglib (Mallet et al., 2022) for dataset preparation, FR3D (Sarver et al., 2008) for base pair family annotations and Forgi (Thiel et al., 2019) for secondary element annotations.

Table 5: Overview statistics of the three RNAcoder autoencoder training datasets (5 Å threshold, RNAglib non-redundant structure set): RNA-structure and pair counts by split, share of chemically modified nucleotides, and the per-epoch training-set size after stratified quota subsampling (train split only; val/test are used unsubsampled).

	Residues	Base pairs	Dinucleotides
RNAs (total / train / val / test)	3,098 / 2,170 / 309 / 619	2,310 / 1,617 / 231 / 462	3,098 / 2,170 / 309 / 619
Total	2,734,538	2,013,248	2,734,538
Train split	1,741,532	1,323,282	1,741,532
Val split	321,978	211,978	321,978
Test split	671,028	477,988	671,028
Containing modified nucleotides, total (%)	26,632 (1.0%)	18,850 (0.9%)	26,632 (1.0%)
Containing modified nucleotides, train (%)	11,529 (0.7%)	9,804 (0.7%)	11,529 (0.7%)
Distinct classes (n)	6 secondary elements	18 LW families	6 secondary elements
Epoch size after subsampling (train)	316,188	343,478	316,188
Epoch reduction vs. full train split	-81.8%	-74.0%	-81.8%

Table 6: Base-pair training set, by Leontis-Westhof (LW) family, before and after word2vec-inspired stratified quota subsampling ($t = 10^{-2}$). "Modified" = pairs containing a chemically modified nucleotide, always kept (phase 1); "Quota (non-mod.)" = phase-2 draw size for canonical-nucleotide pairs of that family, $\mathrm{round}(\sqrt{tN_cN_{total}}) - N_c^{modified}$, capped at $[0, N_c - N_c^{modified}]$; "Post-subsampling" = Modified + Quota, i.e. the family's expected count in one training epoch.

LW family	Pre-subsampling (train)		Modified	Quota (non-mod.)	Post-subsampling	
	n	% of train	n	n	n	% of epoch
cWW	996,228	75.3%	6,944	107,873	114,817	33.4%
tSS	49,916	3.8%	508	25,193	25,701	7.5%
cSS	49,337	3.7%	292	25,259	25,551	7.4%
tSH	37,332	2.8%	111	22,115	22,226	6.5%
tHS	37,332	2.8%	111	22,115	22,226	6.5%
tWH	27,595	2.1%	523	18,586	19,109	5.6%
tHW	27,594	2.1%	523	18,586	19,109	5.6%
tWW	17,944	1.4%	116	15,293	15,409	4.5%
tHH	14,550	1.1%	62	13,814	13,876	4.0%
cSW	11,656	0.9%	77	11,579	11,656	3.4%
cWS	11,656	0.9%	77	11,579	11,656	3.4%
cWH	8,968	0.7%	68	8,900	8,968	2.6%
cHW	8,958	0.7%	68	8,890	8,958	2.6%
tWS	8,747	0.7%	140	8,607	8,747	2.5%
tSW	8,747	0.7%	140	8,607	8,747	2.5%
cHS	2,974	0.2%	22	2,952	2,974	0.9%
cSH	2,974	0.2%	22	2,952	2,974	0.9%
cHH	774	0.1%	0	774	774	0.2%
Total	1,323,282	100.0%	9,804	333,674	343,478	100.0%

Table 7: Residue training set, by secondary structure element, before and after subsampling ($t = 10^{-2}$). "Stem" additionally carries the extra $\times 0.5$ stem-difficulty multiplier before the modified-count subtraction, on top of the word2vec frequency penalty. "Unclassified" = residues with no annotated secondary structure element ($\sim$0.1% of the train split): its 29 modified-nucleotide residues are still guaranteed every epoch via phase 1; its 1,720 non-modified residues are excluded from every quota and never drawn. Column definitions otherwise as in Table 6.

Secondary element	Pre-subsampling (train)		Modified	Quota (non-mod.)	Post-subsampling	
	n	% of train	n	n	n	% of epoch
Stem	711,598	40.9%	2,891	52,771	55,662	17.6%
Junction	373,511	21.4%	3,382	77,270	80,652	25.5%
Internal loop	343,142	19.7%	2,114	75,190	77,304	24.4%
Hairpin	273,896	15.7%	2,963	66,102	69,065	21.8%
Five-prime end	25,045	1.4%	105	20,780	20,885	6.6%
Three-prime end	12,591	0.7%	45	12,546	12,591	4.0%
Unclassified (excl.)	1,749	0.1%	29	0	29	0.0%
Total	1,741,532	100.0%	11,529	304,659	316,188	100.0%

Table 8: Backbone-connection training set, by secondary structure element of the 5' residue, before and after subsampling. Numerically identical to Table 7: both datasets are built from the same underlying 5'→3' backbone (B35) edges of the RNAglib graph – residues take the edge's 5' node, backbone connections take the edge itself – so they share the same RNA-level train/val/test split and the same per-class counts. Column definitions as in Table 6.

Secondary element	Pre-subsampling (train)		Modified	Quota (non-mod.)	Post-subsampling	
	n	% of train	n	n	n	% of epoch
Stem	711,598	40.9%	2,891	52,771	55,662	17.6%
Junction	373,511	21.4%	3,382	77,270	80,652	25.5%
Internal loop	343,142	19.7%	2,114	75,190	77,304	24.4%
Hairpin	273,896	15.7%	2,963	66,102	69,065	21.8%
Five-prime end	25,045	1.4%	105	20,780	20,885	6.6%
Three-prime end	12,591	0.7%	45	12,546	12,591	4.0%
Unclassified (excl.)	1,749	0.1%	29	0	29	0.0%
Total	1,741,532	100.0%	11,529	304,659	316,188	100.0%

D.4 Training Hyperparameters

The three autoencoders reported in the main results were trained with an identical protocol and the same optimization hyperparameters.

All three use the equivariant EquiformerV3 encoder with a distance-threshold atom-level neighbor graph (5 Å cutoff), a plain deterministic (non-variational, non-quantized) bottleneck, and the shared MLP decoder that reconstructs atom37 coordinates, atom type logits, and residue-type logits.

During our ablations, we also explored variational formnulations, as well as different cutoff graphs (10 Å and 15 Å)

Optimization used AdamW (PyTorch defaults: $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-2}) with a constant initial learning rate of 10^{-3}, reduced on plateau (factor 0.1, patience 5 epochs, monitored on the total validation loss) and a batch size of 128. Training ran for up to 100 epochs with early stopping on the total validation loss (patience 10 epochs, minimum improvement $\Delta = 10^{-4}$); the checkpoint with the lowest validation loss was kept for evaluation and embedding extraction. In practice all three runs triggered early stopping well before the 100-epoch cap, with best validation loss reached at epoch 42 (backbone), 54 (base pairs), and 53 (residues).

The splitting applied is time split Sheridan (2013). This ensures no RNA overlap occurs between the train, validation, and test split, ensuring no data leakage, especially for base pairs and dinucleotides

that might share some nucleotides. We did not use structure-based or family-based splitting as our objective is a strictly-local reconstruction objective.

The reconstruction loss combines coordinate MSE, atom type cross-entropy, and residue-type cross-entropy with weights $w_{\text{coords}} = 1.0$, $w_{\text{atom}} = 0.1$, $w_{\text{residue}} = 0.1$; the train split is drawn every epoch through the stratified quota sampler (Table 6–8) rather than uniform shuffling, while validation and test use the full, unsubsampled splits. All three runs use global seed 42 (Python/NumPy/PyTorch).

Table 9: Training hyperparameters for the three RNAcoder autoencoders used in the main results. All hyperparameters not listed here (e.g. EquiformerV3 architecture widths, decoder hidden dimension) are left at their code defaults and are identical across the three runs.

Hyperparameter	Value
Shared across all three encoders	
Encoder	EquiformerV3
Framework (bottleneck)	plain autoencoder (deterministic)
Graph construction	distance threshold, 5 Å cutoff
Optimizer	AdamW ($\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-2})
Learning rate (initial)	10^{-3}
LR schedule	ReduceLROnPlateau (factor 0.1, patience 5 epochs, on val. loss)
Batch size	128
Max. epochs	100
Early-stopping patience	10 epochs (min. improvement $\Delta = 10^{-4}$, on val. loss)
Loss weights (w_{coords}, w_{atom}, w_{residue})	1.0, 0.1, 0.1
Train-split sampling	stratified quota sampler (per epoch); val/test unsubsampled
Seed	42
Encoder-specific	
Latent dimension (backbone)	64
Latent dimension (base pairs)	64
Latent dimension (residues)	32
Best epoch reached (backbone / base pairs / residues)	42 / 54 / 53

E RECONSTRUCTION PERFORMANCE OF THE AUTOENCODERS

This appendix details the reconstruction quality of the three autoencoders used throughout the paper: the single-residue autoencoder, the backbone-connection autoencoder, and the base-pair autoencoder. All three use the EquiformerV3 encoder over a 5 Å atom-neighborhood graph and are trained with latent dimensionality 32 for the single-residue model and 64 for the backbone and base-pair models. All metrics below are computed on the held-out test split at each model's best-validation-loss checkpoint.

Reconstruction quality by structural context and base-pair geometry. Tables 10 and 11 break the coordinate RMSD down by, respectively, the secondary structure context of the residue (stem, hairpin, internal loop, junction, and the two chain termini) and the Leontis-Westhof geometric family of the pair (used for the base-pair autoencoder).

Table 10: Test-set RMSD (Å) by structural context, for the residue and backbone autoencoders.

Structural context	**Residue (lat. 32)**	**Backbone (lat. 64)**
Five-prime end	0.198	0.100
Hairpin	0.170	0.106
Internal loop	0.132	0.100
Junction	0.138	0.110
Stem	0.099	0.042
Three-prime end	0.182	0.230

Table 11: Test-set RMSD (Å) by Leontis-Westhof geometric family, for the base-pair autoencoder (lat. 64). `c`/`t` denote cis/trans orientation; `W`/`H`/`S` denote the Watson-Crick, Hoogsteen, and sugar edges.

LW family	RMSD	LW family	RMSD
cWW	0.039	tWW	0.069
cWH	0.065	tWH	0.063
cWS	0.057	tWS	0.062
cHW	0.069	tHW	0.056
cHH	0.090	tHH	0.053
cHS	0.072	tHS	0.038
cSW	0.062	tSW	0.065
cSH	0.077	tSH	0.039
cSS	0.052	tSS	0.056

F VISUALIZATION OF RNA SUBUNIT RECONSTRUCTIONS

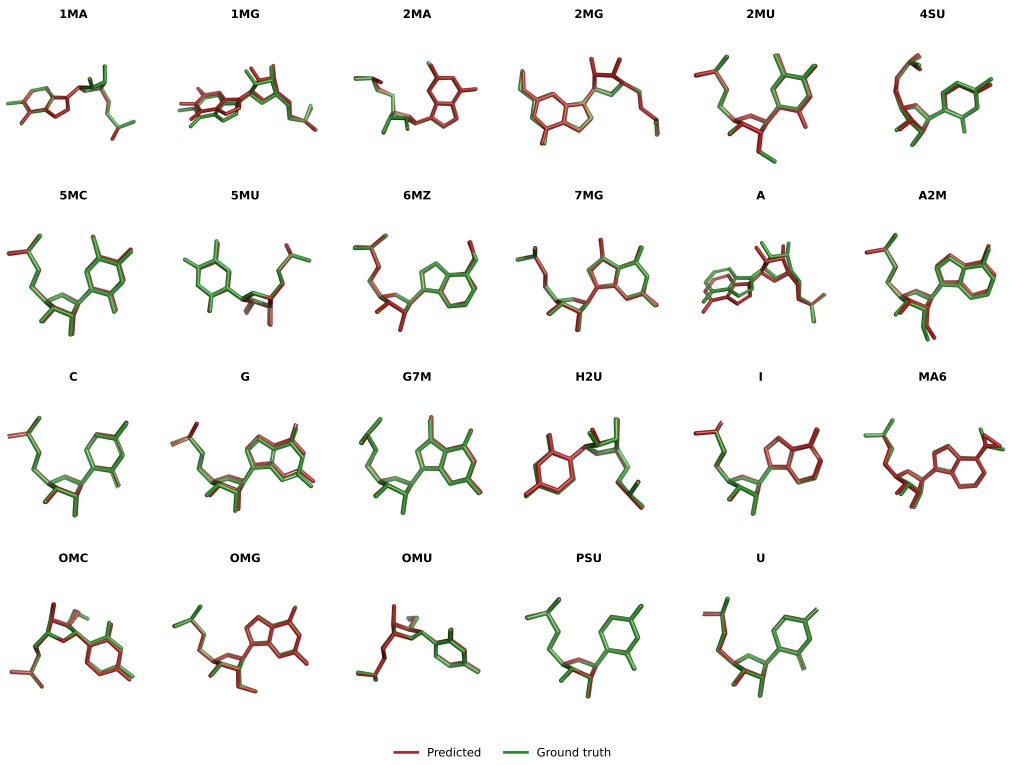

Figure 5: Visualization of ground truth (in green) and reconstructed (in red) nucleotides across the 23 standard and modified nucleotide types supported by RNAtom37 nomenclature

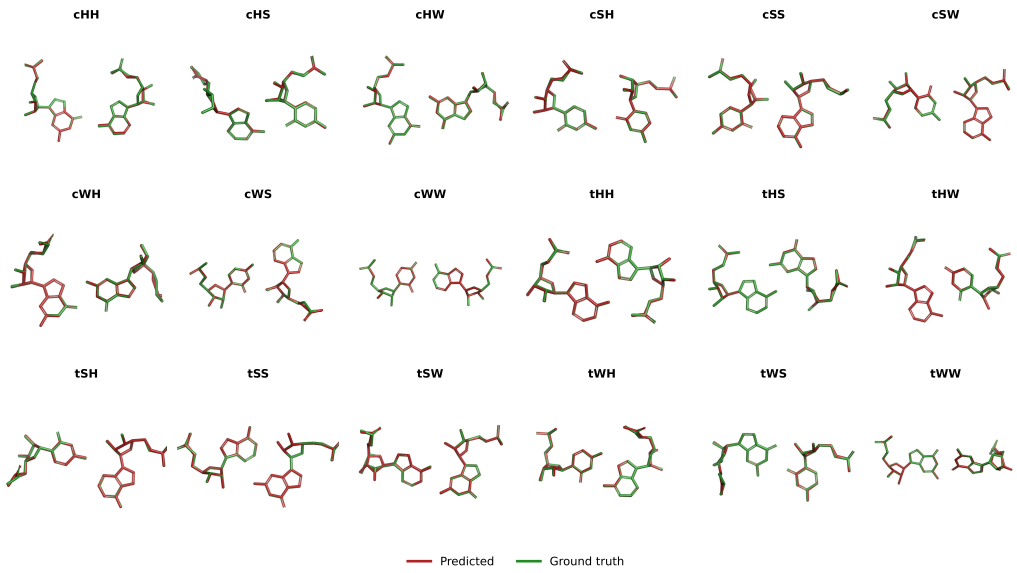

Figure 6: Visualization of ground truth (in green) and reconstructed (in red) base pairs across the 18 Leontis-Westhof geometric families

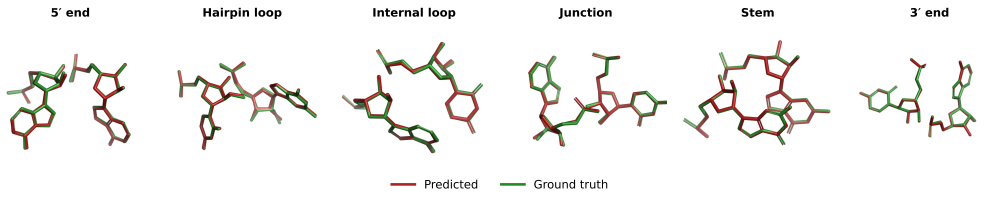

Figure 7: Visualization of ground truth (in green) and reconstructed (in red) dinucleotides across the 5 secondary structure contexts as defined by Forgi (Thiel et al., 2019)

G ADDITIONAL RESULTS ON THE INTERPRETABILITY OF THE LATENT SPACE: DETAILED RESULTS

We detail the results of the compositionality experiment inspired by (Mikolov et al., 2013a). For each 2′-O-methylated nucleotide (a modification that has the advantage of being well represented in our datasets for all nucleotide types), we compute, for all nucleotide types X other than A, *predicted* = centroid of the embeddings of 2′-O-methylated X nucleotides-centroid of X nucleotides+centroid of A nucleotides. Then, we compute the cosine similarities of *predicted* to all modified nucleotides and find the nearest neighbor. In all cases, the closest one is 2MA (2′-O-methylated adenosine), which is the target. We repeated the experiment for all other nucleotide types than A and found the same results. This seems to indicate that RNAcoder has well learned the compositionality rules of nucleotide modifications.

Figure 8 displays the full matrix of cosine similarities between

- on the one hand, the centroids and the centroids of the nucleotides embeddings, of the 2′-O-methylated nucleotide embeddings, and various operations over these embeddings
- on the other hand, the embeddings of genuine 2′-O-methylated nucleotides (in columns)

We observe that the embeddings of operations pointing to the same target (e.g., OMC-C+A, OMG-G+A, and OMU-U+A all theoretically point to A2M, since they follow the pattern methylated X - X +A which should equal "methylated A") are systematically the closest ones, thus forming yellow rows in the diagonal blocks (apart from the first row of each diagonal block which corresponds to the nonmodified nucleotide).

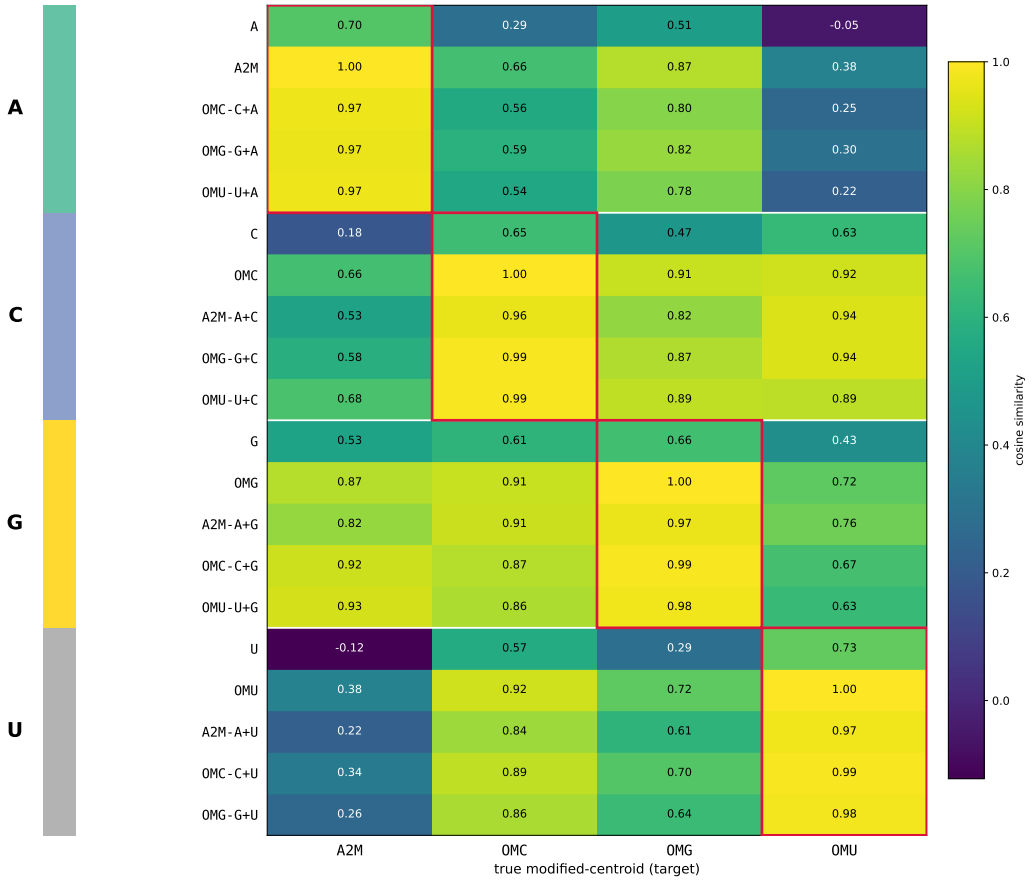

Figure 8: Compositionality of the residue embeddings regarding nucleotide identities and modifications.

H Downstream Tasks

H.1 Datasets

We evaluate the residue-level embeddings produced by our autoencoders on two nucleotide-level binary classification tasks drawn from the RNAglib task library (Wyss et al., 2025): **RNA-Site**, predicting whether a nucleotide belongs to a small-molecule binding site, and **RNA-Prot**, predicting whether a nucleotide belongs to a protein-binding interface. Both tasks assign every nucleotide of an RNA structure a binary label and use the same input feature (nucleotide identity, one-hot over $\{A, C, G, U\}$). We use RNAglib's dataset construction and splitting logic without modification; this Appendix summarizes the underlying label definitions and reports the resulting dataset statistics for reproducibility.

RNA-Site: small-molecule binding-site prediction. A nucleotide is labeled positive if any of its heavy atoms lies within 6.0 Å of a heavy atom of a small-molecule ligand satisfying the HARIBOSS filtering criteria (Panei et al., 2022): molecular weight between 160 and 1000 Da, at least one carbon atom, atomic composition restricted to $\{$C, H, N, O, Br, Cl, F, P, Si, B, Se, S$\}$, and excluding water. Two further exclusions remove spurious "ligands": (i) any heteroatom group that is covalently integrated into a polymer backbone (e.g. a $5'$ cap analog or a charged tRNA's aminoacyl ester) is identified and excluded via a bond-distance check (< 2.0 Å along a curated set of backbone linkages) rather than via its deposited chemical-component type, since several genuine ligands (e.g. ppGpp, argininamide) are themselves typed as polymer-linking components; (ii) a candidate ligand is required to be genuinely bound to the RNA rather than incidentally located near it within a larger RNA–protein assembly, i.e. to contact at least 2 RNA residues and at least as many RNA as protein residues within 4.0 Å. Structures are partitioned into connected components (the same locally-connected structural units used throughout our data pipeline), and a component is retained only if it contains at least 10 positively labeled nucleotides and between 15 and 500 nucleotides in total. The dataset is built at full redundancy, i.e. without collapsing near-identical structures, following the task's default configuration. Splits are assigned with RNAglib's structure-based `ClusterSplitter`, which clusters RNAs by USalign structural similarity so that near-duplicate structures cannot straddle the train/validation/test boundary.

RNA-Prot: protein-binding interface prediction. A nucleotide is labeled positive if any of its heavy atoms lies within 4.0 Å of any heavy atom belonging to a protein chain, where a chain only counts as protein if it contains at least 2 amino-acid residues (excluding free amino acids that share a residue name with one of the twenty canonical amino acids, e.g. a riboswitch's own bound ligand). We additionally relabel as negative any nucleotide whose only contributing protein chain(s) are known crystallization chaperones (domains such as the U1A RRM or Fab/Fv antibody fragments that are routinely co-crystallized to nucleate crystal contacts and are not biological partners of the RNA). Connected components are retained only if they contain at least one positive and one negative nucleotide, and between 15 and 500 nucleotides in total. The dataset is built using the non-redundant (`nr`) resolution proposed by (Leontis & Zirbel, 2012), collapsing near-identical structures to a single representative, following the task's default configuration. Splits are assigned with the same structure-based `ClusterSplitter` as RNA-Site.

Dataset statistics. Table 12 reports the resulting dataset sizes. Both tasks are imbalanced toward the negative class, more markedly so for RNA-Site, whose smaller and stricter ligand definition yields a 22.1% positive rate against 46.3% for RNA-Prot; this is why both tasks default to balanced accuracy rather than raw accuracy as their evaluation metric.

Table 12: Dataset statistics for the two downstream nucleotide-classification tasks. RNA counts are the number of connected structural components (train / validation / test); nucleotide counts and positive rates are pooled over all splits.

Task	Positive label	# RNAs (train/val/test)	# Nucleotides	Positive rate
RNA-Site	within 6.0 Å of a bound ligand	137 (96/20/21)	10,924	2,414 (22.1%)
RNA-Prot	within 4.0 Å of a protein chain	852 (596/128/128)	51,850	24,016 (46.3%)

H.2 TRAINING DETAILS

All downstream results (RNA-Site, RNA-Prot, and their GCN/RGCN baselines) share the same GNN backbone, optimizer, and evaluation protocol; only the input node/edge features change between RNAcoder and the baselines. This Appendix reports the resulting configuration in full.

Architecture. The backbone is a Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2018). The 4-dimensional one-hot nucleotide identity (or, for RNAcoder node-token runs, the pretrained residue embedding) is first mapped to the hidden width by a linear layer followed by ReLU and dropout, then passed through relational graph-convolution layers, each followed by GraphNorm (which was found to significantly stabilize the training), a ReLU nonlinearity, and dropout (rate 0.2). For RNAcoder runs, the pretrained base-pair/backbone embeddings are fused into each layer's message additively (a non-linear MLP projection of the edge embedding added to the source node's feature); for the RGCN baseline, edges instead carry only their discrete relation id (no learned embedding). Every node's final hidden state is passed through a linear classifier head (width $128 \to 1$) followed by a sigmoid to produce one binding probability per nucleotide.

GNN Hyperparameters To ensure that every model is compared at its own best capacity rather than at a shared, possibly favorable-to-one configuration, the GNN hyperparameters were selected by an identical, standardized architecture search, repeated independently for every (task, model) combination: 3 tasks (RNA-Site, RNA-Prot, RNA-Ligand) $\times$ 4 models (GCN and RGCN on the 2.5D graph, RiboSphere embeddings on the RGCN backbone, and RNAcoder). For each combination, we sweep the hidden width h, the number of message-passing layers L and the learning rate. For RNA-Site and RNA-Prot (and for RiboSphere on all three tasks), the grid is the full factorial $h \in \{32, 128, 256\} \times L \in \{2, 4, 6\} \times \text{lr} \in \{10^{-4}, 10^{-3}\}$ (18 configurations). For RNA-Ligand, the search was run in two stages: a first grid $h \in \{32, 128\} \times L \in \{2, 3\} \times \text{lr} \in \{5\times10^{-5}, 10^{-4}, 5\times10^{-4}, 10^{-3}\}$ (16 configurations), followed by a larger-model grid $h \in \{128, 256\} \times L \in \{3, 4, 5, 6\}$ at the best learning rate of the first stage for each model. Every configuration is trained with 3 seeds (seeds 0–2) under exactly the same protocol as the final benchmark: batch size of 16 RNAs, 500 epochs with early stopping disabled, GraphNorm after each layer, class-weighted binary cross-entropy, and restoration of the checkpoint with the best validation balanced accuracy. For every run, we record the peak validation balanced accuracy over the 500 epochs. The selected configuration is the one maximizing this peak averaged over the 3 seeds. Test data plays no role in the selection. The selected configurations, reported in Table 13, are then evaluated with 5 seeds: seeds 0–2 are the architecture-search runs themselves (identical experiments), and seeds 3 and 4 are trained additionally with the same protocol. All other hyperparameters are shared across models and tasks.

Table 13: GNN hyperparameters selected by the architecture search (mean peak validation balanced accuracy over 3 seeds) for each task and model, used in Table 2. h: hidden width; L: number of message-passing layers.

Task	Model	h	L	Learning rate
RNA-Site	2.5D graph (GCN)	32	4	10^{-3}
	2.5D graph (RGCN)	128	6	10^{-4}
	RiboSphere	256	6	10^{-4}
	RNAcoder	256	6	10^{-3}
RNA-Prot	2.5D graph (GCN)	256	6	10^{-4}
	2.5D graph (RGCN)	128	4	10^{-4}
	RiboSphere	32	6	10^{-4}
	RNAcoder	256	4	10^{-4}
RNA-Ligand	2.5D graph (GCN)	128	2	10^{-3}
	2.5D graph (RGCN)	128	5	10^{-4}
	RiboSphere	128	4	10^{-3}
	RNAcoder	256	6	10^{-4}
Shared across all runs		batch size 16, 500 epochs, no early stopping, best-val checkpoint		

Optimization. We use the Adam optimizer with learning rate 5×10^{-5}, no weight decay, and a constant learning-rate schedule (no warmup or annealing), with a batch size of 16 RNAs. Models are trained for up to 500 epochs; the checkpoint with the best validation balanced accuracy is restored for test-set evaluation. The loss is binary cross-entropy on the per-nucleotide logits, with the positive class reweighted by the fixed ratio of negative to positive nucleotides in the task's training split (to counteract the class imbalance reported in Appendix H.1, Table 12).

Evaluation and reporting. We report test-set AUC (`sklearn.metrics.roc_auc_score`) and balanced accuracy (`sklearn.metrics.balanced_accuracy_score`). For nucleotide-level prediction tasks (RNA-Site and RNA-Prot), metrics are computed at the RNA-level then averaged across all RNAs, in order to mimic a virtual screening setting in which drug discovery teams are generally interested in ranking candidate binding site within a specific target RNA. Metrics are reported at the best-validation-balanced-accuracy checkpoint, averaged over 5 random seeds (mean $\pm$ standard deviation across seeds); seeds control model initialization and minibatch ordering only, splits are fixed (Appendix H.1). Each run trains on a single NVIDIA V100 GPU.

I Ablation of autoencoder architectures

Table 14: Edge-only (base-pair / backbone encoder, no residue tokens) gridsearch: downstream GNN mean $\pm$ std balanced accuracy and AUC across seeds, the validation split. Bold = best, underlined = second-best, within each column.

		RNA Site		RNA Prot	
Base-pair enc.	Backbone enc.	Val Bacc.	Val AUC	Val Bacc.	Val AUC
eq lat8	eq lat8	72.0 ± 0.9	$\underline{78.3 \pm 0.6}$	64.0 ± 0.4	67.6 ± 0.2
	eq lat16	70.3 ± 0.8	76.1 ± 1.5	63.6 ± 0.9	67.6 ± 1.1
	eq lat32	70.9 ± 1.3	76.4 ± 1.2	64.0 ± 0.6	68.3 ± 0.9
	eq lat64	70.3 ± 2.4	76.1 ± 1.6	64.3 ± 0.3	68.8 ± 0.7
	tf lat8	69.7 ± 1.5	74.8 ± 2.6	64.6 ± 0.5	68.8 ± 0.6
	tf lat16	69.5 ± 1.9	74.5 ± 2.2	64.0 ± 0.9	67.8 ± 1.7
	tf lat32	67.7 ± 1.7	72.5 ± 2.7	64.3 ± 0.2	68.8 ± 0.5
	tf lat64	69.7 ± 1.2	75.0 ± 1.1	63.7 ± 0.4	67.2 ± 0.6
eq lat16	eq lat8	71.2 ± 0.7	76.8 ± 1.4	63.6 ± 0.5	67.7 ± 0.3
	eq lat16	71.0 ± 1.1	77.1 ± 0.9	65.0 ± 0.3	68.8 ± 0.7
	eq lat32	71.2 ± 1.7	77.8 ± 1.7	64.6 ± 0.4	68.9 ± 0.5
	eq lat64	71.4 ± 1.2	77.2 ± 1.5	65.3 ± 0.5	69.8 ± 0.6
	tf lat8	67.7 ± 1.3	73.4 ± 0.9	63.7 ± 0.5	67.4 ± 0.7
	tf lat16	70.3 ± 1.3	76.5 ± 1.3	65.4 ± 0.3	$\underline{70.3 \pm 0.2}$
	tf lat32	67.6 ± 1.2	73.2 ± 1.8	65.0 ± 0.6	69.6 ± 0.5
	tf lat64	69.1 ± 0.8	74.2 ± 0.9	64.0 ± 0.2	67.9 ± 0.4
eq lat32	eq lat8	71.0 ± 0.5	76.3 ± 1.5	63.7 ± 0.6	67.5 ± 0.8
	eq lat16	71.0 ± 0.4	76.7 ± 1.0	64.7 ± 0.5	68.4 ± 0.9
	eq lat32	$\underline{72.2 \pm 1.1}$	$\mathbf{78.4 \pm 1.5}$	64.7 ± 0.3	69.4 ± 0.3
	eq lat64	69.2 ± 1.0	76.0 ± 0.8	$\underline{65.5 \pm 0.1}$	70.0 ± 0.4
	tf lat8	67.8 ± 1.2	73.7 ± 1.6	63.1 ± 0.3	67.3 ± 0.5
	tf lat16	69.9 ± 0.8	75.3 ± 0.9	64.7 ± 0.7	69.1 ± 1.3
	tf lat32	68.6 ± 0.8	73.7 ± 0.7	$\mathbf{65.6 \pm 0.7}$	70.0 ± 0.6
	tf lat64	68.8 ± 0.3	74.3 ± 1.0	63.7 ± 0.4	67.5 ± 0.6
eq lat64	eq lat8	70.6 ± 0.8	76.7 ± 1.2	63.7 ± 0.4	67.5 ± 0.8
	eq lat16	70.0 ± 1.3	76.5 ± 1.2	64.5 ± 0.2	68.4 ± 0.1
	eq lat32	71.5 ± 2.1	77.7 ± 2.0	64.9 ± 0.7	69.6 ± 0.8
	eq lat64	70.7 ± 0.4	77.1 ± 0.2	64.8 ± 0.3	69.1 ± 0.4
	tf lat8	68.0 ± 0.9	74.7 ± 0.7	63.1 ± 0.6	66.7 ± 0.5
	tf lat16	69.6 ± 1.3	74.5 ± 1.8	64.9 ± 0.5	69.5 ± 0.7

Continued on next page

Table 14 – continued from previous page

Base-pair enc.	Backbone enc.	RNA Site		RNA Prot	
		Val Bacc.	Val AUC	Val Bacc.	Val AUC
	tf lat32	66.4 ± 1.9	72.1 ± 2.1	65.4 ± 0.7	69.9 ± 0.5
	tf lat64	68.0 ± 1.1	73.7 ± 1.0	63.5 ± 0.3	67.5 ± 0.6
tf lat8	eq lat8	70.6 ± 0.9	76.4 ± 0.9	63.7 ± 0.1	68.1 ± 0.5
	eq lat16	69.2 ± 1.8	74.8 ± 1.4	63.2 ± 0.5	66.7 ± 0.9
	eq lat32	68.6 ± 1.5	73.9 ± 1.6	63.8 ± 0.5	67.7 ± 1.0
	eq lat64	69.6 ± 2.7	75.7 ± 2.7	64.3 ± 0.2	68.5 ± 0.4
	tf lat8	69.7 ± 0.6	75.7 ± 0.4	64.9 ± 0.5	69.9 ± 1.0
	tf lat16	68.4 ± 1.5	73.8 ± 2.1	64.0 ± 0.7	67.9 ± 1.1
	tf lat32	67.5 ± 1.8	72.8 ± 1.9	64.2 ± 0.3	68.4 ± 0.4
	tf lat64	68.4 ± 1.5	74.7 ± 1.7	63.6 ± 0.4	67.8 ± 1.1
tf lat16	eq lat8	70.1 ± 1.1	76.0 ± 1.1	63.4 ± 0.6	67.1 ± 0.8
	eq lat16	**72.4 ± 0.9**	78.0 ± 0.9	64.5 ± 0.4	68.1 ± 0.7
	eq lat32	70.3 ± 0.9	75.9 ± 2.1	64.1 ± 0.4	68.5 ± 0.4
	eq lat64	70.4 ± 1.8	76.5 ± 2.1	64.6 ± 0.6	68.8 ± 0.9
	tf lat8	67.1 ± 0.8	72.9 ± 1.1	63.7 ± 0.5	67.4 ± 0.8
	tf lat16	71.4 ± 0.4	77.3 ± 0.3	64.9 ± 0.4	69.9 ± 0.3
	tf lat32	68.4 ± 1.3	73.4 ± 2.0	64.8 ± 0.2	69.6 ± 0.3
	tf lat64	68.7 ± 1.8	74.0 ± 1.3	63.5 ± 0.1	67.2 ± 0.4
tf lat32	eq lat8	69.2 ± 1.8	74.9 ± 2.0	63.1 ± 0.6	67.1 ± 0.7
	eq lat16	69.5 ± 1.3	75.0 ± 1.5	63.9 ± 0.5	67.5 ± 0.8
	eq lat32	71.8 ± 0.4	78.2 ± 1.1	64.3 ± 0.3	68.4 ± 0.3
	eq lat64	68.5 ± 1.1	74.1 ± 1.0	64.9 ± 0.3	69.4 ± 0.6
	tf lat8	67.5 ± 0.9	72.6 ± 2.4	63.4 ± 0.4	67.3 ± 0.6
	tf lat16	70.4 ± 1.1	75.6 ± 1.9	64.2 ± 0.8	68.7 ± 1.0
	tf lat32	70.1 ± 0.9	76.5 ± 0.7	65.3 ± 0.6	69.4 ± 0.8
	tf lat64	68.6 ± 0.9	74.4 ± 1.2	63.7 ± 0.1	67.5 ± 0.6
tf lat64	eq lat8	70.2 ± 0.7	76.2 ± 0.7	63.1 ± 0.6	67.1 ± 0.4
	eq lat16	69.7 ± 1.6	75.7 ± 1.8	63.8 ± 0.5	67.8 ± 0.5
	eq lat32	69.2 ± 1.2	74.4 ± 1.9	64.8 ± 0.6	69.2 ± 0.6
	eq lat64	70.5 ± 0.7	76.9 ± 1.0	64.6 ± 0.3	69.0 ± 0.3
	tf lat8	68.1 ± 1.4	74.6 ± 1.9	63.3 ± 0.4	66.7 ± 0.7
	tf lat16	69.3 ± 2.1	74.8 ± 2.1	64.2 ± 0.7	68.3 ± 1.1
	tf lat32	66.2 ± 0.6	71.8 ± 0.9	65.5 ± 0.5	**70.5 ± 0.7**
	tf lat64	68.1 ± 1.1	74.7 ± 1.5	63.5 ± 0.5	67.5 ± 0.6

By default, we train the three autoencoders independently. We also benchmark a mode that we call *unified encoder* in which we jointly train the three autoencoders: a single encoder is used for all three input types (single residue, base pair, backbone connection), and its decoder always reconstructs three residues: the residue itself, its potential base pair partner, and its potential backbone connection partner. For a single-residue input, only the reconstruction of that one residue contributes to the loss; for a base-pair or dinucleotide input, both the reconstruction of the residue and its partner contribute. We report the results in table 16. We find the unified encoder to provide lower downstream performance than the best of the separate encoders.

Table 15: Residue encoder × base-pair/backbone encoder pair gridsearch: downstream GNN mean ± std balanced accuracy and AUC across seeds, the validation split. Bold = best, underlined = second-best, within each column.

		RNA Site		RNA Prot	
Base-pair / backbone enc.	Residue enc.	Val Bacc.	Val AUC	Val Bacc.	Val AUC
bp eq lat32 / bb eq lat32	eq lat8	70.0 ± 1.2	76.1 ± 1.4	64.1 ± 0.6	68.3 ± 0.6
	eq lat16	71.2 ± 0.5	77.5 ± 1.0	65.1 ± 0.5	70.1 ± 0.5
	eq lat32	70.4 ± 1.2	75.5 ± 2.3	65.6 ± 0.5	70.1 ± 0.5
	eq lat64	71.5 ± 1.0	77.1 ± 1.3	66.3 ± 0.5	**71.4 ± 0.5**
	tf lat8	68.1 ± 1.0	73.8 ± 1.7	64.6 ± 0.3	69.6 ± 0.4
	tf lat16	68.4 ± 0.5	73.1 ± 1.0	65.2 ± 0.5	70.3 ± 0.5
	tf lat32	68.2 ± 1.7	72.9 ± 1.5	65.5 ± 0.5	70.6 ± 0.4
	tf lat64	70.1 ± 1.9	75.0 ± 2.0	64.7 ± 0.8	69.8 ± 0.8
bp eq lat32 / bb eq lat64	eq lat8	70.4 ± 1.4	76.4 ± 1.4	65.0 ± 0.3	69.5 ± 0.7
	eq lat16	70.4 ± 1.2	77.4 ± 1.6	65.5 ± 0.4	69.9 ± 0.5
	eq lat32	70.7 ± 1.0	75.9 ± 0.9	**66.4 ± 0.4**	71.1 ± 0.4
	eq lat64	71.0 ± 1.4	76.0 ± 1.0	66.2 ± 0.4	70.7 ± 0.7
	tf lat8	69.5 ± 1.2	74.4 ± 1.3	65.2 ± 0.3	70.4 ± 0.4
	tf lat16	68.3 ± 1.1	73.9 ± 1.0	65.6 ± 0.7	70.4 ± 0.8
	tf lat32	67.9 ± 1.5	73.3 ± 1.9	65.8 ± 0.7	70.9 ± 0.9
	tf lat64	70.2 ± 1.4	75.7 ± 2.1	65.4 ± 0.5	70.4 ± 1.1
bp eq lat64 / bb eq lat64	eq lat8	70.9 ± 1.1	76.7 ± 1.3	64.4 ± 0.6	69.0 ± 0.4
	eq lat16	71.7 ± 0.9	76.8 ± 0.7	65.1 ± 0.3	69.8 ± 0.4
	eq lat32	71.7 ± 0.6	76.3 ± 0.7	65.9 ± 0.4	71.1 ± 0.2
	eq lat64	70.8 ± 1.2	75.9 ± 1.3	66.3 ± 0.3	71.0 ± 0.3
	tf lat8	70.0 ± 1.3	74.6 ± 2.1	64.7 ± 0.4	69.6 ± 0.3
	tf lat16	67.4 ± 0.7	73.1 ± 0.8	65.0 ± 0.5	70.0 ± 0.3
	tf lat32	69.4 ± 0.6	74.6 ± 0.8	65.7 ± 0.4	71.0 ± 0.4
	tf lat64	70.0 ± 2.2	74.9 ± 1.8	64.8 ± 0.5	70.4 ± 0.6
bp tf lat8 / bb eq lat8	eq lat8	71.6 ± 0.8	76.9 ± 0.7	63.5 ± 0.3	68.1 ± 0.5
	eq lat16	70.5 ± 0.6	76.5 ± 1.5	64.0 ± 0.4	69.1 ± 0.4
	eq lat32	**72.0 ± 0.5**	**77.9 ± 0.5**	64.7 ± 0.4	69.6 ± 0.3
	eq lat64	71.4 ± 1.1	77.6 ± 1.2	65.4 ± 0.6	70.7 ± 0.6
	tf lat8	70.9 ± 1.2	76.3 ± 1.3	64.4 ± 0.3	69.2 ± 0.2
	tf lat16	69.6 ± 0.9	75.0 ± 0.3	64.7 ± 0.5	69.9 ± 0.4
	tf lat32	70.5 ± 0.7	76.4 ± 0.8	64.9 ± 0.3	70.1 ± 0.5
	tf lat64	70.9 ± 1.1	77.0 ± 0.9	64.7 ± 0.4	69.4 ± 0.6

Table 16: Unified encoder (used simultaneously as the residue, base-pair and backbone encoder) GNN search: downstream GNN mean ± std balanced accuracy and AUC across seeds, the validation split. Bold = best, underlined = second-best, within each column.

	RNA Site		RNA Prot	
Unified encoder	Val Bacc.	Val AUC	Val Bacc.	Val AUC
lat8	71.0 ± 0.6	76.1 ± 0.6	**64.6 ± 0.6**	68.7 ± 0.8
lat16	**71.8 ± 0.8**	**78.4 ± 1.3**	64.1 ± 0.3	68.5 ± 0.8
lat32	71.3 ± 0.6	77.2 ± 0.9	64.2 ± 0.5	**69.0 ± 0.4**