Multi-State RNA Design with Geometric Multi-Graph Neural Networks

Chaitanya K. Joshi [1] **Arian R. Jamasb** [1] **Ramon Viñas** [1] **Charles Harris** [1] **Simon Mathis** [1] **Pietro Liò** [1]

Abstract

Computational RNA design has broad applications across synthetic biology and therapeutic development. Fundamental to the diverse biological functions of RNA is its conformational flexibility, enabling single sequences to adopt a variety of distinct 3D states. Currently, computational biomolecule design tasks are often posed as inverse problems, where sequences are designed based on adopting a single desired structural conformation. In this work, we propose **gRNAde**, a **g**eometric **RNA de**sign pipeline that operates on sets of 3D RNA backbone structures to explicitly account for and reflect RNA conformational diversity in its designs. We demonstrate the utility of gRNAde for improving native sequence recovery over single-state approaches on a new large-scale 3D RNA design dataset, especially for multi-state and structurally diverse RNAs. Our code is available at https://github.com/chaitjo/geometric-rna-design

1. Introduction

RNA molecules have been at the forefront of biology and biotechnology in recent years (Doudna and Charpentier, 2014; Pardi et al., 2018). There is growing interest in designing RNA-based nanotechnology with specific functions, including therapeutic aptamers, riboswitch biosensors and ribozyme catalysts (Guo, 2010; Zhu et al., 2022).

On the other hand, proteins have been the primary focus in the artificial intelligence community. Availability of large-scale data combined with advances in geometric deep learning (Bronstein et al., 2021) have revolutionized protein 3D structure prediction (Jumper et al., 2021) and inverse design (Dauparas et al., 2022; Gainza et al., 2023).

However, applications of geometric deep learning for computational RNA design are underexplored compared to proteins due to paucity of 3D structural data (Das, 2021; Townshend et al., 2021). Most tools for RNA design primarily focus on secondary structure without considering 3D geometry (Churkin et al., 2018), or use non-learnt algorithms for aligning 3D RNA motifs (Yesselman et al., 2019).

In addition to limited 3D data, the key technical challenge is that RNA is more dynamic than proteins. The same RNA can adopt multiple distinct conformational states to create and regulate complex biological functions (Ganser et al., 2019; Hoetzel and Suess, 2022; Ken et al., 2023). Computational RNA design pipelines must account for this conformational flexibility to engineer new functionality.

To enable multi-state RNA design, we develop **gRNAde**, a geometric deep learning-based pipeline for RNA sequence design conditioned on *multiple* backbone conformations. gRNAde explicitly accounts for RNA conformational flexibility via a novel multi-Graph Neural Network architecture which independently encodes a set of conformers via message passing, followed by conformer order-invariant pooling and sequence design, illustrated in Figure 1.

We demonstrate the utility of gRNAde's multi-state architecture by improving native sequence recovery over single-state baselines adopted from protein design, especially for structurally diverse RNAs for which we found single-state models to be unreliable.

2. gRNAde: Geometric RNA Design Pipeline

RNA backbones as geometric multi-graphs. We are given a set of k RNA backbone 3D point clouds $\{\vec{B}^{(1)}, \ldots, \vec{B}^{(k)}\}$ and the corresponding sequence of n nucleotides. For each backbone point cloud $\vec{B}^{(k)}$, we first construct an independent geometric graph $\mathcal{G}^{(k)}$ as follows: (1) We build a 3-bead coarse-grained representation of the RNA backbone, retaining the coordinates for P, C4', N1 (pyrimidine) or N9 (purine) for each nucleotide (Dawson et al., 2016).[1] (2) Each nucleotide i is assigned a node in the geometric graph with the 3D coordinate $\vec{x}_i \in \mathbb{R}^3$ corresponding to each C4'. (3) Each node is connected to its 10 nearest neighbours as measured by the distance $\|\vec{x}_i - \vec{x}_j\|_2$ between their C4' atoms. (4) Node features include for-

[1]University of Cambridge, UK. Correspondence to: Chaitanya K. Joshi <chaitanya.joshi@cl.cam.ac.uk>.

---

[1]This 3-bead representation can 'describe RNA backbone conformations fully in most cases' (Wadley et al., 2007), which will be critical for future work on RNA backbone generative models.
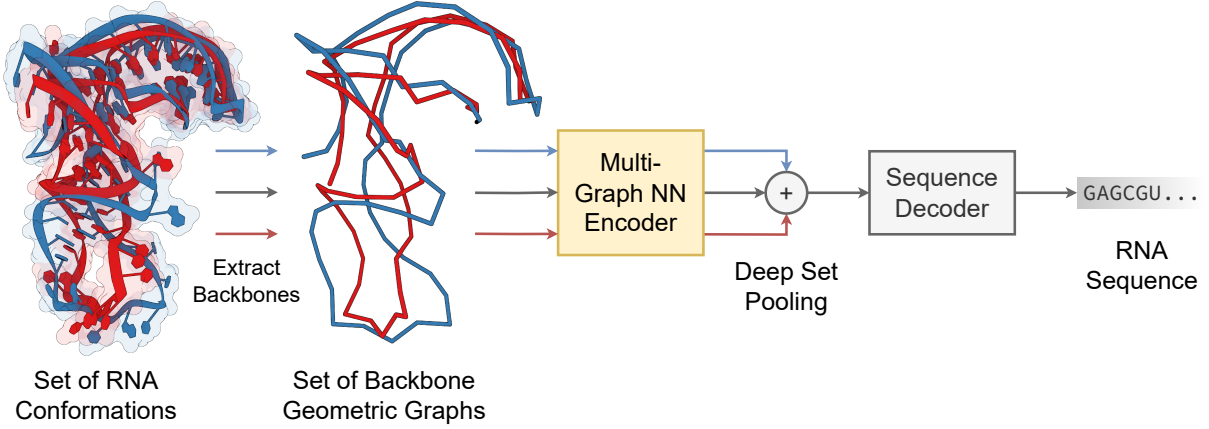
Figure 1: **The gRNAde pipeline for multi-state RNA design.** gRNAde processes a set of conformers via a multi-GNN encoder which is equivariant to conformer order, followed by conformer order-invariant pooling and sequence design.

ward and reverse unit vectors along the backbone sequence ($\vec{x}_{i+1} - \vec{x}_i$ and $\vec{x}_i - \vec{x}_{i-1}$) as well as unit vectors from C4' to P and N1/N9, similar to the featurization used for protein design (Ingraham et al., 2019; Jing et al., 2020). (5) Similarly, edge features include unit vectors from the source to destination node $\vec{x}_j - \vec{x}_i$, the distance $\|\vec{x}_i - \vec{x}_j\|_2$ encoded by 16 RBFs, and the distance along the backbone $j - i$ encoded by 16 sinusoidal encodings. Thus, each RNA backbone is represented as $\mathcal{G}^{(k)} = (\boldsymbol{A}^{(k)}, \boldsymbol{S}^{(k)}, \vec{\boldsymbol{V}}^{(k)})$ with the scalar features $\boldsymbol{S}^{(k)} \in \mathbb{R}^{n \times f}$, vector features $\vec{\boldsymbol{V}}^{(k)} \in \mathbb{R}^{n \times f' \times 3}$, and $\boldsymbol{A}^{(k)}$, an $n \times n$ adjacency matrix. For clear presentation and without loss of generality, we omit edge features and use f, f' to denote scalar/vector feature channels.

The set of geometric graphs $\{\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(k)}\}$ is merged into what we term a 'multi-graph' $\mathcal{M} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{V}})$ by stacking the set of scalar features $\{\boldsymbol{S}^{(1)}, \dots, \boldsymbol{S}^{(k)}\}$ into one tensor $\boldsymbol{S} \in \mathbb{R}^{n \times k \times f}$ along a new axis for the set size k. Similarly, the set of vector features $\{\vec{\boldsymbol{V}}^{(1)}, \dots, \vec{\boldsymbol{V}}^{(k)}\}$ is stacked into one tensor $\vec{\boldsymbol{V}} \in \mathbb{R}^{n \times k \times f' \times 3}$. Lastly, the set of adjacency matrices $\{\boldsymbol{A}^{(1)}, \dots, \boldsymbol{A}^{(k)}\}$ are merged via a union $\cap$ into one single joint adjacency matrix $\boldsymbol{A}$.

Geometric multi-GNN encoder. Message passing on the unified multi-graph adjacency enables efficient and *independent* processing of each conformation, while maintaining permutation equivariance of the updated feature tensors along both the first (no. of nodes) and second (no. of conformations) axes. This works by operating on only the last hidden feature channels and accounting for the extra conformations axis during message propagation and aggregation; see Figure 3 and the pseudocode in the Appendix.

We use multiple $O(3)$-equivariant GVP-GNN (Jing et al., 2020) layers to update scalar features $\boldsymbol{s}_i \in \mathbb{R}^{k \times f}$ and vector features $\vec{\boldsymbol{v}}_i \in \mathbb{R}^{k \times f' \times 3}$ for each node i:

$$\boldsymbol{m}_i, \vec{\boldsymbol{m}}_i := \sum_{j \in \mathcal{N}_i} \text{MSG}\big((\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i), (\boldsymbol{s}_j, \vec{\boldsymbol{v}}_j), \boldsymbol{e}_{ij}\big) \quad (1)$$

$$\boldsymbol{s}_i', \vec{\boldsymbol{v}}_i' := \text{UPD}\big((\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i), (\boldsymbol{m}_i, \vec{\boldsymbol{m}}_i)\big). \quad (2)$$

In principle, gRNAde can use any more expressive geometric GNN layers (Joshi et al., 2023).

Our multi-Graph Neural Network encoder is easy to implement in message passing frameworks (e.g. PyTorch Geometric (Fey and Lenssen, 2019)) and can be used as a *plug-and-play* extension for any geometric GNN pipeline to incorporate the multi-state inductive bias. It serves as an elegant alternative to batching all the conformations, which we found required major alterations to message passing and pooling depending on downstream tasks.

Conformation order-invariant pooling. The final encoder representations in gRNAde account for multi-state information while being invariant to the permutation of the conformational ensemble. To achieve this, we perform a Deep Set pooling (Zaheer et al., 2017) over the conformations axis after the final encoder layer to reduce $\boldsymbol{S} \in \mathbb{R}^{n \times k \times f}$ and $\vec{\boldsymbol{V}} \in \mathbb{R}^{n \times k \times f' \times 3}$ to $\boldsymbol{S}' \in \mathbb{R}^{n \times f}$ and $\vec{\boldsymbol{V}}' \in \mathbb{R}^{n \times f' \times 3}$:

$$\boldsymbol{S}', \vec{\boldsymbol{V}}' := \frac{1}{k} \sum_{i=1}^{k} \big(\boldsymbol{S}[:, i], \vec{\boldsymbol{V}}[:, i]\big). \quad (3)$$

Importantly, a simple sum or average pooling does not introduce any new learnable parameters to the pipeline and is flexible to handle a variable number of conformations.

Sequence decoding. We feed the final encoder representations $\boldsymbol{S}', \vec{\boldsymbol{V}}'$ to autoregressive GVP-GNN decoder layers to predict the probability of the four possible nucleotides for each node. Decoding proceeds according to the RNA

sequence order from the 5' end to 3' end. We can also consider unordered and masked decoding schemes (Dauparas et al., 2022) depending on the task at hand.

3. New Large-scale Dataset for 3D RNA Design

Large-scale RNASolo data. We create a machine learning-ready dataset for RNA inverse design using RNASolo (Adamczyk et al., 2022), a recent repository of RNA 3D structures extracted from solo RNAs, protein-RNA complexes, and DNA-RNA hybrids in the PDB. We used structures at resolution ≤3Å, resulting in 3,764 unique RNA sequences for which a total of 11,538 structures are available (∼100× more RNAs than Townshend et al. (2021)).

Dataset statistics are shown in Figure 2, illustrating the diversity of our dataset in terms of sequence length, number of structures per sequence, as well as structural variations among conformations per sequence.

Data splitting. We split the RNA sequences into training (∼3000 samples), validation (200 samples) and test (200 samples) sets under three different scenarios:

(1) *Average RMSD split.* We split sequences based on the average pairwise RMSD among the structures, with training, validation and test sets getting progressively more structurally diverse. This tests models' ability to design RNA with multiple distinct conformational states.

(2) *Number of Structures split.* We split sequences such that training, validation and test sets have progressively more structures available per sequence. This evaluates whether models can handle large and variable structural ensembles.

(3) *Sequence Identity split.* We cluster the sequences based on nucleotide similarity using CD-HIT (Fu et al., 2012) with an identity threshold of 80%. This allows us to create splits containing biologically dissimilar clusters of RNAs.

Evaluation metric. We evaluate performance using native sequence recovery: the average % of nucleotides correctly recovered in 100 sampled sequences. We also compute average accuracy of identifying the correct nucleotide.

Model hyperparameters. All models use 3 encoder and 3 decoder GVP-GNN layers, with 128 scalar/16 vector node features, and 32 scalar/1 vector edge features. All models are trained for 100 epochs using the Adam optimiser with an initial learning rate of 0.001, which is reduced by a factor 0.9 when validation performance plateaus with patience of 5 epochs. We report the test set performance for the model checkpoint with the best validation set accuracy. All results are averaged across 3 random seeds.

Single-state baseline. A data point from our dataset consists of a pair (set of structures, corresponding sequence). As the number of structures per sequence is highly variable,

we train gRNAde at various settings for maximum number of conformers k per sequence.[2] Thus, we sample exactly k structures per sequence during data loading. Setting $k = 1$ corresponds to a standard single structure inverse design pipeline (Ingraham et al., 2019; Jing et al., 2020), which we use as a baseline to demonstrate the advantage of gRNAde's multi-state design paradigm.

4. Results

Multi-state architecture boosts sequence recovery. We hypothesised that using multiple RNA conformations would improve sequence recovery, especially for multi-state RNA. In Table 1, we evaluated recovery of gRNAde models trained at varying number of conformations sampled per RNA. Overall, we observed higher recovery with multiple conformations across three different splits, demonstrating gRNAde's ability to handle structural diversity, variable number of structures, and biological dissimilarity.

Case study for theophylline aptamer. In Table 2, we consider a scenario where we want to design a new aptamer for theophylline, a common drug for pulmonary disease treatment, based on backbones structures from a known aptamer with multiple conformational states (Wrist et al., 2020). This case study highlights the unreliability of single-state models depending on an arbitrary choice of input structure and designed sequence. gRNAde is explictly designed to predict consistent sequences while accounting for all available conformations, leading to improved performance.

5. Discussion

We introduce gRNAde, a geometric deep learning pipeline for RNA sequence design conditioned on multiple backbone conformations. Future work will consider more realistic RNA design scenarios and experimental validation, such as partial/masked design (Yesselman et al., 2019).

To the best of our knowledge, gRNAde is the first geometric deep learning model for explicit multi-state biomolecule representation learning – the pipeline is generic and can be repurposed for other learning tasks on sets of biomolecular conformations, including multi-state protein design. The multi-graph architecture can be further improved by leverage theoretical advances in deep learning on sets (Wagstaff et al., 2022; Maron et al., 2020).

Finally, we are hopeful that advances in RNA-only structure determination with computationally assisted cryo-EM (Kappel et al., 2020; Bonilla and Kieft, 2022) will further enable geometric deep learning for RNA modelling and design.

---

[2]Note that setting a higher or lower k does not impact the number of parameters for gRNAde; the model is flexible to handle variable k during inference or when not batching multiple samples.

Max. #Conformers	Avg. RMSD Split Recovery@100		#Structures Split Recovery@100		Seq. Identity Split Recovery@100	
k	Mean	Median	Mean	Median	Mean	Median
1 (baseline)	59.58% ±1.06	61.99% ±1.85	63.06% ±1.36	70.19% ±2.27	43.86% ±1.67	46.01% ±2.59
3	62.02% ±1.19	65.03% ±2.38	63.57% ±0.24	70.89% ±1.29	**45.41%** ±0.46	**48.13%** ±1.69
5	61.99% ±2.08	64.59% ±2.60	64.66% ±0.87	72.09% ±0.27	44.41% ±1.20	47.31% ±1.39
8	**63.90%** ±1.68	**66.43%** ±0.99	**65.46%** ±1.28	**72.33%** ±1.20	43.75% ±0.64	47.01% ±0.87

Table 1: **Native sequence recovery for multi-state gRNAde vs. single-state baseline**. Using multiple conformations as input to gRNAde consistently improves sequence recovery, especially for multi-state and structurally diverse design settings (Average RMSD split and #Structures split). For the Sequence Identity split of biologically dissimilar RNAs, we see increased sequence recovery for gRNAde with $k = 3$ even though this setting does not explicitly evaluate multi-state design.

Structure	Accuracy	Recovery@100	Most Probable Sequence	Sequence Logo Plot@$t =0.2$
8D29_1_C	63.64%	33.18%	GCUAGUGGUGCGAAACGCGCAAAGGGUAGCGUC	
8D29_1_R	60.61%	28.21%	GCUAUUCGAUUGAAACGCGCAAGUGGUUACGUA	
8D29_1_F	69.70%	48.85%	GCGCAUCCGGCGAAACACGCUAAUGGUAGCGCA	
8D29_1_J	57.58%	41.67%	GGUGAGUGGCCGAAACACGCUAAUGGUAGCGCC	
8DK7_1_C	81.82%	57.18%	GCUAUUCCGGAGAAACACGCCGAUGGCAGCGUC	
8DK7_1_F	78.79%	47.27%	GGGAUUCCAGAGAGACACGCCCAUGGCAGUGCC	
Ensemble	90.91%	67.55%	GCGAUGCGAGCGAAACACGCUCUUGGCAGCGUC	

Table 2: **Case study for theophylline aptamer** (GCGAUACCAGCGAAACACGCCCUUGGCAGCGUC). We consider a multi-state design scenario for a new theophylline aptamer using multiple backbones from a known aptamer. Results for the first 6 rows are for the single-state baseline. Ensemble results in row 7 are for gRNAde trained with multiple conformations (here, $k =5$). Key takeaways include: (1) Recovery and accuracy for the single-state baseline can fluctuate significantly depending on the choice of conformer, suggesting that **single-state models may struggle to assign geometrically dissimilar motifs to the same sequence.** gRNAde alleviates this by directly 'baking in' the multi-state nature of RNA as an inductive bias in the architecture. (2) Each structure gives a different set of sampled sequences for single-state models – **it is not obvious how to select the input structure as well as designed sequence for single-state models**. gRNAde is explictly designed to predict consistent sequences while accounting for multiple states, leading to improved recovery and accuracy.

References

Jennifer A Doudna and Emmanuelle Charpentier. The new frontier of genome engineering with crispr-cas9. *Science*, 2014.

Norbert Pardi, Michael J Hogan, Frederick W Porter, and Drew Weissman. mrna vaccines—a new era in vaccinology. *Nature reviews Drug discovery*, 2018.

Peixuan Guo. The emerging field of rna nanotechnology. *Nature nanotechnology*, 2010.

Yiran Zhu, Liyuan Zhu, Xian Wang, and Hongchuan Jin. Rna-based therapeutics: An overview and prospectus. *Cell Death & Disease*, 2022.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint*, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, et al. Robust deep learning based protein sequence design using protein-mpnn. *Science*, 2022.

Pablo Gainza, Sarah Wehrle, Alexandra Van Hall-Beauvais, Anthony Marchand, Andreas Scheck, Zander Harteveld, Stephen Buckley, Dongchun Ni, Shuguang Tan, Freyr Sverrisson, et al. De novo design of protein interactions with learned surface fingerprints. *Nature*, 2023.

Rhiju Das. Rna structure: a renaissance begins? *Nature Methods*, 2021.

Raphael JL Townshend, Stephan Eismann, Andrew M Watkins, Ramya Rangan, Maria Karelina, Rhiju Das, and Ron O Dror. Geometric deep learning of rna structure. *Science*, 2021.

Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. Design of rnas: comparing programs for inverse rna folding. *Briefings in bioinformatics*, 2018.

Joseph D Yesselman, Daniel Eiler, Erik D Carlson, Michael R Gotrik, Anne E d'Aquino, Alexandra N Ooms, Wipapat Kladwang, Paul D Carlson, Xuesong Shi, David A Costantino, et al. Computational design of three-dimensional rna structure and function. *Nature nanotechnology*, 2019.

Laura R Ganser, Megan L Kelly, Daniel Herschlag, and Hashim M Al-Hashimi. The roles of structural dynamics in the cellular functions of rnas. *Nature reviews Molecular cell biology*, 2019.

Janis Hoetzel and Beatrix Suess. Structural changes in aptamers are essential for synthetic riboswitch engineering. *Journal of Molecular Biology*, 2022.

Megan L Ken, Rohit Roy, Ainan Geng, Laura R Ganser, Akanksha Manghrani, Bryan R Cullen, Ursula Schulze-Gahmen, Daniel Herschlag, and Hashim M Al-Hashimi. Rna conformational propensities determine cellular activity. *Nature*, 2023.

Wayne K Dawson, Maciej Maciejczyk, Elzbieta J Jankowska, and Janusz M Bujnicki. Coarse-grained modeling of rna 3d structure. *Methods*, 2016.

Leven M Wadley, Kevin S Keating, Carlos M Duarte, and Anna Marie Pyle. Evaluating and learning from rna pseudotorsional space: quantitative validation of a reduced representation for rna structure. *Journal of molecular biology*, 2007.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *NeurIPS*, 2019.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Chaitanya K. Joshi, Cristian Bodnar, Simon V. Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning*, 2023.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *NeurIPS*, 2017.

Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. Rna-solo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 2022.

Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 2012.

Alexandra Wrist, Wanqi Sun, and Ryan M Summers. The theophylline aptamer: 25 years as an important tool in cellular engineering research. *ACS Synthetic Biology*, 2020.

Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. Universal approximation of functions on sets. *Journal of Machine Learning Research*, 2022.

Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International conference on machine learning*, 2020.

Kalli Kappel, Kaiming Zhang, Zhaoming Su, Andrew M Watkins, Wipapat Kladwang, Shanshan Li, Grigore Pintilie, Ved V Topkar, Ramya Rangan, Ivan N Zheludev, et al. Accelerated cryo-em-guided determination of three-dimensional rna-only structures. *Nature methods*, 2020.

Steve L Bonilla and Jeffrey S Kieft. The promise of cryo-em to explore rna structural dynamics. *Journal of Molecular Biology*, 2022.

Appendix

Figure 2: Dataset statistics.

Figure 3: Multi-graph tensor representation of RNA conformational ensembles.

Listing 1: Pseudocode for multi-GNN encoder layer.
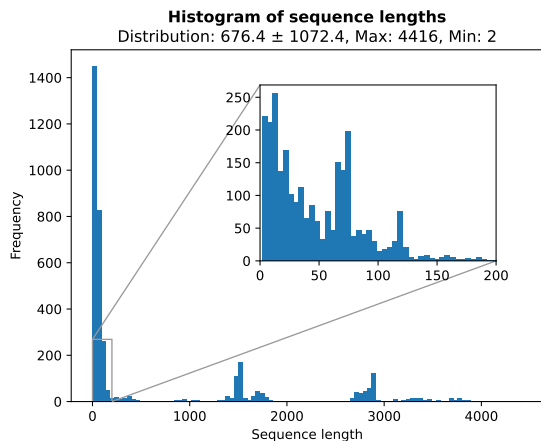
(a) **Sequence length.** The dataset is long-tailed in terms of RNA sequence length, with many short sequences including aptamers, miRNAs, siRNAs and tRNAs (fewer than 200 nucleotides). The dataset also includes several medium length ribosomal RNAs, snRNAs and snoRNAs (thousands of nucleotides).
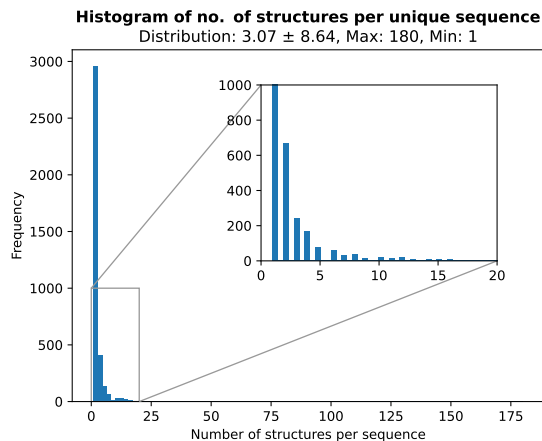
(b) **Number of structures per sequence.** The dataset covers a wide range of RNA conformation ensembles, with on average 3 structures per sequence. There are multiple structures available for 1,477 sequences. The remaining 2,287 sequences have one corresponding structure.
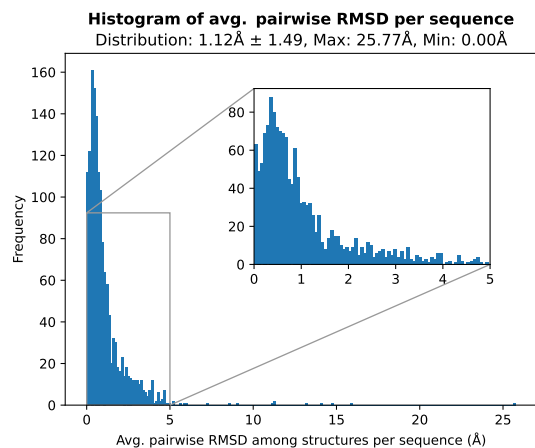
(c) **Average pairwise RMSD per sequence.** For 1,477 sequences with multiple structures, there is significant structural diversity among conformations. On average, the pairwise RMSD among the set of structures for a sequence is greater than 1Å.
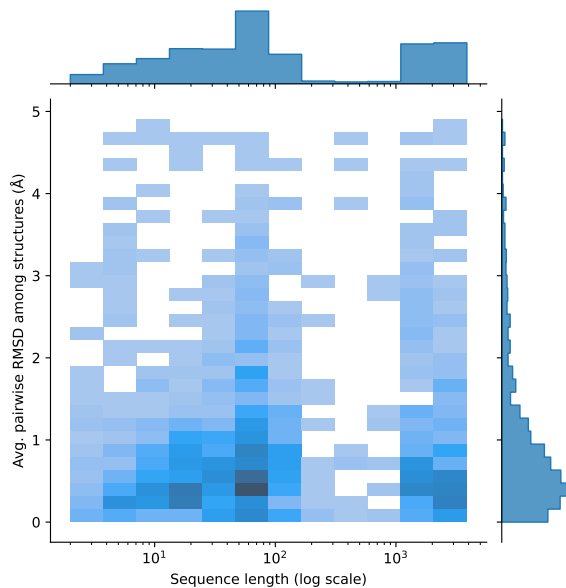
(d) **Bivariate distribution for sequence length vs. avg. RMSD.** The joint plot illustrates how structural diversity (measured by avg. pairwise RMSD) varies across sequence lengths. We notice similar structural variations regardless of sequence length.

Figure 2: **Dataset statistics.** We plot histograms to visualise the dataset diversity in terms of (a) sequence length, (b) number of structures available per sequence, as well as (c) structural variation among conformations for those RNA that have multiple structures. The bivariate distribution plot (d) for sequence length vs. average pairwise RMSD illustrates structural diversity regardless of sequence lengths.
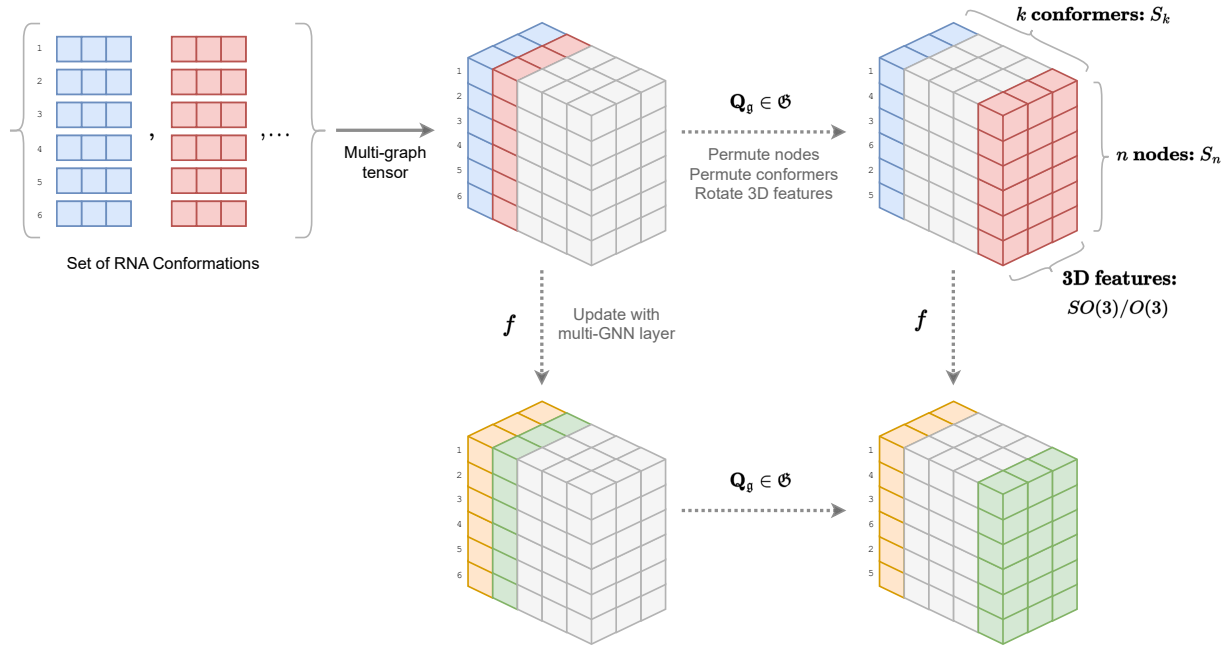
Figure 3: **Multi-graph tensor representation of RNA conformational ensembles, and the associated symmetry groups acting on each axis.** We process a set of k RNA conformations with n nodes each into a tensor representation. Each multi-GNN layer updates the tensor while being equivariant to the underlying symmetries; pseudocode is available in Listing 1. Here, we show a tensor of 3D vector-type features with shape $n \times k \times 3$. As depicted in the equivariance diagram, the updated tensor must be equivariant to permutation S_n of n nodes for axis 1, permutation S_k of k conformers for axis 2, and rotation $SO(3)/O(3)$ of the 3D features for axis 3.

```
1  class MultiGVPConv(MessagePassing):
2      '''GVPConv for handling multiple conformations'''
3
4      def __init__(self, ...):
5          ...
6
7      def forward(self, x_s, x_v, edge_index, edge_attr):
8
9          # stack scalar feats along axis 1:
10         # [n_nodes, n_conf, d_s] -> [n_nodes, n_conf * d_s]
11         x_s = x_s.contiguous().view(x_s.shape[0], x_s.shape[1] * x_s.shape[2])
12
13         # stack vector feat along axis 1:
14         # [n_nodes, n_conf, d_v, 3] -> [n_nodes, n_conf * d_v*3]
15         x_v = x_v.contiguous().view(x_v.shape[0], x_v.shape[1] * x_v.shape[2]*3)
16
17         # message passing and aggregation
18         message = self.propagate(edge_index, s=x_s, v=x_v, edge_attr=edge_attr)
19
20         # split scalar and vector channels
21         return _split_multi(message, d_s, d_v, n_conf)
22
23     def message(self, s_i, v_i, s_j, v_j, edge_attr):
24
25         # unstack scalar feats:
26         # [n_nodes, n_conf * d] -> [n_nodes, n_conf, d_s]
27         s_i = s_i.view(s_i.shape[0], s_i.shape[1]//d_s, d_s)
28         s_j = s_j.view(s_j.shape[0], s_j.shape[1]//d_s, d_s)
29
30         # unstack vector feats:
31         # [n_nodes, n_conf * d_v*3] -> [n_nodes, n_conf, d_v, 3]
32         v_i = v_i.view(v_i.shape[0], v_i.shape[1]//(d_v*3), d_v, 3)
33         v_j = v_j.view(v_j.shape[0], v_j.shape[1]//(d_v*3), d_v, 3)
34
35         # message function for edge j-i
36         message = tuple_cat((s_j, v_j), edge_attr, (s_i, v_i))
37         message = self.message_func(message)   # GVP
38
39         # merge scalar and vector channels along axis 1
40         return _merge_multi(*message)
41
42  def _split_multi(x, d_s, d_v, n_conf):
43      '''
44      Splits a merged representation of (s, v) back into a tuple.
45      '''
46      s = x[..., :-3 * d_v * n_conf].contiguous().view(x.shape[0], n_conf, d_s)
47      v = x[..., -3 * d_v * n_conf:].contiguous().view(x.shape[0], n_conf, d_v, 3)
48      return s, v
49
50  def _merge_multi(s, v):
51      '''
52      Merges a tuple (s, v) into a single `torch.Tensor`, where the vector
53      channels are flattened and appended to the scalar channels.
54      '''
55      # s: [n_nodes, n_conf, d] -> [n_nodes, n_conf * d_s]
56      s = s.contiguous().view(s.shape[0], s.shape[1] * s.shape[2])
57      # v: [n_nodes, n_conf, d, 3] -> [n_nodes, n_conf * d_v*3]
58      v = v.contiguous().view(v.shape[0], v.shape[1] * v.shape[2]*3)
59      return torch.cat([s, v], -1)
```

Listing 1: **PyTorch Geometric-style pseudocode for a multi-GNN encoder layer.** We update node features for each conformer independently while maintaining permutation equivariance of the updated feature tensors along both the first (number of nodes) and second (number of conformations) axes.