gRNAde: Geometric Deep Learning for 3D RNA inverse design

Anonymous Author(s)
Affiliation
Address
email

Abstract

Computational RNA design tasks are often posed as inverse problems, where sequences are designed based on adopting a single desired secondary structure without considering 3D geometry and conformational diversity. We introduce **gRNAde**, a **g**eometric **RNA de**sign pipeline operating on 3D RNA backbones to design sequences that explicitly account for structure and dynamics. Under the hood, gRNAde is a multi-state Graph Neural Network that generates candidate RNA sequences conditioned on one or more 3D backbone structures where the identities of the bases are unknown. On a single-state fixed backbone re-design benchmark of 14 RNA structures from the PDB identified by Das et al. [2010], gRNAde obtains higher native sequence recovery rates (56% on average) compared to Rosetta (45% on average), taking under a second to produce designs compared to the reported hours for Rosetta. We further demonstrate the utility of gRNAde on a new benchmark of multi-state design for structurally flexible RNAs, as well as zero-shot ranking of mutational fitness landscapes in a retrospective analysis of a recent RNA polymerase ribozyme structure. Open source code and tutorials are available at: `anonymous.4open.science/r/geometric-rna-design`

1 Introduction

Why RNA design? Historical efforts in computational drug discovery have focussed on designing small molecule or protein-based medicines that either treat symptoms or counter the end stages of disease processes. In recent years, there is a growing interest in designing new RNA-based therapeutics that intervene earlier in disease processes to cut off disease-causing information flow in the cell [Damase et al., 2021, Zhu et al., 2022]. Notable examples of RNA molecules at the forefront of biotechnology today include mRNA vaccines [Metkar et al., 2024] and CRISPR-based genomic medicine [Doudna and Charpentier, 2014]. Of particular interest for structure-based design are ribozymes and riboswitches in the untranslated regions of mRNAs [Mandal and Breaker, 2004, Leppek et al., 2018]. In addition to coding for proteins (such as the spike protein in the Covid vaccine), naturally occurring mRNAs contain riboswitches that are responsible for cell-state dependent protein expression of the mRNA. Riboswitches act by 'switching' their 3D structure from an unbound conformation to a bound one in the presence of specific metabolites or small molecules. Rational design of riboswitches will enable translation to be dependent on the presence or absence of partner molecules, essentially acting as 'on-off' switches for highly targeted mRNA therapies in the future [Felletti et al., 2016, Mustafina et al., 2019, Mohsen et al., 2023].

Challenges of RNA modelling. Despite the promises of RNA therapeutics, proteins have instead been the primary focus in the 3D biomolecular modelling community. Availability of a large number of protein structures from the PDB combined with advances in deep learning for structured data [Bronstein et al., 2021, Duval et al., 2023] have revolutionized protein 3D structure prediction [Jumper

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

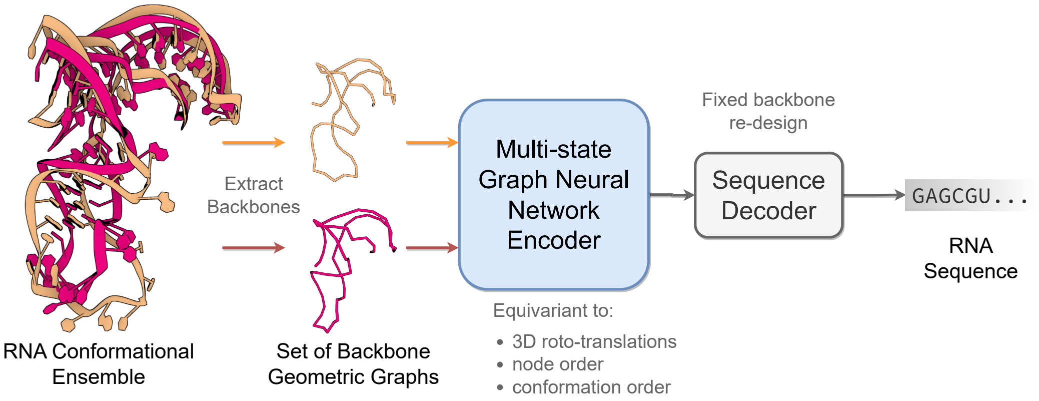


Figure 1: **The gRNAde pipeline for 3D RNA inverse design.** gRNAde is a generative model for RNA sequence design conditioned on backbone 3D structure(s). gRNAde processes one or more RNA backbone graphs (a conformational ensemble) via a multi-state GNN encoder which is equivariant to 3D roto-translation of coordinates as well as conformer order, followed by conformer order-invariant pooling and autoregressive sequence decoding.

et al., 2021] and rational design [Dauparas et al., 2022, Watson et al., 2023]. Applications of deep learning for computational RNA design are underexplored compared to proteins due to paucity of 3D structural data [Schneider et al., 2023]. Most tools for RNA design primarily focus on secondary structure without considering 3D geometry [Churkin et al., 2018] and use non-learnt algorithms for aligning 3D RNA fragments [Han et al., 2017, Yesselman et al., 2019], which can be restrictive due to the hand-crafted nature of the heuristics used.

In addition to limited 3D data for training deep learning models, 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 both the 3D geometric structure and conformational flexibility of RNA to engineer new biological functions.

Our contributions. This paper introduces **gRNAde**, a **g**eometric deep learning-based pipeline for **RNA** inverse **de**sign conditioned on 3D structure, analogous to ProteinMPNN for proteins [Dauparas et al., 2022]. As illustrated in Figure 1, gRNAde generates candidate RNA sequences conditioned on one or more backbone 3D conformations, enabling both single- and multi-state fixed-backbone sequence design.

We demonstrate the utility of gRNAde for the following design scenarios:

- **Improved performance and speed over Rosetta.** We compare gRNAde to Rosetta [Leman et al., 2020], the state-of-the-art physically based tool for 3D RNA inverse design, for single-state fixed backbone design of 14 RNA structures of interest from the PDB identified by Das et al. [2010]. We obtain higher native sequence recovery rates with gRNAde (56% on average) compared to Rosetta (45% on average). Additionally, gRNAde is significantly faster than Rosetta for inference; e.g. sampling 100+ designs in 1 second for an RNA of 60 nucleotides on an A100 GPU, compared to the reported hours for Rosetta.
- **Enables multi-state RNA design**, which was previously not possible with Rosetta. gRNAde with multi-state GNNs improves sequence recovery over an equivalent single-state model on a benchmark of structurally flexible RNAs, especially for surface nucleotides which undergo positional or secondary structural changes.
- **Zero-shot learning of RNA fitness landscape.** In a retrospective analysis of mutational fitness landscape data for an RNA polymerase ribozyme [McRae et al., 2024], we show how gRNAde's perplexity, the likelihood of a sequence folding into a backbone structure, can be used to rank mutants based on fitness in a zero-shot/unsupervised manner and outperforms random mutagenesis for improving fitness over the wild type in low throughput scenarios.

2 The gRNAde pipeline

2.1 The 3D RNA inverse folding problem

Figure 1 illustrates the RNA inverse folding problem: the task of designing new RNA sequences conditioned on a structural backbone. Given the 3D coordinates of a backbone structure, machine learning models must generate sequences that are likely to fold into that shape. The underlying assumption behind inverse folding (and rational biomolecule design) is that structure determines function [Huang et al., 2016]. To the best of our knowledge, gRNAde is the first explicitly multi-state inverse folding pipeline, allowing users to design sequences for backbone conformational ensembles (a set of 3D backbone structures) as opposed to a single structure.

2.2 RNA conformational ensembles as geometric multi-graphs

Featurization. The input to gRNAde is an RNA to be re-designed. For instance, this could be a set of PDB files with 3D backbone structures for the given RNA (a conformational ensemble) and the corresponding sequence of n nucleotides. As shown in Appendix Figure 11, gRNAde builds a geometric graph representation for each input structure:

1. We start with 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]. This 'pseudotorsional' representation describes RNA backbones completely in most cases while reducing the size of the torsional space to prevent overfitting [Wadley et al., 2007].
2. Each nucleotide i is assigned a node in the geometric graph with the 3D coordinate $\vec{\boldsymbol{x}}_i \in \mathbb{R}^3$ corresponding to the centroid of the 3 bead atoms. Random Gaussian noise with standard deviation 0.1Å is added to coordinates during training to prevent overfitting on crystallisation artifacts, following Dauparas et al. [2022]. Each node is connected by edges to its 32 nearest neighbours as measured by the pairwise distance in 3D space, $\|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|_2$.
3. Nodes are initialized with geometric features analogous to the featurization used in protein inverse folding [Ingraham et al., 2019, Jing et al., 2020]: (a) forward and reverse unit vectors along the backbone from the 5' end to the 3' end, ($\vec{\boldsymbol{x}}_{i+1} - \vec{\boldsymbol{x}}_i$ and $\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_{i-1}$); and (b) unit vectors, distances, angles, and torsions from each C4' to the corresponding P and N1/N9.
4. Edge features for each edge from node j to i are initialized as: (a) the unit vector from the source to destination node, $\vec{\boldsymbol{x}}_j - \vec{\boldsymbol{x}}_i$; (b) the distance in 3D space, $\|\vec{\boldsymbol{x}}_j - \vec{\boldsymbol{x}}_i\|_2$, encoded by 32 radial basis functions; and (c) the distance along the backbone, $j - i$, encoded by 32 sinusoidal positional encodings.

Multi-graph representation. As described in the previous section, given a set of k (conformer) structures in the input conformational ensemble, each RNA backbone is featurized as a separate geometric graph $\mathcal{G}^{(k)} = (\boldsymbol{A}^{(k)}, \boldsymbol{S}^{(k)}, \vec{\boldsymbol{V}}^{(k)})$ 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 input to gRNAde is thus a set of geometric graphs $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(k)}\}$ which is merged into what we term a 'multi-graph' representation of the conformational ensemble, $\mathcal{M} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{V}})$, by stacking the set of scalar features $\{\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(k)}\}$ 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 $\{\vec{\boldsymbol{V}}^{(1)}, \ldots, \vec{\boldsymbol{V}}^{(k)}\}$ is stacked into one tensor $\vec{\boldsymbol{V}} \in \mathbb{R}^{n \times k \times f' \times 3}$. Lastly, the set of adjacency matrices $\{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(k)}\}$ are merged via a union $\cup$ into one single joint adjacency matrix $\boldsymbol{A}$.

2.3 Multi-state GNN for representation learning on conformational ensembles

The gRNAde model, illustrated in Appendix Figure 12, processes one or more RNA backbone graphs via a multi-state GNN encoder which is equivariant to 3D roto-translation of coordinates as well as to the ordering of conformers, followed by conformer order-invariant pooling and sequence decoding. We describe each component in the following sections.

Multi-state GNN encoder. When representing conformational ensembles as a multi-graph, each node feature tensor contains three axes: (#nodes, #conformations, feature channels). We perform message passing on the multi-graph adjacency to *independently* process each conformer, while maintaining permutation equivariance of the updated feature tensors along both the first (#nodes) and second (#conformations) axes. This works by operating on only the feature channels axis and generalising the PyTorch Geometric [Fey and Lenssen, 2019] message passing class to account for the extra conformations axis; see Appendix Figure 14 and the pseudocode for details.

We use multiple rotation-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:

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

$$\boldsymbol{s}'_i, \vec{\boldsymbol{v}}'_i := \textsc{Upd}\big((\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i) \ , \ (\boldsymbol{m}_i, \vec{\boldsymbol{m}}_i) \big), \tag{2}$$

where $\textsc{Msg}, \textsc{Upd}$ are Geometric Vector Perceptrons, a generalization of MLPs to take tuples of scalar and vector features as input and apply $O(3)$-equivariant non-linear updates. The overall GNN encoder is $SO(3)$-equivariant due to the use of reflection-sensitive input features (dihedral angles) combined with $O(3)$-equivariant GVP-GNN layers.

Our multi-state GNN encoder is easy to implement in any message passing framework 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 $\vec{\boldsymbol{V}}' \in \mathbb{R}^{n \times f' \times 3}$:

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

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, enabling both single-state and multi-state design with the same model.

Sequence decoding and loss function. We feed the final encoder representations after pooling, $\boldsymbol{S}', \vec{\boldsymbol{V}}'$, to autoregressive GVP-GNN decoder layers to predict the probability of the four possible base identities (A, G, C, U) for each node/nucleotide. Decoding proceeds according to the RNA sequence order from the 5' end to 3' end. gRNAde is trained in a self-supervised manner by minimising a cross-entropy loss (with label smoothing value of 0.05) between the predicted probability distribution and the ground truth identity for each base. During training, we use autoregressive teacher forcing [Williams and Zipser, 1989] where the ground truth base identity is fed as input to the decoder at each step, encouraging the model to stay close to the ground-truth sequence.

Sampling. When using gRNAde for inference and designing new sequences, we iteratively sample the base identity for a given nucleotide from the predicted conditional probability distribution, given the partially designed sequence up until that nucleotide/decoding step. We can modulate the smoothness or sharpness of the probability distribution by using a temperature parameter.

2.4 Evaluation metrics for designed sequences

In principle, inverse folding models can be sampled from to obtain a large number of designed sequences for a given backbone structure. Thus, in-silico metrics to determine which sequences are useful and which ones to prioritise in wet lab experiments are a critical part of the overall pipeline. We currently use the following metrics to evaluate gRNAde's designs, visualised in Appendix Figure 13:

- **Native sequence recovery**, which is the average percentage of native (ground truth) nucleotides correctly recovered in the sampled sequences. Recovery is the most widely used metric for biomolecule inverse design [Dauparas et al., 2022] but can be misleading in the case of RNAs where alternative nucleotide base pairings can form the same structural patterns.
- **Secondary structure self-consistency score**, where we 'forward fold' the sampled sequences using a secondary structure prediction tool (we used EternaFold [Wayment-Steele et al., 2022])

and measure the average Matthew's Correlation Coefficient (MCC) to the groundtruth secondary structure, represented as a binary adjacency matrix. MCC values range between -1 and +1, where +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. This measures how well the designs recover base pairing patterns.

- **Tertiary structure self-consistency scores**, where we 'forward fold' the sampled sequences using a 3D structure prediction tool (we used RhoFold [Shen et al., 2022]) and compute the average RMSD, TM-score and GDT_TS to the groundtruth C4' coordinates to measure how well the designs recover global structural similarity and 3D conformations.
- **Perplexity**, which can be thought of as the average number of bases that the model is selecting from for each nucleotide. Formally, perplexity is the average exponential of the negative log-likelihood of the sampled sequences. A perfect model would have perplexity of 1, while a perplexity of 4 means that the model is making random predictions (the model outputs a uniform probability over 4 possible bases). Perplexity does not require a ground truth structure to calculate, and can also be used for ranking sequences as it is the model's estimate of the compatibility of a sequence with the input backbone structure.

Significance and limitations. Self-consistency metrics, termed 'designability' (eg. scRMSD$\leq$2Å), as well as perplexity have been found to correlate with experimental success in protein design [Watson et al., 2023]. While precise designability thresholds are yet to be established for RNA, pairs of structures with TM-score$\geq$0.45 or GDT_TS$\geq$0.5 are known to correspond to roughly the same fold [Zhang et al., 2022]. Another major limitation for in-silico evaluation of 3D RNA design compared to proteins is the relatively worse state of structure prediction tools [Schneider et al., 2023].

3 Experimental Setup

3D RNA structure dataset. We create a machine learning-ready dataset for RNA inverse design using RNASolo [Adamczyk et al., 2022], a novel repository of RNA 3D structures extracted from solo RNAs, protein-RNA complexes, and DNA-RNA hybrids in the PDB. We used structures at resolution $\leq$4.0Å resulting in 4,223 unique RNA sequences for which a total of 12,011 structures are available (RNASolo date cutoff: 31 October 2023). Dataset statistics are available in Appendix Figure 15, 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.

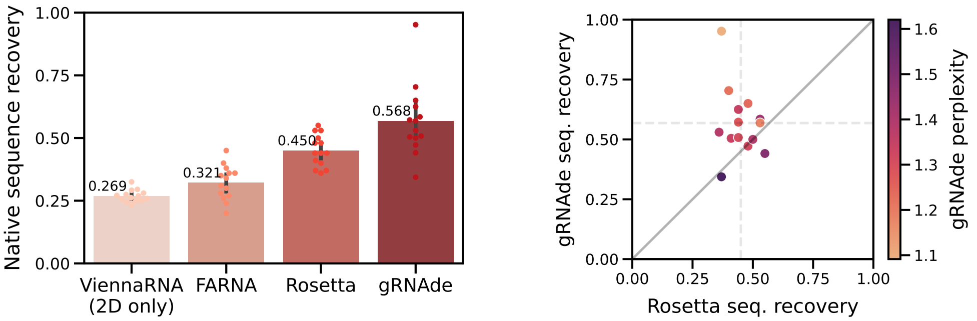
Structural clustering. In order to ensure that we evaluate gRNAde's generalization ability to novel RNAs, we cluster the 4,223 unique RNAs into groups based on structural similarity. We use US-align [Zhang et al., 2022] with a similarity threshold of TM-score >0.45 for clustering, and ensure that we train, validate and test gRNAde on structurally dissimilar clusters (see next paragraph). We also provide utilities for clustering based on sequence homology using CD-HIT [Fu et al., 2012], which leads to splits containing biologically dissimilar clusters of RNAs.

Splits to evaluate generalization. After clustering, we split the RNAs into training ($\sim$4000 samples), validation and test sets (100 samples each) to evaluate two different design scenarios:

1. **Single-state split.** This split is used to fairly evaluate gRNAde for single-state design on a set of RNA structures of interest from the PDB identified by Das et al. [2010], which mainly includes riboswitches, aptamers, and ribozymes. We identify the structural clusters belonging to the RNAs identified in Das et al. [2010] and add all the RNAs in these clusters to the test set (100 samples). The remaining clusters are randomly added to the training and validation splits.
2. **Multi-state split.** This split is used to test gRNAde's ability to design RNA with multiple distinct conformational states. We order the structural clusters based on median intra-sequence RMSD among available structures within the cluster[1]. The top 100 samples from clusters with the highest median intra-sequence RMSD are added to the test set. The next 100 samples are added to the validation set and all remaining samples are used for training.

Validation and test samples come from clusters with at most 5 unique sequences, in order to ensure diversity. Any samples that were not assigned clusters are directly appended to the training set. We

[1] For each RNA sequence, we compute the pairwise C4' RMSD among all available structures. We then compute the median RMSD across all sequences within each structural cluster.



(a) gRNAde outperforms Rosetta.

(b) gRNAde's perplexity correlates with recovery.

Figure 2: **gRNAde compared to Rosetta for single-state design.** (a) We benchmark native sequence recovery of gRNAde, Rosetta, FARNA and ViennaRNA on 14 RNA structures of interest identified by Das et al. [2010]. gRNAde obtains higher native sequence recovery rates (56% on average) compared to Rosetta (45%). (b) Sequence recovery per sample for Rosetta and gRNAde, shaded by gRNAde's perplexity for each sample. gRNAde's perplexity is correlated with native sequence recovery for designed sequences. Full results are available in Appendix Table 2.

also directly add very large RNAs (> 1000 nts) to the training set, as it is unlikely that we want to design very large RNAs. We exclude very short RNA strands (< 10 nts).

Evaluation metrics. For a given data split, we evaluate models on the held-out test set by designing 16 sequences (sampled at temperature 0.1) for each test data point and computing averages for each of the metrics described in Section 2.4: native sequence recovery, structural self-consistency scores and perplexity. We employ early stopping by reporting test set performance for the model checkpoint for the epoch with the best validation set recovery. Standard deviations are reported across 3 consistent random seeds for all models.

Hyperparameters. All models use 4 encoder and 4 decoder GVP-GNN layers, with 128 scalar/16 vector node features, 64 scalar/4 vector edge features, and drop out probability 0.5, resulting in 2,147,944 trainable parameters. All models are trained for a maximum of 50 epochs using the Adam optimiser with an initial learning rate of 0.0001, which is reduced by a factor 0.9 when validation performance plateaus with patience of 5 epochs. Ablation studies of key modelling decisions are available in Appendix Table 1.

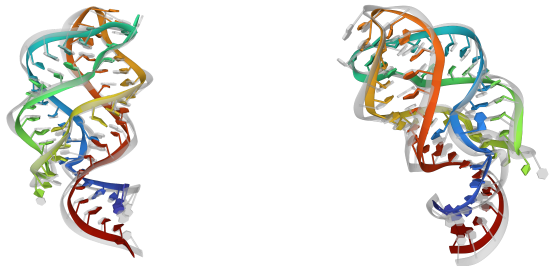
4 Results

4.1 Single-state RNA design benchmark

We set out to compare gRNAde to Rosetta, a state-of-the-art physically based toolkit for biomolecular modelling and design [Leman et al., 2020]. We reproduced the benchmark setup from Das et al. [2010] for Rosetta's fixed backbone RNA sequence design workflow on 14 RNA structures of interest from the PDB, which mainly includes riboswitches, aptamers, and ribozymes (full listing in Table 2). We trained gRNAde on the single-state split detailed in Section 3, explicitly excluding the 14 RNAs as well as any structurally similar RNAs in order to ensure that we fairly evaluate gRNAde's generalization abilities vs. Rosetta.

gRNAde improves sequence recovery over Rosetta. In Figure 2, we compare gRNAde's native sequence recovery for single-state design with numbers taken from Das et al. [2010] for Rosetta, FARNA (a predecessor of Rosetta), and ViennaRNA (the most popular 2D inverse folding method). gRNAde has higher recovery of 56% on average compared to 45% for Rosetta, 32% for FARNA, and 27% for ViennaRNA. See Appendix Table 2 for per-RNA results.

gRNAde is significantly faster than Rosetta. In addition to superior sequence recovery, gRNAde is significantly faster than Rosetta for high-throughout design pipelines. Training gRNAde from scratch takes roughly 2–6 hours on a single A100 GPU, depending on the exact hyperparameters. Once trained, gRNAde can design hundreds of sequences for backbones with hundreds of nucleotides



Design 1:
perplexity: 1.310
recovery: 0.591 (27 edits)
sc2D = 0.923, scRMSD = 1.384
scTM = 0.831, scGDT = 0.830

Design 2:
perplexity: 1.382
recovery: 0.409 (37 edits)
sc2D = 0.922, scRMSD = 2.125
scTM = 0.687, scGDT = 0.678

Design 3:
perplexity: 1.425
recovery: 0.515 (30 edits)
sc2D = 0.923, scRMSD = 3.213
scTM = 0.512, scGDT = 0.526

Figure 3: **Cherry-picked designs for Guanine riboswitch aptamer** (PDB: 4FE5). We show the RhoFold-predicted 3D structure in colour overlaid on the groundtruth structure in grey. Designs recover the base pairing patterns and tertiary structure of the RNA, as measured by high self-consistency score. gRNAde's perplexity is correlated well with 3D self-consistency scores and can be useful for ranking designs. More design visualisations are available in Appendix C.

in ∼1 second with GPU acceleration. On the other hand, Rosetta takes order of hours to produce a single design due to performing expensive Monte Carlo optimisations[2]. Deep learning methods like gRNAde are arguably easier to use since no expert customization is required and setup is easier compared to Rosetta [Dauparas et al., 2022], potentially making RNA design more broadly accessible.

gRNAde's perplexity correlates with sequence and structural recovery. In Figure 2b, we plot native sequence recovery per sample for Rosetta vs. gRNAde, shaded by gRNAde's average perplexity for each sample. Perplexity is an indicator of the model's confidence in its own prediction (lower perplexity implies higher confidence) and appears to be correlated with native sequence recovery. Additionally, visualisations of gRNAde's designs for a riboswitch in Figure 3 show that perplexity is also correlated with structural self-consistency scores. In the subsequent Section 4.3, we further demonstrate the utility of gRNAde's perplexity for zero-shot ranking of RNA fitness landscapes.

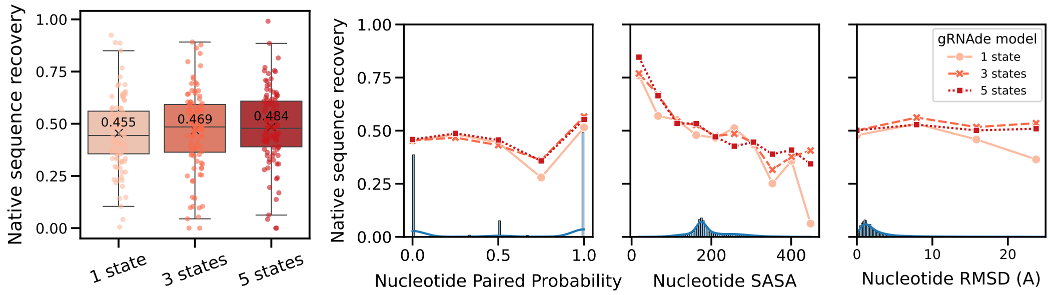
4.2 Multi-state RNA design benchmark

Structured RNAs often adopt multiple distinct conformational states to perform biological functions [Ken et al., 2023]. For instance, riboswitches adopt at least two distinct functional conformations: a ligand bound (holo) and unbound (apo) state, which helps them regulate and control gene expression [Stagno et al., 2017]. If we were to attempt single-state inverse design for such RNAs, each backbone structure may lead to a different set of sampled sequences. It is not obvious how to select the input backbone as well as designed sequence when using single-state models for multi-state design. gRNAde's multi-state GNN, descibed in Section 2.3, directly 'bakes in' the multi-state nature of RNA into the architecture and designs sequences explicitly conditioned on multiple states.

In order to evaluate gRNAde's multi-state design capabilities, we trained equivalent single-state and multi-state gRNAde models on the multi-state split detailed in Section 3, where the validation and test sets contain progressively more structurally flexible RNAs as measured by median RMSD among multiple available states for an RNA.

Multi-state gRNAde boosts sequence recovery. In Figure 4a, we compared a single-state variant of gRNAde with otherwise equivalent multi-state models (with up to 3 and 5 states, respectively) in terms of native sequence recovery. Multi-state variants show marginal improvements, overall. As a caveat, it is worth noting that multi-state models consume more GPU memory than an equivalent single-state model during mini-batch training (approximate peak GPU usage for max. number of states = 1: 12GB, 3: 28GB, 5: 50GB on a single A100 with at most 3000 total nodes in a mini-batch).

[2] While we have not run Rosetta ourselves, we note that its documentation states that "runs on RNA backbones longer than ∼ten nucleotides take many minutes or hours".



(a) Per-sample sequence recovery

(b) Per-nucleotide recovery vs. structural flexibility measures

Figure 4: **Multi-state design benchmark.** (a) Multi-state gRNAde show marginal improvement over an equivalent single-state model in terms of average per-sample sequence recovery over all test RNAs. (b) When plotting sequence recovery per-nucleotide, multi-state gRNAde improves over a single-state model for structurally flexible regions of RNAs, as characterised by nucleotides that tend to undergo changes in base pairing (left), nucleotides with greater average solvent accessible surface area (centre), and nucleotides with higher average RMSD (right) across multiple states. Marginal histograms in blue show the distribution of values. We plot performance for one consistent random seed across all models; collated results and ablations are available in Appendix Table 1.

Improved recovery in structurally flexible regions. In Figure 4b, we evaluated gRNAde's multi-state sequence recovery at a fine-grained, per-nucleotide level. Multi-state GNNs improve sequence recovery over the single-state variant on structurally flexible nucleotides, as characterised by undergoing changes in base pairing/secondary structure, higher average RMSD between 3D coordinates across states, and larger solvent accessible surface area.

4.3 Zero-shot ranking of RNA fitness landscape

Lastly, we explored the use of gRNAde as a zero-shot ranker of mutants in RNA engineering campaigns. Given the backbone structure of a wild type RNA of interest as well as a candidate set of mutant sequences, we can compute gRNAde's perplexity of whether a given sequence folds into the backbone structure. Perplexity is inversely related to the likelihood of a sequence conditioned on a structure, as described in Section 2.4. We can then rank sequences based on how 'compatible' they are with the backbone structure in order to select a subset to be experimentally validated in wet labs.

Retrospective analysis on ribozyme fitness landscape. A recent study by McRae et al. [2024] determined a cryo-EM structure of a dimeric RNA polymerase ribozyme at 5Å resolution[3], along with fitness landscapes of $\sim$75K mutants for the catalytic subunit 5TU and $\sim$48K mutants for the scaffolding subunit t1. We design a retrospective study using this data of (sequence, fitness value) pairs where we simulate an RNA engineering campaign with the aim of improving catalytic subunit fitness over the wild type 5TU sequence.

We consider various design budgets ranging from hundreds to thousands of sequences selected for experimental validation, and compare 4 unsupervised approaches for ranking/selecting variants: (1) random choice from all $\sim$75,000 sequences; (2) random choice from all 449 single mutant sequences; (3) random choice from all single and double mutant sequences (as sequences with higher mutation order tend to be less fit); and (4) negative gRNAde perplexity (lower perplexity is better). For each design budget and ranking approach, we compute the expected maximum change in fitness over the wild type that could be achieved by screening as many variants as allowed in the given design budget. We run 10,000 simulations to compute confidence intervals for the 3 random baselines.

gRNAde outperforms random baselines in low design budget scenarios. Figure 5 illustrates the results of our retrospective study. At low design budgets of up to hundreds of sequences, which are relevant in the case of a low throughput fitness screening assay, gRNAde outperforms all random baselines in terms of the maximum change in fitness over the wild type. The top 10 mutants as ranked by gRNAde contain a sequence with 4-fold improved fitness, while the top 200 leads to a 5-fold improvement. Note that gRNAde is used zero-shot here, i.e. it was not fine-tuned on any assay data.

[3] This RNA was not present in gRNAde's training data, which contains structures at $\leq$4.0Å resolution.

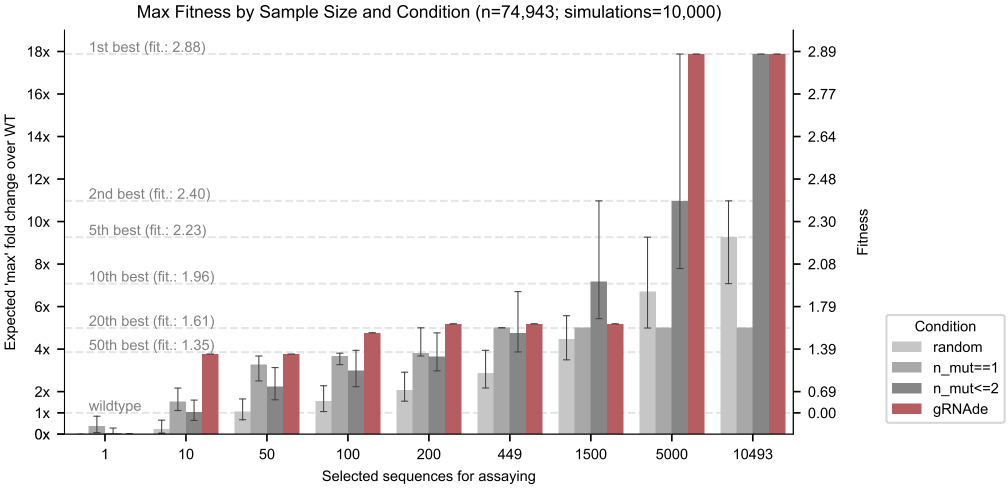


Figure 5: **Retrospective study of gRNAde for ranking ribozyme mutant fitness.** Using the backbone structure and mutational fitness landscape data from an RNA polymerase ribozyme [McRae et al., 2024], we retrospectively analyse how well we can rank variants at multiple design budgets using random selection vs. gRNAde's perplexity for mutant sequences conditioned on the backbone structure (catalytic subunit 5TU). Note that gRNAde is used zero-shot here, i.e. it was not fine-tuned on any assay data. For stochastic strategies, bars indicate median values, and error bars indicate the interquartile range estimated from 10,000 simulations per strategy and design budget. At low throughput design budgets of up to ∼500 sequences, selecting mutants using gRNAde outperforms random baselines in terms of the expected maximum improvement in fitness over the wild type. In particular, gRNAde performs better than single site saturation mutagenesis, even when all single mutants are explored (total of 449 single mutants, 10,493 double mutants for the catalytic subunit 5TU in McRae et al. [2024]). See Appendix Figure 10 for results on scaffolding subunit t1.

Perspective. Overall, it is promising that gRNAde's perplexity correlates with experimental fitness measurements out-of-the-box (zero-shot) and can be a useful ranker of mutant fitness in our retrospective study. In realistic design scenarios, improvements could likely be obtained by fine-tuning gRNAde on a low amount of experimental fitness data. For example, latent features from gRNAde may be finetuned or used as input to a prediction head with supervised learning on fitness landscape data. This study acts as a sanity check before committing to wet lab validation of gRNAde designs. We see random mutagenesis and directed evolution-based approaches as complementary to de-novo design and inverse folding approaches like gRNAde. Random mutagenesis can be thought of as local exploration around a wild type sequence, optimising fitness within an 'island' of activity. Structure-based design approaches are akin to global jumps in sequence space, with the potential to find new islands further away from the wild type [Huang et al., 2016].

5 Conclusion

We introduce gRNAde, a geometric deep learning pipeline for RNA sequence design conditioned on one or more 3D backbone structures. gRNAde is superior to the physically based Rosetta for 3D RNA inverse folding in terms of performance, inference speed, and ease of use. Further, gRNAde enables explicit multi-state design for structurally flexible RNAs which was previously not possible with Rosetta. gRNAde's perplexity correlates with native sequence and structural recovery, and can be used for zero-shot ranking of mutants in RNA engineering campaigns. To the best of our knowledge, gRNAde is also the first geometric deep learning architecture for multi-state biomolecule representation learning; the model is generic and can be repurposed for other learning tasks on conformational ensembles, including multi-state protein design.

Limitations. Key avenues for future development of gRNAde include supporting multiple interacting chains, accounting for partner molecules with RNAs, and supporting negative design against undesired conformations. We discuss practical tradeoffs to using gRNAde in real-world RNA design scenarios in Appendix B, including limitations due to the current state of 3D RNA structure prediction tools.

References

B. Adamczyk, M. Antczak, and M. Szachniuk. Rnasolo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 2022. (Cited on page 5)

M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 2021. (Cited on page 13)

M. Baek, R. McHugh, I. Anishchenko, H. Jiang, D. Baker, and F. DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature Methods*, 2024. (Cited on page 13)

E. Bonnet, P. Rzazewski, and F. Sikora. Designing rna secondary structures is hard. *Journal of Computational Biology*, 2020. (Cited on page 13)

M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint*, 2021. (Cited on page 1)

J. Chen, Z. Hu, S. Sun, Q. Tan, Y. Wang, Q. Yu, L. Zong, L. Hong, J. Xiao, T. Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint*, 2022. (Cited on page 13)

A. Churkin, M. D. Retwitzer, V. Reinharz, Y. Ponty, J. Waldispühl, and D. Barash. Design of rnas: comparing programs for inverse rna folding. *Briefings in bioinformatics*, 2018. (Cited on page 2, 13)

T. R. Damase, R. Sukhovershin, C. Boada, F. Taraballi, R. I. Pettigrew, and J. P. Cooke. The limitless future of rna therapeutics. *Frontiers in bioengineering and biotechnology*, 2021. (Cited on page 1)

R. Das, J. Karanicolas, and D. Baker. Atomic accuracy in predicting and designing noncanonical rna structure. *Nature methods*, 2010. (Cited on page 1, 2, 5, 6, 13, 14, 15, 19)

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. Wicky, et al. Robust deep learning based protein sequence design using proteinmpnn. *Science*, 2022. (Cited on page 2, 3, 4, 7, 14)

W. K. Dawson, M. Maciejczyk, E. J. Jankowska, and J. M. Bujnicki. Coarse-grained modeling of rna 3d structure. *Methods*, 2016. (Cited on page 3)

K. Didi, F. Vargas, S. Mathis, V. Dutordoir, E. Mathieu, U. J. Komorowska, and P. Lio. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design. In *NeurIPS 2023 Machine Learning for Structural Biology Workshop*, 2023. (Cited on page 13)

J. A. Doudna and E. Charpentier. The new frontier of genome engineering with crispr-cas9. *Science*, 2014. (Cited on page 1)

A. Duval, S. V. Mathis, C. K. Joshi, V. Schmidt, S. Miret, F. D. Malliaros, T. Cohen, P. Lio, Y. Bengio, and M. Bronstein. A hitchhiker's guide to geometric gnns for 3d atomic systems. *arXiv preprint*, 2023. (Cited on page 1)

M. Felletti, J. Stifel, L. A. Wurmthaler, S. Geiger, and J. S. Hartig. Twister ribozymes as highly versatile expression platforms for artificial riboswitches. *Nature communications*, 2016. (Cited on page 1)

M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *ICLR 2019 Representation Learning on Graphs and Manifolds Workshop*, 2019. (Cited on page 4)

L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 2012. (Cited on page 5)

L. R. Ganser, M. L. Kelly, D. Herschlag, and H. M. Al-Hashimi. The roles of structural dynamics in the cellular functions of rnas. *Nature reviews Molecular cell biology*, 2019. (Cited on page 2)

D. Han, X. Qi, C. Myhrvold, B. Wang, M. Dai, S. Jiang, M. Bates, Y. Liu, B. An, F. Zhang, et al. Single-stranded dna and rna origami. *Science*, 2017. (Cited on page 2, 13)

S. He, R. Huang, J. Townley, R. C. Kretsch, T. G. Karagianes, D. B. Cox, H. Blair, D. Penzar, V. Vyaltsev, E. Aristova, et al. Ribonanza: deep learning of rna structure through dual crowdsourcing. *bioRxiv*, 2024. (Cited on page 13)

J. Hoetzel and B. Suess. Structural changes in aptamers are essential for synthetic riboswitch engineering. *Journal of Molecular Biology*, 2022. (Cited on page 2)

P.-S. Huang, S. E. Boyken, and D. Baker. The coming of age of de novo protein design. *Nature*, 2016. (Cited on page 3, 9)

J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola. Generative models for graph-based protein design. *NeurIPS*, 2019. (Cited on page 3, 20)

J. B. Ingraham, M. Baranov, Z. Costello, K. W. Barber, W. Wang, A. Ismail, V. Frappier, D. M. Lord, C. Ng-Thow-Hing, E. R. Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 2023. (Cited on page 13)

B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020. (Cited on page 3, 4, 20)

C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen, and P. Lio. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning*, 2023. (Cited on page 17)

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021. (Cited on page 1, 13)

M. L. Ken, R. Roy, A. Geng, L. R. Ganser, A. Manghrani, B. R. Cullen, U. Schulze-Gahmen, D. Herschlag, and H. M. Al-Hashimi. Rna conformational propensities determine cellular activity. *Nature*, 2023. (Cited on page 2, 7)

J. K. Leman, B. D. Weitzner, S. M. Lewis, J. Adolf-Bryfogle, N. Alam, R. F. Alford, M. Aprahamian, D. Baker, K. A. Barlow, P. Barth, et al. Macromolecular modeling and design in rosetta: recent methods and frameworks. *Nature methods*, 2020. (Cited on page 2, 6)

K. Leppek, R. Das, and M. Barna. Functional 5' utr mrna structures in eukaryotic translation regulation and how to find them. *Nature reviews Molecular cell biology*, 2018. (Cited on page 1)

S. Li, S. Moayedpour, R. Li, M. Bailey, S. Riahi, L. Kogler-Anele, M. Miladi, J. Miner, D. Zheng, J. Wang, et al. Codonbert: Large language models for mrna design and optimization. *bioRxiv*, 2023a. (Cited on page 13)

Y. Li, C. Zhang, C. Feng, R. Pearce, P. Lydia Freddolino, and Y. Zhang. Integrating end-to-end learning with deep geometrical potentials for ab initio rna structure prediction. *Nature Communications*, 2023b. (Cited on page 13)

M. Mandal and R. R. Breaker. Gene regulation by riboswitches. *Nature reviews Molecular cell biology*, 2004. (Cited on page 1)

E. K. McRae, C. J. Wan, E. L. Kristoffersen, K. Hansen, E. Gianni, I. Gallego, J. F. Curran, J. Attwater, P. Holliger, and E. S. Andersen. Cryo-em structure and functional landscape of an rna polymerase ribozyme. *Proceedings of the National Academy of Sciences*, 2024. (Cited on page 2, 8, 9, 19)

M. Metkar, C. S. Pepin, and M. J. Moore. Tailor made: the art of therapeutic mrna design. *Nature Reviews Drug Discovery*, 2024. (Cited on page 1)

M. G. Mohsen, M. K. Midy, A. Balaji, and R. R. Breaker. Exploiting natural riboswitches for aptamer engineering and validation. *Nucleic Acids Research*, 2023. (Cited on page 1)

K. Mustafina, K. Fukunaga, and Y. Yokobayashi. Design of mammalian on-riboswitches based on tandemly fused aptamer and ribozyme. *ACS Synthetic Biology*, 2019. (Cited on page 1)

R. J. Penic, T. Vlasic, R. G. Huber, Y. Wan, and M. Sikic. Rinalmo: General-purpose rna language models can generalize well on structure prediction tasks. *arXiv preprint*, 2024. (Cited on page 13)

F. Runge, D. Stoll, S. Falkner, and F. Hutter. Learning to design RNA. In *ICLR*, 2019. (Cited on page 13)

B. Schneider, B. A. Sweeney, A. Bateman, J. Cerny, T. Zok, and M. Szachniuk. When will rna get its alphafold moment? *Nucleic Acids Research*, 2023. (Cited on page 2, 5)

T. Shen, Z. Hu, Z. Peng, J. Chen, P. Xiong, L. Hong, L. Zheng, Y. Wang, I. King, S. Wang, et al. E2efold-3d: End-to-end deep learning method for accurate de novo rna 3d structure prediction. *arXiv preprint*, 2022. (Cited on page 5)

J. Stagno, Y. Liu, Y. Bhandari, C. Conrad, S. Panja, M. Swain, L. Fan, G. Nelson, C. Li, D. Wendel, et al. Structures of riboswitch rna reaction states by mix-and-inject xfel serial crystallography. *Nature*, 2017. (Cited on page 7)

C. Tan, Y. Zhang, Z. Gao, H. Cao, and S. Z. Li. Hierarchical data-efficient representation learning for tertiary structure-based rna design. *arXiv preprint*, 2023. (Cited on page 13, 14)

R. J. Townshend, S. Eismann, A. M. Watkins, R. Rangan, M. Karelina, R. Das, and R. O. Dror. Geometric deep learning of rna structure. *Science*, 2021. (Cited on page 13)

Q. Vicens and J. S. Kieft. Thoughts on how to think (and talk) about rna structure. *Proceedings of the National Academy of Sciences*, 2022. (Cited on page 13, 17)

L. M. Wadley, K. S. Keating, C. M. Duarte, and A. M. Pyle. Evaluating and learning from rna pseudotorsional space: quantitative validation of a reduced representation for rna structure. *Journal of molecular biology*, 2007. (Cited on page 3)

W. Wang, C. Feng, R. Han, Z. Wang, L. Ye, Z. Du, H. Wei, F. Zhang, Z. Peng, and J. Yang. trrosettarna: automated prediction of rna 3d structure with transformer network. *Nature Communications*, 2023. (Cited on page 13)

M. Ward, E. Courtney, and E. Rivas. Fitness functions for rna structure design. *Nucleic Acids Research*, 2023. (Cited on page 13)

A. M. Watkins, R. Rangan, and R. Das. Farfar2: improved de novo rosetta prediction of complex global rna folds. *Structure*, 2020. (Cited on page 13)

J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 2023. (Cited on page 2, 5, 13)

H. K. Wayment-Steele, W. Kladwang, A. I. Strom, J. Lee, A. Treuille, A. Becka, E. Participants, and R. Das. Rna secondary structure packages evaluated and improved by high-throughput experiments. *Nature methods*, 2022. (Cited on page 4)

R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1989. (Cited on page 4)

J. D. Yesselman, D. Eiler, E. D. Carlson, M. R. Gotrik, A. E. d'Aquino, A. N. Ooms, W. Kladwang, P. D. Carlson, X. Shi, D. A. Costantino, et al. Computational design of three-dimensional rna structure and function. *Nature nanotechnology*, 2019. (Cited on page 2, 13, 14)

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. *NeurIPS*, 2017. (Cited on page 4, 20)

C. Zhang, M. Shine, A. M. Pyle, and Y. Zhang. Us-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature methods*, 2022. (Cited on page 5)

Y. Zhu, L. Zhu, X. Wang, and H. Jin. Rna-based therapeutics: An overview and prospectus. *Cell Death & Disease*, 2022. (Cited on page 1)

A Related Work

We attempt to briefly summarise recent developments in RNA structure modelling and design, with an emphasis on deep learning-based approaches.

RNA inverse folding. Most tools for RNA inverse folding focus on secondary structure without considering 3D geometry [Churkin et al., 2018, Runge et al., 2019] and approach the problem from the lens of energy optimisation [Ward et al., 2023]. Rosetta fixed backbone re-design [Das et al., 2010] is the only energy optimisation-based approach that accounts for 3D structure. Deep neural networks such as gRNAde can incorporate 3D structural constraints and are orders of magnitude faster than optimisation-based approaches; this is particularly attractive for high-throughput design pipelines as solving the inverse folding optimisation problem is NP hard [Bonnet et al., 2020].

RNA structure design. Inverse folding models for protein design have often been coupled with backbone generation models which design structural backbones conditioned on various design constraints [Watson et al., 2023, Ingraham et al., 2023, Didi et al., 2023]. Current approaches for RNA backbone design use classical (non-learnt) algorithms for aligning 3D RNA motifs [Han et al., 2017, Yesselman et al., 2019], which are small modular pieces of RNA that are believed to fold independently. Such algorithms may be restricted by the use of hand-crafted heuristics and we plan to explore data-driven generative models for RNA backbone design in future work.

RNA structure prediction. There have been several recent efforts to adapt protein folding architectures such as AlphaFold2 [Jumper et al., 2021] and RosettaFold [Baek et al., 2021] for RNA structure prediction [Li et al., 2023b, Wang et al., 2023, Baek et al., 2024]. A previous generation of models used GNNs as ranking functions together with Rosetta energy optimisation [Watkins et al., 2020, Townshend et al., 2021]. None of these architectures aim at capturing conformational flexibility of RNAs, unlike gRNAde which represents RNAs as multi-state conformational ensembles. Neither can structure prediction tools be used for RNA design tasks as they are not generative models.

RNA language models. Self-supervised language models have been developed for predictive and generative tasks on RNA sequences, including general-purpose models such as RNA FM [Chen et al., 2022] and RiNaLMo [Penic et al., 2024] as well as mRNA-specific CodonBERT [Li et al., 2023a]. RNA sequence data repositories are orders of magnitude larger than those for RNA structure (eg. RiNaLMo is trained on 36 million sequences). However, standard language models can only implicitly capture RNA structure and dynamics through sequence co-occurence statistics, which can pose a chellenge for designing structured RNAs such as riboswitches, aptamers, and ribozymes. RibonanzaNet [He et al., 2024] represents a recent effort in developing structure-informed RNA language models by supervised training on experimental readouts from chemical mapping, although RibonanzaNet cannot be used for RNA design. Inverse folding methods like gRNAde are language models conditioned on 3D structure, making them a natural choice for structure-based design.

Comparison to contemporaneous work. Concurrently, Tan et al. [2023] also developed a deep learning-based 3D RNA inverse folding model. We want to emphasize that this is independent work, but for completeness we include a discussion on key differences to gRNAde:

- Methodology:
  - *New capabilities*: gRNAde enables explicit multi-state design to generate sequences conditioned on multiple backbone structures, which is not possible with Rosetta nor Tan et al. [2023]'s approach. We have also demonstrated the utility of gRNAde's perplexity for zero-shot ranking of mutants in RNA engineering campaigns.
  - *Decoding*: gRNAde uses an autoregressive decoder with rotation-equivariant GNN layers, while Tan et al. [2023] use a non-autoregressive (one-shot) decoder with rotation-invariant layers. In our ablation study (Appendix D), we found autoregressive decoding to show significantly higher 2D and 3D self-consistency scores than non-autoregressive decoding, even though non-autoregressive decoding lead to higher sequence recovery. Autoregressive decoding is more expressive and can condition predictions at each decoding step on past predictions, while one-shot decoders sample from independent probability distributions for each nucleotide. We find autoregressive decoding to be a better inductive bias for predicting base pairing and base stacking interactions that are drivers of RNA structure [Vicens and Kieft, 2022]. For instance, G-C and A-U pairs can often be swapped for one another, but non-autoregressive decoding does not capture such paired constraints.

- Evaluation:
  - *Evaluation metrics*: Tan et al. [2023] focus on measuring native sequence recovery, only. We have additionally introduced structural self-consistency metrics at the 2D and 3D level, which have been shown to better correlate with experimental success in protein design.
  - *Perplexity*: We found gRNAde's perplexity to be correlated with sequence and structural recovery, as well as demonstrated its utility for zero-shot ranking of mutants in RNA engineering. On the other hand, Tan et al. [2023] do not report perplexity and claim that perplexity is an unsuitable metric for RNA design.
  - *Data splitting*: While both studies use structural clustering to evaluate generalisation to structurally dissimilar RNAs, Tan et al. [2023]'s test splits are determined randomly. Our experiments use currated test splits from Das et al. [2010] to fairly compare gRNAde to physically based Rosetta, as well as split based on structural flexibility to benchmark multi-state design.
- Usage and reproducibility: We release open source training and inference code as well as model checkpoints to enable complete reproducibility. We also release Colab notebooks and detailed tutorials to make gRNAde broadly applicable and useful in real-world RNA design campaigns. At present, it is not possible to reproduce the results in Tan et al. [2023] or compare to gRNAde directly as no training code is available.

B FAQs on using gRNAde

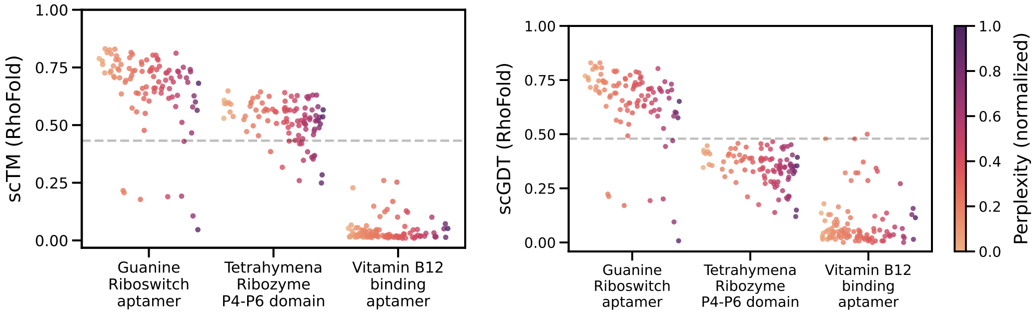
How to chose the number of states to provide as input to gRNAde? In general, this would depend on the design objective. For instance, designing riboswitches may necessitate multi-state design, while a single-state pipeline may be more sensible for locking an aptamer into its bound conformation [Yesselman et al., 2019]. Note that it may be possible to benefit from multi-state gRNAde models even when performing single-state design by using slightly noised variations of the same backbone structure as an input conformational ensemble.

How to prioritise or chose amongst designed sequences? We have currently provided 3 types of evaluation metrics: native sequence recovery, structural self-consistency scores and perplexity, towards this end. We suspect that recovery may not be the ideal choice, except for design scenarios where we require certain regions of the RNA sequence to be conserved or native-like. Self-consistency scores may provide an overall more holistic evaluation metric as they accounts for alternative base pairings which still lead to similar structures as well as better capture the recovery of structural motifs responsible for functionality. However, structural self-consistency scores inherit the limitations of the structure prediction methods used as part of their computation. For instance, computing the self-consistency score between an RNA backbone and its own native sequence provides an upper bounds on the maximum score that designs can obtain under a given structure prediction method. Lastly, gRNAde's perplexity estimates the likelihood of a sequence given a backbone and can be useful for ranking designs and mutants in RNA engineering campaigns (especially for design scenarios where structure prediction tools are not performant).

In real-world design scenarios, we can pair gRNAde with another machine learning model (an 'oracle') for ranking or predicting the suitability of designed sequences for the objective (for instance, binding affinity or some other notion of fitness). We hope to conduct further experimental validation of gRNAde designs in the wet lab in order to better understand these tradeoffs.

Why not average single-state logits over multiple states for multi-state design? ProteinMPNN [Dauparas et al., 2022] proposes to average logits from multiple backbones for multi-state protein design. Here is a simple example to highlight issues with such an approach: Consider two states A and B, and choice of labels X, Y, and Z. For state A: X, Y, Z are assigned probabilities 75%, 20%, 5%. For state B: X, Y, Z are assigned probabilities 5%, 20%, 75%. Logically, label Y is the only one that is compatible with both states. However, averaging the probabilities would lead to label X or Z being more likely to be sampled in designs. As an alternative, gRNAde is based on multi-state GNNs which can take as input one or more backbone structures and generate sequences conditioned on the conformational ensemble directly.

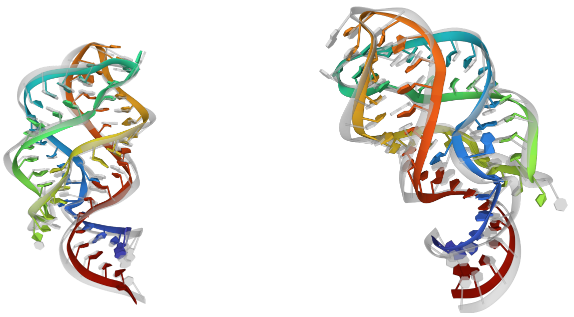
C 3D Visualisation of gRNAde Designs



(a) self-consistency TM-score

(b) self-consistency GDT_TS

Figure 6: **3D self-consistency scores for 3 representative RNAs from Das et al. [2010].** We use RhoFold to 'forward fold' 100 designs sampled at temperature = 0.5 and plot self-consistency TM-score and GDT_TS. Each dot corresponds to one designed sequence and is coloured by gRNAde's perplexity (normalised per RNA). Designs with lower relative perplexity generally have higher 3D self-consistency and can be considered more 'designable'. Dotted lines represent TM-score and GDT_TS thresholds of 0.45 and 0.50, repsectively. Pairs of structures scoring higher than the threshold correspond to roughly the same fold.



Design 1:
GGCAAGUAAUCCCUACGCUAUG
GGUAGGGAGUCUCAGCAGUGAC
CCGUAAAGUUACUACCUUGCCC

perplexity: 1.3097
recovery: 0.5909 (27 edits)
sc2D = 0.9227
scRMSD = 1.3839
scTM = 0.8309
scGDT = 0.8295

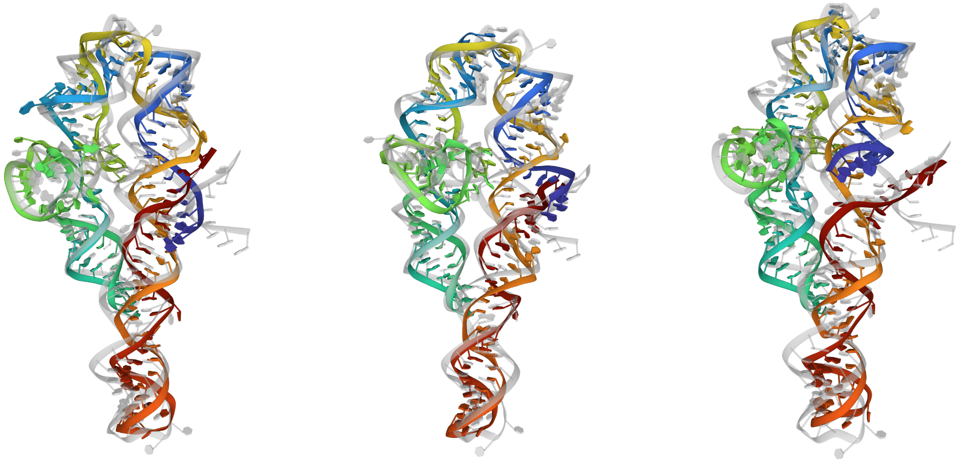
Design 2:
CGGUGGUAAGCCCAACGCUAGG
GGUUGGGCGUCUCAGCACAGUC
CCGUAAAGAUUGUACCCACCGG

perplexity: 1.3815
recovery: 0.4091 (37 edits)
sc2D = 0.9227
scRMSD = 2.1249
scTM = 0.6874
scGDT = 0.6780

Design 3:
AGCAAGUAAUGCCAUCGCUAUG
GGAUGGUAGUGUCAGCACUGAC
CCUUAAAGUUAGUACCUUGCUU

perplexity: 1.4247
recovery: 0.5152 (30 edits)
sc2D = 0.9227
scRMSD = 3.2131
scTM = 0.5118
scGDT = 0.5265

Figure 7: **Cherry-picked designs for Guanine riboswitch aptamer** (PDB: 4FE5, sequence: GGACAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGUCC).



Design 1:
GGGGCUCCGGCGACGCAGUCGAAAG
CCCAGCAGUACCAAGCCUCAGGGGA
AACUUUGAGGUGGCCUAACAAAGGA
UACGGUAAUAAGCUGCGGGAAAAGG
UUGUAAGCCGGAGCGAAGACCUAAG
GCACCGCUUUUGGCGGUGCUAUGGU
UGAAGUUAA

perplexity: 1.2462
recovery: 0.7170 (44 edits)
sc2D = 0.8301
scRMSD = 5.4562
scTM = 0.6481
scGDT = 0.4465

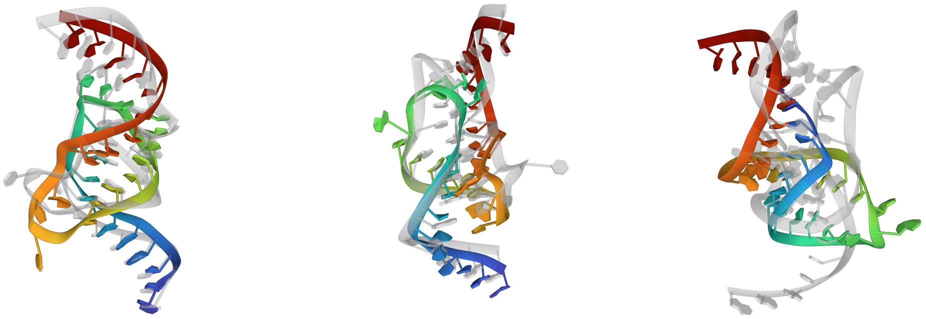
Design 2:
GGGGUACCGGCGACGCAGUCGAAUG
CCCUGUGGUACCAAGCCCCGGGGGA
AACUUCGGGGUGGCCUUACCAAGGA
CACGGUAAUAAGCCACGGGAAAUGG
UUGUAAGCCGGUCCGAAGCCCUAAG
GCCGCGCUUUGGGCGCGGCUAUGGG
UGAAGGCAA

perplexity: 1.3273
recovery: 0.6226 (58 edits)
sc2D = 0.6896
scRMSD = 6.7239
scTM = 0.6300
scGDT = 0.4513

Design 3:
GAGGCCACGGCAACGCAGUCUAACG
CCCUGUGGUACCAAGUCUUAGGAGA
AAUUUUAAGAUGGCCUAAUAAAGGA
UAUGGUAAUAAGCCACGGGAAAAGG
UUGUAAGACGUGACGAAGUCCUAAG
GCCACAGUUUUGCUGUGGCUAUGGA
UGGAGUACA

perplexity: 1.3204
recovery: 0.7044 (45 edits)
sc2D = 0.7922
scRMSD = 8.8211
scTM = 0.4582
scGDT = 0.2909

Figure 8: **Cherry-picked designs for Tetrahymena Ribozyme P4-P6 domain** (PDB: 2R8S, sequence: GGAAUUGCGGGAAAGGGGUCAACAGCCGUUCAGUACCAAGUCUCAGGGGAAACUUUGAGAUGGCCUUGCAAAGGGU AUGGUAAUAAGCUGACGGACAUGGUCCUAACACGCAGCCAAGUCCUAAGUCAACAGAUCUUCUGUUGAUAUGGAUGCAGUUCA).



Design 1:
GUCAAACGCAGCCGAAA
GCGCGAUAGUCCCAGGAA

perplexity = 1.6237
recovery = 0.4571 (16 edits)
sc2D = -0.0074
scRMSD = 3.9505
scTM = 0.2597
scGDT = 0.4786

Design 2:
GGCAAACGCGGCCGAAA
GCGCGUGAGUCCCCGGAC

perplexity = 1.6630
recovery = 0.4857 (16 edits)
sc2D = -0.0099
scRMSD = 3.3549
scTM = 0.2526
scGDT = 0.5000

Design 3:
CGUAGUCGGAGCCGAAG
GGCCGUUAGUCCCAGGAG

perplexity = 1.7020
recovery = 0.4000 (17 edits)
sc2D = 0.4035
scRMSD = 16.4102
scTM = 0.0319
scGDT = 0.0571

Figure 9: **Cherry-picked designs for Vitamin B12 binding aptamer** (PDB: 1ET4, sequence: GGAACCGGUGCGCAUAACCACCUCAGUGCGAGCAA).

Table 1: Ablation study and aggregated benchmark results for gRNAde. We report metrics averaged over 100 test sets samples and standard deviations across 3 consistent random seeds. The percentages reported in brackets for the 3D self-consistency scores are the percentage of designed samples within the 'designability' threshold values (scRMSD≤2Å, scTM≥0.45, scGDT≥0.5).

							Self-consistency metrics			
							2D – EternaFold		3D – RhoFold	
Split	Max. #states	Model	GNN	Max. train length	Perplexity (↓)	Native seq. recovery (↑)	scMCC (↑)	scRMSD (↓)	scTM-score (↑)	scGDT_TS (↑)
Single-state split	1	AR	Equiv	500	1.77±0.07	0.438±0.01	0.624±0.07	13.01±1.18 (0.5%)	0.21±0.0 (14.3%)	0.22±0.0 (12.7%)
	1	AR	Equiv	1000	1.73±0.08	0.453±0.01	0.648±0.01	13.10±0.58 (1.0%)	0.20±0.0 (10.8%)	0.21±0.0 (10.6%)
	1	AR	Equiv	2500	1.41±0.01	0.493±0.01	0.633±0.03	11.76±0.91 (1.4%)	0.27±0.0 (28.8%)	0.27±0.0 (28.0%)
	1	AR	Equiv	5000	1.29±0.02	0.530±0.01	0.585±0.03	11.70±0.56 (1.3%)	0.26±0.0 (24.8%)	0.25±0.0 (20.1%)
	1	AR	Inv	5000	1.32±0.04	0.549±0.00	0.612±0.02	11.50±0.64 (1.9%)	0.28±0.0 (32.1%)	0.28±0.0 (26.2%)
	1	NAR	Inv	5000	1.54±0.04	0.571±0.00	0.430±0.02	14.26±0.51 (1.3%)	0.19±0.0 (15.9%)	0.18±0.0 (12.7%)
	1	NAR	Equiv	5000	1.46±0.06	0.584±0.00	0.473±0.02	13.04±0.88 (1.3%)	0.23±0.0 (24.0%)	0.22±0.0 (17.9%)
	3	AR	Equiv	5000	1.23±0.05	0.539±0.01	0.620±0.01	11.47±1.05 (2.5%)	0.28±0.0 (31.4%)	0.28±0.0 (27.2%)
	5	AR	Equiv	5000	1.25±0.01	0.539±0.02	0.596±0.03	11.90±1.00 (2.9%)	0.27±0.0 (31.6%)	0.26±0.0 (26.4%)
	Groundtruth sequence prediction baseline:				-	1.000±0.00	0.686±0.00	5.23±0.07 (27.9%)	0.56±0.0 (68.7%)	0.55±0.0 (68.7%)
	Random sequence prediction baseline:				-	0.251±0.00	0.012±0.00	24.40±0.34 (0.0%)	0.04±0.0 (0.0%)	0.02±0.0 (0.0%)
	ViennaRNA 2D-only baseline:				-	0.259±0.00	0.611±0.00	20.34±0.10 (0.0%)	0.07±0.0 (0.6%)	0.07±0.0 (1.1%)
Multi-state split	1	AR	Equiv	500	1.87±0.06	0.445±0.01	0.603±0.03	13.08±0.20 (3.5%)	0.10±0.0 (1.2%)	0.25±0.0 (20.7%)
	1	AR	Equiv	1000	1.84±0.01	0.447±0.01	0.580±0.01	13.02±0.56 (2.3%)	0.09±0.0 (0.9%)	0.25±0.0 (20.4%)
	1	AR	Equiv	2500	1.73±0.04	0.480±0.02	0.567±0.01	12.83±0.05 (3.4%)	0.10±0.0 (1.9%)	0.26±0.0 (21.2%)
	1	AR	Equiv	5000	1.68±0.03	0.455±0.01	0.569±0.02	12.88±0.20 (4.1%)	0.11±0.0 (1.6%)	0.26±0.0 (22.6%)
	1	AR	Inv	5000	1.72±0.01	0.463±0.01	0.559±0.03	13.09±0.27 (4.1%)	0.10±0.0 (2.2%)	0.27±0.0 (23.0%)
	1	NAR	Inv	5000	2.01±0.04	0.457±0.01	0.461±0.01	14.06±0.23 (3.2%)	0.08±0.0 (1.7%)	0.23±0.0 (16.5%)
	1	NAR	Equiv	5000	1.89±0.06	0.432±0.01	0.423±0.01	13.63±0.27 (3.6%)	0.09±0.0 (1.2%)	0.24±0.0 (18.3%)
	3	AR	Equiv	5000	1.60±0.03	0.467±0.03	0.561±0.03	13.31±0.38 (3.4%)	0.10±0.0 (2.6%)	0.24±0.0 (19.0%)
	5	AR	Equiv	5000	1.55±0.04	0.473±0.01	0.549±0.03	13.48±0.79 (3.3%)	0.10±0.0 (3.0%)	0.24±0.0 (20.2%)
	Groundtruth sequence prediction baseline:				-	1.000±0.00	0.570±0.01	9.78±0.13 (10.3%)	0.16±0.0 (11.7%)	0.36±0.0 (36.7%)
	Random sequence prediction baseline:				-	0.249±0.00	0.128±0.00	21.15±0.21 (0.9%)	0.02±0.0 (0.0%)	0.09±0.0 (3.3%)
	ViennaRNA 2D-only baseline:				-	0.258±0.00	0.601±0.00	15.47±0.20 (2.4%)	0.05±0.0 (0.2%)	0.19±0.0 (15.2%)
All data	1	AR	Equiv	5000	1.23±0.01	0.733±0.00	0.627±0.02	8.10±0.28 (20.7%)	0.42±0.0 (46.1%)	0.41±0.0 (43.0%)
	2	AR	Equiv	5000	1.21±0.01	0.783±0.01	0.629±0.03	8.40±0.09 (19.1%)	0.42±0.0 (47.8%)	0.41±0.0 (41.7%)
	3	AR	Equiv	5000	1.19±0.01	0.787±0.01	0.606±0.02	7.88±0.68 (20.5%)	0.43±0.0 (47.4%)	0.42±0.0 (44.0%)
	5	AR	Equiv	5000	1.15±0.01	0.811±0.01	0.617±0.02	7.51±0.30 (20.7%)	0.45±0.0 (50.2%)	0.44±0.0 (46.7%)

D Ablation Study

Table 1 presents an ablation study as well as aggregated benchmark for various configurations of gRNAde. Key takeaways are highlighted below. Note that all results in the main paper are reported for models trained on the maximum length of 5000 nucleotides using autoregressive decoding and rotation-equivariant GNN layers, as this lead to the lowest perplexity values.

Max. train RNA length Limiting the maximum length of RNAs used for training can be seen as ablating the use of ribosomal RNA families (which are thousands of nucleotides long and form complexes with specialised ribosomal proteins). We find that training on only short RNAs fewer than 1000s of nucleotides leads to worse sequence recovery and 3D self-consistency scores, even though it improves 2D self-consistency across both evaluation splits. This suggests that tertiary interactions learnt from ribosomal RNAs can generalise to other RNA families to some extent (large ribosomal RNAs were excluded from test sets).

GNN We ablated whether the internal representations of the GVP-GNN are rotation invariant or equivariant. Equivariant GNNs are theoretically more expressive [Joshi et al., 2023] and we do find them more capable at fitting the training distribution (as shown by lower perplexity). However, we do not find significant differences in terms of other performance metrics across different GNN layers.

Model 'AR' implies autoregressive decoding (described in Section 2.3, uses 4 encoder and 4 decoder layers), while 'NAR' implies non-autoregressive, one-shot decoding using an MLP (uses 8 encoder layers). Across both evaluation splits, AR models show significantly higher self-consistency scores than NAR, even though NAR lead to higher sequence recovery. AR is more expressive and can condition predictions at each decoding step on past predictions, while one-shot NAR samples from independent probability distributions for each nucleotide. Thus, AR is a better inductive bias for predicting base pairing and base stacking interactions that are drivers of RNA structure [Vicens and Kieft, 2022]. For instance, G-C and A-U pairs can often be swapped for one another, but non-autoregressive decoding does not capture such paired constraints.

Max. #states We evaluate the impact of increasing the maximum number of states as input to gRNAde. Multi-state models marginally improve native sequence recovery as well as structural self-consistency scores over an equivalent single state variant, even for the single-state benchmark where the multi-state model is being used with only one state as input. This suggests that seeing multiple states during training can be useful for gRNAde's performance even for single-state design tasks.

Non-learnt baselines. We report the performance of two non-learnt baselines to contextualise gRNAde's performance: for each test sample, simply predicting the groundtruth sequence back and predicting a random sequence. Structural self-consistency scores for the Groundtruth baseline provides a rough upper bounds on the maximum score that any gRNAde designs can theoretically obtain given the current state of 2D/3D structure predictors being used. gRNAde always performs better than the random baseline and often reaches 2D self-consistency scores close to the upper bound. Both 2D and 3D self-consistency scores are inherently limited by the performance of the structure prediction methods used.

2D inverse folding baseline. We additionally report results for ViennaRNA's 2D-only inverse folding method to further demonstrate the utility of 3D inverse folding. ViennaRNA has improved 2D self-consistency scores over gRNAde but fails to capture tertiary interactions in its designs, as evident by poor recovery and 3D self-consistency scores similar to the random baseline.

Split. Single- and multi-state splits are described in Section 3; the multi-state split is relatively harder than the single-state split based on overall reduced performance for all baselines and models. Models trained on 'All data' use all RNASolo samples for training, solely for the purpose of releasing the best possible gRNAde checkpoints for real-world usage. Evaluation metrics for 'All data' are reported on the single-state test set.

E Additional Results

Table 2: Full results for Figure 2 comparing gRNAde to Rosetta, FARNA and ViennaRNA for single-state design on 14 RNA structures of interest identified by Das et al. [2010]. Rosetta and FARNA recovery values are taken from Das et al. [2010], Supplementary Table 2.

		ViennaRNA	**FARNA**	**Rosetta**	**gRNAde (single-state)**		
PDB ID	**Description**	Recovery	Recovery	Recovery	Recovery	Perplexity	2D self-cons.
1CSL	RRE high affinity site	0.25	0.20	0.44	0.5719	1.2812	0.8644
1ET4	Vitamin B12 binding RNA aptamer	0.25	0.34	0.44	0.6250	1.3457	-0.0135
1F27	Biotin-binding RNA pseudoknot	0.30	0.36	0.37	0.3437	1.6203	0.4523
1L2X	Viral RNA pseudoknot	0.24	0.45	0.48	0.4721	1.3181	0.5692
1LNT	RNA internal loop of SRP	0.33	0.27	0.53	0.5843	1.4337	0.1379
1Q9A	Sarcin/ricin domain from E.coli 23S rRNA	0.27	0.40	0.41	0.5044	1.3411	0.0597
4FE5	Guanine riboswitch aptamer	0.29	0.28	0.36	0.5300	1.3824	0.9116
1X9C	All-RNA hairpin ribozyme	0.26	0.31	0.50	0.5000	1.3905	0.6630
1XPE	HIV-1 B RNA dimerization initiation site	0.27	0.24	0.40	0.7037	1.2177	0.7768
2GCS	Pre-cleavage state of glmS ribozyme	0.25	0.26	0.44	0.5078	1.3053	0.4062
2GDI	Thiamine pyrophosphate-specific riboswitch	0.25	0.38	0.48	0.6500	1.2363	-0.0251
2OEU	Junctionless hairpin ribozyme	0.23	0.30	0.37	0.9519	1.0913	0.7768
2R8S	Tetrahymena ribozyme P4-P6 domain	0.27	0.36	0.53	0.5689	1.1881	0.7281
354D	Loop E from E. coli 5S rRNA	0.28	0.35	0.55	0.4410	1.4938	0.0430
	Overall recovery:	0.27	0.32	0.45	0.5682		

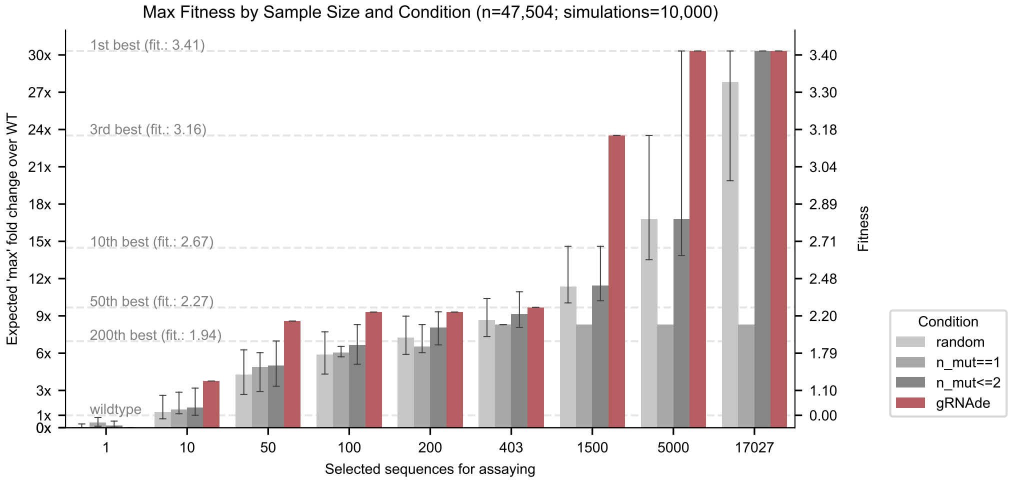


Figure 10: **Retrospective study of gRNAde for ranking ribozyme mutant fitness (t1 subunit).** Using the backbone structure and mutational fitness landscape data from an RNA polymerase ribozyme [McRae et al., 2024], we retrospectively analyse how well we can rank variants at multiple design budgets using random selection vs. gRNAde's perplexity for mutant sequences conditioned on the backbone structure (scaffolding subunit t1). gRNAde performs better than single site saturation mutagenesis, even when all single mutants are explored (total of 403 single mutants, 17,027 double mutants for the scaffolding subunit t1 in McRae et al. [2024]). See Section 4.3 for results on catalytic subunit 5TU and further discussions.

F Additional Figures

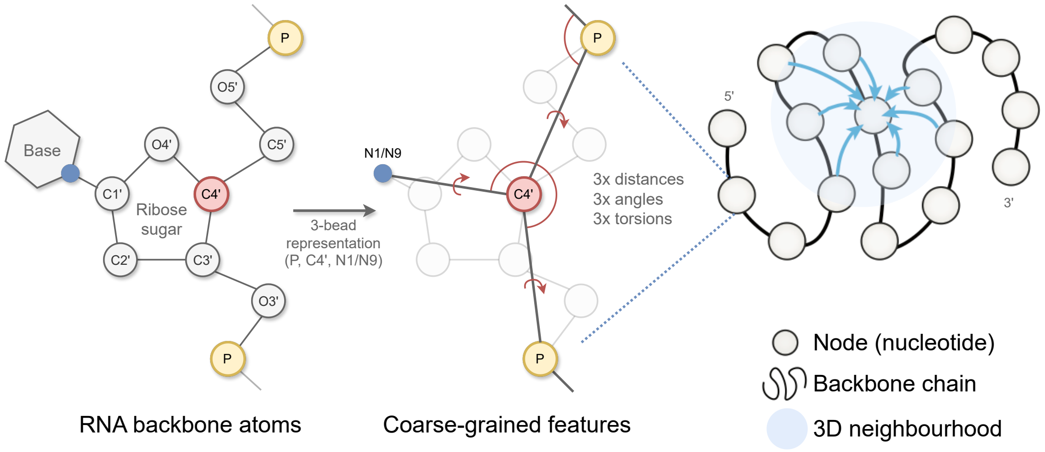


Figure 11: **gRNAde featurizes RNA backbone structures as 3D geometric graphs.** Each RNA nucleotide is a node in the graph, consisting of 3 coarse-grained beads for the coordinates for P, C4', N1 (pyrimidines) or N9 (purines) which are used to compute initial geometric features and edges to nearest neighbours in 3D space. Backbone chain figure adapted from Ingraham et al. [2019].

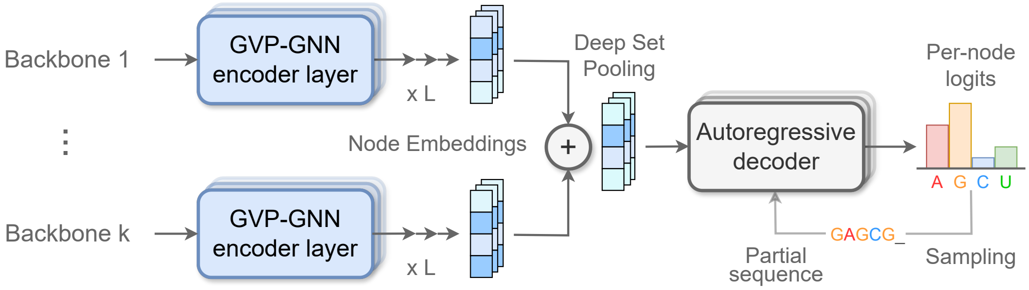


Figure 12: **gRNAde model architecture.** One or more RNA backbone geometric graphs are encoded via a series of SE(3)-equivariant Graph Neural Network layers [Jing et al., 2020] to build latent representations of the local 3D geometric neighbourhood of each nucleotide within each state. Representations from multiple states for each nucleotide are then pooled together via permutation invariant Deep Sets [Zaheer et al., 2017], and fed to an autoregressive decoder to predict a probabilities over the four possible bases (A, G, C, U). The probability distribution can be sampled to design a set of candidate sequences. During training, the model is trained end-to-end by minimising a cross-entropy loss between the predicted probability distribution and the true sequence identity.

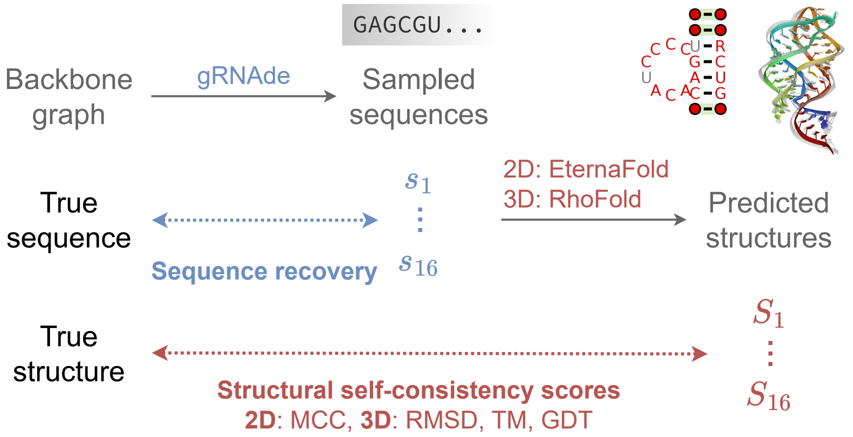


Figure 13: **In-silico evaluation metrics for gRNAde designed sequences.** We consider (1) *sequence recovery*, the percentage of native nucleotides recovered in designed samples, (2) *self-consistency scores*, which are measured by 'forward folding' designed sequences using a structure predictor and measuring how well 2D and 3D structure are recovered (we use EternaFold and RhoFold for 2D/3D structure prediction, respectively). We also report (3) *perplexity*, the model's estimate of the likelihood of a sequence given a backbone.

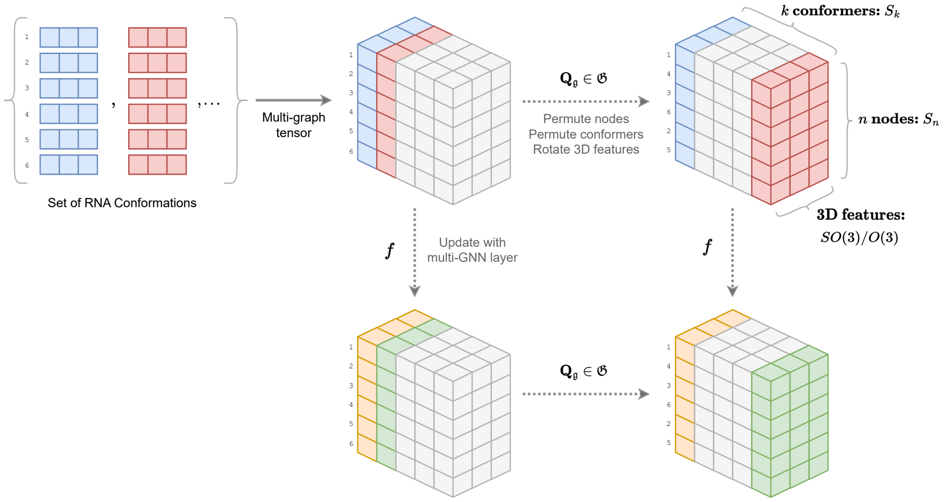
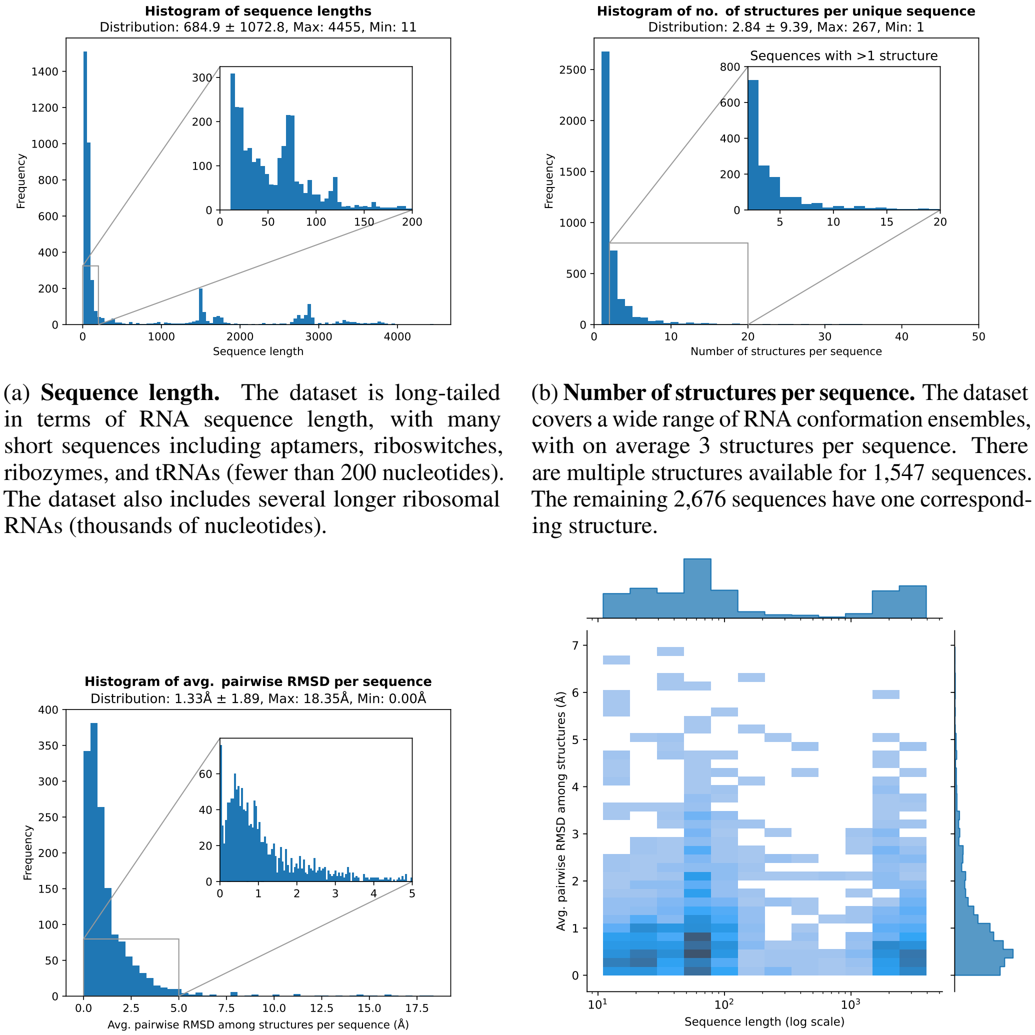

Set of RNA Conformations
Multi-graph tensor
$\mathbf{Q}_\mathfrak{g} \in \mathfrak{G}$
Permute nodes
Permute conformers
Rotate 3D features
k conformers: S_k
n nodes: S_n
3D features: $SO(3)/O(3)$
f
Update with multi-GNN layer
f
$\mathbf{Q}_\mathfrak{g} \in \mathfrak{G}$

Figure 14: **Multi-graph tensor representation of RNA conformational ensembles**, and the associated symmetry groups acting on each axis. We process a set of k RNA backbone conformations with n nodes each into a tensor representation. Each multi-state 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.

```
class MultiGVPConv(MessagePassing):
    ’’’GVPConv for handling multiple conformations’’’

    def __init__(self, ...):
        ...

    def forward(self, x_s, x_v, edge_index, edge_attr):

        # stack scalar feats along axis 1:
        # [n_nodes, n_conf, d_s] -> [n_nodes, n_conf * d_s]
        x_s = x_s.view(x_s.shape[0], x_s.shape[1] * x_s.shape[2])

        # stack vector feat along axis 1:
        # [n_nodes, n_conf, d_v, 3] -> [n_nodes, n_conf * d_v*3]
        x_v = x_v.view(x_v.shape[0], x_v.shape[1] * x_v.shape[2]*3)

        # message passing and aggregation
        message = self.propagate(
            edge_index, s=x_s, v=x_v, edge_attr=edge_attr)

        # split scalar and vector channels
        return _split_multi(message, d_s, d_v, n_conf)

    def message(self, s_i, v_i, s_j, v_j, edge_attr):

        # unstack scalar feats:
        # [n_nodes, n_conf * d] -> [n_nodes, n_conf, d_s]
        s_i = s_i.view(s_i.shape[0], s_i.shape[1]//d_s, d_s)
        s_j = s_j.view(s_j.shape[0], s_j.shape[1]//d_s, d_s)

        # unstack vector feats:
        # [n_nodes, n_conf * d_v*3] -> [n_nodes, n_conf, d_v, 3]
        v_i = v_i.view(v_i.shape[0], v_i.shape[1]//(d_v*3), d_v, 3)
        v_j = v_j.view(v_j.shape[0], v_j.shape[1]//(d_v*3), d_v, 3)

        # message function for edge j-i
        message = tuple_cat((s_j, v_j), edge_attr, (s_i, v_i))
        message = self.message_func(message)  # GVP

        # merge scalar and vector channels along axis 1
        return _merge_multi(*message)

def _split_multi(x, d_s, d_v, n_conf):
    ’’’
    Splits a merged representation of (s, v) back into a tuple.
    ’’’
    s = x[..., :-3 * d_v * n_conf].view(x.shape[0], n_conf, d_s)
    v = x[..., -3 * d_v * n_conf:].view(x.shape[0], n_conf, d_v, 3)
    return s, v

def _merge_multi(s, v):
    ’’’
    Merges a tuple (s, v) into a single ‘torch.Tensor‘,
    where the vector channels are flattened and
    appended to the scalar channels.
    ’’’
    # s: [n_nodes, n_conf, d] -> [n_nodes, n_conf * d_s]
    s = s.view(s.shape[0], s.shape[1] * s.shape[2])
    # v: [n_nodes, n_conf, d, 3] -> [n_nodes, n_conf * d_v*3]
    v = v.view(v.shape[0], v.shape[1] * v.shape[2]*3)
    return torch.cat([s, v], -1)
```

Listing 1: **PyG-style pseudocode for a multi-state GVP-GNN layer.** We update node features for each conformer independently while maintaining permutation equivariance of the updated feature tensors along both the first (no. of nodes) and second (no. of conformations) axes.



(a) **Sequence length.** The dataset is long-tailed in terms of RNA sequence length, with many short sequences including aptamers, riboswitches, ribozymes, and tRNAs (fewer than 200 nucleotides). The dataset also includes several longer ribosomal RNAs (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 1,547 sequences. The remaining 2,676 sequences have one corresponding structure.

(c) **Average pairwise RMSD per sequence.** For 1,547 sequences with multiple structures, there is significant structural diversity among conformations. On average, the pairwise C4' 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 15: **RNASolo data statistics.** We plot histograms to visualise the diversity of RNAs available 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.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract are that the proposed 3D RNA design method, gRNAde, improves sequence recovery over Rosetta, is capable of multi-state design, and can be useful for zero-shot ranking of RNA fitness landscapes. All the claims are supported by empirical results and expanded upon in detail in the rest of the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, we have discussed limitations at several places, including the conclusion section, an FAQ section, as well as a detailed ablation study in the appendix.

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: All code, data, and pretrained models are publicly available, along with detailed instructions on installation, reproducing results, and real-world usage.

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

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

   Answer: [Yes]

   Justification: All data and code is publicly available, including the exact commands and environments needed to access the raw data, process the data, and reproduce the results.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: The experimental setup is presented in detail as a dedicated section in the main text, as well as extensively documented in the code.

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: All results are accompanied by error bars and confidence intervals, along with the factors of variability that the error bars capture.

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: Yes, the paper and code provide information on the computer resources used for this work. However, we currently do not have estimes on the total compute used.

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We hope that our tools contributes to the development of RNA-based therapeutics towards improving health outcomes. We have attempted to make gRNAde as convinient to use as possible towards this end. We do not foresee any immediate negative societal impact of our work.

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Original owners of any assets used as part of our stidy are appropriately credited.

13. **New Assets**

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

    Answer: [Yes]

    Justification: Our datasets, code, and model checkpoints are publicly available under the permissive MIT License.

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.