RESTFLOW: STABLE FEW-STEP RECTIFIED FLOW FOR RNA BACKBONE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA molecules play central roles in biological regulation and therapeutic development, with many of their functions closely tied to three-dimensional structure. Recent diffusion and flow-based generative models provide effective frameworks for unconditional RNA backbone generation. However, existing approaches often require many sampling steps to maintain steric validity, limiting their efficiency in large-scale candidate generation. Here, we introduce ReSTFlow, a rectified flow framework that integrates steric regularization and trajectory correction for few-step, high-quality RNA backbone generation. The first stage trains STFlow, a sterically regularized flow model that represents each nucleotide as a local rigid frame and learns continuous transport from a noise distribution to RNA backbone conformations under clash-aware supervision. In the second stage, the learned flow is rectified using trajectories constructed from STFlow-generated samples, resulting in straighter transport paths that can be accurately integrated with substantially fewer sampling steps. At inference, trajectory correction stabilizes coarse translational updates during sampling. With 20 sampling steps, ReSTFlow achieves 75.0% steric validity, and 69.8% of its sterically valid backbones are designable, demonstrating strong performance in the few-step setting. These results establish ReSTFlow as an effective framework for high-quality RNA backbone generation under limited sampling budgets.

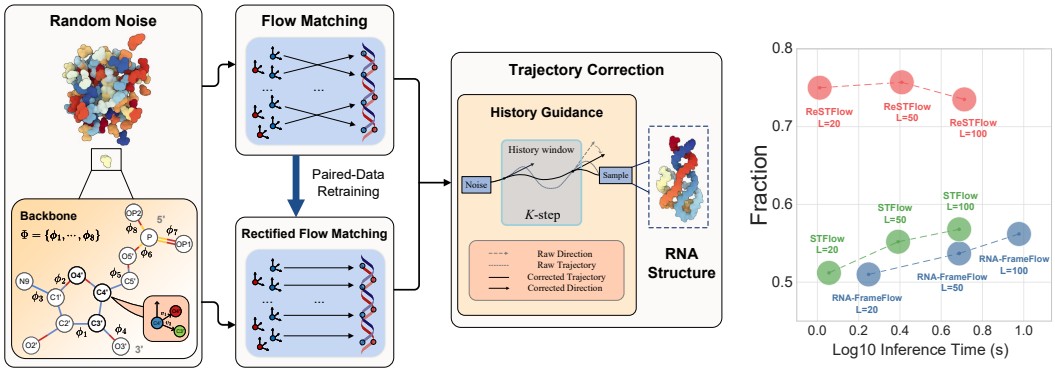
1 INTRODUCTION

RNA molecules perform diverse regulatory and therapeutic functions that are critically shaped by the three-dimensional structures they adopt (Vicens & Kieft, 2022). Structure-guided de novo RNA design seeks to construct new RNA molecules with desired structures or functions (Joshi et al., 2025). A key component of this process is RNA backbone generation, which produces three-dimensional backbone conformations without relying on a predefined sequence or structural template. This task remains challenging because RNA backbones must satisfy stringent local geometric constraints across multiple torsional degrees of freedom while accommodating highly flexible global conformations (Vicens & Kieft, 2022; Anand et al., 2025). The limited availability of experimentally determined RNA structures (Adamczyk et al., 2022; Anand et al., 2025) further increases the difficulty of learning physically realistic backbone distributions.

Recent diffusion and flow-based generative models have provided effective frameworks for RNA structure generation, including RNA backbone generation (Morehead et al., 2024; Nori & Jin, 2024; Anand et al., 2025; Ma et al., 2025). Among these methods, RNA-FrameFlow (Anand et al., 2025) formulates unconditional RNA backbone generation as a continuous transport process from randomly initialized frames to three-dimensional backbone conformations. However, existing approaches often require many sampling steps to maintain steric validity and designability, which limits their efficiency in large-scale candidate generation (Anand et al., 2025; Yue et al., 2025). Reducing the number of integration steps can substantially accelerate sampling, but coarse numerical updates may destabilize sampling trajectories and degrade structural quality. This difficulty in achieving both high quality and efficient sampling remains a key limitation of RNA backbone generation methods.

To address this limitation, we propose ReSTFlow, a two-stage rectified flow framework that integrates steric regularization and trajectory correction for few-step RNA backbone generation. As

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



(a) The framework of ReSTFlow (b) Efficiency-quality comparison

Figure 1: Overview of ReSTFlow and its inference efficiency and validity. (a) The framework combines clash-aware flow matching, rectification training on noise-sample pairs, and history-conditioned trajectory correction at inference. (b) Average inference time versus steric validity (clash-free fraction) for RNA backbones of length $N = 100$ at different numbers of sampling steps L. Dashed lines connect configurations of the same method.

illustrated in Figure 1(a), the first stage trains STFlow, which represents each nucleotide using a local frame and learns continuous transport from random initial frames to RNA backbone conformations. To improve steric plausibility, STFlow incorporates a clash loss based on van der Waals radii into the flow-matching objective, providing clash-aware supervision during training. In the second stage, the learned flow is rectified using trajectories constructed from STFlow-generated samples, resulting in more direct transport paths that can be accurately integrated with fewer sampling steps. At inference, we introduce a history-conditioned trajectory correction inspired by Proximal-Mean Inversion (Wang et al., 2026). This correction stabilizes coarse translational updates of the nucleotide frames. Together, these components improve steric validity and support stable generation with only a few sampling steps.

Experiments demonstrate the effectiveness of ReSTFlow for few-step RNA backbone generation. As shown in Figure 1(b), with only 20 sampling steps, ReSTFlow achieves 75.0% steric validity, the highest among all methods evaluated under the same sampling budget. Under our benchmarking setup, the average inference time is 1.03 s per sample, compared with 1.77 s for RNA-FrameFlow. These results establish ReSTFlow's effectiveness in few-step RNA backbone generation.

2 Related Work

2.1 RNA Structure Prediction

Existing RNA modeling methods broadly span structure prediction and generative design, with most formulations assuming access to sequence information, molecular context, or both. For single-chain structure prediction, RhoFold+ (Shen et al., 2024) infers RNA structures from representations learned by RNA language models, whereas DRfold (Li et al., 2023) predicts local-frame orientations and geometric restraints that are combined into a hybrid energy potential for structure assembly. More general-purpose predictors extend the scope from individual RNA structures to biomolecular complexes. RoseTTAFoldNA (Baek et al., 2024) models both RNA structures and protein–nucleic acid complexes, while AlphaFold 3 (Abramson et al., 2024) supports a broader range of biomolecular complexes.

2.2 RNA Backbone Generation

RNA-FrameFlow (Anand et al., 2025) introduces a flow-matching framework for unconditional RNA backbone generation. It represents each nucleotide as a local $\mathrm{SE}(3)$ frame and learns a continuous transport from a simple prior to the distribution of RNA backbone conformations, enabling generation without predefined sequences or structural templates. However, the generated backbones

still exhibit limitations in structural quality, while performance under substantially reduced sampling budgets remains underexplored. These limitations motivate approaches that improve generation quality while enabling reliable few-step sampling.

2.3 Efficient Sampling

Rectified flow offers a principled route to efficient generation by simplifying the geometry of the learned transport (Liu et al., 2023). Through source–target recoupling, it reduces path crossings and curvature, enabling more reliable integration under coarse time discretization (Liu et al., 2023; Lee et al., 2023). ReQFlow (Yue et al., 2025) brings this principle to protein backbone generation by combining quaternion flow matching with rectification, enabling efficient sampling. However, RNA's richer local geometry and stringent steric constraints make few-step generation particularly susceptible to geometric drift and steric clashes (Anand et al., 2025). ReSTFlow addresses these challenges by combining rectification with clash-aware supervision and trajectory correction.

3 Methods

3.1 RNA Backbone Parameterization

As illustrated in Figure 1(a), each RNA nucleotide contains a phosphate group, a ribose ring, and a nucleobase. Directly generating the Cartesian coordinates of all backbone atoms would be unnecessarily complex because each nucleotide contains multiple atoms and has substantial conformational flexibility. Following prior work (Anand et al., 2025), we represent nucleotide i by a rigid local frame $T_i = (\mathbf{x}_i, R_i) \in \mathrm{SE}(3)$ and internal torsion angles Φ_i. Here, $\mathbf{x}_i \in \mathbb{R}^3$ is the frame origin, and $R_i \in \mathrm{SO}(3)$ is its rotation. A backbone of length N is therefore represented as $\mathcal{B} = \{T_i, \Phi_i\}_{i=1}^N$.

Specifically, we place the frame origin at $C4'_i$ in the ribose ring and use $C4'_i$, $O4'_i$, and $C3'_i$ as reference atoms. The directions $C4'_i \to O4'_i$ and $C4'_i \to C3'_i$ define the local orientation of the nucleotide. The flow state consists only of local frames. A separate head predicts endpoint torsions, which are not numerically integrated. Following RNA-FrameFlow, the coordinate and distance losses are evaluated on $\mathcal{S} = \{C3', C4', O4'\}$. Eight torsion angles $\Phi_i = \{\phi_{i,k}\}_{k=1}^8$ and fixed internal geometry reconstruct the remaining backbone atoms (Gelbin et al., 1996; Anand et al., 2025).

3.2 Flow Matching

Under this local-frame representation, we formulate RNA backbone generation as continuous flow modeling in $\mathrm{SE}(3)^N$ (Lipman et al., 2023; Chen & Lipman, 2024). We represent each $R_i \in \mathrm{SO}(3)$ by a unit quaternion $q_i \in \mathbb{S}^3$ to stabilize rotational interpolation and numerical updates (Solà, 2017; Yue et al., 2025). We write $T_s = \{(\mathbf{x}_{s,i}, q_{s,i})\}_{i=1}^N$ for $s \in \{0, t, 1\}$. Generation is then deterministic transport from a reference distribution p_0 to the empirical data distribution p_{data}. To train the base flow model, we independently sample $T_0 \sim p_0$ and $T_1 \sim p_{\mathrm{data}}$ and construct conditional paths between them.

We construct translation and rotation flows separately in $\mathbb{R}^3$ and $\mathrm{SO}(3)$.

Translation. We draw $\mathbf{e}_i \sim \mathcal{N}(\mathbf{0}, I_3)$, define $\bar{\mathbf{e}} = N^{-1}\sum_{j=1}^N \mathbf{e}_j$, and set $\mathbf{x}_{0,i} = 10(\mathbf{e}_i - \bar{\mathbf{e}})$. We connect this centered initial state to the ground-truth translation $\mathbf{x}_{1,i}$ by

$$\mathbf{x}_{t,i} = (1-t)\mathbf{x}_{0,i} + t\mathbf{x}_{1,i}, \qquad \mathbf{v}_i = \mathbf{x}_{1,i} - \mathbf{x}_{0,i}. \tag{1}$$

During training, we sample $t \in [0, 1]$.

Rotation. We independently sample $q_{0,i}$ from an isotropic $\mathrm{IGSO}(3)$ reference distribution (Yim et al., 2023b) with scale 1.5, while $q_{1,i}$ is obtained from the corresponding ground-truth frame. Before each interpolation or relative-rotation calculation, we select the representative of $q_{1,i}$ satisfying $q_{0,i}^\top q_{1,i} \geq 0$. We then define the rotation path and its angular velocity using spherical linear interpolation expressed through exponential and logarithmic maps:

$$q_{t,i} = q_{0,i} \otimes \mathrm{Exp}\left(t\,\mathrm{Log}\left(q_{0,i}^{-1} \otimes q_{1,i}\right)\right), \qquad \omega_i = 2\,\mathrm{Log}\left(q_{0,i}^{-1} \otimes q_{1,i}\right), \tag{2}$$

where $\otimes$ denotes quaternion multiplication. The operators $\mathrm{Exp}(\cdot)$ and $\mathrm{Log}(\cdot)$ are the exponential and logarithmic maps between unit quaternions and the associated Lie algebra.

Training. Given independently sampled endpoint frames $T_0 \sim p_0$ and $T_1 \sim p_{\mathrm{data}}$, we obtain the intermediate state $T_t = (\mathbf{x}_t, q_t)$ using Eq. (1) and Eq. (2). The flow model predicts endpoint frames and torsions as $(\hat{\mathbf{x}}_1, \hat{q}_1, \hat{\Phi}) = M_\theta(T_t, t)$.

Following RNA-FrameFlow (Anand et al., 2025), M_θ is $\mathrm{SE}(3)$-equivariant and uses node, pair, and time embeddings. Six processing blocks combine invariant point attention (Jumper et al., 2021), sequence Transformer layers (Vaswani et al., 2017), and equivariant frame updates (Yim et al., 2023a). A separate torsion head predicts normalized sine–cosine representations for the eight torsion angles.

The predicted endpoint frames define the translation and angular velocities

$$\hat{\mathbf{v}}_{t,i} = \frac{\hat{\mathbf{x}}_{1,i} - \mathbf{x}_{t,i}}{1-t}, \qquad \hat{\omega}_{t,i} \quad = \frac{2\,\mathrm{Log}\left(q_{t,i}^{-1} \otimes \hat{q}_{1,i}\right)}{1-t}. \tag{3}$$

The torsion head directly predicts the endpoint representation $\hat{\Phi}$ used to reconstruct non-frame atoms. Only frame variables are integrated during sampling.

The basic $\mathrm{SE}(3)$ flow matching objective consists of a translation-flow loss and a rotation-flow loss:

$$\mathcal{L}_{\mathbb{R}^3} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{v}_i - \hat{\mathbf{v}}_{t,i}\|_2^2, \qquad \mathcal{L}_{\mathrm{SO}(3)} = \frac{1}{N}\sum_{i=1}^{N} \|\omega_i - \hat{\omega}_{t,i}\|_2^2. \tag{4}$$

Steric Regularization. In addition to the coordinate, distance, and torsion losses adopted from RNA-FrameFlow (Anand et al., 2025), we introduce an atom-level penalty for steric overlap. These objectives do not explicitly enforce minimum distances between nonbonded atoms in the reconstructed endpoint. We aggregate inter- and intranucleotide violations at the atom level:

$$\mathcal{L}_{\mathrm{clash}} = \frac{1}{Z_c}\sum_{(i,a)\in\mathcal{A}} m_{i,a}\left(c_{i,a}^{\mathrm{inter}} + c_{i,a}^{\mathrm{intra}}\right), \qquad Z_c = \sum_{(i,a)\in\mathcal{A}} m_{i,a} + \epsilon_c. \tag{5}$$

Here, $\mathcal{A}$ is the set of reconstructed atom slots, and $m_{i,a} \in \{0,1\}$ is the valid-atom mask. For each valid atom (i,a), the two c terms denote accumulated hinge penalties from admissible inter- and intranucleotide neighbors. The internucleotide penalty uses the lower bound $r_a + r_b - \delta_{\mathrm{train}}$, where r_a and r_b are van der Waals radii (Bondi, 1964) and δ_{train} is the permitted overlap margin. The intranucleotide penalty uses the template-derived bound $\underline{d}_i^{ab}$. Each admissible pair contributes to the per-atom accumulators of its incident atoms. Thus, $\mathcal{L}_{\mathrm{clash}}$ measures the mean accumulated overlap per valid atom rather than the mean over all candidate pairs. Pair masks and covalent exclusions are detailed in the supplementary material. This differentiable penalty is distinct from the binary clash criterion used for evaluation.

The full training objective is

$$\begin{aligned} \mathcal{L} &= \lambda_{\mathrm{trans}}\mathcal{L}_{\mathbb{R}^3} + \lambda_{\mathrm{rot}}\mathcal{L}_{\mathrm{SO}(3)} + \mathcal{L}_{\mathrm{aux}}, \\ \mathcal{L}_{\mathrm{aux}} &= \lambda_{\mathrm{bb}}\mathcal{L}_{\mathrm{bb}} + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}} + \lambda_{\mathrm{tors}}\mathcal{L}_{\mathrm{tors}} + \lambda_{\mathrm{clash}}\mathcal{L}_{\mathrm{clash}}. \end{aligned} \tag{6}$$

The auxiliary losses are applied only at later flow times. Their weights and the complete definitions of the inherited terms are provided in the supplementary material.

3.3 Rectified Flow

Rectified flow retrains a base model on STFlow-generated endpoint pairs rather than independently sampled noise–data pairs (Liu et al., 2023; Yue et al., 2025). Following ReQFlow, this recoupling reduces path crossings and promotes straighter trajectories that are easier to integrate under a limited sampling budget. We generate the endpoint pairs with the trajectory-corrected sampler described in the next subsection and retain only clash-free endpoints.

Let $\mathcal{F}_{M_\theta}$ denote the terminal sampling map used for pair construction. Given an initial frame set $T_0' \sim p_0$, it returns endpoint frames T_1' and torsion targets Φ_1' as

$$(T_1', \Phi_1') = \mathcal{F}_{M_\theta}(T_0'). \tag{7}$$

The retained tuples (T_0', T_1', Φ_1') define an empirical coupling π_{ret} constructed from STFlow-generated endpoint pairs. Starting from STFlow, we perform one rectification round using the same interpolation paths and training objective as in base flow matching. The endpoint torsions Φ_1' continue to supervise the torsion head. This procedure produces ReSTFlow without changing the model architecture or introducing an additional training loss. Trajectory correction affects rectification only through the generated endpoint pairs.

Under the idealized population formulation and standard rectified-flow assumptions, rectification preserves the time-marginal laws of π_{ret} and yields lower or equal convex transport costs for its translation and quaternion components (Liu, 2022; Yue et al., 2025).

3.4 Trajectory-Corrected Sampling

We use the same trajectory-corrected sampler to construct rectification pairs and generate samples at inference time. In either setting, sampling starts from $T_0 = (\mathbf{x}_0, q_0) \sim p_0$. Throughout this paper, L sampling steps correspond to L Euler updates with a uniform step size $\Delta t = 1/L$. We set $t_j = j\Delta t$ for $j = 0, \ldots, L$. At each t_j, for $j = 0, \ldots, L-1$, the model predicts endpoint frames $\hat{T}_{1,j} = (\hat{\mathbf{x}}_{1,j}, \hat{q}_{1,j})$ and endpoint torsions $\hat{\Phi}_j$. Rotations are advanced with an exponentially weighted quaternion update (Yue et al., 2025):

$$q_{t_{j+1}} = q_{t_j} \otimes \mathrm{Exp}\left(\frac{1}{2}\Delta t\, \gamma e^{-\gamma t_j} \hat{\omega}_{t_j}\right). \tag{8}$$

Here, $\hat{\omega}_{t_j}$ is defined in Eq. (3), and we set $\gamma = 10$. Before computing each relative rotation, we select the representative of $\hat{q}_{1,j}$ that follows the shortest geodesic from q_{t_j}. Translations are advanced using the corrected velocity $\widetilde{\mathbf{v}}_{t_j}$ defined in Eq. (12). When no velocity history is available, we set $\widetilde{\mathbf{v}}_{t_j} = \hat{\mathbf{v}}_{t_j}$.

Coarse Euler integration can amplify inconsistencies between successive translational velocity estimates. To stabilize these updates, we apply a history-conditioned correction inspired by Proximal-Mean Inversion (Wang et al., 2026). The correction acts independently on each nucleotide and adds no trainable parameters. It does not modify quaternion updates or the training objective. During rectification, it influences learning only through the endpoint pairs produced by STFlow.

We use the raw translation velocity $\hat{\mathbf{v}}_t$ defined in Eq. (3). At t_j, let $K_j = \min\{K, j\}$ denote the number of preceding evaluations retained in the history. Their mean is

$$\bar{\mathbf{v}}_{t_j} := \frac{1}{K_j} \sum_{k=1}^{K_j} \hat{\mathbf{v}}_{t_{j-k}}, \qquad K_j > 0. \tag{9}$$

When $K_j = 0$, the sampler uses $\hat{\mathbf{v}}_{t_j}$ without correction.

For $K_j > 0$, we compare the current raw velocity with the preceding estimate and the historical mean through the quadratic consistency gradient

$$\mathbf{g}_{t_j} = 2\hat{\mathbf{v}}_{t_j} - \hat{\mathbf{v}}_{t_{j-1}} - \bar{\mathbf{v}}_{t_j}, \qquad \widetilde{\mathbf{g}}_{t_j} = \frac{\mathbf{g}_{t_j}}{\max\left(\|\mathbf{g}_{t_j}\|_2, \epsilon\right)}. \tag{10}$$

All averages, norms, and correction magnitudes are evaluated independently for each nucleotide. The constant $\epsilon > 0$ prevents division by zero.

The time factor $a_j = (1+t_j)/2$ gradually increases the correction weight. The parameter α controls the correction strength, while β bounds its magnitude relative to the current velocity:

$$s_j = \min\left\{\alpha a_j \|\hat{\mathbf{v}}_{t_j} - \bar{\mathbf{v}}_{t_j}\|_2, \quad \beta \max\left(\|\hat{\mathbf{v}}_{t_j}\|_2, \epsilon\right)\right\}. \tag{11}$$

The corrected velocity takes a bounded step in the negative consistency-gradient direction and is then used in the Euler update:

$$\widetilde{\mathbf{v}}_{t_j} = \hat{\mathbf{v}}_{t_j} - s_j \widetilde{\mathbf{g}}_{t_j}, \qquad \mathbf{x}_{t_{j+1}} = \mathbf{x}_{t_j} + \Delta t\, \widetilde{\mathbf{v}}_{t_j}. \tag{12}$$

Table 1: Comparison of RNA backbone generation performance across methods and sampling steps. Design., Val., Div., and Nov. denote designability, steric validity, diversity, and novelty, respectively. Designability and steric validity are reported as percentages. Bold values indicate the best designability and steric validity for each method across sampling steps.

Method	Steps	Design. ↑	Val. ↑	Div. ↑	Nov. ↓
RNA-FrameFlow	100	57.6	**56.2**	0.413	0.731
	50	59.0	53.7	0.432	0.732
	20	**60.8**	51.0	0.507	0.729
	10	56.2	45.7	0.573	0.725
STFlow	100	63.9	**56.8**	0.483	0.687
	50	**64.7**	55.2	0.487	0.679
	20	59.3	51.2	0.590	0.613
	10	35.1	50.3	0.720	0.532
ReSTFlow	100	**71.4**	73.5	0.350	0.651
	50	70.7	**75.7**	0.338	0.632
	20	69.8	75.0	0.373	0.572
	10	48.7	70.8	0.423	0.495

When a velocity history is available, the correction is applied before each translational Euler update. Rotations remain unchanged because averaging them requires a manifold-aware construction on $\mathrm{SO}(3)$. During rectification-pair construction, the correction is applied to STFlow; at inference, it is applied to both STFlow and ReSTFlow. After the final Euler update, we reconstruct the RNA backbone from the final frames T_{t_L} and the endpoint torsions $\hat{\Phi}_{L-1}$ predicted at t_{L-1}. Detailed history management and hyperparameter settings are provided in the supplementary material.

4 Experiments

To evaluate ReSTFlow, we compare it against RNA-FrameFlow and STFlow, our first-stage model before rectification training. We evaluate steric validity, designability, local geometric quality, and performance across a range of RNA chain lengths. We further examine sensitivity to the number of sampling steps and conduct ablation studies on the steric-clash loss, trajectory correction, and rectification training.

4.1 Experimental Setup

Dataset. We collect experimentally determined RNA structures from RNAsolo (Adamczyk et al., 2022; 2026), a curated database of RNA structures derived from the Protein Data Bank (Berman et al., 2000), using the data available as of September 25, 2025. We retain RNA chains with a resolution of 4 Å or better and apply additional filters based on chain length and structural quality. The resulting training set contains 7,053 RNA chains ranging from 30 to 200 nt. For evaluation, we adopt an unconditional generation setting, in which each model generates RNA backbones from random noise without conditioning on an input structure. We consider target lengths from 40 to 150 nt in increments of 10 nt and generate 50 backbones at each length.

Baselines. We use RNA-FrameFlow (Anand et al., 2025) as the external baseline for unconditional RNA backbone generation because it employs a comparable nucleotide-level local-frame representation. We also include STFlow, our first-stage model before rectification, to isolate the effect of rectification training. For a controlled comparison, trajectory correction is applied to both STFlow and ReSTFlow during inference unless otherwise specified.

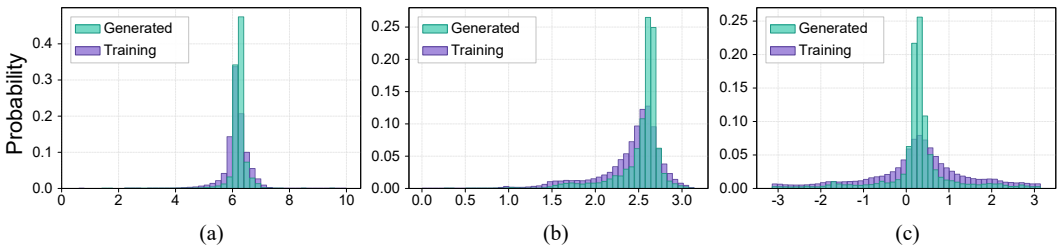


Figure 2: Local geometry evaluation of ReSTFlow-generated RNA backbones. (a) Distribution of distances between adjacent $C4'$ atoms. (b) Distribution of pseudo-bond angles defined by three consecutive $C4'$ atoms. (c) Distribution of pseudo-dihedral angles defined by four consecutive $C4'$ atoms. "Generated" and "Training" denote ReSTFlow-generated backbones and training-set structures, respectively. Greater distributional overlap indicates closer agreement in local $C4'$ geometry.

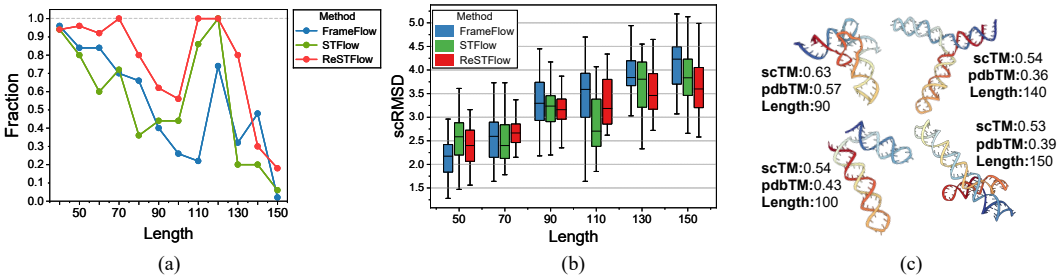


Figure 3: Generation quality by RNA backbone length. (a) Fraction of sterically valid samples at each target length. (b) scRMSD distributions of sterically valid samples after inverse folding and structure prediction; lower values indicate better self-consistency. (c) Representative ReSTFlow backbones annotated with scTM, pdbTM, and length. Higher scTM indicates stronger self-consistency, whereas lower pdbTM indicates lower similarity to the closest training-set structure.

Implementation Details. We first train STFlow on the filtered RNAsolo training set. Using the trained STFlow model, we generate candidate noise–backbone pairs with trajectory correction for RNA lengths ranging from 40 to 150 nt. After steric-clash filtering, we retain 7,850 noise–backbone pairs for rectification training. ReSTFlow is initialized from the trained STFlow checkpoint and further trained on these pairs. Trajectory correction introduces no additional trainable parameters. STFlow and ReSTFlow are optimized with AdamW using a learning rate of 10^{-4} on a single NVIDIA A100 40 GB GPU. Further implementation details are provided in Appendix C.

4.2 Evaluation Metrics

We evaluate generated RNA backbones in terms of steric validity, designability, diversity, and novelty. Steric validity is evaluated over all generated samples. Unless stated otherwise, the remaining metrics are evaluated only on sterically valid backbones, ensuring that downstream sequence-design and structural analyses are restricted to geometrically plausible structures.

Steric Validity. We measure steric validity using the clash-free rate, which captures local steric plausibility. Van der Waals radii are assigned according to atom type, and steric clashes are detected between nonbonded atom pairs after excluding covalently connected atoms. A backbone is considered sterically valid if it contains no detected clashes. The steric-validity rate is the fraction of clash-free backbones among all generated samples.

Designability. For each sterically valid backbone, gRNAde (Joshi et al., 2025) generates eight candidate sequences, whose structures are predicted using RhoFold+ (Shen et al., 2024). Based on $C4'$ atoms, we compute the self-consistency TM-score (scTM) and RMSD (scRMSD) between each prediction and the generated backbone, defining the sample-level scTM as the maximum over the eight candidates. Following prior work (Anand et al., 2025), a backbone is considered designable if

its sample-level scTM is at least 0.45, using the TM-score and nucleic-acid alignment definitions of Zhang & Skolnick (2004) and Zhang et al. (2022). Designability is reported among sterically valid backbones, whereas overall designable yield is the fraction of all generated backbones that are both sterically valid and designable.

Diversity. We use qTMclust from the US-align toolkit (Zhang et al., 2022) to cluster sterically valid backbones based on structural similarity, with a TM-score threshold of 0.45. Backbones assigned to the same cluster are regarded as sharing a similar fold. Diversity is defined as the number of structural clusters divided by the number of sterically valid backbones, with higher values indicating greater structural diversity.

Novelty. We evaluate novelty by comparing each sterically valid backbone against the training structures using US-align (Zhang et al., 2022) and $C4'$ atoms. For each generated backbone, we record its maximum TM-score to the training set and report the mean of these nearest-neighbor similarities. Lower values therefore indicate greater structural novelty. This nearest-neighbor similarity metric is denoted as pdbTM in Figure 3.

4.3 RNA Backbone Generation Comparison

Generation Quality. For each method–step setting, we generate 600 backbones, with 50 samples at each of 12 target lengths ranging from 40 to 150 nt. Table 1 compares performance across methods and sampling budgets. At 20 steps, ReSTFlow achieves 75.0% steric validity, and 69.8% of its sterically valid backbones are designable, compared with 51.0% and 60.8% for RNA-FrameFlow, respectively. This corresponds to overall designable yields of 52.4% for ReSTFlow and 31.0% for RNA-FrameFlow. Notably, ReSTFlow already surpasses RNA-FrameFlow at 100 steps in overall designable yield, despite using only 20 steps. It also exhibits lower training-set similarity while maintaining competitive diversity.

Local Geometry Evaluation. We evaluate local backbone geometry using the $C4'$-based descriptors adopted in RNA-FrameFlow (Anand et al., 2025). Specifically, we compare generated samples with training-set structures in terms of distances between adjacent $C4'$ atoms, pseudo-bond angles defined by three consecutive $C4'$ atoms, and pseudo-dihedral angles defined by four consecutive $C4'$ atoms. As shown in Figure 2, ReSTFlow recovers the principal modes of all three training-set distributions. The adjacent-distance and pseudo-bond-angle distributions show particularly close agreement in their dominant regions, whereas the generated pseudo-dihedral distribution is more concentrated around its central mode. These results complement the global self-consistency evaluation by characterizing local $C4'$ geometry.

Performance Across Backbone Lengths. We conduct a length-resolved evaluation of steric validity and structural self-consistency. Figures 3(a) and 3(b) show that ReSTFlow exhibits the strongest overall steric-validity profile across the evaluated range, with particularly notable results at target lengths of 70, 110, and 120 nt. It also maintains favorable scRMSD distributions over the evaluated lengths. Figure 3(c) further presents representative backbones that are sterically valid, structurally self-consistent, and distinct from their closest training-set matches. Across lengths, ReSTFlow consistently produces sterically valid, self-consistent, and structurally distinct backbones.

4.4 Ablation Study

We conduct ablation studies to examine the contributions of the clash loss, trajectory correction, data filtering, and rectification training. Table 2 compares the corresponding variants at 20 sampling steps, while Figure 4 evaluates trajectory correction across different sampling budgets.

Component Contributions. Table 2 identifies rectification training on STFlow-generated pairs as the main source of performance improvement. Without clash-based pair filtering, rectification increases steric validity from 51.2% to 67.3% and designability from 59.3% to 67.8%. Filtering the training pairs further raises these values to 75.0% and 69.8%. The clash loss and trajectory correction also contribute substantially to ReSTFlow's performance. Clash-aware supervision suppresses atomic overlaps during training, whereas trajectory correction stabilizes coarse-step integration.

Table 2: Ablation study of ReSTFlow at 20 sampling steps. The "w/o data filtering" variant is trained on an equal-sized set of STFlow-generated pairs without clash-based filtering. All metrics follow the definitions in Table 1.

Variant	Design. ↑	Val. ↑	Div. ↑	Nov. ↓
Full ReSTFlow	69.8	75.0	0.373	0.572
w/o traj. corr.	68.0	71.2	0.358	0.601
w/o clash loss	63.8	69.0	0.357	0.625
w/o data filt.	67.8	67.3	0.380	0.571
w/o rect. stage	59.3	51.2	0.590	0.613

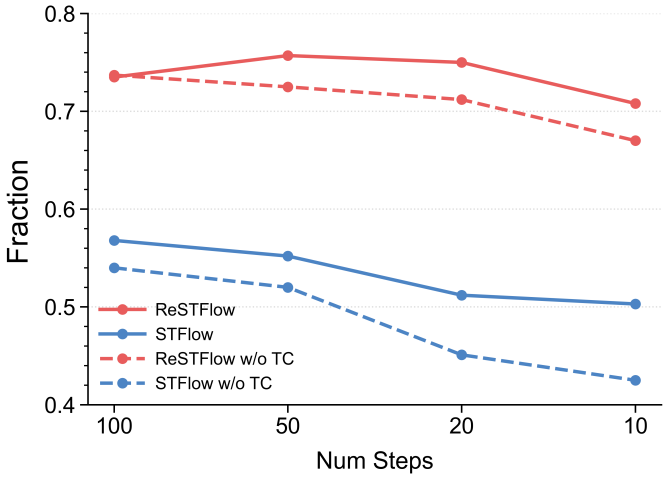


Figure 4: Effect of trajectory correction across different numbers of sampling steps. Solid lines denote sampling with trajectory correction, whereas dashed lines denote sampling without it.

Trajectory Correction Across Sampling Steps. Figure 4 further examines the effect of trajectory correction at 10, 20, 50, and 100 sampling steps. For both STFlow and ReSTFlow, the benefit of trajectory correction is most evident under coarse sampling. The corrected variants retain steric validity more effectively as the step count is reduced, whereas their uncorrected counterparts exhibit a sharper decline. These results demonstrate that trajectory correction is particularly effective at stabilizing coarse translational updates and preserving steric validity in the few-step regime.

5 Conclusion

We presented ReSTFlow, a rectified flow framework that integrates clash-aware supervision, rectification training, and inference-time trajectory correction for few-step RNA backbone generation. Our evaluation covers backbones ranging from 40 to 150 nt, a biologically relevant range encompassing many compact and medium-sized functional RNAs. Within this range, ReSTFlow achieved strong steric validity and designability under limited sampling budgets. These results show that sterically valid and designable RNA backbones can be generated efficiently even under coarse numerical integration, highlighting the effectiveness of ReSTFlow in the few-step regime. Because high-quality structural data for substantially longer RNAs remain limited, performance on longer chains has not yet been systematically evaluated. Moreover, because designability is assessed through an inverse-folding and structure-prediction pipeline, the reported values partly depend on the performance of the downstream models. Future work will evaluate ReSTFlow on longer RNA chains, incorporate richer chemical and stereochemical constraints, extend the framework to conditional generation, and experimentally assess the utility of generated backbones for functional RNA design.

AI USE STATEMENT

We used generative AI tools to assist with LaTeX formatting and to refine word choice in selected sentences for clarity and readability. We also used generative AI tools to assist with code review, including identifying potential implementation issues and checking the supplementary code for potentially identifying information. All AI-assisted changes were reviewed by the authors to ensure that the original scientific meaning was preserved. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the mathematical formulation and training and sampling algorithms in Appendix B. Dataset preparation, model architecture, loss functions, training settings, checkpoint selection, and evaluation protocols are documented in Appendix C. Additional ablation studies, hyperparameter sensitivity analyses, and runtime measurements are reported in Appendix D.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, 2024. doi: 10.1038/s41586-024-07487-w.

Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. RNAsolo: A repository of cleaned PDB-derived RNA 3d structures. *Bioinformatics*, 38(14):3668–3670, 2022. doi: 10.1093/bioinformatics/btac386.

Bartosz Adamczyk, Pawel Boinski, Marta Szachniuk, and Maciej Antczak. RNAsolo 2.0: Multimodal database to study RNAs, their structural families and intermolecular interfaces. *Journal of Molecular Biology*, 438(18):169570, 2026. doi: 10.1016/j.jmb.2025.169570.

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

Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using RoseTTAFoldNA. *Nature Methods*, 21:117–121, 2024. doi: 10.1038/s41592-023-02086-5.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235.

A. Bondi. Van der waals volumes and radii. *The Journal of Physical Chemistry*, 68(3):441–451, 1964. doi: 10.1021/j100785a001.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

Anke Gelbin, Bohdan Schneider, Lester Clowney, Shu-Hsin Hsieh, Wilma K. Olson, and Helen M. Berman. Geometric parameters in nucleic acids: Sugar and phosphate constituents. *Journal of the American Chemical Society*, 118(3):519–529, 1996. doi: 10.1021/ja9528846.

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

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18957–18973. PMLR, 2023. URL https://proceedings.mlr.press/v202/lee23j.html.

Yang Li, Chengxin Zhang, Chenjie Feng, Robin Pearce, P. Lydia Freddolino, and Yang Zhang. Integrating end-to-end learning with deep geometrical potentials for ab initio RNA structure prediction. *Nature Communications*, 14:5745, 2023. doi: 10.1038/s41467-023-41303-9.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022. URL https://arxiv.org/abs/2209.14577.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Runze Ma, Zhongyue Zhang, Zichen Wang, Chenqing Hua, Jiahua Rao, Zhuomin Zhou, and Shuangjia Zheng. RiboFlow: Conditional de novo RNA co-design via synergistic flow matching. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=brV7U7svgS.

Alex Morehead, Jeffrey A. Ruffolo, Aadyot Bhatnagar, and Ali Madani. Towards joint sequence–structure generation of nucleic acid and protein complexes with SE(3)-discrete diffusion, 2024. URL https://arxiv.org/abs/2401.06151.

Divya Nori and Wengong Jin. RNAFlow: RNA structure & sequence design via inverse folding-based flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38395–38408. PMLR, 2024. URL https://proceedings.mlr.press/v235/nori24a.html.

Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, et al. Accurate RNA 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, 21:2287–2298, 2024. doi: 10.1038/s41592-024-02487-0.

Joan Solà. Quaternion kinematics for the error-state Kalman filter, 2017. URL https://arxiv.org/abs/1711.02508.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Quentin Vicens and Jeffrey S. Kieft. Thoughts on how to think (and talk) about RNA structure. *Proceedings of the National Academy of Sciences*, 119(17):e2112677119, 2022. doi: 10.1073/pnas.2112677119.

Chenru Wang, Beier Zhu, and Chi Zhang. Free lunch for stabilizing rectified flow inversion. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.11850.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching, 2023a. URL https://arxiv.org/abs/2310.05297.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40001–40039. PMLR, 2023b. URL https://proceedings.mlr.press/v202/yim23a.html.

Angxiao Yue, Zichong Wang, and Hongteng Xu. ReQFlow: Rectified quaternion flow for efficient and high-quality protein backbone generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73828–73850. PMLR, 2025. URL https://proceedings.mlr.press/v267/yue25c.html.

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

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004. doi: 10.1002/prot.20264.

A Supplementary Overview

This supplement is organized into four substantive parts. Section B specifies the method and algorithms. Section C documents implementation and evaluation protocols. Section D reports additional experimental results, and Section E presents qualitative examples.

STFlow denotes the first-stage sterically regularized model, whereas *ReSTFlow* denotes its rectified counterpart. We count L sampling steps as L Euler updates with a uniform step size $\Delta t = 1/L$. Steric validity is the clash-free fraction over all samples. Designability is the fraction of sterically valid samples with $\mathrm{scTM} \geq 0.45$. For each sterically valid backbone, pdbTM is its maximum TM-score against the training set; a backbone is classified as novel when $\mathrm{pdbTM} < 0.45$.

B Method Details

B.1 RNA Frame and Quaternion Conventions

Each nucleotide i is represented by a local frame $T_i = (\mathbf{x}_i, R_i) \in \mathrm{SE}(3)$ and eight torsion angles Φ_i. The frame origin is placed at $C4'_i$, and the directions toward $O4'_i$ and $C3'_i$ determine its orientation. The eight torsions place $C2'$, $C1'$, $N9/N1$, $O3'$, $O5'$, P, $OP1$, and $OP2$ in this order; fixed residue templates place the remaining modeled atoms (Anand et al., 2025).

During flow matching, R_i is represented by a scalar-first unit quaternion $q_i = (w_i, x_i, y_i, z_i) \in \mathbb{S}^3$. We use the Hamilton product and right-multiply relative rotations, so vectors returned by $2\,\mathrm{Log}(q_t^{-1} \otimes q_1)$ are expressed in the current local frame. Before interpolation or relative-rotation evaluation, the target quaternion is sign-aligned with the current quaternion, $q_1 \leftarrow \mathrm{sign}(q_t^\top q_1) q_1$, to select the shortest representative on $\mathrm{SO}(3)$.

We use N for RNA length, $T_i = (\mathbf{x}_i, q_i)$ for a nucleotide frame, and Φ_i for the eight predicted torsions. The sampling budget is L, the history length is K, and α and β control the correction magnitude.

B.2 Rotation Scheduler

For an aligned pair (q_0, q_1), the training path is

$$q_t = q_0 \otimes \mathrm{Exp}\big(t\,\mathrm{Log}(q_0^{-1} \otimes q_1)\big)\,. \tag{13}$$

For rotational sampling, we use the exponential schedule $\kappa(t) = 1 - e^{-\gamma t}$, as in ReQFlow (Yue et al., 2025). For a fixed endpoint, the scheduled path and its local angular velocity are

$$\begin{aligned} q_t &= q_0 \otimes \mathrm{Exp}\big(\kappa(t)\,\mathrm{Log}(q_0^{-1} \otimes q_1)\big)\,, \\ \omega_t &= \gamma e^{-\gamma t}\omega, \end{aligned} \tag{14}$$

where $\omega = 2\,\mathrm{Log}(q_0^{-1} \otimes q_1)$. This identity follows directly from $\dot\kappa(t) = \gamma e^{-\gamma t}$ and the quaternion kinematic equation.

The corresponding Euler update is

$$q_{t+\Delta t} = q_t \otimes \mathrm{Exp}\left(\frac{1}{2}\Delta t\, \gamma e^{-\gamma t}\hat{\omega}_t\right), \tag{15}$$

where $\hat{\omega}_t$ is the predicted angular velocity. We set $\gamma = 10$.

B.3 Rectification on RNA Frame Space

Let π_{ret} denote the coupling obtained by sampling initial frames, generating endpoints with trajectory-corrected STFlow, and retaining only clash-free endpoints. Rectification training uses samples from π_{ret} rather than independently paired noise and RNAsolo structures. We equip $\mathcal{M} = \mathbb{R}^{3N} \times \mathrm{SO}(3)^N$ with the product metric

$$\|U\|^2_{g_\lambda} = \sum_{i=1}^{N} \|\mathbf{v}_i\|_2^2 + \lambda_R \sum_{i=1}^{N} \|\omega_i\|_2^2, \qquad \lambda_R > 0, \tag{16}$$

where $U = (\mathbf{v}_1, \omega_1, \ldots, \mathbf{v}_N, \omega_N)$ is a tangent vector and ω_i is represented in the local tangent coordinates defined above.

Marginal preservation for the retained coupling. Let Z_t be the translation–SLERP interpolation of $(Z_0, Z_1) \sim \pi_{\mathrm{ret}}$, with tangent velocity U_t, and define the population rectified field $u^*(z,t) = \mathbb{E}[U_t \mid Z_t = z]$. Assume finite kinetic action and sufficient regularity for the continuity equation driven by u^* to have a unique distributional solution. If $Z'_0 \sim \mathrm{Law}(Z_0)$ and $\dot{Z}'_t = u^*(Z'_t, t)$, then

$$\mathrm{Law}(Z'_t) = \mathrm{Law}(Z_t), \qquad t \in [0,1]. \tag{17}$$

To see this, apply the chain rule to any smooth test function h and condition on Z_t:

$$\frac{d}{dt}\mathbb{E}[h(Z_t)] = \mathbb{E}[\langle \nabla h(Z_t), U_t\rangle] = \mathbb{E}[\langle \nabla h(Z_t), u^*(Z_t,t)\rangle]. \tag{18}$$

Thus Z_t and Z'_t satisfy the same weak continuity equation with the same initial law. In particular, rectification preserves the two endpoint marginals of π_{ret}.

Convex transport cost for the retained coupling. For the quaternion component, we follow the formulation of ReQFlow (Yue et al., 2025). For nucleotide i and any convex function $c : \mathbb{R}^3 \to \mathbb{R}$, define

$$C_i(q_{0,i}, q_{1,i}) = c\big(\mathrm{Log}(q_{0,i}^{-1} \otimes q_{1,i})\big)\,. \tag{19}$$

Under the same assumptions as rectified quaternion flow, the induced coupling π'_{ret} satisfies

$$\mathbb{E}_{\pi'_{\mathrm{ret}}}\left[\sum_{i=1}^{N} C_i(q'_{0,i}, q'_{1,i})\right] \le \mathbb{E}_{\pi_{\mathrm{ret}}}\left[\sum_{i=1}^{N} C_i(q_{0,i}, q_{1,i})\right]. \tag{20}$$

The translation component has the analogous lower-or-equal convex-cost property in $\mathbb{R}^{3N}$ (Liu, 2022). Thus, rectification preserves or reduces these transport costs in the idealized population setting. This result does not guarantee a strict cost reduction for a finite trained network.

Boundary introduced by filtering. The preserved marginals are those of π_{ret}. Because retention depends on the generated endpoint, the retained initial marginal need not equal the unconditioned prior p_0, and the retained endpoint marginal is a filtered model distribution rather than p_{data}. We therefore use the proposition to motivate straighter transport under the retained coupling, not to claim that clash filtering preserves the original prior or RNAsolo data marginal.

B.4 Trajectory Correction

At time t_j, ReSTFlow maintains up to K preceding raw translational velocity estimates. For nonuniform steps, we use the step-duration-weighted mean

$$\bar{\mathbf{v}}_{t_j} = \frac{\sum_{k=1}^{K_j} \Delta t_{j-k}\hat{\mathbf{v}}_{t_{j-k}}}{\sum_{k=1}^{K_j} \Delta t_{j-k}}, \qquad K_j = \min(K, j). \tag{21}$$

The uniform grid used here reduces this expression to the arithmetic mean. The correction direction is obtained from the per-nucleotide quadratic consistency objective

$$J_j(\mathbf{v}) = \frac{1}{2}\|\mathbf{v} - \hat{\mathbf{v}}_{t_{j-1}}\|_2^2 + \frac{1}{2}\|\mathbf{v} - \bar{\mathbf{v}}_{t_j}\|_2^2, \tag{22}$$

whose gradient at the current raw velocity is

$$\mathbf{g}_j = 2\hat{\mathbf{v}}_{t_j} - \hat{\mathbf{v}}_{t_{j-1}} - \bar{\mathbf{v}}_{t_j}, \qquad \widetilde{\mathbf{g}}_j = \frac{\mathbf{g}_j}{\max(\|\mathbf{g}_j\|_2, \epsilon)}. \tag{23}$$

With $a_j = (1 + t_j)/2$, the bounded scale and corrected velocity are

$$s_j = \min\left\{\alpha a_j \|\hat{\mathbf{v}}_{t_j} - \bar{\mathbf{v}}_{t_j}\|_2,\ \beta \max(\|\hat{\mathbf{v}}_{t_j}\|_2, \epsilon)\right\}, \tag{24}$$

$$\widetilde{\mathbf{v}}_{t_j} = \hat{\mathbf{v}}_{t_j} - s_j \widetilde{\mathbf{g}}_j. \tag{25}$$

All means, norms, and scales are computed independently for each nucleotide. We use $K = 10$, $\alpha = 0.30$, $\beta = 1.5$, and $\epsilon = 10^{-8}$. The first evaluation has an empty history and therefore uses the raw velocity unchanged. Only raw, rather than corrected, velocities enter the subsequent history.

B.5 Training and Sampling Algorithms

Algorithm 1 STFlow training

Require: Filtered RNAsolo training set $\mathcal{D}$, model M_θ
1: **for** each training minibatch T_1, Φ_1 **do**
2: Sample $t \sim \mathcal{U}[t_{\min}, 1 - t_{\min}]$ and $T_0 \sim p_0$
3: Interpolate translations linearly and rotations by quaternion SLERP
4: Predict endpoint frames and torsions with $M_\theta(T_t, t)$
5: Compute translation, rotation, coordinate, distance, torsion, and clash losses
6: Activate auxiliary losses only when $t > \tau_{\text{aux}}$
7: Update θ using the optimizer in Table 5
8: **end for**
Ensure: Trained STFlow parameters

Algorithm 2 Construction of rectification pairs

Require: Trained STFlow model, target lengths, candidate count
1: **for** each sampled initial frame set $T_0' \sim p_0$ **do**
2: Generate endpoint frames and torsions with Algorithm 4
3: Reconstruct all modeled backbone atoms
4: Evaluate the exact steric-clash criterion in Section C.2
5: **if** the generated backbone is clash-free **then**
6: Retain (T_0', T_1', Φ_1')
7: **end if**
8: **end for**
Ensure: Retained model-induced endpoint pairs

Algorithm 3 ReSTFlow rectification training

Require: Trained STFlow parameters and retained endpoint pairs
1: Initialize ReSTFlow from the selected STFlow checkpoint
2: **for** each rectification-training minibatch (T_0', T_1', Φ_1') **do**
3: Sample t and interpolate between the paired endpoints
4: Predict endpoint frames and torsions
5: Compute the same objective used for STFlow
6: Update the model parameters
7: **end for**
Ensure: Trained ReSTFlow parameters

Algorithm 4 Trajectory-corrected inference

Require: Model M_θ, initial frames T_0, and sampling steps L

Set $\Delta t = 1/L$, $t_j = j\Delta t$, and $\mathcal{H} = \varnothing$
for $j = 0, \ldots, L-1$ **do**
  Predict $(\hat{\mathbf{x}}_{1,j}, \hat{q}_{1,j}, \hat{\Phi}_j) = M_\theta(T_{t_j}, t_j)$
  Set $\hat{\mathbf{v}}_j = (\hat{\mathbf{x}}_{1,j} - \mathbf{x}_{t_j})/(1-t_j)$
  Set $\hat{\omega}_j = 2\,\mathrm{Log}(q_{t_j}^{-1} \otimes \hat{q}_{1,j})/(1-t_j)$
  if $\mathcal{H} \neq \varnothing$ **then**
    Compute the step-weighted history mean $\bar{\mathbf{v}}_j$
    Compute $\widetilde{\mathbf{g}}_j$, s_j, and $\widetilde{\mathbf{v}}_j = \hat{\mathbf{v}}_j - s_j\widetilde{\mathbf{g}}_j$
  else
    Set $\widetilde{\mathbf{v}}_j = \hat{\mathbf{v}}_j$
  end if
  $\mathbf{x}_{t_{j+1}} \leftarrow \mathbf{x}_{t_j} + \Delta t\,\widetilde{\mathbf{v}}_j$
  Update $q_{t_{j+1}}$ with Eq. (15)
  Append $(\mathrm{stopgrad}(\hat{\mathbf{v}}_j), \Delta t)$ to $\mathcal{H}$; retain the latest $K = 10$
end for
Reconstruct the RNA backbone from T_{t_L} and the endpoint torsions $\hat{\Phi}_{L-1}$

Ensure: Generated RNA backbone

Thus, L sampling steps comprise L Euler updates with step size $\Delta t = 1/L$. Rectification-pair generation uses the same trajectory-corrected sampler employed to evaluate STFlow and ReSTFlow.

C Implementation and Evaluation

C.1 Dataset Preparation and Rectification Pairs

RNAsolo data preparation. The training structures were collected from RNAsolo (Adamczyk et al., 2022) using the database snapshot available on September 25, 2025. We retain homomer entries with experimental resolution below 4 Å and modeled lengths from 30 to 200 nucleotides. The resulting training set contains 7,053 RNA chains.

The executable preprocessing pipeline crops each entry to the interval between its first and last modeled nucleotide. Frames and torsions are then derived only for modeled residues. Alternate locations, modified nucleotides, missing atoms, chain breaks, multiple structural models, and duplicate entries receive no additional manual filtering beyond this pipeline. Following the preprocessing protocol of RNA-FrameFlow, we retain the filtered RNAsolo structures for training.

Following common practice in generative structure modeling, we used a fixed monitoring subset sampled from the filtered training metadata to track optimization progress and analyze checkpoint behavior.

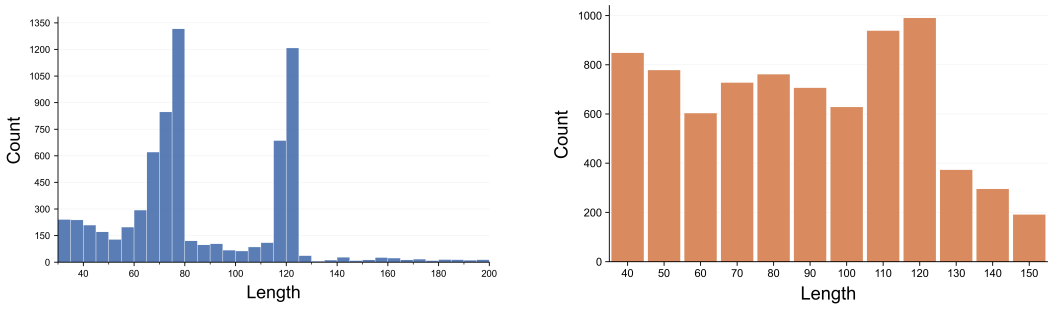


Figure 5: Length distributions of the two training stages. The left panel shows filtered RNAsolo structures used to train STFlow. The right panel shows clash-free noise–backbone pairs used to train ReSTFlow.

Rectification-pair dataset. Candidate pairs are generated by trajectory-corrected STFlow for lengths from 40 to 150 nt at intervals of 10 nt using $L = 50$ sampling steps. After applying the same clash-free criterion used for steric-validity evaluation, 7,850 noise–backbone pairs are retained for rectification training. These generated pairs are used to construct model-induced couplings for rectification training rather than serving as additional structural labels.

Candidate generation is intentionally nonuniform across lengths. We draw more candidates between 80 and 150 nt to compensate for the lower steric-validity yield in this range. All qualifying pairs are retained without coordinate- or similarity-based endpoint deduplication. The retained length distribution is reported in the right panel of Figure 5.

We perform one rectification round, consistent with the single-stage procedure used in ReQFlow (Yue et al., 2025). Additional rectification rounds produced limited gains in preliminary experiments and were therefore not used for the reported ReSTFlow model.

C.2 Architecture and Loss Functions

Model architecture. STFlow and ReSTFlow share the same endpoint-prediction network, so rectification introduces no additional trainable parameters. The network uses a six-block invariant-point-attention (IPA) trunk with interleaved sequence Transformer and equivariant frame updates, followed by an endpoint torsion head. Table 3 summarizes the method-defining settings.

Table 3: Core architecture settings of the shared endpoint-prediction network.

Hyperparameter	Value
Node / pair embedding dimensions	256 / 128
IPA blocks / attention heads	6 / 8
IPA query/key/value points	8 / 8 / 12
Transformer layers per block / heads	2 / 4
Pair distance bins	22
Translation self-conditioning	Enabled
Torsion head (input/hidden, blocks, angles)	256 / 128, 2, 8

Loss functions and steric-clash implementation. Following RNA-FrameFlow (Anand et al., 2025), we retain three endpoint-level geometric objectives that supervise selected atom coordinates, pairwise atomic distances, and torsion angles. Let $\mathcal{S} = \{C3', C4', O4'\}$ denote the supervised atom set. The inherited objectives are

$$\begin{aligned}
\mathcal{L}_{\text{bb}} &= \frac{1}{N|\mathcal{S}|} \sum_{i=1}^{N} \sum_{a \in \mathcal{S}} \|\hat{\mathbf{x}}_{i,a} - \mathbf{x}_{i,a}\|_2^2 , \\
\mathcal{L}_{\text{dist}} &= \frac{1}{Z} \sum_{((i,a),(j,b)) \in \mathcal{P}} \left(\hat{d}_{ij}^{ab} - d_{ij}^{ab} \right)^2 , \\
\mathcal{L}_{\text{tors}} &= \frac{1}{8N} \sum_{i=1}^{N} \sum_{k=1}^{8} \|\hat{\mathbf{u}}_{i,k} - \mathbf{u}_{i,k}\|_2^2 .
\end{aligned} \tag{26}$$

Here, $d_{ij}^{ab} = \|\mathbf{x}_{i,a} - \mathbf{x}_{j,b}\|_2$ and $\hat{d}_{ij}^{ab} = \|\hat{\mathbf{x}}_{i,a} - \hat{\mathbf{x}}_{j,b}\|_2$. The set $\mathcal{P}$ contains valid ordered atom pairs after atom masking and self-pair exclusion, and $Z = |\mathcal{P}|$. For torsion k of nucleotide i, $\mathbf{u}_{i,k} = (\sin \phi_{i,k}, \cos \phi_{i,k})$ and $\hat{\mathbf{u}}_{i,k}$ is the normalized prediction. These objectives are combined with the steric-clash term in the auxiliary endpoint loss defined in the main paper.

Training and steric-validity evaluation use the same 23-atom RNA representation but distinct steric-overlap criteria. In both cases, atoms absent from the corresponding nucleotide type are removed using the nucleotide-specific atom-existence mask. Element-specific van der Waals radii are 1.70 Å for C, 1.55 Å for N, 1.52 Å for O, and 1.80 Å for P and S; an unrecognized element defaults to 1.80 Å.

Training criterion. The final training configuration enables a differentiable flat-bottom structural-violation loss with weight $\lambda_{\text{clash}} = 0.05$, following the AlphaFold-style violation formulation used by related RNA flow models. For atoms a and b belonging to different nucleotides, the soft overlap at the predicted endpoint is

$$\begin{aligned} e_{ij}^{ab} &= \left[r_a + r_b - \delta_{\text{train}} - \hat{d}_{ij}^{ab} \right]_+, \\ \delta_{\text{train}} &= 1.5 \text{ Å}. \end{aligned} \tag{27}$$

Let $\mathcal{A}$ denote the reconstructed atom slots and $m_{i,a}$ the combined valid-atom mask. The implementation first accumulates overlap penalties at each atom and then applies

$$\begin{aligned} \mathcal{L}_{\text{clash}} &= \frac{1}{Z_c} \sum_{(i,a)\in\mathcal{A}} m_{i,a} \left(c_{i,a}^{\text{inter}} + c_{i,a}^{\text{intra}} \right), \\ Z_c &= \sum_{(i,a)\in\mathcal{A}} m_{i,a} + \epsilon_c, \qquad \epsilon_c = 10^{-6}. \end{aligned} \tag{28}$$

For internucleotide clashes, only residue pairs $i < j$ are evaluated. Each admissible atom-pair penalty e_{ij}^{ab} is computed once and added to the per-atom accumulators of both incident atoms. The covalently bonded $O3'_i$–P_{i+1} pair is excluded when the nucleotides are consecutive in the same chain. For intranucleotide clashes, the atom-pair mask retains both off-diagonal orders, (a, b) and (b, a). Row- and column-wise sums form $c_{i,a}^{\text{intra}}$, matching the implemented AlphaFold-style reduction. Template-derived lower and upper bounds restrict this term to nonbonded atom pairs.

The inter- and intranucleotide per-atom tensors are added before the shared valid-atom normalization. The loss therefore measures the mean accumulated overlap per valid atom, rather than a mean over all candidate pairs. When the average clash burden per atom is fixed, both the numerator and denominator scale proportionally with chain length. Connection losses and the broader within-nucleotide geometry penalty are computed by the violation routine but disabled in the optimized objective. This term serves as an auxiliary optimization regularizer: it penalizes severe atomic overlap without imposing excessive costs for small coordinate deviations.

The weighted structural term is clamped at 1.0 before it is added to the coordinate, distance, and torsion terms. The complete auxiliary loss is active only when $t > \tau_{\text{aux}} = 0.25$ and is finally clamped at 5.0. Thus, the configured weight 0.05 applies inside the auxiliary term rather than to the full flow-matching objective.

Evaluation criterion. Steric validity is assessed using the stricter hard-clash criterion of the RNA-FrameFlow baseline protocol (Anand et al., 2025). Evaluation considers only noncovalent atom pairs belonging to different nucleotides; within-nucleotide pairs are not checked. The covalently bonded $O3'_i$–P_{i+1} pair is excluded for adjacent nucleotides. A remaining atom pair is classified as clashing when

$$d_{ij}^{ab} < r_a + r_b - \delta_{\text{eval}}, \qquad \delta_{\text{eval}} = 0.6 \text{ Å}. \tag{29}$$

A generated structure is counted as clash-free only when no evaluated pair satisfies this condition. The training loss is therefore not a differentiable implementation of the reported evaluation metric: the former supplies a smooth optimization signal, whereas the latter defines the binary clash-free rate used for reporting and baseline comparison.

C.3 Training and Checkpointing

Both stages use AdamW with learning rate 10^{-4} and are trained on one NVIDIA A100 40 GB GPU using PyTorch 2.0.0 and CUDA 11.7. The sampler caps a minibatch at eight structures and uses a maximum squared-residue budget of 10^6.

ReSTFlow is warm-started with configuration override enabled from the STFlow checkpoint at epoch 25, optimization step 182,832. This checkpoint is also used for the reported STFlow evaluation. The reported ReSTFlow results use the checkpoint at epoch 28, optimization step 206,382. After observing loss convergence, we monitored model checkpoints by evaluating generated samples from fixed random seeds. The final checkpoints were selected based on steric quality and diversity statistics to avoid degenerate generation behavior. For the selected ReSTFlow checkpoint, the mean

Table 4: Core loss and hard-clash evaluation settings. Training uses differentiable clash supervision, whereas reporting uses the strict binary hard-clash criterion.

Loss or control	Final setting
Training settings	
Flow-loss weights (translation/rotation)	2.0 / 1.0
Geometric auxiliary losses	Coordinate, distance, torsion (1.0)
Structural clash loss weight	0.05
Training clash scope	Inter- and within-nucleotide
Training overlap tolerance	1.5 Å
Auxiliary activation threshold	$\tau_{\mathrm{aux}} = 0.25$
Clash-term clamp	1.0
Total auxiliary-loss clamp	5.0
Evaluation settings	
Evaluation overlap tolerance	0.6 Å
Evaluation atom scope	Nucleotide-type-specific inter-nucleotide mask; adjacent $O3'_i$–P_{i+1} excluded

numbers of clashes per 100 checked atoms and per 100 heavy atoms were both 0.998, below the prespecified threshold of one clash per 100 atoms.

Table 5: Stage-specific training and checkpoint settings. Shared optimizer, batching, and hardware settings are given in the text.

Setting	STFlow	ReSTFlow
Initialization	Random	STFlow checkpoint
Training set	7,053 chains	7,850 retained pairs
Rectification start (epoch/step)	–	25 / 182,832
Reported checkpoint (epoch/step)	25 / 182,832	28 / 206,382

C.4 Baselines and Evaluation Protocol

Baseline configuration and fairness. RNA-FrameFlow (Anand et al., 2025) is the external baseline because it uses a comparable nucleotide-level local-frame representation. We evaluate the default pretrained checkpoint and parameters released in its official repository without retraining the model on the ReSTFlow dataset. Consequently, the baseline retains its original training data, and the performance gap should not be attributed solely to the proposed method.

Evaluation pipeline. For each method–step configuration, 50 backbones are generated at every length from 40 to 150 nt, giving 600 samples. Steric validity is evaluated over all samples. Designability, diversity, and novelty are evaluated only over sterically valid samples. We follow the self-consistency strategy of RNA-FrameFlow (Anand et al., 2025). For each backbone, gRNAde (Joshi et al., 2025) proposes eight sequences, and RhoFold+ (Shen et al., 2024) predicts their structures. Structural comparisons use $C4'$ atoms and US-align (Zhang et al., 2022).

Let $\mathcal{G}$ be the 600 generated backbones for one method–step setting and $\mathcal{V} \subseteq \mathcal{G}$ the clash-free subset. Steric validity is $|\mathcal{V}|/|\mathcal{G}|$. For $b \in \mathcal{V}$, let $s_{b,m}$ and $r_{b,m}$ denote the scTM and scRMSD obtained from candidate sequence $m \in \{1, \ldots, 8\}$. We aggregate the candidates as $s_b = \max_m s_{b,m}$ and

Table 6: Baseline provenance and matched evaluation settings.

Method	Checkpoint source	Training data	Scheduler
RNA-FrameFlow	Official repository default	As released	Repository default
STFlow	Epoch 25 Step 182,832	7,053 RNAsolo chains	Matched
ReSTFlow	Epoch 28 Step 206,382	7,053 + 7,850 pairs	Matched

Note. All methods are evaluated with $L \in \{10, 20, 50, 100\}$ sampling steps.

$r_b = \min_m r_{b,m}$. Designability is

$$\frac{1}{|\mathcal{V}|} \sum_{b \in \mathcal{V}} \mathbb{1}\{s_b \geq 0.45\}. \tag{30}$$

For diversity, qTMclust clusters the valid backbones at TM-score threshold 0.45; the score is the number of clusters divided by $|\mathcal{V}|$. For novelty, each $b \in \mathcal{V}$ is aligned to every training structure using $C4'$ atoms. Let p_b be the largest resulting training-set TM-score. A backbone is classified as novel when $p_b < 0.45$, and the novelty rate is $|\{b \in \mathcal{V} : p_b < 0.45\}|/|\mathcal{V}|$. The reported pdbTM score is the mean of p_b over $\mathcal{V}$ and summarizes nearest-neighbor training-set similarity; lower values indicate greater novelty.

C.5 Statistical and Runtime Protocol

Statistical reporting. Model-quality metrics are reported from the selected checkpoints. For the length-resolved clash analysis in Figure 6, uncertainty is quantified using 95% bootstrap confidence intervals.

Runtime measurement. Following the protocol used for the main-paper timing comparison, each method–budget configuration generates 50 backbones of length 100 nt on one NVIDIA A100 40 GB GPU. We report the mean per-backbone sampling time at $L \in \{10, 20, 50, 100\}$. Timing includes model sampling only and excludes gRNAde, RhoFold+, and the remaining self-consistency evaluation.

D Additional Experimental Results

D.1 Sequential Component Ablation

To assess the contribution of each component, we examine the sequential ablation results in Table 7. Among the configurations evaluated with 50 sampling steps, introducing the clash loss increases steric validity from 45.8% to 52.0%, while designability remains comparable at 62.2%. Adding trajectory correction further improves steric validity to 55.2% and designability to 64.7%, while lowering the training-set similarity score from 0.694 to 0.679, despite a slight decrease in diversity from 0.518 to 0.487. With rectification training, the complete ReSTFlow model achieves 75.0% steric validity and 69.8% designability using only 20 sampling steps, exceeding the strongest 50-step configuration by 19.8 and 5.1 percentage points, respectively. It also obtains the lowest training-set similarity score of 0.572, indicating improved novelty, although its lower diversity reveals a trade-off between sample diversity and the proportion of sterically valid and designable structures. Overall, the results show that the clash loss and trajectory correction provide incremental gains, while their combination with rectification training yields the strongest performance in the target few-step setting.

Table 7: Sequential ablation of clash loss, trajectory correction, and rectification training. The first three configurations are evaluated with 50 sampling steps, whereas the complete ReSTFlow model uses 20 steps to reflect its target few-step setting.

Clash	Traj. Corr.	Rect.	Design. ↑	Val. ↑	Div. ↑	Nov. ↓
No	No	No	62.5	45.8	0.477	0.723
Yes	No	No	62.2	52.0	0.518	0.694
Yes	Yes	No	64.7	55.2	0.487	0.679
Yes	Yes	Yes	69.8	75.0	0.373	0.572

D.2 Pair Filtering and Correction Sensitivity

Rectification-pair filtering. Table 8 examines whether filtering the rectification pairs used in the second training stage contributes to ReSTFlow's performance. Both conditions use 7,850 training pairs and share the same initialization and training configuration. The unfiltered condition retains pairs without clash-free screening, whereas the clash-free condition is the standard ReSTFlow configuration. We evaluated both conditions at the same four sampling budgets using the metrics defined in Section C.4.

Table 8: Ablation of rectification-pair filtering in the second training stage. "Unf." retains unfiltered pairs, whereas "CF" denotes the clash-free ReSTFlow training set. Designability and steric validity are percentages; all entries are point estimates.

L	Designability ↑		Steric validity ↑		Diversity ↑		pdbTM ↓	
	Unf.	CF	Unf.	CF	Unf.	CF	Unf.	CF
10	42.7	48.7	65.2	70.8	0.420	0.423	0.515	0.495
20	67.8	69.8	67.3	75.0	0.380	0.373	0.571	0.572
50	71.5	70.7	68.5	75.7	0.350	0.338	0.613	0.632
100	69.1	71.4	67.5	73.5	0.345	0.350	0.623	0.651

Clash-free pair filtering improved steric validity at every tested budget, with absolute gains of 5.6–7.7 percentage points. The designability effect was smaller and budget dependent: filtering improved designability by 6.0, 2.0, and 2.3 percentage points at $L = 10$, 20, and 100, respectively, whereas the unfiltered model was 0.8 percentage points higher at $L = 50$. Diversity changed by at most 0.012 and showed no consistent direction. The pdbTM differences were likewise non-monotonic: filtering reduced pdbTM by 0.020 at $L = 10$, left it nearly unchanged at $L = 20$, and increased it by 0.019 and 0.028 at $L = 50$ and 100. These results support the narrower conclusion that clash-free pair filtering preferentially improves geometric validity, rather than uniformly improving every generation metric.

The 7,850-pair length distribution used for the standard second-stage training is reported alongside the 7,053-chain first-stage RNAsolo distribution in Figure 5.

Trajectory-correction sensitivity. The default configuration $(K, \alpha, \beta) = (10, 0.30, 1.5)$ was selected empirically through controlled comparisons. We first fix $K = 10$ and jointly evaluate the correction-scale pairs formed by $\alpha \in \{0.10, 0.30, 0.50\}$ and $\beta \in \{1.0, 1.5, 2.0\}$. We then fix the selected pair $(\alpha, \beta) = (0.30, 1.5)$ and compare $K \in \{5, 10, 20\}$. The shared default configuration gives 11 unique settings. Each setting uses the complete ReSTFlow checkpoint at $L = 20$ and generates 600 backbones under the matched evaluation protocol.

Table 9: Trajectory-correction sensitivity at $L = 20$. Each setting evaluates 600 backbones. Validity is the clash-free percentage, and diversity is the number of qTMclust clusters per sterically valid backbone. The default is $(K, \alpha, \beta) = (10, 0.30, 1.5)$.

Paired-scale sweep at fixed $K = 10$

K	α	β	Valid. ↑	Div. ↑
10	0.10	1.0	72.0	0.330
10	0.10	1.5	69.3	0.345
10	0.10	2.0	71.8	0.355
10	0.30	1.0	73.0	0.343
10	0.30	1.5	75.0	0.373
10	0.30	2.0	75.8	0.312
10	0.50	1.0	73.7	0.350
10	0.50	1.5	72.8	0.348
10	0.50	2.0	73.3	0.367

History-window sweep at fixed $(\alpha, \beta) = (0.30, 1.5)$

K	α	β	Valid. ↑	Div. ↑
5	0.30	1.5	73.5	0.348
20	0.30	1.5	74.3	0.352

Across the 11 configurations, steric validity ranged from 69.3% to 75.8%, whereas diversity ranged from 0.312 to 0.373. The default configuration achieved 75.0% steric validity and the highest diversity, 0.373. The setting $(K, \alpha, \beta) = (10, 0.30, 2.0)$ produced the highest validity, 75.8%, but the lowest diversity, 0.312. Changing K from 5 to 20 produced only moderate variation in both metrics. The scale sweep showed no monotonic trend across either α or β. These results identify the default as a balanced setting rather than a uniformly optimal one.

D.3 Runtime Analysis

The runtime comparison reports total model-sampling time for RNA-FrameFlow, STFlow, and ReSTFlow across four matched evaluation budgets.

Table 10: Mean per-backbone sampling time in seconds for 50 generated 100-nt RNA backbones on an NVIDIA A100 40 GB GPU. Timing excludes inverse folding, structure prediction, and self-consistency evaluation.

Method	$L = 10$	$L = 20$	$L = 50$	$L = 100$
RNA-FrameFlow	1.01	1.77	4.86	9.49
STFlow	0.58	1.14	2.46	4.87
ReSTFlow	0.55	1.03	2.56	5.16

D.4 Length-Resolved Steric Analysis

Steric severity is summarized using the same hard-clash criterion as the steric-validity evaluation, and only inter-nucleotide clashes are counted. At each target length, the mean number of clashes per 100 checked atoms is calculated over all 50 generated backbones, including samples with zero clashes. The corresponding RNAsolo values at the evaluated lengths are included as a structural reference in Figure 6.

D.5 Self-Consistency and Training-Set Similarity

Figure 7 reports sample-level self-consistency and training-set similarity for 600 backbones generated by ReSTFlow at $L = 50$. The evaluation includes 50 samples at each target length from 40 to 150 nt. For each sample, scTM is the maximum score over eight inverse-folded sequences. The pdbTM value is the maximum TM-score against all 7,053 RNAsolo training structures; samples with $\mathrm{pdbTM} < 0.45$ are classified as novel. Thus, the lower-right region defined by $\mathrm{scTM} \geq 0.45$ and $\mathrm{pdbTM} < 0.45$ contains samples that are both designable and novel. Colors distinguish the target-length groups. The visualization is descriptive and is not used to estimate a correlation between the two metrics.

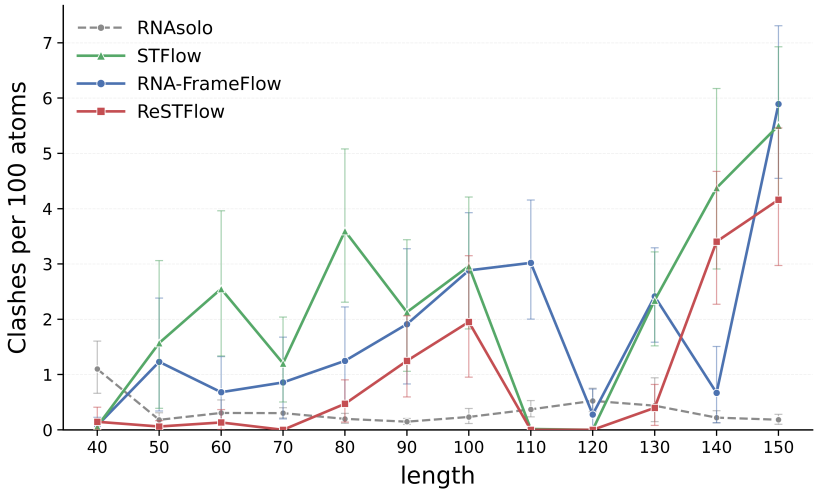


Figure 6: Length-resolved inter-nucleotide steric clashes. Markers show the mean number of clashes per 100 checked atoms for RNAsolo, STFlow, RNA-FrameFlow, and ReSTFlow at each RNA length. Error bars indicate 95% bootstrap confidence intervals. Generated-method results summarize 50 backbones per target length at $L = 20$ and retain all samples, including structures with zero clashes.

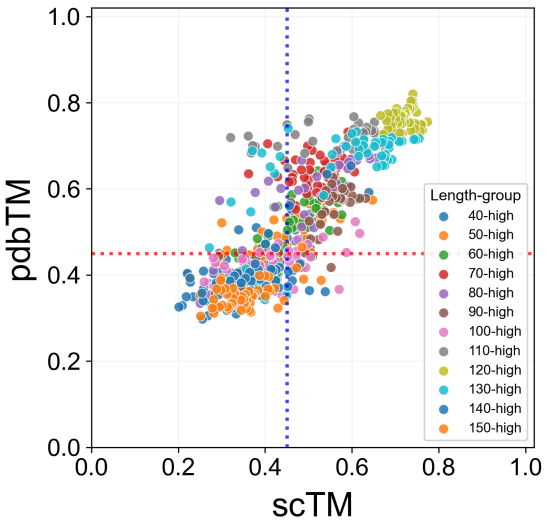
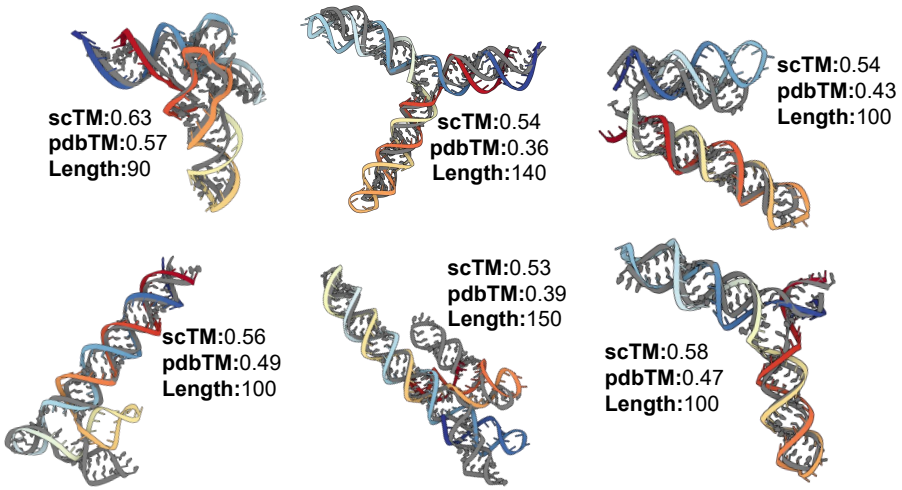


Figure 7: Self-consistency and training-set similarity for 600 ReSTFlow backbones generated at $L = 50$. Each point represents one backbone, and colors distinguish target lengths from 40 to 150 nt. The vertical line at $\mathrm{scTM} = 0.45$ marks the designability threshold. The horizontal line at $\mathrm{pdbTM} = 0.45$ marks the novelty threshold; samples below this line are classified as novel.

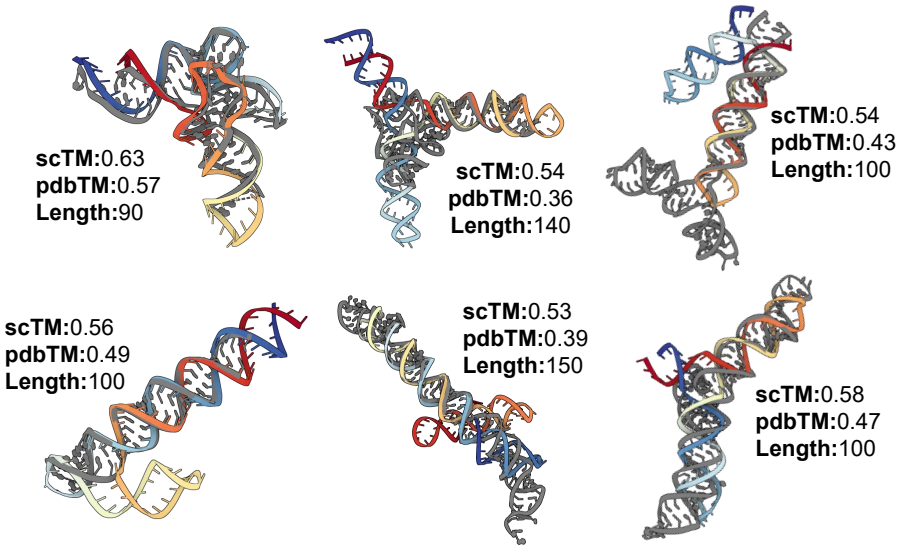
E QUALITATIVE RESULTS

E.1 REPRESENTATIVE GENERATIONS

Figure 8 compares the same six ReSTFlow generations with two structural references. Panel (a) aligns each generated backbone with the corresponding RhoFold+ reconstruction obtained during self-consistency evaluation. Panel (b) aligns the same backbone with its closest RNAsolo training-set structure, identified by the maximum pdbTM score. The examples span 90–150 nt and include structures on both sides of the $\mathrm{pdbTM} = 0.45$ novelty threshold.



(a) Alignment with RhoFold+ reconstructions



(b) Alignment with the closest RNAsolo training-set structures

Figure 8: Representative structural comparisons for six ReSTFlow generations. (a) Each generated backbone is aligned with its corresponding RhoFold+ reconstruction from the self-consistency pipeline. (b) The same backbone is aligned with its closest RNAsolo training-set structure, identified by the maximum pdbTM score. Annotations report RNA length, scTM between the generated backbone and its RhoFold+ reconstruction, and pdbTM against the closest training-set structure.