DTFLOW: DUAL-TRACK FLOW MATCHING FOR GENERATIVE RNA BASE PAIR PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA molecules can adopt multiple base-pairing states, yet most learning-based predictors return a single structure per sequence. We introduce **DTFlow**, a dual-track flow framework that unifies prediction and generation of sparse binary base-pair matrices (BPMs), including non-canonical interactions. A pretrained sequence track and a pair track exchange information through attention at nucleotide-pair resolution. Two time step variables condition this exchange and, together with the input states, gate prediction, conditional BPM generation or auxiliary masked sequence reconstruction within the same architecture. Prediction and generation share a BPM endpoint readout and objective, enabling predictive training to initialise generation from limited structural supervision. We evaluate this framework with **RNA-MultiBP**, which preserves multiple observed BPMs per sequence and excludes target-related sequence groups from training and validation through sequence-, structure- and family-informed clustering. On 56 CASP and RNA-Puzzles sequences, we assess reference coverage and candidate fidelity by comparing predicted and observed base pairs. DTFlow's single-output predictor improves reference coverage over the RiNALMo baseline by 11.1 percentage points, while approaching RNAfold and ProbKnot. With 16 candidates, its generative model improves coverage over retrained RNADiffFold by 9.6 percentage points at the same candidate count. On the 26 multi-reference sequences, generating 16 candidates improves coverage over the single predictive output by 3.1 percentage points. Conditioning trRosettaRNA2 on DTFlow samples brings predictions of a riboswitch closer to deposited conformers than conditioning on the deposited BPMs themselves. These results demonstrate competitive prediction and improved candidate-set coverage within a unified framework under limited structural supervision and controlled sequence separation. Code is available at `https://anonymous.4open.science/r/DTFlow-3C0D/`.

1 INTRODUCTION

RNA structure spans primary sequence, conventional secondary structure, a full base-pair matrix (BPM), and the three-dimensional fold (Figure 1a). The same sequence can adopt multiple conformations (Chowdhury et al., 2006; Reining et al., 2013), which may appear identical in the nested Watson–Crick scaffold of conventional secondary structure. A full BPM additionally records non-canonical interactions through Watson–Crick, Hoogsteen and sugar edges, many of which organize tertiary motifs. Across 3,816 PDB-derived RNAs, 41.4% are observed in two or more distinct BPM states (Figure 1b). We therefore study multi-state RNA structure through BPMs, which preserve interactions relevant to tertiary organization while remaining more tractable and less sensitive to coordinate-level variation than full atomic structures.

Most learning-based methods predict a single structure per sequence (Fu et al., 2022; Penić et al., 2025), leaving alternative BPM states unrepresented. For downstream tertiary prediction, a single BPM also supplies only one pairing hypothesis as structural input. Generative models offer alternative candidates, and a BPM's two-dimensional form suggests adapting image-generation architectures. However, BPMs encode sparse binary relations, where a small set of pairings can distinguish states. Learning these relationships from limited structural data is challenging when closely related training sequences are excluded. On our benchmark, retrained RNADiffFold (Wang et al., 2025), with its U-Net denoiser, and our direct Diffusion Transformer adaptation (Peebles & Xie, 2023)

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

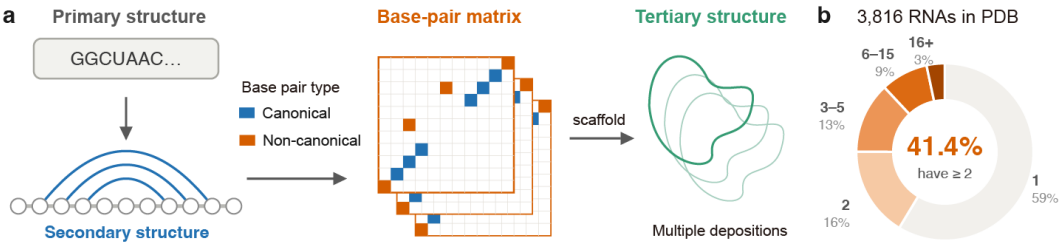


Figure 1: **(a)** A sequence folds into base pairs, recorded as a matrix over both canonical and non-canonical geometries, and those pairs scaffold the tertiary structure; one sequence is deposited in several such matrices. **(b)** Distinct base-pair states per RNA over the 3,816 RNAs in the archive: 41.4% are observed in two or more.

trail strong predictive baselines (Table 1). Since both prediction and generation require plausible sequence–pair relationships, we reuse predictive representations and supervision to support generation.

We introduce **DTFlow**, a dual-track framework unifying RNA BPM prediction and generation. Its pretrained sequence track and full-resolution pair track communicate in the same residue-pair coordinates: pair embeddings bias sequence attention, and attention maps update the pair embeddings. Two time variables, t_{seq} and t_{pair}, modulate this exchange and, together with the input states, select prediction, conditional BPM generation or auxiliary masked sequence reconstruction within the same architecture. Prediction and generation share a BPM endpoint readout and objective, allowing predictive weights to initialize generation. This combines pair-biased attention (Jumper et al., 2021) and predictor-to-generator adaptation (Jing et al., 2024) around an RNA pairing target and a common task interface (Section 3).

We also introduce **RNA-MultiBP**, which groups PDB observations by sequence while preserving distinct BPM references. Sequence, structural and family information define splits anchored on CASP and RNA-Puzzles targets, excluding target-related sequence groups from training and validation. Held-out-family and within-family subsets are reported separately. Experimental supervision covers 1,893 training sequences, supplemented by encoder pretraining and RNAfold-labelled augmentation (Sections D and 4.1). Evaluation measures reference coverage and candidate fidelity at explicit candidate counts: how closely the candidate set reaches each reference, and each candidate matches an observed BPM.

Contributions. Our contributions are **(i)** a unified dual-track predictive and generative framework; **(ii)** a multi-reference benchmark with controlled sequence separation and explicit family strata; and **(iii)** an empirical study under limited structural supervision. On 56 test RNAs, DTFlow-P approaches RNAfold and ProbKnot in All coverage while outperforming RiNALMo at the same single-output budget. At $K = 16$, DTFlow-G achieves All coverage of 0.651 versus 0.555 for retrained RNADiffFold, and Multi coverage of 0.653 versus 0.623 for the single predictive output (Table 1). Training and sampling studies assess the design choices (Sections 4.3 and 4.4). In a riboswitch case study, its generated BPMs also yield trRosettaRNA2 candidates closer to the deposited conformers than conditioning on the deposited BPMs (Section 4.5).

2 Related Work

Secondary structure and base-pair prediction. Thermodynamic methods such as ViennaRNA (Lorenz et al., 2011) and learned methods including language-model structure heads (Chen et al., 2022; Penić et al., 2025), UFold (Fu et al., 2022), SPOT-RNA (Singh et al., 2019) and RibonanzaNet (He et al., 2024) provide strong single-output prediction baselines. A single returned structure does not explicitly represent alternative structures of the same sequence. Output conventions also differ: canonical secondary-structure predictions omit some interactions present in a full BPM target. We distinguish benchmark-trained heads from released structure predictors with additional supervised exposure (Section D.3).

Generative models of base pairing. RNADiffFold (Wang et al., 2025) is the closest comparator, and placing it precisely matters for what our comparison can claim. Its denoiser is a UFold U-Net over the contact map with RNA-FM features as conditioning, and its generative process is two-class multinomial diffusion (Hoogeboom et al., 2021) — on a binary state, exactly the uniform-transition case of a discrete diffusion (Austin et al., 2021). DTFlow instead uses a continuous symmetric source and an endpoint-parameterised flow. These methods differ in process, denoiser and backbone, so their comparison cannot isolate the effect of any single component. Our DiT adaptation supplies an additional image-style baseline; its configuration and differences are disclosed in Section D.3.

Tertiary structure, and methods that consume base pairs. Base pairs are an input to tertiary prediction as well as an output of it. RNAbpFlow (Tarafder & Bhattacharya, 2026) conditions an SE(3) flow on sequence and base-pair information, motivating the study of generated pair maps as structural inputs. Such RNA-specific approaches complement general-purpose biomolecular predictors such as AlphaFold 3 and Boltz-2 (Abramson et al., 2024; Passaro et al., 2025). The downstream study in Section 4.5 evaluates base-pair agreement from tertiary predictions with and without generated structural inputs. This does not establish improved recovery of distinct tertiary states.

Flow matching. Flow matching (Lipman et al., 2023) and rectified flow (Liu et al., 2023) learn a velocity field transporting a simple source to the data, and discrete variants extend this to categorical state spaces (Campbell et al., 2024). Our application uses a structured state: a base-pair matrix is symmetric and sparse, and it is *not* a matching, since a residue pairs on its Watson-Crick, Hoogsteen and sugar edges at once. That last point rules out the categorical treatment a one-partner-per-residue state would allow, and motivates the matrix representation of Section 3.1.

Splits and leakage control. Family- and clan-level grouping using Rfam (Kalvari et al., 2021), and homology-aware component grouping in RNA3DB (Szikszai et al., 2024), provide precedents for controlling relatedness across splits. Our grouping uses MMseqs2 (Steinegger & Söding, 2017) and our reference pairs come from FR3D (Sarver et al., 2008) and DSSR (Lu et al., 2015) annotations. Section A.4 describes our exclusion rules and their limits.

3 METHOD

DTFlow unifies prediction and generation through a shared architecture and BPM endpoint objective. Two time variables, t_{seq} and t_{pair}, condition the network on the state of each track. Together with the corresponding inputs, these variables provide a common interface for structure prediction, conditional BPM generation and auxiliary masked sequence reconstruction. Task changes preserve the network architecture and readout heads; predictive training supplies the weights that initialise the generative stage.

3.1 A DUAL-TRACK ARCHITECTURE

The input is a nucleotide sequence of length L. Its structure is a **base-pair matrix** $x_1 \in \{0,1\}^{L\times L}$, symmetric, with a 1 wherever the two residues pair under any Leontis–Westhof geometry (Leontis & Westhof, 2001); one sequence has several such matrices, each a distinct *base-pair state*. The two sides are corrupted differently, because they are different kinds of object.

$$\begin{aligned} &\text{pairing:} && x_{t_{\text{pair}}} = (1 - t_{\text{pair}})\, x_0 + t_{\text{pair}}\, x_1, \\ &\text{sequence:} && \Pr\big(i \in \mathcal{M} \mid t_{\text{seq}}\big) = 1 - t_{\text{seq}}. \end{aligned}$$

The pairing side is a linear interpolation from a continuous source, x_0 drawn uniformly on $[0,1]$ and mirrored so the state stays symmetric. On the sequence side, uniform masking independently replaces each residue with a mask token with probability $1 - t_{\text{seq}}$. The resulting masked positions form $\mathcal{M}$ (Section 3.3). The sequence time specifies the expected visible fraction; a clean sequence has $t_{\text{seq}} = 1$, while $t_{\text{seq}} = 0$ masks every residue. Conditional tasks keep the conditioning track clean, while Joint corrupts both tracks. Diagonal and padded pairing cells are held at zero throughout (Section C). Prediction uses a *neutral state* with every legal pairing cell fixed at $\frac{1}{2}$, the mean of the

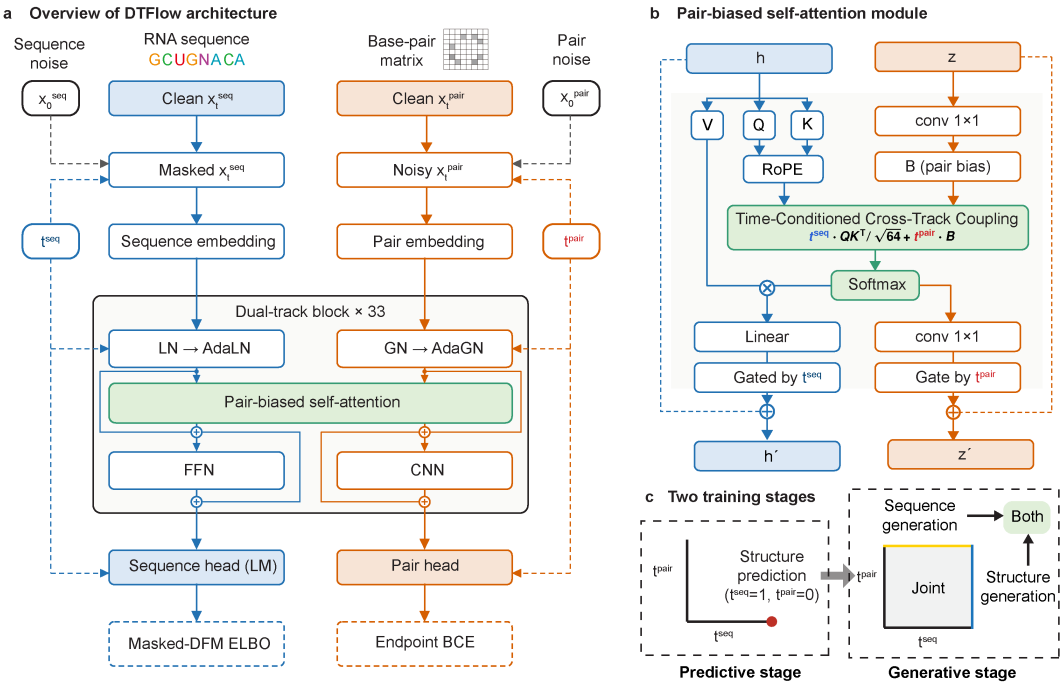


Figure 2: **DTFlow overview. (a)** Sequence and pair tracks predict masked residues and BPM endpoints. **(b)** Pair-biased attention couples the tracks through cross-track time modulation. **(c)** Predictive training initialises generative training. Both combines the conditional tasks on the right and top edges; Joint corrupts both tracks. Uniform sequence masking supplies auxiliary masked-residue supervision (Section 3.3).

source distribution. This value is centred at zero before the pair-input projection. Generation uses a sampled source. The neutral and sampled queries can share the same times $(t_{\text{seq}}, t_{\text{pair}}) = (1, 0)$, while differing in their pairing inputs (Section 4.1).

Figure 2 shows the two tracks. The *sequence track* is a pretrained nucleotide language model (RiNALMo, 651M parameters), with 33 blocks carrying one embedding per residue. The *pair track* has one block per sequence block and carries one embedding per nucleotide pair. A 1×1 convolution embeds the input pairing state at the first block; subsequent blocks update this representation.

The tracks share residue-pair coordinates at their attention interface. At each block, the pair embedding biases sequence attention before the softmax, and the resulting attention map updates the pair embedding. This bidirectional exchange lets the current pairing state influence sequence processing and returns the sequence-derived relationships to the pair track. Cross-track time modulation controls the exchange (Section 3.2). The model preserves nucleotide-pair resolution throughout and enforces BPM symmetry at the input and readout.

3.2 Time-Conditioned Cross-Track Coupling

The two time conditions enter adaptive layer normalisation on the sequence track and adaptive group normalisation on the pair track, as well as the attention-score weights below. The sequence track projects its embeddings to Q, K, V and applies rotary positional encoding. The pair track projects its embeddings through a 1×1 convolution to one channel per attention head, yielding a bias B. The attention logits combine both sources in the same residue-pair coordinates:

$$A \;=\; t_{\text{seq}} \cdot \frac{QK^{\top}}{\sqrt{d}} \;+\; t_{\text{pair}} \cdot B.$$

At $t_{\text{seq}} = 0$, the sequence-derived score term vanishes; at $t_{\text{pair}} = 0$, the additive pair bias vanishes. Prediction uses $(1, 0)$ with the neutral pairing input. Conditional BPM generation keeps $t_{\text{seq}} = 1$

while advancing t_{pair} with the evolving state. Masked sequence reconstruction keeps $t_{\text{pair}} = 1$, with t_{seq} controlling the masking probability; Joint corrupts both tracks. These queries use the same attention interface. Source noise also enters the pair track directly, so a generative query can remain stochastic at $t_{\text{pair}} = 0$ (Section C).

3.3 Objectives

We use the sequence direction as an auxiliary reconstruction task. Uniform masking samples positions independently across the sequence, using the same probability $1 - t_{\text{seq}}$ at every residue. The language head recovers the native residues at the masked positions $\mathcal{M}$, conditioned on the corrupted sequence and the pairing input.

The pair head is supervised by binary cross-entropy on called cells, and the sequence head by token- and time-normalised cross-entropy on $\mathcal{M}$ (Campbell et al., 2024):

$$\mathcal{L}_{\text{pair}} = \frac{1}{|C|} \sum_{(i,j)\in C} \text{BCE}\Big(\sigma(Z_{ij}),\, x_1^{ij}\Big)\,,$$

$$\mathcal{L}_{\text{seq}} = \frac{1}{L \max(1 - t_{\text{seq}}, 1/L)} \sum_{i\in\mathcal{M}} -\log p_\theta\big(s_i \mid \widetilde{s}, x_{t_{\text{pair}}}, t_{\text{seq}}, t_{\text{pair}}\big)\,.$$

Here s and $\widetilde{s}$ are the native and corrupted sequences; Z denotes pair-head logits, distinct from attention scores A. The set C contains called, legal cells: uncalled cells carry no pair supervision. The sequence term is zero when $\mathcal{M}$ is empty. The time normalisation uses the masking probability, with clipping at $1/L$ (Section C). Task indicators a_{pair} and a_{seq} activate the appropriate endpoint losses:

$$\mathcal{L} \;=\; a_{\text{pair}}\mathcal{L}_{\text{pair}} \;+\; w_{\text{seq}} a_{\text{seq}}\, \mathcal{L}_{\text{seq}}, \qquad w_{\text{seq}} \geq 0,$$

where pairing-only training sets $w_{\text{seq}} = 0$. Both combines the pairing and sequence objectives through conditional tasks: each training sample corrupts one track while keeping the other clean. Joint instead combines uniform sequence masking with a corrupted pairing input (Figure 2c).

Neither loss directly regresses a velocity. We use an endpoint-prediction parameterization (Lipman et al., 2023; Ho et al., 2020; Salimans & Ho, 2022): the network emits logits Z whose sigmoid $\hat{x}_1 = \sigma(Z)$ predicts the clean matrix, and the field the sampler integrates is assembled from that prediction at sampling time, on the linear path of Section 3.1,

$$v_t \;=\; \frac{\hat{x}_1 - x_t}{1 - t}, \qquad \text{so that} \qquad x_t + (1 - t)\, v_t \;=\; \hat{x}_1 \;\text{ identically.}$$

A single Euler step over the remaining interval $1 - t$ therefore lands on the predicted endpoint. With multiple steps, the network re-estimates the endpoint along the trajectory; Section 4.4 evaluates how step count affects coverage and fidelity.

For structure generation, the sequence remains clean while the pairing state moves from a symmetric uniform source toward the endpoint. We use T Euler steps and decode the final endpoint with a fixed threshold. Independent source draws yield K candidates; Section C gives the full update.

3.4 Two-stage training

Stage 1 trains structure prediction at $(t_{\text{seq}}, t_{\text{pair}}) = (1, 0)$ with the pairing input held at its neutral value and conditioning-specific parameters fixed at identity initialisation. The sequence encoder and the attention-to-pair pathway, including the pair blocks and output head, remain trainable. We augment the PDB training split by mutating its sequences and using RNAfold to generate base-pair labels for the variants.

Stage 2 initialises from these predictive weights and introduces corrupted inputs, using conditional tasks for Both or simultaneous corruption for Joint. The endpoint readout is shared across stages. Zero-initialised conditioning branches preserve the neutral predictive computation under identical queries; this does not imply equality at arbitrary noisy states or times. The training-strategy comparison in Table 2 includes the additional predictive stage and its data exposure. Sections C and D provides the implementation and protocol details.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and baselines. All main results use **RNA-MultiBP**, our split and evaluation for PDB-derived RNA, with base-pair matrix (BPM) targets. The 10–512 nt window contains 7,886 training records spanning 1,893 unique sequences and 419 validation records spanning 163 sequences. The 187 test observations span 56 exact sequences associated with CASP15 (Das et al., 2023), CASP16 (Kretsch et al., 2026) and RNA-Puzzles (Cruz et al., 2012). Split construction excludes target-related sequence groups using sequence, structural and fragment-reuse links (Section A.2). The test observations include 105 official-target records and 82 additional same-sequence records. We report inter-family (44 sequences) and intra-family (12 sequences) subsets separately; All contains both. Single/Multi is defined by full-BPM reference-group multiplicity, using the same BPM pair sets and observation masks for all methods: 30 sequences have one reference group and 26 have multiple groups. Section A details construction, grouping and leakage-control scope. Baselines include thermodynamic predictors, language-model structure heads, RNADiffFold and **DiT-BPM**, our direct DiT adaptation. **DTFlow-P** denotes the stage-1 predictor and **DTFlow-G** the stage-2 generative model; sampling steps T and candidate count K are reported separately. The main comparison uses structure heads trained on our split and allows sequence-only encoder pretraining. DTFlow's predictive stage additionally uses 20,000 RNAfold-labelled mutants (Section D); these are pseudo-labelled augmentation, separate from the experimentally observed training data. Section D.3 discloses each protocol, including residual supervised weights in the RNADiffFold retrain.

Metrics and comparisons. Base-pair F1 is the harmonic mean of precision and recall for predicted base pairs, evaluated only on the reference's called cells (Equation (1)). For references $y_1, \ldots, y_M$ and candidates $x_1, \ldots, x_K$, let S_{mn} be this score for candidate n against reference m. We report $\mathrm{Cov} = \frac{1}{M}\sum_m \max_n S_{mn}$ and $\mathrm{Fid} = \frac{1}{K}\sum_n \max_m S_{mn}$, then average over exact sequences. Coverage measures agreement with each reference; fidelity measures each candidate's nearest-reference agreement. These quantities differ even at $K = 1$ when several references exist. Coverage is non-decreasing for nested candidate sets, so K must accompany every comparison and fidelity is given alongside it in Section E. Neither score proves recovery of distinct conformations. DTFlow-P uses one neutral-state query; DTFlow-G uses T model evaluations for each of K sampled initial states. All main rows use the common-reference protocol in Section B.3; the released-checkpoint sampling comparison uses a separately identified common cohort (Section E.6).

4.2 PREDICTIVE VERSUS GENERATIVE BASE-PAIR PREDICTION

The predictive mode already approaches strong thermodynamic baselines under the benchmark's sequence-group separation. At $K = 1$, DTFlow-P reaches All coverage of 0.634, compared with 0.641 for both RNAfold and RNAstructure ProbKnot. RiNALMo and RNA-FM reach 0.523 and 0.373, respectively. Thus, the reported DTFlow training recipe substantially narrows the gap between these language-model predictors and the two thermodynamic baselines without relying on additional candidates. On Multi, DTFlow-P reaches 0.623, exceeding RNAfold (0.593) and ProbKnot (0.590) with the same single-output budget.

At $K = 16$, DTFlow-G exceeds retrained RNADiffFold in both All coverage (0.651 versus 0.555) and Multi coverage (0.653 versus 0.558). Our direct DiT adaptation also trails DTFlow, although backbone and training differences limit architectural attribution (Section D.3). Relative to DTFlow-P, DTFlow-G raises Multi coverage from 0.623 to 0.653, while Single coverage changes only from 0.645 to 0.649. The gain therefore concentrates on sequences with multiple BPM references, combining generative training with a larger candidate budget. DTFlow-P remains stronger on intra-family sequences (0.810 versus 0.802), and has higher All fidelity (0.643 versus 0.591; Table 5).

The unified framework therefore supports both competitive single-output prediction and improved candidate-set coverage. Performance still varies across subsets: thermodynamic methods remain stronger on single-reference sequences, held-out families and both CASP subsets. DTFlow-G has the highest Multi coverage in the main comparison (0.653, versus 0.593 for RNAfold, the stronger thermodynamic row on this subset) and the highest RNA-Puzzles coverage. An additional thermodynamic operating point in Section E.5 has higher All coverage than DTFlow-G. The gains should

Table 1: **BPM reference coverage on RNA-MultiBP**, 56 RNAs, 10–512 nt, with shared BPM references. Challenge subsets overlap. K: candidates; T: sampling steps. Coverage can increase with K; fidelity is in Table 5. RNAstructure: ProbKnot; RNAfold: ViennaRNA MFE. ‡Retrained RNADiffFold; §our DiT adaptation. Training protocols are detailed in Section D.3. Bold marks the highest coverage per column.

				by states		by family		by challenge		
method	K	T	all	single	multi	inter	intra	Puzzles	C15	C16
			56	30	26	44	12	39	10	9
Learned, structure head trained here										
RiNALMo	1	—	0.523	0.501	0.549	0.459	0.757	0.529	0.456	0.523
RNA-FM	1	—	0.373	0.395	0.348	0.324	0.554	0.386	0.336	0.374
Thermodynamic										
RNAstructure	1	—	0.641	**0.685**	0.590	**0.623**	0.708	0.618	**0.660**	**0.695**
RNAfold	1	—	0.641	0.682	0.593	0.611	0.751	0.619	0.634	0.692
Generative										
RNADiffFold‡	16	20	0.555	0.553	0.558	0.523	0.674	0.584	0.384	0.586
DiT-BPM§	1	10	0.150	0.149	0.151	0.104	0.318	0.177	0.058	0.105
Ours										
DTFlow-P	1	1	0.634	0.645	0.623	0.587	**0.810**	0.646	0.618	0.625
DTFlow-G	16	2	**0.651**	0.649	**0.653**	0.610	0.802	**0.658**	0.654	0.642

Table 2: **Two-stage training and combined supervision.** Coverage (Cov) and fidelity (Fid) on Single (30 RNAs) and Multi (26 RNAs), with $T = 2$, $K = 16$ throughout. Direct starts from the pretrained encoder; Two-stage starts from DTFlow-P. Training time pairs identify the task queries: u is the BPM corruption time and s is the sequence corruption time (Sections 3.1 and 3.3). Both combines conditional tasks; Joint corrupts both tracks. Bold marks the best task configuration within each training route, subset and metric.

	Training	Single-state (30)				Multi-state (26)			
	time pair	Direct		Two-stage		Direct		Two-stage	
Task	$(t_{\text{seq}}, t_{\text{pair}})$	Cov	Fid	Cov	Fid	Cov	Fid	Cov	Fid
Pairing only	$(1, u)$	**0.627**	**0.560**	0.615	0.552	**0.651**	**0.606**	0.626	0.583
Both	$(1, u)$ or $(s, 1)$	0.619	0.548	**0.649**	**0.581**	0.627	0.584	**0.653**	**0.603**
Joint	(s, u)	0.621	0.517	0.618	0.535	0.631	0.556	0.603	0.535

therefore be read alongside the subset results and per-candidate fidelity. Section 4.4 examines the sampling-budget contribution.

4.3 Effect of two-stage training and combined supervision

The time-pair column describes training queries; evaluation keeps $t_{\text{seq}} = 1$ and advances t_{pair} for every row. At the fixed budget $T = 2$, $K = 16$, the effect of two-stage training depends on the objective (Table 2). With Both, initializing from DTFlow-P raises Single coverage from 0.619 to 0.649 and Multi coverage from 0.627 to 0.653, with fidelity improving on both subsets. However, pairing-only and Joint coverage decrease under two-stage training. Correspondingly, pairing-only supervision performs best under direct training, whereas Both performs best under two-stage training. Across all six configurations, two-stage Both has the highest coverage on both subsets and the highest Single fidelity; direct pairing-only retains slightly higher Multi fidelity (0.606 versus 0.603) at nearly the same Multi coverage (0.651 versus 0.653). Joint has the lowest fidelity within each training route and subset. Its larger joint corruption space may be harder to learn from limited data, but this is a hypothesis rather than an isolated causal finding. The results support the combination of predictive initialization and Both for coverage, not a general benefit of staging every objective. Training history, augmentation and task exposure remain potential confounders (Section D). A supplementary RNA-FM comparison under a separate training-log evaluation is reported in Section E.2.

Table 3: **Sampling steps and sample count on All (56 sequences).** All rows share BPM references. T: steps per candidate; K: candidate count. ‡Retrained RNADiffFold, using 20 diffusion steps. §Our DiT adaptation (Section D.3). Bold marks the highest value per metric column.

		Coverage					Fidelity
Model	T	K=1	K=2	K=4	K=8	K=16	K=16
RNADiffFold‡	20	0.445	0.481	0.509	0.534	0.555	0.458
DiT§	10	0.232	0.251	0.273	0.283	0.297	0.241
DTFlow-G	1	**0.611**	**0.621**	**0.630**	0.637	0.645	**0.624**
	2	0.581	0.605	0.626	**0.639**	0.651	0.591
	4	0.580	0.603	0.620	0.638	0.649	0.583
	8	0.575	0.601	0.621	0.636	**0.652**	0.576

Use of pair-to-attention feedback. Holding the DTFlow-G checkpoint fixed, we suppress the pair-to-attention bias only at inference. At $T = 2$, $K = 16$ and a shared decoding threshold of 0.15, removing feedback lowers All coverage from 0.6508 to 0.6450 and fidelity from 0.5921 to 0.5861; Multi coverage falls from 0.6541 to 0.6469. The $T = 1$ control produces identical candidates. This supports use of feedback by the trained model. One-to-one reference matching changes only modestly, from 66 to 65 of 85 Multi reference groups at F1 ≥ 0.5, providing limited evidence of additional state recovery. Paired uncertainty and the decoding-setting difference from the main table are reported in Section E.4.

4.4 Effect of sampling steps and sample count

We vary the integration steps per candidate T and the number of independent candidates K while holding the DTFlow-G checkpoint fixed (Table 3). Increasing K improves reference coverage at each tested T: at $T = 2$, coverage rises from 0.581 to 0.651 as K increases from 1 to 16, while fidelity remains nearly unchanged ($0.590 \rightarrow 0.591$). Increasing T offers a smaller coverage gain: at $K = 16$, moving from 1 to 8 steps raises coverage from 0.645 to 0.652, while fidelity decreases from 0.624 to 0.576. The budget allocation comparison is clearer at fixed $T \times K = 16$: configurations $(T, K) = (1, 16), (2, 8), (4, 4), (8, 2)$ achieve coverage of 0.645, 0.639, 0.620, and 0.601, respectively, with $(1, 16)$ also achieving the highest fidelity. Thus, within the tested range, allocating the same number of model evaluations to more candidates gives higher reference coverage and fidelity. These query counts do not measure wall-clock speedups within or across models. DTFlow-G also exceeds both trained generative baselines in coverage at every tested K.

4.5 Case study: generated BPMs let a tertiary predictor reach riboswitch conformers

Tertiary structure predictors such as trRosettaRNA2 produce a single structure for each input BPM; conditioning them on multiple DTFlow-G samples therefore yields an ensemble of tertiary candidates. We illustrate this with the guanidine-III riboswitch (RNA-Puzzles 21, 41 nt), which is deposited as 11 chains from six crystal structures and is represented in Figure 3 by three conformers: 5NWQ A, 5NWQ B and 5O62 B. When conditioned on its own predicted BPM, trRosettaRNA2 attains TM-scores of $0.57/0.57/0.39$ against these conformers. Conditioning on the deposited BPMs, one per chain, yields no improvement, where the best resulting structure attains TM-scores of $0.61/0.59/0.40$, and the base pairs extracted from it reach an F1 of $0.79/0.78/0.78$, lower than the $0.84/0.83/0.84$ obtained with the predictor's own input. In contrast, conditioning on four DTFlow-G samples produces a structure with TM-scores of $0.85/0.80/0.62$, the highest of the three inputs for every conformer, and base-pair F1 of $0.95/0.91/0.94$. It is also the only input that recovers a base pair specific to 5NWQ A. Over all 16 DTFlow-G samples, TM-score coverage (Equation (2)) increases from 0.461 to 0.770 and fidelity from 0.547 to 0.709, compared with 0.505 and 0.557 for the deposited BPMs. These results indicate that a single input map, even a correct one, constrains the predictor to one preferred static fold, whereas diverse DTFlow-G samples provide alternative pairings that allow it to reach the deposited multiple conformations.

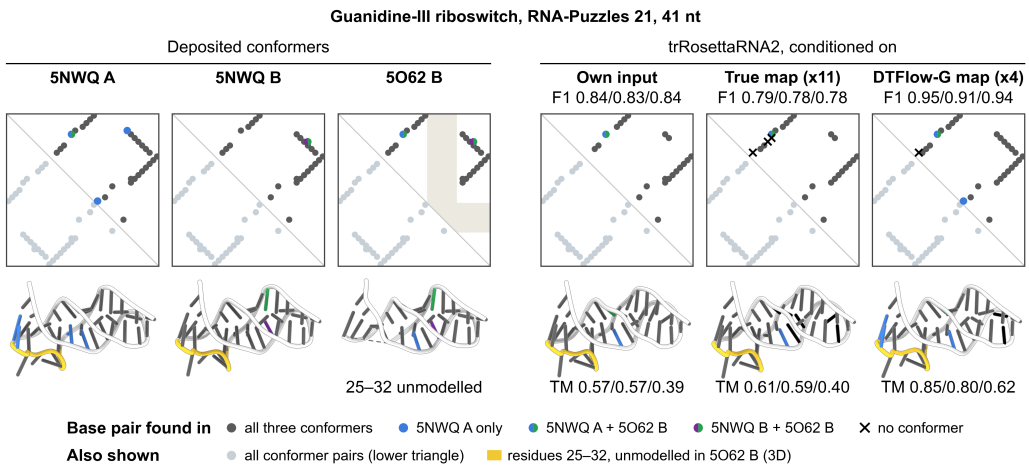


Figure 3: **Case study: the guanidine-III riboswitch (RNA-Puzzles 21, 41 nt), deposited in three conformers.** Top, base-pair maps: the upper triangle holds each structure's pairs, coloured by the conformers that carry them; the lower triangle holds all conformer pairs for comparison. Bottom, the same structures in 3D, rungs coloured as in the maps. F1 and TM are against 5NWQ A / 5NWQ B / 5O62 B, for the candidate closest to any conformer.

5 Conclusion

We introduced DTFlow to unify RNA base-pair prediction and generation. Sequence and pair tracks interact at nucleotide-pair resolution, with two time conditions and corresponding input states selecting tasks within a shared architecture. A common BPM endpoint objective connects predictive training to generative fine-tuning. RNA-MultiBP evaluates this framework under sequence- and structure-aware data separation while retaining multiple observed pairing states. In this setting, DTFlow's single-output predictor approaches RNAfold and ProbKnot in All reference coverage and improves over the language-model baselines. Its generative model exceeds retrained RNADiffFold at equal candidate count and improves Multi coverage over the predictive output when given a larger candidate budget. An inference-time diagnostic supports the trained model's use of pair-to-attention feedback. These findings support reusing learned sequence–pair relationships for both prediction and generation with limited structural supervision.

Limitations and future work. Evaluation covers 56 RNAs of up to 512 nucleotides, on which thermodynamic methods remain stronger on held-out families and both CASP subsets, and an additional operating point exceeds DTFlow-G in All coverage (Section E.5). Coverage gains trade off against per-candidate fidelity, the feedback diagnostic changes one-to-one reference recovery only modestly, and the isolated benefit of cross-track time weighting remains untested. Deposited structures do not establish equilibrium populations, and in tertiary prediction the folders' own inputs remain the stronger single prediction on single-conformer targets. The sequence direction currently provides only auxiliary reconstruction; future work will scale sequence–structure data, use this direction for structure-conditioned design, and test whether the shared predictive–generative construction transfers to other tasks with coupled sequence and pair representations.

AI use statement

We used generative AI tools substantially in preparing this work, and describe the scope rather than characterising it in general terms.

Writing. Generative AI assisted in revising and polishing this manuscript, including the organisation of sections and the prose of the method and evaluation descriptions. Every claim was checked against our own experimental records, and numerical values were transcribed from the recorded result files rather than produced by a model.

Code. Generative AI assisted in writing and refactoring training, evaluation and analysis code, and in auditing existing code. Several findings reported here originated in such an audit—among them a missing pretrained normalisation in our transformer loop, and the observation that a previously reported gap had been read from the split the arms were trained on. Each was confirmed by reading the code and re-measuring before it was written up.

Analysis. Generative AI assisted in reading our own logs and result files and in proposing checks. It did not select which results to report.

Not used. We did not use generative AI to produce experimental results, to generate or modify data, or to write the review of a paper.

We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, claims and artifacts.

Ethics statement

This work uses publicly deposited RNA structures from the Protein Data Bank and publicly released model checkpoints, under their respective terms. It involves no human subjects, no personally identifying information, and no newly collected data.

One point deserves stating because it bears on how results here could be misread. The structures we treat as an RNA's "states" are consolidations of deposited annotations, not measured conformational populations, and deposition frequency reflects what has been crystallised rather than what a molecule does in solution. Claims derived from this benchmark are about the set of structures the archive records, and should not be carried into biological or clinical settings as statements about physiological ensembles.

Reproducibility statement

Data construction, split assignment and the scored cohort are documented in Section A. Section B defines masked pair-level F1, sequence-level coverage and fidelity, readout budgets and the shared BPM reference groups. Section C gives the model, task objectives and sampling updates. The manifest checksum, baseline adaptations and training settings appear in Section D; supplementary results are in Section E.

References

Josh Abramson, Jonas Adler, Jack Dunger, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Stanislav Bellaousov and David H. Mathews. ProbKnot: Fast prediction of RNA secondary structure including pseudoknots. *RNA*, 16(10):1870–1880, 2010. doi: 10.1261/rna.2125310. URL https://doi.org/10.1261/rna.2125310.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5453–5512. PMLR, 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, Irwin King, and Yu Li. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. *arXiv preprint*

arXiv:2204.00300, 2022. doi: 10.48550/arXiv.2204.00300. URL https://arxiv.org/abs/2204.00300.

Saheli Chowdhury, Christophe Maris, Frédéric H.-T. Allain, and Franz Narberhaus. Molecular basis for temperature sensing by an RNA thermometer. *The EMBO Journal*, 25(11):2487–2497, 2006. doi: 10.1038/sj.emboj.7601128. URL https://doi.org/10.1038/sj.emboj.7601128.

José Almeida Cruz, Marc-Frédérick Blanchet, Michal Boniecki, et al. RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*, 18(4):610–625, 2012. doi: 10.1261/rna.031054.111. URL https://doi.org/10.1261/rna.031054.111.

Rhiju Das, Rachael C. Kretsch, Adam J. Simpkin, Thomas Mulvaney, Phillip Pham, Ramya Rangan, Fan Bu, Ronan M. Keegan, Maya Topf, Daniel J. Rigden, Zhichao Miao, and Eric Westhof. Assessment of three-dimensional RNA structure prediction in CASP15. *Proteins: Structure, Function, and Bioinformatics*, 91(12):1747–1770, 2023. doi: 10.1002/prot.26602. URL https://doi.org/10.1002/prot.26602.

Laiyi Fu, Yingxin Cao, Jie Wu, Qinke Peng, Qing Nie, and Xiaohui Xie. UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Research*, 50(3):e14, 2022. doi: 10.1093/nar/gkab1074. URL https://doi.org/10.1093/nar/gkab1074.

Shujun He, Rui Huang, Jill Townley, et al. Ribonanza: deep learning of RNA structure through dual crowdsourcing. *bioRxiv*, 2024. doi: 10.1101/2024.02.24.581671. URL https://www.biorxiv.org/content/10.1101/2024.02.24.581671.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 12454–12465, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. AlphaFold meets flow matching for generating protein ensembles. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22277–22303. PMLR, 2024. URL https://proceedings.mlr.press/v235/jing24a.html.

John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Ioanna Kalvari, Eric P. Nawrocki, Nancy Ontiveros-Palacios, et al. Rfam 14: expanded coverage of metagenomic, viral and microrna families. *Nucleic Acids Research*, 49(D1):D192–D200, 2021. doi: 10.1093/nar/gkaa1047. URL https://doi.org/10.1093/nar/gkaa1047.

Rachael C. Kretsch, Alissa M. Hummer, Shujun He, Rongqing Yuan, Jing Zhang, Thomas Karagianes, Qian Cong, Andriy Kryshtafovych, and Rhiju Das. Assessment of nucleic acid structure prediction in CASP16. *Proteins: Structure, Function, and Bioinformatics*, 94(1):192–217, 2026. doi: 10.1002/prot.70072. URL https://doi.org/10.1002/prot.70072.

Neocles B. Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512, 2001. doi: 10.1017/s1355838201002515. URL https://doi.org/10.1017/s1355838201002515.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

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 (ICLR)*, 2023. URL https://arxiv.org/abs/2209.03003.

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

Xiang-Jun Lu, Harmen J. Bussemaker, and Wilma K. Olson. DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Research*, 43(21):e142, 2015. doi: 10.1093/nar/gkv716. URL https://doi.org/10.1093/nar/gkv716.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707. URL https://doi.org/10.1101/2025.06.14.659707.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

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

Anke Reining, Senada Nozinovic, Kai Schlepckow, Florian Buhr, Boris Fürtig, and Harald Schwalbe. Three-state mechanism couples ligand and temperature sensing in riboswitches. *Nature*, 499(7458):355–359, 2013. doi: 10.1038/nature12378. URL https://doi.org/10.1038/nature12378.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.00512.

Michael Sarver, Craig L. Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B. Leontis. FR3D: finding local and composite recurrent structural motifs in RNA 3d structures. *Journal of Mathematical Biology*, 56(1-2):215–252, 2008. doi: 10.1007/s00285-007-0110-x. URL https://doi.org/10.1007/s00285-007-0110-x.

Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, 10(1):5407, 2019. doi: 10.1038/s41467-019-13395-9. URL https://doi.org/10.1038/s41467-019-13395-9.

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

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

Sumit Tarafder and Debswapna Bhattacharya. RNAbpFlow: base pair-augmented SE(3) flow matching for conditional RNA 3D structure generation. *Nature Methods*, 23(7):1349–1358, 2026. doi: 10.1038/s41592-026-03128-4. URL https://doi.org/10.1038/s41592-026-03128-4.

Zhen Wang, Yizhen Feng, Qingwen Tian, Ziqi Liu, Pengju Yan, and Xiaolin Li. RNADiffFold: generative RNA secondary structure prediction using discrete diffusion models. *Briefings in Bioinformatics*, 26(1):bbae618, 2025. doi: 10.1093/bib/bbae618. URL https://doi.org/10.1093/bib/bbae618.

Sebastian Will, Kristin Reiche, Ivo L Hofacker, Peter F Stadler, and Rolf Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Computational Biology*, 3(4):e65, 2007. doi: 10.1371/journal.pcbi.0030065. URL https://doi.org/10.1371/journal.pcbi.0030065.

Sebastian Will, Tejal Joshi, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs. *RNA*, 18(5):900–914, 2012. doi: 10.1261/rna.029041.111. URL https://doi.org/10.1261/rna.029041.111.

APPENDIX

This appendix documents the dataset and split construction (Section A), evaluation protocol (Section B), model and sampling details (Section C), experimental configurations (Section D), and supplementary results (Section E). Supplementary numerical results use the same 10–512 nt evaluation window as the main tables.

A RNA-MULTIBP: DATA CONSTRUCTION AND SPLITS

A.1 OBSERVATIONS, SEQUENCES, AND STRUCTURAL TARGETS

RNA-MultiBP groups PDB-derived RNA observations by exact nucleotide sequence while retaining their observed base-pair annotations. Structures are retrieved as biological assemblies and annotated with DSSR (Lu et al., 2015). Integrated functional elements (IFEs) distinguish structurally integrated RNA units from arbitrary assembly membership; the present benchmark uses single-chain observations. Sequence registration is checked before grouping, including container fragments and unresolved terminal duplications. An IFE identifier retains the provenance of each observation, whereas an exact sequence identifies the RNA used for aggregate evaluation.

Each observation provides a base-pair matrix (BPM) and a dot-bracket (DBN) projection. BPM entries are 1 for an annotated pair under any Leontis–Westhof geometry, 0 for a called non-pair, and -1 for an uncalled cell. The binary target is therefore accompanied by a separate observation mask. The nested canonical DBN projection used in this dataset discards interactions outside that projection; this is a property of our projection, not a limitation of every extended dot-bracket notation. In particular, two observations can have the same projected DBN but different BPMs. A residue may participate in several annotated interactions, so BPM decoding does not impose a one-partner constraint.

An unresolved residue or cell is not labelled unpaired. The DBN symbol – denotes no-call, and uncalled BPM cells are excluded from both the loss and the pair-level evaluation. Grouping observations with identical supervised content reduces deposition redundancy without assigning higher weight to frequently deposited RNAs. Reference equivalence and the distinction between a recorded annotation and a biological conformational state are specified in Section B.

A.2 GROUPING WITHOUT COMPLETE FAMILY ANNOTATION

The split construction has two levels: sequence and structure similarity identify related RNA groups, and additional fragment links prevent close subsequences from crossing partitions. This permits inclusion of RNAs without Rfam annotations. Rfam labels (Kalvari et al., 2021) provide calibration and audit information where available, rather than being required for every RNA.

The first level combines sequence and structural evidence. Sequence links include exact identity, substring containment, MMseqs2 clustering at identity 0.5 and coverage 0.3, and local alignments at identity 0.95 with a 30 nt minimum aligned length (Steinegger & Söding, 2017). Structural links include identical pair sets, offset containment, and global and local LocARNA alignments (Will et al., 2007; 2012). Coverage and recurrence filters limit merges caused by short motifs shared among unrelated RNAs. A regularised quadratic logistic model proposes further cluster merges, subject to a linkage constraint; family annotations then calibrate remaining fragmented families. These calibration operations are part of data construction, not independent validation of the clustering procedure.

The second level adds fragment-reuse links at identity 0.98 over at least 50 nt and identity 0.95 over at least 20 nt. Connected groups are used to restrict split assignments. CASP15, CASP16 and RNA-Puzzles targets anchor the test partition, including additional deposited observations of the same sequence. Smaller target-associated families and unlabelled clusters are held out as inter-family test groups. For large annotated families with at least 100 exact sequences, target-related sequence groups are held out as intra-family targets while other members can remain in training. The remaining data are partitioned by clan-linked components into training and validation.

Table 4: Manifest entries after the 10–512 nt filter. Training and validation entries are retained BPM-parent records; test entries preserve observations. They must not all be interpreted as unique structural states.

Partition	Manifest entries	Exact sequences
Training	7,886	1,893
Validation	419	163
Test	187	56

A.3 Manifest and evaluation cohort

The scored length window is narrower than the source manifest's 1022 nt upper bound. Its 187 test records comprise 105 official-target records and 82 additional same-sequence observations. They span 56 exact sequences: 44 inter-family and 12 intra-family. Challenge subsets contain 39 RNA-Puzzles, 10 CASP15 and 9 CASP16 sequences. Challenge membership can overlap, so these counts must not be summed. Single/Multi stratification is defined by the number of distinct BPM reference groups per exact sequence (Section B.3). The 187 observations form 115 BPM reference groups: 30 sequences have one group and 26 have more than one.

A.4 Scope of leakage control

The split addresses exact-sequence reuse, related sequence groups, and local fragment reuse in the supervised dataset. Inter-family and intra-family results answer different generalisation questions and are reported separately; All includes both. Family exclusion does not imply that every higher-level clan is absent from training, and intra-family evaluation deliberately retains training members of the target family.

These controls do not audit every corpus used to pretrain a released model. In particular, sequence-only language-model pretraining and supervised structure-head pretraining are separate sources of exposure. Baseline initialisation is disclosed in Section D.3. PDB observations also vary in experimental conditions and completeness; the dataset is an archive of observed annotations, not an estimate of equilibrium state probabilities or a condition-controlled conformational ensemble.

B Evaluation protocol

B.1 Masked base-pair F1

For reference m of RNA s, let R_{sm} be its set of annotated unordered pairs and C_{sm} the set of called, legal upper-triangular cells. Given a predicted pair set P_{sn}, restrict it to $P_{sn}^{(m)} = P_{sn} \cap C_{sm}$. The pair-level similarity is

$$S_{smn} = \frac{2|P_{sn}^{(m)} \cap R_{sm}|}{|P_{sn}^{(m)}| + |R_{sm}|}. \tag{1}$$

The implementation assigns 0 when either set is empty, including when both are empty; empty targets are therefore not automatically perfect matches. The BPM scorer admits $i < j$ with minimum separation 1, including crossing and non-canonical pairs. It neither converts missing annotations into negative labels nor requires conversion of a predicted pair set into DBN. Each reference uses its own observation mask.

B.2 Coverage and fidelity at the RNA level

For M_s references and K_s generated candidates, we report

$$\mathrm{Cov}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} \max_{1 \le n \le K_s} S_{smn}, \qquad \mathrm{Fid}_s = \frac{1}{K_s} \sum_{n=1}^{K_s} \max_{1 \le m \le M_s} S_{smn}. \tag{2}$$

Dataset scores are unweighted means over the scored exact sequences. Coverage measures how well the candidate set approximates each reference; fidelity measures each candidate's agreement

with its closest reference. These are F1-based nearest-reference summaries, not the distributional precision/recall estimators used for natural images. Fidelity does not establish physical validity, and a low-fidelity candidate may be either an error or a structure absent from the observed reference set.

At $K_s = 1$, coverage averages the predictor's F1 over references, whereas fidelity takes its best reference match. They coincide when $M_s = 1$, but need not coincide for a single-output predictor on a multi-reference RNA. Increasing K_s with nested candidate sets cannot decrease coverage by definition. Consequently, every candidate-count comparison should be read together with fidelity and the declared sampling budget. Duplicate candidates remain in these summaries; their frequency affects fidelity.

B.3 Reference grouping and reported strata

Both reference grouping and Single/Multi stratification use BPMs. For each exact sequence, pool all eligible test-manifest observations, including official-target and same-sequence records. Collapse observations only when their complete BPM pair sets and called-cell masks are identical; retain one representative of each group. Let M_s be this number of BPM reference groups. Single contains RNAs with $M_s = 1$, and Multi contains RNAs with $M_s > 1$. The same groups define the references in Equation (2) for every method.

Two observations with identical canonical DBN projections but different non-canonical BPM interactions therefore remain separate references. No DBN projection is used as a proxy for BPM state identity. The mask is retained to avoid equating unresolved cells with called non-pairs; mask-only differences identify different supervised observations, not proven biological conformations.

A sequence's candidates are evaluated against every BPM reference group, regardless of which deposited observation supplied the prediction record. A deterministic predictor supplies one common candidate per sequence. All methods must cover the same sequence cohort and reference scope, and candidate-set comparisons require the declared number of draws for each RNA.

The common cohort contains 56 sequences and 115 reference groups, partitioned into 30 Single and 26 Multi sequences. Removing the called mask from state identity would yield 31 Single and 25 Multi sequences; thus one member of Multi differs only in annotation coverage. We retain the mask to preserve each observation's scoring support and do not interpret this case as evidence of alternative biological states. All reported main configurations cover the full cohort. Predictive scores are either recomputed from saved predictions or re-aggregated from complete per-observation masked BPM scores; the latter are checked to agree within each reference group.

B.4 Predictive and generative readouts

A DTFlow-P prediction uses the clean sequence and the neutral pairing input $x_{\text{neutral}} = \frac{1}{2}E$ at $(t_{\text{seq}}, t_{\text{pair}}) = (1, 0)$, where E is the legal-cell mask. This is a deterministic query in evaluation mode. It is counted as one model evaluation in the tables, even when historical logs call it `ode0`. DTFlow-G instead starts from a sampled pairing matrix and performs T Euler steps, independently for each of K candidates. At $T = 1$, zero pair time suppresses pair-to-attention bias, but the sampled input can still affect the pair track and its output. A one-step generative query is therefore distinct from the neutral predictive query.

The K sweeps retain the first K candidates in sample-index order, giving nested candidate sets. RNADiffFold uses its 20-step discrete schedule and DiT uses the indicated integration budget. The product TK counts model queries for these sweeps; it is not a measurement of wall-clock cost across different architectures.

B.5 What these metrics do not establish

Coverage permits one candidate to match several references and therefore does not count independently recovered states. A dedicated state-recovery analysis would deduplicate candidates, compare references on compatible called regions, and use a declared similarity criterion with one-to-one assignment. No such result is inferred from the coverage tables. The current aggregates also do not provide variation across training seeds or confidence intervals. These limitations are particularly relevant to small CASP subsets and to small reported differences.

C MODEL, OBJECTIVES, AND SAMPLING

C.1 PAIR-STATE REPRESENTATION

Let $E_{ij} = \mathbf{1}[i \neq j]$ within the unpadded RNA, with $E = E^\top$. The source is sampled independently in the legal upper triangle as $x_0^{ij} \sim \mathcal{U}[0,1]$ and mirrored. Illegal cells are zero. For an observed binary endpoint x_1, the pairing path is $x_t = (1-t)x_0 + tx_1$. The network reads the centred state $\phi(x_t) = (x_t - \frac{1}{2}) \odot E$. The neutral query $x_{\text{neutral}} = \frac{1}{2}E$ consequently has zero centred input. The observation mask specifies supervision support and is distinct from the sequence-derived legality mask used at inference.

C.2 SEQUENCE–PAIR INTERACTION

The sequence track uses the pretrained RiNALMo encoder. A full-resolution pair track is updated alongside its 33 Transformer blocks. A 1×1 projection embeds the centred input once, producing the initial pair features. At each layer, normalised pair features are projected into an attention bias with one channel per head. For residue-to-residue scores,

$$A^\ell = t_{\text{seq}} \frac{Q^\ell (K^\ell)^\top}{\sqrt{d}} + t_{\text{pair}} B^\ell, \qquad H^\ell = \text{softmax}(A^\ell). \tag{3}$$

Padding masks and the encoder's special-token handling are applied by the attention implementation. Residue attention maps feed the pair-update blocks; the final pair head is symmetrised before conversion to endpoint probabilities. The interaction therefore preserves nucleotide-pair resolution throughout, without image patchification or spatial pooling.

Flow times enter through adaptive normalisation and the explicit score weights above. The adaptive sequence and pair branches use layer and group normalisation, respectively. At $t_{\text{pair}} = 0$, the additive pair bias is zero, but the pair features can still depend on x_0. This separates *feedback into sequence attention* from *state dependence of the pair output*. It also explains why a sampled one-step query is not required to equal a neutral-state prediction.

C.3 TRAINING TASKS AND LOSSES

Let $U = t_{\text{seq}}$ and $V = t_{\text{pair}}$, with 1 denoting a clean track. For the sequence task, uniform masking draws $m_i \sim \text{Bernoulli}(1-U)$ independently for each nucleotide and sets $\mathcal{M} = \{i : m_i = 1\}$. Positions in $\mathcal{M}$ are replaced with mask tokens; special tokens and padding are excluded. The same masking probability applies to every residue, independently of its pairing status. Thus $\mathbb{E}[|\mathcal{M}| \mid U] = L(1-U)$, while the realised masked count may differ. The input time U controls corruption rather than being recomputed from that count. At $U = 0$ all residues are masked, and at $U = 1$ the sequence is clean. Both and Joint use this sequence corruption with clean and corrupted pairing inputs, respectively. The generative task configurations are:

Configuration	Corruption	Active supervision
Pairing only	Clean sequence, corrupted BPM	Pair endpoint
Both	Clean-sequence pairing task or clean-BPM masked sequence reconstruction	Pair endpoint or masked residues
Joint	Corrupted BPM and uniformly masked sequence	Pair endpoint and masked residues

Both and Joint consequently differ in conditioning and task exposure, not only in whether a loss coefficient is nonzero. Their comparison is not an isolated loss-term ablation.

Write Z_θ for pair-head logits, distinct from attention scores A^ℓ. On called, legal cells C,

$$\mathcal{L}_{\text{pair}} = \frac{1}{|C|} \sum_{(i,j) \in C} \text{BCE}(\sigma(Z_\theta^{ij}), x_1^{ij}). \tag{4}$$

For the uniformly sampled masked positions $\mathcal{M}$, the token-normalised sequence loss is

$$\mathcal{L}_{\text{seq}} = \frac{\sum_{i \in \mathcal{M}} -\log p_\theta(s_i \mid s_U, x_V, U, V)}{L \max(1-U, 1/L)}. \tag{5}$$

An empty masked set contributes zero. The denominator uses the expected masked count, clipped to at least one; it does not in general equal the realised $|\mathcal{M}|$. This is the token- and time-normalised masked endpoint objective (Campbell et al., 2024). Here it supplies auxiliary sequence supervision; structure-conditioned sequence design has not been evaluated. Task indicators a_{pair} and a_{seq} activate the appropriate losses: $\mathcal{L} = a_{\text{pair}}\mathcal{L}_{\text{pair}} + w_{\text{seq}}a_{\text{seq}}\mathcal{L}_{\text{seq}}$. The mixture probabilities, loss weights and number of updates determine task exposure and must accompany each training configuration.

C.4 Predictive initialisation

Stage 1 uses clean sequences and neutral BPM inputs at $(U, V) = (1, 0)$. Conditioning-specific parameters are frozen or held at their identity initialisation. The sequence backbone and attention-to-pair prediction pathway, including pair-update blocks and the pair head, remain trainable. Zero centred input does not imply zero gradient in the pair track.

Stage 2 initialises the generative model from this predictor and activates the selected corrupted-track tasks. Zero-initialised conditioning projections preserve the neutral predictive computation at the handoff when inputs, times, normalisation and numerical settings are identical. Such a check does not establish equality for arbitrary noisy states or arbitrary flow times. The training-strategy comparison includes the additional predictive training and its data exposure; a causal estimate of initialisation alone would require matched total optimisation and data budgets.

C.5 Endpoint parameterisation and the flow field

Under the linear pairing path, the conditional target velocity is $x_1 - x_0 = (x_1 - x_t)/(1 - t)$. For a binary endpoint, the population minimiser of cellwise BCE is its conditional mean. If $\hat{x}_{1,\theta}$ approximates that mean, it induces the corresponding conditional mean velocity

$$v_\theta(x_t, t \mid s) = \frac{\hat{x}_{1,\theta}(x_t, t \mid s) - x_t}{1 - t}, \qquad 0 \le t < 1. \tag{6}$$

This explains the endpoint parameterisation; it does not assert that endpoint BCE and velocity mean-squared error have identical finite-model optimisation.

For structure generation, hold the sequence clean throughout. For each of K independent sources, set $t_j = j/T$ and iterate

$$x_{j+1} = x_j + \frac{1}{T}\,\frac{\hat{x}_{1,\theta}(x_j, t_j \mid s) - x_j}{1 - t_j}, \qquad j = 0, \ldots, T - 1. \tag{7}$$

The model is never queried at the singular time $t = 1$. The last step lands on the last predicted endpoint. Symmetric legal-cell probabilities are decoded with the configuration's fixed threshold into an unordered pair set. The reported sweeps start from the uniform source; they do not use predictor restarts or additional source-shrinkage controls.

D Experimental configurations and reproducibility

D.1 Data and result provenance

The current BPM manifest is `pdb_rna_v4_containment_l1022_bpm.csv`, filtered to 10–512 nt for the reported cohort. Its SHA-256 is

```
df47f109bd3e6d06f78b791f635fc8b9beee
87192ef1d1b0825c2b05336c9302
```

Counts after filtering are in Table 4. The adopted reference partition is recorded in `state_definition_bpm_mask.json`. DTFlow's training and sampling tables use `kt_sweep_l512_dtflow_unified.json`; the predictive row is verified against `dtflow_p_mutaug2_l512_unified.json`. Retrained RNADiffFold uses `k_sweep_l512_rnadifffold_ft_unified.json`, and the additional thermodynamic comparison uses `thermodynamic_panel_l512_unified.json`. The remaining predictive rows are re-aggregated from complete per-observation BPM scores under the same partition. All observations within a reference group

have identical F1 in these exports. The audit `review_unified_20260926.json` records source hashes and subset counts. DiT's DBN-trained sampling checkpoint is independently rescored from its saved candidates in `k_sweep_l512_dit_verified_20260926.json`; this file records the prediction, manifest and checkpoint hashes. Replaying retrained RNADiffFold at $K = 1$ through the same reference scorer reproduces its unified export exactly. The feedback diagnostic uses `feedback_gate_ep12_l512.json`; independent saved-candidate scoring, checkpoint and output hashes, prediction-deduplication checks, common-observation-support matching and paired-bootstrap intervals are recorded in `feedback_gate_ep12_audit_20260926.json`. Its actual decoding threshold is 0.15, as recorded by the evaluator; passing a threshold grid alone did not override the stored checkpoint threshold. The main-grid export separately reselected 0.3 on validation. The supplementary RNA-FM comparison (Section E.2) is transcribed from training logs, with config/log hashes, selected epochs and the threshold-selector audit recorded in `rnafm_generative_audit_20260926.json`. It is kept separate from the unified main-table exports.

D.2 DTFlow training and model selection

For predictive training, we generate $20{,}000$ mutants from sequences in the PDB training split and label them with RNAfold, sampling original and augmented examples one-to-one per epoch. The two-stage generative configurations initialise from this predictive checkpoint; direct configurations start without that predictive stage. The comparison therefore includes differences in training history and augmentation exposure. Table 2 fixes the sampling budget at $T = 2$, $K = 16$ across training configurations. Table 3 varies $T \in \{1, 2, 4, 8\}$ and $K \in \{1, 2, 4, 8, 16\}$ for the same generative model.

Checkpoint selection, decoding-threshold selection and sampling-budget selection are separate decisions. The current main comparison reports $T = 2$, $K = 16$ for DTFlow-G, while the complete test sweep is presented as sensitivity analysis. A completed validation-selection record for this default has not yet been verified; it must not be inferred from the test-set optimum.

Sequence reconstruction uses uniform masking with probability $1 - t_{\text{seq}}$ per residue, while BPM supervision is restricted to called, legal cells. The task-specific time settings are summarised in Table 2, and the objectives and sampling procedure in Section C. The supplementary RNA-FM training settings are given in Section E.2.

D.3 Baseline implementations

Language-model predictors. RiNALMo and RNA-FM use pretrained nucleotide encoders and randomly initialised ResNet structure heads trained on the benchmark training split. The reported RiNALMo configuration fine-tunes the encoder; the RNA-FM configuration trains the head with the encoder frozen. Released structure heads are outside the main comparison. Their additional supervised pretraining is not equivalent to the sequence-only pretraining allowed in the main comparison.

Thermodynamic predictors. RNAfold uses the ViennaRNA $2.7.2$ minimum-free-energy prediction. The RNAstructure row uses RNAstructure 6.6 ProbKnot (Bellaousov & Mathews, 2010). Both are evaluated as single-output methods without fitting a structure head on the benchmark. Their predictions are scored against the same type of full-BPM target, although their output representations do not include every non-canonical interaction in that target.

RNADiffFold. The retrained baseline uses RNADiffFold's denoising network and discrete noise process with a benchmark-specific training loop. This adaptation provides manifest input, no-call masking, sequence-grouped sampling and validation-based checkpoint/threshold selection. The BPM objective uses minimum pair separation 1 and its per-cell observation mask; generation uses 20 diffusion steps. The available initialisation audit identifies residual normalisation tensors from the supplied UFold weights after reset. This is a supervised carryover limitation, in addition to the explicitly retained RNA-FM encoder pretraining. Its effect cannot be assigned a favourable or unfavourable direction without a clean reinitialisation control. The row denotes our retraining protocol, not an exact reproduction of the authors' published training recipe.

Table 5: Coverage and fidelity for the configurations in the main comparison. All contains 56 RNAs; Multi is the 26-RNA stratum of sequences with more than one distinct BPM state. K differs across rows. RNAstructure uses ProbKnot.

			All		Multi	
Method	T	K	Cov	Fid	Cov	Fid
RiNALMo	—	1	0.523	0.532	0.549	0.567
RNA-FM	—	1	0.373	0.379	0.348	0.360
RNAstructure	—	1	0.641	0.647	0.590	0.603
RNAfold	—	1	0.641	0.647	0.593	0.607
RNADiffFold, retrained	20	16	0.555	0.458	0.558	0.468
DiT-BPM	10	1	0.150	0.154	0.151	0.160
DTFlow-P	1	1	0.634	0.643	0.623	0.641
DTFlow-G	2	16	0.651	0.591	0.653	0.603

Direct DiT adaptation. DiT-BPM is our adaptation of an image-style Diffusion Transformer, conditioned on RNA-FM features and using axial attention with 8×8 pair-map patches. It differs from DTFlow in backbone, representation and training configuration. The main comparison uses a BPM-trained checkpoint. The sampling sweep uses a different, DBN-trained checkpoint with full encoder fine-tuning, selected at epoch 35 with threshold 0.6 and scored against BPM references. Its checkpoint SHA-256, recorded in full in the scoring artifact, starts with `0e9efde8b424`. The two tables therefore do not form a K sweep of the same DiT checkpoint. These results are evidence about these particular transfers, not a controlled test of patchification or a bound on all image-based generators.

D.4 Interpretation and remaining evidence

The current tables give point estimates on a small, length-restricted cohort. Differences in training exposure, retained supervised weights and training objectives limit causal attribution across methods. Candidate coverage does not establish recovery of distinct conformations or thermodynamic populations. The downstream conditioning study in Section 4.5 measures base-pair agreement read from predicted coordinates. Its three-dimensional coverage comparison uses unequal candidate budgets and does not establish improved multi-state recovery.

E Supplementary results

E.1 Coverage and fidelity for the main configurations

Table 5 supplements the main coverage table with fidelity for All and Multi under the shared BPM reference protocol in Section B.3. Higher coverage at larger K is not equivalent to higher per-candidate agreement or to independently recovered conformations.

E.2 RNA-FM backbone: direct and two-stage training

We additionally train DTFlow with RNA-FM as the sequence backbone. Direct training starts from the pretrained encoder; two-stage training starts from an RNA-FM-based predictor trained with PDB labels and RNAfold-labelled sequence mutants. Both the encoder and pair modules are fine-tuned, using 64 pair channels, a maximum length of 512 nucleotides, 20 generative epochs, seed 17, and an encoder learning rate of 10^{-5}. The sequence-loss weight is 0.11 for Both and Joint. This experiment uses uniform sequence masking; Both samples its two conditional tasks with equal probability. Each test score in Table 6 uses the epoch with the highest validation coverage in that run; bold marks column maxima.

Two-stage training gives higher logged test coverage for all three objectives. Pairing-only performs best under direct training; under two-stage training, Both has the highest test coverage, while pairing-only has the slightly higher validation score. Additional predictive training, augmentation and unequal task exposure prevent attribution to initialisation alone. Direct Both and Joint peak at the final epoch, so incomplete optimisation remains a possible explanation for their low scores. The

Table 6: Direct and two-stage DTFlow training with RNA-FM ($T = 4$, $K = 1$).

	Validation coverage		Test coverage	
Task	Direct	Two-stage	Direct	Two-stage
Pairing only	**0.545**	**0.573**	**0.493**	0.529
Both	0.212	0.571	0.137	**0.535**
Joint	0.271	0.525	0.126	0.463

Table 7: DTFlow-G (two-stage, Both): the complete recorded $T \times K$ grid. All: 56 RNAs; Multi: 26 RNAs. No checkpoint changes across rows.

		All		Multi	
T	K	Cov	Fid	Cov	Fid
1	1	0.611	0.620	0.621	0.640
1	2	0.621	0.620	0.631	0.639
1	4	0.630	0.622	0.638	0.641
1	8	0.637	0.623	0.644	0.642
1	16	0.645	0.624	0.651	0.642
2	1	0.581	0.590	0.581	0.599
2	2	0.605	0.591	0.610	0.604
2	4	0.626	0.591	0.630	0.603
2	8	0.639	0.590	0.644	0.605
2	16	0.651	0.591	0.653	0.603
4	1	0.580	0.588	0.572	0.589
4	2	0.603	0.588	0.605	0.594
4	4	0.620	0.584	0.620	0.589
4	8	0.638	0.583	0.635	0.589
4	16	0.649	0.583	0.647	0.589
8	1	0.575	0.584	0.568	0.586
8	2	0.601	0.580	0.600	0.587
8	4	0.621	0.578	0.616	0.585
8	8	0.636	0.578	0.632	0.584
8	16	0.652	0.576	0.649	0.582

initial predictor reaches validation/test coverage of $0.603/0.580$ at $T = 1$, $K = 1$; these results do not establish a single-candidate generative advantage.

These training-log scores retain the original evaluation protocol and have not been independently re-aggregated under the main tables' common BPM-plus-mask reference partition. A threshold-selection defect prevented independent calibration of the $T = 1$ and $T = 4$ readouts, so we do not interpret their difference as loss of predictive ability. This is a separate single-candidate training comparison; cross-backbone generative gains and distinct-state recovery require a matched multi-candidate evaluation.

E.3 Complete DTFlow sampling grid

Table 7 extends the main sampling analysis with both metrics on All and Multi. Within this fixed-checkpoint export, the comparison uses a common reference-grouping implementation. At $K = 16$, $T = 2$ gives the highest Multi coverage, whereas $T = 8$ gives the highest All coverage. The one-step sampler has higher fidelity. This trade-off does not establish a universally optimal step count or a validation-selected default.

E.4 Inference-time pair-to-attention feedback

We evaluate $A = t_{\text{seq}} S + g\, t_{\text{pair}} B$ with $g = 1$ (normal feedback) or $g = 0$ (feedback suppressed). The multiplier acts after adaptive modulation of the entire pair bias. Weights, source

Table 8: Feedback on ($g = 1$) versus off ($g = 0$) at $T = 2$, $K = 16$ and decoding threshold 0.15. Δ is on minus off, computed before rounding.

	Coverage			Fidelity		
Subset	On	Off	Δ	On	Off	Δ
All (56)	0.6508	0.6450	+0.0057	0.5921	0.5861	+0.0059
Single (30)	0.6479	0.6434	+0.0045	0.5812	0.5756	+0.0056
Multi (26)	0.6541	0.6469	+0.0072	0.6046	0.5983	+0.0063

Table 9: One-to-one reference matches on the 26-RNA Multi cohort. Predictions are deduplicated; common-support analysis also merges reference pair sets that become identical on shared observed cells.

	Original groups (85)		Common-support groups (80)	
τ	On	Off	On	Off
0.3	83	83	79	79
0.4	83	81	79	77
0.5	66	65	67	67
0.6	58	56	57	55
0.7	35	34	33	32
0.8	20	20	18	18
0.9	2	2	2	2

draws, time conditions, the pair track and its output head are unchanged. Both settings use the epoch-12 DTFlow-G snapshot from the main sampling grid, $T = 2$, $K = 16$ and seed 17. The replay uses the checkpoint's stored decoding threshold 0.15 for both settings, whereas the main grid uses 0.3 selected on validation. Thus Table 8 is a paired diagnostic at a separately specified operating point, not an exact replay of the main scores. At $T = 1$, the feedback term is zero and the generated candidate files are identical across settings.

With $10{,}000$ paired RNA-bootstrap replicates, the 95% percentile interval for the All coverage difference is $[-0.00002, 0.01117]$, narrowly including zero; the fidelity interval is $[0.00315, 0.00877]$. For Multi, the coverage and fidelity intervals are $[0.00199, 0.01303]$ and $[0.00261, 0.01030]$, respectively. These intervals condition on the trained checkpoint and fixed sampled candidates; they do not capture training-seed, source-resampling or family-cluster uncertainty. This intervention tests whether the trained model uses feedback, not whether retraining an architecture without feedback would produce a worse model.

To examine state recovery, we deduplicate predicted pair sets within each RNA and count the maximum one-to-one matching to reference groups with base-pair F1 at least τ (Table 9). The original Multi cohort has 85 BPM-plus-mask reference groups over 26 RNAs. A second analysis intersects observation masks within each RNA and deduplicates both reference and predicted pair sets on the common support; this leaves 80 reference pair sets, with 25 RNAs retaining multiple states. Prediction deduplication leaves the original matching counts unchanged.

Feedback adds at most two one-to-one matches across these thresholds, and paired RNA-bootstrap intervals for the mean count differences all include zero. The lowest threshold is nearly saturated in both settings. These results provide limited evidence that feedback expands distinct-reference recovery in this cohort; they do not establish equivalence or absence of an effect on conformational diversity.

E.5 Additional thermodynamic comparison

Under the same BPM scoring protocol, ViennaRNA centroid achieves All coverage of 0.656 with one candidate, exceeding DTFlow-G's 0.651 with 16 candidates; its Multi coverage is lower at 0.603 versus 0.653.

E.6 RELEASED CHECKPOINTS OUTSIDE THE MAIN TRAINING PROTOCOL

Released structure heads were not trained exclusively under our supervised split; overlap is known for some corpora and unverified for others. Exclusion is a protocol decision, not evidence that every checkpoint memorised every target. Released-versus-retrained gaps conflate initialisation, supervised exposure and optimisation, so they do not isolate the effect of leakage.

A separate comparison at $K = 16$ on the 54 sequences covered by both released RNADiffFold and DTFlow-G gives All coverage 0.660 versus 0.640, Multi coverage 0.649 versus 0.653, and All fidelity 0.567 versus 0.579, respectively. This comparison uses equal candidate counts and its own common cohort, separate from the 56-RNA main table.