# Order-Agnostic Decoding for Sample-Efficient RNA Inverse Folding

**Antonia Panescu** [1] **Shujun He** [2] **Yixuan He** [3] **Rex Ying** [1]

## Abstract

RNA inverse folding, the task of designing a sequence that folds into a target structure, is a core computational problem in RNA synthetic biology. However, current methods rely on best-of-$N$ screening to compensate for low per-sample reliability and on left-to-right autoregressive sequence generation, which locks in upstream nucleotides before downstream pairing partners are observed and cannot natively support in-painting tasks. Here, we introduce an order-agnostic decoder for 2D secondary-structure inverse design, trained under a uniform-permutation distribution over decoding orders. On the OpenKnot Round 7b 240-mer pseudoknot benchmark, our method produces five times as many perfect-structure designs per sample as the strongest autoregressive baseline, while being more sample-efficient, and recovers diversity through order randomization rather than token sampling. On the out-of-distribution Eterna100 benchmark, the same model transfers without retraining, exceeding sampling from the base model at a matched budget. To showcase in-painting capability, we perform both motif preservation and motif redesign. Together, these capabilities could enable functional aptamer and riboswitch design, and reframe sample-efficient generation as a principled alternative to the dominant best-of-$N$ regime in RNA inverse folding.

## 1. Introduction

RNA secondary structure determines function across ribozymes, viral elements, riboswitches, and aptamers. Designing a sequence that folds into a target structure is a central computational problem in synthetic biology, for example redesigning the regulatory scaffold around an aptamer ligand-binding pocket, or engineering a riboswitch around a fixed regulatory motif.

The dominant evaluation paradigm for data-driven RNA inverse folding is best-of-$N$ screening. State-of-the-art pseudoknot results (He & Sun, 2026) were obtained by drawing $K{=}98\,000$ samples per target on 256 A100 GPUs and post-hoc filtering the top candidates. In this protocol each individual sample from an autoregressive (AR) RNA decoder has low probability of reproducing the target structure, so brute-force sampling is required to find any. The same workaround scales poorly to longer or more constrained design problems, because per-sample success rates drop multiplicatively under additional constraints.

We argue that sample efficiency, not sample volume, is the right axis. Concretely, we replace the L→R AR decoder of Struct2SeQ (He & Sun, 2026) with an order-agnostic decoder trained under a uniform-permutation distribution over decoding orders, and at inference draw a fresh permutation per sample. The same trained checkpoint then supports three distinct sequence-design tasks without retraining or architectural changes (Figure 1): (i) random-permutation generation of full sequences; (ii) *motif preservation*, where we fix a structural motif to wild-type and redesign the surrounding scaffold (Watson et al., 2023; Dauparas et al., 2022); (iii) *motif redesign*, where we fix the scaffold and redesign only the motif itself, the natural setup for aptamer/riboswitch engineering (Wachsmuth et al., 2013; Penchovsky, 2014). Modes (ii) and (iii) are inaccessible under L→R decoding alone.

**Contributions.** Based on these ideas, our main contributions are:

- We adapt order-agnostic decoding to RNA inverse folding conditioned on secondary structure topology, unlocking native motif in-painting, as described above.
- Our method (1) produces $5\times$ as many perfect-structure designs per sample as the strongest published AR baseline at matched compute on the OpenKnot Round 7b 240-mer pseudoknot benchmark; (2) generates diverse designs from argmax-only inference via random order decoding, without the token-level sampling strategies used in best-of-$N$ protocols; (3) transfers to the out-of-distribution Eterna100 benchmark with no retraining or fine-tuning, solving $64/100$ puzzles with a modest

[1]Yale University [2]Texas A&M University [3]Arizona State University. Correspondence to: Antonia Panescu <antonia.panescu@yale.edu>.

*Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)*

sampling budget, under ViennaRNA-2; and (4) supports motif preservation *and* motif redesign as native inference-time operations from a single checkpoint, the latter strictly impossible under AR decoding (Figure 1).

- We provide a mechanistic explanation for the asymmetric utility of post-hoc rescue (a standard step in best-of-$N$ pipelines where near-miss designs are repaired by enumerating local nucleotide mutations): rescue helps AR decoders, whose failures are locally repairable over-pair errors, but not random-permutation decoders, whose failures are context-dependent under-pair errors that local repair cannot create.

## 2. Related work

Protein inverse design tools such as ProteinMPNN (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022) exploit a 20-letter vocabulary and abundant 3D structures; RNA's 4-letter alphabet creates a more degenerate sequence-to-structure mapping with a flatter thermodynamic landscape, so these recipes do not transfer cleanly. High-quality 3D structures are also scarce (Townshend et al., 2021), which limits both 3D structure prediction (Shen et al., 2024) and the 3D inverse-design tools (Joshi et al., 2024; Kubaney et al., 2025) that rely on folding-back evaluation against the target.

For 2D RNA inverse folding, chemical-mapping experiments (Lee et al., 2014) and learned predictors such as RibonanzaNet (He et al., 2024) make structural rewards cheap and reliable. While classical physics-based approaches (Lorenz et al., 2011) struggle with pseudoknots, and Monte-Carlo-based approaches (Yang et al., 2017) are slow, reinforcement-learning approaches (LEARNA / MetaLEARNA (Runge et al., 2019), RNA-DCGen (Hossain et al., 2024)) and the recent Struct2SeQ (He & Sun, 2026) have closed much of this gap. Struct2SeQ in particular handles pseudoknotted topologies and matches human Eterna players on OpenKnot Round 7, but it is fundamentally a left-to-right AR model, which limits per-sample reliability and rules out partial-sequence conditioning (in-painting). Our work removes this limitation while keeping Struct2SeQ's deep-$Q$-learning training signal.

## 3. Method

**Setup.** We adopt the encoder-decoder formulation, RibonanzaNet-SS-derived dense per-position reward, twice-shifted deep-$Q$-learning training with experience replay, and Hungarian-matched Jaccard scoring of Struct2SeQ (He & Sun, 2026; He et al., 2024; Mnih et al., 2015); we refer the reader there for full details.

**Algorithm 1** Random-permutation argmax decoding.

1: **Input:** target structure $S$, length $L$, policy $\pi_\theta$.
2: Sample $\sigma \sim \mathrm{Uniform}(S_L)$.
3: Initialise output $y \leftarrow [\perp]^L$, KV cache $\mathcal{K} \leftarrow \varnothing$.
4: **for** $t = 0, \ldots, L-1$ **do**
5: $\quad q \leftarrow \pi_\theta(\sigma_t \mid y, S, \sigma, \mathcal{K})$ {$Q$-values at position $\sigma_t$}
6: $\quad q \leftarrow q + \mathrm{Mask}(\sigma_t, y, S)$ {partner / repeat / A-budget masks}
7: $\quad y_{\sigma_t} \leftarrow \arg\max q$; update $\mathcal{K}$.
8: **end for**
9: **return** $y$.

**Algorithm 2** In-painting via fixed-mask pre-fill.

1: **Input:** target $S$, length $L$, fixed mask $f \in (\{\perp\} \cup \{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\})^L$, policy $\pi_\theta$.
2: **Pre-fill:** initialise $y \leftarrow f$ at all positions where $f_i \neq \perp$. {Fixed nucleotides visible to partner-mask from step 0}
3: Sample $\sigma \sim \mathrm{Uniform}(S_L)$.
4: **for** $t = 0, \ldots, L-1$ **do**
5: $\quad$ Decode $\sigma_t$ as in Algorithm 1, line 5–6.
6: $\quad$ **if** $f_{\sigma_t} \neq \perp$ **then**
7: $\quad\quad y_{\sigma_t} \leftarrow f_{\sigma_t}$ {teacher-force; KV cache stays consistent}
8: $\quad$ **else**
9: $\quad\quad y_{\sigma_t} \leftarrow \arg\max q$.
10: $\quad$ **end if**
11: $\quad$ Update $\mathcal{K}$.
12: **end for**
13: **return** $y$.

**Architecture and training.** The encoder follows He & Sun (2026) unchanged. We replace its LSTM-PE plus causal-conv decoder with a pure Transformer decoder using a learned relative-position bias on the self- and cross-attention scores, keyed on the decoding-order permutation $\sigma$ (Figure 2). This decouples position from recurrence, letting any $\sigma$ share parameters. For each training example of length $L$ we draw $\sigma \sim \mathrm{Uniform}(S_L)$; the causal mask is in step order, not RNA-position order. We continue training from the released Struct2SeQ checkpoint for 5 additional episodes (5 epochs each) on 4×A100 80 GB GPUs, with bf16 mixed precision, FlashAttention / memory-efficient SDPA kernels, and `torch.compile` yielding a $\sim$7× training-throughput improvement compared to He & Sun (2026). Algorithm 1 is the inference procedure.

**Native in-painting vs. teacher-forced AR.** The pre-fill in Algorithm 2 (line 2) writes constrained nucleotides into the output buffer before the decode loop starts. Subsequent decode steps therefore read these identities through the partner-mask irrespective of their position in $\sigma$. Setting $f$ to wild-type at motif positions yields motif preservation;

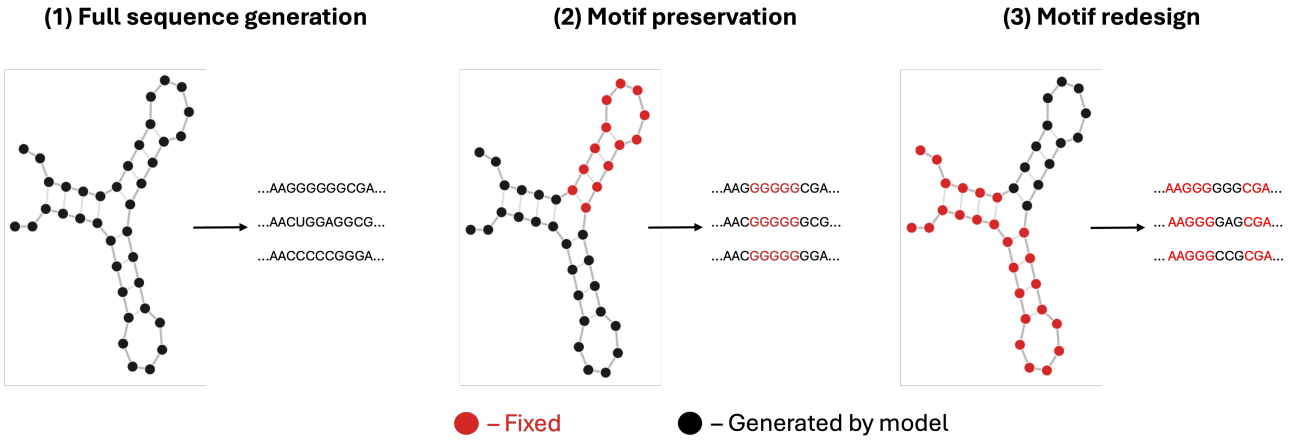



*Figure 1.* **One order-agnostic decoder, three inverse-design modalities.** Because the decoder is trained to generate nucleotides in an arbitrary order, a single model can condition on *any* subset of fixed positions and design the remainder, unifying three tasks under one network. *(1) Full design:* every position is generated to fold the sequence into the target structure. *(2) Motif preservation:* the nucleotides of a known functional motif (e.g. an aptamer or riboswitch element) are held fixed at their wild-type (WT) identities while the surrounding scaffold is generated around them. *(3) Motif redesign:* the flanking context is held fixed and only the motif is regenerated. This natively requires conditioning on fixed sequence *both* upstream and downstream of the redesigned region, and is therefore not expressible by left-to-right autoregressive generation.

setting $f$ to wild-type at scaffold positions yields motif redesign. An L→R AR decoder with teacher-forced motif positions $[a, b]$ cannot replicate this: at decode steps $t < a$, the motif positions have not yet been visited and are absent from the KV cache, so upstream scaffold tokens are sampled without the motif identity as conditioning context; only at $t > b$ does the overridden motif enter the KV cache and condition downstream sampling. For pseudoknotted targets with long-range pairs crossing the motif boundary, this is particularly limiting. Table 1 reports a $1.4\times$ perfect-Jaccard ratio in our favour on motif preservation.

## 4. Experiments

**OpenKnot240 Benchmark.** We evaluate on the 20 OpenKnot Round 7b 240-mer pseudoknot puzzles (Eterna Game, 2024a; Lee et al., 2014). Each method generates $K{\sim}1000$ samples per target. Sequences are scored identically across all rows: predicted secondary structure from RibonanzaNet-SS (He et al., 2024) → Hungarian-matched base-pair set → Jaccard against the target. We report perfect-Jaccard rate (fraction of samples with Jaccard $= 1.0$) and puzzles solved ($\geq 1$ perfect at $K{=}1000$) on every row.

**Headline results.** The main comparison in Table 1 tests our central sample-efficiency claim. At matched $K{\sim}1000$, our random-permutation argmax decoder produces $33.24\%$ perfect designs and solves $13/20$ puzzles. The best published AR protocol (3-strategy + rescue, faithful to the Struct2SeQ decoding strategy; see Table 1 caption for the protocol definition) reaches $6.64\%$ / $17(/20)$: per-sample reliability is $5.0\times$ higher for our model, while AR's coverage lead of 4 puzzles comes entirely from rescue (§5). Under matched protocol, the per-sample ratio is $2.0\times$ in our favour and AR leads coverage by 2 puzzles.

**Architecture vs. decoding order.** To isolate the contribution of decoding order from architecture and additional training, we run a same-checkpoint ablation. Holding weights and architecture fixed, L→R + argmax reaches $23.72\%$ / 10 and random-perm + argmax reaches $33.24\%$ / 13 ($+9.52$ pp from order alone). Holding decoding order fixed at L→R + argmax, the original 10-episode checkpoint reaches $14.35\%$ / 5 versus our $23.72\%$ / 10 ($+9.37$ pp from architecture plus 5 extra training episodes; we control for this in the next section). Order and architecture+training each contribute roughly half of the total perfect-rate gain.

**Controlling for training budget.** The architecture comparison above is confounded by the five additional episodes our checkpoint receives. To remove this confound, we continue the *original* Struct2SeQ for five more episodes in its own regime: resuming from its released episode-10 checkpoint and using the same training data, reward (RibonanzaNet-SS structure match), architecture, learning rate, and converged exploration rate. We see that further training does not help: the 15-episode AR baseline is uniformly *worse* than at 10 episodes (Table 1), indicating that the AR model has saturated. At this matched budget, our random-permutation decoder still solves $15/20$ versus $12/20$, so the gain is not an artefact of training duration. The fact that additional AR training is *counterproductive*

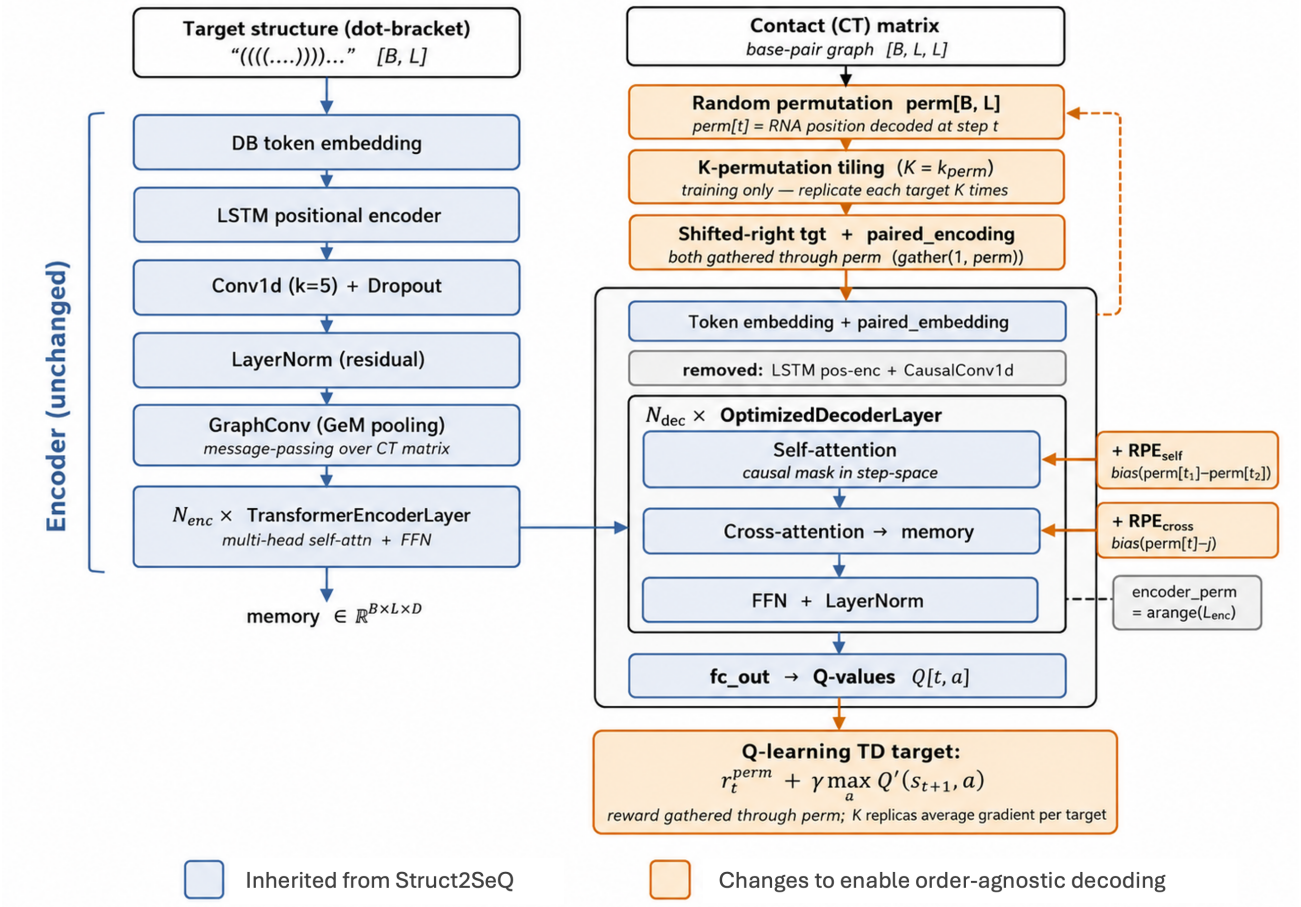



*Figure 2.* **Order-agnostic decoder architecture.** The structure-conditioned decoder predicts nucleotides in a randomly permuted order, conditioning on the target structure and the already-decoded positions. The network is trained by deep Q-learning against a structure-match reward.

further suggests that training order-agnostically from the outset, rather than only at inference, would widen the gap, which we leave to future work.

**Per-strategy AR ablation.** We next decompose the 3-strategy protocol's coverage gain over argmax-only decoding. Reporting each strategy alone at $K = 333$, the two $\epsilon$-greedy strategies each recover most of the gain on their own, solving 12 and 11 puzzles for original Struct2SeQ AR (vs. 5 for argmax-only), and 13 and 14 for our random-perm checkpoint, whereas $Q$-softmax recovers almost none (2 and 3). $Q$-softmax's samples are near-worthless (0.07% perfect) because $Q$-values are not calibrated probabilities (He & Sun, 2026, § 2.7): sampling a softmax over them drifts off the argmax peak. Such samples pay off only at extreme budgets: 0.07% is $\sim$ 70 perfects per target at $K = 98{,}000$ but $\sim$ 0.2 at $K = 333$. Token-level sampling thus buys diversity at the cost of correctness, motivating a diversity mechanism that leaves the token distribution untouched.

**Diversity engine.** If random decoding order alone generates sample diversity, token-level sampling becomes optional. We test this by counting unique sequences across $K$=1000 samples per target. Random-permutation + argmax produces 20,032 unique sequences, one per sample. The 3-strategy AR protocol of Struct2SeQ is the analogous diversity mechanism on the L→R side ( argmax-only L→R is naturally deterministic, producing only ∼17 unique sequences per target on the same budget) but it injects token-level noise. Random-perm therefore composes with argmax instead of requiring token-level sampling to recover diversity, which is why our argmax-only result outperforms every other single configuration in Table 1.

**In-painting.** A single random-permutation checkpoint performs both motif preservation and motif redesign as native inference-time operations. For each target we extract a structural-element inventory—hairpins, internal loops, multi-loops, and pseudoknot stems—via a stem-grouping algorithm over OpenKnotScorePipeline's pair-crossing classification (Eterna Game, 2024b), and draw one motif per

*Table 1.* **OpenKnot Round 7b 240-mer pseudoknot benchmark.** Perfect-Jaccard % is the fraction of designs reproducing the target exactly (Hungarian on RibonanzaNet-SS predicted base-pair probabilities, $\theta$=0.5). Puzzles solved is the count of targets with $\geq 1$ perfect-Jaccard design at $K$=1000. We report the original Struct2SeQ both at its released 10-episode checkpoint and continued for 5 additional episodes (15 total) in its *own* published recipe, data, reward, and exploration schedule; "Ours" is our order-agnostic Transformer-decoder variant, also trained for 15 episodes (10 + 5). The 3-strategy protocol is $\epsilon$-greedy with $p$=0.05, $\epsilon$-greedy with $p$=0.10, and $Q$-softmax sampling at $p$=1.0, each with $K\approx$333, matching the protocol of He & Sun (2026). Rescue is the post-hoc $4^k$ local repair on the per-puzzle best non-perfect seed when $|\text{diff_pos}| \leq 4$.

| Method | Perfect-Jaccard (%) | Puzzles solved |
|---|---|---|
| *Plain generation. Original Struct2SeQ AR (L→R, 10 episodes)* | | |
| argmax-only | 14.35 | 5/20 |
| 3-strategy | 6.55 | 13/20 |
| 3-strategy + rescue | 6.64 | 17/20 |
| *Plain generation. Original Struct2SeQ AR (L→R, 10 + 5 episodes)* | | |
| argmax-only | 5.62 | 3/20 |
| 3-strategy | 3.23 | 12/20 |
| 3-strategy + rescue | 3.23 | 12/20 |
| *Plain generation. Ours (bidirectional, 10 + 5 episodes)* | | |
| L→R argmax-only | 23.72 | 10/20 |
| L→R + 3-strategy | 9.38 | 12/20 |
| L→R + 3-strategy + rescue | 9.13 | 13/20 |
| **Random-perm argmax-only** | **33.24** | **13/20** |
| Random-perm + 3-strategy | 13.65 | 14/20 |
| Random-perm + 3-strategy + rescue | 13.44 | 15/20 |
| *In-painting. Ours (random-perm) and AR teacher-forced reference* | | |
| **Motif preservation** | **25.69** | **13/20** |
| Motif preservation + rescue | 25.28 | 13/20 |
| Motif redesign (AR-impossible natively) | 19.82 | 13/20 |
| *In-painting. Ours L→R + teacher-forced motif (architecture vs. order ablation)* | | |
| L→R + teacher-forced motif (argmax) | 18.41 | 11/20 |
| L→R + teacher-forced motif + 3-strategy | 7.60 | 12/20 |
| L→R + teacher-forced motif + 3-strategy + rescue | 7.31 | 12/20 |
| L→R + teacher-forced motif (argmax) + rescue | 18.13 | 11/20 |

sample uniformly at random (sizes capped at 50% of $L$, mean $\sim$ 15%). Motif preservation reaches 25.69% / 13 at $K$ = 1000 and redesign 19.82% / 13 on the same inventory, with fixed positions held exactly in 100% of a 500-sequence audit. Crucially, an L→R decoder can mimic preservation by teacher-forcing the motif, a workaround our decoder still beats by 1.40× on per-sample reliability, but redesign is structurally impossible for the AR paradigm, since it commits scaffold positions before reaching the motif.

**Eterna100: out-of-distribution.** To test whether order-agnostic decoding generalizes beyond pseudoknot design, we evaluate on the Eterna100 benchmark (Anderson-Lee et al., 2016; Koodli et al., 2021), which consists of 100 secondary-structure synthetic puzzles spanning a wide difficulty range. Unlike OpenKnot, Eterna100 targets are pseudoknot-free, so we score under the benchmark's standard oracle, ViennaRNA-2 (Lorenz et al., 2011): a design solves a puzzle iff its minimum-free-energy structure matches the target exactly (base-pair Jaccard = 1.0). This oracle is independent of the RibonanzaNet-SS reward used in training, making Eterna100 a genuinely out-of-distribution test under a different folding model.

Order-agnostic decoding transfers without retraining: it solves 52/100 puzzles at $K$=1000 and 64/100 at $K$=10 000, beating the original Struct2SeQ 3-strategy protocol at both budgets (45 and 50; Table 2). The sharp budget scaling (52 → 64) indicates these misses are within the model's reach but unsampled at low budget—a search limit rather than a capacity one. Consistent with this, post-hoc rescue helps little (+1 at each budget): most residual failures are distributed near-misses rather than the locally repairable errors rescue corrects.

Beyond drawing more samples, the two most natural routes to closing this gap are to specialize the model to each target and to remove search-irrelevant (albeit physiologically-informed) sequence generation constraints; we test both. Per-target fine-tuning (Appendix A) is the natural test-time alternative to brute-force screening; across a portfolio of folding-aware rewards it lifts coverage from 52 to 62/100 at $K$=1000 (63 with rescue). Separately, the training pipeline inherits synthesizability caps (limits on stretches of single repeated base and adenine-content limits) that aid wet-lab synthesis but are irrelevant to folding in this computational benchmark, and bias designs toward high GC content; dis-

*Table 2.* **Eterna100 secondary-structure design benchmark**: a design solves a puzzle iff its minimum-free-energy structure matches the target exactly (base-pair Jaccard = 1.0). We report puzzles solved (out of 100) at two sampling budgets, scored under ViennaRNA 2. The original Struct2SeQ uses the 3-strategy autoregressive protocol, whereas "Ours" generates by random permutation. Rescue denotes the post-hoc $4^k$ local repair procedure ($|\mathrm{diff_pos}| \leq 4$). Per-target fine-tuning (FT) briefly continues training on a single target before sampling. The synthesis-constraint rows disable the homopolymer-run and adenine-content caps, which are relevant for synthesis, but not folding. †Result not yet available under ViennaRNA-2 scoring.

| Method | Solved ($K = 1000$) | Solved ($K = 10{,}000$) |
|---|---|---|
| *Original Struct2SeQ AR ($L \to R$, 10 episodes)* | | |
| 3-strategy | 45/100 | 50/100 |
| 3-strategy + rescue | 49/100 | 53/100 |
| *Original Struct2SeQ AR ($L \to R$, 10 + 5 episodes)* | | |
| 3-strategy | 51/100 | —† |
| 3-strategy + rescue | 54/100 | —† |
| *Ours (order-agnostic, 10 + 5 episodes), synthesis constraint ON* | | |
| **Random-perm** | **52/100** | **64/100** |
| Random-perm + rescue | 53/100 | 65/100 |
| Random-perm + per-target FT | 62/100 | 64/100 |
| Random-perm + per-target FT + rescue | 63/100 | —† |
| *Ours (order-agnostic), synthesis constraint OFF* | | |
| **Random-perm** | **59/100** | 64/100 |
| Random-perm + rescue | 62/100 | 64/100 |
| Random-perm + per-target FT | 63/100 | 65/100 |

abling them recovers exactly those GC-biased failures, lifting the base model at no added compute.

Both gains, however, are confined to the low-budget regime. At $K{=}10\,000$ the fine-tuned, constraint-free, and base models all converge to $64/100$ (rescue, 65): each intervention reaches this coverage at a smaller budget, but none exceeds plain sampling from the base model. We analyze why test-time adaptation saturates here, and what would move the ceiling, in §5.

## 5. Discussion

**Per-sample reliability vs. coverage Pareto.** Per-sample reliability and puzzle coverage are distinct operating points. Prior evaluations have favored coverage at large sampling budgets, leaving per-sample reliability under-reported. Best-of-$N$ at $K{=}98\,000$ optimizes *coverage*; our random-perm checkpoint optimizes *per-sample reliability* ($33.24\%$ of single samples perfect at $K{=}1000$, with no rescue or sampling temperature), while still being able to offer both extremes from one checkpoint (reliability-optimized: $33.24\%$ / 13; coverage-optimized: $13.44\%$ / 15). Per-sample reliability is the axis that determines wet-lab synthesis cost and constrained-design feasibility, especially as design problems get longer or more challenging.

**Order randomisation as a diversity engine.** Random-permutation decoding obtains diversity from the decoding order itself: distinct $\sigma$'s induce distinct conditioning paths through the same trained policy, producing $1000$ distinct argmax outputs from $K{=}1000$ samples without any token-level randomness. The 3-strategy AR protocol used by Struct2SeQ ($\epsilon$-greedy at two values plus $Q$-softmax) achieves the analogous outcome via a different mechanism: injecting token-level noise into the decoder, but at a per-sample reliability cost. Order randomisation is orthogonal to per-sample reliability rather than substituting for it.

**Reward alignment versus more sampling.** More training and more sampling both reliably raise coverage, so the question is whether aligning the model to the benchmark's native reward would outperform increasing the sampling budget. We test this with per-target fine-tuning against a portfolio of folding-aware rewards (Appendix A). At $K{=}1000$ this appears to be the case: per-target fine-tuning adds 10 solved puzzles over the base model, and constraint relaxation and rescue add 7 and 1. However, at $K{=}10\,000$, fine-tuning matches the base model's $64/100$ exactly, and the other interventions converge as well ($64$ and $65$). Reward alignment thus reaches the coverage that more sampling reaches on its own, not beyond it, and at higher cost: fine-tuning spends $\sim 10^4$ training generations per target, none of it shared across targets. We read this as evidence that the bottleneck is the base model being stuck in local minima, not the reward. The model is pretrained on a large genome-derived corpus far from the hand-designed Eterna100 puzzles, and a few per-target episodes cannot realign a model with that much pretraining behind it. Closing the gap on this benchmark therefore likely calls for realigning the *pretraining* to the benchmark (Section 6), or a more principled search, rather than further reward engineering.

**Why rescue is one-sided: over-pair vs. under-pair failures.** Symmetric rescue ($4^k$ local enumeration on the per-puzzle best non-perfect seed) reaches Jaccard $= 1.0$ on $4/5$ AR seeds but $0/6$ random-perm seeds. This is failure-mode asymmetry. AR's narrow exploration produces *over-pair* errors (an extra predicted pair the target lacks), which rescue removes by breaking Watson-Crick complementarity. Random-perm produces *under-pair* errors: the target wants a pair the network refuses to predict from the surrounding $\sim$200 bases of context, and no local mutation creates one. Manual verification on puzzle 12 confirms all 16 mutations of the relevant G-C pair fail (max Jaccard $0.984$); multi-seed model-guided in-paint rescue (top-5 seeds, $K{=}100$) hits the same ceiling. While rescue is a useful tool, its utility tracks failure-mode locality, which is set by decoding regime.

## 6. Limitations and conclusion

**Limitations and future work.** Our evaluation is computational throughout: predicted secondary structure serves as the design oracle, with experimental validation deferred to follow-up; 3D-conditioned and joint 2D/3D generation are complementary directions with their own data bottlenecks. Within our current scope, four observations shape the path forward:

- *The residual gap is a last-pair problem.* On both benchmarks, the unsolved puzzles are near-misses: the seven unsolved OK7b puzzles are under-pair errors (max Jaccard $0.984$), and the unsolved Eterna100 puzzles are likewise high-Jaccard distributed near-misses, typically a single pair short of perfect, so the remaining work is precise local correction.
- *The pretraining corpus is largely out-of-distribution relative to Eterna100.* The model is trained against a RibonanzaNet-SS reward yet generalizes to Vienna-scored Eterna100, so part of the gap is likely a mismatch that *pretraining-time* alignment would close, where test-time fine-tuning cannot.
- *The repair step is a useful but unprincipled heuristic.* The $4^k$ rescue enumerates local mutations blindly and structurally cannot create the under-pair errors that dominate our failures. Because the order-agnostic decoder can fix any subset of positions and regenerate the rest, it is strictly more general than a learned mutation engine, and a trained, position-selecting repair that chooses *where* to intervene is a natural next step.
- *Sampling budget is compute-limited.* The experiments outlined in this paper ran on four A100 GPUs at $K{\leq}10\,000$. We expect scaling to larger computational budgets to enable increased sampling, resulting in lifting both benchmarks further, as is suggested by the $52 \rightarrow 64$ Eterna scaling showing headroom remains. However, on Eterna puzzles, the out-of-distribution gap described above may cap gains until pretraining is aligned to the benchmark's native oracle and training data.

**Conclusion.** Order-agnostic decoding for RNA inverse folding delivers $5\times$ higher per-sample reliability than autoregressive baselines on the OpenKnot 240-mer pseudoknot benchmark, generalizes out-of-distribution to a second oracle, and supports motif preservation and motif redesign natively from a single checkpoint. These capabilities target aptamer and riboswitch engineering: fix the binding pocket as the constrained motif and redesign the surrounding regulatory scaffold (or vice versa) via fixed-mask pre-fill, then validate by chemical mapping (SHAPE) (Merino et al., 2005) and in-vitro folding. More broadly, our results argue for measuring and improving RNA inverse-folding methods on per-sample reliability, as opposed to traditional best-of-$N$ coverage.

## Impact Statement

This work targets RNA inverse folding for pseudoknot-bearing targets, with downstream relevance to mRNA therapeutic design (Damase et al., 2021; Zhang et al., 2023), riboswitch and aptamer engineering (Wachsmuth et al., 2013; Penchovsky, 2014; Wayment-Steele et al., 2022). We choose 2D secondary-structure design because chemical-mapping experiments make experimentally-grounded structural feedback abundant in a way 3D crystallographic data is not (Lee et al., 2014; He et al., 2024), allowing the resulting neural oracles to be incorporated directly into the design loop. By raising per-sample reliability and lowering the compute requirement for high-quality candidates ($\sim$100$\times$ less per target than the published Struct2SeQ baseline at matched coverage), the method reduces both compute and wet-lab cost, lowering the barrier for academic labs to participate in RNA design challenges. We see no immediate dual-use concern beyond those generic to nucleic-acid design tools; downstream synthesis is gated by physical reagent access. Interactive deployment on platforms such as Eterna (Lee et al., 2014; Wayment-Steele et al., 2022) is a promising longer-term direction: order-agnostic decoding is a natural fit for a setting in which a human fixes an arbitrary subset of nucleotides and the model fills in the rest in real time, placing the design tool in a transparent collaborative role rather than as an opaque generative oracle.

## References

Anderson-Lee, J., Fisker, E., Kosaraju, V., Wu, M., Kong, J., Lee, J., Lee, M., Zada, M., Treuille, A., and Das, R. Principles for predicting RNA secondary structure design difficulty. *Journal of Molecular Biology*, 428(5, Part A):

748–757, 2016. doi: 10.1016/j.jmb.2015.11.013.

Damase, T. R., Sukhovershin, R., Boada, C., Taraballi, F., Pettigrew, R. I., and Cooke, J. P. The limitless future of RNA therapeutics. *Frontiers in Bioengineering and Biotechnology*, 9:628137, 2021. doi: 10.3389/fbioe.2021.628137.

Dauparas, J., Anishchenko, I., Bennett, N., others, and Baker, D. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Eterna Game. OpenKnot AI Design Data. https://github.com/eternagame/OpenKnotAIDesignData, 2024a. GitHub repository.

Eterna Game. OpenKnotScorePipeline. https://github.com/eternagame/OpenKnotScorePipeline, 2024b. GitHub repository.

He, S. and Sun, Q. Struct2SeQ: RNA inverse folding with deep Q-learning. *bioRxiv*, 2026. doi: 10.64898/2026.01.16.700031.

He, S., Huang, R., Townley, J., Quarmby, C., others, Wayment-Steele, H. K., and Das, R. Ribonanza: deep learning of RNA structure through dual crowdsourcing. *bioRxiv*, 2024. doi: 10.1101/2024.02.24.581671.

Hossain, H. S. et al. RNA-DCGen: Dual constrained RNA sequence generation with LLM-attack. *bioRxiv*, 2024. doi: 10.1101/2024.09.23.614570.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/hsu22a.html.

Joshi, C. K., Jamasb, A. R., Viñas, R., Harris, C., Mathis, S. V., Morehead, A., Anand, R., and Liò, P. grnade: Geometric deep learning for 3D RNA inverse design. In *ICLR 2024 Generative and Experimental Perspectives for Biomolecular Design Workshop*, 2024.

Koodli, R. V., Rudolfs, B., Romano, J., Wayment-Steele, H. K., Dunlap, W. A., Eterna Participants, and Das, R. Redesigning the Eterna100 for the Vienna 2 folding engine. *bioRxiv*, 2021. doi: 10.1101/2021.08.26.457839.

Kubaney, A., Favor, A., McHugh, L., Mitra, R., Pecoraro, R., Dauparas, J., Glasscock, C., and Baker, D. RNA sequence design and protein–DNA specificity prediction with NA-MPNN. *bioRxiv*, 2025. doi: 10.1101/2025.10.03.679414.

Lee, J., Kladwang, W., Lee, M., Cantu, D., others, Treuille, A., and Das, R. RNA design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, 111(6):2122–2127, 2014. doi: 10.1073/pnas.1313039111.

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6:26, 2011. doi: 10.1186/1748-7188-6-26.

Merino, E. J., Wilkinson, K. A., Coughlan, J. L., and Weeks, K. M. RNA structure analysis at single nucleotide resolution by selective 2'-hydroxyl acylation and primer extension (SHAPE). *Journal of the American Chemical Society*, 127(12):4223–4231, 2005. doi: 10.1021/ja043822v.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., others, and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Penchovsky, R. Computational design of allosteric ribozymes as molecular biosensors. *Biotechnology Advances*, 32(5):1015–1027, 2014. doi: 10.1016/j.biotechadv.2014.05.005.

Runge, F., Stoll, D., Falkner, S., and Hutter, F. Learning to design RNA. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=ByfyHh05tQ.

Shen, T., Hu, Z., Sun, S., Liu, D., Wong, F., Wang, J., Chen, J., Wang, Y., Hong, L., Xiao, J., et al. Accurate RNA 3D structure prediction using a language model-based deep learning approach. *Nature Methods*, 21(12):2287–2298, 2024. doi: 10.1038/s41592-024-02487-0.

Townshend, R. J. L., Eismann, S., Watkins, A. M., Rangan, R., Karelina, M., Das, R., and Dror, R. O. Geometric deep learning of RNA structure. *Science*, 373(6558):1047–1051, 2021. doi: 10.1126/science.abe5650.

Wachsmuth, M., Findeiß, S., Weissheimer, N., Stadler, P. F., and Mörl, M. De novo design of a synthetic riboswitch that regulates transcription termination. *Nucleic Acids Research*, 41(4):2541–2551, 2013. doi: 10.1093/nar/gks1330.

Watson, J. L., Juergens, D., Bennett, N. R., others, and Baker, D. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Wayment-Steele, H. K., Wu, M., Gotrik, M., and Das, R. Crowdsourced RNA design discovers diverse, reversible, efficient, self-contained molecular switches. *Proceedings of the National Academy of Sciences*, 119(18):e2112979119, 2022. doi: 10.1073/pnas.2112979119.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Yang, X., Yoshizoe, K., Taneda, A., and Tsuda, K. RNA inverse folding using Monte Carlo tree search. *BMC Bioinformatics*, 18(1):468, 2017. doi: 10.1186/s12859-017-1882-7.

Zhang, H., Zhang, L., Lin, A., Xu, C., Li, Z., Liu, K., others, Mathews, D. H., and Huang, L. Algorithm for optimized mRNA design improves stability and immunogenicity. *Nature*, 621(7978):396–403, 2023. doi: 10.1038/s41586-023-06127-z.

## A. Per-target fine-tuning for Eterna100 alignment

For each Eterna100 target left unsolved by the base model, we briefly continue training the shared base policy on that *single* target before sampling: starting from the base checkpoint, we run 20 episodes of the same update rule on the lone target (exploration annealed $p\colon\ 0.6 \to 0.1$, learning rate $10^{-3}$), then decode as in Algorithm 1. This is test-time adaptation: each target yields its own checkpoint, at a cost of $\sim 10^4$ generations per target, not amortized. Table 3 lists the folding-aware rewards we evaluate; each solves a partially distinct subset of targets, so the $K{=}1000$ figure in Section 4 is their union. As noted there, this advantage does not persist at $K{=}10{,}000$.

*Table 3.* **Per-target fine-tuning reward portfolio.** $x$: a designed sequence; $p_i^\star$ / $\hat{p}_i$: target / predicted pairing state at position $i$; $E(x, S)$: free energy of $x$ constrained to fold into $S$; $E_{\mathrm{MFE}}(x){=}\min_S E(x, S)$; $\hat{S}(x){=}\arg\min_{S\neq S^\star} E(x, S)$ the lowest-energy competing structure. Structure (per-position) rewards train a DQN; energy (scalar) rewards use REINFORCE (Williams, 1992).

| Reward | Definition | Optimizer | Enforces |
|---|---|---|---|
| Structure match | $r_i = \mathbb{1}[\hat{p}_i = p_i^\star]$ | DQN | correct topology |
| Weighted structure match | $r_i \in \{\, {+}10 \text{ (pair)},\ {+}1 \text{ (loop)},\ {\leq}0 \text{ (wrong)} \,\}$ | DQN | base pairs over loops |
| Free-energy gap | $R(x) = -\big(E(x, S^\star) - E_{\mathrm{MFE}}(x)\big)$ | REINFORCE | $S^\star$ is an MFE fold |
| Negative-design margin | $R(x) = E\Big(x, \hat{S}(x)\Big) - E(x, S^\star)$ | REINFORCE | $S^\star$ is the *unique* MFE |