A REASONING-INFORMED VERIFIER-GUIDED EDITING AGENT FOR RNA INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA function follows from RNA structure, so designing sequences that reliably fold into a prescribed shape is central to RNA therapeutics and synthetic biology. Inverse folding is difficult because the sequence-to-structure map is highly degenerate and rugged, candidate quality is observable only after folding, and optimizing against a single folding oracle invites overfitting. Existing designers emit a sequence in one open-loop pass, provide no rationale for individual nucleotide choices, and accept candidates through hand-tuned thresholds. We present RIVET, which recasts RNA inverse folding as verifier-guided, model-based reinforcement learning over interpretable edits. Instead of emitting a sequence directly, RIVET repairs one through a propose, imagine, select, verify, and refold loop built from four modules. SKETCH is a reasoning policy that proposes typed, biophysically grounded edits with explicit rationales and is trained by group-relative policy optimization. ECHO is a learned world model that ranks edits without folding. SCOUT forwards only the most promising edits to costly verification. an energy-based verifier that turns folding feasibility into a calibrated dense reward and conformal accept-reject gate. Two guarantees make the loop safe: potential-based shaping provably cannot bias the design objective, and the gate bounds false acceptance at a chosen rate. The same loop supports tertiary-structure design, secondary self-consistency, and binding-conditioned design for RNA-ligand and RNA-RBP targets. Under a multi-tool protocol whose reporting predictors are disjoint from the loop-internal folder, RIVET attains state-of-the-art native recovery of 56.3%, the best tertiary and secondary self-consistency, and generalizes robustly to held-out RNA-Puzzles and CASP15 targets. Our code is available at `https://anonymous.4open.science/r/RIVET-4517/`.

1 INTRODUCTION

RNA function is coupled to three-dimensional structure. Riboswitches sense metabolites through ligand-binding pockets, ribozymes catalyze reactions via tertiary contacts, and aptamers recognize targets through specific interfaces (Cech & Steitz, 2014; Warner et al., 2018). This coupling makes RNA a therapeutic and synthetic-biology modality (Damase et al., 2021) and motivates RNA inverse folding, in which a sequence is designed to fold into a prescribed structure. Because many functional elements, including binding pockets, A-minor motifs, and base triples, are defined by tertiary geometry rather than base pairing alone, design must condition on three-dimensional backbones instead of only secondary-structure diagrams (Joshi et al., 2025; Huang et al., 2024).

Three properties make the problem hard. The sequence to structure map is degenerate and rugged, so native recovery is an imperfect proxy for structural fidelity (Hu et al., 2026). Candidate quality becomes observable only after folding, which makes feedback expensive and delayed. Solved RNA tertiary structures are scarce relative to proteins (Adamczyk et al., 2022), and folding predictors disagree, so a design tuned to one oracle may fail under another (Shen et al., 2024; Wang et al., 2023; Passaro et al., 2025). Prior work addresses parts of this but not their interaction. Classical solvers search sequence space under thermodynamic secondary-structure models (Hofacker et al., 1994; Zadeh et al., 2011; Kleinkauf et al., 2015; Yang et al., 2017; Rubio-Largo et al., 2024), learned policies improve that search (Runge et al., 2018; Eastman et al., 2018; Runge et al., 2024; Li et al., 2025), and tertiary designers condition on backbones through geometric encoders, autoregressive or diffusion decoders, and RNA foundation models (Tan et al., 2024; Joshi et al., 2025; Huang et al.,

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.

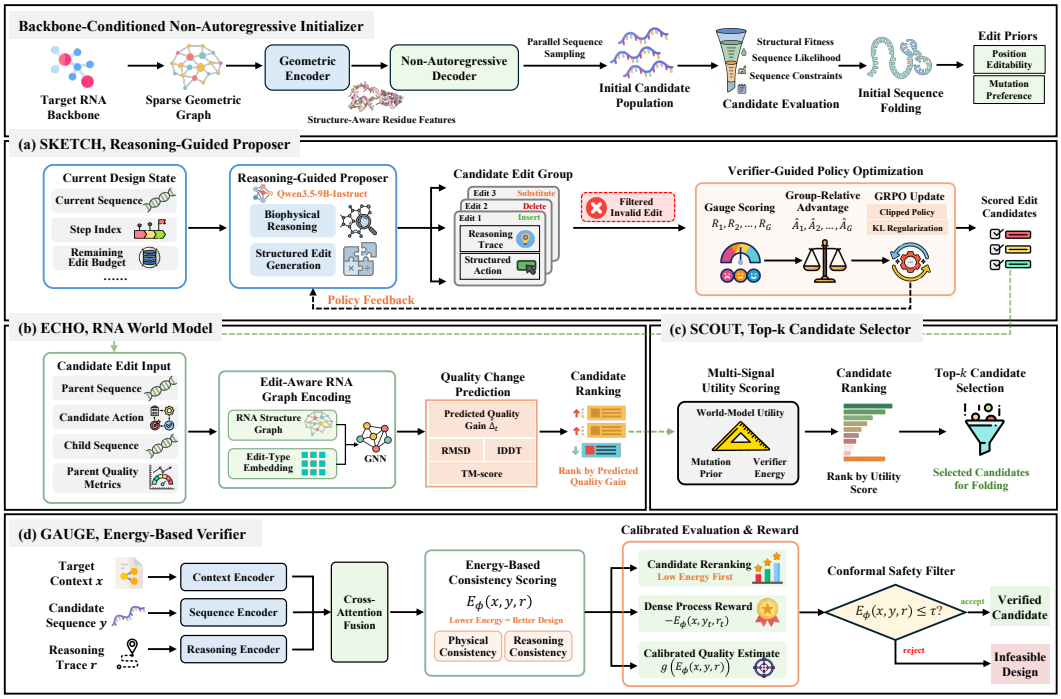


Figure 1: **Architecture of the RIVET modules.** (a) SKETCH generates initial candidates via encoding and non-autoregressive decoding, with a proposer for structured edits. (b) ECHO predicts changes using edit-aware graph encoding without folding simulation. (c) SCOUT ranks candidates through utility scoring for top-k selection. (d) rforms verification with calibrated evaluation, filtering infeasible designs through conformal safety guarantees.

2024; Kim et al., 2026; Tan et al., 2025; Bibekar et al., 2025; Zhang et al., 2024). Reward-driven fine-tuning has also been used to optimize self-consistency (Hu et al., 2026). These systems remain open-loop and opaque: they emit a sequence in one pass without inspecting its folded state, offer no rationale checkable at the level of individual nucleotides, use separate objectives for structural and binding design, and accept candidates through uncalibrated thresholds that provide no control over how often an infeasible design is admitted.

We introduce RIVET, a closed-loop agent that repairs candidate sequences rather than emitting them, formulating design as verifier-guided, model-based reinforcement learning over interpretable edits. SKETCH is a reasoning policy that proposes typed edits from a grounded biophysical operator vocabulary together with explicit rationales, so every step is auditable and every stated effect is checkable. ECHO is a learned world model that predicts the quality change of an edit without invoking a folding tool. SCOUT ranks edits by world-model utility, inverse-folding prior gain, and verifier energy, and forwards only the top candidates to the real folder, keeping expensive oracle calls sparse. an energy-based verifier that scores physical consistency, reasoning validity, and optional binding, and converts these signals into both a dense process reward and a conformal accept-reject gate. Potential-based shaping guarantees that the verifier reward cannot change the optimal design objective, while the conformal gate bounds false acceptance at a user-specified rate, so the loop gains dense supervision and calibrated reliability without altering what it optimizes. Because the agent operates over interpretable edits and verifier rewards rather than a task-specific decoder, one loop serves tertiary design, secondary self-consistency, and binding-conditioned design for RNA-ligand and RNA-RBP targets.

Our contributions are threefold: we formulate RNA inverse folding as verifier-guided, model-based reinforcement learning over interpretable edits; provide auditable rationales, potential-invariant verifier shaping, and conformal false-acceptance control; and reduce oracle use by screening edits with a world model and top-k selector. Under a multi-tool protocol disjoint from the loop-internal folder,

RIVET reaches 56.3% native recovery, the best mean tertiary and secondary self-consistency, and strong held-out transfer.

2 Related Works

RNA inverse folding began with secondary-structure solvers and learned policies that optimize sequences under thermodynamic base-pair models (Hofacker et al., 1994; Zadeh et al., 2011; Kleinkauf et al., 2015; Runge et al., 2018). Tertiary design was introduced by fixed-backbone Rosetta protocols (Das et al., 2010) and is now dominated by geometric or generative backbone-to-sequence models (Tan et al., 2024; Joshi et al., 2025; Huang et al., 2024; Kim et al., 2026). These methods largely remain open-loop: they emit a sequence without inspecting its folded state, giving nucleotide-level rationales, or performing local repair. Recent work optimizes structural rewards directly (Hu et al., 2026), while broader advances in verifiable RL, world models, energy scores, potential shaping, and conformal prediction motivate reliable closed-loop design (Shao et al., 2024; Schrittwieser et al., 2020; LeCun et al., 2006; Ng et al., 1999; Angelopoulos & Bates, 2023). Unlike post-hoc folding or binding evaluation pipelines (Team et al., 2025; Passaro et al., 2025; Carvajal-Patiño et al., 2025; Jiang & Chen, 2022), RIVET places calibrated verification inside the edit loop. Further discussion appears in Appendix A.1.

3 Methodology: RIVET

RIVET is a closed-loop agent with four modules (Figure 1): SKETCH proposes interpretable edits, ECHO predicts edit-induced quality changes without folding, SCOUT ranks candidates for verification, and ores consistency, reasoning validity, and optional binding. Together they form a verifier-guided, model-based reinforcement learner in which ECHO supports planning, pplies calibrated rewards and accept-reject control, and SKETCH is optimized by group-relative policy optimization. Shaping by eserves the design objective (Proposition 1), and the gate controls false acceptance (Proposition 2). Pseudocode appears in Appendix A.2.

3.1 Problem Formulation

We cast 3D RNA inverse folding as a finite-horizon decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, f, R, \gamma)$ in which an agent iteratively edits a candidate sequence until its folded conformation matches a target backbone; 2D design is the special case where the context carries a base-pair target. The state is $s_t = (y_t, \hat{C}_t, x)$, comprising the current sequence, its predicted conformation, and the context x, which contains the target backbone C, an optional 2D target T, an optional binding target z, and constraints such as GC content and forbidden motifs. An action $a_t \in \mathcal{A}$ is an interpretable edit from a grounded operator set, and the transition $s_{t+1} = f(s_t, a_t)$ refolds the edited sequence and is approximated by ECHO during planning. The reward couples structure, sequence likelihood, and an optional binding term, $R(s_t) = \mathrm{scTM}(\hat{C}_t, C) + \eta\, \ell_\omega(y_t \mid C) + \zeta\, B(y_t, z)$, where ℓ_ω is the structure-conditioned log-likelihood from the initializer and B is a binding score from etting $\zeta{=}0$ recovers pure structural design, and the 2D task replaces scTM with base-pair agreement to T.

We optimize the reasoning policy by KL-regularized reinforcement learning (Schulman et al., 2017), maximizing $\mathbb{E}_{\pi_\psi}[\sum_t \gamma^t \tilde{R}(s_t, a_t)] - \beta_{\mathrm{KL}} \mathbb{E}_{s_t}[\mathrm{KL}(\pi_\psi \| \pi_{\mathrm{ref}})]$ subject to $E_\phi(s_t) < \tau$, where π_{ref} is a frozen supervised-initialized reference that keeps reasoning traces fluent and on-manifold. Because the task reward moves only after a successful refold, we shape it with a potential $\Phi(s) = -E_\phi(x, y, r)$,

$$\tilde{R}(s_t, a_t) = \underbrace{\big[R(s_{t+1}) - R(s_t)\big]}_{\text{task progress}} + \underbrace{\big[\gamma\Phi(s_{t+1}) - \Phi(s_t)\big]}_{\text{verifier shaping}},$$

so verifier energy supplies an every-step learning signal.

Proposition 1 (shaping is policy-invariant). *For any bounded potential Φ, the optimal policy under $\tilde{R}$ equals the optimal policy under R* (Ng et al., 1999). Hence celerates credit assignment but provably cannot steer RIVET toward designs that the true objective R would reject. A restatement adapted to our constrained setting appears in Appendix A.11.

3.2 INITIALIZATION AND STRUCTURE-CONDITIONED PRIORS

RIVET does not begin editing from a sequence. A backbone-conditioned non-autoregressive inverse-folding model q_ω maps a target backbone to a sequence distribution and remains frozen during reinforcement learning. Each nucleotide is represented by three sequence-independent backbone atoms, so the encoder receives no nucleotide identity, and a geometric Transformer yields per-residue scalar and vector states. Because all positions are decoded in parallel, a candidate population $\mathcal{Y}_0$ is obtained from one encoding pass. en selects the starting sequence subject to feasibility, $y_0 = \arg\max_{y \in \mathcal{Y}_0} S_{\text{init}}(y)$ such that $E_\phi(x, y, r_{\text{init}}) < \tau$, where S_{init} combines the verifier energy, the sequence log-likelihood, and a constraint-satisfaction term, and the selected sequence is folded to form s_0. The architecture, features, attention formulation, training objective, and the definition of S_{init} are given in Appendix A.3.

Beyond initialization, q_ω supplies two priors reused each step. The log-likelihood $\ell_\omega(y \mid C) = |V_C|^{-1} \sum_{i \in V_C} \log q_\omega(y_i \mid C)$ enters the reward as the sequence-prior term, where V_C contains residues with valid geometric inputs. The log-prior gain $\Delta_\omega(i, b \mid y_t, C) = \log q_\omega(y_i{=}b \mid C) - \log q_\omega(y_i{=}y_{t,i} \mid C)$ is positive when a substitution is more compatible with the target backbone and is used by SCOUT to prioritize candidates. A position-level editability prior from the predictive entropy of q_ω, which concentrates proposals on ambiguous positions, is defined in Appendix A.3.

3.3 SKETCH: REASONING-GUIDED PROPOSER

SKETCH is an instruction-tuned language model (Qwen3.5-9B-Instruct) acting as the policy π_ψ. At each step it receives the target identifier, the current sequence, the step index, the remaining edit budget, and the legal alphabet $\{A, C, G, U\}$, and must return a structured edit in a fixed JSON schema. It first emits a reasoning trace and then a structured action, $(r_t, a_t) \sim \pi_\psi(\cdot \mid s_t)$, constrained to a grounded operator vocabulary that includes enforcing a canonical pair implied by backbone geometry, strengthening base stacking, satisfying an A-minor or base-triple contact, resolving a steric clash, stabilizing a stem, tuning a loop, adjusting GC content, installing an aptamer motif, and shaping a binding pocket. Every operator maps to a deterministic edit, carries an explicit biophysical rationale, and parses into a typed program that also feeds o grounding keeps the reasoning faithful and blocks edits whose stated effect cannot be checked. The complete vocabulary, argument signatures, preconditions, and prompt template are given in Appendix A.7.

We train π_ψ with Group-Relative Policy Optimization (Shao et al., 2024), which fits our setting because SKETCH already proposes a group of candidate edits per state, so no separate critic is required. At state s we draw G traces $\{(r_i, a_i)\}_{i=1}^G \sim \pi_{\text{old}}$, score each by its shaped return R_i, standardize the advantage within the group as $\hat{A}_i = (R_i - \text{mean}_j R_j)/\text{std}_j R_j$, and minimize the clipped surrogate with importance ratio ρ_i,

$$\mathcal{L}_\pi = -\mathbb{E}\Big[\tfrac{1}{G}\sum_{i=1}^{G} \min\big(\rho_i \hat{A}_i,\ \text{clip}(\rho_i, 1{-}\epsilon, 1{+}\epsilon)\,\hat{A}_i\big)\Big] + \beta_{\text{KL}}\,\text{KL}(\pi_\psi \| \pi_{\text{ref}}).$$

Because $\hat{A}_i$ scores a trace against its siblings on the same target, the policy learns which biophysical rationale actually improves structural quality rather than memorizing target-specific sequences.

3.4 ECHO: RNA WORLD MODEL

Refolding every candidate is expensive, so ECHO learns a surrogate for the quality change of an edit. We summarize the quality of a folded sequence by a single scalar built from three standardized structural metrics, $Q(y) = 0.2\,\tilde{s}_{\text{rmsd}}(y) + 0.4\,\tilde{s}_{\text{lddt}}(y) + 0.4\,\tilde{s}_{\text{tm}}(y)$, oriented so that higher is always better. For a candidate edit, ECHO receives the parent sequence, the child sequence, the action, and the parent metrics, and predicts $\hat{\Delta}_t = f_\theta(s_t, a_t) \approx Q(y_{t+1}) - Q(y_t)$, with an auxiliary head predicting the child metric triple $\hat{s}_c$. It is trained by regression against folding-derived targets,

$$\mathcal{L}_{\text{ECHO}} = \mathbb{E}\big[(f_\theta(s_t, a_t) - \Delta Q_t)^2\big] + \lambda_{\text{aux}} \sum_c \mathbb{E}\big[(\hat{s}_c - s_c)^2\big],$$

where the delta-state term defines the ranking signal and the auxiliary term regularizes the shared representation. Supplying the parent metrics as inputs rather than predicting them lets the surrogate

concentrate on the effect of the edit rather than on absolute target difficulty. The graph construction and edit-type encoding are given in Appendix A.5.

3.5 SCOUT: TOP-k CANDIDATE SELECTOR

Because SKETCH proposes more candidates than the folder can evaluate, SCOUT selects which edits to verify. It combines world-model utility, prior gain, and verifier energy as $U(s,a) = \hat{\Delta}(s,a) + \mu\,\Delta_\omega(a) - \beta\,E_\phi(x,y',r)$, where y' is the edited sequence, and keeps the top-k candidates for folding. This limits oracle calls while using all three signals. Slate construction, legality filtering, and deduplication are given in Appendix A.8.

3.6 NERGY-BASED VERIFIER

We define an energy $E_\phi(x,y,r)$ in which a low value denotes a design that is consistent with x and supported by r. The sequence, the reasoning trace, and the context are encoded and fused by cross-attention into one scalar head. Training uses three terms whose total is $\mathcal{L}_E = \mathcal{L}_{\text{con}} + \lambda_1\mathcal{L}_{\text{pref}} + \lambda_2\mathcal{L}_{\text{cal}}$. A contrastive term pushes valid designs below structure-breaking perturbations, $\mathcal{L}_{\text{con}} = \mathbb{E}[(m + E_\phi(x,y^+,r) - E_\phi(x,y^-,r))_+]$. A preference term ranks a trace below a hallucinated one, $\mathcal{L}_{\text{pref}} = -\mathbb{E}[\log\sigma(E_\phi(x,y,r^-) - E_\phi(x,y,r^+))]$. A calibration term regresses a monotone map g of the energy onto a quality signal q, $\mathcal{L}_{\text{cal}} = \mathbb{E}[(g(E_\phi(x,y,r)) - q)^2]$, where q is averaged over the folding ensemble so that the energy does not collapse onto a single oracle. Given a binding target z, poses a binding head $B(y,z)$ trained on the data described under Setup. Architecture, negative construction, and calibration details are given in Appendix A.6.

The calibrated energy is also the process reward that drives SKETCH: the step signal $-E_\phi(x,y_t,r_t)$ scores every intermediate reasoning step and is exactly the potential Φ of Proposition 1, so credit reaches feasibility-improving edits long before a refold confirms success. Finally, we turn the threshold into a guarantee. Given a calibration set of n known-infeasible designs with energies $\{e_j\}_{j=1}^n$, we set τ to their empirical α-quantile and accept if and only if $E_\phi \le \tau$.

Proposition 2 (false-acceptance control). *Under exchangeability of calibration and test infeasible designs, the probability that the filter accepts a new infeasible design is at most* $\alpha + \frac{1}{n+1}$ (Angelopoulos & Bates, 2023). The gate therefore carries a distribution-free reliability guarantee rather than a hand-tuned cutoff. At inference ts as a reranker, as the dense reward $-E_\phi$ plus ζB, and as this calibrated filter. The construction of the calibration set and the exchangeability assumption it requires are discussed in Appendix A.11.

3.7 CLOSED-LOOP TRAINING

The learnable modules are trained jointly as $\mathcal{L}_{RIVET} = \mathcal{L}_\pi + \nu_1\mathcal{L}_{\text{ECHO}} + \nu_2\mathcal{L}_E$ in an alternating loop. SKETCH proposes a group of edits, ECHO scores them, and SCOUT selects the top-k for verification. The true folder is queried only on those candidates, and each verified transition enters a replay buffer that refreshes the regression targets of ECHO, the calibration set of nd the shaped returns driving the policy update, after which the best verified child advances the state. Real-tool labels thus enter training only through a thin, top-k channel, which is what makes verifier-guided reinforcement learning affordable for an expensive oracle. Optimizer settings, loss weights, and search budgets are given in Appendix A.12.

4 EXPERIMENTS

We study five questions. (Q1) Does RIVET improve native recovery and multi-tool self-consistency over state-of-the-art tertiary designers? (Q2) Do the gains hold across sequence lengths and on out-of-distribution families? (Q3) Do they extend to secondary-structure agreement? (Q4) Can the same loop produce sequences with desired binding properties? (Q5) Which modules drive the gains, and at what folding cost? All evaluation is *in silico* and no wet-laboratory validation is claimed.

Setup. We construct the main tertiary benchmark from RNAsolo[2] (Adamczyk et al., 2022; 2025), merging the BGSU and Rfam repositories, deduplicating by PDB, model, and chain identifier, and retaining single-chain RNAs of 10 to 500 nucleotides at resolution 3.0 Å or better. To suppress

Table 1: **Main tertiary inverse folding** on the structure-similarity split. All methods here condition on a three-dimensional backbone, so every cell is defined and directly comparable. Best and second-best recovery, scTM, and F1 are in **bold** and underlined. The dispersion σ is descriptive and is not ranked. Thermodynamic and learned *secondary*-structure solvers are evaluated separately in Table 2.

Method	Rec.%↑	Tertiary self-consistency scTM ↑				σ	Secondary self-consistency F1 ↑			
		Protenix	trRosettaRNA	Boltz-2	Mean		Protenix	trRosettaRNA	Boltz-2	Mean
RIDER	0.418	0.150	0.122	0.181	0.151	0.030	0.068	0.092	0.091	0.084
gRNAde	0.408	0.253	0.235	0.264	0.251	0.015	0.046	0.047	0.057	0.050
RDesign	<u>0.523</u>	0.283	0.286	0.369	0.312	0.049	0.248	0.305	0.359	0.304
RiboDiffusion	0.454	<u>0.310</u>	<u>0.368</u>	<u>0.370</u>	<u>0.355</u>	0.023	<u>0.455</u>	<u>0.542</u>	<u>0.500</u>	<u>0.499</u>
RIVET (ours)	**0.563**	**0.366**	**0.416**	**0.473**	**0.418**	0.053	**0.538**	**0.635**	**0.637**	**0.603**

Note: Means are computed per target over the predictors that returned a valid prediction and then averaged over targets; σ is the per-target cross-predictor standard deviation of scTM, likewise averaged. Structure-conditioned perplexity is model-specific and not comparable across designers: it is 3.621 (gRNAde) and 3.104 (RDesign), and we do not rank it.

redundancy we cluster chains at 30% sequence identity and additionally at TM-score 0.5, take the union of the two relations to form connected components, and split components into training, validation, and test sets in an approximate 8:1:1 ratio, so that chains related under either criterion never cross a split boundary. Generalization uses held-out RNA-Puzzles (Cruz et al., 2012) and cross-family Rfam targets (Kalvari et al., 2021). Binding benchmarks are curated from HARIBOSS (Panei et al., 2022), ROBIN (Yazdani et al., 2023), and Rfam aptamer domains for RNA-ligand, and from RNAcompete (Ray et al., 2013) and ENCODE eCLIP (Van Nostrand et al., 2020) for RBP-RNA. Full preprocessing is given in Appendix A.9. Tertiary baselines are RDesign (Tan et al., 2024), gRNAde (Joshi et al., 2025), RiboDiffusion (Huang et al., 2024), and RIDER (Hu et al., 2026); we additionally evaluate secondary-structure solvers on the same tertiary split to quantify how far thermodynamic design alone transfers.

Metrics and protocol. At the sequence level we report native recovery, and we record structure-conditioned perplexity in the note to Table 1 without ranking it, since each value is computed under a different generative model. At the tertiary level we report scTM under three predictors, Protenix (Team et al., 2025), trRosettaRNA (Wang et al., 2023), and Boltz-2 (Passaro et al., 2025), and at the secondary level base-pair F1 under the same three. For binding we use predictors disjoint from the loop: RNAmigos2 (Carvajal-Patiño et al., 2025), RLDOCK (Jiang & Chen, 2022), and AnnapuRNA (Stefaniak & Bujnicki, 2021) for ligands, with docking energies min-max normalized so higher is better, and DeepBind (Alipanahi et al., 2015) and ZHMolGraph for RBPs. We also report the cross-predictor standard deviation σ of scTM, treated as a dispersion statistic that indicates how the panel agrees, with no preferred direction, because its magnitude scales with the mean. The folding tool used inside the loop is kept disjoint from the reporting panel, so reported self-consistency is not the quantity the agent optimized, and a method can lead only if its advantage survives three correlated views of the same design.

Ablation protocol. Ablations reuse one checkpoint and perform inference only on 60 test targets, with expansion and ranking driven by learned scores and real verification performed only on the exported top-1 leaf. The protocol and search budget are given in Appendix A.10 and Table 7.

5 RESULTS

Tertiary inverse folding (Q1). Table 1 reports the main comparison among backbone-conditioned designers. RIVET attains the highest native recovery at 0.563, ahead of the strongest baseline RDesign at 0.523, and leads under all three tertiary predictors, raising mean scTM from 0.355 to 0.418, with per-column margins of +0.056, +0.048, and +0.103 under Protenix, trRosettaRNA, and Boltz-2 relative to the best baseline in that column. The ordering is preserved under each predictor separately, which is what a genuine improvement rather than single-oracle tuning produces. Secondary agreement improves in parallel, with mean base-pair F1 rising from 0.499 to 0.603 and RIVET leading under each predictor individually, so the tertiary gains are not purchased by degrading base-pairing fidelity. The dispersion column tracks the scale of the corresponding means, so we

Table 2: **Secondary-structure solvers evaluated on the tertiary split**, which calibrates how far base-pair-level design transfers to three-dimensional targets. Rows are ordered by recovery. RIVET is repeated from Table 1 as a reference line and is not part of this comparison class. Base-pair F1 is deliberately omitted: these solvers receive the base-pair target as their design objective, so base-pair self-consistency is not an independent measurement for them and is reported only for the backbone-conditioned designers in Table 1.

Method	**Rec.**%↑	**Tertiary self-consistency** scTM ↑				σ
		Protenix	trRosettaRNA	Boltz-2	Mean	
antaRNA	0.262	0.220	0.199	0.243	0.221	0.022
LEARNA	0.262	0.255	0.250	0.285	0.263	0.019
MCTS-RNA	0.293	0.245	0.239	0.280	0.255	0.022
RNAinverse	0.296	0.258	0.235	0.285	0.259	0.025
eM2dRNAs	0.308	0.262	0.191	0.273	0.264	0.025
Meta-LEARNA	0.312	0.248	0.226	0.271	0.248	0.023
RIVET (reference)	*0.563*	*0.366*	*0.416*	*0.473*	*0.418*	*0.053*

Table 3: **Full performance breakdown across RNA characteristics.** Recovery and mean scTM over the three tertiary predictors. Superscripts name the best baseline for each cell, and Δ is the change of RIVET relative to that baseline. Subset sizes are given so that small-sample entries can be weighted accordingly. Per-predictor versions of this breakdown are given in Appendix A.13.

Category	**Recovery (%) ↑**			**Mean scTM ↑**			**Count**
	Best baseline	RIVET	Δ	Best baseline	RIVET	Δ	
By length							
Short ($\leq$50 nt)	47.8^{RDesign}	**54.9**	+7.1	$0.270^{\dagger}$	**0.270**	$\sim$0	1,247
Medium (50 to 100 nt)	54.8^{RDesign}	**55.3**	+0.5	0.419^{RDesign}	**0.527**	+0.108	2,163
Long (>100 nt)	52.4^{RDesign}	**58.9**	+6.5	0.355^{RiboDiff}	**0.630**	+0.275	891
Highly structured RNAs							
tRNA (RF00005)	63.5^{RDesign}	61.8	−1.7	0.526^{RiboDiff}	**0.576**	+0.050	428
5S rRNA (RF00001)	61.7^{RiboDiff}	**65.1**	+3.4	0.305^{RDesign}	**0.723**	+0.418	156
LSU rRNA, bacteria (RF02541)	58.1^{RiboDiff}	**70.1**	+12.0	0.272^{RiboDiff}	**0.447**	+0.175	89
Riboswitch aptamer domains							
Lysine (RF00168)	51.9^{RDesign}	44.8	−7.1	0.225^{RDesign}	**0.278**	+0.053	47
NsiR1 (RF02399)	$44.2^{\text{RNAinverse}}$	**50.6**	+6.4	$0.165^{\text{RNAinverse}}$	**0.185**	+0.020	32
THF	31.9^{RiboDiff}	**44.9**	+13.0	0.284^{RiboDiff}	**0.412**	+0.128	28
TPP	42.2^{LEARNA}	**46.5**	+4.3	0.199^{LEARNA}	**0.224**	+0.025	24
c-di-GMP-I	$46.9^{\text{Meta-LEARNA}}$	**62.3**	+15.4	$0.251^{\text{Meta-LEARNA}}$	**0.290**	+0.039	19
Held-out generalization							
RNA-Puzzles	47.1^{RDesign}	**63.1**	+16.0	0.294^{RiboDiff}	**0.428**	+0.134	24
CASP15	35.1^{RiboDiff}	**39.0**	+3.9	0.241^{RiboDiff}	0.241	$\sim$0	8
Overall test set	52.3^{RDesign}	**56.3**	+4.0	0.355^{RiboDiff}	**0.418**	+0.063	4,301

Note: RiboDiff abbreviates RiboDiffusion. † The strongest baseline matches RIVET to three decimals, so no advantage is claimed on short targets. RNA-Puzzles and CASP15 are held out from training; subsets should be weighted by count.

draw no ranking from it. Table 2 calibrates the benchmark from the other side: every secondary-structure solver evaluated on this split remains below 0.32 recovery and 0.27 mean scTM, showing that matching a base-pair projection is insufficient for these targets.

Length, family, and generalization (Q2, Q3). Table 3 gives the full breakdown by length, family, and held-out benchmark; the corresponding per-predictor tables are in Appendix A.13. The advantage increases with target length: mean scTM improves by +0.108 on medium targets and by +0.275 beyond 100 nucleotides, while short targets are essentially unchanged. This matches the

Table 4: **Ablation and reliability** on the 60-target inference-only subset. The first row reports absolute values; other rows report changes relative to it. The dispersion σ is the cross-predictor standard deviation of scTM.

Configuration	Sequence	Tertiary scTM ↑				σ
	Rec.↑	Protenix	trRosettaRNA	Boltz-2	Mean	
Reference, no edit (absolute)	0.549	0.366	0.416	0.473	0.418	0.0530
Effect of edit depth, full RIVET						
depth 1	∼0.0000	+0.0081	+0.0010	+0.0018	+0.0038	0.0042
depth 2	−0.0038	+0.0086	+0.0025	+0.0049	+0.0051	0.0040
depth 3 (**full**)	−0.0042	**+0.0093**	**+0.0030**	**+0.0051**	**+0.0058**	0.0032
Module ablations						
Single-step (depth 1, one parent)	−0.0051	+0.0018	+0.0048	−0.0001	+0.0021	0.0025
ECHO-only guidance	−0.0042	+0.0033	+0.0011	+0.0016	+0.0020	0.0012
Policy-prior-only ranking	+0.0028	−0.0020	−0.0012	+0.0037	+0.0002	0.0031
Uniform-random slate	−0.0023	+0.0021	+0.0027	+0.0016	+0.0021	0.0006

Table 5: **Binding-conditioned design.** Left: RNA-ligand binding with normalized docking scores and success rate. Right: RBP-RNA binding probability. Unconditioned methods ignore the binding target; RIVET$_{\zeta=0}$ also omits the binding head. Best values are in **bold**, second best are underlined; σ is descriptive.

Method	RNA-ligand binding				σ	RBP-RNA binding			σ
	RNAmigos2↑	RLDOCK↑	AnnapuRNA↑	Succ.%↑		DeepBind↑	ZHMolGraph↑	Mean↑	
RIDER	0.697	0.367	0.393	40.0	0.234	0.5551	0.510	0.533	0.031
gRNAde (unconditioned)	0.701	0.504	**0.548**	62.5	0.203	0.5417	0.654	0.598	0.079
RDesign (unconditioned)	0.659	0.550	0.492	61.9	0.222	0.5468	0.674	0.610	0.089
RiboDiffusion (unconditioned)	0.701	0.506	0.531	60.0	0.186	0.5564	0.604	0.580	0.033
RIVET$_{\zeta=0}$ (structure only)	**0.710**	**0.561**	0.531	**64.5**	0.259	**0.5619**	**0.694**	**0.628**	0.093

motivation for iterative repair, since a longer backbone admits more independent local defects that one forward pass must resolve simultaneously but an edit loop can address sequentially, and the short-target result also shows the loop does not degrade cases where a one-pass designer is already adequate. On highly structured families the improvements are largest where baselines are weakest, with mean scTM rising by $+0.418$ on 5S rRNA and $+0.175$ on bacterial LSU rRNA, and recovery by $+12.0$ points on the latter; riboswitch aptamer domains follow the same direction, including $+13.0$ recovery on THF and $+15.4$ on c-di-GMP-I. Because several subsets are small, we treat individual family values as indicative and rely on the consistency of the direction across families.

Two families show negative recovery deltas, tRNA at -1.7 and lysine riboswitch at -7.1, while their mean scTM deltas remain positive at $+0.050$ and $+0.053$. We read this as a property of the objective rather than a defect: the loop accepts an edit only when verified structural utility improves, so on families where the native sequence is not the unique structurally adequate solution it moves away from native identity. On held-out benchmarks, RNA-Puzzles recovery improves by $+16.0$ points to 0.631 and mean scTM by $+0.134$ to 0.428; per-method values for both held-out sets are given in Table 8. CASP15 is the most demanding setting: recovery improves by $+3.9$ points to 0.390 while mean scTM matches RiboDiffusion at 0.241, so on those eight targets we report a sequence-level improvement together with parity in structural self-consistency and claim no structural advantage. Regarding Q3, the secondary columns of Table 1 show the highest base-pair F1 under all three predictors without a separate secondary objective, so tertiary repair carries over to secondary agreement; we do not report Eterna100 or bpRNA-1m TS0 and therefore make no claim about RIVET as a dedicated secondary-structure solver.

Binding-conditioned design (Q4). Table 5 reports a conservative binding evaluation: the baselines are unconditioned and ignore z, and the reported variant is RIVET$_{\zeta=0}$, which optimizes structure only and does not use the binding head, so any advantage is attributable to structural accuracy rather than binding supervision. On RNA-ligand targets RIVET$_{\zeta=0}$ attains the best consensus under RNAmigos2 at 0.710 and RLDOCK at 0.561, is second under AnnapuRNA, and attains the highest

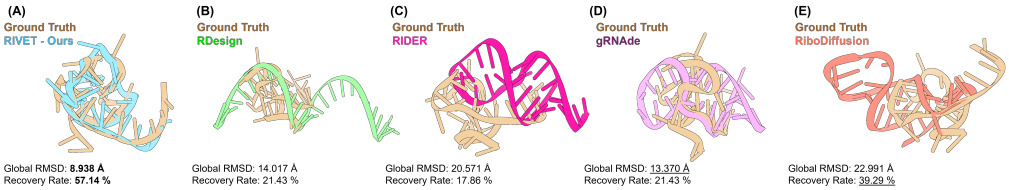


Figure 2: **Case study of closed-loop repair on PDB 6V9B.** Each panel traces one RIVET trajectory from the initializer output to the accepted design, showing the applied grounded operator with its rationale, the predicted conformation superposed on the target backbone, and the per-step verifier energy and multi-tool scTM. Edits are retained only when the verified utility improves, so rejected steps roll back to the previous sequence.

success rate at 64.5% against 62.5% for the strongest unconditioned baseline; on RBP targets it is best under both predictors, giving the highest mean at 0.628. The consistent reading is that geometrically accurate designs tend to be more binding-competent, which is what one expects when the interaction surface is defined by tertiary geometry. We are explicit about the limits: margins are modest, one baseline leads under AnnapuRNA, all scores come from predictors rather than measured affinity, and the binding-aware variant with $\zeta > 0$ is not included, so the claim is only that structural conditioning alone already matches or exceeds unconditioned designers on these panels.

Ablation and reliability (Q5). Table 4 isolates each module. Absolute values are subset-specific, and all other rows are within-table contrasts. Editing improves structural quality monotonically with depth, with mean scTM increasing by $+0.0038$, $+0.0051$, and $+0.0058$ at depths 1, 2, and 3, positive under every predictor at every depth; the magnitudes are small on this subset, so we base the conclusion on sign consistency across three predictors and three depths rather than any individual entry. Recovery moves oppositely over the same depths, the same dissociation observed at the family level and expected when acceptance is governed by verified utility. Dispersion decreases with depth, from 0.0530 to 0.0032, which we interpret as increasing agreement among the independent predictors about the accepted designs, and which is relevant because it addresses the concern that a verifier-driven loop might improve one folding model at the expense of others. Figure 2 shows a representative accepted trajectory, in which each retained edit is one grounded operator whose stated rationale can be compared against the verified change it produced.

The module ablations separate two functions. Removing multi-step search reduces the mean gain from $+0.0058$ to $+0.0021$, indicating that a substantial part of the benefit comes from recursive repair rather than a single well-chosen edit; this row is not budget-matched to the depth-1 configuration, since single-step verifies only the surrogate-selected top-1 candidate whereas depth 1 verifies a larger set and selects by verified utility, so the gap should not be attributed to depth alone. Removing the learned ranking signals is more damaging: policy-prior-only ranking reduces the mean gain to $+0.0002$ and turns two of three predictors negative, showing that the policy alone cannot reliably identify which of its own proposals deserve verification. ECHO-only guidance retains most of the gain at $+0.0020$, locating the dominant ranking signal in the world model while leaving a smaller contribution to d the value head. The uniform-random slate reaches $+0.0021$, which we read as evidence that part of the benefit already resides in the grounded operator vocabulary, since even random selection from a legal, biophysically constrained slate is not harmful. On cost, all intermediate expansion and ranking is handled by the learned modules, so the oracle budget is independent of tree size, which is what makes verifier-guided search practical when the folder is expensive.

6 Conclusion

RIVET reframes RNA inverse folding as closed-loop repair: a reasoning policy proposes typed grounded edits, a world model decides which deserve folding, and a calibrated energy verifier supplies a potential-invariant dense reward and a conformal accept-reject gate. It attains 56.3% native recovery and the best mean tertiary and secondary self-consistency, with the ordering preserved under every predictor rather than a single oracle and gains concentrated on longer, densely structured targets. More broadly, verifier-in-the-loop repair offers a template for design problems whose quality is revealed only by an expensive oracle. The outlook is set out in Appendix A.14.

AI USE STATEMENT

Generative AI tools were used solely for manuscript preparation, specifically language polishing and organization of already-written text.

ETHICS STATEMENT

The authors have read and adhere to the ICLR Code of Ethics. This work is computational and involves no human subjects and no personally identifiable information. All structure and interaction data are drawn from existing public resources, listed in Appendix A.9. The outputs of RIVET are candidate sequences rather than validated biological products, and downstream use should follow standard biosafety and regulatory procedures. Limitations are discussed in Appendix A.15.

REPRODUCIBILITY STATEMENT

The anonymous repository linked in the abstract provides the code, the configuration files used in the reported runs, the grounded operator specification, and the dataset preprocessing scripts. Dataset construction and splits are described in Appendix A.9; the initializer in Appendix A.3; the transition bank and stage ordering in Appendix A.4; ECHO and Appendices A.5 and A.6; search and selection in Appendix A.8, with the search configuration in Table 7; the ablation protocol in Appendix A.10; and the implementation and hardware configuration in Appendix A.12. Assumptions and proofs for Propositions 1 and 2 are given in Appendix A.11, and Algorithms 1 and 2 state the inference and training loops.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. Rnasolo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 38(14):3668–3670, 2022.

Bartosz Adamczyk, Pawel Boinski, Marta Szachniuk, and Maciej Antczak. Rnasolo 2.0: multi-modal database to study rnas, their structural families and intermolecular interfaces. *Journal of Molecular Biology*, pp. 169570, 2025.

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33 (8):831–838, 2015.

Rishabh Anand, Chaitanya K Joshi, Alex Morehead, Arian R Jamasb, Charles Harris, Simon V Mathis, Kieran Didi, Rex Ying, Bryan Hooi, and Pietro Liò. Rna-frameflow: Flow matching for de novo 3d rna backbone design. *ArXiv*, pp. arXiv–2406, 2025.

Jeff Anderson-Lee, Eli Fisker, Vineet Kosaraju, Michelle Wu, Justin Kong, Jeehyung Lee, Minjae Lee, Mathew Zada, Adrien Treuille, Rhiju Das, et al. Principles for predicting rna secondary structure design difficulty. *Journal of molecular biology*, 428(5):748–757, 2016.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Parth Bibekar, Lucien F Krapp, and Matteo Dal Peraro. Context-aware geometric deep learning for rna sequence design. *bioRxiv*, pp. 2025–06, 2025.

Juan G Carvajal-Patiño, Vincent Mallet, David Becerra, Luis Fernando Niño Vasquez, Carlos Oliver, and Jérôme Waldispühl. Rnamigos2: accelerated structure-based rna virtual screening with deep graph learning. *Nature Communications*, 16(1):2799, 2025.

Thomas R Cech and Joan A Steitz. The noncoding rna revolution—trashing old rules to forge new ones. *Cell*, 157(1):77–94, 2014.

Angelica Chen, Samuel Don Stanton, Robert G Alberstein, Andrew Martin Watkins, Richard Bonneau, Vladimir Gligorijevic, Kyunghyun Cho, and Nathan C Frey. Llms are highly-constrained biophysical sequence optimizers. In *NeurIPS 2024 Workshop on AI for New Drug Modalities*, 2024.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *BioRxiv*, pp. 2022–08, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

José Almeida Cruz, Marc-Frédérick Blanchet, Michal Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Song Cao, Rhiju Das, Feng Ding, Nikolay V Dokholyan, Samuel Coulbourn Flores, et al. Rna-puzzles: a casp-like evaluation of rna three-dimensional structure prediction. *Rna*, 18(4):610–625, 2012.

Tulsi Ram Damase, Roman Sukhovershin, Christian Boada, Francesca Taraballi, Roderic I Pettigrew, and John P Cooke. The limitless future of rna therapeutics. *Frontiers in bioengineering and biotechnology*, 9:628137, 2021.

Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bprna: large-scale automated annotation and analysis of rna secondary structure. *Nucleic acids research*, 46(11):5381–5394, 2018.

Rhiju Das, John Karanicolas, and David Baker. Atomic accuracy in predicting and designing non-canonical rna structure. *Nature methods*, 7(4):291–294, 2010.

Peter Eastman, Jade Shi, Bharath Ramsundar, and Vijay S Pande. Solving the rna design problem with reinforcement learning. *PLoS computational biology*, 14(6):e1006176, 2018.

Alireza Ghafarollahi and Markus J Buehler. Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 3(7): 1389–1409, 2024.

Girolamo Giudice, Fátima Sánchez-Cabo, Carlos Torroja, and Enrique Lara-Pezzi. Attract—a database of rna-binding proteins and associated motifs. *Database*, 2016:baw035, 2016.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of rna secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188, 1994.

Tianmeng Hu, Yongzheng Cui, Biao Luo, and Ke Li. Rider: 3d rna inverse design with reinforcement learning-guided diffusion. *arXiv preprint arXiv:2602.16548*, 2026.

Han Huang, Ziqian Lin, Dongchen He, Liang Hong, and Yu Li. Ribodiffusion: tertiary structure-based rna inverse folding with generative diffusion models. *Bioinformatics*, 40(Supplement_1): i347–i356, 2024.

Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. Linearfold: linear-time approximate rna folding by 5'-to-3'dynamic programming and beam search. *Bioinformatics*, 35(14):i295–i304, 2019.

Yangwei Jiang and Shi-Jie Chen. Rldock method for predicting rna-small molecule binding modes. *Methods*, 197:97–105, 2022.

Chaitanya Joshi, Arian Jamasb, Ramon Viñas, Charles Harris, Simon Mathis, Alex Morehead, Rishabh Anand, and Pietro Liò. grnade: Geometric deep learning for 3d rna inverse design. In *International Conference on Learning Representations*, volume 2025, pp. 10166–10191, 2025.

Ioanna Kalvari, Eric P Nawrocki, Nancy Ontiveros-Palacios, Joanna Argasinska, Kevin Lamkiewicz, Manja Marz, Sam Griffiths-Jones, Claire Toffano-Nioche, Daniel Gautheret, Zasha Weinberg, et al. Rfam 14: expanded coverage of metagenomic, viral and microrna families. *Nucleic acids research*, 49(D1):D192–D200, 2021.

Nayoung Kim, Giuliano De Carluccio, Kehan Zhang, and James J Collins. Generative ai for synthetic biology: Designing biological parts, circuits, and genomes. *Cell Systems*, 17(2), 2026.

Robert Kleinkauf, Martin Mann, and Rolf Backofen. antarna: ant colony-based rna sequence design. *Bioinformatics*, 31(19):3114–3121, 2015.

Rohan V Koodli, Boris Rudolfs, Jonathan Romano, Hannah K Wayment-Steele, William A Dunlap IV, Eterna Participants, and Rhiju Das. Redesigning the eterna100 for the vienna 2 folding engine. *BioRxiv*, pp. 2021–08, 2021.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Yang Li, Chengxin Zhang, Chenjie Feng, Robin Pearce, P Lydia Freddolino, and Yang Zhang. Integrating end-to-end learning with deep geometrical potentials for ab initio rna structure prediction. *Nature Communications*, 14(1):5745, 2023.

Yichong Li, Xiaoyong Pan, Hongbin Shen, and Yang Yang. Drag: design rnas as hierarchical graphs with reinforcement learning. *Briefings in Bioinformatics*, 26(2):bbaf106, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

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.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

Daniel Maticzka, Sita J Lange, Fabrizio Costa, and Rolf Backofen. Graphprot: modeling binding preferences of rna-binding proteins. *Genome biology*, 15(1):R17, 2014.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Francesco P Panei, Rachel Torchet, Herve Menager, Paraskevi Gkeka, and Massimiliano Bonomi. Hariboss: a curated database of rna-small molecules structures to aid rational drug design. *Bioinformatics*, 38(17):4185–4193, 2022.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Debashish Ray, Hilal Kazan, Kate B Cook, Matthew T Weirauch, Hamed S Najafabadi, Xiao Li, Serge Gueroussov, Mihai Albu, Hong Zheng, Ally Yang, et al. A compendium of rna-binding motifs for decoding gene regulation. *Nature*, 499(7457):172–177, 2013.

Álvaro Rubio-Largo, Laura Escobar-Encinas, Nuria Lozano-García, and José Maria Granado-Criado. Evolutionary strategy to enhance an rna design tool performance. *IEEE Access*, 12: 15582–15593, 2024.

Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design rna. *arXiv preprint arXiv:1812.11951*, 2018.

Frederic Runge, Jörg Franke, Daniel Fertmann, Rolf Backofen, and Frank Hutter. Partial rna design. *Bioinformatics*, 40(Supplement_1):i437–i445, 2024.

Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications*, 12(1):941, 2021.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, Yixuan Wang, Liang Hong, Jin Xiao, et al. Accurate rna 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, 21(12):2287–2298, 2024.

Filip Stefaniak and Janusz M Bujnicki. Annapurna: A scoring function for predicting rna-small molecule binding poses. *PLoS computational biology*, 17(2):e1008309, 2021.

Lei Sun, Kui Xu, Wenze Huang, Yucheng T Yang, Pan Li, Lei Tang, Tuanlin Xiong, and Qiangfeng Cliff Zhang. Predicting dynamic cellular protein–rna interactions by deep learning using in vivo rna structures. *Cell research*, 31(5):495–516, 2021.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Bozhen Hu, Siyuan Li, Zicheng Liu, and Stan Z Li. Rdesign: Hierarchical data-efficient representation learning for tertiary structure-based rna design. In *International Conference on Learning Representations*, volume 2024, pp. 10304–10333, 2024.

Cheng Tan, Yijie Zhang, Zhangyang Gao, Hanqun Cao, Siyuan Li, Siqi Ma, Mathieu Blanchette, and Stan Z Li. R3design: deep tertiary structure-based rna sequence design and beyond. *Briefings in Bioinformatics*, 26(1):bbae682, 2025.

ByteDance AML AI4Science Team, Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pp. 2025–01, 2025.

Eric L Van Nostrand, Peter Freese, Gabriel A Pratt, Xiaofeng Wang, Xintao Wei, Rui Xiao, Steven M Blue, Jia-Yu Chen, Neal AL Cody, Daniel Dominguez, et al. A large-scale binding and functional map of human rna-binding proteins. *Nature*, 583(7818):711–719, 2020.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Wenkai Wang, Chenjie Feng, Renmin Han, Ziyi Wang, Lisha Ye, Zongyang Du, Hong Wei, Fa Zhang, Zhenling Peng, and Jianyi Yang. trrosettarna: automated prediction of rna 3d structure with transformer network. *Nature communications*, 14(1):7266, 2023.

Yizhou Wang, Chen Tang, Han Deng, Jiabei Xiao, Jiaqi Liu, Jianyu Wu, Jun Yao, Pengze Li, Encheng Su, Lintao Wang, et al. Scireasoner: Laying the scientific reasoning ground across disciplines. *arXiv preprint arXiv:2509.21320*, 2025.

Katherine Deigan Warner, Christine E Hajdin, and Kevin M Weeks. Principles for targeting rna with drug-like small molecules. *Nature reviews Drug discovery*, 17(8):547–558, 2018.

Xiufeng Yang, Kazuki Yoshizoe, Akito Taneda, and Koji Tsuda. Rna inverse folding using monte carlo tree search. *BMC bioinformatics*, 18(1):468, 2017.

Kamyar Yazdani, Deondre Jordan, Mo Yang, Christopher R Fullenkamp, David R Calabrese, Robert Boer, Thomas Hilimire, Timothy EH Allen, Rabia T Khan, and John S Schneekloth Jr. Machine learning informs rna-binding chemical space. *Angewandte Chemie*, 135(11):e202211358, 2023.

Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks, and Niles A Pierce. Nupack: Analysis and design of nucleic acid systems. *Journal of computational chemistry*, 32(1):170–173, 2011.

Zaixi Zhang, Ruofan Jin, Linlin Chao, Guangxue Xu, Yikun Zhang, Guowei Zhou, Di Yin, Yingqing Guo, Yaqi Fu, Yukang Yang, et al. Rnagenesis: a generalist foundation model for functional rna therapeutics. *bioRxiv*, pp. 2024–12, 2024.

A Appendix

This appendix supplements the main paper and is organized as a single section with numbered subsections. Appendix A.1 extends the discussion of related work. Appendix A.2 gives pseudocode for the inference and training loops. Appendices A.3 to A.7 specify the components summarized in the main paper: the backbone-conditioned initializer, the verified transition bank and training pipeline, ECHO, nd the grounded operator vocabulary. Appendices A.8 and A.9 describe search and selection, and the datasets used in the experiments. Appendix A.10 gives the ablation protocol, Appendix A.11 the proofs and assumptions behind the two guarantees, Appendix A.12 the implementation configuration, and Appendix A.13 the per-predictor result tables omitted from the main paper for space. Appendix A.14 gives the outlook, and Appendices A.15 and A.16 state limitations, ethical considerations, and generative AI usage.

A.1 Extended Related Work

A.1.1 Secondary-structure inverse folding

The problem was originally posed as designing a sequence whose thermodynamic minimum free energy structure matches a target base-pair diagram. RNAinverse introduced adaptive local search over the ViennaRNA energy model (Hofacker et al., 1994; Lorenz et al., 2011), NUPACK generalized the formulation to multi-strand ensembles (Zadeh et al., 2011), antaRNA applied ant colony optimization with explicit GC-content control (Kleinkauf et al., 2015), MCTS-RNA introduced Monte Carlo tree search over partial assignments (Yang et al., 2017), and eM2dRNAs used multi-objective evolutionary optimization (Rubio-Largo et al., 2024). Learned policies replaced hand-designed search with reinforcement learning: LEARNA and Meta-LEARNA amortize a policy across targets (Runge et al., 2018), libLEARNA extends the action space to partial designs (Runge et al., 2024), and graph-based agents exploit the structure of the target as a relational input (Eastman et al., 2018; Li et al., 2025). These methods are typically evaluated on Eterna100 and bpRNA-derived splits (Anderson-Lee et al., 2016; Koodli et al., 2021; Danaee et al., 2018). Their common assumption is that a thermodynamic model is an adequate stand-in for folding, which holds for canonical pairing but not for the tertiary contacts that define many functional elements.

A.1.2 Tertiary inverse folding

Fixed-backbone RNA design was established with Rosetta-based protocols (Das et al., 2010), which score candidate sequences with a physics-based energy on a given backbone. Learned designers now dominate. RDesign uses a hierarchical geometric encoder with a contrastive objective (Tan et al., 2024), gRNAde applies a multi-state geometric vector-perceptron network to conformational ensembles (Joshi et al., 2025), RiboDiffusion formulates design as conditional diffusion in sequence space (Huang et al., 2024), RhoDesign couples a geometric encoder to a Transformer decoder (Kim et al., 2026), and R3Design and related context-aware variants refine the geometric representation (Tan et al., 2025; Bibekar et al., 2025). RNA foundation models supply pretrained sequence representations that can be adapted to design (Zhang et al., 2024; Chen et al., 2022). All of these emit a sequence in a single conditional pass. Consequently they cannot inspect the folded state of what

they produced, cannot localize which nucleotide choice caused a structural defect, and cannot repair such a defect without regenerating the whole sequence.

A.1.3 REWARD-DRIVEN AND AGENTIC DESIGN

Because likelihood and native recovery are incomplete proxies for foldability, recent work optimizes structural rewards directly. RIDER fine-tunes a diffusion designer against three-dimensional self-consistency (Hu et al., 2026). In adjacent domains, reinforcement learning with chemically informed rewards and tool-using language-model agents have been applied to molecular and protein design (Wang et al., 2025; Chen et al., 2024; Ghafarollahi & Buehler, 2024). RIVET differs in three ways. First, the action space consists of typed, precondition-checked edits rather than free generation, so a proposal can be rejected as illegal before any scoring. Second, the reward is a calibrated energy rather than a raw tool score, and it is injected as a potential so that it provably cannot change the optimal policy. Third, acceptance is governed by a conformal gate with an explicit false-acceptance level rather than by a tuned threshold.

A.1.4 REINFORCEMENT LEARNING FROM VERIFIABLE REWARDS AND MODEL-BASED PLANNING

Our training procedure draws on the observation that verifiable outcome signals are effective policy rewards for reasoning models (Shao et al., 2024; Guo et al., 2025; Cobbe et al., 2021; Lightman et al., 2024), and on KL-regularized policy optimization (Schulman et al., 2017). Group-Relative Policy Optimization is a natural fit here because the design loop already samples a group of sibling candidates at each state, which supplies the within-group baseline and removes the need for a separate critic. The role of ECHO follows the model-based tradition of learning a transition surrogate and planning inside it (Sutton, 1991; Ha & Schmidhuber, 2018; Schrittwieser et al., 2020; M. Moerland et al., 2023), with the specific difference that we do not model the full folded structure but only the scalar quality change induced by an edit, which is the minimal quantity needed to rank candidates.

A.1.5 VERIFICATION, ENERGY SCORES, AND CALIBRATION

an energy-based model in the sense of LeCun et al. (2006), trained contrastively so that feasible designs receive lower energy than structure-breaking perturbations. The use of the negative energy as a shaping potential relies on the policy-invariance theorem of Ng et al. (1999). The acceptance gate uses split conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023) applied one-sidedly to infeasible calibration designs. Prior RNA and protein design pipelines use independent folding and binding predictors (Anand et al., 2025; Shen et al., 2024; Wang et al., 2023; Li et al., 2023; Team et al., 2025; Passaro et al., 2025; Abramson et al., 2024; Huang et al., 2019; Sato et al., 2021; Carvajal-Patiño et al., 2025; Jiang & Chen, 2022; Stefaniak & Bujnicki, 2021; Alipanahi et al., 2015; Maticzka et al., 2014; Sun et al., 2021) almost exclusively as post-hoc evaluators. Our contribution at this level is to move calibrated verification inside the loop, where the same scalar simultaneously ranks candidates, shapes the reward, and controls acceptance.

A.2 ALGORITHMS

Algorithm 1 states the inference loop and Algorithm 2 the training loop referenced in the main paper. Both are written in prose rather than in symbols, and control flow is stated in words rather than in nested blocks, so that each line names the module responsible for it and the quantity it consumes; the corresponding formulas are those of Section 3 and are not repeated here. Indentation marks the body of a loop.

Two design choices are worth stating explicitly. First, real-tool labels enter training only through the thin top-k channel, so the number of folding calls per round is bounded by the verification budget regardless of how many candidates the policy proposes; this is what makes verifier-guided reinforcement learning affordable when the folder is expensive. Second, ECHO is never trained on its own predictions, only on folded outcomes, which prevents the surrogate from drifting toward self-confirming estimates over rounds.

Algorithm 1 RIVET closed-loop design at inference time

Input: design context, namely the target backbone, the optional secondary and binding targets, and the sequence constraints
Input: step budget, verification budget, and the calibrated acceptance threshold of Output: an accepted sequence that passes the conformal gate, or a reported failure
Draw a candidate population from the frozen backbone-conditioned initializer in one encoding pass
Select the highest-scoring feasible candidate as the starting sequence, then fold it to form the initial state
Repeat the following block until the step budget is exhausted:
    SKETCH samples a group of proposals for the current state, each pairing a reasoning trace with one grounded operator
    Discard proposals that fail to parse, name an unknown operator, violate a precondition, or duplicate an existing child
    ECHO predicts the quality change of every surviving proposal without calling the folder
    SCOUT keeps the highest-utility proposals, combining world-model utility, initializer prior gain, and verifier energy
    Fold only the retained proposals with the loop-internal folder
    signs an energy to each folded candidate, and each candidate receives a verified utility
    Take the candidate with the best verified utility as the proposal for this step
    If the verified utility of the proposal exceeds that of the current state, accept the proposal and advance the state; **otherwise** roll back and retain the current state
    If the accepted state passes the conformal gate and satisfies the design goal, return that sequence and stop
End of repeat block
Return the best gate-passing state visited, or report failure if no visited state passed the gate

Algorithm 2 RIVET verifier-guided model-based reinforcement learning at training time

Input: pool of design contexts, group size, verification budget, and step budget
Input: frozen reference policy obtained by offline policy initialization, and a replay buffer of verified transitions
Output: trained proposer SKETCH, world model ECHO, and verifier Repeat the following block for every training round:
    Draw a design context from the pool and form the initial state as in Algorithm 1
    Repeat the following inner block until the step budget is exhausted:
        SKETCH proposes a group of reasoning traces paired with grounded operators for the current state
        ECHO scores every proposal, and SCOUT keeps the highest-utility ones within the verification budget
        Fold the retained proposals and score them with Form the shaped return of each proposal as task progress plus the verifier shaping term of Section 3.1
        Append every verified transition to the replay buffer
        Advance the state to the best verified candidate, or roll back if no candidate improves the verified utility
    End of inner repeat block
    Standardize the shaped returns within each proposal group to obtain group-relative advantages
    Update SKETCH on the clipped group-relative surrogate with a divergence penalty toward the reference policy
    Refresh ECHO by regression on the replay buffer, using folded outcomes only and never its own predictions
    Refresh verified outcomes together with constructed negatives
    Recompute the acceptance threshold from the refreshed set of known-infeasible calibration designs
End of repeat block
Return the trained proposer, world model, and verifier

A.3 BACKBONE-CONDITIONED NON-AUTOREGRESSIVE INITIALIZER

This subsection gives the full specification of the initializer summarized in the main paper. The model is pretrained independently and remains frozen during verifier-guided reinforcement learning, where it serves three roles: it defines the initial candidate population, it supplies the structure-conditioned sequence log-likelihood used as the reward prior, and it supplies the position-level and base-level mutation priors used by SKETCH and SCOUT.

A.3.1 SEQUENCE-INDEPENDENT GEOMETRIC REPRESENTATION

Let the target RNA backbone be $C = \{\mathbf{c}_i\}_{i=1}^{L}$. For each nucleotide we extract three sequence-independent atoms,

$$\mathbf{c}_i = \left(\mathbf{c}_i^{P}, \mathbf{c}_i^{C4'}, \mathbf{c}_i^{C1'}\right) \in \mathbb{R}^{3\times 3}.$$

In contrast to representations that use N1 for pyrimidines and N9 for purines, the selected atoms do not reveal the native nucleotide class, so the structure encoder receives no explicit or atom-type-induced nucleotide identity. This matters for the design task, since an encoder that could infer purine or pyrimidine identity from atom typing would leak part of the answer.

We construct a directed sparse graph $\mathcal{G}_C = (\mathcal{V}, \mathcal{E})$, where each node represents one valid nucleotide, and the edge set is the union of self edges, directed adjacent-sequence edges, long-range sequence edges, and spatial edges obtained from a k-nearest-neighbour graph over the $C4'$ coordinates,

$$\mathcal{E} = \mathcal{E}_{\text{self}} \cup \mathcal{E}_{\text{seq}} \cup \mathcal{E}_{\text{long}} \cup \mathcal{E}_{\text{spatial}}.$$

To encode local RNA orientation we construct a right-handed coordinate frame per residue. Taking $C4'$ as the frame origin,

$$\begin{aligned}
\mathbf{x}_i &= \operatorname{Norm}\left(\mathbf{c}_i^{P} - \mathbf{c}_i^{C4'}\right), \\
\widetilde{\mathbf{y}}_i &= \mathbf{c}_i^{C1'} - \mathbf{c}_i^{C4'} - \left[\left(\mathbf{c}_i^{C1'} - \mathbf{c}_i^{C4'}\right)^{\top} \mathbf{x}_i\right] \mathbf{x}_i, \\
\mathbf{y}_i &= \operatorname{Norm}(\widetilde{\mathbf{y}}_i), \qquad \mathbf{z}_i = \operatorname{Norm}(\mathbf{x}_i \times \mathbf{y}_i),
\end{aligned}$$

and the local frame F_i stacks $\mathbf{x}_i$, $\mathbf{y}_i$, and $\mathbf{z}_i$ row-wise.

Each node carries invariant scalar features $\mathbf{x}_i^s$ and equivariant vector features $\mathbf{x}_i^v$. The scalar representation describes local backbone geometry, atom validity, distances, local curvature, and frame-relative geometric quantities. The vector representation contains the local frame axes and the backbone direction, $\mathbf{x}_i^v = [\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i, \mathbf{t}_i]$, where $\mathbf{t}_i$ denotes the local backbone tangent.

For each directed edge (j, i), the scalar edge representation contains radial-basis encodings of inter-residue distances, relative sequence-position encodings, source and target directions expressed in the corresponding local frames, the relative frame rotation $F_i F_j^{\top}$, frame-validity indicators, and a learned edge-type embedding. The normalized displacement between the source and target $C4'$ atoms is retained as an equivariant edge vector.

A.3.2 SPARSE GEOMETRIC TRANSFORMER ENCODER

The scalar node, vector node, and edge features are embedded as

$$\begin{aligned}
\mathbf{h}_i^{(0)} &= \mathbf{h}_{\text{mask}} + \operatorname{MLP}_n(\mathbf{x}_i^s), \\
\mathbf{V}_i^{(0)} &= \operatorname{VecLinear}(\mathbf{x}_i^v), \qquad \mathbf{e}_{ji}^{(0)} = \operatorname{MLP}_e(\mathbf{x}_{ji}^e),
\end{aligned}$$

where the shared mask embedding $\mathbf{h}_{\text{mask}}$ is used at every residue, so no nucleotide identities enter the encoder.

We then apply K sparse geometric Transformer layers, each jointly updating invariant scalar states, equivariant vector states, and edge states,

$$\begin{aligned}
\left(\mathbf{h}_i^{(\ell+1)}, \mathbf{V}_i^{(\ell+1)}, \mathbf{e}_{ji}^{(\ell+1)}\right) = \operatorname{GeoTransformer}_{\ell} \\
\left(\mathbf{h}_i^{(\ell)}, \mathbf{V}_i^{(\ell)}, \Xi_i^{(\ell)}\right),
\end{aligned}$$

where $\Xi_i^{(\ell)} = \{\mathbf{h}_j^{(\ell)}, \mathbf{V}_j^{(\ell)}, \mathbf{e}_{ji}^{(\ell)}\}_{j\in\mathcal{N}(i)}$ collects the neighbourhood messages.

For attention head m, the score from source residue j to target residue i is

$$\begin{aligned}
a_{ji}^m = \frac{\left(\mathbf{q}_i^m\right)^{\top}\left(\mathbf{k}_j^m + \mathbf{k}_{ji}^m\right)}{\sqrt{d_h}} + b_{ji}^m + \lambda_v^m \left\langle \mathbf{Q}_i^m, \mathbf{K}_j^m \right\rangle \\
+ \lambda_t^m \left\langle \mathbf{Q}_i^m, \mathbf{d}_{ji} \right\rangle + \lambda_s^m \left\langle \mathbf{K}_j^m, \mathbf{d}_{ji} \right\rangle,
\end{aligned}$$

where $\mathbf{k}_{ji}^m$ and b_{ji}^m come from the edge state, $\mathbf{Q}_i^m$ and $\mathbf{K}_j^m$ are vector queries and keys, and $\mathbf{d}_{ji}$ is the equivariant edge direction. The normalized weight is

$$\alpha_{ji}^m = \frac{\exp(a_{ji}^m)}{\sum_{j' \in \mathcal{N}(i)} \exp(a_{j'i}^m)}.$$

Vector information affects attention only through invariant inner products and directional projections, so the scalar attention weights are rotation invariant while the resulting vector updates remain rotation equivariant.

After K layers, vector-channel magnitudes are projected back into the scalar representation,

$$\overline{\mathbf{h}}_i = \mathrm{LN}\left[\mathbf{h}_i^{(K)} + \mathrm{MLP}_v\left(\left\|\mathbf{V}_i^{(K)}\right\|\right)\right].$$

A.3.3 Non-autoregressive sequence decoder

A residue-wise classification head maps each encoded nucleotide to four logits, $\mathbf{z}_i = f_{\mathrm{NAR}}(\overline{\mathbf{h}}_i) \in \mathbb{R}^4$, over the alphabet $\mathcal{A} = \{A, G, C, U\}$, with

$$q_\omega(y_i = b \mid C) = \mathrm{Softmax}(\mathbf{z}_i)_b, \qquad q_\omega(y \mid C) = \prod_{i=1}^{L} q_\omega(y_i \mid C).$$

Predicting all positions in parallel avoids imposing an arbitrary left-to-right generation order and allows many complete sequences to be sampled after a single backbone-encoding pass, which is what makes population initialization inexpensive.

A.3.4 Training objective

For a native structure and sequence pair (C, y^*), the initializer is trained with residue-level cross-entropy,

$$\mathcal{L}_{\mathrm{NAR}} = -\frac{1}{|V_C|} \sum_{i \in V_C} \log q_\omega(y_i^* \mid C),$$

where V_C contains residues with valid geometric inputs. With label smoothing coefficient ϵ_{ls}, the target distribution is $\widetilde{p}_{i,b} = (1 - \epsilon_{\mathrm{ls}})\mathbb{I}[b = y_i^*] + \epsilon_{\mathrm{ls}}/4$. The encoder optionally predicts the clean $C4'$ distance for each non-self edge,

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{|\mathcal{E}'|} \sum_{(j,i) \in \mathcal{E}'} \mathrm{SmoothL1}\left(\widehat{d}_{ji}, \left\|\mathbf{c}_j^{C4'} - \mathbf{c}_i^{C4'}\right\|_2\right),$$

where $\mathcal{E}'$ excludes self edges, giving the complete objective $\mathcal{L}_{\mathrm{init}} = \mathcal{L}_{\mathrm{NAR}} + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}}$. In our main configuration the model is optimized primarily with the sequence loss, and the distance term is treated as an optional geometric regularizer.

A.3.5 Parallel candidate initialization and selection

Given a sampling temperature T_0, a candidate nucleotide at position i is drawn as

$$y_i^{(k)} \sim \mathrm{Categorical}\left[\mathrm{Softmax}\left(\frac{\mathbf{z}_i}{T_0}\right)\right], \qquad k = 1, \ldots, K_0,$$

with optional top-k or nucleus filtering. Because the logits are computed once and positions are conditionally independent, the population $\mathcal{Y}_0$ is generated in parallel. en scores each candidate by

$$S_{\mathrm{init}}(y) = -\lambda_E E_\phi(x, y, r_{\mathrm{init}}) + \eta \ell_\omega(y \mid C) + \lambda_{\mathrm{con}} C_{\mathrm{seq}}(y),$$

where $C_{\mathrm{seq}}(y)$ measures satisfaction of sequence constraints such as fixed residues, GC-content requirements, and forbidden motifs, and the highest-scoring feasible candidate becomes y_0.

A.3.6 STRUCTURE-CONDITIONED PRIORS

The structure-conditioned perplexity corresponding to ℓ_ω is $\mathrm{PPL}(y \mid C) = \exp[-\ell_\omega(y \mid C)]$. The residue-wise predictive entropy is

$$u_i = -\sum_{b \in \mathcal{A}} q_\omega(y_i = b \mid C) \log q_\omega(y_i = b \mid C),$$

where low entropy indicates that the target geometry strongly supports one nucleotide identity and high entropy indicates that several identities remain structurally plausible. We therefore define the position-level editability prior

$$p_{\mathrm{edit}}(i \mid C) = \frac{\exp(\kappa u_i)}{\sum_{j=1}^{L} \exp(\kappa u_j)},$$

which concentrates proposal mass on geometrically ambiguous positions, and combine it with the base-level prior gain Δ_ω defined in the main paper.

A.4 VERIFIED TRANSITION BANK AND TRAINING PIPELINE

The modules are trained in three stages: the initializer is pretrained first, a bank of folding-verified transitions is then constructed, and ECHO, nd SKETCH are trained against that bank. This subsection describes the bank and the stage ordering; the module-specific details follow in Appendices A.5 and A.6.

A.4.1 TRANSITION REPRESENTATION

We represent an RNA design state as a node and a single sequence edit as a directed edge, so one training example is the triple

$$\big(y_{\mathrm{parent}},\ a,\ y_{\mathrm{child}}\big).$$

Both endpoints are submitted to the real folding tools, and for each endpoint we record TM-score, lDDT, RMSD, the number of tools that returned a valid prediction, and the edit type, edit positions, and mutation mask that produced the child. The three structural metrics are standardized using training-set statistics and oriented so that higher is better, giving the scalar quality state

$$Q(y) = 0.4\, z_{\mathrm{TM}}(y) + 0.4\, z_{\mathrm{lDDT}}(y) + 0.2\, z^{+}_{\mathrm{RMSD}}(y),$$

and the supervision target for ECHO is the difference

$$\Delta Q = Q(y_{\mathrm{child}}) - Q(y_{\mathrm{parent}}).$$

The bank is therefore a dataset of verified transitions rather than a dataset of independently labelled sequences, which is what allows the surrogate to be trained on the effect of an edit rather than on the absolute difficulty of a target.

A.4.2 EDIT SAMPLING FOR COVERAGE

Two choices keep the bank from collapsing onto easy or one-sided transitions. First, edits are sampled so that roughly half move the sequence toward the native identity and roughly half move it away, so the surrogate observes both improving and degrading transitions rather than only the improving direction that a greedy generator would produce. Second, edit multiplicity is mixed, with approximately 40 percent single-position edits, 40 percent two-position edits, and 20 percent edits touching more than two positions, so that the surrogate is not fitted exclusively to minimal perturbations.

A.4.3 STAGE ORDERING

Stage 0 pretrains the initializer as described above and freezes it. Stage 1 constructs the transition bank by applying sampled edits to initializer outputs and folding both endpoints. Stage 2 trains ECHO offline on the bank, trains the same verified outcomes together with constructed negatives, and initializes SKETCH from structured edit examples derived from the bank. Stage 3 is the verifier-guided policy optimization of Algorithm 2, in which fresh verified transitions are appended to the bank as they are produced.

A.4.4 TRAINING STATE USED IN THE REPORTED EXPERIMENTS

The results reported in the main paper are produced with all modules held fixed rather than updated online. The initializer, SKETCH, and ECHO are loaded from their trained checkpoints and frozen, used for candidate scoring and for the acceptance gate without further updating, and SCOUT is a fixed tree planner with no learned parameters of its own. Real folding results are still written to the cache and to the verified-transition files during evaluation, but they are not consumed to update any module within the reported runs. The reported comparisons therefore measure the inference-time contribution of each module at a common trained state, and they do not measure the effect of retraining a separate model for each configuration.

A.5 ECHO ARCHITECTURE AND TRAINING DETAILS

ECHO operates on an edit-aware graph derived from the parent state. Node features reuse the geometric scalar features of the initializer graph, augmented with the current nucleotide identity, a binary indicator of whether the position is touched by the candidate action, and the parent per-metric quality values broadcast to every node. Edge features reuse the initializer edge construction. The applied action is encoded by a learned embedding of its operator type from the grounded vocabulary together with the substituted base identities and the mutation mask, and this embedding is concatenated to the node features of the affected positions and to a global graph token. Message passing is followed by mean pooling over nodes and concatenation with the global action token, and two heads read out from the pooled representation: a scalar head producing $\hat{\Delta}_t$ and a three-way head producing $\hat{s}_c$. Layer counts, hidden widths, and loss weights are fixed by the released configuration files rather than tuned per experiment.

Three input restrictions matter for the validity of the surrogate. ECHO receives the parent sequence, the child sequence, the edit action and mask, and the already-verified scalar quality state of the parent. It does not receive the target structure, it does not predict coordinates, and it never receives the verified quality of the child, which is the quantity it is asked to predict. Supplying the parent metrics as inputs rather than predicting them removes the need for the surrogate to model absolute target difficulty and lets it concentrate on the effect of the edit; withholding the child metrics prevents the prediction target from leaking into the input.

An earlier version of the module predicted all nine combinations of three tools and three metrics. We replaced it with the present scalar-utility transition model because a single standardized delta is both the quantity actually needed for ranking and a more stable regression target than nine correlated per-tool values. Training uses only the verified transitions in the bank, so ECHO is refreshed from folded outcomes and never from its own predictions, and the standardization statistics are fixed on the training distribution so that the scale of $\hat{\Delta}$ does not drift across rounds. The checkpoint used in all reported experiments is the scalar-utility model selected by validation loss.

A.6 CHITECTURE AND CALIBRATION DETAILS

codes three inputs separately. The context x is encoded from the target backbone graph using the same geometric features as the initializer, with additional channels for the optional 2D target and constraint specification. The candidate sequence y is encoded by a sequence encoder over the nucleotide alphabet. The reasoning trace r is encoded by a text encoder applied to the emitted rationale together with the parsed structured action. The three representations are fused by cross-attention, with the sequence representation attending to both the context and the reasoning representation, and a scalar head reads the fused state to produce $E_\phi(x, y, r)$. The objective is a learned energy in which a low value denotes a design that is physically consistent with the context and logically supported by its stated rationale, rather than a direct regression onto a scalar reward.

Negative examples for the contrastive term are constructed by structure-breaking perturbations of a valid design, including disruption of canonical pairs implied by the backbone, substitutions that introduce steric conflicts at close contacts, and randomized replacement of nucleotides at low-entropy positions identified by p_{edit}. Negative traces for the preference term are generated by corrupting a coherent rationale so that its stated biophysical effect no longer matches the action it accompanies, which trains the verifier to penalize rationales that cannot be checked against the edit they justify. The calibration term maps the energy onto folding quality averaged over the design-time folding en-

Table 6: SKETCH grounded reasoning operators. † also apply to 2D design; ⋆ are binding-conditioned operators.

Operator	Biophysical rationale	Concrete edit
`enforce_canonical_pair`	backbone geometry implies a Watson-Crick pair	set bases to G-C or A-U
`place_stacking_purine`	strengthen base stacking	substitute toward purine stack
`satisfy_tertiary_contact`	support an A-minor motif or base triple	edit interacting nucleotides
`resolve_steric_clash`	relieve groove clash	swap to a smaller base
`stabilize_stem`†	increase helix stacking energy	mutate pair to G-C
`tune_loop`†	satisfy loop length or identity	edit loop nucleotides
`adjust_GC`†	meet GC constraint	substitute A/U ↔ G/C
`install_aptamer_motif`⋆	seed a binding-competent motif	graft or edit pocket nucleotides
`shape_binding_pocket`⋆	open or close pocket for ligand or RBP	edit bulge or loop geometry

semble rather than onto a single predictor, which is deliberate: calibrating to one oracle would make the energy a proxy for that oracle, and the resulting acceptance decisions would inherit its idiosyncrasies. The acceptance threshold τ is then set on a calibration set of designs known to be infeasible for their target, following the one-sided split-conformal construction analysed in Appendix A.11.

At inference ays three roles that are all exercised in the reported runs: it contributes an energy penalty to the candidate ranking used by SCOUT, it supplies the dense shaping signal $-E_\phi$, and it acts as the conformal accept-reject gate over exported leaves. We are explicit about one gap. The runs confirm that connected to the tree-search scoring and to the gate, but we do not have training logs complete enough to report its optimization schedule or a specific final checkpoint identifier, and the reported experiments do not perform the online alternating update of scribed in Algorithm 2. We therefore describe an offline-trained energy verifier used in a frozen state, and we do not claim that the reported numbers demonstrate the fully online variant of the training loop.

A.7 GROUNDED OPERATOR VOCABULARY AND PROMPT FORMAT

Table 6 lists the grounded operators. Each operator is specified by a name, an argument signature, a precondition that must hold in the current state, and a deterministic edit rule, so that an emitted action either applies unambiguously or is rejected as illegal before any scoring takes place. The vocabulary is partitioned into tertiary operators, operators that also apply to secondary design, and binding-conditioned operators, and the complete specification with argument signatures is released with the code.

SKETCH is prompted with the target identifier, the current sequence, the step index, the remaining edit budget, and the legal alphabet, and is required to return a JSON object containing a free-text rationale field and a structured action field. The action field names one operator and supplies its arguments. Generation attempts that fail to parse, that name an operator outside the vocabulary, that violate an operator precondition, or that duplicate an already-proposed child are discarded and resampled up to the attempt limit. Because the rationale and the action are emitted together and the action is executable, every proposal is auditable in the sense that its stated justification can be compared against the edit actually performed and against the resulting verified quality change.

A.8 SEARCH AND SELECTION DETAILS

At each state, SKETCH is sampled repeatedly until a target number of legal and non-duplicate children has been collected or the attempt limit is reached. Duplicates are detected on the resulting child sequence rather than on the action, so two different operators that produce the same sequence contribute only one candidate. All collected children are scored by ECHO and ranked by the utility U defined in the main paper, and only the top-k are refolded.

Expansion proceeds over a bounded tree. The number of parents that may be expanded, the number of children retained per parent, the number of simulations, and the maximum depth are all capped, with the reported configuration listed in Table 7. Since several parents may be expanded, the tree contains more nodes than the parent cap. After search terminates, the leaf with the highest aggregate score is exported as the proposal for real verification.

Acceptance is decided only after verification. The proposal is accepted when its verified utility exceeds that of the current sequence and is otherwise rolled back, so a step that the surrogate favoured but the folder did not confirm cannot advance the state. This is the mechanism that keeps surrogate error from accumulating across depth.

A.9 Dataset Construction and Preprocessing

We construct the main tertiary dataset from RNAsolo[2] (Adamczyk et al., 2022; 2025). To maximize structural coverage, we combine the structures collected under the BGSU and Rfam repositories. Since the two repositories partially overlap, the merged collection is deduplicated according to the PDB identifier, model identifier, and RNA chain identifier before further processing. We retain structures satisfying the RNAsolo[2] resolution filter of at most 3.0 Å. Each RNA chain is extracted and treated as an individual design target, and only single-chain RNAs with lengths between 10 and 500 nucleotides are retained. Structures with invalid coordinates, severe backbone incompleteness, or unsupported residues are removed.

To reduce both sequence and structural redundancy, we first cluster RNA chains at a sequence-identity threshold of 30 percent, and then perform structure-based clustering using a TM-score threshold of 0.5. The union of the sequence-similarity and structure-similarity relations is used to form disjoint connected components, ensuring that RNA chains related by either criterion are assigned to the same split. The resulting components are divided into training, validation, and test sets with an approximate ratio of 8:1:1. Generalization sets are held out entirely: RNA-Puzzles targets (Cruz et al., 2012), CASP15 RNA targets, and cross-family Rfam targets (Kalvari et al., 2021).

For binding, we curate two benchmarks. The RNA-ligand set combines HARIBOSS structures of RNA-small-molecule complexes (Panei et al., 2022), ROBIN binders (Yazdani et al., 2023), and Rfam riboswitch aptamer domains. The RBP-RNA set combines RNAcompete in-vitro binding specificities (Ray et al., 2013) and ENCODE eCLIP in-vivo binding maps (Van Nostrand et al., 2020), with motif references taken from ATtRACT (Giudice et al., 2016). In both cases the target supplies a structure context together with the binding partner, and the binding predictors used for reporting are disjoint from any predictor used inside the loop.

A.10 Ablation Protocol Details

All ablations reuse one model checkpoint and perform inference only, so differences isolate the search and scoring modules rather than training. We fix a stratified subset of 60 targets drawn from the 369 test targets, retain one initializer sequence per target as the root, and reuse the root structural metrics without refolding. Folding results are cached and reused across configurations, keyed by the tuple of target, sequence, tool, and configuration, so repeated states incur no additional oracle calls and the number of new folding calls is generally below the nominal budget.

Tree search never queries the true folder. Candidate expansion and ranking use only the learned scores, and real verification is performed once per target after search terminates, on the exported top-1 leaf. Table 7 lists the search configuration. Because up to four parents may be expanded and each expanded parent may retain up to four children, the search tree contains substantially more than four nodes; the parent cap bounds how many nodes are expanded, not how many nodes exist.

We compare five configurations. *Full* enables SKETCH proposal, ECHO utility, ergy, and the value score at depth 3. *Single-step* sets the maximum depth to 1 and the parent cap to 1, leaving all candidate-generation and scoring settings unchanged, so the root is expanded once, the resulting first-layer children are themselves the leaves, and none of them is expanded further. This is not a zero-expansion setting, and its search budget is smaller than that of the full loop because only one parent is ever expanded. ECHO-*only guidance* retains ECHO utility but removes the ergy and value contributions from ranking. *Policy-prior-only ranking* removes ECHO, nd value from ranking and orders candidates by the policy prior alone. *Uniform-random slate* replaces proposal selection by uniform sampling from the same legal candidate slate, which perturbs selection rather than introducing random mutations. When the conformal gate remains active, ill filters leaves, so the corresponding rows remove the ore rather than tirely.

Table 7: **Search configuration used in all reported runs.** The parent cap bounds the number of nodes that may be expanded, not the size of the tree. Real verification is performed after search terminates rather than inside it.

Parameter	Value	Meaning
max depth	3	longest edit chain from root to a leaf
simulations	4	tree traversals per target
max expanded parents	4	nodes allowed to generate children
max children per parent	4	legal non-duplicate children retained
expansion group size	4	legal candidates sought per expansion
expansion attempts	8	generation attempts per expansion
exported leaves	1	proposals sent to real verification
in-search folding calls	0	folder is never queried during search

A.10.1 WHY THE DEPTH ROWS AND THE SINGLE-STEP ROW ARE NOT BUDGET-MATCHED

The depth-1, depth-2, and depth-3 rows are recorded from the full closed-loop trajectory, in which each round collects up to eight candidates per state, submits the top five by predicted utility to real folding, and selects the proposal by verified utility. The single-step row instead collects up to four children, selects the top-1 by surrogate score alone, and folds only that one candidate. Two differences follow. The single-step configuration sees a smaller candidate pool, and it commits to a surrogate-selected proposal without the opportunity to compare several verified outcomes. The gap between the single-step row and the depth-1 row therefore reflects both search depth and verification budget, and it should not be read as an estimate of the value of depth alone. We report both rows because they answer different questions, the depth rows asking whether recursive repair helps and the single-step row asking how much is lost when only one candidate is ever verified.

A.10.2 VERIFICATION, UTILITY, AND ACCEPTANCE

Real verification uses Boltz-2, trRosettaRNA2, and Protenix, giving at most $60 \times 3 = 180$ tool-level results per configuration before cache reuse. For each sequence we aggregate TM-score, lDDT, and RMSD across the tools that returned a valid prediction and form a single utility with a fixed root normalizer,

$$U = 0.4\, z_{\mathrm{TM}} + 0.4\, z_{\mathrm{lDDT}} + 0.2\, z^{+}_{\mathrm{RMSD}},$$

where the RMSD direction is inverted so that every component is higher-better. A proposal is accepted when $U_{\mathrm{proposal}} - U_{\mathrm{current}} > 0$ and is otherwise rolled back. Each round records three sequences: the sequence at the start of the round, the proposal selected by search, and the accepted sequence after verification and possible rollback. All reported structural and recovery quantities are read from the accepted sequence. We compute recovery against the native sequence for all three states and report the change from the round-start sequence to the accepted sequence, retaining the change to the proposal as a secondary quantity; reading recovery from the round-start field instead would report the unedited root and understate the effect of editing.

A.11 PROOFS AND DISCUSSION OF THE TWO GUARANTEES

A.11.1 PROPOSITION 1

The result is the potential-based shaping theorem of Ng et al. (1999). For any bounded Φ and shaping term $F(s, a, s') = \gamma\Phi(s') - \Phi(s)$, the optimal policies of the shaped and unshaped problems coincide, because the return of any trajectory differs between the two problems only by the telescoping quantity $\Phi(s_0)$ up to discounting, which does not depend on the policy. In our setting $\Phi(s) = -E_\phi(x, y, r)$ is bounded because the energy head has bounded output, and the shaping term is added to a task-progress difference rather than replacing it, so the argument applies unchanged. The practical consequence is that an imperfect verifier can slow learning or waste folding budget, but it cannot make a design preferable that the true objective R would reject. We regard this as the property that makes it acceptable to use a learned verifier as a dense reward at all.

A.11.2 PROPOSITION 2

The result is the standard split-conformal guarantee (Angelopoulos & Bates, 2023) applied to a one-sided threshold. If the calibration energies $\{e_j\}_{j=1}^{n}$ of known-infeasible designs and the energy of a new infeasible design are exchangeable, then the rank of the new energy among the calibration energies is uniform, so setting τ to the empirical α-quantile bounds the probability that a new infeasible design falls below τ by $\alpha + \frac{1}{n+1}$.

Two assumptions deserve emphasis. First, the guarantee is a statement about infeasible designs only, so it controls false acceptance and says nothing about how many feasible designs are rejected; a conservative α therefore trades yield for reliability. Second, exchangeability requires that the calibration set be drawn from the same distribution as the infeasible designs encountered at inference. This holds by construction while the calibration set is refreshed from verified transitions of the same target distribution, but it would be violated if the agent were deployed on a family of targets whose failure modes differ systematically from those seen during calibration. We therefore treat the stated bound as valid within the evaluated distribution rather than as a universal certificate.

A.12 IMPLEMENTATION AND CONFIGURATION

SKETCH is Qwen3.5-9B-Instruct, adapted with LoRA rather than full fine-tuning. Training proceeds in two steps. Offline policy initialization constructs structured edit examples from verified transitions and ECHO candidate slates and trains the LoRA adapter on them; the resulting adapter initializes both the trainable policy and the frozen reference policy. Verifier-guided optimization then samples a group of edits at each state, forms group-relative advantages from ECHO, he value score, and the verified outcome, updates the policy in the GRPO style of the main paper, and regularizes toward the frozen reference with a KL penalty.

Training ran on V100 hardware, which constrained the configuration. The stable setting enabled gradient checkpointing with a maximum sequence length of 384, a per-device batch size of 1, and 16 gradient-accumulation steps. Training past this point produced non-finite gradients and was not adopted, so all reported editing and ablation results use the round-16 checkpoint of the joint training run, and no further policy updates were applied.

The remaining modules are used in the state described in Appendix A.4: the initializer is loaded from its pretrained checkpoint and frozen, ECHO is loaded from the scalar-utility checkpoint and frozen, used for energy scoring and conformal gating without online updating, and SCOUT is a fixed selector that combines ECHO utility, the policy prior, the ergy, and an optional value score under a PUCT-style tree expansion. Folding verification uses Boltz-2, trRosettaRNA2, and Protenix, with results cached by target, sequence, tool, and configuration so that repeated states do not incur additional oracle calls. Layer counts, hidden widths, loss weights, learning rates, and search budgets are recorded in the released configuration files, which are published together with the code, the operator specification, and the preprocessing scripts at the anonymous repository given in the abstract of the main paper.

A.13 ADDITIONAL RESULTS

This subsection reports the per-predictor breakdowns omitted from the main paper for space. Table 8 reports the two held-out generalization benchmarks per method. Table 9 gives the per-predictor breakdown by sequence length, and Table 10 the per-predictor breakdown by Rfam family. The comparison against thermodynamic and learned secondary-structure solvers on the tertiary split is reported in Table 2, and the full category-level breakdown is reported in Table 3, so neither is duplicated here.

A.14 OUTLOOK

Three directions follow directly from the present results. First, all evidence reported here is computational, so prospective validation through chemical probing, binding assays, and functional measurement is the natural next step; the folding and binding panels used for reporting are predictors, not measurements, and agreement across three disjoint predictors bounds only predictor-level consistency. Second, the binding evaluation covers only the structure-only variant RIVET$_{\zeta=0}$, so the

Table 8: **Generalization** to held-out RNA-Puzzles and CASP15. scTM is the mean over the three tertiary predictors. Rows are ordered by RNA-Puzzles recovery. Best per column in **bold** and second best underlined.

Method	RNA-Puzzles		CASP15	
	Rec.%↑	scTM↑	Rec.%↑	scTM↑
RIDER	0.358	0.189	0.271	0.149
gRNAde	0.407	0.190	0.288	0.143
RiboDiffusion	0.465	<u>0.294</u>	<u>0.351</u>	**0.241**
RDesign	<u>0.471</u>	0.229	0.324	0.190
RIVET (ours)	**0.631**	**0.428**	**0.390**	**0.241**

Table 9: **Sequence recovery and structural self-consistency across RNA lengths.** Recovery is reported as a percentage, and structural self-consistency is scTM computed from structures predicted by Protenix, Boltz-2, and trRosettaRNA.

	Short ($L \leq 50$)				**Medium** ($50 < L \leq 100$)				**Long** ($L > 100$)				**All**
Method	**Rec.**	**Ptx**	**B2**	**trR**	**Rec.**	**Ptx**	**B2**	**trR**	**Rec.**	**Ptx**	**B2**	**trR**	**Rec.**
RIDER	38.4	0.168	0.176	0.133	42.5	0.158	0.168	0.127	44.1	0.131	0.197	0.109	41.8
RDesign	47.8	0.250	0.253	0.201	54.8	0.377	0.477	0.419	52.4	0.212	0.573	0.264	52.3
gRNAde	37.2	0.251	0.258	0.220	42.6	0.268	0.287	0.226	40.7	0.215	0.223	0.206	40.8
RiboDiffusion	36.9	0.256	0.257	0.219	46.4	0.411	0.480	0.357	51.1	0.320	0.670	0.793	45.4
RNAinverse	29.8	0.230	0.242	0.184	28.5	0.287	0.333	0.264	31.8	0.237	0.252	0.239	29.6
MCTS-RNA	26.9	0.234	0.239	0.188	31.0	0.278	0.334	0.281	29.0	0.207	0.237	0.219	29.3
antaRNA	25.6	0.222	0.233	0.176	26.6	0.251	0.283	0.233	26.3	0.177	0.198	0.173	26.2
LEARNA	25.1	0.220	0.229	0.183	26.2	0.290	0.330	0.293	29.3	0.209	0.255	0.256	26.2
Meta-LEARNA	28.5	0.241	0.244	0.191	31.7	0.267	0.293	0.243	32.7	0.226	0.260	0.227	31.2
eM2dRNAs	25.1	0.248	0.258	0.180	33.2	0.290	0.322	0.275	33.3	0.228	0.267	0.248	30.8
RIVET (ours)	**54.9**	0.270	0.276	0.235	**55.3**	0.555	0.547	0.477	**58.9**	0.364	0.734	0.793	**56.3**

Note: Rec. denotes sequence recovery in percent. Ptx, B2, and trR denote scTM scores computed from structures predicted by Protenix, Boltz-2, and trRosettaRNA, respectively.

incremental value of explicit binding conditioning with $\zeta > 0$, and whether the grounded binding operators recover pocket geometry that structural accuracy alone does not, remain open. Third, the folding budget is currently spent uniformly on the top-k candidates; an uncertainty-aware world model would instead allocate verification where the surrogate is least confident, which should raise the yield of each oracle call and matters most on the long targets where the loop already gains the most. A fourth and longer-term direction is to refresh d the conformal calibration set online during deployment, which Algorithm 2 specifies but the reported runs do not exercise.

A.15 Limitations and Ethical Considerations

RIVET is a computational framework, and its results require prospective experimental validation. Chemical probing, binding assays, and functional measurements are needed to confirm designed structures and activities in vitro or in cellular settings. Its performance may also depend on the coverage of available RNA structure and interaction data, especially for rare tertiary motifs, ligand classes, or protein-binding contexts. Future work should incorporate broader datasets, uncertainty-aware models, and active experimental feedback.

RIVET is intended for beneficial RNA design applications, including therapeutic discovery, molecular probes, and synthetic biology research. Its outputs are candidate sequences rather than validated biological products. Downstream use should follow appropriate biosafety, biosecurity, and regula-

Table 10: **Sequence recovery and structural self-consistency across Rfam families.** For each family, the first line reports sequence recovery and the second line reports scTM in the order Protenix / Boltz-2 / trRosettaRNA. Low-frequency families are grouped into Others.

Method	RF00005	RF00001	RF02541	RF00168	RF02399	Others	All
	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR	Rec. Ptx/B2/trR
RIDER	0.478 0.161/0.190/0.131	0.524 0.146/0.278/0.124	0.446 0.119/0.131/0.101	0.382 0.117/0.126/0.094	0.382 0.158/0.202/0.142	0.367 0.155/0.155/0.122	0.418 0.150/0.181/0.122
RDesign	0.635 0.419/0.626/0.481	0.607 0.263/0.506/0.305	0.443 0.205/0.226/0.197	0.519 0.239/0.273/0.225	0.622 0.180/0.224/0.157	0.458 0.255/0.261/0.272	0.523 0.283/0.369/0.286
gRNAde	0.489 0.282/0.307/0.269	0.454 0.261/0.265/0.236	0.380 0.194/0.190/0.180	0.408 0.249/0.260/0.198	0.159 0.091/0.114/0.090	0.368 0.252/0.253/0.214	0.408 0.253/0.264/0.235
RiboDiffusion	0.561 0.462/0.638/0.519	0.617 0.463/0.731/0.747	0.581 0.336/0.268/0.272	0.350 0.259/0.801/0.221	0.191 0.203/0.191/0.164	0.375 0.258/0.275/0.225	0.454 0.310/0.370/0.368
RNAinverse	0.288 0.287/0.363/0.267	0.315 0.259/0.273/0.275	0.316 0.197/0.168/0.148	0.316 0.181/0.196/0.157	0.442 0.168/0.179/0.165	0.280 0.262/0.271/0.221	0.296 0.258/0.285/0.235
MCTS-RNA	0.325 0.294/0.391/0.331	0.301 0.225/0.285/0.289	0.272 0.157/0.167/0.125	0.281 0.209/0.239/0.209	0.212 0.171/0.194/0.151	0.287 0.248/0.258/0.211	0.293 0.245/0.280/0.239
antaRNA	0.259 0.251/0.301/0.248	0.269 0.189/0.218/0.198	0.251 0.149/0.152/0.132	0.258 0.192/0.223/0.179	0.223 0.160/0.181/0.172	0.262 0.231/0.241/0.192	0.262 0.220/0.243/0.199
LEARNA	0.244 0.293/0.357/0.318	0.322 0.241/0.287/0.327	0.269 0.215/0.134/0.165	0.233 0.182/0.262/0.152	0.232 0.165/0.172/0.156	0.267 0.252/0.260/0.215	0.262 0.255/0.285/0.250
Meta-LEARNA	0.294 0.276/0.303/0.252	0.339 0.245/0.314/0.283	0.291 0.174/0.188/0.152	0.329 0.248/0.256/0.211	0.360 0.163/0.191/0.148	0.311 0.258/0.263/0.215	0.312 0.248/0.271/0.226
eM2dRNAs	0.339 0.294/0.269/0.247	0.335 0.231/0.310/0.215	0.329 0.196/0.276/0.141	0.333 0.146/0.197/0.156	0.172 0.178/0.181/0.128	0.307 0.272/0.181/0.178	0.308 0.262/0.273/0.191
RIVET (ours)	**0.618** **0.479/0.665/0.583**	**0.650** **0.523/0.805/0.842**	**0.701** **0.402/0.689/0.251**	**0.448** **0.286/0.305/0.243**	**0.505** **0.193/0.191/0.173**	**0.514** **0.282/0.311/0.254**	**0.563** **0.366/0.473/0.416**

Note: Rec. denotes sequence recovery. Ptx, B2, and trR denote scTM computed from structures predicted by Protenix, Boltz-2, and trRosettaRNA. Others contains families whose sample counts fall below the category threshold. All denotes results over the complete test set. RF00005 is tRNA, RF00001 is 5S rRNA, RF02541 is bacterial LSU rRNA, RF00168 is the lysine riboswitch, and RF02399 is NsiR1.

tory procedures, including checks for off-target effects, immunogenicity, unintended interactions, and context-dependent activity.

A.16 Generative AI Usage

Generative AI tools were used solely for manuscript preparation, specifically language polishing and organization of already-written text.