RECURRENT GEOMETRIC COUPLING REFINEMENT FOR RNA INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA inverse folding seeks nucleotide sequences compatible with a target three-dimensional structure. Base-pairing and stacking interactions couple nucleotide choices in RNA, yet sampling positions independently within each generation step can fail to preserve their joint compatibility. We introduce Recurrent Geometric Coupling Refinement (RGCR), which builds on masked diffusion language modeling (MDLM) to learn joint nucleotide distributions over geometry-derived pairs and triples. Estimates of their expected log-score gains over singleton predictions guide geometric message passing that recurrently refines both the joint distributions and these estimates. The final scores select disjoint groups for joint sampling among the positions generated at each step. We establish a normalized generative process and decompose its predictive gains into contributions from dependence retention and marginal prediction. RGCR improves structure fidelity and recovery performance on the RNA inverse-folding benchmark. Code and checkpoints are available at https://anonymous.4open.science/r/rgcr-5241/.

1 INTRODUCTION

RNA inverse folding seeks nucleotide sequences compatible with a target 3D structure, posing a geometry-conditioned generation problem over discrete variables (Joshi et al., 2025). Base pairing and stacking couple nucleotide choices across the folded molecule, including positions distant along the sequence (Leontis & Westhof, 2001; Lu et al., 2015). Individually plausible nucleotide choices can therefore form incompatible combinations, motivating coordinated sequence generation.

Geometric autoregressive models and structure-conditioned diffusion have advanced RNA inverse folding (Joshi et al., 2025; Huang et al., 2024; Si et al., 2026). Autoregressive decoding captures dependencies through sequential conditioning; independent masked sampling can discard within-step dependencies despite shared contextual representations (Liu et al., 2025). Joint denoising and dependency-aware reveal selection address this limitation in discrete generation (Hersche et al., 2026; Ringel et al., 2026). For few-step RNA generation, the key decision is which geometry-derived groups benefit from joint sampling within each reveal set.

Geometric dependence alone does not ensure predictive benefit: learned joint predictors also incur error. We therefore use their gains over singletons as predictive utility to guide geometric messages; revised predictions then update utility.

RGCR builds on MDLM's absorbing-mask representation (Sahoo et al., 2024), coupling recurrent geometric refinement with joint generation (Figure 1). RNA's four-letter alphabet permits normalized pair and triple distributions alongside singleton predictions at each generation step. Predicted joint-over-singleton log-score gains gate geometric messages that recurrently refine predictions and utilities within a fixed masked context. Final scores select disjoint groups within the reveal set; selected groups are sampled jointly, with singleton predictions for remaining positions.

Contributions. We introduce utility-guided recurrent geometric refinement and joint sampling for RNA inverse folding. Building on analyses of dependence loss (Liu et al., 2025; Hersche et al., 2026), we establish normalized generation and exact schedule-conditioned likelihood, and decompose predictive gains into dependence retention and marginal prediction. Experiments demonstrate improved structural self-consistency, recovery, and foldable diversity, with controlled interventions supporting joint sampling, learned selection, and recurrent feedback.

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.

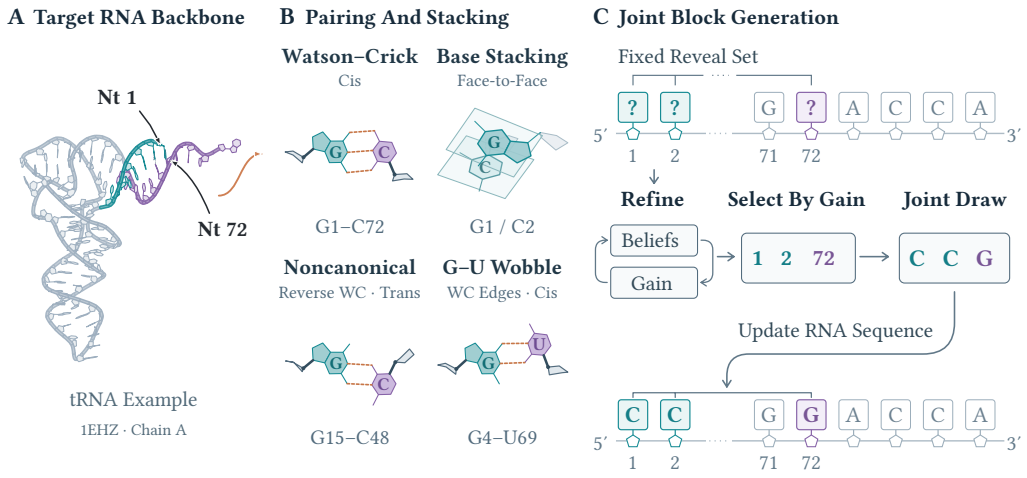

Figure 1: Recurrent Geometric Coupling Refinement (RGCR). (A) Fixed target backbone. (B) Motivation for joint nucleotide prediction. (C) Recurrent refinement estimates predictive gains and selects disjoint groups for joint sampling within the reveal set.

2 Related Work

2.1 Geometric Learning for RNA Inverse Folding

Geometric vector perceptrons connect RNA backbone geometry to nucleotide prediction (Jing et al., 2021). gRNAde uses multi-state graphs; GALS-Fold combines local equivariance with global attention (Joshi et al., 2025; Wen et al., 2026). RDesign combines contrastive learning with secondary-structure-guided base-pair correction; HyperRNA uses higher-order hypergraph encoding followed by autoregressive decoding (Tan et al., 2024; Yang & Fan, 2025). RiboDiffusion uses self-conditioned continuous denoising, RIdiffusion hyperbolic discrete diffusion, and SOLD structural-reward optimization of latent diffusion (Huang et al., 2024; Hou et al., 2025; Si et al., 2026). RNAFlow couples inverse folding and structure prediction for protein-conditioned co-design (Nori & Jin, 2024). RGCR learns which geometry-derived joint predictions to execute.

2.2 Dependency-Aware Discrete Generation and Adaptive Factorization

MDLM provides our masked-diffusion foundation (Sahoo et al., 2024); related formulations include D3PM, SEDD, and order-agnostic ARDM (Austin et al., 2021; Lou et al., 2024; Hoogeboom et al., 2022). Mask-Predict and MaskGIT develop iterative masked decoding (Ghazvininejad et al., 2019; Chang et al., 2022). DCD and CoDiLA formalize dependence lost through factorized denoising, using copulas and local autoregression for joint prediction; EDLM uses residual energies and CoDD probabilistic circuits (Liu et al., 2025; Hersche et al., 2026; Xu et al., 2025; Li et al., 2026a). APD uses joint predictions for adaptive parallel decoding (Israel et al., 2025). Block Diffusion combines autoregression across blocks with diffusion within blocks, and Set Diffusion permits flexible token sets and orderings (Arriola et al., 2025; Arriola & Kuleshov, 2026). Swordsman and GeoBlock adapt contiguous boundaries using entropy shifts and attention-derived dependencies (Zhang et al., 2026; Wan et al., 2026); DEMASK learns conditional influences to select reveal sets with low estimated dependence for independent parallel sampling (Ringel et al., 2026). RGCR selects geometric groups by predicted joint-over-singleton gain, incorporating predictive error and dependence.

2.3 Recurrent Refinement and Learned Structured Inference

Universal Transformers and Huginn share transformations across depth, while TRM alternates latent and answer updates (Dehghani et al., 2019; Geiping et al., 2025; Jolicoeur-Martineau, 2025). Learned structured inference spans repeated message updates in CRF-RNN, structure2vec, neural belief propagation, and Recurrent Relational Networks (Zheng et al., 2015; Dai et al., 2016; Kuck et al., 2020; Garcia Satorras & Welling, 2021; Palm et al., 2018), iterative posterior refine-

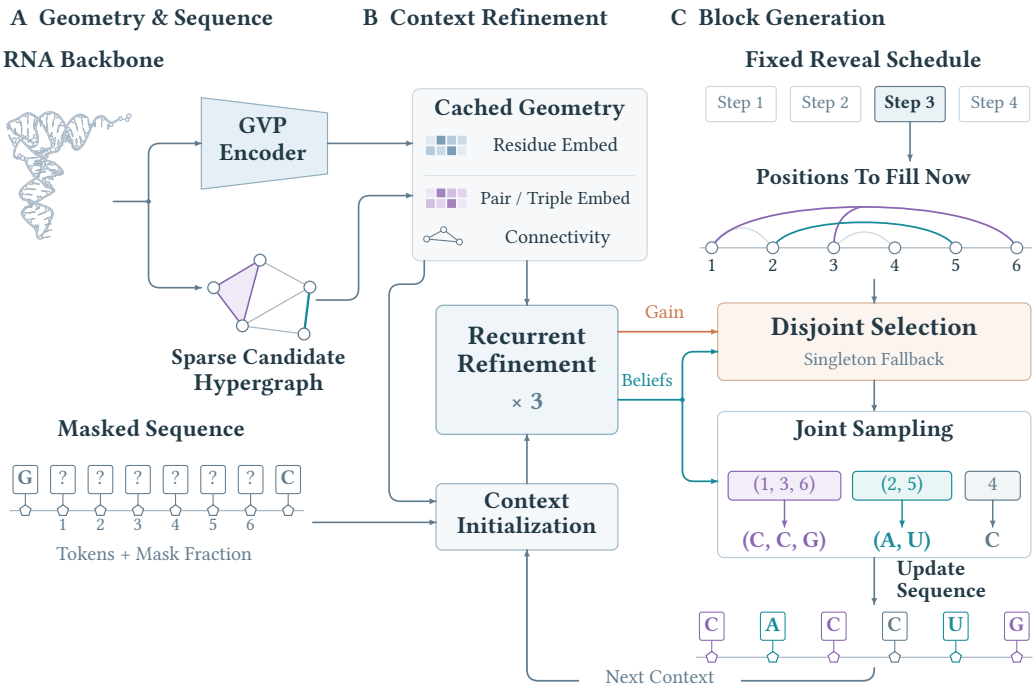

Figure 2: RGCR architecture. (A) Geometry defines cached features and candidate groups. (B) Recurrent updates refine local predictions and predictive gains in the masked context. (C) Temperature-adjusted gains select disjoint groups for joint sampling at each reveal step.

ment (Marino et al., 2018), and latent interaction graphs for dynamical prediction in NRI and dNRI (Kipf et al., 2018; Graber & Schwing, 2020). RAFT and CoTracker3 recurrently refine flow and tracks (Teed & Deng, 2020; Karaev et al., 2025).

AlphaFold2, ESMFold, and RhoFold+ recycle representations and features derived from predicted coordinates (Jumper et al., 2021; Lin et al., 2023; Shen et al., 2024), whereas AlphaFold3 recycles single/pair representations before coordinate diffusion (Abramson et al., 2024). RFdiffusion self-conditions RoseTTAFold-based denoising on preceding clean-structure predictions; BioEmu conditions diffusion on cached AlphaFold2 representations to sample protein equilibrium ensembles (Watson et al., 2023; Lewis et al., 2025). RefineGNN feeds antibody structures back into sequence prediction (Jin et al., 2022). RGCR feeds sequence beliefs and predictive utilities into refinement, then uses the utilities to select joint sampling blocks.

3 Methodology

Problem and scope. Given a fixed RNA backbone X of length L, RGCR generates $S \in \mathcal{A}^L$, where $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$, through recurrent geometric refinement and utility-guided joint sampling (Figure 2).

3.1 Geometric candidates and local predictive distributions

Geometry representation. Following gRNAde (Joshi et al., 2025), each residue is represented by P, C4′, and a common glycosidic-nitrogen coordinate role for N1/N9. A GVP encoder (Jing et al., 2021) produces cached invariant residue features g_i. Relative-geometry features are also invariant to global rotations and translations; their construction and encoder inputs are specified in Appendix C.

Sparse candidate hypergraph. The candidate hypergraph $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \mathcal{H}_3$ contains all singleton (one-position) blocks and geometry-derived pairs and triples. The initial pair graph combines covalent connectivity with a symmetrized C4′ k-nearest-neighbor graph. Each anchor and two of its

neighbors in this initial graph form a triple candidate, with at most k_3 retained per anchor. All three pairwise faces of each retained triple are added to $\mathcal{H}_2$, making its pair alternatives available. Candidates and probability-tensor axes follow increasing chain-position order. The construction retains $O(L(k + k_3))$ candidates; ranking and validity rules are specified in Appendix C.

For each pair $i < j$ in $\mathcal{H}_2$, let ϕ_{ij} denote its invariant relative-geometry features. Shared order-specific MLPs define pair and triple geometric features as

$$g_{\{i,j\}} = f_{g,2}(\phi_{ij}), \qquad g_{\{i,j,k\}} = f_{g,3}([\phi_{ij}, \phi_{ik}, \phi_{jk}]), \quad i < j < k. \tag{1}$$

Masked contexts. Let $\omega = (U_1, \ldots, U_T)$ be an external randomized reveal schedule whose nonempty sets partition $\{1, \ldots, L\}$. At step t, define $O_t = \bigcup_{u<t} U_u$, $M_t = \{1, \ldots, L\} \setminus O_t$, and $c_t = (X, \omega, t, S_{O_t})$. The network observes the structure, current masked sequence, and unresolved fraction; the schedule supplies $U_t \subseteq M_t$, and likelihoods are conditioned on ω. Candidates contained in M_t support belief refinement, whereas only those contained in U_t can be jointly sampled at step t.

Explicit local heads. Initialize $h_i^{(1)}$ with a geometric context pass over revealed and masked residues in the pair graph $\mathcal{H}_2$, using g_i, current token embeddings, and the unresolved fraction. At refinement stage r, shared heads predict

$$q_i^{(r)} = \mathrm{softmax}(f_1(h_i^{(r)})), \qquad Q_e^{(r)} = \mathrm{softmax}\left(f_{|e|}(g_e, [h_i^{(r)}]_{i\in e})\right), \quad |e| \in \{2, 3\}. \tag{2}$$

Singleton, pair, and triple heads have 4, 16, and 64 entries; their parameters are shared across candidates of the same order and refinement stages. Joint prediction and supervision apply only to candidates contained in M_t.

3.2 Predictive coupling utility and recurrent refinement

A common singleton reference. For a fully masked candidate e and its native training label S_e^*, define the stage-specific log-score gain

$$u_e^{(r)}(S_e^*; c_t) = \log Q_e^{(r)}(S_e^* \mid c_t) - \sum_{i\in e} \log q_i^{(r)}(S_i^* \mid c_t). \tag{3}$$

The learned coupling $C_e^{(r)}$ approximates the expected gain $G_e^{(r)}(c_t) = \mathbb{E}_{S_e\sim P(\cdot|c_t)}[u_e^{(r)}(S_e; c_t)]$ under the data distribution P. Gains are measured in nats per block, with zero singleton gain.

Belief features. The coupling predictor uses model total correlation $D_e^{(r)}$ and summed Jensen–Shannon disagreement $A_e^{(r)}$ between joint-head marginals and singleton predictions, defined in Appendix C.

The recurrent coupling loop. Starting from $C_e^{(0)} = 0$, each stage first computes the heads in (2) and then updates the coupling state (Figure 3):

$$z_e^{(r)} = \left[g_e, [h_i^{(r)}]_{i\in e}, \mathrm{Emb}(Q_e^{(r)}), D_e^{(r)}, A_e^{(r)}, C_e^{(r-1)}\right], \tag{4}$$

$$C_e^{(r)} = C_e^{(r-1)} + f_{C,|e|}(z_e^{(r)}). \tag{5}$$

Here Emb projects the canonically ordered probability table through an order-specific map. For $r < R$, coupling-gated messages refine the hidden states:

$$m_{e\to i}^{(r)} = \sigma(a_{|e|} C_e^{(r)} + b_{|e|}) f_{M,|e|}(h_i^{(r)}, z_e^{(r)}), \tag{6}$$

$$h_i^{(r+1)} = \mathrm{LN}\left(h_i^{(r)} + f_U\left(h_i^{(r)}, g_i, \frac{\sum_{e\in\mathcal{N}_i(M_t)} m_{e\to i}^{(r)}}{\max(1, |\mathcal{N}_i(M_t)|)}\right)\right). \tag{7}$$

Here $\mathcal{N}_i(M) = \{e \in \mathcal{H}_2 \cup \mathcal{H}_3 : i \in e,\ e \subseteq M\}$. Gate parameters and update maps are shared across $R = 3$ stages; the final $C^{(R)}$ estimates the utility of the same-stage $Q^{(R)}$. The masked context remains fixed during refinement, and coupling states reset at each generation step.

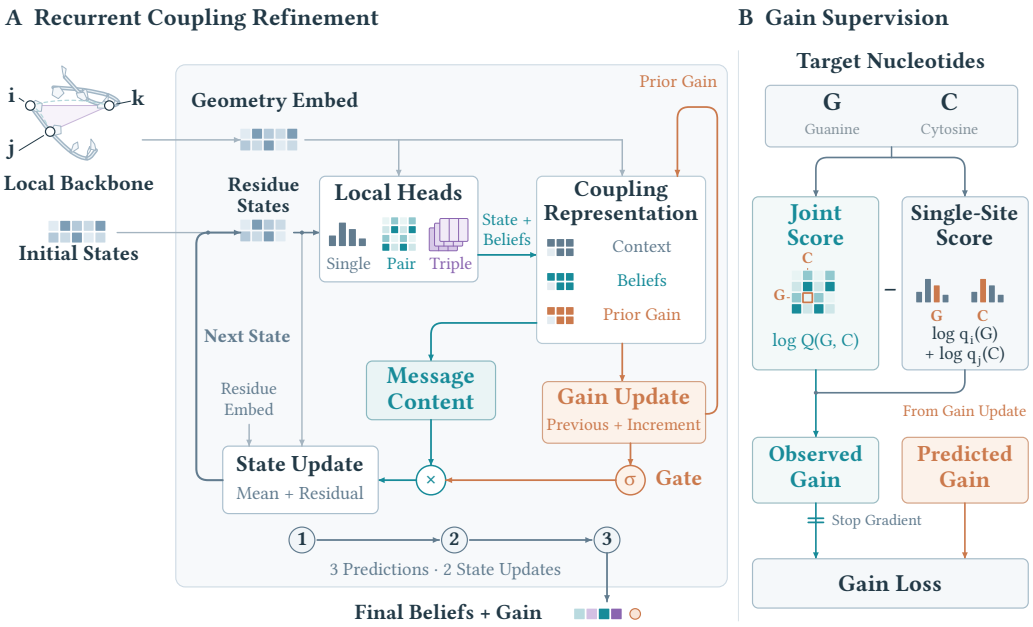

Figure 3: Recurrent refinement and gain supervision. (A) Predictions update gains, which gate geometric messages for the next stage within a fixed context. (B) Native joint-minus-singleton log scores provide stop-gradient targets for gain regression.

3.3 Training Objective

For each training sequence, draw a reveal schedule and sample $t \sim \mathrm{Unif}\{1, \ldots, T\}$, retaining the native prefix $S^*_{O_t}$ and masking M_t; $t = 1$ gives the all-masked context. Using the absorbing-mask representation of MDLM (Sahoo et al., 2024), we optimize a supervised composite denoising objective. For $j \in \{2, 3\}$, let $\mathcal{E}_j(c_t) = \{e \in \mathcal{H}_j : e \subseteq M_t\}$ contain all eligible candidates, independently of selection. With $w_r = 1/R$ and $\mathrm{avg}_{e \in \varnothing}(\cdot) = 0$, the objective is

$$\mathcal{L}_{\text{sing}} = \sum_{r=1}^{R} w_r \frac{-1}{|M_t|} \sum_{i \in M_t} \log q_i^{(r)}(S_i^* \mid c_t), \tag{8}$$

$$\mathcal{L}_{\text{joint}} = \sum_{r=1}^{R} w_r \sum_{j=2}^{3} \mathrm{avg}_{e \in \mathcal{E}_j(c_t)} \left[-\frac{1}{j} \log Q_e^{(r)}(S_e^* \mid c_t) \right], \tag{9}$$

$$\mathcal{L}_{\text{coupling}} = \sum_{r=1}^{R} w_r \sum_{j=2}^{3} \mathrm{avg}_{e \in \mathcal{E}_j(c_t)} \left(C_e^{(r)} - \mathrm{sg}[u_e^{(r)}(S_e^*; c_t)] \right)^2, \tag{10}$$

$$\mathcal{L} = \mathbb{E}_{(X, S^*), t, \omega}[\mathcal{L}_{\text{sing}} + \lambda_J \mathcal{L}_{\text{joint}} + \lambda_C \mathcal{L}_{\text{coupling}}]. \tag{11}$$

Order-wise averaging and $1/j$ balance training losses; selection uses block gains. The stop-gradient operator sg detaches only the realized-gain targets; gradients through coupling inputs and recurrent messages still update shared representations and predictive heads. Appendix A.4 gives the regression interpretation.

3.4 Executable Adaptive Block Generation

Temperature-adjusted selection. Training and likelihood diagnostics use $\tau = 1$. For sampling at $\tau > 0$, write $Q_B = Q_B^{(R)}$ and $Q_{\{i\}} = q_i^{(R)}$, and define

$$Z_B(\tau) = \sum_s Q_B(s \mid c_t)^{1/\tau}, \qquad Q_{B,\tau}(s \mid c_t) = \frac{Q_B(s \mid c_t)^{1/\tau}}{Z_B(\tau)}. \tag{12}$$

With singleton normalizer $Z_i(\tau)$, the adjusted score is

$$C^{(R)}_{B,\tau} = \frac{C^{(R)}_B}{\tau} - \log Z_B(\tau) + \sum_{i \in B} \log Z_i(\tau), \qquad C^{(R)}_{\{i\},\tau} = 0. \tag{13}$$

Only final selection uses this correction; recurrent messages retain untempered $C^{(r)}$ (Appendix A.3).

Partition and sampling kernel. Let $\mathfrak{P}(U_t)$ contain partitions of U_t into singletons and available pair/triple candidates, with selection objective

$$\max_{\pi \in \mathfrak{P}(U_t)} \sum_{B \in \pi} C^{(R)}_{B,\tau}. \tag{14}$$

A deterministic greedy approximation processes positive-scoring candidates in decreasing adjusted-score order, accepts nonoverlapping blocks, and fills uncovered positions with singletons. The resulting partition $\pi_{t,\tau}(c_t)$ is determined before any identities in U_t are sampled; tie-breaking is specified in Appendix C. The step kernel is

$$K_{\theta,\tau}(S_{U_t} \mid c_t) = \prod_{B \in \pi_{t,\tau}(c_t)} Q_{B,\tau}(S_B \mid c_t). \tag{15}$$

Sample one categorical assignment independently per block, commit U_t simultaneously, and repeat until every position has been sampled once. The generator is normalized for every $\tau > 0$, even when overlapping candidate heads disagree (Section 4). Algorithm 1 summarizes generation; Appendix C gives its computational cost.

4 Theoretical Interpretation

Fix X and $P(S \mid X)$, with ω independent of S given X. We use the contexts and deterministic pre-sampling partitions defined above, suppress R, and set $\tau = 1$, writing $\pi_t = \pi_{t,1}$. The following identities connect normalized generation to conditional predictive risk.

Proposition 1 (Normalized generation and exact likelihood). The executed process defines a normalized conditional distribution

$$p^{\omega}_{\theta}(S \mid X) = \prod_{t=1}^{T} q_{\theta,t}(S_{U_t} \mid c_t), \qquad q_{\theta,t}(S_{U_t} \mid c_t) = \prod_{B \in \pi_t(c_t)} Q_B(S_B \mid c_t). \tag{16}$$

Replaying the deployed selector on native prefixes gives the exact sequence log-likelihood conditional on ω. The expected schedule-conditioned NLL upper-bounds the schedule-marginalized NLL (Appendix A.1).

Proposition 2 (Dependence, prediction error, and block utility). At a fixed context c and reveal set U, write $P_J = P(S_J \mid c)$ and $\mathrm{TC}(P_J) = \sum_{i \in J} H(P_i) - H(P_J)$ for $J \subseteq U$. For an admissible partition π, let $Q_\pi = \prod_{B \in \pi} Q_B(\cdot \mid c)$. When log scores are finite on the data support,

$$\mathrm{KL}(P_U \| Q_\pi) = \underbrace{\mathrm{TC}(P_U) - \sum_{B \in \pi} \mathrm{TC}(P_B)}_{\text{dependence discarded across blocks}} + \underbrace{\sum_{B \in \pi} \mathrm{KL}(P_B \| Q_B)}_{\text{local predictive error}}. \tag{17}$$

Relative to the common singleton reference $q_0 = \prod_{i \in U} q_i$, the expected gains defined in Section 3.2 satisfy

$$G_B(c) = \mathrm{TC}(P_B) + \sum_{i \in B} \mathrm{KL}(P_i \| q_i) - \mathrm{KL}(P_B \| Q_B), \tag{18}$$

$$\mathrm{KL}(P_U \| q_0) - \mathrm{KL}(P_U \| Q_\pi) = \sum_{B \in \pi} G_B(c). \tag{19}$$

Thus maximizing true expected gain minimizes conditional KL, even with misspecified heads. Learned scores and greedy selection approximate gain estimation and partition optimization, respectively; Appendix A.6 bounds their combined effect on true gain. With exact joint and singleton predictors ($Q_B = P_B$ and $q_i = P_i$), $G_B = \mathrm{TC}(P_B)$.

Dependence retention and marginal prediction. With $Q_{B,i} = \mathrm{Marg}_i\, Q_B$, the pointwise gain decomposes as

$$u_B = \delta_B^{\mathrm{dep}} + \delta_B^{\mathrm{marg}},$$
$$\delta_B^{\mathrm{dep}} = \log Q_B - \sum_{i \in B} \log Q_{B,i}, \qquad \delta_B^{\mathrm{marg}} = \sum_{i \in B} \log Q_{B,i} - \sum_{i \in B} \log q_i. \tag{20}$$

The two signed terms separate dependence retention at fixed model marginals from changes in marginal prediction. Both propositions extend to normalized tempered heads; proofs appear in Appendix A.

5 Experiments

Setup. We evaluate on DAS55, following gRNAde (Joshi et al., 2025). Each method generates 16 unranked designs per target under 3 generation seeds; scores are averaged with equal target weights. Controlled comparisons share reveal schedules, and operating temperatures are chosen on validation data. Appendices B and C give the data and implementation details.

Metrics. We report structural self-consistency (scTM) averaged across RhoFold+, AlphaFold3, and DRfold2 using US-align (Shen et al., 2024; Abramson et al., 2024; Li et al., 2026b; Zhang et al., 2022). We also report sequence recovery, foldable diversity (FD), oracle disagreement, and base-pair recovery (BP F1). FD weights sequence diversity by structural validity (Ektefaie et al., 2024). Native-prefix NLL is schedule-conditioned at $\tau = 1$ and reported in nats per nucleotide.

5.1 RQ1: How well does RGCR perform on RNA inverse folding?

Table 1: RNA inverse-folding performance. scTM averages three forward predictors; Disagr. is their per-design sample standard deviation. FD uses RhoFold+ scTM ≥ 0.45.

Method	scTM ↑	Rec. (%) ↑	FD ↑	Disagr. ↓
RDesign (Tan et al., 2024)	0.310	41.884	0.000	0.097
RIdiffusion (Hou et al., 2025)	0.412	47.389	0.012	**0.075**
gRNAde (Joshi et al., 2025)	0.437	56.473	0.026	0.119
GALS-Fold (Wen et al., 2026)	0.404	56.450	0.014	0.146
RGCR (ours)	**0.480**	**64.684**	**0.067**	0.109

As shown in Table 1, RGCR leads in scTM, recovery, and FD, improving over gRNAde by 9.8%, 8.21 percentage points, and $2.6\times$, respectively. Higher FD strengthens performance on the joint objective of sequence diversity and structural validity.

5.2 RQ2: Does joint generation help beyond improved marginals?

Using one full RGCR checkpoint, we replace each selected tempered joint head $Q_{B,\tau}$ by the product of its marginals (Table 2). This preserves the blocks and one-site marginals at each fixed context. Rollout arms share reveal traces and recompute predictions from their own prefixes.

Joint sampling improves scTM and BP F1 while lowering NLL (Table 2A), linking dependence retention to better native-sequence prediction and generated structural quality.

Let N_q, N_{marg}, and N_{joint} denote this checkpoint's singleton-head, product-of-marginals, and joint NLLs. Equation (20) gives $N_q - N_{\mathrm{joint}} = (N_q - N_{\mathrm{marg}}) + (N_{\mathrm{marg}} - N_{\mathrm{joint}})$. Dependence retention contributes 0.073 of the 0.074 nats/nt gain (Table 2B), identifying preserved nucleotide combinations as the principal source of improvement.

Table 2: Joint generation at matched one-site marginals. (A) Sampling kernels and the singleton-head NLL reference. (B) Signed native NLL gains (nats/nt), computed before rounding.

Sampling kernel / reference	NLL ↓	scTM ↑	BP F1 ↑
A. Sampling kernels and same-model reference			
Joint sampling	**0.597**	**0.480**	**0.390**
Product-of-marginals sampling	0.670	0.448	0.377
Singleton-head reference	0.671	–	–
B. Signed native likelihood gains (nats/nt)			
Total gain, $N_q - N_{\text{joint}}$			+0.074
Marginal prediction, $N_q - N_{\text{marg}}$			+0.001
Dependence retention, $N_{\text{marg}} - N_{\text{joint}}$			+0.073

5.3 RQ3: How do learned utilities and recurrent feedback contribute?

Table 3A compares ablations with matched geometric encoders, inputs, hidden widths, and training budgets; the hidden-only policies share one checkpoint. Panels B and C isolate final selection and cross first- and final-stage heads with selection scores.

Table 3: Components of RGCR. (A) Matched ablations. (B) Final-selector replacement in full RGCR; pair/triple coverage is measured at native contexts with $\tau = 1$. (C) Crossed recurrence at fixed native contexts. All NLLs are in nats/nt; rollouts share reveal traces.

Configuration	scTM ↑	NLL ↓
A. Component ablations		
Hidden-only: singleton ($R = 3$)	0.426	0.687
Hidden-only: geometry-greedy pairs ($R = 3$)	0.426	0.684
Hidden-only: learned gain ($R = 3$)	0.431	0.624
One-shot RGCR ($R = 1$)	0.438	0.677
Pair-only RGCR ($R = 3$)	0.463	0.646
Full RGCR ($R = 3$)	**0.480**	**0.597**

Final selector	NLL ↓	scTM ↑	BP F1 ↑	Pair/triple (%)
B. Final-selector intervention				
Learned gain	**0.597**	**0.480**	**0.390**	79.313/6.285
TC-greedy	0.608	0.456	0.383	77.548/15.815

C. Crossed recurrence: NLL (nats/nt) ↓		
Selection scores	First-stage heads $Q^{(1)}$	Final-stage heads $Q^{(R)}$
First-stage $C^{(1)}$	0.612	0.599
Final-stage $C^{(R)}$	0.604	**0.597**
Changed partner sets, stages $1 \to R$ (%)		42.164

Selecting by predictive gain. With the recurrent model, greedy packing, and temperature fixed, learned gain improves scTM, BP F1, and NLL over TC-greedy (panel B). This supports selection by predictive utility, which accounts for dependence and predictive accuracy.

Recurrent belief–utility feedback. Full RGCR improves scTM over hidden-only recurrence with learned-gain execution (0.431) and one-shot RGCR (0.438), reaching 0.480 (panel A). Hidden-only recurrence retains three stages and auxiliary supervision, while removing belief and utility feedback from messages. The gain demonstrates the value of explicit belief–utility feedback.

The crossed evaluation attributes NLL improvements to both updated heads and updated selection, with the final-stage combination improving NLL from 0.612 to 0.597 (panel C). Recurrence thus refines both block predictions and execution choices.

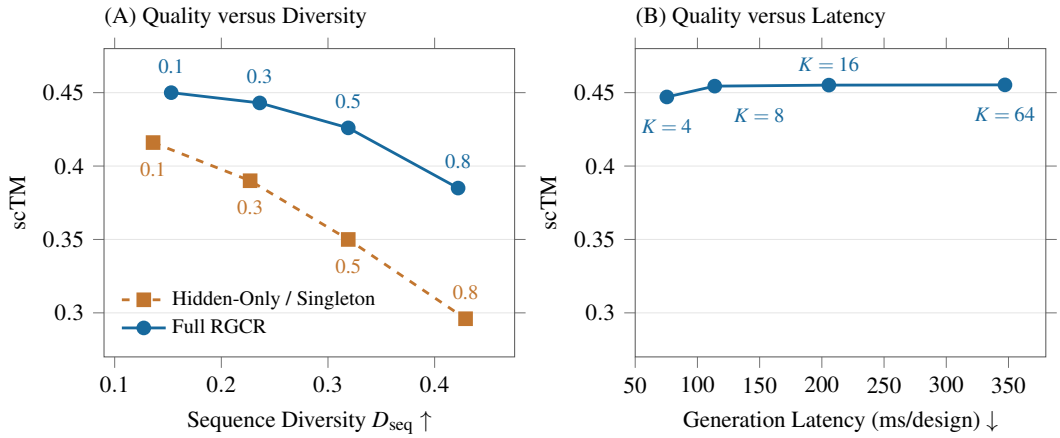

Figure 4: Generation trade-offs evaluated with RhoFold+. (A) scTM versus sequence diversity; labels indicate temperature. (B) scTM versus generation latency for full RGCR at $\tau = 0.1$; labels indicate reveal-step cap K. Latency excludes forward structure prediction.

The triple pathway. The triple pathway improves scTM and NLL over pair-only RGCR (panel A). The control retains all pair candidates, including triple-face closure, while removing triple heads, losses, and messages. The gains support triple modeling across refinement and joint execution.

5.4 RQ4: What are the quality–diversity and quality–compute trade-offs?

Against hidden-only singleton generation, RGCR improves structural quality throughout the temperature sweep and increases sequence diversity at low temperatures (Figure 4A).

A reveal-step cap of 4 already achieves most of the structural quality attained at larger budgets; quality then plateaus as latency rises (Figure 4B).

6 Discussion and Limitations

Predictive utility connects geometric modeling to generation decisions by assessing the benefit of executing local joint predictions. RGCR realizes this principle through recurrent feedback and disjoint joint sampling. The matched-marginal intervention identifies dependence retention as the main source of its same-model likelihood gain, linking joint nucleotide compatibility to improved structural self-consistency. RNA's small alphabet makes explicit local joint distributions practical.

Limitations. RGCR targets sequence design for fixed backbone inputs and represents joint choices through pair and triple blocks. Shared geometric and sequence context informs all block predictions, while the executed distribution factorizes across blocks within each reveal step. Structural evaluation uses computational self-consistency under forward predictors; physical folding and biological function still require wet-lab validation.

AI USE STATEMENT

In this work, we used generative AI tools to refine hypotheses and the theoretical framework, to formulate mathematical claims and assist in writing their proofs, to provide feedback on the research methodology and the experimental protocol, to implement methods and execute experiments, and to clean and reformat datasets. We have not used generative AI tools to generate the core research ideas, the original experimental design, or the evaluation design; the remaining tasks requiring disclosure, including generating synthetic datasets, translation, and qualitative or thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of this paper and to search and summarize related literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available RNA structures and involves no human subjects, personal data, or wet-lab experiments. RNA design is dual-use in principle; our method is conditioned on an experimentally determined backbone, predicts nucleotide identities only, and is not experimentally validated, and any experimental use of designed sequences should follow standard synthesis-screening and institutional biosafety practice.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix C specify the model, geometric features, candidate construction, training objective, optimization, and checkpoint-selection procedure. Appendix A states the assumptions and gives complete proofs of the claims in Section 4. Appendix B and Table 4 document dataset curation, splits, position filtering, forward-predictor configurations, metric definitions, and aggregation. Code, dataset manifests, stored reveal traces, and evaluation scripts are available at the anonymous repository linked in the abstract.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Marianne Arriola and Volodymyr Kuleshov. Set Diffusion: Interpolating Token Orderings Between Autoregression and Diffusion for Fast and Flexible Decoding. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2607.01775.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked Generative Image Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://masked-generative-image-transformer.github.io/.

Hanjun Dai, Bo Dai, and Le Song. Discriminative Embeddings of Latent Variable Models for Structured Data. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2702–2711. PMLR, 2016. URL https://proceedings.mlr.press/v48/daib16.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Yasha Ektefaie, Olivia Viessmann, Siddharth Narayanan, Drew Dresser, J. Mark Kim, and Armen Mkrtchyan. Reinforcement learning on structure-conditioned categorical diffusion for protein inverse folding. *arXiv preprint arXiv:2410.17173*, 2024. doi: 10.48550/arXiv.2410.17173. URL https://arxiv.org/abs/2410.17173.

Víctor Garcia Satorras and Max Welling. Neural Enhanced Belief Propagation on Factor Graphs. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 685–693. PMLR, 2021. URL https://proceedings.mlr.press/v130/garcia-satorras21a.html.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1380. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3b01972cf31e6fa0fe29e4b8b5c2a0a1-Abstract-Conference.html.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

Colin Graber and Alexander G. Schwing. Dynamic Neural Relational Inference. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8513–8522, 2020. doi: 10.1109/cvpr42600.2020.00854. URL https://doi.org/10.1109/cvpr42600.2020.00854.

Michael Hersche, Nicolas Menet, Ronan Tanios, and Abbas Rahimi. Locally Coherent Parallel Decoding in Diffusion Language Models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.20216.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive Diffusion Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.02037.

Dongyue Hou, Shuai Zhang, Mengyao Ma, Hanbo Lin, Zheng Wan, Hui Zhao, Ruian Zhou, Xiao He, Xian Wei, Dianwen Ju, and Xian Zeng. A Hyperbolic Discrete Diffusion 3D RNA Inverse Folding Model for Functional RNA Design. *Journal of Chemical Information and Modeling*, 65 (13):6568–6584, 2025. doi: 10.1021/acs.jcim.5c00527. URL https://doi.org/10.1021/acs.jcim.5c00527.

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. doi: 10.1093/bioinformatics/btae259. URL https://doi.org/10.1093/bioinformatics/btae259.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating Diffusion LLMs via Adaptive Parallel Decoding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 58825–58843, 2025. doi: 10.52202/085713-1763. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4c7d31b2da05a17f15101db5b37a1e24-Abstract-Conference.html.

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative Refinement Graph Neural Network for Antibody Sequence-Structure Co-design. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.04624.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron O. Dror. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

Alexia Jolicoeur-Martineau. Less is More: Recursive Reasoning with Tiny Networks, 2025. URL https://arxiv.org/abs/2510.04871.

Chaitanya K. Joshi, Arian R. Jamasb, Ramon Viñas, Charles Harris, Simon V. Mathis, Alex Morehead, Rishabh Anand, and Pietro Liò. gRNAde: Geometric Deep Learning for 3D RNA inverse design. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b96f01343ff10150e6719eb163e1536-Abstract-Conference.html.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and Better Point Tracking by Pseudo-Labeling Real Videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10, Honolulu, HI, USA, 2025. doi: 10.1109/ICCV51701.2025.00568. URL https://doi.org/10.1109/ICCV51701.2025.00568.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural Relational Inference for Interacting Systems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2688–2697. PMLR, 2018. URL https://proceedings.mlr.press/v80/kipf18a.html.

Jonathan Kuck, Shuvam Chakraborty, Hao Tang, Rachel Luo, Jiaming Song, Ashish Sabharwal, and Stefano Ermon. Belief Propagation Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 667–678, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/07217414eb3fbe24d4e5b6cafb91ca18-Abstract.html.

Neocles B. Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512, 2001. doi: 10.1017/S1355838201002515. URL https://doi.org/10.1017/S1355838201002515.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Adam E. Foster, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Roberto Sordillo, Ryota Tomioka, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 389(6761):eadv9817, 2025. doi: 10.1126/science.adv9817. URL https://doi.org/10.1126/science.adv9817.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the Factorization Barrier in Diffusion Language Models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2603.00045.

Yang Li, Chenjie Feng, Xi Zhang, Sho Tsukiyama, Duanyu Feng, and Yang Zhang. DRfold2 is a deep learning-based tool that enables efficient and accurate RNA structure prediction. *PLOS Biology*, 24(2):e3003659, 2026b. doi: 10.1371/journal.pbio.3003659. URL https://doi.org/10.1371/journal.pbio.3003659.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://doi.org/10.1126/science.ade2574.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete Copula Diffusion. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/dd1fef536655685898a6602bfbf16857-Abstract-Conference.html.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

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

Joe Marino, Yisong Yue, and Stephan Mandt. Iterative Amortized Inference. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3403–3412. PMLR, 2018. URL https://proceedings.mlr.press/v80/marino18a.html.

Divya Nori and Wengong Jin. RNAFlow: RNA Structure & Sequence Design via Inverse Folding-Based Flow Matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38395–38408, 2024. URL https://proceedings.mlr.press/v235/nori24a.html.

Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent Relational Networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/b9f94c77652c9a76fc8a442748cd54bd-Abstract.html.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-Guided Parallel Decoding in Discrete Diffusion Language Models. *arXiv preprint arXiv:2604.02560*, 2026. doi: 10.48550/arXiv.2604.02560. URL https://arxiv.org/abs/2604.02560.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, Yixuan Wang, Liang Hong, Jin Xiao, Liangzhen Zheng, Tejas Krishnamoorthi, Irwin King, Sheng Wang, Peng Yin, James J. Collins, and Yu Li. Accurate RNA 3D structure prediction using a language model-based deep learning approach. *Nature Methods*, 21(12):2287–2298, 2024. doi: 10.1038/s41592-024-02487-0. URL https://doi.org/10.1038/s41592-024-02487-0.

Qi Si, Xuyang Liu, Penglei Wang, Xin Guo, Yuan Qi, and Yuan Cheng. Structure-based RNA Design by Step-wise Optimization of Latent Diffusion Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1006–1014, 2026. doi: 10.1609/aaai.v40i2.37070. URL https://ojs.aaai.org/index.php/AAAI/article/view/37070.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ad9a1a6ffac3dd72cc1df96019eca01-Abstract-Conference.html.

Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision – ECCV 2020*, pp. 402–419, 2020. doi: 10.1007/978-3-030-58536-5_24. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/3526_ECCV_2020_paper.php.

Lipeng Wan, Junjie Ma, Jianhui Gu, Zeyang Liu, Xuyang Lu, and Xuguang Lan. GeoBlock: Inferring Block Granularity from Dependency Geometry in Diffusion Language Models. *arXiv preprint arXiv:2603.26675*, 2026. doi: 10.48550/arXiv.2603.26675. URL https://arxiv.org/abs/2603.26675.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

Xiangyu Wen, Yujing Bian, Hengrui Gu, and Kaixiong Zhou. GALS-Fold: Geometry-Aware Long-Short RNA Inverse Folding with Linear Scaling. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 12372–12383, 2026. doi: 10.1145/3770855.3818980. URL https://doi.org/10.1145/3770855.3818980.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-Based Diffusion Language Models for Text Generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/535baca36883a4e0450423b26b150a48-Abstract-Conference.html.

Guang Yang and Lei Fan. Harnessing Hypergraphs in Geometric Deep Learning for 3D RNA Inverse Folding. *arXiv preprint arXiv:2512.03592*, 2025. doi: 10.48550/arXiv.2512.03592. URL https://arxiv.org/abs/2512.03592.

Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature Methods*, 19(9): 1109–1115, 2022. doi: 10.1038/s41592-022-01585-1. URL https://doi.org/10.1038/s41592-022-01585-1.

Yu Zhang, Xinchen Li, Jialei Zhou, Hongnan Ma, Zhongwei Wan, Yiwei Shi, Duoqian Miao, Qi Zhang, and Longbing Cao. Swordsman: Entropy-Driven Adaptive Block Partition for Efficient Diffusion Language Models. *arXiv preprint arXiv:2602.04399*, 2026. doi: 10.48550/arXiv.2602.04399. URL https://arxiv.org/abs/2602.04399.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1529–1537, 2015. doi: 10.1109/iccv.2015.179. URL https://doi.org/10.1109/iccv.2015.179.

APPENDIX

A THEORETICAL ANALYSIS AND PROOFS

A.1 EXACT LIKELIHOOD AND TRAJECTORY KL

Fix X, the external schedule ω, and the model parameters. At context $c_t = (X, \omega, t, S_{O_t})$, the selector has not observed any identity in U_t. Its output $\pi_t(c_t)$ is therefore fixed while summing over S_{U_t}. Since the selected blocks partition U_t,

$$\sum_{s_{U_t}} q_{\theta,t}(s_{U_t} \mid c_t) = \prod_{B \in \pi_t(c_t)} \sum_{s_B} Q_B(s_B \mid c_t) = 1. \tag{21}$$

Successively summing Equation (16) over $S_{U_T}, S_{U_{T-1}}, \ldots, S_{U_1}$ gives one. For a complete sequence S, its revealed values determine every selector output uniquely. No additional probability factor or marginalization is required for the deterministic route:

$$-\log p_\theta^\omega(S \mid X) = -\sum_{t=1}^{T} \sum_{B \in \pi_t(c_t(S))} \log Q_B(S_B \mid c_t(S)). \tag{22}$$

Evaluating this expression requires replaying the deployed selector on native prefixes.

The data distribution admits the corresponding chain-rule factorization,

$$P(S \mid X) = \prod_{t=1}^{T} P(S_{U_t} \mid X, S_{O_t}). \tag{23}$$

The external schedule is independent of S given X, so conditioning additionally on ω leaves this distribution unchanged. Taking the log ratio with Equation (16) and applying iterated expectation yields

$$\mathrm{KL}(P \| p_\theta^\omega) = \sum_{t=1}^{T} \mathbb{E}_{S \sim P} \log \frac{P(S_{U_t} \mid c_t)}{q_{\theta,t}(S_{U_t} \mid c_t)} \tag{24}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{S_{O_t} \sim P} \mathrm{KL}(P(S_{U_t} \mid c_t) \,\|\, q_{\theta,t}(\cdot \mid c_t)) \,. \tag{25}$$

The true distribution within U_t need not factor over the selected blocks. The expectations are over data prefixes; errors under generated prefixes can differ. The same argument applies to any deterministic prefix-dependent reveal policy that generates each site once; the external schedule additionally separates grouping from position selection.

Random schedules. For $\omega \sim \mu(\cdot \mid X)$ independent of S given X, the marginal generator and Jensen's inequality give

$$p_\theta(S \mid X) = \mathbb{E}_{\omega \sim \mu(\cdot|X)}[p_\theta^\omega(S \mid X)], \tag{26}$$

$$-\log p_\theta(S \mid X) \le \mathbb{E}_{\omega \sim \mu(\cdot|X)}[-\log p_\theta^\omega(S \mid X)]. \tag{27}$$

The expected schedule-conditioned NLL therefore upper-bounds the schedule-marginalized NLL. Each fixed-schedule likelihood is evaluated exactly; the expectation over schedules is estimated using the shared trace bank.

A.2 PARTITION ERROR AND COMMON-REFERENCE PREDICTIVE GAIN

Fix c, U, and a partition π. All data entropies and marginals below are conditional on c. Assume the predictive probabilities are positive on the data support, making the log-score differences finite. Softmax heads satisfy this condition. Marginalizing expectations of blockwise functions gives

$$\mathrm{KL}(P_U \| Q_\pi) = -H(P_U) - \sum_{B \in \pi} \mathbb{E}_{P_B} \log Q_B(S_B) \tag{28}$$

$$= \sum_{B \in \pi} H(P_B) - H(P_U) + \sum_{B \in \pi} \mathrm{KL}(P_B \| Q_B). \tag{29}$$

Because the blocks partition U,

$$\sum_{B\in\pi} H(P_B) - H(P_U) = \mathrm{TC}(P_U) - \sum_{B\in\pi} \mathrm{TC}(P_B). \tag{30}$$

This proves Equation (17). The entropy difference is also $\mathrm{KL}(P_U \| \prod_{B\in\pi} P_B)$, so it is nonnegative.

For an individual block, the realized and expected gains are

$$\begin{aligned} u_B(S_B; c) &= \log Q_B(S_B \mid c) - \sum_{i\in B} \log q_i(S_i \mid c), \\ G_B(c) &= \mathbb{E}_{S_B\sim P_B}[u_B(S_B; c)]. \end{aligned} \tag{31}$$

Expanding the expected log scores gives

$$G_B(c) = -H(P_B) - \mathrm{KL}(P_B\|Q_B) + \sum_{i\in B} \left[H(P_i) + \mathrm{KL}(P_i\|q_i)\right] \tag{32}$$

$$= \mathrm{TC}(P_B) + \sum_{i\in B} \mathrm{KL}(P_i\|q_i) - \mathrm{KL}(P_B\|Q_B). \tag{33}$$

Every site occurs in exactly one block, hence

$$\sum_{B\in\pi} G_B(c) = \mathbb{E}_{P_U} \log \frac{\prod_{B\in\pi} Q_B(S_B)}{\prod_{i\in U} q_i(S_i)} \tag{34}$$

$$= \mathrm{KL}(P_U\|q_0) - \mathrm{KL}(P_U\|Q_\pi). \tag{35}$$

This proves Equations (18) and (19). The reference q_0 is constant across partitions, so maximizing the true gain sum minimizes the corresponding KL. No correct-specification or marginal-consistency assumption is required.

Exact joint and singleton predictors. When $Q_B = P_B$ and $q_i = P_i$, the predictive-error terms vanish, giving $G_B = \mathrm{TC}(P_B)$; the optimal partition therefore maximizes true within-block total correlation. Zeros outside the data support are allowed under the usual zero-mass conventions.

Decomposing the observed gain. Adding and subtracting $\sum_i \log Q_{B,i}$ proves Equation (20) pointwise. Its conditional expectations are

$$\mathbb{E}_P \delta_B^{\mathrm{dep}} = \mathrm{TC}(P_B) + \sum_{i\in B} \mathrm{KL}(P_i\|Q_{B,i}) - \mathrm{KL}(P_B\|Q_B), \tag{36}$$

$$\mathbb{E}_P \delta_B^{\mathrm{marg}} = \sum_{i\in B} \left[\mathrm{KL}(P_i\|q_i) - \mathrm{KL}(P_i\|Q_{B,i})\right]. \tag{37}$$

The first measures the benefit of preserving dependence with the learned one-site marginals held fixed and generally differs from both true and model total correlation; the second measures the change in marginal prediction. Both can be negative, so we report signed differences rather than percentages.

From local gains to sequence likelihood. Retain the full model's recurrent computations and conditional singleton functions, but replace its executed step kernel by $q_{0,t} = \prod_{i\in U_t} q_i(\cdot \mid c_t)$. This defines a same-model reference generator p_0^ω. Subtracting the two trajectory KL decompositions gives

$$\mathrm{KL}(P\|p_0^\omega) - \mathrm{KL}(P\|p_\theta^\omega) = \mathbb{E}_{S\sim P} \sum_{t=1}^{T} \sum_{B\in\pi_t(c_t(S))} G_B(c_t(S)). \tag{38}$$

For each native trace, its realized counterpart is the sum of u_B over the selected blocks. Inserting the product of each selected head's own marginals as an intermediate kernel separates that NLL difference into the sums of δ_B^{dep} and δ_B^{marg}. All three scores use the same native contexts and existing predictions. The reference shares the full model's parameters, unlike singleton execution of the hidden-only checkpoint, and structural quality is evaluated separately.

A.3 TEMPERATURE CORRECTION

At a fixed context and $\tau > 0$, the normalized heads satisfy

$$\log Q_{B,\tau}(s) = \frac{\log Q_B(s)}{\tau} - \log Z_B(\tau), \qquad \log q_{i,\tau}(a) = \frac{\log q_i(a)}{\tau} - \log Z_i(\tau). \tag{39}$$

Subtracting the singleton log scores yields

$$u_{B,\tau}(S_B; c) = \frac{u_{B,1}(S_B; c)}{\tau} - \log Z_B(\tau) + \sum_{i \in B} \log Z_i(\tau). \tag{40}$$

The normalizers depend on context, not the unknown assignment. Taking conditional expectation under P_B gives the expected gain relative to the tempered singleton reference:

$$G_{B,\tau} = \frac{G_{B,1}}{\tau} - \log Z_B(\tau) + \sum_{i \in B} \log Z_i(\tau). \tag{41}$$

For $C_{B,1} := C_B^{(R)}$, transforming the final score as

$$C_{B,\tau} = \frac{C_{B,1}}{\tau} - \log Z_B(\tau) + \sum_{i \in B} \log Z_i(\tau) \tag{42}$$

therefore gives the exact error relation

$$C_{B,\tau} - G_{B,\tau} = \frac{C_{B,1} - G_{B,1}}{\tau}. \tag{43}$$

Singleton scores remain zero, and $\tau = 1$ recovers the original scores. The correction changes only final selection; untempered beliefs and couplings remain the inputs to the recurrent gates. The same-marginal intervention at temperature τ instead uses $\mathrm{Marg}_i\, Q_{B,\tau}$, which generally differs from $q_{i,\tau}$.

The proofs of both propositions apply to the normalized tempered heads without change.

A.4 UTILITY REGRESSION AND ITS INFORMATION RESTRICTION

Freeze the predictive functions and feature map, including the recurrence that produces them. Fix a candidate order $j \in \{2, 3\}$ with positive probability of eligible training contexts. All expectations in this subsection use the training-context distribution conditioned on $\mathcal{E}_j(c) \neq \varnothing$, stages sampled with probabilities w_r, and candidates sampled uniformly from $\mathcal{E}_j(c)$. The order-j coupling risk equals the resulting mean-squared risk multiplied by the probability of an eligible context, a constant that does not affect its minimizer. To include shared prediction across candidates and stages, write $V = (c, e, r)$, $Y = u_e^{(r)}(S_e; c)$, $G(V) = G_e^{(r)}(c)$, and $Z = z_e^{(r)}(c)$. Assume finite second moments. For any scalar a,

$$\mathbb{E}[(a - Y)^2 \mid V] = (a - G(V))^2 + \mathrm{Var}(Y \mid V). \tag{44}$$

An unrestricted predictor measurable with respect to the router features has the population optimum

$$C^*(Z) = \mathbb{E}[Y \mid Z] = \mathbb{E}[G(V) \mid Z]. \tag{45}$$

The second equality follows because Z is a deterministic function of V. The residual update does not change this optimum: its previous coupling is already a component of Z, so an unrestricted residual readout can represent any such $C^*(Z)$. Restricted neural function classes and shared parameterization can introduce further approximation error.

The tower property also gives

$$\mathbb{E}[Y \mid C^*(Z)] = C^*(Z). \tag{46}$$

This identity holds under the training sampling measure. Calibration concerns averages over contexts with equal scores rather than accuracy at each context. Because each context has a single native label, estimating the conditional mean relies on sharing across contexts.

During joint training, the predictive functions and features change. Detached targets remove only their direct derivative branch; differentiable inputs and recurrent messages still transmit gradients to shared parameters. For one target, the coupling-loss contribution to belief parameters ϕ is $2(C_\phi - u_\phi)\nabla_\phi C_\phi$, without the derivative of u_ϕ. The conditional-mean statement is therefore a fixed-function regression interpretation, not a convergence theorem for joint training.

A.5 Greedy selection on predicted weights

At a fixed context let w_e be the final score supplied to the selector, using the temperature correction when applicable. Singleton weights are zero. Consequently, a maximum-weight partition can omit all nonpositive nonsingleton candidates and fill their positions with singletons. Let $\mathcal{G}$ be the positive blocks accepted by decreasing-weight greedy selection and $\mathcal{O}$ a maximum-weight disjoint family of positive-weight nonsingleton candidates. Each block of $\mathcal{O}$ is either accepted or first blocked by an earlier accepted greedy block whose weight is at least as large. Charge its weight to that greedy block. Since the optimal blocks are mutually disjoint, a greedy block e receives at most $|e| \leq 3$ charges. Hence

$$\sum_{e\in\mathcal{O}} w_e \leq \sum_{e\in\mathcal{G}} |e| w_e \leq 3 \sum_{e\in\mathcal{G}} w_e. \tag{47}$$

This gives a $1/3$ approximation to the predicted-score objective, or $1/2$ when only pairs are available; the zero-optimum case is immediate. The bound relies on the zero-weight singleton fallback and the absence of quotas, and concerns the predicted scores rather than the true utilities.

A.6 From score error to selection regret

Fix a context, reveal set, predictive heads, and temperature τ, and use the same feasible partitions $\mathfrak{P}(U)$ for both objectives. Write $F(\pi) = \sum_{B\in\pi} G_{B,\tau}$ and $\widehat{F}(\pi) = \sum_{B\in\pi} C_{B,\tau}$, with zero singleton gains. For a true-gain maximizer $\pi^\star$ and the executed greedy partition π_g, define

$$E = \max_{\pi\in\mathfrak{P}(U)} \sum_{B\in\pi} |G_{B,\tau} - C_{B,\tau}|, \qquad \Delta_{\text{pack}} = \max_{\pi\in\mathfrak{P}(U)} \widehat{F}(\pi) - \widehat{F}(\pi_g). \tag{48}$$

Then

$$F(\pi^\star) - F(\pi_g) \leq 2E + \Delta_{\text{pack}}. \tag{49}$$

Indeed, insert $\widehat{F}(\pi^\star)$ and $\widehat{F}(\pi_g)$ between the two true objectives: the two score-error terms are each at most E, and the predicted-objective difference is at most Δ_{pack}. The bound separates utility-estimation error from packing error.

B Experimental protocol

Data curation and separation. We use the gRNAde DAS-Split release (Joshi et al., 2025), removing exact retained-sequence matches to training from the 100 source validation groups leaves 98 selection-validation targets. This common development split is separate from the test selection below.

Compared generators share chain extraction, canonical-parent mapping of modified nucleotides, and retained positions with complete P, C4′, and glycosidic-N coordinates. Targets require at least three retained positions and an unambiguous A/C/G/U reference at every design position. Source-position mappings, chain boundaries, and covalent gaps are preserved; coordinate masks encode bead roles rather than nucleotide identity. Target masks are fixed before generation, and native refolding does not determine eligibility.

DAS55 input selection. Before generation, we select DAS55 from the 98 source test sequence groups, each represented by one fixed conformation. We exclude 20 multi-source-chain targets and 23 single-chain targets with discontinuous covalent connectivity (13 with internal source-position gaps and 10 with other author-label connectivity breaks), leaving $98 - 20 - 23 = 55$ targets. Covalent edges require consecutive retained source indices on one chain and either consecutive author residue numbers or a changed insertion code at the same number. All retained targets satisfy the compared systems' atom and source-secondary-structure requirements, with no further exclusions. DAS55 contains 4,214 design positions across targets of length 18–159; these input checks require no additional position deletion or coordinate reconstruction.

Checkpoint and evaluator provenance. Common-DAS retraining uses shared conformers and design positions; optimization recipes and historical tuning budgets differ across methods. RDesign and RIdiffusion retain released-checkpoint provenance and the adaptations in Appendix C.

Table 4: DAS data composition. Development counts are sequence groups; the test count is the common-supported target set.

Dataset	Train	Validation	Test
DAS development / DAS55 test	4,012	98	55

Reveal traces and native-prefix scoring. For each target and generation seed, we draw and store 16 independent residue permutations. Each permutation defines one of the 16 generated designs and is partitioned according to (58). All fixed-schedule controls reuse this exact trace bank. Training contexts follow the same length-dependent schedule. At inference, the current context contains each method's generated prefix; at native-prefix evaluation, earlier chunks contain the native nucleotide identities. Unknown identities in U_t never enter selection. Each conditional NLL sums log scores along a complete trace, divides by L, and is averaged over the 16 traces within each generation seed, then across seeds and targets with equal target weights. Each trace has an exact schedule-conditioned NLL at $\tau = 1$; averaging the finite trace bank estimates its expectation over schedules, rather than the schedule-marginalized likelihood.

Fixed-context stage scores are aggregated with weights $|U_t|/L$. At every stage r, singleton blocks use $Q^{(r)}_{\{i\}} = q^{(r)}_i$ and $C^{(r)}_{\{i\}} = 0$. The same-model singleton reference reuses the full model's conditional q_i functions, without changing its refinement computation or training parameters.

Candidate availability and executed coverage. For order $j \in \{2, 3\}$, the selected-position coverage is

$$s_j = \frac{1}{L} \sum_t \sum_{\substack{B \in \pi_{t,\tau} \\ |B| = j}} |B|. \tag{50}$$

Table 3B reports $100s_2$ and $100s_3$ at native contexts and $\tau = 1$, averaged over complete traces within each generation seed, then across seeds and targets. Candidates in M_t can contribute to refinement, whereas execution requires all candidate positions to lie in U_t.

Crossed recurrence diagnostics. At identical native contexts and reveal sets, we cross the first- and final-stage predictive heads with the partitions selected by their scores. Writing $\mathcal{B}^{(r)}$ for the partition selected at stage r, define

$$\mathcal{L}_{r,s}(c_t) = -\frac{1}{|U_t|} \sum_{B \in \mathcal{B}^{(r)}} \log Q^{(s)}_B(S^*_B \mid c_t), \qquad r, s \in \{1, R\}. \tag{51}$$

At $\tau = 1$, scores aggregate as $L^{-1} \sum_t |U_t| \mathcal{L}_{r,s}(c_t)$ per trace, then average over traces, generation seeds, and targets. Table 3C reports the resulting NLLs. With final heads fixed, updated selection scores reduce NLL from 0.599 to 0.597; with the final partition fixed, updated heads reduce NLL from 0.604 to 0.597. For recurrence diagnostics, a position's partner set is the other positions in its selected block, and is empty for a singleton. The changed-partner fraction is $L^{-1} \sum_t \sum_{i \in U_t} \mathbf{1}\{P^{(1)}_i \neq P^{(R)}_i\}$, where $P^{(r)}_i$ is the partner set under stage-r selection, with the same trace, generation-seed, and target averaging.

Intervention definitions. Run the unmodified recurrent loop before either selector intervention. The learned policy uses $C^{(R)}_{B,\tau}$; TC-greedy replaces only this final score with $\mathrm{TC}(Q_{B,\tau})$. Both use the same greedy nonoverlap rule and singleton fallback. Recurrent gates are unchanged. Differences in selected coverage are part of this policy comparison: nonnegative TC and signed predictive utility need not make the same abstention decisions.

At a fixed native context and selected partition, the same-marginals intervention replaces $Q_{B,\tau}$ by the product of its own marginals, computed after temperature adjustment. In a complete rollout, each arm recomputes its states and routes from its own generated prefix while retaining the same external reveal sets. The fixed-context comparison reports predictive scores; rollout comparisons report scTM and BP F1. Signed dependence-retention and marginal-prediction gains are reported in nats per nucleotide.

Forward prediction. For RhoFold+, we use single-sequence inputs (code revision `e36efb1`, released `rhofold_pretrained_params.pt`), with predictor seed 0, 10 recycles, and no relaxation. For the three-oracle comparisons, all 55 DAS55 targets are evaluated with RhoFold+, AlphaFold3, and DRfold2 on identical designs, without resampling or retuning for each evaluator. Prediction failure never changes the target denominator and receives zero scTM.

AlphaFold3 receives one RNA chain with `unpairedMsa: ""`, without `pairedMsa` or `templates` fields. The input contains the designed sequence and no partner chain, ligand, target coordinates, or supplied alignment. We use model seed 0, 10 recycles, and five diffusion samples per input sequence. The highest internally ranked structure is evaluated; ranking never accesses the target. All 16 designed sequences per target and generation seed remain in the unranked design evaluation. AlphaFold3's structural training cutoff is 2021-09-30.

Structural metric and coordinate correspondence. Let I_X and $I_{\widehat{X}}$ contain the design positions with observed target and predicted C4′ coordinates, respectively. Define $n_X = |I_X|$, $I_{\text{match}} = I_X \cap I_{\widehat{X}}$, and $n_{\text{match}} = |I_{\text{match}}|$. Usable coordinates are matched target–prediction C4′ pairs at these same design positions. Target eligibility requires $n_X \geq 3$. We evaluate scTM on $[0, 1]$ using US-align (Zhang et al., 2022) in RNA mode. Target and prediction indices are mapped to design positions before alignment; `-TMscore 1` fixes this correspondence and `-atom " C4'"` selects the representative atom. We set `-u` to n_X and read the score normalized by this user-specified length. Positions outside I_X are excluded; missing predicted coordinates within I_X contribute zero to the numerator without reducing the denominator. Complete prediction failures and predictions with $n_{\text{match}} < 3$ receive zero scTM. Normalization remains n_X, not n_{match}. Sequence length L remains the denominator for native sequence NLL and recovery.

Cross-oracle scTM and disagreement. For design k of target x and generation seed s, let a_{xsko} denote its target-referenced scTM under oracle $o \in \{\text{RhoFold+}, \text{AF3}, \text{DRfold2}\}$. On the same three-oracle design records, define

$$\bar{a}_{xsk} = \frac{1}{3}\sum_{o=1}^{3} a_{xsko}, \qquad d_{xsk} = \left[\frac{1}{2}\sum_{o=1}^{3}(a_{xsko} - \bar{a}_{xsk})^2\right]^{1/2}. \tag{52}$$

Tables 1–3 average $\bar{a}_{xsk}$ within each target and generation seed, then across the three seeds; Table 1 applies the same aggregation to d_{xsk}. The resulting target scores are averaged equally. Disagreement is the mean per-design sample standard deviation, not the standard deviation of aggregated method scores. Lower disagreement indicates closer oracle scores, rather than necessarily higher structural accuracy. All three oracles evaluate the same saved sequences without resampling or oracle-specific design selection; duplicate sequences retain their original design multiplicities. FD retains the RhoFold+ validity threshold $\gamma = 0.45$ and is computed separately within each target and seed.

Base-pair recovery. The BP F1 results in Tables 2 and 3B use MC-Annotate 2.0.0 for both structures, retaining the same position correspondence, target-defined observable base-atom masks, and F1 definition below. Let $\mathcal{P}_X$ and $\mathcal{P}_{\widehat{X}}$ contain the canonical and noncanonical base pairs within a target-defined observable-position mask. The same target mask applies to every prediction and is not reduced when predicted coordinates are missing. Base-pair F1 is

$$F_{\text{BP}} = \frac{2|\mathcal{P}_X \cap \mathcal{P}_{\widehat{X}}|}{|\mathcal{P}_X| + |\mathcal{P}_{\widehat{X}}|}. \tag{53}$$

The empty-set convention applies only to targets with valid annotations and a nonempty scored pair universe; unscorable targets are recorded separately. For successful predictions and annotations on these targets, two empty sets receive a score of one, and exactly one empty set receives zero. Forward-prediction or prediction-annotation failures receive zero regardless of the target pair count. Scores are averaged over designs within each target and generation seed, then across seeds and scorable targets with equal target weights; target eligibility is determined independently of generated predictions. This endpoint measures pair recovery without requiring an identical interaction class.

Aggregation, diversity, and operating points. For target x, generation seed $s \in \{1, 2, 3\}$, and design $k \in \{1, \ldots, M\}$ with $M = 16$, let a_{xsk} be its scTM, including zeros for failed predictions. The target-macro endpoint on target set D is $|D|^{-1}\sum_{x\in D} 3^{-1}\sum_{s=1}^{3} M^{-1}\sum_{k=1}^{M} a_{xsk}$.

For a fixed target and generation seed, write the sequences as $S^{(1)}, \ldots, S^{(M)}$ and their RhoFold+ scTM scores as $a_1, \ldots, a_M$. Over the same L design positions, normalized Hamming distance is $d_{ab} = L^{-1} \sum_{i=1}^{L} \mathbf{1}[S_i^{(a)} \neq S_i^{(b)}]$. With structural threshold $0 < \gamma \leq 1$, set $v_a = \mathbf{1}[a_a \geq \gamma]$ and define sequence diversity and foldable diversity (adapted from Ektefaie et al., 2024) as

$$D_{\text{seq}} = \binom{M}{2}^{-1} \sum_{a<b} d_{ab}, \qquad \text{FD}_\gamma = \binom{M}{2}^{-1} \sum_{a<b} v_a v_b d_{ab}. \tag{54}$$

Repeated sequences are retained, and all $\binom{M}{2} = 120$ pairs remain in both denominators. FD combines structural validity with sequence diversity and is zero when fewer than two designs pass the threshold. The scTM reference is the experimental target structure. Each diversity score is computed within one target and generation seed, then averaged across seeds and targets with equal target weights; sequences from different targets or seeds are never pooled. Recovery follows the same aggregation after averaging designs within each target and seed and is reported as a percentage. Sampling temperatures are selected on validation data; released-checkpoint reference rows retain their documented settings. Native-prefix NLL uses $\tau = 1$, with checkpoint selection specified in Appendix C. For RhoFold+ on DAS55, FD uses the fixed threshold $\gamma = 0.45$. The validity threshold is shared across methods, and failed predictions are never valid designs. Hidden-only checkpoints select temperature under singleton execution; their geometry-greedy and learned-gain policies inherit it. Full RGCR selects temperature under learned-gain joint execution; same-checkpoint product-of-marginals and TC-greedy interventions inherit it. Cross-evaluator comparisons reuse the same designs without retuning.

Fixed-checkpoint schedule sensitivity. The full-RGCR latency sweep uses step caps $K \in \{4, 8, 16, 64\}$ and $\tau = 0.1$. Each residue permutation is divided into $T_K(L) = \max\{1, \min(K, \lfloor L/3 \rfloor)\}$ balanced chunks. The checkpoint and refinement depth remain fixed while the inference schedule changes. This changes both inference cost and the distribution of masked contexts, and measures schedule transfer at a fixed checkpoint. Latency includes cached geometry encoding, context initialization, prediction, recurrence, and selection, measured at the same batch size and hardware with device synchronization and warmup excluded. For $T = 4, R = 3$, generation comprises 12 prediction stages, eight hidden-state updates, four context initializations, and one geometry encoding.

C Implementation and Reproducibility Details

Generation cost. Let $E = |\mathcal{H}_2| + |\mathcal{H}_3|$ and $E_t = |\{e \in \mathcal{H}_2 \cup \mathcal{H}_3 : e \subseteq U_t\}|$. With fixed hidden widths and alphabet size, a generation step costs $O(R(L+E) + E_t \log(1+E_t))$, excluding the once-per-target geometry encoding and candidate construction. The first term includes context initialization, local heads, and recurrent messages; the second bounds sorting for greedy selection. Parameters are shared across candidates, so candidate count affects computation rather than the number of independently parameterized heads.

Geometric features and coordinate validity. For residue i, let p_i, c_i, and n_i denote the P, C4′, and common glycosidic-nitrogen coordinates. The GVP encoder receives coordinates, chain positions, covalent connectivity, and bead-validity masks. A residue-local frame is defined by

$$a_i = p_i - c_i, \qquad e_{i,1} = a_i / \|a_i\|, \tag{55}$$

$$b_i^{\perp} = (n_i - c_i) - e_{i,1} e_{i,1}^{\top} (n_i - c_i), \tag{56}$$

$$e_{i,2} = b_i^{\perp} / \|b_i^{\perp}\|, \qquad e_{i,3} = e_{i,1} \times e_{i,2}, \qquad F_i = [e_{i,1}, e_{i,2}, e_{i,3}]. \tag{57}$$

The frame is valid only when all three coordinates are observed and both normalization denominators exceed a positive tolerance ϵ_{geom}; each validity test precedes its corresponding division. Local displacement $F_i^{\top}(c_j - c_i)$ requires a valid frame at i and an observed C4′ at j, whereas relative orientation $F_i^{\top} F_j$ requires both frames. Undefined features are zero-filled with an explicit validity

Algorithm 1 RGCR generation under an external reveal schedule

Require: Structure X, partition $\omega = (U_1, \ldots, U_T)$, temperature $\tau > 0$, stages R

Cache geometric features and candidates $\mathcal{H}_2, \mathcal{H}_3$ from X
Initialize every sequence position as masked
for $t = 1, \ldots, T$ **do**
  Form c_t and unresolved set M_t from the current sequence
  Initialize $h^{(1)}$ by the context pass; set $C^{(0)} = 0$
  for $r = 1, \ldots, R$ **do**
    Predict $q_i^{(r)}$ and $Q_e^{(r)}$ for candidates $e \subseteq M_t$ using (2)
    Form $z_e^{(r)}$ and update $C_e^{(r)}$ using (5)
    if $r < R$ **then**
      Refine $h^{(r+1)}$ with fully masked candidate messages using (7)
    end if
  end for
  Temper final heads and adjust gains using (12)–(13)
  Sort candidates $e \subseteq U_t$ with $C_{e,\tau}^{(R)} > 0$ by decreasing score
  Break ties by canonical indices; greedily accept disjoint candidates
  Complete the partition π_t with singleton blocks
  for $B \in \pi_t$ **do**
    Draw $S_B \sim Q_{B,\tau}$ independently across blocks, using $Q_{\{i\},\tau} = q_{i,\tau}^{(R)}$
  end for
  Commit all sampled identities in U_t simultaneously
end for
return Complete sequence S

mask. Inter-bead distances remain available whenever their constituent coordinates are observed, independently of frame validity. Under a global rotation $R \in SO(3)$ and translation, F_i transforms as RF_i, so these local-frame features and the distance features are invariant. The pair feature ϕ_{ij} concatenates radial-basis distance encodings, local-frame features, sequence separation, and validity masks in canonical residue order. Coordinates are measured in Å, with $\epsilon_{\text{geom}} = 10^{-6}$ Å. Each distance uses 16 radial basis functions $\exp[-((d-\mu)/2)^2]$, with centers $\mu = 0, 2, \ldots, 30$ Å; local displacements are divided by 30 Å. Signed source-position and retained-position gaps use $\operatorname{sgn}(\Delta)\log(1+|\Delta|)/\log(513)$, with chain and source-mapping validity indicators.

Candidate construction. The initial pair graph combines covalent neighbors with a symmetrized $k = 32$ C4′ nearest-neighbor graph. Spatial edges and triples require valid C4′ coordinates; covalent candidates remain defined by chain connectivity. For each anchor, every unordered pair of its neighbors in the initial graph forms a triple candidate, irrespective of whether the two neighbors share an initial edge. Candidates are ranked by the sum of their three C4′ pair distances, retaining at most $k_3 = 8$ triples per anchor. Distance ties are broken by chain position. Triples are deduplicated, their three pairwise faces are added to $\mathcal{H}_2$, and all candidate axes follow increasing chain-position order. The $O(L(k + k_3))$ bound counts retained candidates, not the cost of nearest-neighbor search.

Reveal schedule. We draw a random permutation of the positions and divide it into nearly equal consecutive chunks with

$$T(L) = \max\{1, \min(4, \lfloor L/3 \rfloor)\}. \tag{58}$$

Chunk sizes differ by at most one, and each chunk contains at least three positions for $L \geq 3$. Training and default evaluation use this rule; fixed-schedule controls share the same reveal traces. Appendix B varies the inference schedule at fixed checkpoints.

Belief features. For $Q_{e,i}^{(r)} = \mathrm{Marg}_i\, Q_e^{(r)}$, the coupling predictor uses

$$D_e^{(r)} = D_{\mathrm{KL}}\left(Q_e^{(r)} \,\middle\|\, \prod_{i\in e} Q_{e,i}^{(r)}\right), \tag{59}$$

$$A_e^{(r)} = \sum_{i\in e} \mathrm{JS}\left(Q_{e,i}^{(r)}, q_i^{(r)}\right). \tag{60}$$

Here $D_e^{(r)}$ is model total correlation (mutual information for a pair), and $A_e^{(r)}$ is summed Jensen–Shannon disagreement between joint-head marginals and singleton predictions. Both are computed exactly from the local categorical tables.

Architecture. Table 5 summarizes the geometric candidate and refinement settings. Geometry is cached across generation steps, while sequence-conditioned states and coupling scores are reinitialized at each step. Predictive heads, coupling updates, and message maps share parameters across refinement stages, with separate maps for pairs and triples. Order-specific message maps return the same hidden dimension. Gate scalars a_j and b_j are unconstrained learned parameters, so utility conditions message transmission without imposing monotonicity in the utility score. The initial context pass includes revealed and masked residues; subsequent candidate messages use fully masked pairs and triples. The GVP encoder has three layers with $(h, 16)$ scalar/vector node channels and $(h/4, 1)$ edge channels; the sequence-conditioned context uses two message-passing layers. Head and message MLPs have two linear layers with hidden width h and SiLU activation; candidate geometry maps instead use hidden and output width 128. Scalar states use LayerNorm, vector states use root-mean-square normalization, and dropout is zero. Probability tables are linearly embedded into 32 dimensions without bias; log-softmax and log-sum-exp reductions use float32.

Table 5: RGCR architectural configurations.

Setting	Value
Hidden width, h	128 or 256
Candidate geometry width	128
GVP layers / context layers	3 / 2
Node vector channels / belief embedding width	16 / 32
Dropout	0
C4′ nearest neighbors, k	32
Retained triples per anchor, k_3	8
Refinement stages, R	3
Stage supervision weights, w_r	$1/R$
Singleton / pair / triple categorical entries	4 / 16 / 64
Initial coupling state, $C_e^{(0)}$	0

Training and selection settings. Training uses seed 0 and AdamW with weight decay 0.01 and gradient-norm clipping at 1.0. Each optimizer update accumulates four uniformly sampled targets with replacement, weighting targets equally. For each target, we sample a uniform residue permutation and a uniform pre-reveal stage of the schedule in (58). Checkpoints are selected using native-prefix validation NLL at $\tau = 1$. No gradient is taken through discrete block selection. Algorithm 1 specifies greedy selection; score ties are broken lexicographically by canonical candidate indices.

Controlled configurations. Table 6 distinguishes changes to the trained architecture from changes to its execution policy. Singleton execution, geometry-greedy pair execution, and learned-gain execution share the training-seed-0 hidden-only checkpoint and its auxiliary prediction and utility heads. Its messages exclude belief and coupling feedback, its gates are constant, and each stage has a one-shot utility readout. Geometry-greedy execution ranks pairs by increasing C4′ distance, breaks ties by chain position, and gives missing distances lowest priority. All policies use singleton fallback. One-shot RGCR, full RGCR, and pair-only RGCR are trained separately. The pair-only model removes triple heads, losses, and messages while retaining all pair candidates, including triple-face closure.

Table 6: Controlled architectures and execution policies. The three hidden-only policies share the training-seed-0 checkpoint. Feedback is absent (No) or includes beliefs and utilities (Both).

Configuration	R	Feedback	Execution
Hidden-only: singleton	3	No	Singleton heads
Hidden-only: geometry-greedy	3	No	Geometric pairs
Hidden-only: learned gain	3	No	Gain-selected pairs/triples
One-shot RGCR	1	No	Gain-selected pairs/triples
Full RGCR (Ours)	3	Both	Gain-selected pairs/triples
Pair-only RGCR	3	Both	Gain-selected pairs

System configurations. Table 7 summarizes the input and decoding configurations of the complete systems. The gRNAde comparison uses its single-backbone configuration, consistent with the single-target task. The common-DAS GALS-Fold configuration uses code revision `970157d`, the shared DAS development manifests, and validation operating-point selection. RDesign uses the released network with an SS-only correction adapter and P/O5′/C5′/C4′/C3′/O3′ coordinates from the same source conformation. Its source-record secondary-structure pairs are projected onto retained positions, dropping pairs with missing endpoints without native-base compatibility filtering; deterministic repeated designs are retained. RGCR uses coordinates, coordinate-validity masks, chain positions, and covalent connectivity without secondary-structure annotations.

Table 7: Input and decoding configurations for the main comparison.

System	Input and design procedure
gRNAde (Joshi et al., 2025)	Single target backbone; geometric autoregressive decoding.
GALS-Fold (Wen et al., 2026)	Target geometry; local equivariance and global attention.
RDesign (Tan et al., 2024)	Source backbone and secondary-structure pairs; SS-only correction adapter.
RIdiffusion (Hou et al., 2025)	Target backbone; structure-conditioned discrete diffusion with adapted released weights.
RGCR (Ours)	Three-coordinate residue representation; gain-selected local joint generation.