ALIGNING GENOME LANGUAGE WITH STRUCTURAL DIFFUSION FOR NUCLEIC ACID DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Genome language models and structure-conditioned inverse-folding networks learn complementary views of nucleic acids, linking sequence variation to the three-dimensional folds and molecular environments in which RNA and DNA operate. We introduce EVO-IF, which aligns genome-language representations with a compact structural diffusion model for prescribed backbones and nucleic-acid assembly contexts. A 1.18M-parameter context-aware interface enables geometric residue states to retrieve compressed representations from a frozen EVO2 encoder, with staged adaptation across molecular contexts. EVO-IF achieves 86.81% native sequence recovery on the Das Benchmark, 18.94 percentage points above NA-MPNN, and 76.96% across 121 DNA-containing assemblies. The capacity-matched training comparison shows gains of 8.49 and 9.68 percentage points over the structure-only reference on RNA and DNA, respectively. Among the tested benchmark decoding settings, one-pass decoding achieves the highest recovery while reducing measured latency by avoiding costly diffusion refinement. Together, EVO-IF extends genome-language representations to macromolecular design under explicit geometric and assembly constraints.

1 INTRODUCTION

Nucleic-acid inverse folding seeks sequences compatible with a prescribed three-dimensional backbone and its molecular environment. It provides a computational basis for designing structured RNAs and studying sequence preferences at protein–nucleic-acid interfaces (Joshi et al., 2025; Kubaney et al., 2025). A successful sequence must satisfy local backbone geometry, long-range base pairing and stacking, and contacts with neighboring chains. RNA conformational diversity further complicates this task. Compatibility with one local motif need not imply compatibility with the complete fold or its alternative states (Tinoco & Bustamante, 1999; Ganser et al., 2019). Designing within heterogeneous assemblies therefore requires combining geometric constraints with sequence dependencies across molecular contexts (Ma et al., 2025; **?**).

Structure-conditioned learning has made substantial progress on this problem. ProteinMPNN and ESM-IF established effective approaches to protein inverse folding (Dauparas et al., 2022; Hsu et al., 2022). For RNA, RDesign develops data-efficient structural representations, gRNAde incorporates multiple backbone conformations, and RiboDiffusion learns sequence generation through denoising (Tan et al., 2024; Joshi et al., 2025; Huang et al., 2024). NA-MPNN extends geometric modeling to a shared representation of proteins, DNA, and RNA (Kubaney et al., 2025). These approaches provide increasingly expressive structural models, but paired sequence–structure data remain limited compared with the sequence data available for pretraining. Genome language models such as EVO2 offer a complementary source of learned sequence regularities (Brixi et al., 2026); using those representations for a specified molecular backbone requires an explicit interface to geometry.

The central modeling challenge is to relate sequence representations learned across genomes to the geometric constraints of a particular molecular assembly. Sequence order and spatial neighborhoods define different contexts, where residues far apart along a chain can meet in three dimensions, and neighboring polymers introduce contacts absent from an isolated sequence. We address this mismatch by making geometric residue states retrieve a shared genome representation, with polymer context controlling its contribution to sequence prediction. Absorbing-state diffusion provides the conditional sequence-reconstruction objective and supports both one-pass prediction and iterative refinement.

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.

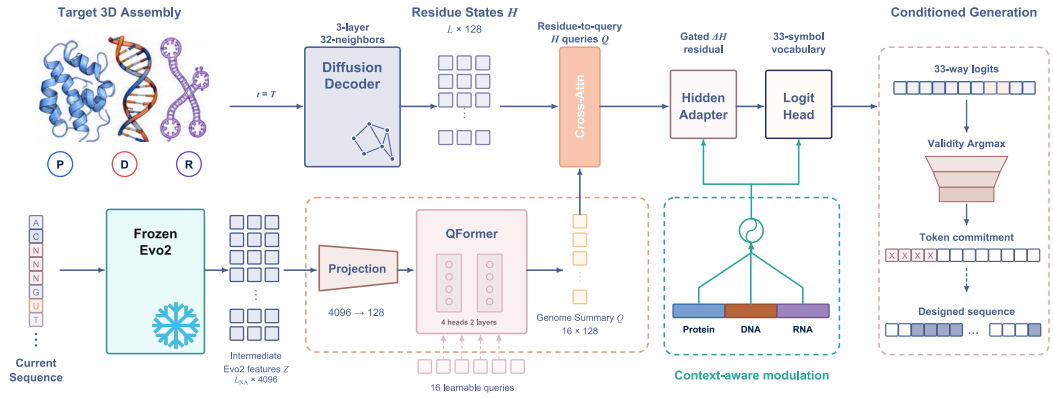

Figure 1: **Overview of EVO-IF.** A structural diffusion decoder represents the target assembly, while frozen EVO2 encodes the current nucleic-acid sequence. A learned-query bridge compresses genome features for residue-level cross-attention. Polymer-context modulation controls residual fusion and output corrections. Polymer-valid predictions yield the designed sequence in one pass, or update the candidate for re-encoding during iterative decoding.

We introduce EVO-IF, a genome-geometry framework aligning nucleic-acid sequence design (Figure 1). A 1.18M-parameter interface connects frozen EVO2-7B to the inverse-folding specific structural diffusion decoder. Sixteen learned queries summarize genome activations, which each structural residue retrieves through geometry-conditioned cross-attention. Protein, DNA, and RNA presence bits modulate this alignment and the residual corrections to sequence predictions. We optimize the coupled model for backbone-conditioned sequence reconstruction, using near-identity initialization and staged adaptation across polymer contexts. The same architecture supports one-pass prediction from a fully masked input and iterative decoding with re-encoding of the evolving candidate.

EVO-IF sets the highest mean recovery among the compared methods on the 14-structure Das Benchmark, where 86.81% in a single pass, 18.94 percentage points above NA-MPNN, with gains on all 14 targets. It also reaches 76.96% DNA recovery across 121 assemblies. Relative to the structure-only reference, the gains are 8.49 points on RNA and 9.68 on DNA, establishing a consistent advantage across both modalities. One-pass decoding further reduces measured latency by $30.7\times$ relative to the same model's Iter-30 refinement. We evaluate the complete system through benchmark comparisons and examine its behavior through matched training comparisons, chain-serialization controls, and fixed-weight genome-branch interventions. These experiments distinguish performance after adaptation from dependence on individual inputs during decoding. Self-consistency, multi-state, pseudoknot, temporal, and predicted-backbone evaluations examine fold compatibility, conformational diversity, and changes in structural input.

Our main contributions are:

- We connect genome-scale sequence modeling with molecular-scale nucleic-acid design. By coupling a genome language model to a structural decoder, we make target backbone geometry and surrounding molecular partners explicit conditions for sequence generation, providing a shared framework for RNA and DNA design within prescribed folds and assembly environments.
- We develop residue-level alignment that couples a genome encoder, learned-query compression, context-modulated structural diffusion, and staged adaptation across RNA, DNA, and protein-nucleic-acid contexts.
- We achieve leading Das recovery and substantial DNA gains across isolated and protein-associated assemblies, with one-pass decoding improving both recovery and latency over 30-step refinement for the released checkpoint.
- We provide controlled analyses of training, chain serialization, and inference-time branch dependence, together with evaluations of structural robustness, conformational diversity, and protein retention.

2 Related Work

Structure-conditioned inverse folding. Inverse folding seeks sequences compatible with a given three-dimensional backbone. Learning-based approaches represent spatial relationships between residues and learn conditional sequence distributions from paired sequence-structure data. ProteinMPNN uses message passing over backbone geometry for protein sequence design (Dauparas et al., 2022), while ESM-IF combines geometric representations with large-scale training on predicted structures (Hsu et al., 2022). NA-MPNN extends message passing to a unified biopolymer graph whose nodes can represent protein residues, DNA bases, or RNA bases (Kubaney et al., 2025). This representation includes contacts between molecular partners and supports both RNA sequence design and fixed-dock protein-DNA specificity prediction.

Classical RNA design methods commonly optimize sequences against secondary-structure targets, as in ViennaRNA (Lorenz et al., 2011). Rosetta fixed-backbone redesign incorporates three-dimensional constraints by sampling sequence and base-conformation changes under an all-atom energy function (Das et al., 2010; Leman et al., 2020). Recent approaches learn from RNA geometry and conserved structural patterns (Wang et al., 2026). RDesign develops hierarchical structural representations for data-efficient tertiary-structure-based design (Tan et al., 2024). gRNAde uses equivariant graph neural networks and represents multiple conformations of the same RNA as a geometric multigraph, allowing sequence design to account for conformational variability (Joshi et al., 2025). RBFN combines geometric algebra features with a Bayesian flow network to model nucleotide distributions and evaluates both single-state and multi-state inverse design (Wang et al., 2025a).

Non-autoregressive and diffusion-based generation. Non-autoregressive methods predict multiple sequence positions in parallel, with iterative updates introducing dependencies between predictions. MaskGIT and Muse progressively fill masked positions using bidirectional context (Chang et al., 2022; 2023), while discrete diffusion models learn to reverse a specified corruption process over categorical tokens (Austin et al., 2021). These ideas also support biomolecular generation and structure-guided optimization (Si et al., 2026; Hu et al., 2026). RiboDiffusion learns to recover RNA sequences from noisy inputs conditioned on tertiary structure (Huang et al., 2024). DPLM-2 extends discrete diffusion to joint modeling of protein sequences and tokenized structures, supporting unconditional generation as well as folding and inverse folding (Wang et al., 2025b). SearchDiff augments masked diffusion with inference-time search to improve satisfaction of user-specified constraints and properties (Ta et al., 2026).

Nucleic-acid language models. Self-supervised language models learn from sequence collections that are substantially larger than the available structural datasets. RNA-FM and RiNALMo use masked language modeling on unannotated RNA sequences to learn representations for downstream structure and function prediction (Chen et al., 2022; Penić et al., 2025). Evo2 extends sequence modeling to genomes across all domains of life, supporting sequence generation and prediction of variant effects (Brixi et al., 2026). These pretraining objectives capture structural regularities through sequence statistics without explicitly conditioning on target coordinates.

Cross-modal alignment. Structure-conditioned sequence design and image-to-text generation share a similar modeling challenge. Translating spatial features into conditioning representations that a pretrained sequence generator can use. Vision-language models attempt to narrow this gap that emphasize both selecting relevant information and preserving spatial detail. BLIP-2 introduces Q-Former, whose learnable queries are trained through representation alignment and generation objectives to extract language-relevant visual features, enabling a compact interface between frozen image and language models (Li et al., 2023). Qwen-VL combines learned-query compression with two-dimensional positional encodings in its cross-attention adapter to retain spatial information for fine-grained visual understanding (Bai et al., 2023). InstructPLM aligns backbone embeddings with a pretrained protein language model (Qiu et al., 2024); OmniGenome learns mappings between RNA sequences and secondary structures (Yang et al., 2025).

3 METHOD

3.1 MIXED-TYPE MACRO-MOLECULE INVERSE FOLDING

Let an assembly be represented by a residue graph $\mathcal{G} = (\mathbf{X}, \mathbf{A}, \mathbf{p})$, where $\mathbf{X}$ contains observed atomic coordinates, $\mathbf{A}$ denotes geometric neighborhood features, and $\mathbf{p}$ specifies chain and polymer identity. A binary design mask $\mathbf{m}$ selects the residues whose native categorical sequence $\mathbf{s}_0$ is to be recovered. The model estimates

$$p_\theta(\mathbf{s}_0 \mid \mathcal{G}, \mathbf{m}) = \prod_{i:m_i=1} p_\theta(s_{0,i} \mid \mathcal{G}, \mathbf{s}_t, t) \tag{1}$$

while keeping non-designed residues available as structural and sequence context. The shared 33-symbol vocabulary represents protein, DNA, RNA, unknown, and mask tokens. Output validity masks prevent a residue from receiving a token of the wrong polymer type.

Absorbing structural diffusion. We use a compact encoder–decoder as the geometric backbone. Its forward process independently replaces a designable token by an absorbing mask state. For cumulative retention probability $\bar{\alpha}_t$,

$$q(s_{t,i} \mid s_{0,i}) = \mathrm{Cat}\big(\bar{\alpha}_t\, \mathbf{e}_{s_{0,i}} + (1-\bar{\alpha}_t)\, \mathbf{e}_{\text{MASK}}\big) \tag{2}$$

The structure encoder builds residue states from a 32-neighbor geometric graph. A three-layer non-causal decoder combines those states with the partially observed sequence, producing $\mathbf{H}_t \in \mathbb{R}^{L\times 128}$.

3.2 FROZEN GENOME-LANGUAGE REPRESENTATION

At each reverse step we extract the current nucleic-acid subsequence from $\mathbf{s}_t$. RNA display symbols are mapped to the DNA alphabet (U $\mapsto$ T), unknown or masked symbols become N, and nucleic-acid chains are concatenated without separator tokens. For an entirely unknown input, the default implementation initializes the genome path with a structure-predicted candidate while keeping the diffusion state masked. The resulting encoder input $\mathbf{s}_t^{\mathrm{NA}}$ is passed through frozen EVO2-7B. We capture the intermediate activation at `blocks.28.mlp.l3blocks.28.mlp.l3`

$$\mathbf{Z}_t = f_{\text{Evo}}^{(28)}(\mathbf{s}_t^{\mathrm{NA}}) \in \mathbb{R}^{L_{\mathrm{NA}}\times 4096} \tag{3}$$

with gradients disabled. Recomputing $\mathbf{Z}_t$ is essential. It makes the sequence prior conditional on decisions already made by the reverse process. During iterative decoding, we recompute $\mathbf{Z}_t$ from the updated candidate as predicted bases replace masks. For protein-only assemblies, no nucleic-acid string exists and the entire EVO2 branch is bypassed.

3.3 LEARNED-QUERY ALIGNMENT BRIDGE

Alignment links genome representations to the residue states encoding a target molecular geometry. Following the learned-query interfaces reviewed in Section 2, we project the genome states and summarize them with $K = 16$ learned queries,

$$\mathbf{E}_t = \mathbf{Z}_t \mathbf{W}_E, \qquad \mathbf{E}_t \in \mathbb{R}^{L_{\mathrm{NA}}\times 128}, \tag{4}$$

$$\mathbf{Q}_t = \mathrm{QFormer}(\mathbf{Q}_0, \mathbf{E}_t) \qquad \mathbf{Q}_t \in \mathbb{R}^{16\times 128} \tag{5}$$

The query encoder has two pre-normalized Transformer-decoder layers, four heads, and zero dropout. Each residue then retrieves a geometry-dependent combination of the global genome summaries,

$$\mathbf{U}_t = \mathrm{LN}(\mathrm{MHA}(\mathbf{H}_t, \mathbf{Q}_t, \mathbf{Q}_t)) \tag{6}$$

Thus, sequence-length-dependent computation is compressed before residue-level fusion, and all assembly residues can access the same global summary.

3.4 CONTEXT-AWARE MODULATION

We condition the alignment on the polymer composition of the target assembly, encoded by three presence bits,

$$r = 4\mathbb{I}[\text{protein}] + 2\mathbb{I}[\text{DNA}] + \mathbb{I}[\text{RNA}], \qquad \mathbf{c} = \mathrm{onehot}_8(r) \tag{7}$$

The polymer context modulates a scalar fusion gate, a channel-wise scale, and rank-16 residual adapters:

$$\gamma(\mathbf{c}) = \sigma(g_0 + \mathbf{w}_g^\top \mathbf{c}) \tag{8}$$

$$\mathbf{a}(\mathbf{c}) = 1 + \tanh(\mathbf{W}_s \mathbf{c}) \tag{9}$$

$$\widetilde{\mathbf{H}}_t = \mathbf{H}_t + \gamma(\mathbf{c})\,[\mathbf{a}(\mathbf{c}) \odot \mathbf{U}_t + \mathbf{A}_r(\mathbf{U}_t)] \tag{10}$$

where $\mathbf{A}_r(\mathbf{x}) = \mathbf{U}_r\,\mathrm{GELU}(\mathbf{D}_r\mathbf{x})$ is a context-conditioned low-rank map. A second context-conditioned rank-16 map corrects the vocabulary logits,

$$\boldsymbol{\ell}_t = \mathbf{W}_{\mathrm{out}}\widetilde{\mathbf{H}}_t + \mathbf{B}_r(\widetilde{\mathbf{H}}_t) \tag{11}$$

We initialize $g_0 = -4$ and all adapter output matrices to zero. Consequently, the initial fusion gate is $\sigma(-4) \approx 0.018$ and the adapter corrections are exactly zero. Alignment therefore starts as a small correction to the existing geometry-conditioned predictions.

3.5 Optimization and decoding

We minimize masked categorical cross-entropy on the native identity of designable residues,

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{\sum_i m_i} \sum_{i:m_i=1} \log p_\theta(s_{0,i} \mid \mathcal{G}, \mathbf{s}_t, t) \tag{12}$$

The genome model remains frozen throughout. Training proceeds through bridge alignment with the structural decoder frozen, composition-adapter training, and a low-learning-rate structural-decoder phase with the bridge fixed. The final balanced phase contains 600 updates at learning rate 5×10^{-6}, uses fully masked design positions, and covers RNA, DNA-only, protein–DNA, protein–DNA–RNA, and heterogeneous-complex training examples. The full protocol and evaluation panels appear in Appendices B.2 and A.1.

At inference, one-pass decoding predicts all designed tokens from a fully masked input in one forward evaluation. Iterative decoding follows a cosine MaskGIT-style schedule (Chang et al., 2022): at each of 30 steps it predicts all sites, commits the highest-confidence subset, optionally remasks low-confidence sites, and re-encodes the updated nucleic-acid sequence with Evo2. We use argmax decisions and a fixed seed for the released checks. Algorithmic details and the complete configuration appear in Appendix B.1.

4 Experiments

We evaluate Evo-IF across RNA and DNA inverse folding, heterogeneous nucleic-acid assemblies, and a protein-retention stress test. The evaluations examine sequence recovery, compatibility with target folds, and behavior across molecular contexts. Complete protocols and numerical results appear in Appendix A.

4.1 Experimental setup

Datasets. Training uses mixed-polymer structural data under the curriculum described in Appendix B.2. For RNA sequence recovery, the Das Benchmark (Das et al., 2010) comprises 14 experimentally determined structures spanning aptamers, riboswitches, ribozymes, and pseudoknots; each target supplies a single backbone for sequence design. RNA structural self-consistency is evaluated on the released gRNAde single-state benchmark (Joshi et al., 2025). Multi-state design uses the 100-group StructSim benchmark following Joshi et al. (2025); Wang et al. (2025a), where each group contains conformations of the same RNA sequence and the model receives up to one, two, or three conformers as input. For DNA design, DNA-121 contains 121 biological assemblies, including 28 DNA-only, 78 protein-DNA, and 15 protein-DNA-RNA structures. The full assembly provides molecular context, while recovery is scored exclusively at DNA positions. Additional evaluations cover 10 pseudoknotted RNAs, RhoFold-predicted versions of the Das backbones, 116 CATH 4.2 short protein chains, and a temporal panel of 12 structures released after checkpoint selection. Task-specific protocols are detailed in Appendices A and C.

Metrics and aggregation. We assess both agreement with native sequences and compatibility with target structures. For target j, native sequence recovery and perplexity are

$$R_j = \frac{1}{|\mathcal{M}_j|} \sum_{i \in \mathcal{M}_j} \mathbb{I}[\widehat{s}_{j,i} = s_{j,i}], \tag{13}$$

$$\mathrm{PPL}_j = \exp\left[-\frac{1}{|\mathcal{M}_j|} \sum_{i \in \mathcal{M}_j} \log p_{j,i}(s_{j,i}) \right], \tag{14}$$

where $\mathcal{M}_j$ contains the evaluated design positions, $s_{j,i}$ and $\widehat{s}_{j,i}$ are native and designed residue identities, and $p_{j,i}$ is the model's residue distribution under the evaluated decoding setting. Recovery is reported as a percentage; lower perplexity indicates greater support for the native sequence. Following gRNAde (Joshi et al., 2025), we compute structural self-consistency by forward-folding designed RNA sequences. EternaFold (Wayment-Steele et al., 2022) supplies secondary-structure predictions. RhoFold (Shen et al., 2024) supplies tertiary-structure predictions, which are aligned to the target backbone. We report scRMSD, the coordinate root-mean-square deviation in Å, and scTM-score, a length-normalized measure of fold similarity. For aligned residue distances $d_{j,i}$ in Å over n_j evaluated residues, scGDT_TS averages agreement across four distance thresholds:

$$\mathrm{scGDT}_{\mathrm{TS},j} = \frac{1}{4n_j} \sum_{\tau \in \{1,2,4,8\}} \sum_{i=1}^{n_j} \mathbb{I}[d_{j,i} \le \tau]. \tag{15}$$

Higher scMCC, scTM-score, and scGDT_TS and lower scRMSD indicate better structural agreement. For pseudoknots, OpenKnot Score additionally measures agreement between RibonanzaNet-predicted chemical reactivity and the target pairing pattern (Kubaney et al., 2025). Unless otherwise specified, we average per-target scores with equal weight across structures or RNA groups and report their standard deviations. Pseudoknot recovery uses the median across 10 designs per structure, followed by the median across structures. OKS aggregation is specified in Appendix A.3.

Baselines. The Das comparison includes ViennaRNA and FARNA, together with the learned inverse-folding methods RDesign, gRNAde, RhoDesign, RBFN, NA-MPNN (Das et al., 2010; Tan et al., 2024; Joshi et al., 2025; Shen et al., 2024; Wang et al., 2025a; Kubaney et al., 2025). DNA comparisons use NA-MPNN on the same 121 assemblies. For RNA self-consistency, gRNAde and RBFN provide published reference values, while NA-MPNN and EVO-IF are evaluated with the reproduced folding pipeline. Capacity-matched training comparisons and fixed-weight branch interventions separately examine the effects of adaptation and inference-time conditioning. EVO-IF uses one-pass decoding unless explicitly labeled Iter-30. The comparison settings and numerical results accompany the tables in Appendix A.

4.2 NUCLEIC ACID INVERSE FOLDING

Figure 2 summarizes the field comparison on RNA inverse folding.

Single-state RNA recovery. EVO-IF reaches 86.81% recovery on Das Benchmark, exceeding NA-MPNN by 18.94 percentage points (Figure 2a). More importantly, its margin over NA-MPNN remains positive for every target in the panel, showing that the overall improvement is not concentrated in a few favorable cases (Figure 2b). The magnitude is nevertheless target-dependent, with the largest improvements on 1XPE, 1Q9A, and 1F27 (Figure 2b). Appendix Table 1 reports the full aggregate and per-structure results.

Single-state structural self-consistency. We assess compatibility with the target fold by predicting secondary and tertiary structures from designed RNA sequences (Figure 3). Under the same reproduced evaluation pipeline, EVO-IF improves all four mean self-consistency metrics over NA-MPNN: scMCC increases from 0.450 to 0.564, scTM-score from 0.409 to 0.487, and scGDT_TS from 0.443 to 0.516, while scRMSD decreases from 3.30 to 2.87 Å. These results complement native sequence recovery with evidence of compatibility with the target secondary and tertiary structures.

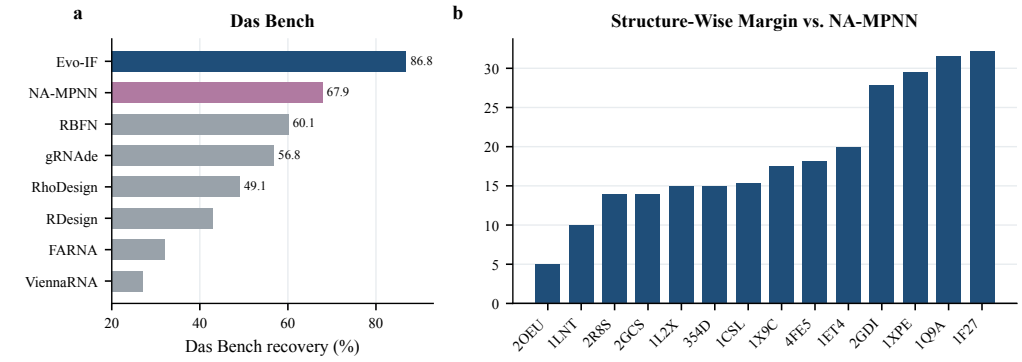

Figure 2: **RNA inverse-folding performance. a**, Das Benchmark recovery for NA inverse-folding against baselines. **b**, structure-wise margin over NA-MPNN.

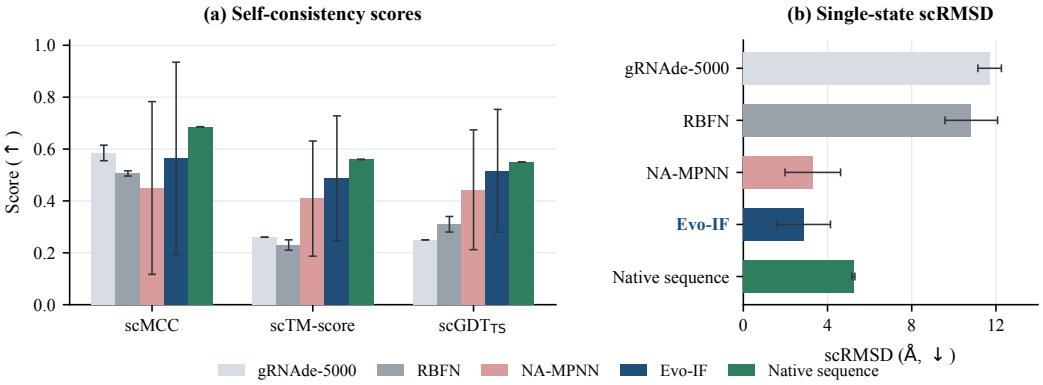

Figure 3: **Single-state RNA self-consistency. a**, secondary- and tertiary-structure consistency scores (higher is better). **b**, scRMSD (lower is better). Bars show means with structure-wise standard deviations. The gRNAde AR model with maximum training length 5,000 and RBFN are prior-reported references; NA-MPNN and EVO-IF use the reproduced evaluation pipeline. Native-sequence predictions provide a reference. Full numerical results appear in Appendix Table 2.

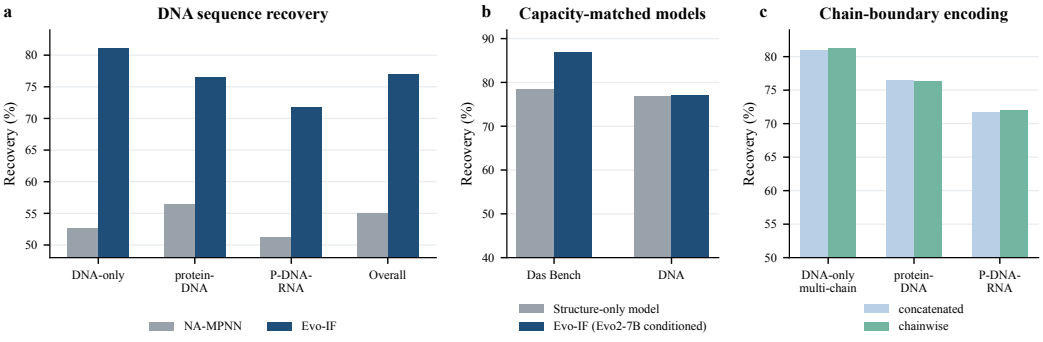

Figure 4: **Controlled model comparisons. a**, DNA benchmark comparisons. **b**, capacity-matched training comparison. **c**, concatenated versus chainwise EVO2 inputs. Protocols and tables are in Appendix Tables 6, 12, and 13.

DNA controlled comparisons. EVO-IF consistently outperforms NA-MPNN across DNA-only and protein-containing DNA contexts, with recovery gains ranging from 19.98 to 28.34 percentage points (Figure 4a). The advantage extends to ternary protein–DNA–RNA assemblies, where recovery reaches 71.80%, a 20.60-point gain over NA-MPNN. In the capacity-matched training comparison, EVO-IF exceeds the structure-only model by 8.49 recovery points on Das Benchmark and 9.68 points on DNA (Figure 4b). These gains measure the benefit of the full genome–geometry adaptation procedure. Fixed-weight interventions probe branch use during decoding (Section 5). The resullt indicates that the effect of genome-language conditioning is not uniform across modalities. Concatenated and chainwise EVO2 inputs differ by at most 0.33 points (Figure 4c), placing the cross-modal gains well above the variation due to chain serialization. The detailed experimental results are given in Appendix Tables 6, 12, and 13.

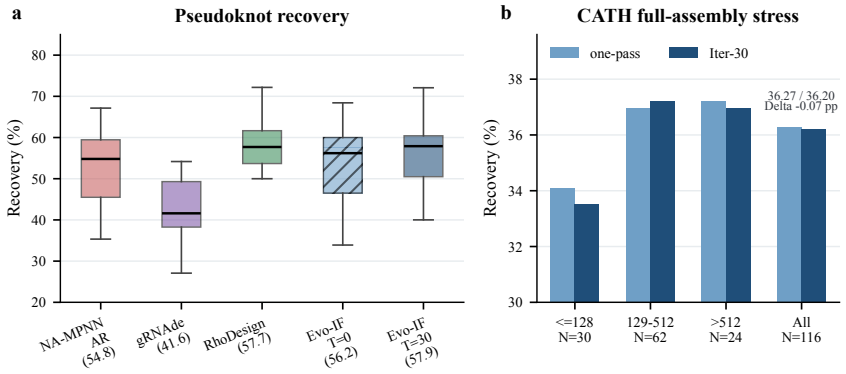

Figure 5: **Broader validation. a**, pseudoknot challenge performance. **b**, one-pass versus Iter-30 protein recovery by full-assembly size on CATH. Protocols and results appear in Appendix A.3 and Tables 8 and 11.

Broader validation and failure modes. On the pseudoknot challenge, EVO-IF remains competitive with the leading baselines but does not uniformly dominate them. EVO-IF reaches 56.2% median recovery with one-pass decoding and 57.9% with Iter-30, compared with 57.7% for RhoDesign and 54.8% for NA-MPNN (Figure 5a).

This sensitivity contrasts with the protein-retention test, in which the adapted decoder remains close to the base model (Figure 5b). Together, these results suggest that EVO-IF transfers across task contexts more reliably than across substantial perturbations to the conditioning backbone. Detailed values are reported in Appendix Tables 8, 7, and 10, respectively.

4.3 ABLATION STUDY

Figure 6 traces the training and decoding milestones from the structure-only model to the EVO-IF checkpoint.

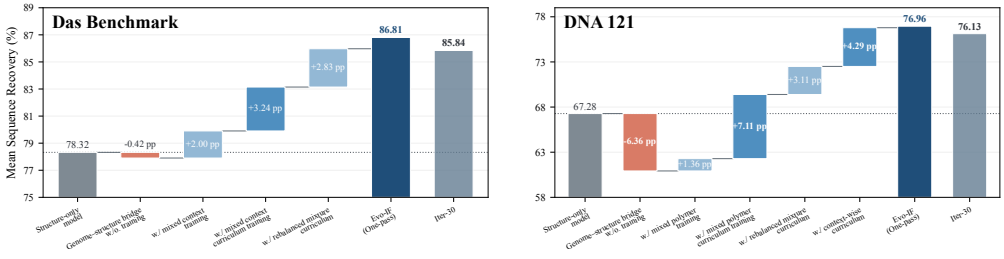

Figure 6: **Training and decoding trajectories on Das Benchmark and DNA-121.** Adjacent labels show changes in mean sequence recovery along the curriculum from the structure-only reference to EVO-IF. The milestone tables appear in Appendix Tables 14 and 15.

Gains along the training curriculum. Recovery increases through mixed-context training and subsequent curriculum stages on both benchmarks. Initial bridge insertion lowers recovery, with a larger decrease on DNA. The largest subsequent DNA improvement occurs during the mixed-polymer curriculum, followed by further gains with mixture and context-wise rebalancing. One-pass decoding gives the strongest endpoint on both benchmarks, while Iter-30 provides no additional gain for the selected checkpoint. Each training stage continues the preceding model under a new configuration. The exact values and adjacent changes are listed in Appendix Tables 14 and 15.

5 Discussion and Future Work

Alignment and adaptation. EVO-IF learns an interface between genome activations and geometric residue states through backbone-conditioned sequence reconstruction. The training comparison shows an 8.49-point recovery gain on Das Benchmark and a 9.68-point gain on DNA (Table 12). At the intermediate checkpoint, removing the coupled branch lowers one-pass recovery by 5.02 points on Das and 4.24 points on DNA, while permuting available bases has smaller effects under Iter-30 (Table 18). The training comparison and fixed-weight interventions demonstrate improved recovery after genome-geometry adaptation and active use of the coupled path during decoding.

Molecular context in shared-decoder training. The curriculum trajectory highlights the role of molecular context in adaptation (Figure 6). Recovery increases as training expands and rebalances the polymer contexts, with the largest DNA increment occurring during the mixed-polymer curriculum. Chain serialization has a smaller effect, changing recovery by at most 0.33 points (Table 13). The successive training stages characterize cumulative adaptation, making data composition and the update schedule explicit parts of the method and its reproducible specification.

One-pass decoding as the benchmark operating point. For the EVO-IF checkpoint, one-pass decoding matches or exceeds Iter-30 on every Das target, with a 0.96-point aggregate recovery advantage and 30.7× lower measured latency than the same model's 30-step decoder (Tables 5 and 20). This supports one-pass decoding as the default for the reported benchmarks. The temporal continuation experiment shows a 2.03-point iterative gain at its starting checkpoint (Table 19), demonstrating that the value of refinement depends on the checkpoint and evaluation setting.

Recovery and structural compatibility remain complementary. Single-state self-consistency improves across all four mean metrics relative to NA-MPNN under the reproduced protocol (Figure 3). Multi-state recovery increases from 44.1% to 47.1% with one to three input conformers, while structural consistency changes only modestly (Table 3). Pseudoknot recovery and OpenKnot Score rankings also differ (Tables 8 and 9). Sequence recovery and predicted fold compatibility capture complementary aspects of design quality (Hu et al., 2026; Si et al., 2026). Extending the framework to uncertain backbones and incorporating experimental feedback would connect this computational evaluation more directly to the structures and functions of designed molecules. Appendix E defines the scope of the present evidence.

6 Conclusion

We presented EVO-IF, a framework that connects genome-scale language modeling with three-dimensional nucleic-acid design. A compact, context-aware alignment bridge couples a frozen genome encoder to structural diffusion, supporting RNA and DNA sequence design across isolated molecules and protein-nucleic-acid assemblies. EVO-IF leads the Das comparison at 86.81% native sequence recovery and reaches 76.96% across the evaluated DNA assemblies, improving on the structure-only reference by 8.49 and 9.68 percentage points, respectively. For the selected checkpoint, one-pass decoding delivers the strongest benchmark recovery among the tested decoding settings while reducing measured latency by 30.7× relative to 30-step refinement. These gains establish EVO-IF as an effective interface between sequence foundation models and the geometric constraints of biomolecular design. Looking ahead, training on more diverse structures and sequence contexts, incorporating explicit design constraints, and integrating experimental feedback offer a path toward greater control over the structure and function of designed nucleic acids.

AI USE STATEMENT

In this work, we used generative AI tools for text polishing. We have not used generative AI tools to develop the core research ideas or algorithms, or to clean the data. This work does not include wet-lab experiments. Additionally, we used generative AI tools to assist with giving figure color recipes. 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

Nucleic-acid sequence design is dual-use. The present work concerns reconstruction from known macromolecular structures and does not report newly designed biological agents, wet-lab synthesis, pathogen optimization, or human-subject data.

Nevertheless, the same technical direction could lower barriers to designing functional nucleic acids. Any future release should retain sequence-level screening, access controls appropriate to capability, and review of generated candidates before synthesis.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model and optimization objective. Appendices B.1 and B.2 document the architecture, preprocessing, training configuration, and decoding procedure. Appendix A provides evaluation protocols and detailed numerical results. We will submit the source code as an archive as supplementary material, with author-identifying information removed from the code and accompanying files. Other assets will be publicly released after the paper is accepted.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/abs/2308.12966.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A. Gonzalez, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652:1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

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*, pp. 11315–11325, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin P. Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4055–4075. PMLR, 2023.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, Irwin King, and Yu Li. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022. URL https://arxiv.org/abs/2204.00300.

Rhiju Das, John Karanicolas, and David Baker. Atomic accuracy in predicting and designing noncanonical RNA structure. *Nature Methods*, 7(4):291–294, 2010. doi: 10.1038/nmeth.1433. URL https://doi.org/10.1038/nmeth.1433.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

Laura R. Ganser, Megan L. Kelly, Daniel Herschlag, and Hashim M. Al-Hashimi. The roles of structural dynamics in the cellular functions of RNAs. *Nature Reviews Molecular Cell Biology*, 20 (8):474–489, 2019.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pp. 8946–8970. PMLR, 2022.

Tianmeng Hu, Yongzheng Cui, Biao Luo, and Ke Li. RIDER: 3d RNA inverse design with reinforcement learning-guided diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lDyS4Qg5Ww.

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.

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

A. Kubaney, A. Favor, L. McHugh, R. Mitra, R. Pecoraro, J. Dauparas, C. Glasscock, and D. Baker. Rna sequence design and protein-dna specificity prediction with na-mpnn. *bioRxiv*, 2025. doi: 10.1101/2025.10.03.679414. URL https://doi.org/10.1101/2025.10.03.679414. Preprint.

J. K. Leman, B. D. Weitzner, S. M. Lewis, et al. Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nature Methods*, 17:665–680, 2020. doi: 10.1038/s41592-020-0848-2.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language–image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742. PMLR, 2023.

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.

Runze Ma, Zhongyue Zhang, Zichen Wang, Will Hua, Jiahua Rao, Zhuomin Zhou, and Shuangjia Zheng. Riboflow: Conditional de novo RNA co-design via synergistic flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=brV7U7svgS.

R. J. Penić, T. Vlašić, R. G. Huber, et al. RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks. *Nature Communications*, 16:5671, 2025. doi: 10.1038/s41467-025-60872-5. URL https://doi.org/10.1038/s41467-025-60872-5.

Jiezhong Qiu, Junde Xu, Jie Hu, Hanqun Cao, Liya Hou, Zijun Gao, Xinyi Zhou, Anni Li, Xiujuan Li, Bin Cui, Fei Yang, Shuang Peng, Ning Sun, Fangyu Wang, Aimin Pan, Jie Tang, Jieping Ye, Junyang Lin, Jin Tang, Xingxu Huang, Pheng Ann Heng, and Guangyong Chen. Instructplm: Aligning protein language models to follow protein structure instructions. *bioRxiv*, 2024. doi: 10.1101/2024.04.17.589642. URL https://doi.org/10.1101/2024.04.17.589642.

T. Shen, Z. Hu, S. Sun, et al. Accurate rna 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, 21:2287–2298, 2024. doi: 10.1038/s41592-024-02487-0.

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.

Huu Binh Ta, Michael Cardei, Alvaro Velasquez, and Ferdinando Fioretto. Search-augmented masked diffusion models for constrained generation. *arXiv preprint arXiv:2602.02727*, 2026. URL https://arxiv.org/abs/2602.02727.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RemfXx7ebP.

Ignacio Tinoco, Jr. and Carlos Bustamante. How rna folds. *Journal of Molecular Biology*, 293(2): 271–281, 1999. ISSN 0022-2836. doi: https://doi.org/10.1006/jmbi.1999.3001.

Rubo Wang, Xingyu Gao, and Peilin Zhao. Geometric algebra-enhanced bayesian flow network for RNA inverse design. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=x0KcjteNds.

Shengfan Wang, Jun Wang, Xiaojian Liu, Weimin Zhu, Jiaqi Huang, Yiming Xue, Junchi Yan, Hong-Bin Shen, Jie Song, and Xiaoyong Pan. Structure-alignment-driven cross-graph modeling for functional RNA design. *Nature Computational Science*, 6:826–841, 2026. doi: 10.1038/s43588-026-01029-2. URL https://doi.org/10.1038/s43588-026-01029-2.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A multimodal diffusion protein language model. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=5z9GjHgerY.

Hannah K. Wayment-Steele, Wipapat Kladwang, Alexandra I. Strom, Jeehyung Lee, Adrien Treuille, Alex Becka, Eterna Participants, and Rhiju Das. RNA secondary structure packages evaluated and improved by high-throughput experiments. *Nature Methods*, 19(10):1234–1242, 2022.

Heng Yang, Renzhi Chen, and Ke Li. Bridging sequence-structure alignment in RNA foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21929–21937, 2025. doi: 10.1609/aaai.v39i20.35500. URL https://ojs.aaai.org/index.php/AAAI/article/view/35500.

A Additional Experimental Results

This section expands the aggregate plots in the main text. We report the complete values for reproducibility and examine consistency across individual targets, behavior under external structure predictors, robustness to input and context shifts, and matched training controls.

A.1 Evaluation Panels

Sequence recovery on Das Benchmark Table 1 reports the Das Benchmark recovery comparison. Literature baseline values are taken from Das et al. (2010); Joshi et al. (2025); Wang et al. (2025a); NA-MPNN and EVO-IF are included as additional comparison columns in the same table. EVO-IF is the selected one-pass checkpoint.

Table 1: Complete per-structure results on the single-state Das RNA benchmark. Bold marks the best value in each row; underlining marks the second-best distinct value, including ties.

PDB ID	Description	ViennaRNA	FARNA	RDesign	gRNAde	RhoDesign	RBFN	NA-MPNN	EVO-IF
1CSL	RRE high affinity site	0.25	0.20	0.4455	**0.5719**	0.4643	0.4832	0.3462	0.5000
1ET4	Vitamin B12 binding RNA aptamer	0.25	0.34	0.3929	0.6250	0.5428	0.4482	0.6857	**0.8857**
1F27	Biotin-binding RNA pseudoknot	0.30	0.36	0.3013	0.3437	0.4211	0.5208	0.6786	**1.0000**
1L2X	Viral RNA pseudoknot	0.24	0.45	0.3727	0.4721	0.2593	0.5833	0.8500	**1.0000**
1LNT	RNA internal loop of SRP	0.33	0.27	0.5556	0.5843	0.3636	0.6375	0.6500	**0.7500**
1Q9A	Sarcin/ricin domain from E.coli 23S rRNA	0.27	0.40	0.4417	0.5044	0.8148	0.8588	0.6842	**1.0000**
4FE5	Guanine riboswitch aptamer	0.29	0.28	0.4112	0.5300	0.8209	0.5326	0.8182	**1.0000**
1X9C	All-RNA hairpin ribozyme	0.26	0.31	0.3967	0.5000	0.3833	0.5187	0.7895	**0.9649**
1XPE	HIV-1 B RNA dimerization initiation site	0.27	0.24	0.3834	0.7037	0.6957	0.6522	0.7045	**1.0000**
2GCS	Pre-cleavage state of glmS ribozyme	0.25	0.26	0.4518	0.5078	0.2049	0.4990	0.5664	**0.7063**
2GDI	Thiamine pyrophosphate-specific riboswitch	0.25	0.38	0.3523	0.6500	0.2436	0.6042	0.6835	**0.9620**
2OEU	Junctionless hairpin ribozyme	0.23	0.30	0.5000	**0.9519**	0.1905	0.5580	0.6667	0.7167
2R8S	Tetrahymena ribozyme P4-P6 domain	0.27	0.36	0.5641	0.5689	0.6415	0.7172	0.7278	**0.8671**
354D	Loop E from E. coli 5S rRNA	0.28	0.35	0.4458	0.4410	**0.8261**	0.8031	0.6500	0.8000
	Overall recovery:	0.27	0.32	0.4296	0.5682	0.4909	0.6012	0.6787	**0.8681**

The per-structure results show that the 86.81% aggregate is broad rather than outlier-driven. At the aggregate level, EVO-IF exceeds NA-MPNN by 18.94 percentage points. EVO-IF improves over NA-MPNN on every target and ranks first among the listed methods on 11 of 14 structures, including both pseudoknots 1F27 and 1L2X. At the same time, 1CSL, 2OEU, and 354D retain a stronger alternative baseline. These exceptions are useful because they span different structural classes and therefore argue against treating a single decoder as uniformly optimal for all RNA motifs.

Single-state RNA self-consistency evaluations For single-state self-consistency metrics (Table 2), the gRNAde and RBFN rows are taken from the published reports of Joshi et al. (2025); Wang et al. (2025a), while NA-MPNN and EVO-IF are evaluated under our reproduced protocol. Native-sequence, random-sequence, and ViennaRNA reference rows are also prior-reported values. Because prior work does not provide all per-design folded structures and exact alignment metadata, copied 3D self-consistency scores are included as historical references. Controlled 3D comparisons are made between NA-MPNN and EVO-IF under the same evaluation pipeline.

Under the reproduced rows, EVO-IF improves each reported self-consistency metric relative to NA-MPNN. The agreement across all four metrics connects the recovery gains to improved secondary- and tertiary-structure compatibility under a shared evaluation pipeline. Recovery measures agreement with the native sequence, while self-consistency assesses the predicted fold against the target geometry.

Multi-state RNA design Table 3 reports the multi-state RNA design results across 100 test samples, following the evaluation protocol in Wang et al. (2025a). The column "Max. states" denotes the maximum number of conformers provided as input for each RNA group. For EVO-IF, aligned conformers are evaluated separately with shared design positions, and their per-residue probabilities are averaged before argmax sequence prediction. Increasing K therefore provides additional structural observations of the same sequence constraint. For EVO-IF, recovery rises modestly from 0.441 at $K{=}1$ to 0.471 at $K{=}3$, while the external fold-consistency metrics remain within narrow ranges. At $K{=}3$, EVO-IF gives the highest reported scTM-score and scGDT_TS, while RBFN leads recovery. Relative to the reproduced NA-MPNN comparison, recovery is lower at each K, with a small tertiary-consistency advantage at $K{=}3$.

Table 2: Self-consistency metrics on the single-state gRNAde released benchmark. Values are reported as mean ± structure-wise standard deviation across the benchmark RNAs.

Model	Max. train length	Self-consistency metrics			
		2D-EternaFold	3D-RhoFold		
		scMCC (↑)	scRMSD (↓)	scTM-score (↑)	scGDT_TS (↑)
gRNAde (AR)	500	0.624±0.07	13.01±1.18	0.21±0.0	0.22±0.0
gRNAde (AR)	1000	0.648±0.01	13.10±0.58	0.20±0.0	0.21±0.0
gRNAde (AR)	2500	0.633±0.03	11.76±0.91	0.27±0.0	0.27±0.0
gRNAde (AR)	5000	0.585±0.03	11.70±0.56	0.26±0.0	0.25±0.0
gRNAde (NAR)	5000	0.473±0.02	13.04±0.88	0.23±0.0	0.22±0.0
RBFN	5000	0.506±0.01	10.83±1.25	0.23±0.02	0.31±0.03
NA-MPNN	5000	0.450±0.333	3.30±1.32	0.409±0.222	0.443±0.231
EVO-IF	5000	0.564±0.371	**2.87±1.27**	**0.487±0.241**	**0.516±0.237**
Groundtruth sequence prediction:		0.686±0.00	5.23±0.07	0.56±0.0	0.55±0.0
Random sequence prediction:		0.012±0.00	24.40±0.34	0.04±0.0	0.02±0.0
ViennaRNA 2D-only:		0.611±0.00	20.34±0.10	0.07±0.0	0.07±0.0

Table 3: Multi-state RNA design results following the `structsim v1` evaluation protocol in Wang et al. (2025a). Metrics are reported as mean ± group-wise standard deviation across 100 test samples. The results of gRNAde and RBFN are taken from Joshi et al. (2025); Wang et al. (2025a). NA-MPNN and EVO-IF are evaluated under the reproduced evaluation pipeline.

Model	Max. states	Native seq. recovery (↑)	Self-consistency metrics			
			2D-EternaFold	3D-RhoFold		
			scMCC (↑)	scRMSD (↓)	scTM-score (↑)	scGDT_TS (↑)
gRNAde (AR)	1	0.481±0.00	0.573±0.04	21.83±0.53	0.12±0.0	0.15±0.0
gRNAde (AR, DS)	3	0.531±0.00	0.573±0.03	22.19±0.28	0.12±0.0	0.15±0.0
gRNAde (AR, DSS)	3	0.540±0.03	**0.574±0.03**	22.20±0.43	0.12±0.0	0.15±0.0
gRNAde (AR, DS)	5	0.510±0.00	0.514±0.00	21.80±0.08	0.12±0.0	0.14±0.0
gRNAde (NAR)	1	0.489±0.00	0.372±0.03	24.18±0.63	0.09±0.0	0.12±0.0
gRNAde (NAR, DS)	3	0.506±0.01	0.346±0.02	24.06±0.43	0.08±0.0	0.11±0.0
gRNAde (NAR, DSS)	3	0.520±0.02	0.352±0.03	24.18±0.55	0.09±0.0	0.12±0.0
gRNAde (NAR, DS)	5	0.517±0.01	0.339±0.01	24.16±0.75	0.08±0.0	0.10±0.0
RBFN	1	0.497±0.01	0.528±0.02	21.68±0.63	0.14±0.01	0.17±0.01
RBFN	2	0.536±0.01	0.515±0.04	22.33±0.98	0.12±0.02	0.14±0.01
RBFN	3	**0.548±0.02**	0.461±0.05	22.58±0.95	0.11±0.03	0.13±0.01
NA-MPNN	1	0.452±0.200	0.322±0.375	2.68±1.36	0.228±0.096	0.400±0.186
NA-MPNN	2	0.479±0.213	0.246±0.359	2.65±1.47	0.230±0.103	0.397±0.188
NA-MPNN	3	0.482±0.206	0.285±0.356	2.64±1.35	0.229±0.100	0.406±0.187
EVO-IF	1	0.441±0.194	0.280±0.357	2.64±1.40	0.225±0.103	0.394±0.187
EVO-IF	2	0.460±0.222	0.278±0.359	2.67±1.37	0.224±0.105	0.393±0.193
EVO-IF	3	0.471±0.214	0.277±0.386	**2.63±1.30**	**0.233±0.103**	**0.411±0.179**
Groundtruth sequence prediction baseline:			0.525±0.00	17.52±0.32	0.25±0.0	0.29±0.0
Random sequence prediction baseline:			0.013±0.00	31.00±0.20	0.03±0.0	0.02±0.0
ViennaRNA 2D-only baseline:			0.470±0.00	29.10±0.00	0.05±0.0	0.05±0.0

Table 4 summarizes the evaluation panels used for the released checkpoint.

Table 4: Evaluation panels. One DNA manifest entry has no designable DNA and is not scored.

Panel	Scored Candidates	Published after checkpoint	Role
Das Benchmark	14	No	benchmark
DNA assemblies	121	No	benchmark
Post-checkpoint temporal	12	Yes	temporal transfer

A.2 ONE-PASS VERSUS 30-STEP ITERATIVE RE-WALK

Table 5: Paired Das comparisons on the same 14 structures. Structure-only denotes the NAIAD reference. SR is mean recovery (%); Δ and wins/ties/losses compare the method with the reference named in each row. All paired values use four-decimal recovery precision, as in Table 1; mean differences are computed before percentage rounding.

Method	Reference	Method SR	Ref. SR	Δ (pp)	Wins	Ties	Losses
EVO-IF one-pass	Structure-only	86.81	78.32	+8.49	11	1	2
EVO-IF Iter-30	Structure-only	85.84	78.32	+7.53	9	2	3
EVO-IF Iter-30	EVO-IF one-pass	85.84	86.81	−0.96	0	8	6

Relative to the 78.32% structure-only reference, EVO-IF reaches 86.81% in one pass: an 8.49-point mean gain with 11 wins, one tie, and two losses. Iter-30 reaches 85.84%, a 7.53-point gain with nine wins, two ties, and three losses against the same reference. Comparing decoding budgets within EVO-IF, Iter-30 yields eight ties and six losses relative to one-pass, for a mean change of −0.96 points. The structure-only comparison establishes the system-level gain, while the paired decoding comparison identifies one-pass as the stronger operating point on this panel.

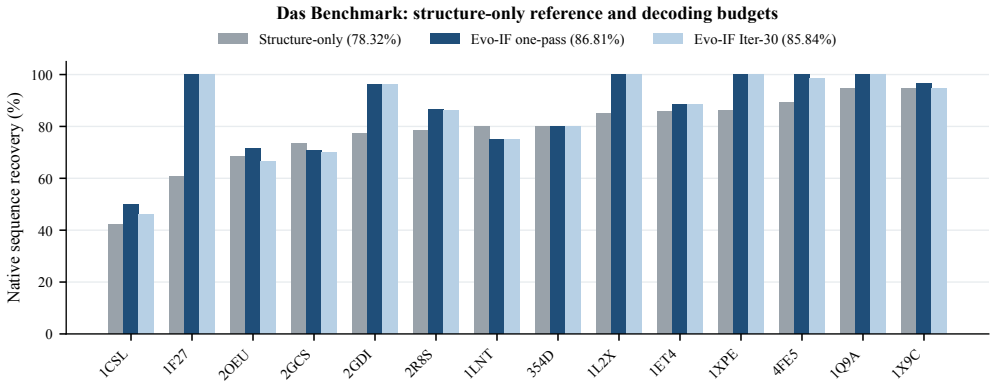

Figure 7: Das recovery for the structure-only reference and EVO-IF with one-pass or Iter-30 decoding. Targets are ordered by structure-only recovery; legend values are means over all 14 structures. One-pass values follow Table 1; paired gains and counts appear in Table 5.

A.3 Additional Evaluations on Nucleic Acid

DNA design benchmark. Table 6 reports the DNA sequence recovery test. The comparison includes NA-MPNN and Evo-IF under the same split. This DNA-specific test split contains 28 DNA-only, 78 protein–DNA, and 15 protein–DNA–RNA assemblies. The Evo-IF checkpoint is evaluated on the same 28, 78, and 15 structures in the same context buckets, respectively.

Table 6: Performance on DNA tasks for NA-MPNN and Evo-IF. All rows are evaluated on the complete 121-assembly split. Recovery is computed on DNA positions only over 121 held-out DNA-containing biological assemblies.

Method	N	DNA-only ($N = 28$)	Protein–DNA ($N = 78$)	Protein–DNA–RNA ($N = 15$)	Overall
NA-MPNN	121	52.7 ± 20.5	56.5 ± 18.2	51.2 ± 15.7	55.0 ± 18.6
Evo-IF	121	81.04 ± 17.09	76.48 ± 17.82	71.80 ± 16.95	76.96 ± 17.75

Evo-IF exceeds NA-MPNN in all three DNA contexts, with gains of 28.34, 19.98, and 20.60 percentage points, respectively. The decline from 81.04% on DNA-only structures to 71.80% on protein–DNA–RNA assemblies shows that heterogeneous interfaces remain the more difficult regime. Importantly, the advantage persists in that hardest bucket rather than arising only from isolated DNA chains.

Transfer to predicted RNA backbones To test sensitivity to input quality, we replace each resolved Das backbone with a RhoFold prediction while retaining the same target and evaluation pipeline. Table 7 reports the backbone-replacement comparison, with its own resolved-input reference for each model. The structure-only row is the control for this experiment. Table 12 reports the principal training comparison.

Table 7: Transfer from experimentally resolved to RhoFold-predicted input backbones on the 14-structure Das panel. Values are structure-wise means; recovery is reported in percent.

Model	Input backbone	SR ↑	scMCC ↑	scRMSD ↓	scTM ↑	scGDT_TS ↑
Structure-only matched	Resolved mmCIF	86.28	0.548	1.969	0.337	0.571
Structure-only matched	RhoFold-predicted	26.11	0.040	2.524	0.172	0.334
Evo-IF	Resolved mmCIF	86.28	0.548	1.969	0.337	0.571
Evo-IF	RhoFold-predicted	26.07	−0.009	2.628	0.176	0.341

Both models lose roughly 60 recovery points on predicted inputs, and scMCC falls to approximately zero. The nearly identical degradation indicates that the bottleneck is the input geometry rather than a model-specific decoding choice; genome-language conditioning does not compensate for a backbone that has moved far from the experimentally resolved target. This experiment is consequently a boundary test for practical deployment, not evidence that predicted and resolved structures are interchangeable inputs.

Pseudoknot challenge decoding sequence recovery Evo-IF (T=30) has the highest median in this comparison. A separate matched run compares one-pass and Iter-30 decoding on the same 10 structures. Table 8 gives the aggregate values, while Figures 5a and 8 show their distributions and per-structure breakdown. Evo-IF (T=30) reaches 57.9%, followed by RhoDesign at 57.7%, Evo-IF (T=0) at 56.2%, and NA-MPNN AR at 54.8%. gRNAde sits noticeably lower than others at 41.6%. Evo-IF (T=0) and Evo-IF (T=30) track each other within 1.7 points of median recovery. Table 8 and Figure 5a summarize the recovery distribution under this matched protocol.

Table 8: Median sequence recovery for the matched pseudoknot decoding comparison (per-PDB median across 10 designs). $T = 0$ denotes one-pass decoding without iterative refinement. All methods use the same RCSB CIF featurisation (`parse_protein=0`).

Method	Median recovery (%)
RhoDesign	57.7
gRNAde	41.6
NA-MPNN	54.8
Evo-IF (T=0)	56.2
Evo-IF (T=30)	**57.9**

Iter-30 improves on one-pass by 1.7 points and exceeds the NA-MPNN aggregate by 3.1 points in this run. The per-structure comparison is less uniform. Iter-30 exceeds NA-MPNN on eight of ten structures, but trails it on 3Q3Z and 7KGA.

The per-structure breakdown (Figure 8) shows that the relative method ordering varies across puzzles, with Evo-IF (T=30) outperforming NA-MPNN on 4PLX, 4ZNP, 2M8K, and 7QR4, while trailing on 3Q3Z (0.55 vs. 0.66) and 7KGA (0.46 vs. 0.51).

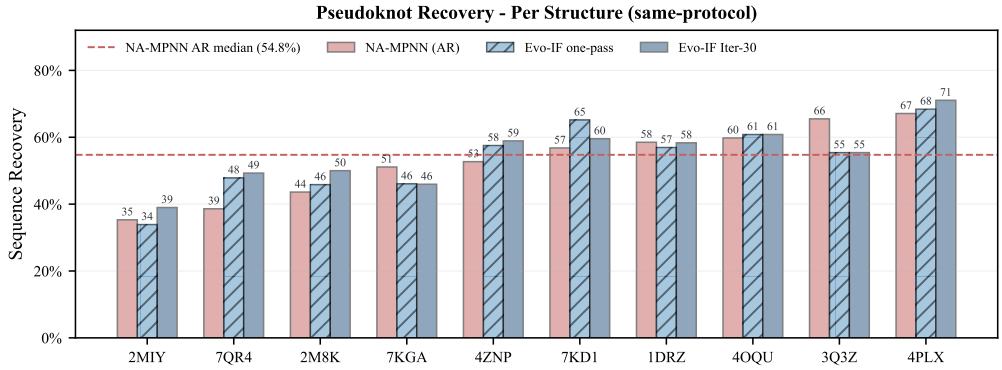

Figure 8: **Per-structure pseudoknot recovery.** Bars show the per-PDB median of 10 designs. Dashed line marks the NA-MPNN AR median (54.8%) for reference.

OpenKnot Score (OKS) Comparison Each designed sequence is folded by RibonanzaNet into a predicted reactivity profile and scored against the DSSR ground-truth dot-bracket of the native crystal. gRNAde leads the OKS distribution at 86.4, followed among model rows by Evo-IF (T=30) at 76.6, Evo-IF (T=0) at 75.9, RhoDesign at 75.3, and NA-MPNN at 71.4. For reference, the WT baseline achieves 73.8. Table 9 and Figure 9 summarize the OKS comparison.

Table 9: Median OKS on the 10-structure pseudoknot test set, n=100 designs per generated-sequence method .

Method	Median OKS
WT (native, RibonanzaNet)	73.8
NA-MPNN	71.4
gRNAde	86.4
RhoDesign	75.3
Evo-IF (T=0)	75.9
Evo-IF (T=30)	76.6

The gRNAde protocol additionally uses 2D structural information and OKS-based rejection sampling during design. These differences in conditioning and candidate selection accompany the OKS comparison.

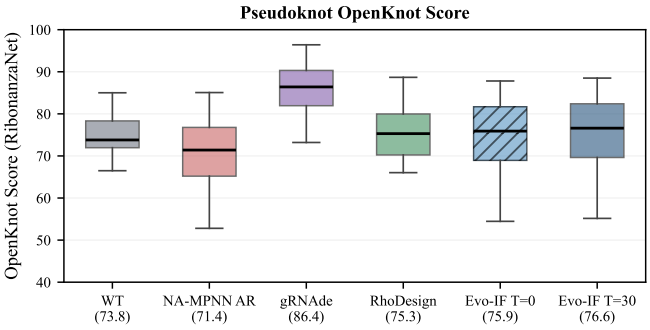

Figure 9: **Pseudoknot OpenKnot Score distributions, same-protocol evaluation.** Five sequence design methods over n=100 designs each, plus the WT reference (n=9). gRNAde is conspicuously high because its training pipeline filters candidates by OKS; the other listed methods train only for sequence recovery.

A.4 Protein Stress Tests

Standard CATH base/adapted protein retention The standard comparison isolates each nominated CATH 4.2 short chain, designs every residue, and supplies no assembly partner. The base and adapted structural decoders use the same featurizer, one-pass inference, and amino-acid vocabulary. Protein inputs bypass Evo2. Table 10 reports the same recovery and PPL convention across both decoders.

Table 10: Matched CATH 4.2 chain-level protein retention. SR is recovery (%). Short-chain PPL includes mean and median. All-chain PPL reports the mean.

Model	Short N	Short SR	Short PPL mean ↓	Short PPL median ↓	All-chain SR	All-chain PPL ↓
Base structural decoder	116	30.87	35.190	13.088	31.55	35.068
Adapted structural decoder	116	30.14	49.905	16.639	29.65	49.768

The 0.73-point recovery decrease after adaptation is small relative to the absolute protein recovery, supporting retention of most discrete sequence decisions. Perplexity is less stable, however: both its mean and median increase for the adapted decoder. Because protein-only inputs bypass Evo2, this comparison probes what nucleic-acid training does to the shared structural decoder rather than whether genome-language features transfer directly to amino acids.

Protein full-assembly stress test Table 11 compares decoding budgets across assembly sizes, retaining the full parsed molecular context.

Table 11: CATH 4.2 short chain full-assembly protein recovery by parsed assembly size. Values are unweighted means. Δ is Iter-30 minus one-pass, calculated from the means.

Assembly positions	N	One-pass	Iter-30	Δ
≤ 128	30	34.09	33.52	-0.57
129–512	62	36.97	37.21	$+0.24$
> 512	24	37.22	36.96	-0.26
All assemblies	116	36.27	36.20	-0.07

A.5 Training controls and curriculum details

The following experiments compare trained configurations, nucleic-acid chain serialization, and successive curriculum stages. The published-baseline comparison measures complete-system performance, while these analyses characterize specific training and implementation choices.

Capacity-matched attribution. Evo-IF raises RNA recovery from 78.32% to 86.81% and DNA recovery from 67.28% to 76.96%: gains of 8.49 and 9.68 percentage points across the two modalities.

Table 12: Capacity-matched training comparison shown in Figure 4b. Entries are native sequence recovery (%). Δ is EVO-IF minus the structure-only reference.

Model	Das Benchmark	DNA
Structure-only inverse folding model	78.32	67.28
EVO-IF (EVO2-7B conditioned)	**86.81**	**76.96**
Δ	+8.49	+9.68

EVO-IF exceeds the structure-only model by 8.49 recovery points on Das Benchmark and 9.68 points on DNA. These gains extend across both RNA and DNA under the reported training protocols. Inference-time branch dependence is evaluated separately through the fixed-weight interventions in Table 18.

Chain-boundary encoding. RNA U-to-T conversion is a one-to-one alphabet mapping required by the frozen tokenizer and preserves base identity. The scientifically relevant preprocessing question is chain boundaries. We train two matched models from the same initialization: one concatenates nucleic-acid chains as in the selected configuration, while the other encodes each chain independently and concatenates the resulting activations in assembly order. Training order, parameter count, masks, optimizer, and update budget are identical. Table 13 reports the resulting native recoveries; single-chain structures are excluded from the boundary-specific aggregate.

Table 13: Matched chain-boundary ablation; entries are native recovery (%).

EVO2 input	Temporal multi-chain	DNA-only multi-chain	Protein–DNA	Protein–DNA–RNA
Concatenate chains	53.99	80.99	76.44	71.75
Encode chains independently	54.33	81.31	76.34	72.08

There is no consistent winner between the two serialization schemes: independent encoding is slightly higher in three buckets, while concatenation is slightly higher for protein–DNA. All differences are at most 0.34 points, far smaller than the benchmark-level changes above. This control supports treating chain serialization as an implementation choice rather than a primary source of accuracy.

Curriculum ladder. Tables 14 and 15 report the successive training and decoding milestones.

Table 14: Das Benchmark ablation milestones for the matched-control ladder. Mean SR is native sequence recovery (%); Δ reports the change from the preceding milestone for floating bars only.

	Native seq. recovery (↑)	Step Δ
Structure-only model	78.32	–
Genome–structure bridge w./o. training	77.90	−0.42
w./ mixed-context training	79.90	+2.00
w./ mixed-context curriculum training	83.14	+3.24
w./ rebalanced-mixture curriculum	85.97	+2.83
EVO-IF (One-pass)	**86.81**	–
Iter-30	85.84	−0.96

On Das Benchmark, attaching the bridge before task-specific training initially lowers recovery, after which each curriculum stage recovers and then exceeds the structure-only reference. The largest increments occur during the mixed-context curriculum and mixture rebalancing.

Table 15: DNA-121 training and decoding milestones. Mean SR is native sequence recovery on DNA positions (%); Δ reports the change from the preceding milestone for floating bars.

	Native seq. recovery (↑)	Step Δ
Structure-only model	67.28	–
Genome–structure bridge w./o. training	60.92	-6.36
w./ mixed-polymer training	62.29	$+1.36$
w./ mixed-polymer curriculum training	69.39	$+7.11$
w./ rebalanced-mixture curriculum	72.50	$+3.11$
w./ context-wise curriculum	76.79	$+4.29$
EVO-IF (One-pass)	**76.96**	–
Iter-30	76.13	-0.83

The DNA ladder is even more curriculum-sensitive: mixed-polymer curriculum supplies the largest individual gain, and context-wise curriculum adds a further 4.29 points after mixture rebalancing. In Table 15, the successive stages describe cumulative adaptation under changing training contexts. Across both ladders, Iter-30 finishes below the selected one-pass checkpoint, consistent with the paired Das analysis in Table 5.

B Architecture and Implementation Details

B.1 Configurations

EVO-IF checkpoint comprises three components, the frozen EVO2-7B base model, a pretrained structure-only inverse folding model, and the fusion adapter `adapter.pt` bridging them. In particular, `adapter.pt` contains the bridge state and can be used only with the supplied inverse-folding model. It does not contain base-model capabilities.

Table 16: Architecture instantiated by the `sg273a_s040` release.

Component	Released setting
Genome encoder	Frozen EVO2-7B; activations from `blocks.28.mlp.l3`; feature width 4096; nucleic-acid sequence only.
Structural denoiser	Structure-only absorbing-state diffusion model; node, edge, and hidden width 128; 3 geometric encoder layers; 3 bidirectional sequence-aware decoder layers; 32 nearest neighbors; 33 output symbols with shared DNA/RNA token indices.
Learned-query bridge	Linear $4096 \to 128$ projection; 16 learned queries; 2 pre-normalized Transformer-decoder layers with 4 heads, feed-forward width 512, GELU activation, and zero dropout; 4-head residue-to-query cross-attention.
Context routing	8-way one-hot encoding of protein/DNA/RNA presence; a context-adjusted scalar fusion gate, a 128-dimensional channel scale, and separate rank-16 hidden-state and $128 \to 33$ logit adapters for every context.
Inference mode	All three components in evaluation mode with gradients disabled; dropout and coordinate augmentation disabled; polymer-specific output masks applied before normalization.
Reference runtime	Linux, Python 3.11, CUDA 12.8, PyTorch 2.7.1, FlashAttention 2.8.0.post2, `evo2==0.6.0`, and `vtx==1.1.0`.

Both the inverse-folding and genome inference commands require a CUDA-capable GPU. Optional EVO2 optimized kernels are disabled unless `-use-kernels` is supplied.

Structural input and geometric features. EVO-IF accepts PDB, PDB.gz, mmCIF, and mmCIF.gz inputs. For mmCIF it uses the first declared biological assembly, whereas PDB coordinates are treated as an already assembled structure. Protein, DNA, RNA, and DNA–RNA-hybrid chains are retained. The mmCIF parser excludes water and the ions `NA`, `CL`, `K`, and `BR`. RNA will not be parsed as relabeled as DNA, even though DNA and RNA bases share sequence token indices in the denoiser. The structure path carries the polymer type and chain label separately, so this token sharing does not erase molecular or chain identity.

The featurizer loads the protein backbone atoms `N`, `CA`, `C`, and `O`, together with the nucleic-acid atoms `OP1`, `OP2`, `P`, `O5'`, `C5'`, `C4'`, `O4'`, `C3'`, `O3'`, `C2'`, `O2'`, and `C1'`. A residue is treated as structurally valid only when its required backbone atoms exceed the 0.8 occupancy cutoff, while the per-atom coordinate mask uses a 0.5 cutoff. Protein `CA` and nucleic-acid `C1'` coordinates define the 32-neighbor graph. Edge features combine a 16-dimensional embedding of relative residue offsets (clipped to $[-32, 32]$ and augmented with a same-chain indicator) with 16 radial basis functions spanning 2–22 Å over all available atom pairs, including a constructed protein C_β and a constructed nucleobase nitrogen. Node features encode macromolecules context type. When a residue cap is enabled, larger nucleic-acid assemblies are reduced with a seed-controlled, NA-centered nearest-residue crop.

Sequence serialization and fusion. At every denoising evaluation, the current DNA and RNA tokens are selected in assembly order and concatenated across chains without separator tokens. RNA `U` is mapped to `T`, while unknown and masked bases are mapped to `N`, yielding an `ACGTN` string for EVO2. The resulting 4096-dimensional activation is projected to width 128 and compressed by the 16 learned queries. The assembly-level route is

$$r = 4\mathbb{I}[\text{protein}] + 2\mathbb{I}[\text{DNA}] + \mathbb{I}[\text{RNA}] \in \{0, \dots, 7\}, \tag{16}$$

which selects the context modulation and the two low-rank adapter paths described in Section 3.3. Before softmax, logits that do not belong to the residue's polymer alphabet are set to $-\infty$, where

protein positions admit the 20 standard amino acids, and DNA/RNA positions admit the four canonical bases. RNA output identities are converted back to `A/C/G/U`, and chain separators are restored only when writing results.

Exact decoding procedure. The design mask can select protein, DNA, RNA, all nucleic-acid, or all polymer positions, where the released default is `na`. Selected positions are first replaced by the absorbing mask token, while unselected positions retain their input identities as decoder context. With M designable positions and T denoising steps, the default cosine schedule targets

$$M_k = \left\lfloor M \, \frac{1 + \cos(\pi k/T)}{2} \right\rfloor \quad \text{masked positions after step } k. \tag{17}$$

At step k, the model predicts every position, ranks currently masked sites by their maximum categorical probability, and reveals the highest-confidence sites until M_k masks remain. The updated nucleic-acid sequence is then serialized and re-encoded by EVO2 before the next step. The optional *rewalk* operation re-masks each designable site whose confidence is below 0.3 with probability 0.1, except after the last step. Any residual masks are filled by a final argmax prediction.

Unless explicitly labeled ITER-30, reported released-checkpoint results use one-pass decoding ($T = 1$) and argmax token decisions. For stochastic multi-sample use, the command-line implementation supports categorical sampling.

B.2 TRAINING DETAILS

Protocol All post-training stages keep EVO2 frozen. We use the full EVO2-7B checkpoint and read hidden states from `blocks.28.mlp.l3`. The alignment interface uses a query encoder with 16 learned queries, 4 attention heads, 2 layers, zero dropout, 4096-dimensional EVO2 inputs, 128-dimensional structure-only outputs, an 8-way biomolecular-context one-hot, and context adapters. We inherit the structural diffusion configuration, including a 33-token vocabulary, shared nucleic-acid tokens, protein/DNA/RNA parsing, `C1'` as the nucleic-acid reference atom, hidden dimension 128, 3 encoder layers, 3 decoder layers, 32 neighbors, and 0.1 dropout. During decoder-only stages, the bridge is frozen and the full structural diffusion decoder is trainable.

Curriculum training Staged curriculum fine-tuning adapts the coupled model across polymer contexts. Training starts from the structure-only checkpoint `s1836.pt/pure0718_2`. Table 17 specifies the data mixtures, trainable modules, and update counts. Figure 10 places the recorded training blocks on a cumulative update axis. Therefore the stage changes and their transient loss shifts can be read together from these learning trace fragments.

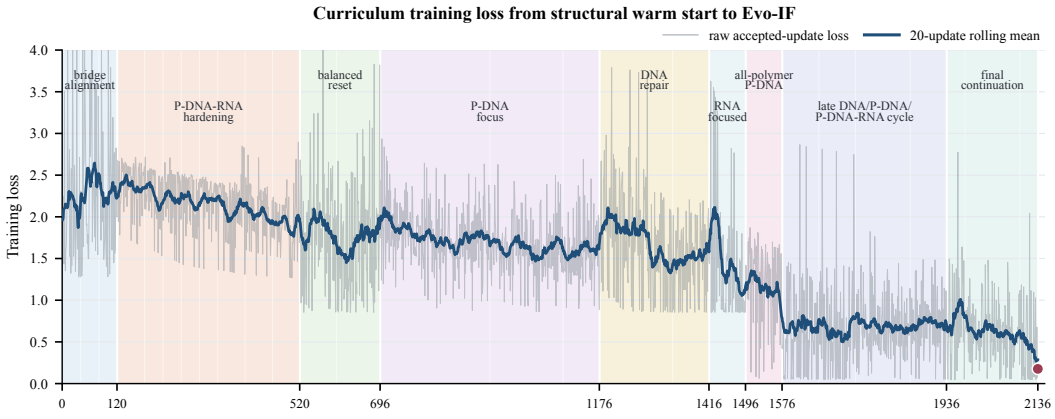

Figure 10: Fluctuations of training loss from the structural warm start to the released EVO-IF adapter. Raw points are the step-wise logged update losses, and the dark curve is a 20-update rolling mean. Background colors distinguish the curriculum stages summarized in Table 17.

The loss jumps at several boundaries reflect the dataset context distribution switches with optimizer restarts. For example, the curriculum moves from bridge alignment to protein–DNA-RNA hardening,

later narrows to protein-DNA, and finally alternates DNA, protein-DNA, and RNA repair tuning before achieving our released version.

Table 17: Curriculum blocks used in the reconstructed loss trace. Context buckets: 1=RNA, 2=DNA, 5=protein–RNA, 6=protein–DNA, and 7=protein–DNA–RNA.

Block	Ancestor runs	Updates	Context mix	Training policy and data balance
Bridge alignment	`run252-r0002`, `sg252r_s040`, `sg252s_s040`	120	1:14, 2:25, 5:16, 6:44, 7:21	bridge and structural decoder both trainable; full-training deduplicated pool followed by a static 1/2/6/7-balanced pool; cosine mask schedule; balanced loss weighting; bridge LR 5×10^{-8}–10^{-7} and decoder LR 10^{-7}–2×10^{-7}
Protein–DNA–RNA hardening	`sg253e`, `sg256a`-`sg256g`	400	7:400	bridge frozen, full structural decoder trainable; fixed full masking, position-balanced loss, uniform example weights; decoder LR 10^{-5} where recorded
Balanced reset	`sg257a_rerun`, `sg257b_s096`	176	1:48, 2:37, 5:1, 6:50, 7:40	equal-four-way alias manifest mixing RNA, DNA-only, protein–DNA, and protein-DNA-RNA examples; bridge frozen; fixed full masking; decoder LR 5×10^{-6} for the recorded rerun
Protein-DNA focus	`sg258a`, `sg258b`, `sg258c`	480	6:480	protein–DNA alias manifest; three decoder-only passes of 240, 160, and 80 accepted updates; fixed full masking and decoder LR 10^{-5}
DNA repair	`sg259a_s160`, `sg260a`	240	1:17, 2:176, 5:2, 6:25, 7:20	DNA-only continuation from the protein–DNA model, followed by an equal-four-way low-LR rebalance at 2×10^{-6}
RNA-focused	`sg261a_s080`	80	1:76, 5:4	RNA-oriented continuation after the DNA/equal-four-way repair; bridge frozen; fixed full masking; decoder LR 10^{-5}
All-polymer protein–DNA	`sg267a`	80	6:80	protein–DNA pass with `design_mode=all` and `loss_balance=polymer_equal`; decoder LR 10^{-6}
Late DNA/protein–DNA cycle	`sg268a`, `sg268b`, `sg269a`, `sg270a_s080`	360	2:160, 6:120, 7:80	DNA-only repair at 5×10^{-6}, protein–DNA expansion at 10^{-5}, then protein-DNA-RNA refresh
Final continuation	`sg271a_s040`, `sg272a`, `sg273a_s040`	200	1:37, 2:40, 5:3, 6:120	final DNA-only, protein–DNA, and RNA sequence

Except for the initial bridge-alignment block and the all-context `sg267a` pass, the plotted continuation blocks freeze the bridge and update the full structural decoder under `design_mode=na`. The final 200-update block comprises 40 DNA-only, 120 protein–DNA, and 40 RNA-oriented updates, and its endpoint is `sg273a_s040`.

Balanced-pool and final-stage settings For the controlled balanced training pool, candidates are drawn only from the training split and are materialized as mmCIF. The audited pool contains balanced active biomolecular-context categories, including category 1 (RNA-only), category 2 (DNA-only), category 5 (protein-RNA), category 6 (protein-DNA), and category 7 (protein-DNA-RNA). The context index is computed as $4 \cdot \mathbb{I}\{\text{protein}\} + 2 \cdot \mathbb{I}\{\text{DNA}\} + \mathbb{I}\{\text{RNA}\}$ from the parsed structure.

Large assemblies are cropped (`crop_large_structures`), with `max_total_L=2048`, `max_na_len=2048`, and `max_cif_bytes=50,000,000`.

The final RNA-oriented stage uses AdamW with weight decay 10^{-4}, learning rate 5×10^{-6} for the trainable parameters. The design mode is `na`, where only DNA/RNA positions are masked and supervised. For the EVO-IF checkpoint, the diffusion state is fully masked at train time (`mask_ratio_schedule=fixed`, `mask_ratio=1.0`, `full_mask_ratio=1.0`). Training and inference use the same candidate-serialization rule for the frozen EVO2 input. Nucleic-acid chains are concatenated in assembly order for the selected configuration.

B.3 INFERENCE

Unless labeled Iter-30 or described as a multi-sample evaluation, results use one-pass decoding. Each control retains the parser and scoring configuration of its paired comparison. Evaluation uses frozen EVO2-7B and the structural configuration in Table 16.

C TEMPORAL AND POST-CHECKPOINT GENERALIZABILITY

C.1 FIXED-CHECKPOINT GENOME-PATH INTERVENTIONS

These interventions use the intermediate `sg272a_s120` checkpoint. Bridge-off calls its adapted structural decoder directly, with no retraining. The permuted-input model shuffles current A/C/G/T symbols deterministically within each record while retaining mask positions and base counts. Jointly permuting already-computed query summaries would be vacuous because residue-to-memory attention has no memory-position encoding. The intervention is therefore applied before EVO2. Complete aggregates appear in Table 18.

Table 18: Fixed-weight path interventions. Values are mean native recovery (%). Δ is full minus intervention.

Panel	Decoder	Full	Bridge off	Δ_b	Permuted	Δ_p	N
Post-checkpoint	One-pass	47.21	44.16	$+3.05$	–	–	12
Post-checkpoint	Iter-30	50.01	45.64	$+4.37$	48.79	$+1.22$	12
Das Benchmark	One-pass	85.28	80.26	$+5.02$	–	–	14
Das Benchmark	Iter-30	85.27	79.86	$+5.41$	84.34	$+0.93$	14
DNA	One-pass	76.83	72.59	$+4.24$	–	–	121
DNA	Iter-30	75.78	74.64	$+1.14$	75.76	$+0.02$	121

C.2 MATCHED CORRUPTION CONTINUATION

The full-mask and mixed-corruption branches start from the same immutable checkpoint and receive the same ordered 80 structures, design masks, context quotas, optimizer state, learning rate, and trainable parameters. The full-mask branch uses mask ratio 1.0. The mixed branch samples uniformly from $[0.2, 1.0]$ and forces a fully masked example with probability 0.25. Table 19 reports the fixed temporal evaluation. The $+0.26$-point change in iterative margin rules out a simple short-continuation explanation.

Table 19: Matched corruption continuation on the post-checkpoint temporal panel. Recovery is in percent and Δ is Iter-30 minus one-pass.

Checkpoint	One-pass	Iter-30	Δ
Starting checkpoint	47.57	49.60	$+2.03$
$+80$ full-mask updates	47.75	49.43	$+1.68$
$+80$ mixed-corruption updates	47.87	49.81	$+1.94$

D RUNTIME AND RESOURSE USAGE

D.1 END-TO-END RUNTIME

Timing includes EVO2 encoding, genome-geometry fusion, structural decoding, and token commitment, while excluding model loading and mmCIF parsing. Eight length-stratified temporal structures are profiled on the same device and software stack.

Table 20: Synchronized end-to-end latency on the temporal profile. Relative cost is calculated from the mean latencies.

Decoder	Latency per structure (s)	Relative cost
One-pass	0.080	$1.0\times$
Iter-30	2.453	$30.7\times$

E LIMITATIONS

Coverage of molecular diversity. The evaluated RNA, DNA, and mixed-polymer assemblies cover a subset of the sequence families, fold topologies, and complex sizes encountered in nucleic-acid design. The behavior of genome–geometry alignment across this broader diversity remains incompletely characterized, particularly for rare folds and large assemblies with multiple interacting partners.

Resolution of genome-geometry alignment. The learned-query interface provides a compact representation shared across assembly residues. Its correspondence to specific sequence motifs, local geometric constraints, and long-range contacts is not yet fully resolved. Encoder scale, query allocation, and molecular-context routing offer useful dimensions for understanding and refining this correspondence.

Uncertain backbones and molecular chemistry. Performance depends on the quality of the conditioning geometry (Table 7). The current representation does not explicitly model structural uncertainty or the effects of noncanonical bases, chemical modifications, and ionic or ligand environments. Backbone ensembles and richer chemical context could extend design to these settings.

Functional and conformational design. The present objective emphasizes sequence reconstruction and compatibility with prescribed structures. The designed sequences have not yet been validated in wet-lab experiments. Binding affinity, chemical stability, and conformational populations impose additional constraints that are only indirectly captured by these criteria. Connecting structural design with functional measurements and explicit ensemble objectives would broaden the range of design outcomes.

Computational scale. The 1.18M-parameter interface operates alongside a frozen 7B encoder, so the computational footprint remains sensitive to sequence length and decoding budget. Smaller genome encoders, selective context retrieval, and distillation into the structural decoder could improve efficiency for larger assemblies and high-throughput design. Protein adaptation preserves most aggregate recovery while increasing perplexity (Table 10). Since protein-only inputs bypass EVO2, this test measures retention in the shared decoder.