SCALING RNA FOUNDATION MODELS THROUGH ITERATIVE REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological function in RNA is encoded across scales, from local sequence motifs to long-range structural dependencies. Conventional RNA foundation models couple parameter capacity and inference depth within a fixed architecture, prescribing a fixed computation graph for each inference. Here we present Eco-RNA, an 80M-parameter RNA foundation model that makes inference depth adjustable through learned iterative refinement. Its shared, input-conditioned Transformer core refines sequence representations through repeated application. The iteration count is selected after training and controls effective depth at fixed model parameters. We train across recurrent depths with masked reconstruction, state regularization, and a one-sided loss that anchors deeper reconstruction to a shallow reference. On the 70-assay RNAGym fitness benchmark, increasing refinement from one to four iterations raises aggregate absolute Spearman correlation from 0.242 to 0.286, with the gain retained through eight iterations. At four iterations, Eco-RNA surpasses the current SOTA Evo 2 40B on all three aggregate fitness metrics despite 500× fewer parameters. Its iteration configurations advance the compute–performance Pareto frontier among the evaluated models. It also outperforms the evaluated baselines in non-coding RNA family classification, RNA secondary-structure prediction, and chemical probing profile prediction. These results establish internal inference computation as an effective scaling dimension for RNA foundation models.

1 INTRODUCTION

RNA function depends on information distributed across a sequence, from local recognition motifs to interactions between distant nucleotides. Foundation models learn transferable representations of this information from unannotated sequences and support the prediction of RNA structure and function (Chen et al., 2022; Penić et al., 2025; Zou et al., 2024). Progress has accompanied increases in model capacity, pretraining data, and context length (Penić et al., 2025; Zou et al., 2024; Brixi et al., 2026). Conventional backbones, however, apply a fixed stack of transformations at inference. Each trained checkpoint therefore offers limited flexibility to trade additional computation for better predictions.

Internal test-time scaling makes execution depth an additional resource for prediction. In language modeling, inference computation can improve predictions through extended generation or candidate selection (Brown et al., 2024; Snell et al., 2025). Recurrent-depth architectures realize internal scaling by repeatedly applying a shared transformation to a latent representation (Dehghani et al., 2019; Geiping et al., 2025; Alabdulmohsin & Zhai, 2025). For a nucleotide encoder, this provides a direct way to allocate additional computation while preserving the sequence input and prediction interface. The central learning challenge is to make the shared update predictive across computational budgets: it must retain access to sequence information, control the scale of the recurrent state, and produce useful representations at different depths.

Here, we introduce **Eco-RNA**, an 80M-parameter RNA foundation model trained for iterative refinement (Figure 2). Building on recurrent-depth architectures (Geiping et al., 2025; Prairie et al., 2026), Eco-RNA reuses an input-conditioned Transformer core with an iteration count chosen at inference. Persistent input injection supplies sequence information at each iteration, while bounded state retention and normalization regulate the recurrent update. Pretraining combines masked reconstruction

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.

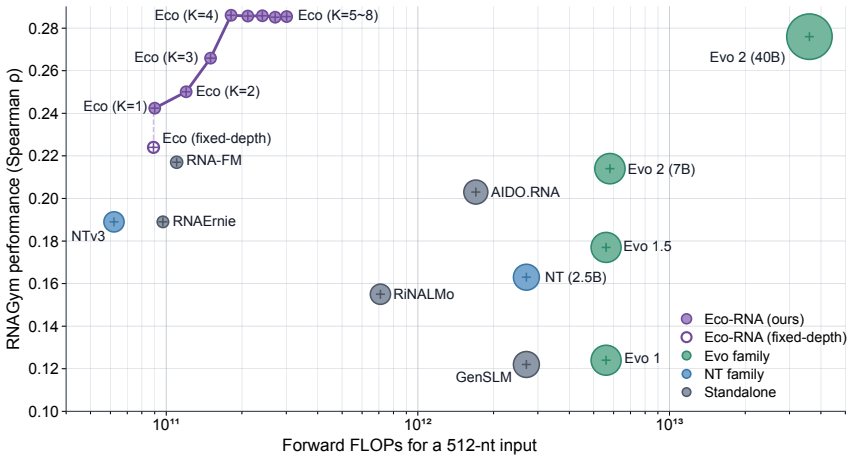

Figure 1: **Inference compute versus RNA fitness prediction.** Estimated FLOPs for one forward pass at 512 nucleotides (log scale) and type-balanced aggregate absolute Spearman correlation on RNAGym. Colors identify model families; marker size increases nonlinearly with parameter count. The line connects $K=1,\ldots,8$ of one Eco-RNA checkpoint, and the hollow marker denotes the fixed-depth control. FLOP accounting is given in Appendix A.4.

at sampled terminal depths with relative terminal-state regularization (Viakhirev et al., 2026). Our one-sided reconstruction anchor adds supervision when an endpoint has higher reconstruction loss than a shallower reference on the same trajectory. This formulation trains iterative computation within a self-supervised sequence backbone; RNAformer and RhoFold+ apply recurrence within task-specific structure predictors (Franke et al., 2023; Shen et al., 2024).

Eco-RNA advances the Pareto frontier of estimated forward FLOPs and RNA fitness prediction performance among the evaluated models (Figure 1). Under the original 70-assay RNAGym protocol (Arora et al., 2025), its one- to four-iteration configurations provide four Pareto-optimal operating points from one checkpoint. Increasing depth from one to four iterations raises aggregate absolute Spearman correlation from 0.242 to 0.286, an 18% relative improvement at approximately twice the forward FLOPs. The gain is retained through eight iterations. At four iterations, Eco-RNA exceeds Evo 2 40B on all three aggregate fitness metrics with approximately 500× fewer parameters (Brixi et al., 2026).

The depth benefit also appears in held-out reconstruction across pretraining checkpoints and persists after supervised task adaptation. Task-specific models initialized from Eco-RNA achieve the highest family-classification accuracy, secondary-structure F1, and probe-mean chemical-reactivity AUC among the compared models. Within each task, the adapted checkpoint and prediction head remain fixed throughout the depth sweep. The resulting curves exhibit different saturation depths, showing that a shared computational interface supports distinct prediction tasks while allowing their inference budgets to be chosen separately.

Our contributions are:

- **Internal test-time scaling for RNA foundation models.** Eco-RNA learns an input-conditioned shared update that supports adjustable inference depth within an 80M-parameter sequence backbone.
- **Cross-depth training with one-sided reconstruction anchoring.** The anchor selectively reinforces deeper predictions relative to a reference loss, complementing supervision at sampled depths and terminal-state regularization.
- **An improved compute–performance frontier.** Eco-RNA advances the RNAGym frontier through adjustable recurrence. Fixed-weight evaluations also establish depth-dependent gains in pretraining reconstruction and supervised RNA prediction.

Appendix B reviews related work on recurrent computation and biological foundation models.

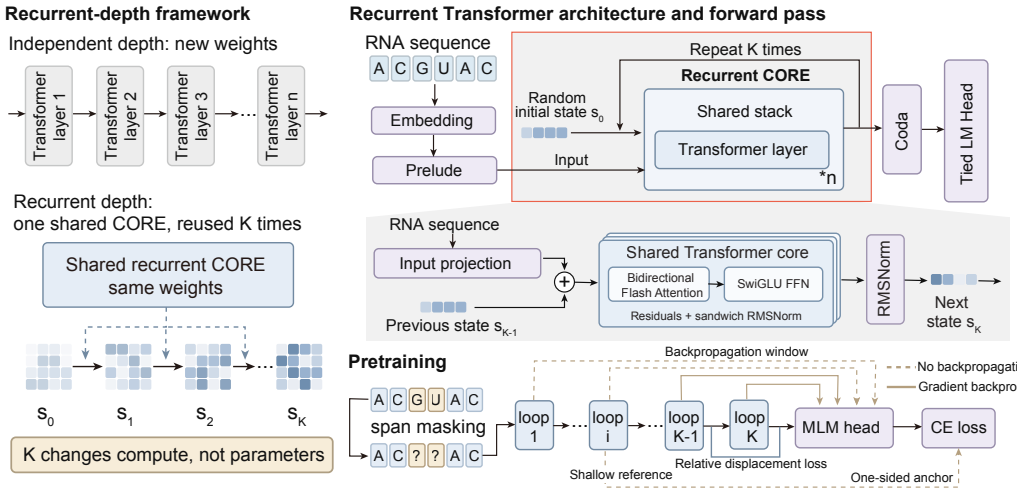

Figure 2: **Eco-RNA.** Left: a conventional stack assigns distinct parameters to successive layers, whereas Eco-RNA applies one shared core for K iterations, decoupling executed depth from parameter count. Right, top: a prelude encodes the input once; the recurrent core updates a randomly initialized state for K iterations; and the coda with a tied language-model head reads out the terminal state. Right, middle: each update combines retained state with the projected prelude representation, applies the shared Transformer core, and normalizes the next state. Bottom: pretraining applies masked reconstruction at each sequence's sampled terminal depth, relative displacement regularization to the final two states, and a one-sided reconstruction anchor that compares eligible deep endpoints with a detached reference.

2 Method

2.1 Recurrent depth as an inference budget

Eco-RNA exposes inference depth as an explicit control over internal computation. Its recurrent depth K is the number of times a shared Transformer core is applied; increasing K allocates more computation to the same input without changing the learned parameters or prediction interface. A prelude encodes the sequence once, the shared core iteratively updates a latent state, and a coda maps the state selected at depth K to a prediction.

Let $x = (x_1, \ldots, x_L)$ be a tokenized nucleotide sequence, $\tilde{x}$ its corrupted input, and $\mathcal{M}$ the positions selected for reconstruction. Let $\mathcal{V}$ denote the vocabulary, d the hidden width, and Θ all model parameters. Eco-RNA comprises a prelude P, an input-conditioned shared core R_θ, and a coda C, with L_p, L_r, and L_c Transformer blocks, respectively; $\theta \subset \Theta$ denotes the parameters of the shared core. The prelude and coda each execute once, whereas the core executes K times, so

$$D(K) = L_\mathrm{p} + KL_\mathrm{r} + L_\mathrm{c} \tag{1}$$

Transformer blocks are applied. Thus, recurrent depth changes the executed computation while leaving parameter count fixed (Dehghani et al., 2019; Geiping et al., 2025). Architecture values and the corresponding compute model appear in Appendices A.1 and A.4.

2.2 An input-conditioned shared update

Each recurrent update must combine the original sequence information with the result of preceding computation. Eco-RNA supplies a fixed prelude representation at every iteration and maintains a separate state that evolves along the trajectory. With embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ and prelude normalization N_p, these two inputs are initialized as

$$e = N_\mathrm{p}(P(E(\tilde{x}))), \qquad s^{(0)} \sim \mathcal{N}(0, \sigma_\mathrm{init}^2 I), \tag{2}$$

where $e, s^{(k)} \in \mathbb{R}^{L\times d}$ and the hyperparameter σ_{init} controls the initial-state scale. Padding entries of the sampled state are set to zero.

Eco-RNA bounds the retained fraction of each state channel while learning the scale of persistent input injection. Learnable channel vectors $a, b \in \mathbb{R}^d$ parameterize a positive input scale δ and a retention coefficient α:

$$\begin{aligned} \delta &= \text{softplus}(b), \\ \alpha &= \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \exp[-\delta \odot \exp(\min(a, a_{\text{cap}}))] \,. \end{aligned} \tag{3}$$

The hyperparameters satisfy $0 < \alpha_{\min} < \alpha_{\max} < 1$, and the numerical cap a_{cap} limits the exponent elementwise. The exponential factor is related to the diagonal transition parameterization studied in Parcae (Prairie et al., 2026); the affine mapping sets the retention interval. The learned coefficients are shared across sequences, positions, and iterations. With a learned input projection B and recurrent output normalization N_{r}, the state update is

$$s^{(k)} = N_{\text{r}}\Big(R_\theta\Big(\alpha \odot s^{(k-1)} + \delta \odot Be\Big)\Big), \qquad k = 1, \ldots, K. \tag{4}$$

Here $\odot$ denotes channelwise multiplication broadcast over positions, and B acts on each position's feature vector. The term $\alpha \odot s^{(k-1)}$ retains a channelwise fraction of the preceding state, whereas $\delta \odot Be$ makes the input-conditioned prelude representation directly available at every iteration. Each direct retention coefficient lies strictly below one; the nonlinear core and normalization determine the dynamics of the full update.

Normalization acts at two complementary boundaries. Sandwich RMS normalization regulates the attention and feed-forward residual branches within each core execution (Geiping et al., 2025); N_{r} normalizes the state handed to the next iteration. The Transformer blocks use bidirectional grouped-query attention (Ainslie et al., 2023), rotary positional embeddings (Su et al., 2024), and SwiGLU layers (Shazeer, 2020). Appendix A.1 specifies the block equations and initialization.

For masked reconstruction at depth k, a learned projection W_{o} maps the recurrent state into the coda, followed by output RMS normalization N_{o}. Writing this map as $\Psi_\Theta(s) = N_{\text{o}}(C(W_{\text{o}}s))$, the tied embedding matrix produces nucleotide logits $g^{(k)}$ and probabilities

$$\begin{aligned} h^{(k)} &= \Psi_\Theta(s^{(k)}), \qquad g^{(k)} = h^{(k)}E^\top, \\ p_{\Theta,k}(x_t \mid \tilde{x}) &= \text{softmax}\Big(g_t^{(k)}\Big)_{x_t}. \end{aligned} \tag{5}$$

The coda and tied reconstruction head are shared across depths; only the recurrent state supplied to this pretraining interface changes with k.

2.3 Training Objective

We train the shared update to produce useful predictions across inference budgets. Each sequence i independently samples a terminal depth K_i and receives masked-reconstruction supervision at that endpoint; the sampling law is given in Section 2.4. We couple this task supervision to two criteria on the recurrent trajectory: relative displacement regulates the final state update, and a one-sided reconstruction anchor reinforces deep endpoints whose reconstruction loss exceeds a shallower reference from the same trajectory. The combined objective is

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \lambda_{\text{fp}}\mathcal{L}_{\text{fp}} + \lambda_{\text{a}}\mathcal{L}_{\text{anchor}}, \tag{6}$$

where $\lambda_{\text{fp}}, \lambda_{\text{a}} \geq 0$ weight the auxiliary terms.

Terminal reconstruction. For a sequence i with supervised positions $\mathcal{M}_i \neq \varnothing$, define its reconstruction loss at depth k as

$$\ell_i^{(k)} = -\frac{1}{|\mathcal{M}_i|} \sum_{t\in\mathcal{M}_i} \log p_{\Theta,k}(x_{it} \mid \tilde{x}_i). \tag{7}$$

Let $\mathcal{I}_{\text{M}}$ be the supervised sequences in a microbatch and $Z_{\text{M}} = \sum_{i\in\mathcal{I}_{\text{M}}} |\mathcal{M}_i|$. The masked language modeling (MLM) objective (Devlin et al., 2019) averages terminal reconstruction over supervised tokens:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{Z_{\text{M}}} \sum_{i\in\mathcal{I}_{\text{M}}} |\mathcal{M}_i| \ell_i^{(K_i)}. \tag{8}$$

Sampling distributes terminal reconstruction supervision across depths over sequences and training updates.

Relative terminal displacement. To regulate the magnitude of the final recurrent update, we penalize relative terminal displacement (Viakhirev et al., 2026). For attention-valid positions V_i, let $\|u\|_{V_i}^2 = \sum_{t \in V_i} \|u_t\|_2^2$. The relative displacement penalty is

$$\mathcal{L}_{\text{fp}} = \frac{1}{|\mathcal{I}_{\text{fp}}|} \sum_{i \in \mathcal{I}_{\text{fp}}} \frac{\|s_i^{(K_i)} - s_i^{(K_i-1)}\|_{V_i}^2}{\max\left(\|s_i^{(K_i-1)}\|_{V_i}^2, \varepsilon_{\text{fp}}\right)}, \tag{9}$$

where $\mathcal{I}_{\text{fp}} = \{i : K_i \geq 2\}$ and $\varepsilon_{\text{fp}} > 0$ is a numerical floor. Per-sequence normalization measures the terminal update relative to the scale of its preceding state. The penalty covers all attention-valid positions, including positions without reconstruction targets, and differentiates both terminal states within the configured backward window.

The readout's local sensitivity relates terminal state displacement to the corresponding change in logits (Appendix A.2). Our one-sided reconstruction anchor complements this state-level criterion by acting directly on reconstruction loss.

One-sided reconstruction anchoring. We introduce a one-sided reconstruction anchor that uses a shallow reference loss to determine when a deeper prediction receives additional supervision. Let K_{a} be the reference depth, $m_{\text{a}} \geq 0$ a tolerance, and $\mathcal{I}_{\text{a}} = \{i \in \mathcal{I}_{\text{M}} : K_i > K_{\text{a}}\}$ the supervised sequences whose terminal depth exceeds the reference. We define

$$\mathcal{L}_{\text{anchor}} = \frac{1}{|\mathcal{I}_{\text{a}}|} \sum_{i \in \mathcal{I}_{\text{a}}} \left[\ell_i^{(K_i)} - \text{stopgrad}\left(\ell_i^{(K_{\text{a}})}\right) - m_{\text{a}}\right]_+, \tag{10}$$

where $[u]_+ = \max(u, 0)$ and stopgrad removes differentiation through its argument. The reference and terminal losses use the same corrupted input, initial state, supervised positions, and model parameters. The reference state is captured from the same recurrent trajectory and read out without gradient tracking.

The optimization effect of this one-sided form is explicit in its gradient. Denote the per-sequence hinge in Equation (10) by r_i^{a}. Along the retained training graph, its gradient is

$$\nabla_\Theta r_i^{\text{a}} = \mathbf{1}\left\{\ell_i^{(K_i)} > \ell_i^{(K_{\text{a}})} + m_{\text{a}}\right\} \nabla_\Theta \ell_i^{(K_i)}. \tag{11}$$

Here $\mathbf{1}\{\cdot\}$ is the indicator function. When an eligible deep endpoint's loss exceeds its detached reference by more than m_{a}, the anchor adds the gradient of that endpoint's reconstruction loss; λ_{a} controls its contribution to Equation (6). The observed nucleotides specify the learning target, and the reference determines when to strengthen that supervision. Improvements beyond the reference remain unpenalized, leaving the representations and predictions free to evolve with depth.

MLM is averaged over supervised tokens, and each auxiliary objective over its eligible sequences. Appendix A.2 specifies empty-set conventions and gradient paths.

2.4 Depth sampling and truncated backpropagation

We sample the terminal depth independently for every sequence, exposing the shared update and readout to different computational budgets within a microbatch (Prairie et al., 2026). We draw a base depth from a Poisson–lognormal distribution (Geiping et al., 2025) and impose a minimum depth $K_{\min}$:

$$\begin{aligned} \tau_i &\sim \mathcal{N}\left(\log \mu_K - \frac{\sigma_K^2}{2}, \sigma_K^2\right), \\ \widetilde{K}_i &\sim \text{Poisson}(e^{\tau_i}), \qquad K_i = \max(K_{\min}, \widetilde{K}_i). \end{aligned} \tag{12}$$

Here μ_K is the expected Poisson intensity before applying the lower bound $K_{\min}$, and σ_K controls dispersion in log intensity. This law places most probability on shorter trajectories while retaining a tail of larger compute budgets.

Forward depth and differentiated depth serve distinct roles. The sampled K_i determines the complete trajectory used for prediction, while a terminal window W_{bwd} controls recurrent activation storage. All K_i iterations execute in the forward pass, and gradients are retained through the final $\min(K_i, W_{\text{bwd}})$ iterations (Geiping et al., 2025; Prairie et al., 2026). A prefix outside this window is detached before the differentiated suffix begins.

The detached prefix still contributes its terminal state to the prediction; gradient propagation is restricted to the retained suffix, where persistent input injection also provides a path to the prelude. The configured suffix retains both states required by the displacement objective, while the anchor readout remains detached. Sampling K_i before applying the backward window preserves the distribution of forward budgets independently of this memory control. Appendix A.2 gives the wavefront schedule and the distributed normalization.

2.5 Pretraining and inference

Stage A pretrains the complete Eco-RNA backbone end to end from random initialization with masked reconstruction. Its data mixture is dominated by OpenGenome2 genomic sequences (Brixi et al., 2026) and RNAcentral (RNAcentral Consortium, 2019). Stage B continues from the Stage-A checkpoint on the same sources while shifting the sampling mixture toward RNA, without changing the architecture, objective, or prediction interface. Downstream results use the Stage-B checkpoint, whereas the pretraining analysis in Section 3.2 follows Stage-A checkpoints along training. Appendix A.3 describes the data sources and training configurations.

At inference, K is specified explicitly. A fixed, position-indexed initial-state table is reused throughout each depth sweep, isolating the effect of additional core applications. Fitness prediction uses the pretrained checkpoint directly. Each supervised task starts from the same Stage-B pretrained backbone; after task adaptation, its checkpoint and task head are held fixed across K. Section 3.1 defines the task-specific readouts.

3 Experiments

3.1 Protocol

We begin with self-supervised reconstruction to establish whether repeated computation is useful for the objective used to train Eco-RNA. We then test whether this benefit extends to RNA fitness prediction at a competitive computational cost. Assay-level analyses examine where the aggregate gains arise. Finally, supervised tasks assess whether additional iterations remain useful across prediction scales after task-specific fine-tuning.

Within each downstream depth sweep, we vary K from 1 to 8 while holding model weights, inputs, initial states, and any prediction heads fixed. Fitness prediction uses the pretrained model without task-specific training and follows the original 70-assay RNAGym protocol. Absolute Spearman correlation, orientation-adjusted AUC, and absolute MCC are averaged within RNA type and then equally across the five types. Family classification, chemical reactivity, and secondary-structure prediction use separately fine-tuned copies of the same backbone. Evaluation details appear in Appendices A.5–A.7.

3.2 Progressive latent refinement emerges from pretraining

We start with held-out reconstruction to assess whether repeated computation helps Eco-RNA recover masked nucleotides during self-supervised pretraining. We evaluate four Stage-A checkpoints between 50k and 300k updates, comparing one, four, and eight iterations within each checkpoint (Table 1). At each measured checkpoint, increasing depth from one to four iterations reduces reconstruction loss by 0.040–0.042, while the reduction from four to eight iterations remains below 0.001. Later checkpoints have lower losses, but each also benefits from additional iterations at inference. Thus, the shared update supports more accurate reconstruction when applied repeatedly, and this benefit persists throughout the measured pretraining trajectory. This establishes that extra iterations are useful for the training objective before we assess their value for downstream RNA prediction.

Table 1: **Reconstruction loss across pretraining and depth.** Held-out RNA masked-reconstruction loss for Stage-A checkpoints at $K=1, 4, 8$. Masked inputs and initial states are matched across depth within each checkpoint; the final column reports the reduction from $K=1$ to 4.

Pretraining updates	$K=1$	$K=4$	$K=8$	$\Delta_{1\to4}$
50k	0.9357	0.8958	0.8955	0.040
100k	0.9070	0.8673	0.8667	0.040
200k	0.8829	0.8416	0.8408	0.041
300k	0.8772	0.8357	0.8348	0.042

3.3 Recurrent depth advances the compute-performance frontier

We next evaluate whether the reconstruction gains are accompanied by better prediction of RNA function at a competitive computational cost. RNA fitness prediction evaluates how well sequence-based scores rank experimentally measured variant effects. RNAGym comprises 70 assays across five RNA types (Arora et al., 2025). With model weights held fixed, increasing depth from one to four iterations raises aggregate absolute Spearman correlation from 0.242 to 0.286, an 18% relative gain, while MCC rises by 22% (Appendix Table A4). A fixed-depth control matched in parameter count, block count, data, and training schedule reaches 0.224 absolute Spearman correlation. Eco-RNA therefore exceeds this control after one iteration and improves further by repeating computation with the same learned weights.

To assess the cost of these gains, we estimate FLOPs for one forward pass on a 512-nucleotide input (Appendix A.4). Four Eco-RNA iterations require 2.0× the FLOPs of one iteration, while eight require 3.3×. Among the evaluated models, the $K=1$ through 4 configurations lie on the Pareto frontier of estimated forward FLOPs and aggregate absolute Spearman correlation (Figure 1). At $K=4$, Eco-RNA exceeds Evo 2 40B on all three aggregate fitness metrics (Brixi et al., 2026), with 500× fewer parameters and approximately 200× fewer forward FLOPs. One trained model therefore provides several competitive choices of prediction quality and computation, selected simply by changing the iteration count (Appendix Table A4).

3.4 Assay-level analysis of depth gains

Additional depth produces different benefits across assays, even when the checkpoint and scoring rule remain fixed. Among the five selected examples in Figure 3, the largest improvements occur in the splicing and aptamer assays, which also have the weakest first-iteration absolute Spearman correlations. A single additional iteration raises the splicing score from 0.045 to 0.280, showing that repeated application of the shared core can substantially improve variant ranking. The coding assay improves more gradually, whereas the tRNA and ribozyme examples begin with stronger predictions and show more modest gains.

The magnitude of a gain and the depth over which it develops are distinct aspects of refinement. The aptamer assay improves sharply and then levels off, while the coding assay continues improving across the displayed depths. The preferred depth can also depend on the evaluation criterion within the same assay: splicing Spearman correlation and AUC peak at $K=3$, whereas absolute MCC continues improving at $K=4$. These examples reveal both rapid correction and gradual improvement under the same recurrent computation rule. They motivate choosing depth according to the assay and the target metric, using validation to identify where additional iterations remain useful. Adjustable depth makes this choice possible without retraining the model for each inference budget.

3.5 Optimal execution depth scales with prediction granularity

Finer-grained predictions retain benefits from refinement after sequence-level classification has largely saturated. We examine this pattern using separately fine-tuned copies of Eco-RNA for family classification, per-nucleotide chemical reactivity, and pairwise secondary-structure prediction. Within each task, the adapted backbone and prediction head remain fixed throughout the depth sweep.

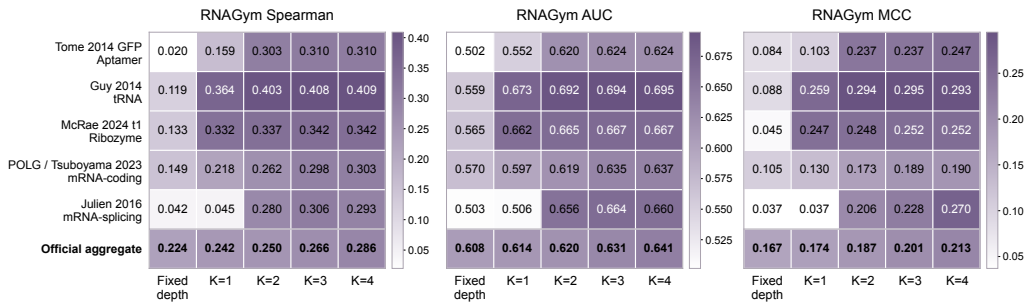

Figure 3: **Depth-dependent fitness prediction on five selected assays.** Columns: the fixed-depth control and $K{=}1, \ldots, 4$ of one Eco-RNA checkpoint. Rows: one assay per RNA type; the last row (labeled official aggregate) is the type-balanced aggregate over all 70 assays.

Sequence-level classification reaches its best observed accuracy within a few iterations. On the 13-class nRC benchmark, accuracy reaches 98.69% at $K{=}4$ (Table A5). At this reporting depth, Eco-RNA also leads the compared models (Table 2). The early plateau shows that additional iterations provide little further benefit for family assignment once this level of performance is reached.

Table 2: **RNA family classification and secondary-structure prediction.** Models are fully fine-tuned from their pretrained checkpoints on each benchmark. Eco-RNA reports its separately adapted task checkpoints at the shared reporting depth of $K{=}4$; column maxima are bolded.

Model	**Params**	**ncRNA accuracy (%)↑**	**SSP F1 (%)↑**
RNA-MSM	96M	84.85	57.98
3UTRBERT	87M	94.32	59.41
RNA-FM	100M	96.88	68.50
RiNALMo	150M	91.89	75.02
RNAGenesis	866M	97.81	73.72
Eco-RNA ($K{=}4$)	80M	**98.69**	**75.40**

Per-nucleotide reactivity shows a longer tail of improvement. After adaptation on Ribonanza (He et al., 2024), we evaluate DMS and 2A3 predictions on 114,836 held-out RNAGym sequences (Arora et al., 2025) (Appendix A.7). Eco-RNA leads all three probe-mean metrics at $K{=}4$ (Table 3), while smaller gains persist to $K{=}7$, where all three means reach their best observed values (Table A6). Thus, nucleotide-level prediction continues to benefit from refinement at depths beyond the classification plateau, with improvements in both discrimination and continuous reactivity estimation.

Pairwise structure prediction links the benefit of refinement to recovery of missing relationships. After fine-tuning on the bpRNA training split and evaluation on TS0 (Danaee et al., 2018; Singh et al., 2019), most of the F1 gain appears early, followed by smaller improvements at greater depths (Table A5). From $K{=}1$ to $K{=}4$, recall increases by 3.3 percentage points while precision remains around 83–84%, indicating that refinement recovers additional reference pairs at sustained precision. Eco-RNA exceeds the compared models at $K{=}4$ (Table 2). Figure 4 illustrates a concrete correction in bpRNA-RFAM-33646: the local pairing-register error is resolved by $K{=}3$, and the corrected stem is retained at $K{=}4$. This example shows how additional iterations can revise specific pairing assignments. Appendix Figure A1 provides further structural examples.

Together, these results suggest that the saturation of refinement gains is related to prediction granularity. A family label summarizes an entire sequence in one categorical decision, whereas nucleotide-level and pairwise outputs require accurate predictions at many positions or position pairs. Their continued improvement is consistent with additional iterations refining distinctions that matter for dense outputs after the benefits for coarse classification have plateaued. Across these evaluated tasks, coarse labels reach their performance plateau early, while finer-grained outputs retain smaller opportunities for correction at greater depth.

Table 3: **Chemical-reactivity prediction.** Per-nucleotide AUC and macro-F1 use global-median binarization; MAE is computed on continuous DMS and 2A3 reactivity, and Mean weights the two probes equally. We fine-tune the learnable baselines (Chen et al., 2022; Fu et al., 2022; Sato et al., 2021) on Ribonanza under a shared supervised-adaptation protocol and evaluate the structure-based methods (Wayment-Steele et al., 2022; Do et al., 2006; Lorenz et al., 2011; Reuter & Mathews, 2010) through the same pipeline. Eco-RNA reports $K=4$; the complete sweep and adaptation details appear in Table A6 and Appendix A.7. Best displayed scores are in **bold**.

	DMS			2A3			Mean		
Method	**AUC↑**	**F1↑**	**MAE↓**	**AUC↑**	**F1↑**	**MAE↓**	**AUC↑**	**F1↑**	**MAE↓**
RNAstructure	0.646	0.605	0.326	0.721	0.667	0.300	0.683	0.636	0.313
RNAfold (ViennaRNA)	0.645	0.604	0.324	0.725	0.670	0.296	0.685	0.637	0.310
CONTRAfold	0.659	0.608	0.372	0.740	0.684	0.330	0.700	0.646	0.351
EternaFold	0.647	0.598	0.360	0.760	0.702	0.309	0.703	0.650	0.335
UFold	0.841	0.766	0.182	0.727	0.668	0.244	0.784	0.717	0.213
MXfold2	0.843	0.766	0.187	0.730	0.668	0.234	0.787	0.717	0.210
RNA-FM	**0.853**	**0.780**	0.173	0.799	0.722	0.201	0.826	0.751	0.187
Eco-RNA ($K=4$)	**0.853**	0.774	**0.156**	**0.874**	**0.791**	**0.160**	**0.864**	**0.782**	**0.158**

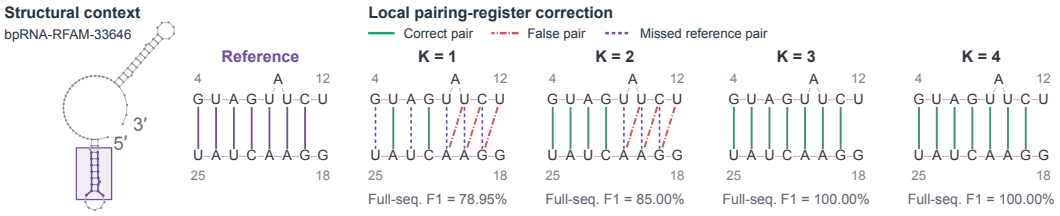

Figure 4: **Local pairing refinement across recurrent depth.** The left panel locates the displayed stem in bpRNA-RFAM-33646; the remaining panels compare the reference with predictions at $K=1,\ldots,4$. F1 scores are computed over the full sequence; the structural views show only the highlighted region.

4 Discussion and Conclusion

Eco-RNA establishes internal inference computation as a scaling dimension for RNA foundation models. A single recurrent checkpoint provides several compute–performance operating points and advances the RNAGym Pareto frontier. The ability to improve predictions by reusing learned parameters gives model users direct control over execution depth after training. This complements the capacity and data choices made during pretraining and extends the computational flexibility of RNA sequence models.

The results connect this inference capability to both representation learning and downstream prediction. Additional depth reduces held-out reconstruction loss at multiple stages of pretraining, and depth-dependent gains persist after adaptation to sequence labels, nucleotide pairs, and chemical reactivity. The tasks differ in their response to additional computation: classification saturates early, whereas structure and reactivity show smaller gains at higher budgets. For secondary structure, increased recall at sustained precision accounts for most of the improvement. These observations establish the value of evaluating computational depth for each prediction task, with a common backbone supporting several useful inference budgets.

The training formulation links recurrent state dynamics to a criterion in prediction space. Relative terminal displacement regularizes state movement, while the one-sided reconstruction anchor directs additional supervision toward deeper endpoints that underperform a reference depth. Because the anchor retains the observed nucleotide as its target, improvements beyond the reference remain unpenalized. The reference identifies depth-related degradation, while the reconstruction target specifies the desired prediction. This separation provides a task-level criterion for training additional recurrent computation within a shared sequence model.

AI USE STATEMENT

Generative AI tools were used solely to polish the language of the manuscript, including grammar, wording, and readability. They were not used to develop the research methodology, formulate scientific claims, design experiments, or interpret results. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

This study uses previously published sequence databases and computational benchmarks. It reports sequence classification, fitness scoring, and chemical-reactivity prediction, without new biological experiments or human-subject research. The authors declare no competing interests.

REPRODUCIBILITY STATEMENT

Section 2 defines the recurrent model and training objectives. Appendices A.1–A.3 specify the architecture, gradient paths, normalization, data sources, and pretraining configuration. Appendices A.5, A.6, and A.7 describe task scoring and adaptation, and Appendix A.4 derives the inference-compute estimate. Training and evaluation use published sequence and benchmark resources, including OpenGenome2, RNAcentral, RNAGym, and Ribonanza (Brixi et al., 2026; RNAcentral Consortium, 2019; Arora et al., 2025; He et al., 2024). Model weights and evaluation configurations will be made publicly available upon publication.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298.

Ibrahim Alabdulmohsin and Xiaohua Zhai. Recursive inference scaling: A winning path to scalable inference in language and multimodal systems. *arXiv preprint arXiv:2502.07503v4*, 2025. doi: 10.48550/arXiv.2502.07503.

Rohit Arora, Murphy Angelo, Christian Andrew Choe, Courtney A Shearer, Aaron W Kollasch, Fiona Qu, Ruben Weitzman, Artem Gazizov, Sarah Gurev, Erik Xie, et al. RNAGym: Large-scale benchmarks for RNA fitness and structure prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.16.660049. URL https://www.biorxiv.org/content/10.1101/2025.06.16.660049v1.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-Recursions: Learning dynamic recursive depths for adaptive token-level computation. *arXiv preprint arXiv:2507.10524v1*, 2025. doi: 10.48550/arXiv.2507.10524.

Sam Boshar, Benjamin Evans, Ziqi Tang, Armand Picard, Yanis Adel, Franziska K Lorbeer, Chandana Rajesh, Tristan Karch, Shawn Sidbon, David Emms, et al. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. *bioRxiv*, 2025. doi: 10.64898/2025.12.22.695963.

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.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. doi: 10.48550/arXiv.2407.21787.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology. *bioRxiv*, 2026. doi: 10.64898/2026.06.03.729735.

Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022. doi: 10.48550/arXiv.2204.00300.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P De Almeida, Hassan Sirelkhatim, et al. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z.

Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Research*, 46(11):5381–5394, 2018. doi: 10.1093/nar/gky285.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Chuong B Do, Daniel A Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, 2006. doi: 10.1093/bioinformatics/btl246.

Antonino Fiannaca, Massimo La Rosa, Laura La Paglia, Riccardo Rizzo, and Alfonso Urso. nRC: non-coding RNA classifier based on structural features. *BioData Mining*, 10(1):27, 2017. doi: 10.1186/s13040-017-0148-2.

Veniamin Fishman, Yuri Kuratov, Aleksei Shmelev, Maxim Petrov, Dmitry Penzar, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. GENA-LM: a family of open-source foundational DNA language models for long sequences. *Nucleic Acids Research*, 53(2):gkae1310, 2025. doi: 10.1093/nar/gkae1310.

Jörg K. H. Franke, Frederic Runge, and Frank Hutter. Scalable deep learning for RNA secondary structure prediction. *arXiv preprint arXiv:2307.10073*, 2023. doi: 10.48550/arXiv.2307.10073.

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

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. *arXiv preprint arXiv:2502.05171*, 2025. doi: 10.48550/arXiv.2502.05171.

Shujun He, Rui Huang, Jill Townley, Rachael C Kretsch, Thomas G Karagianes, David B T Cox, Hamish Blair, Dmitry Penzar, Valeriy Vyaltsev, Elizaveta Aristova, et al. Ribonanza: deep learning of RNA structure through dual crowdsourcing. *bioRxiv*, 2024. doi: 10.1101/2024.02.24.581671.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.

Shuzhen Li, Yifan Zhang, Jiacheng Guo, Quanquan Gu, and Mengdi Wang. DeepLoop: Depth scaling for looped transformers. *arXiv preprint arXiv:2607.13491v2*, 2026. doi: 10.48550/arXiv.2607.13491.

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*, 6:26, 2011. doi: 10.1186/1748-7188-6-26.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in Neural Information Processing Systems*, 36:43177–43201, 2023.

Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

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

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y. Fu. Parcae: Scaling laws for stable looped language models. *arXiv preprint arXiv:2604.12946v1*, 2026. doi: 10.48550/arXiv.2604.12946.

Jessica S Reuter and David H Mathews. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 11:129, 2010. doi: 10.1186/1471-2105-11-129.

RNAcentral Consortium. RNAcentral: a hub of information for non-coding RNA sequences. *Nucleic Acids Research*, 47(D1):D221–D229, 2019. doi: 10.1093/nar/gky1034.

Emanuele Rossi, Federico Monti, Michael Bronstein, and Pietro Liò. ncRNA classification with graph convolutional networks. *arXiv preprint arXiv:1905.06515*, 2019. doi: 10.48550/arXiv.1905.06515.

Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. RNA secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, 12(1):941, 2021. doi: 10.1038/s41467-021-21194-4.

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range DNA sequence modeling. In *International Conference on Machine Learning (ICML)*, 2024.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. doi: 10.48550/arXiv.2002.05202.

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.

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Ivan Viakhirev, Kirill Borodin, Amirah Almutairi, Serguei Barannikov, Maxim Abramov, and Grach Mkrtchian. Think shallow, solve deep: Controlling recurrent dynamics for reliable test-time depth. *arXiv preprint arXiv:2608.18222*, 2026. doi: 10.48550/arXiv.2608.18222.

Ning Wang, Jiang Bian, Yuchen Li, Xuhong Li, Shahid Mumtaz, Linghe Kong, and Haoyi Xiong. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nature Machine Intelligence*, 6(5):548–557, 2024. doi: 10.1038/s42256-024-00836-4.

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. doi: 10.1038/s41592-022-01605-0.

Xiao-Wen Yang, Ziyu Han, Xi-Hua Zhang, Wen-Da Wei, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Stabilizing recurrent dynamics for test-time scalable latent reasoning in looped language models. *arXiv preprint arXiv:2605.26733v1*, 2026. doi: 10.48550/arXiv.2605.26733.

Chengting Yu, Xiaobo Shu, Yadao Wang, Yizhen Zhang, Haoyi Wu, You Wu, Rujiao Long, Ziheng Chen, Yuchi Xu, Wenbo Su, and Bo Zheng. SpiralFormer: Looped transformers can learn hierarchical dependencies via multi-resolution recursion. *arXiv preprint arXiv:2602.11698v2*, 2026. doi: 10.48550/arXiv.2602.11698.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *International Conference on Learning Representations (ICLR)*, 2024.

Shuxian Zou, Tianhua Tao, Sazan Mahbub, Caleb N Ellington, Robin Algayres, Dian Li, Yonghao Zhuang, Hongyi Wang, Le Song, and Eric P Xing. A large-scale foundation model for RNA function and structure prediction. *bioRxiv*, 2024. doi: 10.1101/2024.11.28.625345.

Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, et al. GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications*, 37(6):683–705, 2023. doi: 10.1177/10943420231201154.

A METHOD AND EVALUATION DETAILS

A.1 ARCHITECTURE AND NUMERICAL SPECIFICATION

During training, the Gaussian state in Equation (2) is sampled independently and zeroed at padding positions. Each evaluation sweep instead reuses one seed-fixed, position-indexed Gaussian table at every depth, keeping initialization invariant as K changes.

The retention coefficient evaluates the exponent as $\exp(\min(a, a_{\text{cap}}))$, bounding the numerical transition rate. Transition vectors and boundary maps are shared across all iterations.

Each Transformer block combines bidirectional grouped-query attention with QK normalization, RoPE, SwiGLU, and sandwich-normalized residual updates (Ainslie et al., 2023; Su et al., 2024; Shazeer, 2020; Geiping et al., 2025). Writing N_j for RMS normalization, the block computes

$$\widehat{u} = N_2(u + \text{Attn}(N_1(u))), \qquad u' = N_4(\widehat{u} + \text{MLP}(N_3(\widehat{u}))). \tag{A.1}$$

Embedding and linear weights use a zero-mean Gaussian initializer, with a residual-scaled variant for attention and MLP output projections. Biases are initialized to zero, RMSNorm gains to one, and B and W_{o} to the identity. Input embeddings and terminal vocabulary logits share the same matrix. Table A1 gives the complete backbone configuration. Its $6+3K$ executed blocks count one terminal path and exclude the detached anchor readout used only during training.

Table A1: **Eco-RNA backbone.**

Setting	**Value**
Trainable backbone parameters	80,856,192
Prelude / recurrent core / coda	3/3/3 Transformer blocks
Distinct / executed blocks	$9 \,/\, 6+3K$
Hidden / feed-forward width	768/3,072
Query / key–value heads; head width	12/6; 64
Context length / active vocabulary	8,192 tokens / 23 tokens
Tokenization	Single-nucleotide IUPAC; distinct DNA T and RNA U
Normalization / position encoding	RMSNorm ($\epsilon = 10^{-6}$); RoPE base 10^6
Initial state / evaluation state	$\sigma_{\text{init}} = 1$; position table with seed 0
Retention bounds / exponent cap	$\alpha_{\min} = 0.05,\ \alpha_{\max} = 0.95$; $a_{\text{cap}} = 30$
Weight initialization	Gaussian standard deviation 0.02; residual output scale 0.06455
Boundary maps / prediction head	B, W_{o} identity-initialized; input/output embeddings tied

A.2 VARIABLE-DEPTH TRAINING AND OBJECTIVES

Right-aligned execution and differentiation. Terminal depths K_i are sampled independently as in Section 2.4. Let $K_{\max} = \max_i K_i$. At wavefront step $t \in \{1, \ldots, K_{\max}\}$, the active rows are

$$\mathcal{A}_t = \{i : t > K_{\max} - K_i\}. \tag{A.2}$$

Each row enters the wavefront late enough to complete exactly K_i iterations at the common endpoint. All iterations execute forward. When $K_i > W_{\text{bwd}}$, the first $K_i - W_{\text{bwd}}$ iterations omit the autograd graph, and the boundary state is detached before differentiating the final W_{bwd} iterations. Shorter trajectories are differentiated in full.

Each training update proceeds as follows:

1. Corrupt the input and compute e_i; sample $s_i^{(0)}$ and K_i.
2. Execute the right-aligned recurrent wavefront, retaining the terminal and penultimate states and, when reached, the state at anchor depth K_{a}.
3. Read the terminal state with gradient tracking. Read a detached anchor state through the same coda and tied head under no-gradient evaluation.
4. Form the masked-token and auxiliary objectives below, accumulate their token-normalized gradients, and update the shared parameters.

Supervision and corruption. Let V_i be the attention-valid positions, $U_i \subseteq V_i$ the non-special positions eligible for corruption, and $\mathcal{M}_i \subseteq U_i$ the positions selected for MLM supervision. Special tokens participate in attention and state regularization but not in MLM. The terminal reconstruction loss and its supervised-token normalization are defined in Section 2.3.

For span corruption, let p_{sel} be the target selection fraction and μ_{span} the exponential span scale. A nonempty row requests $q_i = \max\{1, \min(\lfloor p_{\text{sel}}|U_i| \rfloor, |U_i|)\}$ unique positions. Randomly placed, boundary-clipped spans with lengths $\max\{1, \lfloor Y \rfloor\}$, $Y \sim \text{Exp}(1/\mu_{\text{span}})$, are unioned until this count is reached; $q_i = 0$ when U_i is empty. Conditional on selection, the replacement rule in Tables A2 and A3 is applied independently at each selected position. All selected positions contribute to reconstruction.

The state-displacement term uses $\mathcal{I}_{\text{fp}} = \{i : K_i \geq 2\}$. For these sequences, define

$$r_i^{\text{fp}} = \frac{\|s_i^{(K_i)} - s_i^{(K_i-1)}\|_{V_i}^2}{\max\{\|s_i^{(K_i-1)}\|_{V_i}^2, \varepsilon_{\text{fp}}\}}, \qquad \mathcal{L}_{\text{fp}} = \frac{1}{|\mathcal{I}_{\text{fp}}|} \sum_{i \in \mathcal{I}_{\text{fp}}} r_i^{\text{fp}}, \tag{A.3}$$

where $\varepsilon_{\text{fp}} > 0$ is the numerical floor. Anchor eligibility is $\mathcal{I}_{\text{a}} = \{i : K_i > K_{\text{a}}, |\mathcal{M}_i| > 0\}$. The captured anchor state is detached, and its output projection, coda, and tied head execute under no-gradient evaluation. An auxiliary mean is defined as zero when its eligible set is empty; an optimizer update requires a positive accumulated supervised-token count. At hinge equality, the implementation selects the zero subgradient.

Relative displacement and readout change. Fix the attention mask and view Ψ_Θ as a map over valid-position representations. If its derivative has operator norm at most $L_{\Psi,i}$ along the segment from $s_i^{(K_i-1)}$ to $s_i^{(K_i)}$, integrating the derivative along this segment and applying the tied embedding matrix gives

$$\left\|g_i^{(K_i)} - g_i^{(K_i-1)}\right\|_{V_i} \leq \|E\|_2 L_{\Psi,i} \left\|s_i^{(K_i)} - s_i^{(K_i-1)}\right\|_{V_i}. \tag{A.4}$$

Here $\|E\|_2$ is the spectral norm of the embedding matrix. Define the preceding-state scale

$$\begin{aligned} \zeta_i &= \sqrt{\max\left\{\|s_i^{(K_i-1)}\|_{V_i}^2, \varepsilon_{\text{fp}}\right\}}, \\ \|s_i^{(K_i)} - s_i^{(K_i-1)}\|_{V_i} &= \zeta_i \sqrt{r_i^{\text{fp}}}. \end{aligned} \tag{A.5}$$

Substituting Equation (A.5) yields the corresponding relative relation

$$\frac{\|g_i^{(K_i)} - g_i^{(K_i-1)}\|_{V_i}}{\zeta_i} \leq \|E\|_2 L_{\Psi,i} \sqrt{r_i^{\text{fp}}}. \tag{A.6}$$

These bounds relate relative terminal displacement to local changes in the readout. The reconstruction anchor complements this state-level criterion by acting directly on task loss through Equation (11).

Accumulation and DDP normalization. Dynamic token batches are normalized over all supervised tokens across ranks and accumulation microbatches. Let S_{rb} and m_{rb} denote summed token cross-entropy and supervised-token count on rank r, microbatch b; let Q be the world size and $Z_{\text{acc}} = \sum_{r,b} m_{rb}$. With $A_{rb} = \lambda_{\text{fp}} \mathcal{L}_{\text{fp},rb} + \lambda_{\text{a}} \mathcal{L}_{\text{anchor},rb}$, the update uses

$$\begin{aligned} \mathcal{L}_{rb}^{\text{model}} &= \frac{S_{rb}}{Z_{\text{acc}}} + \frac{m_{rb}}{Z_{\text{acc}}} A_{rb}, \\ \mathcal{L}^{\text{update}} = \frac{1}{Q} \sum_{r=1}^{Q} Q \sum_b \mathcal{L}_{rb}^{\text{model}} &= \frac{1}{Z_{\text{acc}}} \sum_{r,b} (S_{rb} + m_{rb} A_{rb}). \end{aligned} \tag{A.7}$$

The factor Q cancels DDP's rank average, while weighting each local auxiliary mean by m_{rb} aligns it with that microbatch's contribution to supervised-token count.

A.3 PRETRAINING DATA AND CONFIGURATION

Stage A trains the complete Eco-RNA backbone from random initialization on unlabeled sequences from four length-partitioned OpenGenome2 genomic sources, its non-coding-RNA partition, and a filtered RNAcentral corpus (Brixi et al., 2026; RNAcentral Consortium, 2019). Its schedule progressively increases RNA sampling while retaining genomic exposure. DNA and RNA sources use T and U, respectively, and sequences exceeding the context limit are randomly cropped. Table A2 reports the Stage-A configuration.

Stage B continues self-supervised pretraining with greater exposure to RNA, using the architecture and objective established in Stage A. Table A3 reports its configuration.

Table A2: **Stage-A pretraining configuration.**

Setting	**Value**
Data	OpenGenome2 genomic buckets and non-coding-RNA partition, filtered RNAcentral
Updates / maximum length	300,000 / 8,192 input tokens
Position selection	$p_{\text{sel}} = 0.15$; exponential span scale $\mu_{\text{span}} = 5$
Replacement after selection	80% mask / 10% modality-valid random base / 10% unchanged
Depth distribution	$\mu_K = 4,\ \sigma_K = 0.5,\ K_{\min} = 2$
Backward window	$W_{\text{bwd}} = 2$ terminal iterations
Displacement / anchor	$\lambda_{\text{fp}} = 0.05,\ \lambda_{\text{a}} = 0.10$
Anchor depth / margin	$K_{\text{a}} = 4,\ m_{\text{a}} = 0$
Fixed-point floor	$\varepsilon_{\text{fp}} = 10^{-12}$
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$
Weight decay / clipping	0.1 / global norm 1.0
Learning-rate schedule	15,000-update warmup to 8×10^{-5}, then cosine; floor 8×10^{-6}
Training system	8 NVIDIA H200 GPUs, BF16 DDP, activation checkpointing

Table A3: **Stage-B pretraining configuration.**

Setting	**Value**
Initialization	Stage-A checkpoint (300,000 updates)
Data	OpenGenome2 non-coding-RNA partition and RNAcentral at increased sampling weight; genomic buckets retained
Updates / maximum length	200,000 / 8,192 input tokens
Position selection	Span masking; $p_{\text{sel}} = 0.15$, exponential span scale $\mu_{\text{span}} = 4$
Replacement after selection	80% mask / 10% random nucleotide / 10% unchanged
Depth distribution / backward window	$\mu_K = 4,\ \sigma_K = 0.5,\ K_{\min} = 2$; $W_{\text{bwd}} = 2$
Displacement / anchor	$\lambda_{\text{fp}} = 0.05,\ \lambda_{\text{a}} = 0.10$
Anchor depth / margin	$K_{\text{a}} = 4,\ m_{\text{a}} = 0$
Optimizer	Fused AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$
Weight decay / clipping	0.1 / global norm 1.0
Learning-rate schedule	5,400-update warmup to 8×10^{-5}, then cosine; floor 8×10^{-6}
Training system	8 NVIDIA H200 GPUs; BF16 DDP, activation checkpointing

A.4 COMPUTE ACCOUNTING

Let L be sequence length, d the hidden width, d_{kv} the total key/value width, d_{f} the feed-forward width, and $|\mathcal{V}|$ the vocabulary size. Counting a fused multiply–add as two floating-point operations, the leading matrix-multiplication cost of one dense bidirectional Transformer block is

$$F_{\text{blk}}(L) = 4Ld(d + d_{\text{kv}}) + 4L^2 d + 6Ldd_{\text{f}}. \tag{A.8}$$

The input projection Be is recomputed at each iteration, W_{o} is applied once, and the tied vocabulary head is applied once. The resulting single-forward count is

$$F_{\text{Eco}}(K, L) = D(K)F_{\text{blk}}(L) + 2(K + 1)Ld^2 + 2Ld|\mathcal{V}|. \tag{A.9}$$

For $L = 512$, the count is 90.011, 180.608, and 301.404 GFLOPs at $K = 1, 4, 8$; it covers the dense matrix multiplications of one forward pass, which dominate cost at this scale. The compared models in Figure 1 are counted the same way from their released checkpoints and inference code.

A.5 NON-CODING RNA CLASSIFICATION AND SECONDARY-STRUCTURE PREDICTION

Family classification follows the published train, validation, and test split of the 13-class nRC benchmark (Fiannaca et al., 2017; Rossi et al., 2019). SSP uses the bpRNA training split and TS0 test set (Danaee et al., 2018; Singh et al., 2019). Both tasks adapt the full Eco-RNA backbone with a task-specific head: a lightweight classifier over the terminal `[CLS]` representation for nRC and a pairwise head over terminal coda features for SSP. SSP reports strict global full-matrix micro-F_1, pooling valid cells of the complete $L \times L$ matrices after directly thresholding pair probabilities at 0.5. Within each task, the adapted checkpoint and head remain fixed throughout the $K = 1, \ldots, 8$ sweep.

A.6 RNAGYM FITNESS PREDICTION

Fitness is evaluated zero-shot on the 70 RNAGym deep mutational scanning assays (Arora et al., 2025). Let w be an assay's wild-type sequence, v a variant, and $\mathcal{D}(v)$ its substituted positions. The wild-type-marginal score uses the uncorrupted wild-type context and sums the log-odds of the mutant nucleotide against the wild-type nucleotide at each substitution:

$$F_K(v; w) = \sum_{j \in \mathcal{D}(v)} \left[\log p_{\Theta,K}(v_j \mid w, j) - \log p_{\Theta,K}(w_j \mid w, j)\right]. \tag{A.10}$$

Here $p_{\Theta,K}(\cdot \mid w, j)$ is the vocabulary distribution at position j under the wild-type input w. Every substitution in a variant is scored under this shared context.

For each assay a, the measured-fitness vector y is binarized above its median for AUC and MCC, and predicted scores are binarized above their median for MCC. The reported orientation-invariant assay metrics are

$$q_{a,\rho} = |\,\mathrm{Spearman}(F_K, y)|, \quad q_{a,\mathrm{AUC}} = \max\{\mathrm{AUC}_a, 1 - \mathrm{AUC}_a\}, \quad q_{a,\mathrm{MCC}} = |\mathrm{MCC}_a|. \tag{A.11}$$

Let $\mathcal{T}$ contain the five RNA types: mRNA-coding, ribozyme, tRNA, mRNA-splicing, and aptamer, and let $\mathcal{A}_t$ be the assays of type t. Each headline metric $m \in \{\rho, \mathrm{AUC}, \mathrm{MCC}\}$ first averages assays within type and then weights the five types equally:

$$Q_m = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{|\mathcal{A}_t|} \sum_{a \in \mathcal{A}_t} q_{a,m}. \tag{A.12}$$

Checkpoint, alphabet convention, initial-state table, and scoring rule remain fixed throughout each depth sweep.

The fixed-depth control matches the recurrent model's parameter count, block count, data, and training schedule. It executes nine distinct blocks once per sequence as a conventional masked language model and uses the same fitness scoring and aggregation. Table A4 reports the model comparison with the complete fitness depth sweep. Table A5 reports the corresponding sweeps after supervised adaptation.

A.7 CHEMICAL-REACTIVITY PREDICTION

The RNAGym secondary-structure track evaluates per-nucleotide DMS and 2A3 chemical reactivity (Arora et al., 2025). For our supervised evaluation, we fine-tune learnable models on the Ribonanza chemical-mapping corpus (He et al., 2024) and evaluate on 114,836 held-out sequences. We evaluate RNA-FM, UFold, and MXfold2 (Chen et al., 2022; Fu et al., 2022; Sato et al., 2021) with matched reactivity heads; thermodynamic predictors supply unpaired-probability reactivity scores from their structure outputs.

Eco-RNA is adapted jointly with a four-block convolutional head whose input concatenates the terminal representation and its difference from the first-iteration representation; this difference is zero at $K = 1$. The adapted checkpoint, head, and initial-state table remain fixed throughout the $K = 1, \ldots, 8$ sweep.

AUC and macro-F_1 use global-median binarization, MAE uses continuous reactivity, and each mean weights DMS and 2A3 equally. Table 3 reports Eco-RNA at $K = 4$ alongside the baselines; Tables A5 and A6 provide the complete depth results as probe means and per-probe scores, respectively.

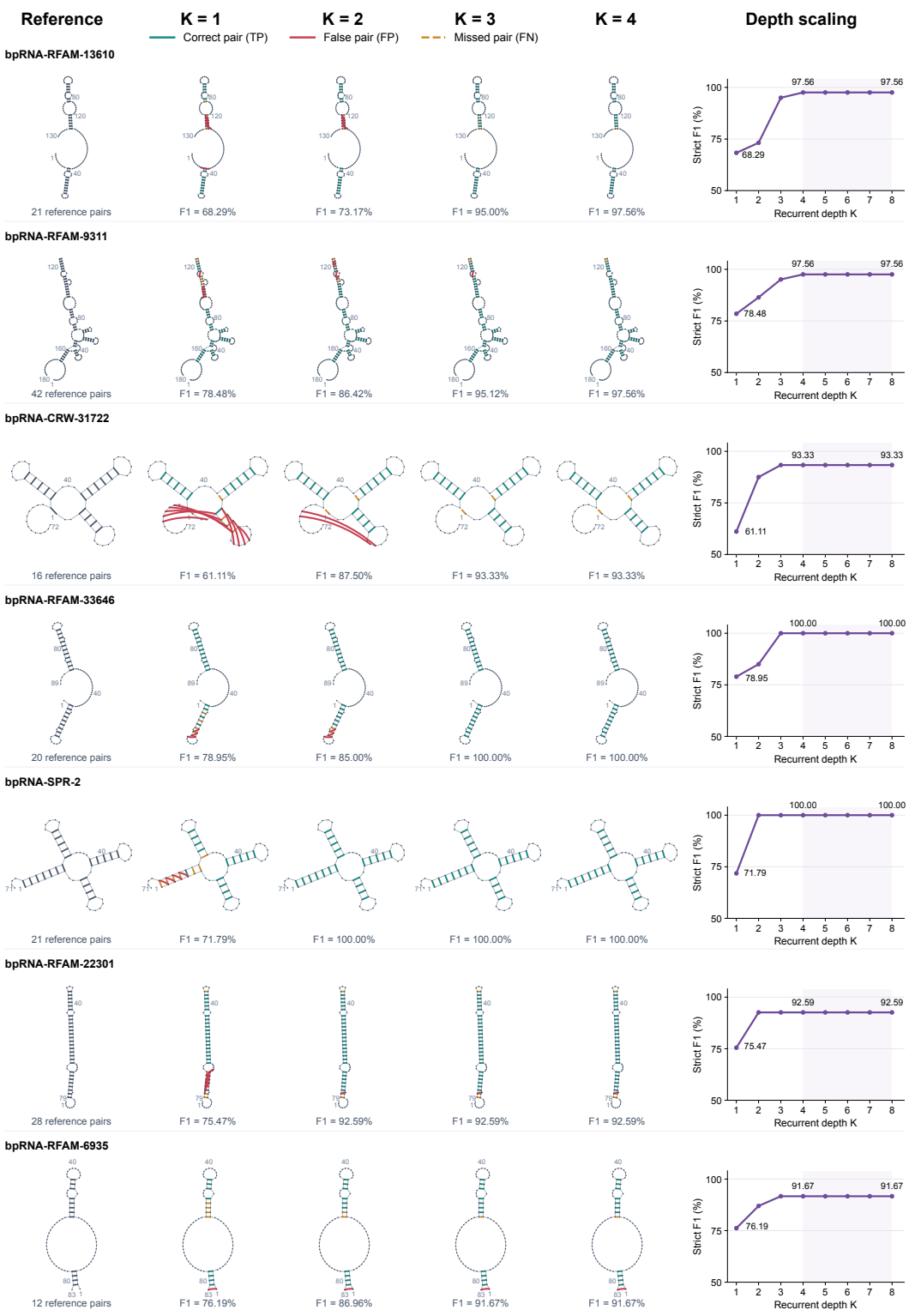

Figure A1: **Secondary-structure case studies across recurrent depths.** Rows show seven selected bpRNA sequences, with reference structures, Eco-RNA predictions at $K{=}1, \ldots, 4$, and full-sequence strict F1 over $K{=}1, \ldots, 8$. Teal, red, and dashed orange edges denote correct, false, and missed base pairs, respectively.

Table A4: **Zero-shot RNA fitness prediction across models and recurrent depths.** Evaluation uses RNAGym (Arora et al., 2025) with the type-balanced aggregation described in Appendix A.6. Eco-RNA evaluates one checkpoint at $K=1,\ldots,8$, with inputs, initial states, and scoring fixed; the matched fixed-depth control omits recurrence. Baseline and control scores retain three decimal places; recurrent Eco-RNA scores use four. Bold marks column maxima before rounding.

Model	Params	K	Spearman↑	AUC↑	MCC↑
GenSLM (Zvyagin et al., 2023)	2.5B	–	0.122	0.558	0.083
Evo 1 (Nguyen et al., 2024)	6.5B	–	0.124	0.561	0.091
RiNALMo (Penić et al., 2025)	650M	–	0.155	0.582	0.122
Nucleotide Transformer (Dalla-Torre et al., 2025)	2.5B	–	0.163	0.577	0.115
Evo 1.5 (Arora et al., 2025)	6.5B	–	0.177	0.583	0.131
Nucleotide Transformer v3 (Boshar et al., 2025)	650M	–	0.189	0.593	0.133
RNAErnie (Wang et al., 2024)	87M	–	0.189	0.598	0.154
AIDO.RNA (Zou et al., 2024)	1.6B	–	0.203	0.596	0.135
Evo 2 7B (Brixi et al., 2026)	7B	–	0.214	0.604	0.161
RNA-FM (Chen et al., 2022)	100M	–	0.217	0.606	0.155
Evo 2 40B (Brixi et al., 2026)	40B	–	0.276	0.636	0.212
Eco-RNA, fixed-depth control	80M	–	0.224	0.608	0.167
Eco-RNA (ours)	80M	1	0.2424	0.6138	0.1741
Eco-RNA (ours)	80M	2	0.2501	0.6199	0.1870
Eco-RNA (ours)	80M	3	0.2659	0.6309	0.2007
Eco-RNA (ours)	80M	4	**0.2861**	0.6409	**0.2130**
Eco-RNA (ours)	80M	5	0.2857	0.6407	0.2108
Eco-RNA (ours)	80M	6	0.2858	**0.6409**	0.2083
Eco-RNA (ours)	80M	7	0.2851	0.6406	0.2059
Eco-RNA (ours)	80M	8	0.2854	0.6407	0.2074

Table A5: **Fixed-weight depth sweeps after supervised adaptation.** The checkpoint, task head, inputs, and initial states remain fixed within each sweep. Classification and SSP are percentages; reactivity metrics average the DMS and 2A3 probes. Bold marks each row's best value before rounding.

K	1	2	3	4	5	6	7	8
nRC family classification								
Accuracy (%)↑	97.96	98.46	98.62	**98.69**	98.65	98.65	**98.69**	**98.69**
bpRNA TS0 secondary structure								
F_1 (%)↑	73.10	75.10	75.38	75.40	75.47	**75.48**	75.47	75.48
Precision (%)↑	83.00	83.66	83.70	83.67	83.69	83.69	83.69	**83.70**
Recall (%)↑	65.31	68.13	68.57	68.62	68.72	**68.74**	68.72	68.72
RNAGym chemical reactivity, mean over probes								
AUC↑	0.8212	0.8500	0.8604	0.8636	0.8651	0.8651	**0.8655**	0.8654
F_1 ↑	0.7491	0.7715	0.7798	0.7825	0.7840	0.7841	**0.7843**	0.7842
MAE↓	0.1945	0.1685	0.1608	0.1580	0.1569	0.1567	**0.1565**	0.1565

Table A6: **Chemical-reactivity prediction across recurrent depths.** The fine-tuned checkpoint and head are fixed across K. Metrics follow Table 3; Mean weights DMS and 2A3 equally. Each column's best value is in **bold**.

	DMS			2A3			Mean		
K	AUC↑	F1↑	MAE↓	AUC↑	F1↑	MAE↓	AUC↑	F1↑	MAE↓
1	0.836217	0.767908	0.179242	0.806089	0.730326	0.209712	0.821153	0.749117	0.194477
2	0.846327	0.771044	0.162480	0.853681	0.771873	0.174473	0.850004	0.771458	0.168476
3	0.854592	**0.776349**	0.156537	0.866237	0.783168	0.165125	0.860415	0.779758	0.160831
4	0.853233	0.773936	0.156149	0.873936	0.791049	0.159765	0.863584	0.782492	0.157957
5	**0.855257**	0.775581	**0.154953**	0.875030	0.792345	0.158763	0.865144	0.783963	0.156858
6	0.854139	0.774719	0.155325	0.876121	0.793491	0.158016	0.865130	0.784105	0.156670
7	0.854830	0.775125	0.155065	0.876097	0.793524	0.157951	**0.865464**	**0.784325**	**0.156508**
8	0.854335	0.774483	0.155280	**0.876384**	**0.793841**	**0.157747**	0.865359	0.784162	0.156514

B Related work

RNA and nucleotide foundation models. Scaling nucleotide foundation models has primarily meant increasing pretraining data, parameter capacity, or accessible context. RNA-FM (Chen et al., 2022), RiNALMo (Penić et al., 2025), and AIDO.RNA (Zou et al., 2024) learn general-purpose representations from unannotated RNA, while Evo 2 extends nucleotide modeling across genomic domains at substantially larger scale (Brixi et al., 2026). DNA-focused models have likewise expanded the design space through new tokenizations and sequence mixers, including DNABERT-2 (Zhou et al., 2024), Nucleotide Transformer (Dalla-Torre et al., 2025), HyenaDNA (Nguyen et al., 2023), and Caduceus (Schiff et al., 2024). Across these architectures, the pretrained backbone ordinarily executes a prescribed number of transformations for every prediction. Eco-RNA introduces inference depth as a separate scaling axis: one checkpoint supports multiple computational budgets while preserving its input and prediction interface.

Recurrent depth and computation allocation. Test-time computation can expand an output trace or search over additional candidates (Brown et al., 2024; Snell et al., 2025); recurrent depth allocates that computation to further transformation of the latent representation. Universal Transformers established the repeated application of a shared transition, optionally coupled to adaptive halting (Dehghani et al., 2019), whereas ALBERT shares parameters across a fixed number of layers to improve parameter efficiency (Lan et al., 2020). Huginn studies recurrent-depth language modeling with a shared middle core between a prelude and coda and an iteration count that remains selectable after training (Geiping et al., 2025). Other designs allocate recurrence at different granularities. RINS repeats an early network segment and combines stochastic recursion with depth-indexed adapters (Alabdulmohsin & Zhai, 2025); Mixture-of-Recursions assigns computation at the token level (Bae et al., 2025); and SpiralFormer couples a shared core to coarse-to-fine resolution changes (Yu et al., 2026). Eco-RNA applies a full-resolution sequence update with a sequence-wide iteration count chosen at inference.

Training shared updates across depths. A reusable update must learn over multiple prediction horizons while remaining tractable to optimize. Huginn combines Poisson–lognormal depth sampling with backpropagation through a terminal suffix (Geiping et al., 2025). Parcae uses an input-driven diagonal state transition, samples terminal depth per sequence, and separates executed forward depth from the retained backward window (Prairie et al., 2026). Complementary work examines the dynamics of repeated parameter use: DeepLoop derives residual and initialization scaling rules (Li et al., 2026), while STARS regularizes an estimate of the recurrent Jacobian's spectral radius (Yang et al., 2026). Think Shallow, Solve Deep analyzes finite-time behavior through terminal state displacement and latent anchoring (Viakhirev et al., 2026).

Eco-RNA formulates cross-depth learning around the predictive role of the recurrent state. Persistent input injection and bounded retention condition the shared Transformer update, while normalization controls its scale. Relative terminal displacement regulates state motion, and a one-sided reconstruction anchor connects the trajectory to masked-token prediction. State matching and output distillation supervise a target representation or distribution; the reconstruction anchor uses shallow loss to gate additional supervision at the deeper endpoint. The observed nucleotides remain the targets, leaving deeper predictions free to improve beyond the reference.

Iterative computation in biological models. Biological prediction systems have also benefited from iteration, although the recurrent object and learning objective differ across settings. RNAformer recycles a two-dimensional pairing representation under supervised secondary-structure training (Franke et al., 2023), and RhoFold+ feeds a predicted three-dimensional structure back through successive refinement rounds (Shen et al., 2024). In protein modeling, ESMFold2 applies looped folding layers to pretrained sequence representations and evaluates additional loops in selected settings (Candido et al., 2026). GENA-LM instead carries memory tokens between consecutive genomic segments during fine-tuning (Fishman et al., 2025). Eco-RNA places recurrence within the self-supervised sequence backbone, making adjustable inference depth available across multiple RNA prediction tasks.