OPTIMRNA: STRUCTURE-AWARE RNA LANGUAGE MODELING WITH FACTORIZED PAIR STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

RNA function is strongly shaped by pairwise structure, yet RNA language models (RNA-LMs) rely on BERT-style Transformers that only maintain per-token representations between layers. To explicitly integrate prior knowledge of RNA pairing into sequence learning, we introduce OPTIMRNA, an architecture that augments the token stream with a pair state maintained and refined across Transformer layers. Initialized from canonical pairs supported by stem-compatible neighborhoods, the pair state is progressively updated from contextual token representations using signed low-rank transformations propagated by antidiagonal convolutions encoding RNA-stem patterns. Every Transformer layer updates the pair state, while later layers read it as a bounded attention bias. Its low-rank formulation enables memory-efficient pair modeling with negligible parameter overhead compared with standard RNA-LMs. We show that OPTIMRNA outperforms all evaluated RNA-LMs in zero-shot secondary-structure prediction and, after supervised fine-tuning, competes with substantially larger foundation models without compromising performance on tasks less directly tied to secondary structure. Controlled ablations and mechanistic analyses support the complementary roles of the pair pathway and stem-aware propagation. Together, these results establish OPTIMRNA as an efficient biologically-informed architecture for RNA language modeling.

1 INTRODUCTION

RNA is typically represented as a linear nucleotide sequence, yet many of its biological functions depend on secondary structures formed by base pairing between nucleotides that may be far apart in sequence. This structure is organized into motifs such as stems and loops (Hendrix et al., 2005). These structures influence many processes such as RNA stability, translation efficiency and conformational switching (Cao et al., 2024). Learning comprehensive RNA representations which can be used to study all of these processes therefore requires models to capture not only sequence content, but also the latent network of pairwise interactions through which an RNA molecule folds and functions.

RNA language models (RNA-LMs) pursue this goal at scale by training BERT-style Transformers (Devlin et al., 2019) on unannotated RNA sequences with masked language modeling, thereby learning structural and functional representations without explicit folding supervision (Zablocki et al., 2025). However, biologically plausible base pairing is highly constrained rather than arbitrary. Canonical Watson–Crick pairs, together with G–U wobble pairs, account for most stem-forming interactions, and viable pairs tend to occur in contiguous, properly oriented arrangements that form stems. Standard RNA-LMs do not encode these constraints explicitly and must instead recover them from sequence statistics. Incorporating known RNA pairing rules could be an effective inductive bias to improve RNA-LMs' structural learning without direct supervision. Explicit pair representations provide a natural mechanism for integrating and progressively refining such biological priors. They are central to modern biomolecular structure-prediction architectures, where pairwise evidence is learned under direct structural supervision (Jumper et al., 2021). Transferring this principle to RNA language modeling is nevertheless challenging: masked language modeling provides no direct supervision for the pair representation, while explicitly maintaining a pairwise state requires quadratic memory and commonly used triangle-based updates can require cubic computation. ERNIE-RNA (Yin et al., 2025) partially addresses this challenge by initializing attention from nucleotide com-

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.

patibility and passing structurally informed attention maps between consecutive layers. However, its implementation materializes large quadratic intermediates and it does not provide a separate pair state that accumulates structural evidence across layers. Current RNA-LMs therefore lack an efficient mechanism for integrating, preserving, and progressively refining biological pairing priors.

We introduce OPTIMRNA, which realizes this principle with a new mechanism that maintains a biologically initialized pair state alongside the conventional token stream. The state is seeded from canonical pairs supported by local stem-compatible neighborhoods (Xia et al., 1998). Every Transformer layer derives a low-rank pairwise update from its contextual token representations and adds it to the state, allowing structural evidence to be retained and refined across depth. We further introduce an antidiagonal convolution that propagates each update along the spatial pattern formed by neighboring base pairs in RNA stems (Hendrix et al., 2005). The late Transformer layers read the resulting pair state as an attention bias, after the earlier layers have first contextualized the token embeddings and accumulated pairwise evidence. Because these updates are combined linearly, the complete state retains an exact sum-of-low-rank representation enabling a memory-efficient implementation that limits intermediate quadratic objects, and can even reconstruct the required pair scores on demand for linear-memory complexity when needed.

Contributions. Our main contributions can be summarized as follows:

- **Biologically grounded and efficient pair modeling.** We introduce an RNA-specific architecture that integrates known regularities of RNA pairing into masked language modeling and maintain and refine pairwise evidence across Transformer layers. Its factorized formulation avoids storing a dense pair state, while exposing a pair-level output that can be used directly for structure prediction or supervised for other pair-level tasks.
- **Structure-aware representation learning.** OPTIMRNA achieves the strongest zero-shot secondary-structure prediction among the evaluated RNA language models and competes with substantially larger models under supervised fine-tuning and cross-family generalization. Controlled ablations isolate the contributions of the persistent pair pathway and antidiagonal propagation, while layerwise analyses characterize how structural information is organized across token, attention, and pair representations.
- **Transferability.** On tasks whose dependence on secondary structure is more indirect, OPTIMRNA shows no systematic degradation relative to matched token-only baselines. These results indicate that the structural inductive bias improves structure-sensitive representations without sacrificing broader sequence information.

2 BACKGROUND AND RELATED WORK

Transformer-based RNA Language models. Standard RNA-LMs such as AIDO.RNA (Zou et al., 2024), RiNALMo (Penić et al., 2025) and RNA-FM (Chen et al., 2022) use BERT-style architectures trained with masked language modeling on large RNA sequence databases such as RNACentral (The RNAcentral Consortium, 2026) and Rfam (Ontiveros-Palacios et al., 2025). These models mainly differ in their encoder size and pretraining data rather than their architectural components. Given an RNA sequence of N nucleotides, denoted $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, they first assign it an initial token embedding matrix $\boldsymbol{X}^0 \in \mathbb{R}^{N\times d}$. Then a Transformer encoder produces contextual token representations $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^L$, where $\boldsymbol{X}^\ell \in \mathbb{R}^{N\times d}$. For attention head h at layer ℓ, the input $\boldsymbol{X}^\ell$ is normalized before being projected into queries, keys, and values:

$$\boldsymbol{H}^\ell = \mathrm{LN}(\boldsymbol{X}^{\ell-1}); \quad \boldsymbol{Q}^{\ell,h} = \boldsymbol{H}^\ell \boldsymbol{W}_Q^{\ell,h}; \quad \boldsymbol{K}^{\ell,h} = \boldsymbol{H}^\ell \boldsymbol{W}_K^{\ell,h}; \quad \boldsymbol{V}^{\ell,h} = \boldsymbol{H}^\ell \boldsymbol{W}_V^{\ell,h}. \tag{1}$$

The conventional attention logits and head output are

$$\boldsymbol{Z}^{\ell,h} = \frac{\boldsymbol{Q}^{\ell,h}(\boldsymbol{K}^{\ell,h})^\top}{\sqrt{d_h}} + \boldsymbol{M}; \quad \boldsymbol{O}^{\ell,h} = \mathrm{softmax}_j(\boldsymbol{Z}^{\ell,h})\boldsymbol{V}^{\ell,h}, \tag{2}$$

where $\boldsymbol{M}$ denotes the padding and attention mask. The head outputs are concatenated and combined with the output projection, residual connection, and feed-forward component to obtain $\boldsymbol{X}^{(\ell)}$. Although attention logits $\boldsymbol{Z}^{\ell,h} \in \mathbb{R}^{N\times N}$ are indexed by pairs of positions, they are transient: they are computed, then used to route information between tokens, and discarded. What is transferred

from one layer to the next is only the updated token states $\boldsymbol{X}^\ell \in \mathbb{R}^{N\times d}$. Any structural information must therefore be folded into per-token embeddings, which may constrain how an RNA-LM can represent and refine the underlying structure of a sequence.

Incorporating pairwise biology into RNA models. Known RNA pairing rules offer ways to inform learning from sequences. Existing approaches have incorporated biological knowledge through pretraining objectives, via motif-aware masking (Wang et al., 2024), structure-aware masking (Zhu et al., 2026) or direct structural supervision (Chu et al., 2024). These approaches differ in what information they provide to the model and whether they require structural information during training. Our interest is in introducing pairing constraints while retaining sequence-only MLM as the training objective. ERNIE-RNA is a particularly relevant precedent. It initializes attention with a bias derived from potential nucleotide pairing and uses the preceding layer's attention map to bias subsequent attention layers (Yin et al., 2025). This carries pairwise information between layers, but ties it to the attention computation and requires a pair-indexed attention map to be available for propagation. Residual connections between attention scores have also been studied outside RNA modeling, notably in RealFormer (He et al., 2021).

A distinct approach is to maintain a pair representation as a model state. AlphaFold's Evoformer and Pairformer refine such representations under direct structural supervision (Jumper et al., 2021; Abramson et al., 2024). RNAformer likewise operates on a two-dimensional representation for supervised RNA secondary-structure prediction (Franke et al., 2024). Dense pair representations also incur memory costs that grow quadratically with sequence length; some commonly used pair-update operations require cubic computation. These examples establish the value of explicitly updating pairwise evidence, but their training setting differs from an RNA-LM supervised only through masked nucleotides. OptimRNA follows the latter, but introduces a dedicated pair state and refines the base-pairing priors to only likely stems in the secondary structure.

3 OptimRNA: Efficient pairwise representations

OptimRNA augments the conventional token stream $\boldsymbol{X}^0, \ldots, \boldsymbol{X}^L$, defined in Section 2, with a pair state that is accumulated over layers $\boldsymbol{P}^0, \ldots, \boldsymbol{P}^L$, where $\boldsymbol{P}^\ell \in \mathbb{R}^{N\times N}$ associates a scalar pair score with every token pair. This pair state is seeded with a biologically-motivated prior, and the model progressively refines this initial state by extracting low-rank pairwise information from contextual token representations, propagating it along stem-compatible antidiagonals, and accumulating the resulting updates across layers. The refined state is read by subsequent layers as a bounded attention bias. We define these operations in detail and summarize them in Figure 1.

3.1 Pair state and antidiagonal convolution

Pair state initialization. The pair state is initialized using a base-complementarity prior. Let $C_{ij} = \mathbf{1}[(x_i, x_j) \in \mathcal{W}]$, with $\mathcal{W} = \{\mathrm{GC}, \mathrm{CG}, \mathrm{AU}, \mathrm{UA}\}$ the Watson–Crick pairs, indicate that nucleotides i and j are complementary. Stacked base pairs lie along antidiagonals, $(i, j) \to (i{+}1, j{-}1)$, so we call the pair (i, j), $i < j$, a *candidate stem* pair if it lies on a run of at least k consecutive complementary pairs along its antidiagonal:

$$P^0_{ij} = P^0_{ji} = \alpha\,\Pi_{ij}, \quad \Pi_{ij} = \max_{0\le a<k} \prod_{t=0}^{k-1} C_{\,i-a+t,\; j+a-t}\, \mathbf{1}[\,i-a+t < j+a-t\,], \tag{3}$$

where out-of-range indices contribute 0, $k = 4$, and $\alpha = 5$ is a fixed scalar modulating the strength of the prior. Equivalently, Π_{ij} is 1 if and only if (i, j) belongs to a stem of 4 or more Watson–Crick. This prior does not specify the secondary structure: it identifies candidate stems, many of which are spurious. In the folded RNA, most of them will not materialize, so it is up to the model to distinguish them through pretraining and recognize other base pairs which are not contained in $\boldsymbol{P}^0$.

A shared latent pair geometry. We hypothesize that pairwise dependencies arising across RNA sequences and layers can be expressed through a shared geometry that respects the symmetry of RNA structure. Accordingly, all layers add to the pair state through one shared symmetric, low-rank, bilinear form. Only its input, the layer's token representations, changes from one layer to the

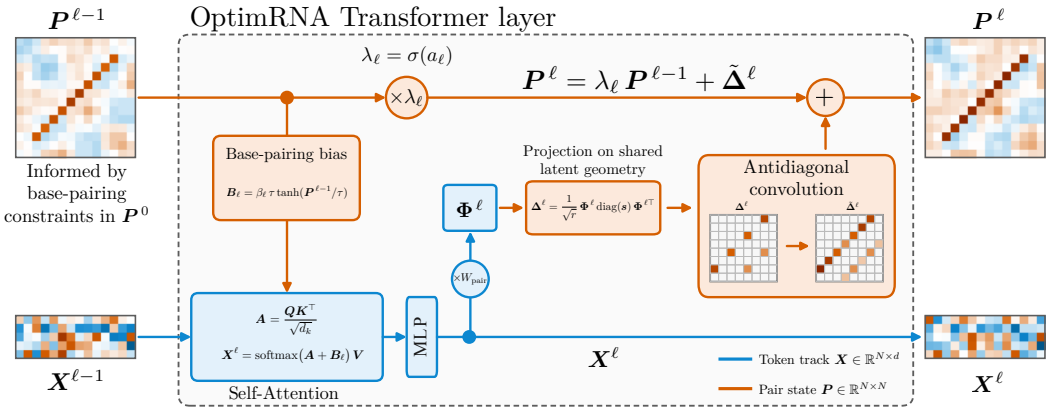


Figure 1: Mechanism of the factorized pairwise representation with antidiagonal convolution

next. We parameterize the shared geometry as

$$\boldsymbol{G} = \frac{1}{\sqrt{r}} \boldsymbol{W}_{\text{pair}}^{\top} \operatorname{diag}(\boldsymbol{s}) \boldsymbol{W}_{\text{pair}} \in \mathbb{R}^{d\times d}, \tag{4}$$

where $\boldsymbol{W}_{\text{pair}} \in \mathbb{R}^{r\times d}$ and $\boldsymbol{s} \in \mathbb{R}^{r}$ are learned and shared across layers with $r = 32$. Given the token representations normalized by a LayerNorm shared across layers, $\boldsymbol{U}^{\ell} = \text{LN}_{\text{pair}}(\boldsymbol{X}^{\ell})$, the pairwise write produced by layer ℓ is

$$\boldsymbol{\Delta}^{\ell} = \boldsymbol{U}^{\ell} \boldsymbol{G} (\boldsymbol{U}^{\ell})^{\top} = \frac{1}{\sqrt{r}} \boldsymbol{\Phi}^{\ell} \operatorname{diag}(\boldsymbol{s}) (\boldsymbol{\Phi}^{\ell})^{\top}, \qquad \boldsymbol{\Phi}^{\ell} = \boldsymbol{U}^{\ell} \boldsymbol{W}_{\text{pair}}^{\top} \in \mathbb{R}^{N\times r}. \tag{5}$$

Thus, each layer expresses its contextual pairwise evidence in a common r-dimensional feature space. By construction, $\boldsymbol{G}$ and $\boldsymbol{\Delta}^{\ell}$ are symmetric and have rank at most r, providing a compact pairwise write rather than a complete prediction of the RNA secondary structure.

Signed low-rank pairwise writes. A single RNA base pair has the zero-diagonal interaction pattern $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, with one positive and one negative eigenvalue. A positive-semidefinite write cannot express this pattern exactly. We therefore use signed component weights $\boldsymbol{s}$, initializing half as positive and half as negative, so that the shared geometry $\boldsymbol{G}$ can provide both types of direction to the layerwise writes $\boldsymbol{\Delta}^{\ell}$. This initialization does not prescribe the spectrum of any particular write except if $\boldsymbol{W}_{\text{pair}}$ and $\boldsymbol{\Phi}^{\ell}$ are full rank, as detailed in Lemma 2 in the Appendix. Empirically, the signs and magnitudes of $\boldsymbol{s}$ remain stable during training, preserving this signed capacity across layers.

Antidiagonal convolution. Base pairs belonging to the same RNA stem occur at neighboring coordinates $(i, j), (i+1, j-1), (i+2, j-2), \ldots$ and therefore form antidiagonal patterns in a pair map. We encode this structural regularity through

$$\mathcal{A}_R(\boldsymbol{\Delta}^{\ell})_{ij} = \sum_{q=-R}^{R} w_q \boldsymbol{\Delta}^{\ell}_{i+q,j-q}, \tag{6}$$

where R is the propagation radius, w_q are learned coefficients, and out-of-range entries are set to zero. We use $R = 4$ to match our seed prior. We impose $w_q = w_{-q}$ to preserve symmetry. Importantly, this operation does not introduce a different latent geometry at each offset: every shifted component uses the same shared $\operatorname{diag}(\boldsymbol{s})$, but evaluates it at neighboring pairs of positions.

Decayed running sum of the pair state. $\mathcal{A}_R(\boldsymbol{\Delta}^{\ell})$ is incorporated into $\boldsymbol{P}^{\ell}$ through a residual-style recurrence

$$\boldsymbol{P}^{\ell} = \lambda_\ell \boldsymbol{P}^{\ell-1} + \mathcal{A}_R(\boldsymbol{\Delta}^{\ell}), \qquad \lambda_\ell = \sigma(a_\ell), \tag{7}$$

where a_ℓ is learned and its by-product $\lambda_\ell \in (0, 1)$ controls the retention of previously accumulated evidence. It is initialized at $\lambda_\ell = 0.97$. This update allows each layer to refine the existing pair

state rather than replace it, while directly preserving information written at preceding layers. The convolution can raise the rank of Δ^ℓ from r to at most $(2R+1)r$, and the accumulation across layers raises it further, so $\boldsymbol{P}^\ell$ can approach full rank while remaining a sum of low rank factors. The corresponding factorization, rank bounds and conditions under which these bounds are attained are provided in Appendix D.8.

Pair-biased attention. To bound the effect of the pair state on the attention mechanism, we apply a nonlinearity before using the pair state to bias attention:

$$\boldsymbol{O}^{\ell,h} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}^{\ell,h}(\boldsymbol{K}^{\ell,h})^\top}{\sqrt{d_h}} + \boldsymbol{M} + \boldsymbol{B}^\ell\right)\boldsymbol{V}^{\ell,h}, \quad \text{with } \boldsymbol{B}^\ell = \beta_\ell \tau \tanh\left(\frac{\boldsymbol{P}^{\ell-1}}{\tau}\right), \tag{8}$$

where $\boldsymbol{M}$ is the padding mask. The fixed temperature of the tanh $\tau > 0$ and bias term β_ℓ bound the bias $|\boldsymbol{B}^\ell_{ij}| \le |\beta_\ell|\tau$ regardless of the magnitude of $\boldsymbol{P}$. This prevents huge attention bias magnitudes as pairwise writes accumulate across depth. The coefficient β_ℓ is learned per layer, and the bias term $\boldsymbol{B}^\ell$ is shared across attention heads. We use $\tau = 8$.

Every layer adds updates to the pair state, but only the last five of the twelve layers read it ($\ell = 8, \ldots, 12$): the first seven layers accumulate pairwise evidence without being biased by it, and the remaining layers attend under the bias $\boldsymbol{B}^\ell$. The early layers therefore build the state from increasingly contextualized token representations before any layer consumes it, which is the regime we analyse in Section 4.2.

3.2 Memory efficient implementation

OptimRNA is implemented to minimize its memory consumption: the only quadratic object is the pair state itself, which is one $N \times N$ object per layer. Attention maps are never fully materialized thanks to Flash Attention (Dao et al., 2022) and Flex Attention (Dong et al., 2025).

Pair-biased attention We compute the pair-biased attention (equation 8) using FlexAttention which adds the bias term $\boldsymbol{B}^\ell$ to each tile of scores inside a fused, tiled attention kernel which also handles masking of padding tokens. For the pair-biased layers, the attention maps are thus never fully materialized, only the bias term needs to be, which is only one $N \times N$.

Fused pair-state update Because the pair state keeps its sum-of-low-rank form, its updates can be applied in the low-rank form, which reduces the amount of $N \times N$ intermediate objects needed to be stored in memory for backpropagation. In particular, the low rank update combined with the antidiagonal convolution (equation 6) is equivalent to the product of two $N \times (2R+1)r$ matrices, each consisting of $2R+1$ shifted, rescaled copies of $\boldsymbol{\Phi}^\ell$.

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell) = \frac{1}{\sqrt{r}}\,\boldsymbol{\Phi}^\ell_A\,(\boldsymbol{\Phi}^\ell_B)^\top, \qquad \boldsymbol{P}^\ell = \lambda_\ell \boldsymbol{P}^{\ell-1} + \frac{1}{\sqrt{r}}\,\boldsymbol{\Phi}^\ell_A\,(\boldsymbol{\Phi}^\ell_B)^\top. \tag{9}$$

where $\boldsymbol{\Phi}^\ell_A = \big[\, w_k\, \mathcal{S}_k(\boldsymbol{\Phi}^\ell)\,\mathrm{diag}(\boldsymbol{s})\big]_{-R\le k\le R}$ and $\boldsymbol{\Phi}^\ell_B = \big[\,\mathcal{S}_k(\boldsymbol{\Phi}^\ell)\big]_{R\ge k\ge -R}$, denoting $\mathcal{S}_q$ the operator that shifts the rows of a matrix by q, as $(\mathcal{S}_q\boldsymbol{\Phi})_i = \boldsymbol{\phi}_{i+q}$. The rank is thus at most $(2R+1)r$ and this implementation requires no $N \times N$ intermediate: each layer's update is a single batched matrix product added to $\lambda_\ell \boldsymbol{P}^{\ell-1}$. $\boldsymbol{\Phi}^\ell_A$ and $\boldsymbol{\Phi}^\ell_B$ differ, but the tied convolution weights $w_q = w_{-q}$ keep the product symmetric. In the backward pass, only $\boldsymbol{\Phi}^\ell$ and $\boldsymbol{P}^{\ell-1}$ are stored, we recompute $\boldsymbol{\Phi}^\ell_A$ and $\boldsymbol{\Phi}^\ell_B$ since they are cheaply computed from $\boldsymbol{\Phi}$ and consume considerable memory. As an extension to this approach, we provide a Flash Attention style kernel which recomputes $\boldsymbol{P}^\ell$ in a tiled fashion without ever fully materializing it, leading to linear memory complexity, which can be particularly useful for extensions to mRNAs or lncRNAs. We provide more detail in Appendix C.2.

4 Results

In the following section, we first evaluate OptimRNA on zero-shot and fine-tuned structure prediction, including generalization across RNA families. We then use controlled ablations and layerwise analyses to assess how its structural components affect prediction and learned representations. Next, we examine performance on tasks less directly tied to secondary structure. Finally, we quantify the

pair pathway's computational overhead and compare throughput and memory use with similarly sized RNA-LMs.

4.1 SECONDARY STRUCTURE PREDICTION

Experimental setup. We first evaluate zero-shot secondary-structure prediction on bpRNA TS0 using nucleotide dependency, which scores candidate base pairs from changes in model predictions under nucleotide perturbations (Tomaz da Silva et al., 2025). Applying the same pipeline to all RNA-LMs assesses structural information available without training a downstream predictor. We report both AUPRC, which evaluates pair ranking independently of score scale and threshold choice, and F1 at a validation-selected threshold to cope with current practice in the field. F1 additionally depends on how sharply precision and recall vary near that threshold and how well it transfers to test data, therefore is likely less robust than AUPRC.

We then evaluate supervised adaptation on bpRNA, fine-tuning on TR0 and testing on TS0 (Singh et al., 2019), and cross-family generalization on ArchiveII (Sloma & Mathews, 2016), systematically training on eight families and testing on the ninth. Following RiNALMo and AIDO.RNA, the prediction head uses final-layer token embeddings; our +P variant additionally receives OptimRNA's pair state. We report shift-tolerant F1 with validation-selected thresholds, averaged across held-out families for ArchiveII.

We harmonize evaluation procedures where possible and distinguish re-evaluated checkpoints from published scores. We also retrain RiNALMo-micro with our pretraining recipe to assess the influence of training choices. Full protocols are provided in Appendix D.

Structural information before fine-tuning. OptimRNA-88M achieves the highest zero-shot AUPRC and F1 among the evaluated RNA-LMs, outperforming substantially larger models, including RiNALMo-giga and AIDO.RNA (Figure 2, Table 1). ERNIE-RNA, which also incorporates biological pairing priors, is the closest competitor. However, OptimRNA-34M already surpasses its AUPRC and matches its F1 with approximately 40% of its parameters, while OptimRNA-88M improves both metrics at a comparable model size and at a significantly lower computational cost (see details in Section 4.4). Together, these results establish OptimRNA's strong performance–parameter trade-off for recovering RNA structure without downstream supervision.

Supervised adaptation and cross-family transfer. OptimRNA retains a favorable performance–parameter trade-off after supervised adaptation, outperforming RiNALMo-giga and approaching AIDO.RNA on bpRNA despite their substantially larger parameter counts (Table 1). These comparisons follow established fine-tuning procedures, although differences in prediction heads and optimization sensitivity mean that they reflect performance under the evaluated settings rather than each model's optimal adaptation. ERNIE-RNA achieves the best performance, but it uses its own attention-map head, making the comparison difficult. Interestingly, within OptimRNA, adding the pair state to the token-based prediction head yields only marginal gains on bpRNA but substantially improves cross-family generalization on ArchiveII. The +P readout increases F1 by 4.5 and 5.6 points for the 34M and 88M models, respectively, with both exceeding the reported results of the evaluated baselines. The consistent benefit across scales suggests that explicit access to the learned pair representation is particularly useful when predicting structures from families excluded from downstream training.

Impact of the pretraining recipe. Retraining RiNALMo-micro with our pretraining recipe substantially improves its zero-shot structure prediction. This gain may reflect the increased exposure to underrepresented RNA types through balanced sampling, together with the use of Muon optimization (see Appendix B). Nevertheless, both OptimRNA models outperform the retrained RiNALMo-micro, with OptimRNA-34M retaining a clear advantage at a comparable parameter count. OptimRNA's performance therefore extends beyond the improvements obtained by applying our training recipe to an existing RNA-LM.

4.2 CONTRIBUTIONS OF THE STRUCTURAL PATHWAY

OptimRNA performs competitively across the structure prediction benchmarks, but comparisons with published models do not permit systematic isolation of the effect of its structural components. In particular, retraining RiNALMo-micro revealed a substantial effect of the pretraining

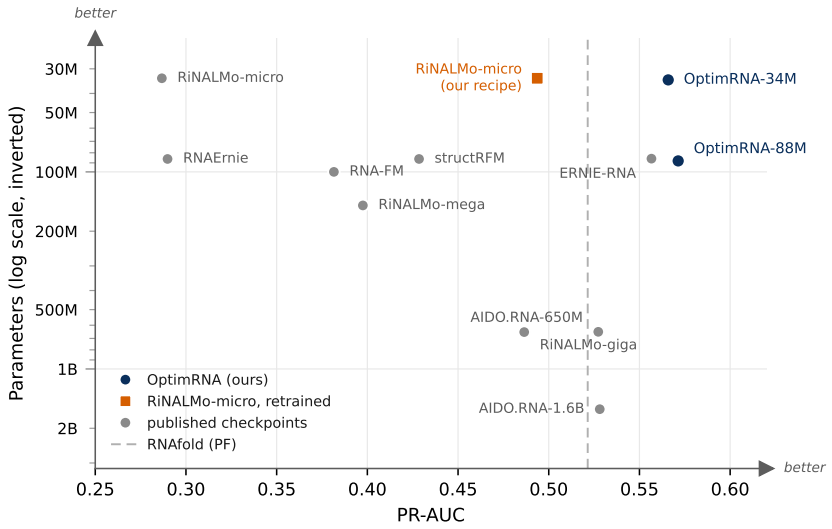


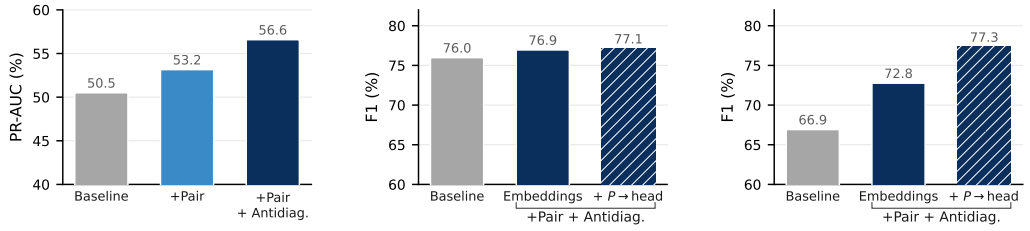
Figure 2: PR-AUC on bpRNA TS0 vs number of parameters for different RNA-LMs.

Table 1: **Secondary-structure prediction.** F1 (%) on zero-shot (ZS) bpRNA TS0, and shift-tolerant F1 (%) on fine-tuned (FT) bpRNA TS0 and ArchiveII leave-one-family-out (mean over the nine held-out families). All ZS cells are measured with our probe; †transcribed from the original publication; ‡released fine-tuned checkpoint re-scored under our protocol. Best in bold, second underlined.

Model	Params	bpRNA TS0		ArchiveII (LOFO)
		ZS	FT	
ERNIE-RNA	85.7M	<u>51.7</u>	**79.1**‡	–
RNA-FM	100M	33.6	67.6†	29.8†
RiNALMo-giga	650M	44.1	74.7†	72.0†
AIDO.RNA	1.6B	50.9	<u>78.3</u>†	73.3†
RiNALMo-micro (our recipe)	33.5M	45.7	76.4	69.5
OptimRNA-34M	34M	<u>51.7</u>	76.9	72.8
+ $P \rightarrow$ head			77.1	**77.3**
OptimRNA-88M	88M	**52.5**	77.6	70.7
+ $P \rightarrow$ head			77.6	<u>76.3</u>

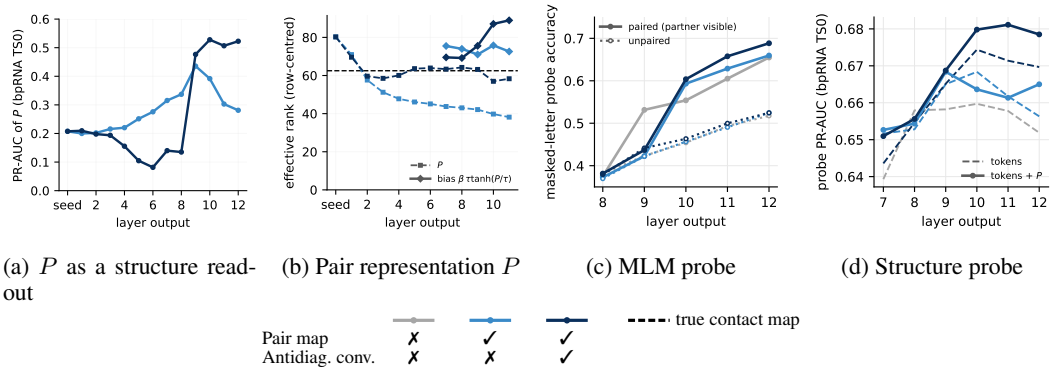
recipe, while supervised results also depend on the adaptation protocol. We therefore compare three 34M models pretrained and evaluated on structure prediction: *i)* a matched token-only backbone; *ii)* a variant with the pair state but without antidiagonal propagation; *iii)* the complete OptimRNA model. They are evaluated under the same conditions, in zero-shot for all and fine-tuning within- and across-family settings for the baseline and the complete OptimRNA model.

Where the structural pathway matters. We report comprehensive performance of structure prediction tasks in Figure 3. In the matched zero-shot comparison, the pair state improves structure prediction, and antidiagonal propagation adds a further gain. Both components therefore help the model recover pairing information before it sees structural labels. The gains are present but more modest after fine-tuning on bpRNA, where a prediction head can learn from annotated structures drawn from the same benchmark distribution. However, under ArchiveII family holdout, the full model improves even when the head reads only token embeddings, and giving it direct access to the pair state yields a larger gain. The clearest benefit of the explicit pair representation thus emerges when the downstream predictor must generalize to an RNA family excluded from its structural training data.



(a) Zero-Shot, bpRNA-TS0 (b) Fine-tuning, bpRNA-TS0 (c) Fine-Tuning, ArchiveII (LOFO)

Figure 3: Structure prediction performance of OptimRNA (+Pair+Antidiag) and ablated models without the antidiagonal convolution (+Pair), and without the pair state entirely (Baseline).



(a) P as a structure readout (b) Pair representation P (c) MLM probe (d) Structure probe

Figure 4: Performance of $\boldsymbol{P}$ as a structure readout, effective rank of $\boldsymbol{P}$ and the attention bias $\boldsymbol{B}$ and performance of a MLM probe and a structure probe trained on TR0, across layers.

Structural information transfers from Pair state to token stream. Across layers, we report in Figure 4 the performance of P as a structure predictor, the pair state and its attention bias' effective rank, the performance of an MLM probe and a structure probe trained on token embeddings only or jointly with the pair state. We observe that $\boldsymbol{P}$ becomes an immediate structure readout, in particular in the late layers. This is reflected in the effective rank of the bias term $\boldsymbol{B}^{\ell}$, in particular for the model equipped with the antidiagonal convolution. Layer 9 is where this structure emerges: the structure readout of its output pair state jumps. It is then read by layer 10, where the pair-state models start making structure-based predictions: at this layer's output, the MLM probe's accuracy on masked nucleotides increases substantially for paired nucleotides whose partner is visible. Token embeddings contain this structural information: the structure probe trained on token-only peaks at layer 10. Still, $\boldsymbol{P}$ brings additional information, based on the probe's increased performance when using $\boldsymbol{P}$ as an additional input.

4.3 Transfer beyond secondary-structure prediction

A structural inductive bias is broadly useful if it improves structure prediction without weakening sequence representations. Splice sites and nucleotide-modification sites are primarily identified through local sequence context, while RNA structure can further influence their recognition. These tasks therefore test whether OptimRNA preserves information beyond the pairwise interactions it is designed to model. We compare ablated variants of OptimRNA, and RiNALMo-micro pretrained with the same recipe, as well as ERNIERNA as our conceptually closest competitor.

Experimental protocol. For modification prediction, we fit linear probes to frozen token embeddings on the MultiRM datasets (Song et al., 2021). We select the embedding layer on validation data and report test AUPRC averaged across five modification types (Table 2). For splice-site prediction, we fine-tune separate donor and acceptor predictors using the Spliceator benchmark (Scalzitti et al., 2021) and report AUPRC averaged across the evaluated species (Table 2). Per-modification and per-species results, together with broader model comparisons, are provided in Appendix D.4.

Table 2: **Splice-site and modification prediction.** Splice site: mean ± sd over 3 fine-tuning seeds; ‡released fine-tuned weights re-scored under our protocol (a single measurement). Best in bold.

Model	Pair state	PR-AUC (%)	
		Splice site	Modification
RiNALMo-giga (650M)	–	**98.92**‡	90.2
RiNALMo-micro (33.5M)	–	97.24 ±0.66	**91.4**
RiNALMo-micro (33.5M, our pretraining)	–	98.36 ±0.18	91.2
ERNIE-RNA (85.7M)	–	98.13 ±0.03	88.8
OptimRNA-34M (ours)	✗	98.16 ±0.13	88.0
	✓	98.05 ±0.24	89.6
OptimRNA-88M (ours)	✗	97.82 ±0.33	88.8
	✓	98.19 ±0.17	89.4

Results. The structural pathway causes no consistent deterioration relative to the matched token-only backbone, although its effects differ across tasks and model scales. Coupled with its gains in structure prediction, these findings suggest that explicit pairing information can improve an RNA-LM's overall range of capabilities without confining its representations to structural tasks.

4.4 Memory consumption and throughput

Table 3 reports the median time per step and memory consumption of our 88M model and comparably-sized ERNIE-RNA on 200 training steps with real sequences, with 30 steps of warmup. The pair state doubles OptimRNA's training time and increases its memory usage by 1.5×. However it still uses 19% less time per step and 4× less memory than ERNIE-RNA, while achieving comparable zero-shot performance. ERNIE-RNA's pair propagation materializes full attention maps at every layer, which is memory costly. Overall, our method enables the training of models with a pair state at a considerably lower memory footprint, enabling bigger batches and faster training. In practice, OptimRNA-88M with the pair state was trained on a single H100 GPU for less than 40 hours.

Table 3: **Training-step cost** (one H100, 48 sequences per step).

Model	Pair	s / step	peak GiB
ERNIE-RNA	–	0.233	68.6
OptimRNA-88M	✗	0.083	10.2
	✓	0.189	15.8

5 Conclusion

We introduced OptimRNA, which incorporates RNA pairing knowledge into sequence-only masked language modeling through a pair state refined across Transformer layers. It achieves the strongest zero-shot secondary-structure prediction among the evaluated RNA-LMs. Compared with ERNIE-RNA, its closest architectural competitor, OptimRNA-88M improves prediction while requiring less training time and GPU memory. After fine-tuning, OptimRNA competes with much larger models, and direct access to its pair state substantially improves cross-family structure prediction. Matched ablations support the contributions of both pair-state modeling and stem-aware propagation, without showing a negative transfer on tasks less related to RNA structure.

Our study also revealed how strongly the pretraining recipe affects structural performance. A useful next step will be to study sampling, automated data curation, and structural priors together to identify which choices produce the most transferable RNA representations. At the architectural level, antidiagonal propagation creates stem-aligned variations of one shared latent pair geometry, drastically augmenting the representational capacities of our model. Nonetheless, limitations may remain, and future models could investigate modeling several latent geometries according to sequence context, potentially using evolutionary information from multiple sequence alignments when available. This would test whether different RNA families or interaction patterns benefit from distinct pairwise relations while retaining efficient pair-state updates.

AI USE STATEMENT

In this work, we used generative AI tools to:

1. Design or provide feedback on research methodology or experiments. The core ideas of all experiments are human generated. We use AI to improve the rigor of experiments.
2. Implement methods. We use coding agents for implementation. We check all code and take full responsibility for its correctness.
3. Support qualitative and thematic data analysis. We use agents to help with making plots and visualizations.
4. Assist in the writing of proofs. All mathematical claims and proofs were checked by the authors, and core ideas were developed by the authors.
5. Interpret results. AI was used to help create intermediary plots and tables to help with interpretation.
6. Draft sections of the paper. We used AI to help rephrase some sentences for clarity, which were then reworked by the authors, and to create LaTeX code for the methods section. No section was entirely drafted by AI.

We have not used generative AI tools to:

1. Help develop theoretical models or conceptual frameworks.
2. Propose or refine hypotheses. The idea that a pair state can help with structural learning in RNA-LM originates from the authors.
3. Clean and reformat datasets.
4. Formulate mathematical claims. AI was not used to formulate claims: Lemma 1 is a standard consequence of Sylvester's law of inertia, and the intuition behind Lemma 2 and the rank capacity of the pair state under antidiagonal convolution and residual accumulation originated from the authors.
5. Provide critical ingredients for proving mathematical claims. The core idea behind the mathematical proofs originated from the authors, AI helped write the proofs.

These required disclosure tasks are not applicable to this work:

1. Generate synthetic data sets.
2. Assist with translation.

Additionally, we used generative AI tools for:

1. Editing the research paper to improve readability. We use agents to find and correct grammar mistakes and point out sections which require more human editing.
2. Identifying relevant literature. We use agents to perform an additional literature search on top of our human search.

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.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate

structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, June 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w.

Xinang Cao, Yueying Zhang, Yiliang Ding, and Yue Wan. Identification of RNA structures and their roles in RNA functions. *Nature Reviews Molecular Cell Biology*, 25(10):784–801, October 2024. ISSN 1471-0080. doi: 10.1038/s41580-024-00748-6.

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, August 2022. arXiv:2204.00300 [q-bio].

Yanyi Chu, Dan Yu, Yupeng Li, Kaixuan Huang, Yue Shen, Le Cong, Jason Zhang, and Mengdi Wang. A 5' UTR language model for decoding untranslated regions of mRNA and function predictions. *Nature Machine Intelligence*, 6(4):449–460, April 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00823-9.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359. Curran Associates, Inc., 2022.

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.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flexattention: A programming model for generating fused attention variants. In *Eighth Conference on Machine Learning and Systems*, 2025.

Jörg K. H. Franke, Frederic Runge, Ryan Köksal, Dominika Matus, Rolf Backofen, and Frank Hutter. RNAformer: A Simple yet Effective Model for Homology-Aware RNA Secondary Structure Prediction, November 2024. Pages: 2024.02.12.579881 Section: New Results.

Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. RealFormer: Transformer likes residual attention. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 929–943, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.81.

Donna K. Hendrix, Steven E. Brenner, and Stephen R. Holbrook. RNA structural motifs: building blocks of a modular biomolecule. *Quarterly Reviews of Biophysics*, 38(3):221–243, August 2005. ISSN 1469-8994, 0033-5835. doi: 10.1017/S0033583506004215.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Nancy Ontiveros-Palacios, Emma Cooke, Eric P Nawrocki, Sandra Triebel, Manja Marz, Elena Rivas, Sam Griffiths-Jones, Anton I Petrov, Alex Bateman, and Blake Sweeney. Rfam 15: RNA families database in 2025. *Nucleic Acids Research*, 53(D1):D258–D267, January 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1023.

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, July 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-60872-5.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, September 2007.

Nicolas Scalzitti, Arnaud Kress, Romain Orhand, Thomas Weber, Luc Moulinier, Anne Jeannin-Girardon, Pierre Collet, Olivier Poch, and Julie D. Thompson. Spliceator: multi-species splice site prediction using convolutional neural networks. *BMC Bioinformatics*, 22(1):561, November 2021. ISSN 1471-2105. doi: 10.1186/s12859-021-04471-3.

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, November 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13395-9.

Michael F. Sloma and David H. Mathews. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA*, 22(12):1808–1818, December 2016. ISSN 1355-8382, 1469-9001. doi: 10.1261/rna.053694.115.

Zitao Song, Daiyun Huang, Bowen Song, Kunqi Chen, Yiyou Song, Gang Liu, Jionglong Su, João Pedro de Magalhães, Daniel J. Rigden, and Jia Meng. Attention-based multi-label neural networks for integrated prediction and interpretation of twelve widely occurring RNA modifications. *Nature Communications*, 12(1):4011, June 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-24313-3.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063.

The RNAcentral Consortium. RNAcentral in 2026: genes and literature integration. *Nucleic Acids Research*, 54(D1):D303–D313, January 2026. ISSN 1362-4962. doi: 10.1093/nar/gkaf1329.

Pedro Tomaz da Silva, Alexander Karollus, Johannes Hingerl, Gihanna Sta Teresa Galindez, Nils Wagner, Xavier Hernandez-Alias, Danny Incarnato, and Julien Gagneur. Nucleotide dependency analysis of genomic language models detects functional elements. *Nature Genetics*, 57(10):2589–2602, October 2025. ISSN 1546-1718. doi: 10.1038/s41588-025-02347-3.

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*, pp. 1–10, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00836-4.

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Cunxiao Du, Chao Du, Tianyu Pang, Zhuoran Yang, Mingyi Hong, and Vincent Tan. Muon outperforms adam in tail-end associative memory learning. In *International Conference on Learning Representations*, volume 2026, pp. 37382–37419, 2026.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.127.

T. Xia, J. SantaLucia, M. E. Burkard, R. Kierzek, S. J. Schroeder, X. Jiao, C. Cox, and D. H. Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37(42):14719–14735, October 1998. ISSN 0006-2960. doi: 10.1021/bi9809425.

Weijie Yin, Zhaoyu Zhang, Shuo Zhang, Liang He, Ruiyang Zhang, Rui Jiang, Gan Liu, Jingyi Wang, Xuegong Zhang, Tao Qin, and Zhen Xie. ERNIE-RNA: an RNA language model with structure-enhanced representations. *Nature Communications*, 16(1):10076, November 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-64972-0.

Luciano I Zablocki, Leandro A Bugnon, Matias Gerard, Leandro Di Persia, Georgina Stegmayer, and Diego H Milone. Comprehensive benchmarking of large language models for RNA secondary structure prediction. *Briefings in Bioinformatics*, 26(2):bbaf137, March 2025. ISSN 1477-4054. doi: 10.1093/bib/bbaf137.

Zaixi Zhang, Ruofan Jin, Linlin Chao, Guangxue Xu, Yikun Zhang, Guowei Zhou, Di Yin, Yingqing Guo, Yaqi Fu, Yukang Yang, Kaixuan Huang, Xiaotong Wang, Junze Zhang, Yujie Yang, Qirong Yang, Ziyao Xu, E. Weinan, Ruhong Zhou, Xiaoming Zhang, Mengdi Wang, and Le Cong. RNA-Genesis: A Generalist Foundation Model for Functional RNA Therapeutics, July 2025. ISSN: 2692-8205, 2692-8205 Pages: 2024.12.30.630826 Section: New Results.

Heqin Zhu, Ruifeng Li, Ao Chang, Haobin Chen, Feng Zhang, Fenghe Tang, Tong Ye, Xin Li, Yunjie Gu, Peng Xiong, and S. Kevin Zhou. A fully open structure-guided RNA foundation model for robust structural and functional inference, May 2026. ISSN: 2692-8205 Pages: 2025.08.06.668731 Section: New Results.

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, November 2024. Pages: 2024.11.28.625345 Section: New Results.

Table 4: **Backbone comparison.** Three 34M backbones pretrained with the same recipe on the same corpus (Appendix B.2); none has the pair map. Validation MLM loss at step 200k on the held-out RNAcentral split; zero-shot bpRNA TS0 structure prediction with the probe of Appendix D.1 (PR-AUC); training-step cost on real RNAcentral batches (192 sequences of up to 1024 nt median over 200 steps): wall time per step, throughput in non-padding tokens per second, and peak memory. Best in bold.

		Pretraining	Zero-shot TS0	Training step (B=192)		
Model	Params	Val. loss	ZS PR-AUC	s / step	ktok / s	peak GiB
RiNALMo-micro (retrained)	33.5M	0.6606	0.494	0.308	261	36.8
OptimRNA-34M (no ConvNeXt)	33.4M	0.6593	0.493	**0.170**	**474**	**22.6**
OptimRNA-34M, baseline	34.1M	**0.6549**	**0.505**	0.189	426	24.0

A Choice of backbone

Our backbone uses a ModernBERT (Warner et al., 2025) backbone which combines local attention and global attention as well as gated MLP. We use RoPE positional encodings Su et al. (2024). In Table 4, we compare the performance of an OptimRNA backbone and a RiNALMo-micro backbone which we retrained with our recipe. On validation loss and zero-shot structure prediction, these models are equivalent, while OptimRNA consumes less memory and is faster. RNAGenesis (Zhang et al., 2025) uses convolution layers before its first transformer layer. Following this, we add ConvNext (Liu et al., 2022) layers before our first transformer encoder layer, and we use this as our baseline. Their cost in time and memory is low, and they help achieve lower validation loss and better zero-shot structure prediction. This latter version is what we use as our baseline.

B Pretraining

Training procedures for RNA Language Models vary and there is no consensus on how to train them optimally. RiNALMo, AIDO.RNA, ERNIERNA, RNA-FM all use RNACentral as their main pretraining dataset and a masked language modeling objective is generally used with 15% of nucleotides masked, but models vary in size, training hyperparameters and data sampling. We decide to train our model with a Masked Language Modeling objective with 15% of nucleotides masked, as this has proven to work well for other models. We use RNACentral's family labels to under-sample over-represented families, and over-sample under-represented ones, and we use the Muon optimizer. The latter may play a huge role in the performance of our baseline model as recent work shows that it not only improves convergence but also helps learning more effectively on rare classes (Wang et al., 2026). We now provide specific details about our training procedure.

B.1 Pretraining data

All OptimRNA models in this paper are pretrained on a corpus built from **RNAcentral release 21.0**, which contains 39,048,508 unique RNA sequences. We remove exact duplicate sequences which leaves 39,035,267 sequences. Each sequence carries the RNA type assigned by RNAcentral. We keep nineteen RNA types as families and collapse all remaining types, into a single *other* family. We follow the scheme of AIDO.RNA (Zou et al., 2024) for validation and test sets which are drawn per family with a fixed quota rather than proportionally. 10,000 sequences are drawn per family for rRNA, lncRNA, tRNA, misc_RNA and *other*, and 5,000 for the remaining fifteen, giving 125,000 sequences each, so that the held-out sets are quasi-uniform over families and validation loss is not dominated by ribosomal RNA. The remaining 38,785,267 sequences form the training corpus; Table 5 reports its composition.

Because the corpus is extremely unbalanced, we do not sample sequences uniformly. Instead, batches are drawn with family-damped probabilities $p \propto \mathrm{count}^{0.7}$, which lowers the rRNA share of a batch to 50.5% and raises under-represented families by a factor of two to seven (Table 5, last column). Sequences longer than 1022 nucleotides are cropped to a uniformly drawn window of

Table 5: Composition of the RNAcentral 21.0 pretraining corpus (training split, 38,785,267 sequences). *Corpus* is the share of sequences in the pool; *sampled* is the share of a training batch under the family-damped weights $p \propto \mathrm{count}^{0.7}$ actually used to train every model in this paper.

Family	Sequences	Corpus (%)	Sampled (%)
rRNA	28,109,595	72.48	50.50
lncRNA	3,130,784	8.07	10.86
tRNA	2,758,264	7.11	9.94
misc_RNA	2,175,946	5.61	8.42
sRNA	427,295	1.10	2.70
pre_miRNA	385,474	0.99	2.51
other	319,929	0.83	2.20
snRNA	290,430	0.75	2.06
snoRNA	275,347	0.71	1.98
ncRNA	235,731	0.61	1.78
SRP_RNA	170,142	0.44	1.42
hammerhead_ribozyme	131,555	0.34	1.18
miRNA	97,052	0.25	0.96
antisense_RNA	57,203	0.15	0.66
RNase_P_RNA	51,218	0.13	0.61
tmRNA	50,633	0.13	0.61
precursor_RNA	45,606	0.12	0.56
siRNA	33,385	0.09	0.45
ribozyme	28,394	0.07	0.40
Y_RNA	11,284	0.03	0.21
Total	38,785,267	100.00	100.00

1024 tokens; the resulting average is 430 tokens per sequence, or approximately 1.7×10^{10} tokens over the single pass described in Appendix B.2.

B.2 Training hyperparameters

All OptimRNA models and the retrained RiNALMo-micro are pretrained on the RNAcentral corpus of Appendix B.1. Sequences are masked with the standard BERT scheme (15% of positions, of which 80% are replaced by the mask token, 10% by a random token and 10% left unchanged). We train for 202k steps at a batch size of 192 sequences, a single pass over the corpus, in bf16 mixed precision. Matrix parameters are optimized with Muon (momentum 0.95, weight decay 0.1) and the remaining parameters with AdamW (learning rate 10^{-4}, $\beta = (0.9, 0.95)$, weight decay 0.01). The Muon learning rate is tuned per scale: 1.5×10^{-3} for the 34M model, 3.5×10^{-4} for the 88M model, and 7×10^{-4} for the retrained RiNALMo-micro. The learning rate follows a cosine schedule with 1000 warm-up steps, decaying to 10% of its peak at the final step. All ablation arms share this recipe; only the architecture differs.

Table 6 collects the hyperparameters of the three pretrained models.

C OptimRNA Transformer

We describe elements of the OptimRNA Transformer layer in more detail.

C.1 Fused pair state update

We provide more details for the equation 5

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell)_{ij} = \frac{1}{\sqrt{r}} \sum_{q=-R}^{R} w_q \, (\boldsymbol{\phi}^\ell_{i+q})^\top \, \mathrm{diag}(\boldsymbol{s}) \, \boldsymbol{\phi}^\ell_{j-q} = \frac{1}{\sqrt{r}} \, (\boldsymbol{a}^\ell_i)^\top \boldsymbol{b}^\ell_j, \tag{10}$$

Table 6: Hyperparameters for pretraining. All three models are trained on the same corpus (Appendix B.1) for one epoch on a single GPU.

	OptimRNA-34M	OptimRNA-88M	RiNALMo-micro
dataset		RNAcentral 21.0	
num-layers	12	12	12
hidden-size	416	768	480
num-attn-heads	16	12	20
ffn-hidden-size	1,664	2,048	1,920
max-seq-len		1,024	
batch size		192 sequences	
steps (epochs)		202k (1)	
optimizer		Muon (matrices) + AdamW (rest)	
Muon learning rate	1.5×10^{-3}	3.5×10^{-4}	7×10^{-4}
AdamW learning rate		1×10^{-4}, $\beta = (0.9, 0.95)$	
lr scheduler		cosine, decaying to 10% of peak	
warmup steps		1,000	
weight decay		0.1 (Muon) / 0.01 (AdamW)	
hidden dropout	0	0	0.1
attention dropout	0	0	0.1
clipnorm		1.0	
precision		bf16	
attention kernel	FlexAttention	FlexAttention	FlashAttention-2

where

$$\boldsymbol{a}_i^\ell = \begin{bmatrix} w_{-R}\,\mathrm{diag}(\boldsymbol{s})\,\boldsymbol{\phi}_{i-R}^\ell \\ \vdots \\ w_R\,\mathrm{diag}(\boldsymbol{s})\,\boldsymbol{\phi}_{i+R}^\ell \end{bmatrix}, \qquad \boldsymbol{b}_j^\ell = \begin{bmatrix} \boldsymbol{\phi}_{j+R}^\ell \\ \vdots \\ \boldsymbol{\phi}_{j-R}^\ell \end{bmatrix} \in \mathbb{R}^{(2R+1)r}.$$

Stacking the vectors a_i^ℓ and b_j^ℓ as rows gives two matrices $\boldsymbol{\Phi}_A^\ell, \boldsymbol{\Phi}_B^\ell \in \mathbb{R}^{N\times(2R+1)r}$. With $\mathcal{S}_q$ the operator that shifts the rows of a matrix by q, $(\mathcal{S}_q\boldsymbol{\Phi})_i = \boldsymbol{\phi}_{i+q}$, with zero rows out of range,

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell) = \frac{1}{\sqrt{r}}\,\boldsymbol{\Phi}_A^\ell\,(\boldsymbol{\Phi}_B^\ell)^\top, \qquad \boldsymbol{P}^\ell = \lambda_\ell \boldsymbol{P}^{\ell-1} + \frac{1}{\sqrt{r}}\,\boldsymbol{\Phi}_A^\ell\,(\boldsymbol{\Phi}_B^\ell)^\top. \tag{11}$$

C.2 Linear memory implementation

Flash Attention (Dao et al., 2022) is a tiled algorithm for computing self-attention in linear memory complexity. Since the bias term $[\boldsymbol{B}^\ell]_{i,j}$ is only dependent on $[\boldsymbol{P}^{\ell-1}]_{i,j}$ and its antidiagonal convolution partners, we can recompute it from the $[\boldsymbol{\Phi}_A^l]_i$ and $[\boldsymbol{\Phi}_B^l]_j$ terms for $l < \ell$ on-demand. By unrolling the recurrence (Eq. 66):

$$\boldsymbol{P}^{\ell-1} = \alpha c_0^{\ell-1}\Pi + \frac{1}{\sqrt{r}}\sum_{t=1}^{\ell-1} c_t^{\ell-1}\,\boldsymbol{\Phi}_A^t(\boldsymbol{\Phi}_B^t)^\top = \alpha c_0^{\ell-1}\Pi + \frac{1}{\sqrt{r}}\,\boldsymbol{\Psi}_A^\ell(\boldsymbol{\Psi}_B^\ell)^\top, \tag{12}$$

where $\boldsymbol{\Psi}_A^\ell = [c_1^{\ell-1}\boldsymbol{\Phi}_A^1, \ldots, c_{\ell-1}^{\ell-1}\boldsymbol{\Phi}_A^{\ell-1}]$ and $\boldsymbol{\Psi}_B^\ell = [\boldsymbol{\Phi}_B^1, \ldots, \boldsymbol{\Phi}_B^{\ell-1}]$, and $c_t^{\ell-1}$ the retention products of Eq. 65. The pair state read by layer ℓ is then a single product of two $N\times(\ell-1)(2R+1)r$ matrices. As in the default implementation, these are never stored: their rows are shifted, rescaled copies of the rows of $\boldsymbol{\Phi}^1, \ldots, \boldsymbol{\Phi}^{\ell-1}$, so storing the projections themselves ($N(\ell-1)r$ numbers, linear in N) suffices.

We use this formulation to perform a tile-wise reconstruction of P in a FlashAttention-style kernel. For a tile of query positions I and key positions J, the kernel forms $\boldsymbol{P}_{IJ}^{\ell-1} = \alpha c_0^{\ell-1}\Pi_{IJ} + \frac{1}{\sqrt{r}}(\boldsymbol{\Psi}_A^\ell)_I(\boldsymbol{\Psi}_B^\ell)_J^\top$ in on-chip memory, which only requires the rows of each $\boldsymbol{\Phi}^t$ within distance R of I and J. It then applies the nonlinear read, adds the result to the attention scores, and discards the

tile after the softmax update. The backward pass recomputes the tiles in the same way and accumulates the gradients of the projections, the coefficients and β_ℓ tile by tile, so no $N \times N$ tensor is stored in either pass. Time complexity remains quadratic in N, like attention itself, but memory becomes linear. We show this in Figure 5 where we measure memory consumption for a single sequence of increasing length for the default implementation and the linear one.

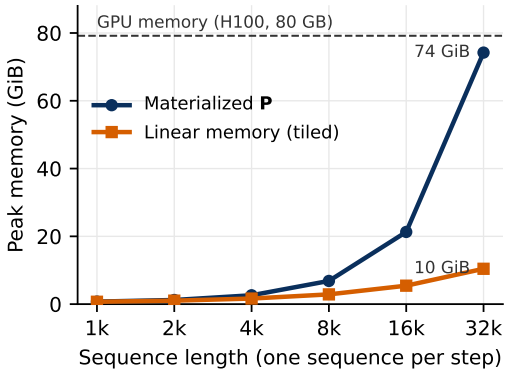


Figure 5: Peak memory of one training step of OptimRNA-34M on a single full-length sequence, with the default or linear implementation

D Experimental setup

D.1 bpRNA TS0

Zero-Shot structure prediction To probe the internally captured dependencies of a model, we follow the in-silico mutagenesis approach of (Tomaz da Silva et al., 2025). In an RNA sequence of length L, to quantify the dependency between two positions i and j, the nucleotide at position i is substituted by q and we record the change in the log-odds that the model assigns to nucleotide k at position j.

$$D_{ij}(q,k) = \log_2 \left(\frac{\widehat{\text{odds}}(n_j = k \mid n_1, \ldots, n_i = q, \ldots, n_L)}{\widehat{\text{odds}}(n_j = k \mid n_1, \ldots, n_L)} \right), \tag{13}$$

where $q, k \in \{A, C, G, U\}$ are the query and key nucleotides and $\widehat{\text{odds}}$ are the odds inferred by the model from the unmasked sequence. Tomaz da Silva et al. (2025) use $e_{ij} = \max_{q,k} |D_{ij}(q,k)|$ as the value of the dependency map; we have found this to be suboptimal and since base pairing is our target, we instead weight complementary substitutions. This does not change the ranking of the models but improves their performance across the board.

$$e_{ij} = \sum_{q,k} W_{qk}\, D_{ij}(q,k), \qquad W_{qk} = \begin{cases} 1 & (q,k) \in \{AU, UA, CG, GC\}, \\ -\frac{1}{12} & \text{otherwise}, \end{cases} \tag{14}$$

and use the symmetrised map $e + e^\top$, greedily decoded to at most one partner per nucleotide, as the structure prediction. Our aggregation function does not help our models more than others, as can be seen in the comparison table 7.

To give every model a fair chance and prevent threshold artifacts from skewing the results, we report the F1 score on the TS0 test set by sweeping over thresholds and select the best on a subset of 200 samples of VL0. We do not use the full VL0 dataset since nucleotide dependency calculation is expensive. We also report the PR-AUC, which is threshold-free. RNAfold and RNAstructure (partition function) and CONTRAfold (posteriors) give pair probabilities, which we cut at 0.5. Unlike the language models, whose threshold is selected on VL0, the folders' F1 is therefore not tuned and may be underestimated; their PR-AUC is unaffected. LinearFold and the MFE/MEA structures are discrete, so they have no tunable threshold and no PR-AUC. In Table 8, we report the metrics for an extended set of models, the F1 score is the exact F1 as opposed to the shift-tolerant F1 used

Table 7: **Zero-shot secondary-structure prediction** on bpRNA TS0 ($n = 1305$) using different aggregation methods. PR-AUC is reported.

Model	Ours	Original (max |log-odds|)
OptimRNA 34M, with pair state	**0.566**	**0.493**
OptimRNA 34M, without pair state	0.505	0.416
ERNIE-RNA	0.557	0.466

Table 8: **Zero-shot structure prediction on bpRNA TS0** (Appendix D.1). Best in bold, second underlined.

Model	Params	Architecture	P	R	F_1	PR-AUC
CONTRAfold (posteriors)	–	pair probabilities	**0.5922**	0.4593	0.4887	0.5415
RNAstructure (MFE)	–	MFE structure	0.4501	0.5695	0.4898	–
LinearFold (beam 100)	–	MFE structure	0.4498	0.5782	0.4928	–
RNAstructure (PF)	–	pair probabilities	0.4836	0.5637	0.5050	0.5132
RNAfold (PF)	–	pair probabilities	0.4884	0.5689	0.5091	0.5215
CONTRAfold (MEA)	–	MEA structure	0.4839	**0.5975**	<u>0.5216</u>	–
RiNALMo-micro	33.5M	–	0.3011	0.2680	0.2535	0.2867
RNAErnie	86M	–	0.3387	0.2916	0.2730	0.2898
RNA-FM	100M	–	0.3908	0.3508	0.3362	0.3816
RiNALMo-mega	148M	–	0.3853	0.3861	0.3537	0.3975
structRFM	86M	–	0.4490	0.4069	0.3885	0.4285
RiNALMo-giga	650M	–	0.4808	0.4678	0.4407	0.4865
RiNALMo-micro (retrained)	33.5M	–	0.5042	0.4848	0.4566	0.4937
AIDO.RNA	648M	–	0.5197	0.5507	0.5068	0.5273
AIDO.RNA	1.6B	–	0.5372	0.5426	0.5095	0.5281
ERNIE-RNA	85.7M	–	0.5344	0.5591	0.5169	0.5566
OptimRNA (ours)	34M	no pair map (control)	0.4949	0.4858	0.4511	0.5049
		+ pair map (no antidiag.)	0.5053	0.5256	0.4755	0.5315
		+ pair map	0.5301	0.5803	0.5173	<u>0.5659</u>
OptimRNA (ours)	88M	no pair map (control)	0.5098	0.5405	0.4867	0.5375
		+ pair map	<u>0.5418</u>	<u>0.5880</u>	**0.5254**	**0.5714**

in fine-tuned structure prediction inherited from RiNALMo and AIDO.RNA. In Figure 2 we report the PR-AUC of relevant models on bpRNA TS0 compared with their parameter count, which we measure from the built models. RNAfold is reported as a dashed gray line.

Fine-tuned structure prediction We follow RiNALMo's procedure to fine-tune our model, with one difference in the unfreezing schedule described below. We use the TR0 training set to train on, select the threshold on VL0 (n=1300) and report metrics on TS0 (n=1305). We report precision, recall and the shift-tolerant F1 score, which allows for a shift of one nucleotide in the predicted structure. This is the same F1 score reported in RiNALMo and AIDO.RNA, according to their respective released evaluation code. We use the same structure prediction head as RiNALMo and AIDO.RNA, and we progressively unfreeze the backbone during training, starting with only the structure prediction head trainable. Newly unfrozen layers start at one tenth of the base learning rate, whereas RiNALMo starts them at one tenth of the current, already decayed learning rate. This latter schedule underfit our models in preliminary runs on this dataset specifically. The structure prediction head consists of two (1x1 $\rightarrow$ 3x3 $\rightarrow$ 1x1) convolution blocks, and the loss is a binary cross-entropy on the upper triangle of the prediction matrix. It takes as input the outer concatenation of the last layer's embeddings, with possibly the pair representation added for the models that are equipped. If we pass the pair state to the structure prediction head alongside the token embeddings, we signify it with $P \rightarrow head$. We train for 60 epochs, picking the best model on the validation set, this is the procedure that AIDO.RNA uses and is more appropriate for an in-family structure prediction task, as opposed to RiNALMo's 15 epochs. We use a learning rate of 5e-4 for OptimRNA-34M and 2.5e-4 for OptimRNA-88M with Adam as an optimizer and no dropout, except for the retrained RiNALMo-micro which has a dropout of 0.1, to match its pretraining setup. We use a batch size of 1, and a

Table 9: **ArchiveII leave-one-family-out.** Shift-tolerant F1 (%) on each held-out family after fine-tuning on the other eight (Appendix D.2). $P \rightarrow$ head: the pair map is passed to the prediction head alongside the token embeddings. Best in bold, second underlined. All models that have not been trained by us are reported from RiNALMo and AIDO.RNA's papers.

Model	Params	Architecture	5S	SRP	tRNA	tmRNA	RNaseP	grpI	16S	23S	telo.	Avg.
RNAstructure (physics)	–	–	63.0	63.0	70.0	43.0	55.0	54.0	57.0	73.0	50.0	58.7
CONTRAfold	–	–	67.0	60.0	76.0	44.0	60.0	59.0	60.0	75.0	<u>54.0</u>	61.7
MXfold2	–	–	54.0	50.0	64.0	46.0	51.0	45.0	55.0	64.0	34.0	51.4
UFold	–	–	53.0	26.0	26.0	40.0	41.0	45.0	41.0	45.0	**80.0**	44.1
RNA-FM	100M	–	52.0	25.0	78.0	29.0	30.0	16.0	13.0	17.0	8.0	29.8
RiNALMo	650M	–	88.0	70.0	93.0	80.0	80.0	**66.0**	74.0	85.0	12.0	72.0
AIDO.RNA	1.6B	–	85.3	73.9	**94.5**	83.8	80.4	<u>64.4</u>	79.5	89.6	8.5	73.3
RiNALMo-micro (our recipe)	33.5M	–	86.5	68.1	92.6	83.1	79.3	47.9	73.6	82.6	11.8	69.5
OptimRNA (ours)	34M	no pair map (control)	82.4	66.6	87.3	80.4	78.3	38.7	70.7	85.0	12.9	66.9
		+ pair map	88.1	75.6	88.3	83.7	82.9	51.8	77.5	87.0	19.8	72.8
		+ pair map, $P \rightarrow$ head	<u>92.4</u>	**81.1**	90.6	**85.7**	**87.4**	54.7	**83.7**	**93.0**	27.3	**77.3**
OptimRNA (ours)	88M	no pair map (control)	83.5	67.2	86.9	79.8	79.3	39.9	66.0	83.8	10.9	66.4
		+ pair map	86.7	72.4	88.5	83.3	79.4	43.9	71.6	87.8	22.6	70.7
		+ pair map, $P \rightarrow$ head	**93.9**	<u>77.9</u>	<u>93.5</u>	<u>85.2</u>	<u>86.8</u>	47.3	<u>81.9</u>	<u>91.4</u>	28.6	<u>76.3</u>

Linear learning rate schedule which finishes at 0.1x the peak LR. We sweep thresholds (0.01–0.29) and select on VL0. Predictions are greedy-decoded one-partner-per-base with non-canonical pairs and $|i - j| < 4$ prohibited. Published rows are transcribed from their papers and symbolized as † in the tables, not re-measured here. For ERNIE-RNA, whose paper reports an exact-match F1 (0.748) on the same TS0 split, we instead run its released bpRNA-1m fine-tuned checkpoint (trained on TR0 only) on VL0 and TS0 and score the output of its own prediction head (2D convolutions over the last layer's attention map, not the RiNALMo head that the RiNALMo, AIDO.RNA and OptimRNA rows share) under the protocol above, i.e. greedy canonical decoding at a threshold selected on VL0, which gives a shift-tolerant F1 of 79.1. Our models are fine-tuned on a single seed.

D.2 Cross-family performance on ArchiveII

ArchiveII cross-family structure prediction We follow the same procedure as RiNALMo to evaluate our model on the cross-family structure prediction task. The archiveII dataset is used, one family is held out and our model is fine-tuned on the other families with a learning rate of 5e-4 for OptimRNA-34M and 2.5e-4 for OptimRNA-88M and the Adam optimizer. This time, we train for 15 epochs and select the last epoch. The model is fine-tuned with progressive unfreezing and a ResNet prediction head exactly as RiNALMo does. Following the RiNALMo protocol we report the *shift-tolerant* F1, average the per-sequence scores unweighted within a family and the nine family scores unweighted across families, so that each family has equal importance in the average F1 score, regardless of its size. Predictions are greedy-decoded one-partner-per-base with non-canonical pairs and $|i - j| < 4$ prohibited. We select the decision threshold on each fold's validation split over the grid 0.01–0.29, retuned every epoch, reporting the final epoch. Our rows are means over three seeds. Published rows are transcribed from the RiNALMo paper, except for AIDO.RNA's numbers which are taken directly from their paper.

D.3 Ablation on the pair mechanism

We report the bpRNA TS0 zero-shot PR-AUC following the protocol stated in Appendix D.1 for our baseline model, which does not have our pair mechanism, our "+Pair" model which has said mechanism, but does not have the antidiagonal convolution, and the "+Pair+Antidiag." which has both. We see that the gains in zero-shot structure prediction are complementary for both of those mechanisms. We use the 34M sized model for this comparison. Everything else beyond the pair representation and the antidiagonal convolution is shared: training dataset, optimizer, learning rate, GPU type, training step count.

Table 10: **Splice-site prediction (Spliceator), per species.** AUPRC (%), mean of the donor and acceptor tasks, on the four benchmark species; mean ± sd over 3 fine-tuning seeds. The 'Mean' column is the splice-site column of Table 2. ‡released fine-tuned weights re-scored under our protocol (a single measurement).

Model	Pair state	Fish	Fly	Plant	Worm	Mean
RiNALMo-giga‡ (650M)	–	**99.39**	**98.82**	**98.87**	**98.58**	**98.92**
RiNALMo-micro (33.5M)	–	98.84 ±0.28	97.52 ±0.58	96.79 ±0.66	95.82 ±1.15	97.24 ±0.66
RiNALMo-micro (33.5M, our pretraining)	–	99.24 ±0.04	98.61 ±0.05	98.00 ±0.36	97.61 ±0.38	98.36 ±0.18
ERNIE-RNA (85.7M)	–	99.06 ±0.05	98.30 ±0.03	97.65 ±0.12	97.51 ±0.06	98.13 ±0.03
OptimRNA-34M (ours)	✗	98.96 ±0.06	98.31 ±0.10	97.93 ±0.12	97.46 ±0.30	98.16 ±0.13
	✓	98.99 ±0.10	98.22 ±0.29	97.86 ±0.22	97.12 ±0.37	98.05 ±0.24
OptimRNA-88M (ours)	✗	98.97 ±0.12	98.04 ±0.34	97.65 ±0.29	96.63 ±0.70	97.82 ±0.33
	✓	99.07 ±0.01	98.44 ±0.13	97.88 ±0.24	97.38 ±0.36	98.19 ±0.17

D.4 Splice site prediction and modification prediction

This splice site prediction task was first defined in Spliceator (Scalzitti et al., 2021) which defines positive and negative classes. The task is thus a binary classification task for donor site prediction and acceptor site prediction separately. We report the average of the area under the precision-recall curve (AUPRC) on the donor prediction task and the acceptor prediction tasks on the four test species (fish/fly/plant/worm); Table 10 breaks the same runs out per species. We report AUPRC rather than F1 at the fixed 0.5 threshold because on these shifted species the latter measures calibration as much as ranking; the F1 numbers, together with the values transcribed from RiNALMo's paper, are given in Table 11. We use RiNALMo's fine-tuning recipe (learning rate of 10^{-5}, 5 epochs, best validation epoch reported). We fine-tune our own 34M and 88M sized models, ERNIE-RNA, and RiNALMo-micro from its published pretrained weights as well as our retrained version, under exactly this recipe. We re-score the released fine-tuned RiNALMo-giga (650M) weights through the same evaluation path, so that every splice-site entry of Table 2 is measured rather than transcribed. We report the mean and standard deviation over three fine-tuning seeds; the released RiNALMo-giga weights involve no fine-tuning seed and are therefore a single measurement.

For modification prediction, the evaluation sets are those of MultiRM: for each of the five modifications, sequences centred on a known modified site and on an unmodified site of the same nucleotide, split into training, validation and test sets. We extract the token embedding of the candidate nucleotide at every layer of a frozen backbone and fit one elastic logistic probe per (model, layer, modification) on standardised features, selecting the regularisation strength and type and the layer on the validation split. Table 12 reports the test PR-AUC per modification behind the modification column of Table 2. Each probe is fit with three seeds and we report their mean; the probes are near-deterministic (standard deviation below 0.1 points at the selected layers), so no seed interval is reported.

D.5 Mechanism analysis

In Figure 4, we perform various analysis of structural components. We describe them in more details.

Rank analysis Using bpRNA TS0 and the two models equipped with a pair representation, we measure the effective rank (Roy & Vetterli, 2007) of $\boldsymbol{P}$ and the attention bias $\boldsymbol{B}$ at every layer after row-centering them.

Direct structure prediction with P We report the PR-AUC on bpRNA TS0 obtained by using the entries of $\boldsymbol{P}^{\ell}$ directly as pair scores, without any correction, after every write. We compare the models with and without the antidiagonal convolution.

MLM and Structure probe To assess the ability of token embeddings to recover masked nucleotide identity across layers, we train one linear probe per layer, using TR0 sequences as a training

Table 11: **Splice-site prediction, F1 at the fixed 0.5 threshold (%)**, mean of donor and acceptor. Published rows transcribed from RiNALMo (Fig. 5c/5d); the released RiNALMo weights re-scored on our pipeline are listed separately (its acceptor model is miscalibrated at 0.5, AUPRC intact). Ours: mean ± sd over 3 seeds. †transcribed from the original publication. ‡released fine-tuned weights re-scored under our protocol (a single measurement).

Model	Pair state	Fish	Fly	Plant	Worm	Mean
RiNALMo-giga† (650M)	–	97.70	96.10	96.25	95.63	96.42
SpliceBERT† (19M)	–	96.70	95.83	94.95	94.82	95.58
RNA-FM† (100M)	–	94.94	93.20	89.93	90.79	92.22
Spliceator† (CNN)	–	93.50	92.90	92.85	91.60	92.71
RiNALMo-giga‡ (650M)	–	96.38	94.99	93.73	93.94	94.76
RiNALMo-micro (33.5M)	–	94.31 ±1.63	91.73 ±1.88	91.03 ±1.28	89.55 ±2.09	91.66 ±1.71
RiNALMo-micro (33.5M, our pretraining)	–	94.70 ±0.21	92.73 ±0.75	92.21 ±0.98	90.89 ±1.05	92.63 ±0.73
ERNIE-RNA (85.7M)	–	95.75 ±0.43	93.75 ±0.48	92.51 ±0.66	92.14 ±0.67	93.54 ±0.45
OptimRNA-34M (ours)	✗	92.05 ±1.45	90.59 ±1.16	90.92 ±0.60	88.35 ±0.77	90.48 ±0.99
	✓	93.47 ±0.74	91.51 ±0.79	91.94 ±0.92	89.03 ±1.31	91.49 ±0.90
OptimRNA-88M (ours)	✗	93.18 ±0.58	90.92 ±0.77	91.17 ±0.28	88.61 ±1.09	90.97 ±0.66
	✓	93.75 ±1.23	91.85 ±1.63	91.91 ±1.06	89.84 ±2.04	91.84 ±1.48

Table 12: **Modification-site prediction (MultiRM).** Test PR-AUC (%) of a linear probe on the frozen token embedding of the candidate nucleotide, best layer selected on the validation split. ✓ / ✗: with and without the pair state – the two rows at each scale are otherwise identical (same width, depth, ConvNeXt embeddings, optimiser, corpus and step count). Mean over three probe seeds (sd below 0.1 points at the selected layer), so no seed interval is shown. Best in bold.

Model	Pair state	m1A	m5C	m6A	m7G	Ψ	Average
RiNALMo-giga (650M)	–	95.0	96.8	89.0	76.9	**93.5**	90.2
RiNALMo-micro (33.5M)	–	95.9	**96.9**	92.8	**79.4**	92.0	**91.4**
RiNALMo-micro (33.5M, our pretraining)	–	95.7	96.7	**94.4**	79.2	89.9	91.2
ERNIE-RNA (85.7M)	–	95.1	92.8	90.8	76.3	88.9	88.8
OptimRNA-34M (ours)	✗	88.8	95.2	90.9	78.9	86.2	88.0
	✓	92.5	94.7	93.7	78.2	89.0	89.6
OptimRNA-88M (ours)	✗	**96.1**	95.2	93.2	71.6	87.8	88.8
	✓	93.2	94.1	91.6	78.6	89.6	89.4

set and evaluating on TS0. We report nucleotide prediction accuracy across layers and split between positions that are paired and unpaired, according to the bpRNA ground truth.

The structure probe is a small head, meant to measure how much structural information its input carries rather than to achieve the best performance. It uses four bilinear maps to combine token embeddings into a pairwise object, which then goes into three 3×3 convolutions (16 channels, GELU) and a 1×1 output convolution. One probe is trained per layer, with AdamW for 12 epochs on TR0 sequences of up to 500 nucleotides. We keep the epoch with the best VL0 performance and evaluate on TS0. When measuring the effect of adding the pair state, we concatenate the bounded bias $\tau \tanh(\boldsymbol{P}/\tau)$ of the pair state written by that layer (which is consumed by the next layer's attention) as an extra input channel to the convolution

D.6 Throughput and memory measurements

We report time and GB per sequence for our 34M-sized models, as well as ERNIE-RNA and our 88M-sized models as it is a published model of similar size and our most direct comparison. We report the median value over 200 steps after 30 warm-up steps, with a batch size of 48. We use bf16 precision, padding at the longest sequence, measured on a single H100 GPU. We report time per step and peak memory consumption.

Table 13: **Training-step cost.** Measures on real RNACentral batches (48 sequences of up to 1024 nt per step, no gradient accumulation, bf16) on a single H100 for 200 optimizer steps after 30 warm-up steps: median wall time per step, peak memory, and peak memory per sequence.

Model	Params	Pair state	s / step	peak GiB	GiB / sequence
OptimRNA-34M	34M	✗	0.074	6.9	0.14
		✓	0.162	11.1	0.23
ERNIE-RNA	85.7M	–	0.233	68.6	1.43
OptimRNA-88M	88M	✗	0.083	10.2	0.21
		✓	0.189	15.8	0.33

D.7 Signature of the signed low-rank pairwise writes

We first recall a standard result concerning rectangular congruence transformations. For a real symmetric matrix $\boldsymbol{A}$, let $n_+(\boldsymbol{A})$, $n_-(\boldsymbol{A})$, and $n_0(\boldsymbol{A})$ denote its numbers of positive, negative, and zero eigenvalues, counted with multiplicity.

Lemma 1 (Inertia under a rectangular congruence)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be symmetric and let $\boldsymbol{B} \in \mathbb{R}^{m\times n}$. Then*

$$n_+(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) \le n_+(\boldsymbol{A}), \tag{15}$$

$$n_-(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) \le n_-(\boldsymbol{A}). \tag{16}$$

If $\boldsymbol{B}$ has full column rank n, then

$$n_+(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) = n_+(\boldsymbol{A}), \tag{17}$$

$$n_-(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) = n_-(\boldsymbol{A}), \tag{18}$$

$$n_0(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) = n_0(\boldsymbol{A}) + m - n. \tag{19}$$

Proof. Let $\mathcal{U}_+ \subseteq \mathbb{R}^m$ be a subspace on which the quadratic form induced by $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top$ is positive definite. For every nonzero $\boldsymbol{x} \in \mathcal{U}_+$,

$$\boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top \boldsymbol{x} = (\boldsymbol{B}^\top \boldsymbol{x})^\top \boldsymbol{A}(\boldsymbol{B}^\top \boldsymbol{x}) > 0. \tag{20}$$

In particular, $\boldsymbol{B}^\top \boldsymbol{x} \neq 0$ for every nonzero $\boldsymbol{x} \in \mathcal{U}_+$, so the map $\boldsymbol{x} \mapsto \boldsymbol{B}^\top \boldsymbol{x}$ is injective on $\mathcal{U}_+$. Its image is a subspace of $\mathbb{R}^n$ on which the quadratic form induced by $\boldsymbol{A}$ is positive definite. Since the largest dimension of such a subspace is $n_+(\boldsymbol{A})$,

$$\dim(\mathcal{U}_+) \le n_+(\boldsymbol{A}). \tag{21}$$

Taking $\mathcal{U}_+$ to have maximal dimension gives

$$n_+(\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top) \le n_+(\boldsymbol{A}). \tag{22}$$

Applying the same argument to $-\boldsymbol{A}$ proves the corresponding inequality for the negative eigenvalues.

If $\boldsymbol{B}$ has full column rank, its thin QR decomposition can be written as

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R}, \tag{23}$$

where $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns and $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is invertible. It follows that

$$\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^\top = \boldsymbol{Q}(\boldsymbol{R}\boldsymbol{A}\boldsymbol{R}^\top)\boldsymbol{Q}^\top. \tag{24}$$

By Sylvester's law of inertia, the invertible congruence $\boldsymbol{R}\boldsymbol{A}\boldsymbol{R}^\top$ has the same numbers of positive, negative, and zero eigenvalues as $\boldsymbol{A}$. Multiplication by the semi-orthogonal matrix $\boldsymbol{Q}$ preserves these eigenvalues on $\mathrm{col}(\boldsymbol{Q})$ and introduces $m - n$ additional zero eigenvalues on its orthogonal complement. This proves equation 19. □

Lemma 2 (Signature capacity of an OptimRNA pairwise write)**.** *Let*

$$\boldsymbol{D}_{\boldsymbol{s}} = \mathrm{diag}(\boldsymbol{s}), \tag{25}$$

$$\boldsymbol{G} = \frac{1}{\sqrt{r}} \boldsymbol{W}_{\mathrm{pair}}^{\top} \boldsymbol{D}_{\boldsymbol{s}} \boldsymbol{W}_{\mathrm{pair}}, \tag{26}$$

$$\boldsymbol{\Delta}^{\ell} = \boldsymbol{U}^{\ell} \boldsymbol{G} (\boldsymbol{U}^{\ell})^{\top} = \frac{1}{\sqrt{r}} \boldsymbol{\Phi}^{\ell} \boldsymbol{D}_{\boldsymbol{s}} (\boldsymbol{\Phi}^{\ell})^{\top}, \tag{27}$$

where

$$\boldsymbol{\Phi}^{\ell} = \boldsymbol{U}^{\ell} \boldsymbol{W}_{\mathrm{pair}}^{\top}. \tag{28}$$

Define

$$p_{\boldsymbol{s}} = \#\{a : s_a > 0\}, \tag{29}$$

$$n_{\boldsymbol{s}} = \#\{a : s_a < 0\}, \tag{30}$$

$$z_{\boldsymbol{s}} = \#\{a : s_a = 0\}. \tag{31}$$

Then

$$n_+(\boldsymbol{\Delta}^{\ell}) \le n_+(\boldsymbol{G}) \le p_{\boldsymbol{s}}, \tag{32}$$

$$n_-(\boldsymbol{\Delta}^{\ell}) \le n_-(\boldsymbol{G}) \le n_{\boldsymbol{s}}. \tag{33}$$

Moreover:

1. *if $\boldsymbol{W}_{\mathrm{pair}}$ has full row rank r, then*

$$n_+(\boldsymbol{G}) = p_{\boldsymbol{s}}, \qquad n_-(\boldsymbol{G}) = n_{\boldsymbol{s}}, \qquad n_0(\boldsymbol{G}) = d - r + z_{\boldsymbol{s}}; \tag{34}$$

2. *if $\boldsymbol{\Phi}^{\ell}$ has full column rank r, then*

$$n_+(\boldsymbol{\Delta}^{\ell}) = p_{\boldsymbol{s}}, \qquad n_-(\boldsymbol{\Delta}^{\ell}) = n_{\boldsymbol{s}}, \qquad n_0(\boldsymbol{\Delta}^{\ell}) = L - r + z_{\boldsymbol{s}}. \tag{35}$$

Proof. The positive scalar $1/\sqrt{r}$ does not change the signs of the eigenvalues. Applying Lemma 1 with

$$\boldsymbol{A} = \boldsymbol{D}_{\boldsymbol{s}}, \qquad \boldsymbol{B} = \boldsymbol{W}_{\mathrm{pair}}^{\top} \tag{36}$$

gives

$$n_+(\boldsymbol{G}) \le n_+(\boldsymbol{D}_{\boldsymbol{s}}) = p_{\boldsymbol{s}}, \tag{37}$$

$$n_-(\boldsymbol{G}) \le n_-(\boldsymbol{D}_{\boldsymbol{s}}) = n_{\boldsymbol{s}}. \tag{38}$$

Since

$$\boldsymbol{\Delta}^{\ell} = \boldsymbol{U}^{\ell} \boldsymbol{G} (\boldsymbol{U}^{\ell})^{\top}, \tag{39}$$

a second application with

$$\boldsymbol{A} = \boldsymbol{G}, \qquad \boldsymbol{B} = \boldsymbol{U}^{\ell} \tag{40}$$

gives

$$n_+(\boldsymbol{\Delta}^{\ell}) \le n_+(\boldsymbol{G}), \tag{41}$$

$$n_-(\boldsymbol{\Delta}^{\ell}) \le n_-(\boldsymbol{G}). \tag{42}$$

Combining the inequalities proves Equation equation 33.

If $\boldsymbol{W}_{\mathrm{pair}}$ has full row rank, then $\boldsymbol{W}_{\mathrm{pair}}^{\top} \in \mathbb{R}^{d \times r}$ has full column rank. The equality case of Lemma 1 therefore shows that $\boldsymbol{G}$ preserves the positive and negative indices of $\boldsymbol{D}_{\boldsymbol{s}}$ and gains $d - r$ additional zero eigenvalues.

Likewise, if $\boldsymbol{\Phi}^{\ell} \in \mathbb{R}^{L \times r}$ has full column rank, applying the equality case directly to

$$\boldsymbol{\Delta}^{\ell} = \frac{1}{\sqrt{r}} \boldsymbol{\Phi}^{\ell} \boldsymbol{D}_{\boldsymbol{s}} (\boldsymbol{\Phi}^{\ell})^{\top} \tag{43}$$

shows that $\boldsymbol{\Delta}^{\ell}$ preserves the positive and negative indices of $\boldsymbol{D}_{\boldsymbol{s}}$ and gains $L - r$ additional zero eigenvalues. □

Remark 1. *Lemma 2 concerns the signs of the eigenvalues, not their numerical values. The entries of $\boldsymbol{s}$ are not generally the eigenvalues of either $\boldsymbol{G}$ or $\boldsymbol{\Delta}^\ell$, because the corresponding factor matrices are not constrained to be orthonormal. The lemma instead establishes that $\boldsymbol{s}$ determines the maximum numbers of positive and negative directions available to the shared geometry and its layerwise realizations. Rank deficiencies in $\boldsymbol{W}_{\mathrm{pair}}$ or $\boldsymbol{\Phi}^\ell$ may suppress some of these directions but cannot create new ones.*

D.8 Factorization and rank capacity of the structured pair state

This section establishes the factorized form and rank capacity of the pair state obtained by combining antidiagonal propagation with residual accumulation. Throughout, let

$$\boldsymbol{D}_{\boldsymbol{s}} = \mathrm{diag}(\boldsymbol{s}) \in \mathbb{R}^{r\times r}, \tag{44}$$

$$\rho = \mathrm{rank}(\boldsymbol{D}_{\boldsymbol{s}}) = \#\{a : s_a \neq 0\}. \tag{45}$$

At Transformer layer ℓ, the raw pairwise write is

$$\boldsymbol{\Delta}^\ell = \frac{1}{\sqrt{r}}\boldsymbol{\Phi}^\ell \boldsymbol{D}_{\boldsymbol{s}} (\boldsymbol{\Phi}^\ell)^\top, \qquad \boldsymbol{\Phi}^\ell \in \mathbb{R}^{L\times r}. \tag{46}$$

It follows directly that

$$\mathrm{rank}(\boldsymbol{\Delta}^\ell) \leq \rho \leq r. \tag{47}$$

D.8.1 Antidiagonal propagation

For an integer offset q, define the sequence-shift operator $\boldsymbol{T}_q \in \mathbb{R}^{L\times L}$ such that

$$(\boldsymbol{T}_q \boldsymbol{\Phi})_i = \begin{cases} \boldsymbol{\Phi}_{i+q}, & 1 \leq i+q \leq L, \\ \boldsymbol{0}, & \text{otherwise.} \end{cases} \tag{48}$$

The antidiagonal propagation operator of radius R is

$$\mathcal{A}_R(\boldsymbol{\Delta})_{ij} = \sum_{q=-R}^{R} w_q \boldsymbol{\Delta}_{i+q,j-q}, \tag{49}$$

where entries outside the valid sequence range are zero.

Proposition 1 (Factorization and rank of an antidiagonally propagated write)**.** *The propagated write generated from equation 46 satisfies*

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell) = \frac{1}{\sqrt{r}} \sum_{q=-R}^{R} w_q (\boldsymbol{T}_q \boldsymbol{\Phi}^\ell) \boldsymbol{D}_{\boldsymbol{s}} (\boldsymbol{T}_{-q} \boldsymbol{\Phi}^\ell)^\top. \tag{50}$$

Consequently,

$$\mathrm{rank}\left(\mathcal{A}_R(\boldsymbol{\Delta}^\ell)\right) \leq \min\{L, (2R+1)\rho\} \leq \min\{L, (2R+1)r\}. \tag{51}$$

If $w_q = w_{-q}$ for every q, then $\mathcal{A}_R(\boldsymbol{\Delta}^\ell)$ is symmetric.

Moreover, suppose that:

1. *$\boldsymbol{D}_{\boldsymbol{s}}$ is nonsingular, so $\rho = r$;*
2. *$w_q \neq 0$ for every $q \in \{-R, \ldots, R\}$;*
3. *$(2R+1)r \leq L$; and*
4. *the concatenated shifted-factor matrix*

$$\mathcal{F}_R^\ell = \begin{bmatrix} \boldsymbol{T}_{-R}\boldsymbol{\Phi}^\ell & \cdots & \boldsymbol{T}_R \boldsymbol{\Phi}^\ell \end{bmatrix} \in \mathbb{R}^{L\times(2R+1)r} \tag{52}$$

has full column rank.

Then the bound is attained:

$$\operatorname{rank}\left(\mathcal{A}_R(\boldsymbol{\Delta}^\ell)\right) = (2R+1)r. \tag{53}$$

Proof. Substituting equation 46 into equation 49 gives

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell)_{ij} = \frac{1}{\sqrt{r}} \sum_{q=-R}^{R} w_q (\phi_{i+q}^\ell)^\top \boldsymbol{D_s} \phi_{j-q}^\ell, \tag{54}$$

which is equivalent to equation 50.

Each term in the sum satisfies

$$\operatorname{rank}\left((\boldsymbol{T}_q \boldsymbol{\Phi}^\ell) \boldsymbol{D_s} (\boldsymbol{T}_{-q} \boldsymbol{\Phi}^\ell)^\top\right) \le \operatorname{rank}(\boldsymbol{D_s}) = \rho. \tag{55}$$

Subadditivity of matrix rank therefore gives

$$\operatorname{rank}\left(\mathcal{A}_R(\boldsymbol{\Delta}^\ell)\right) \le \sum_{q=-R}^{R} \rho = (2R+1)\rho. \tag{56}$$

Since the propagated write belongs to $\mathbb{R}^{L\times L}$, its rank is also bounded by L, proving equation 51.

To establish symmetry, transpose equation 50:

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell)^\top = \frac{1}{\sqrt{r}} \sum_{q=-R}^{R} w_q (\boldsymbol{T}_{-q} \boldsymbol{\Phi}^\ell) \boldsymbol{D_s} (\boldsymbol{T}_q \boldsymbol{\Phi}^\ell)^\top. \tag{57}$$

Replacing q by $-q$ gives

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell)^\top = \frac{1}{\sqrt{r}} \sum_{q=-R}^{R} w_{-q} (\boldsymbol{T}_q \boldsymbol{\Phi}^\ell) \boldsymbol{D_s} (\boldsymbol{T}_{-q} \boldsymbol{\Phi}^\ell)^\top. \tag{58}$$

Thus, $w_q = w_{-q}$ implies

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell)^\top = \mathcal{A}_R(\boldsymbol{\Delta}^\ell). \tag{59}$$

It remains to establish tightness. Index the block columns of $\mathcal{F}_R^\ell$ by offsets $q \in \{-R, \dots, R\}$ and define $\boldsymbol{C}_R \in \mathbb{R}^{(2R+1)r\times(2R+1)r}$ by

$$[\boldsymbol{C}_R]_{q,p} = \begin{cases} w_q \boldsymbol{D_s}, & p = -q, \\ \mathbf{0}_{r\times r}, & \text{otherwise.} \end{cases} \tag{60}$$

Then

$$\mathcal{A}_R(\boldsymbol{\Delta}^\ell) = \frac{1}{\sqrt{r}} \mathcal{F}_R^\ell \boldsymbol{C}_R (\mathcal{F}_R^\ell)^\top. \tag{61}$$

The matrix $\boldsymbol{C}_R$ is a block permutation matrix whose nonzero blocks are $w_q \boldsymbol{D_s}$. If every w_q is nonzero and $\boldsymbol{D_s}$ is nonsingular, then $\boldsymbol{C}_R$ is nonsingular. Since $\mathcal{F}_R^\ell$ has full column rank, rectangular congruence by $\mathcal{F}_R^\ell$ preserves the rank of $\boldsymbol{C}_R$. Therefore,

$$\operatorname{rank}\left(\mathcal{F}_R^\ell \boldsymbol{C}_R (\mathcal{F}_R^\ell)^\top\right) = (2R+1)r, \tag{62}$$

which proves equation 53. □

Remark 2. *The tightness conditions distinguish representational capacity from the rank realized by a trained model. Antidiagonal propagation does not necessarily increase rank: shifted factors may be linearly dependent, some coefficients may vanish, and signed terms may cancel. Proposition 1 instead shows that the increase from r to $(2R+1)r$ is attainable when the shifted factor subspaces are sufficiently diverse.*

D.8.2 RESIDUAL ACCUMULATION ACROSS LAYERS

The propagated writes are accumulated according to

$$\boldsymbol{P}^{\ell} = \lambda_{\ell}\boldsymbol{P}^{\ell-1} + \mathcal{A}_R(\boldsymbol{\Delta}^{\ell}), \qquad \lambda_{\ell} = \sigma(a_{\ell}). \tag{63}$$

Define

$$c_0^{\ell} = \prod_{u=1}^{\ell} \lambda_u, \tag{64}$$

$$c_t^{\ell} = \prod_{u=t+1}^{\ell} \lambda_u, \qquad 1 \le t \le \ell, \tag{65}$$

where an empty product is equal to one, so $c_{\ell}^{\ell} = 1$. Unrolling equation 63 gives

$$\boldsymbol{P}^{\ell} = c_0^{\ell}\boldsymbol{P}^0 + \sum_{t=1}^{\ell} c_t^{\ell}\mathcal{A}_R(\boldsymbol{\Delta}^t). \tag{66}$$

To isolate the contribution of the learned writes from the fixed initial prior, define

$$\widetilde{\boldsymbol{P}}^{\ell} = \boldsymbol{P}^{\ell} - c_0^{\ell}\boldsymbol{P}^0. \tag{67}$$

Then

$$\widetilde{\boldsymbol{P}}^{\ell} = \sum_{t=1}^{\ell} c_t^{\ell}\mathcal{A}_R(\boldsymbol{\Delta}^t). \tag{68}$$

Proposition 2 (Rank capacity of the accumulated pair state)**.** *The learned component of the pair state satisfies*

$$\operatorname{rank}\left(\widetilde{\boldsymbol{P}}^{\ell}\right) \le \min\{L, \ell(2R+1)\rho\} \le \min\{L, \ell(2R+1)r\}. \tag{69}$$

Suppose that $\boldsymbol{D_s}$ is nonsingular, all propagation coefficients w_q are nonzero, and

$$\ell(2R+1)r \le L. \tag{70}$$

If the complete shifted-factor bank

$$\mathcal{F}_R^{1:\ell} = \begin{bmatrix}\mathcal{F}_R^1 & \cdots & \mathcal{F}_R^{\ell}\end{bmatrix} \in \mathbb{R}^{L\times\ell(2R+1)r} \tag{71}$$

has full column rank, then

$$\operatorname{rank}\left(\widetilde{\boldsymbol{P}}^{\ell}\right) = \ell(2R+1)r. \tag{72}$$

Thus, before saturation at the ambient dimension L, the attainable rank can grow by $(2R+1)r$ with every accumulated layer.

Proof. Applying Proposition 1 to every layer gives

$$\operatorname{rank}\left(\mathcal{A}_R(\boldsymbol{\Delta}^t)\right) \le (2R+1)\rho. \tag{73}$$

Using equation 68 and rank subadditivity,

$$\operatorname{rank}\left(\widetilde{\boldsymbol{P}}^{\ell}\right) \le \sum_{t=1}^{\ell} \operatorname{rank}\left(c_t^{\ell}\mathcal{A}_R(\boldsymbol{\Delta}^t)\right) \tag{74}$$

$$\le \ell(2R+1)\rho. \tag{75}$$

The ambient bound $\operatorname{rank}(\widetilde{\boldsymbol{P}}^{\ell}) \le L$ gives equation 69.

For the equality case, let $\boldsymbol{C}_R$ be defined as in equation 60, and define the block-diagonal matrix

$$\boldsymbol{C}_R^{1:\ell} = \operatorname{diag}\left(c_1^{\ell}\boldsymbol{C}_R, \ldots, c_{\ell}^{\ell}\boldsymbol{C}_R\right). \tag{76}$$

Equation 68 can be written as

$$\widetilde{\boldsymbol{P}}^{\ell} = \frac{1}{\sqrt{r}} \mathcal{F}_R^{1:\ell} \boldsymbol{C}_R^{1:\ell} (\mathcal{F}_R^{1:\ell})^{\top}. \tag{77}$$

Because $\lambda_u = \sigma(a_u) \in (0, 1)$, every c_t^{ℓ} is strictly positive. Under the stated assumptions, $\boldsymbol{C}_R$ and hence $\boldsymbol{C}_R^{1:\ell}$ are nonsingular. Since $\mathcal{F}_R^{1:\ell}$ has full column rank, the rectangular congruence in equation 77 preserves the rank of $\boldsymbol{C}_R^{1:\ell}$. Therefore,

$$\operatorname{rank}\left(\widetilde{\boldsymbol{P}}^{\ell}\right) = \ell(2R+1)r. \tag{78}$$

□

Remark 3 (Approaching full rank)**.** *Proposition 2 establishes exact linear growth while the total number of factor directions does not exceed L. Once $\ell(2R+1)r \geq L$, the accumulated state can in principle be full rank. Full rank is not guaranteed solely by the factor bank having full row rank because the signed coupling matrix is indefinite and cancellations may occur. It is attained whenever the restriction of the coupling form $\boldsymbol{C}_R^{1:\ell}$ to the row space of $\mathcal{F}_R^{1:\ell}$ is nondegenerate.*