

A dual context-aware basecaller for nanopore direct RNA sequencing

Corresponding Author: Professor Zexuan Zhu

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The authors present Coral, a basecaller for nanopore DRS reads. Coral uses an encoder-decoder architecture with auto-regressive to learn the contextual dependencies at both signal and sequence levels, outperforming the encoder-CTC architecture and the compared methods (Guppy and RODAN) in basecalling accuracy. The authors demonstrate that the improved basecalling accuracy benefits several downstream analyses, including transcript quantification, ASE analysis and haplotype phasing. I agree with the authors that this work provide a benchmark for accuracy in nanopore DRS applications. However, I also have several serious concerns.

1. I believe one of the main contributions of this work is the auto-regressive decoder which considers the context dependencies at both signal and sequence levels. The authors have shown the next base prediction accuracy given a correct 5-mer context (Fig. 2g). I wonder if a 5-mer was wrongly predicted, how would it influence the prediction of the next 5-mers or bases?

2. The datasets used in this study are either RNA001 or RNA002 DRS reads. However, both RNA001 and RNA002 are now considered outdated, replaced by the RNA004 kit. There are already publicly available RNA004 data now. It is necessary to test Coral using RNA004 data.

- Following this, I suggest that the authors develop the Coral tool to support POD5 format.

3. The authors compared Coral with Guppy and RODAN. However, ONT has already released another basecaller Dorado for practical use. Also, a third-party basecaller GCRTcall was released and published recently (<https://doi.org/10.3389/fgene.2024.1443532>).

- The authors should compare Coral with Dorado and GCRTcall using the datasets used in this study.

- Dorado, Guppy, and GCRTcall all support RNA004 reads. A comparison on RNA004 data should also be performed between Coral and all other methods.

4. As we all know, basecalling of nanopore reads is a very computationally intensive task, so computational efficiency is an important consideration in the development of basecalling tools. However, I didn't find any description about the computational efficiency of Coral.

- Does Coral run faster and need less memory than other methods? The author should compare the time and memory usage of different tools.

- The authors demonstrated that the auto-regressive decoder in Coral outperforms the CTC-decoder in decoding accuracy. As CTC enables parallel decoding, the authors may also compare the two approaches in terms of computational efficiency for further investigation.

Minor

1. Regarding the tool, I suggest the authors output some statistics in the log/stderr of Coral, such as the number of reads that processed, successfully basecalled, and failed.

2. In Figure 4, the font in panel (e) is inconsistent with other panels.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)

The manuscript from Xie S. et al described a new base-caller Coral for base-calling direct RNA sequencing data. They show that Coral is more accurate than two other base callers, Guppy and RODAN, and showed that this increase in accuracy enables the authors to identify downstream events such as splicing isoforms and allele-specific transcripts better. While direct RNA sequencing has been useful in the RNA field, its usage has been hampered by its relatively poor accuracy. As such, this work is timely and important. However, additional controls need to be done to truly demonstrate Coral's improvements over existing tools. Below are my main comments:

1) The authors benchmarked Coral against Guppy and RODAN, unfortunately due to the rapid speed of advancements in the nanopore field, Nanopore now has a more recent base caller Dorado. It would be important for the authors to include Dorado in their benchmarking to show that Coral is still the best.

2) It is unclear whether the sequencing in this manuscript is done using the direct RNA sequencing 002 kit or the 004 kit. Nanopore has shown improvements in base calling accuracy for direct RNA sequencing based on the 004 kit vs the 002 kit. The authors would need to show compatibility with 004 kit and whether their base-calling for the reads from the 004 kit are also more accurate.

3) The authors used annotated isoforms from NA12878 as benchmark- while this is useful, it is difficult to know the absolute truth in this dataset. It might be more useful to benchmark on a synthetic RNA pool (such as using SIRV synthetic RNA pools) whereby we know the truth- to determine the ability to detect transcript isoforms and low abundance transcripts. Using this synthetic RNA pool, the authors could also determine the abundance level cutoffs that enable effective detection by the three methods.

4) For the newly identified isoforms (for eg RPL14-201)- can the authors do some experimental validation, such as RT-PCR, followed by sanger sequencing to determine whether this new isoform indeed exists? I have the same comment for fusion genes.

(Remarks on code availability)

NA

Reviewer #3

(Remarks to the Author)

The paper introduces "Coral," a novel dual context-aware basecaller designed to enhance the accuracy of nanopore direct RNA sequencing (DRS). The authors propose an encoder-decoder network architecture that captures dependencies at both the signal and sequence levels, improving basecalling accuracy over existing tools such as Guppy and RODAN. The study demonstrates that Coral delivers up to a 4.48% improvement in median accuracy on human RNA samples, enhances transcript isoform detection by 23.2%, and improves haplotype phasing accuracy by reducing switch errors by 66.5% and Hamming errors by 60%.

Major comments:

-The authors compare Coral with Guppy which is deprecated by Oxford Nanopore. To show usefulness of their model they should compare Coral with Dorado. Also, there are new nanopores and sequence kit SQK-RNA004, which are not analysed in this paper

-The encoder-decoder architecture with latent alignment variables likely introduces computational overhead compared to traditional CTC-based basecallers. While the paper mentions dynamic programming and beam search optimizations, a more detailed discussion of runtime performance and memory consumption would be beneficial.

-Ablation study should be standard for each deep learning model. The authors should perform an ablation study to better understand the relevance of each component to the model

Minor comment:

-Figure 1a it should read "Raw signal" instead "Raw singal"

(Remarks on code availability)

The authors use old python version 3.7.11 which is not supported on newer Mac processors and it would take significant amount of time for a workaround. I suggest authors to update the python version or provide a container image

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

I thank the authors for their hard work. Most of my concerns have been addressed. I have one remaining concern regarding

the computational efficiency of Coral. Fig. S9 indicates that Coral/Coral-FAST requires 2-8 times more processing time than Dorado. I suggest the authors further elaborate in the Discussion section on the specific benefits that users can gain from utilizing Coral over Dorado.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)

Remarks to the author:

The authors make adjustments to the initially proposed architecture by replacing the CNN-RNN encoder with the CNN-Transformer encoder and the LSTM-decoder with the Transformer decoder. These new architecture brings improvements to the initially proposed architecture in both accuracy and decoding time. Coral supports RNA001/002 and RNA004 chemistry, outperforming existing basecallers across various chemistries, datasets, and evaluation metrics. The authors responded to the concerns previously raised, including reviewer 2's comments 1 and 2 regarding the support for RNA004 and its comparison with Dorado. The authors also benchmark on the SIRV benchmark, showing that the obtained results are comparable with Dorado, with improvements compared with other basecallers. Comment 4 is partially addressed with a complementary in-silico and orthogonal validation analysis due to limitations in data access.

However, there are some concerns regarding the generalizability of the model and the transparency of the ablation and model size.

Major comments:

The authors evaluate the proposed models on multiple RNA001/002 datasets and a human RNA004 dataset. While the evaluation sets correspond to different cell lines, and the benefits of having a model trained on as much data as possible are obvious, a comprehensive argumentation of the model's generalization ability needs to include a leave-one-out approach. The authors should evaluate the model on data that was not seen by the model during the training in any way. This would correspond to removing one species from the RNA001/002 training dataset, and evaluating the model on that species, as well as removing one chromosome from the RNA004 training dataset and evaluating the model on that chromosome. Such an analysis would show that the model does not memorize reference-specific details and can be used on previously unseen species without the risk of collapsing.

The authors support the proposed encoder-decoder architecture by performing a set of ablation experiments. However, they do not provide many details of the performed ablation experiments. How is the Transformer-decoder replaced by the CTC-decoder? How does the architecture of the obtained model compare to Dorado, which incorporates Transformer architecture and CTC loss? If Coral's CTC decoder architecture corresponds closely to Dorado, what brings the improvement compared with Dorado; is it the size of the model? How does the size of the ablated model compare to the original Coral model? Are there differences in that aspect that would affect the accuracy difference between the original and ablated model? The proposed encoder architecture includes a 12-fold down-sampling, does that impose issues to CTC decoding?

Minor comments

The authors do not comment on the overall size of the proposed Coral model compared with other architectures. This information would benefit the understanding of all advancements proposed by Coral's architecture.

The authors propose the architecture where the input is 12-fold down-sampled before being passed to the Transformer encoder, and up-sampled at the output of the Transformer. Justification for this approach, along with the details of the up-sampling module, would be beneficial for understanding the architecture.

The proposed regularization loss FRE aims to penalize non-monotonic alignments by comparing the consecutive expected alignment positions β . Can the authors comment on the cases where values obtained for β seem to be monotonic but are actually averaged values of distant and unrelated positions in the signal?

It would be beneficial if the authors could comment and support the design choice for training the Coral-FAST model using part of the weights trained with the original Coral approach. Have you tried training from zero? How do the pretrained weights help the training, given the objective shift?

It would be important to mention that experimental validation of new isoforms and fusion genes is necessary.

(Remarks on code availability)

We have installed and run the code without any issues.

Reviewer #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

(Remarks on code availability)

The installation and running instructions on Github are well written. The conda installation and running the model were seamless. The output was well formatted. I recommend integrating support for running the model on multiple GPUs. Given the complexity and amount of data that we usually have for basecalling, this would make the model more practically applicable.

Version 2:

Reviewer comments:

Reviewer #3

(Remarks to the Author)

We thank the authors for their hard work and extensive experimental efforts. Most of our concerns regarding the model's generalizability have been addressed with the leave-one-out experiments. Additionally, a substantial portion of the model's size and design details have been provided. Our concerns regarding the proposed regularization loss FRE, and training of the Coral-FAST model have been addressed fully.

However, several concerns remain regarding the evaluation setup when comparing the models, as well as the lack of comparisons for the RNA004 model.

1. It would be beneficial if the authors would include the performance of Guppy/Dorado on the same subset of reads for the leave-one-out generalization evaluation for comparison. We understand that these models are not retrained for generalization evaluation, but this would provide the information on the exact comparison.

2, Response Table 3 and Figure 1 show the relation between the model size and performance for RNA002 chemistry, where we see that the full Coral model is ~20 times larger in size, and only Coral-S can compare in size with the comparable performance. On the contrary, we see no such comparison for RNA004, where the size of the Dorado model is far more comparable to that of Coral. We believe those comparisons would be beneficial.

3. The authors motivate the down-then up-sampling approach by the need to preserve sufficient temporal resolution for robust decoding, with the up-sampling done along the embedding dimension rather than the temporal dimension. It should be discussed why a smaller scale down-sampling would not be a good choice compared with this approach, and how exactly this up-sampling approach helps preserve sufficient temporal resolution.

(Remarks on code availability)

we did this in previous reviews

Reviewer #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

(Remarks on code availability)

Version 3:

Reviewer comments:

Reviewer #3

(Remarks to the Author)

The authors have answered all of our comments. We support the publication of this manuscript.

(Remarks on code availability)

no further comments

Reviewer #4

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

(Remarks on code availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Dear Reviewers,

Thank you for your thoughtful and constructive feedback on our manuscript. We sincerely appreciate the opportunity to address your comments and further enhance our work. Following the suggestions, we have carried out additional experiments to demonstrate that **Coral is competitive or superior to Guppy, RODAN, Dorado and GCRTcall on both RNA001/002 and RNA004 reads** (Response Figs. 2-3) in terms of basecalling accuracy. We also expanded validation to **SIRV spike-in controls**, where Coral delivers the highest concordance among all basecallers tested (Response Figure 7). **To address the concerns regarding computational efficiency raised by two reviewers, we have improved the decoding efficiency of Coral by replacing the original LSTM-based latent-alignment decoder with a Transformer decoder that integrates cross-attention alignment and beam-search inference.** This change enables highly parallel decoding while improving alignment precision. Accordingly, the CNN-RNN encoder is updated to a **CNN-Transformer** architecture, so that both encoding and decoding now benefit from multi-head self-attention, improving accuracy. These changes together lead to a reduction of decoding time by nearly **four-fold** and an improvement of basecalling accuracy (Response Fig. 4).

With the new version of Coral, the median read identity on the NA12878 RNA sample increased from 95.56% to 97.69% (Response Fig. 3), which also facilitated the transcript isoforms discovery, haplotype phasing, and allele-specific expression. These refinements are attributed to the better **long-range signal and sequence dependencies** modeling capability of **Transformers than the recurrent networks**.

All requested analyses have been re-performed with the new version of Coral and the results have been updated in this revision. In summary, we have introduced the following major changes:

- 1) To reinforce the cross-species generalization, we have expanded the training corpus from four to eight species with RNA001/002 kits, then tested across six independent datasets. Coral achieved an average median accuracy of approximately **97%**, outperforming other compared basecallers (Fig. 2a–f).
- 2) Coral was trained on a human reference RNA004 sample and evaluated on an independent HEK293T RNA004 dataset. Coral achieved a median accuracy of 99.52% (Fig. 2g), which surpassed other basecallers, including Dorado (98.28%).
- 3) As suggested, we have involved Dorado (v0.8.3) and GCRTcall (v1) in comparison studies. Results were consolidated in revised Fig. 2 (panels a–g) and Response Figs. 2-3. Detailed ablation studies (Tables S4–S7) affirmed the contributions of the

transformer encoder-decoder architecture.

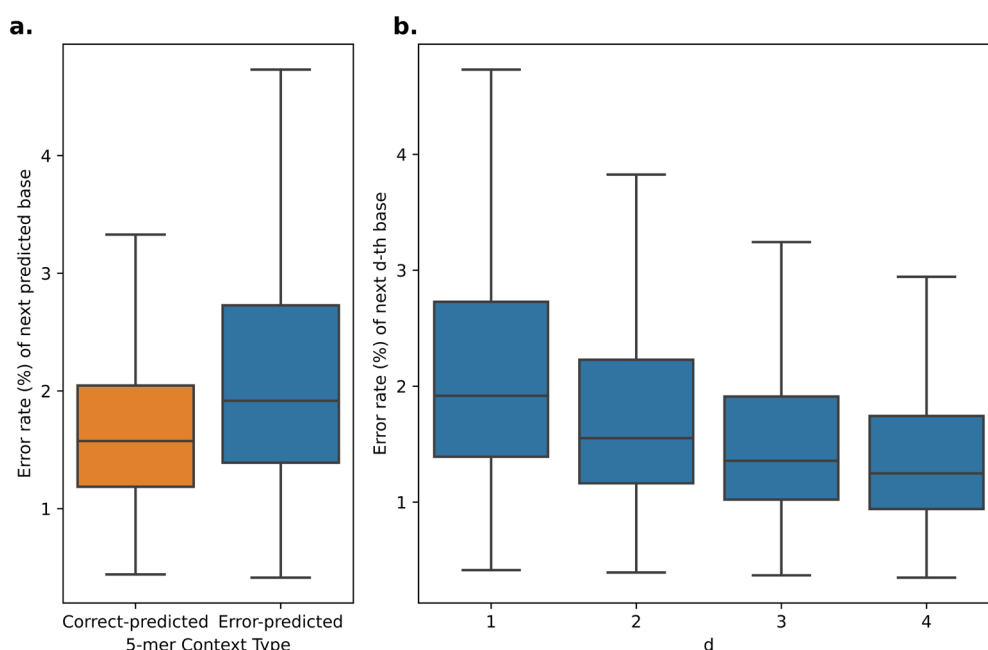
- 4) We benchmarked transcript-level quantification using SIRV spike-ins and Illumina deep RNA-seq data, and performed additional analysis on MCF-7 cell fusion genes (validated by RT-PCR and Sanger sequencing) detection, where Coral showed the highest concordance.
- 5) We now offer two modes, i.e., **Coral** and **Coral-FAST**. The normal mode (Coral), predicting one base at a time, is the used by default to optimize the basecalling accuracy. The FAST mode (Coral-FAST) prioritizes speed by predicting k consecutive bases at a time, optimizing for throughput.

We hope that these improvements, together with the comprehensive revisions that follow, fully address all your concerns and substantially strengthen the manuscript. The detailed point-by-point responses are provided as follows.

Reviewer #1

Comment 1: *I believe one of the main contributions of this work is the auto-regressive decoder which considers the context dependencies at both signal and sequence levels. The authors have shown the next base prediction accuracy given a correct 5-mer context (Fig. 2g). I wonder if a 5-mer was wrongly predicted, how would it influence prediction of the next 5-mers or bases?*

Response: We appreciate this insightful question regarding the robustness of our auto-regressive decoder in the presence of mis-predicted k-mers. To address this concern, we have conducted experiments to evaluate the propagation of errors when k-mer contexts are incorrect. Our findings indicate that an incorrectly predicted 5-mer context increases the error rate of the immediately subsequent base prediction compared to predictions from correct 5-mer contexts (**Response Fig. 1a**). However, as the encoder-decoder architecture of Coral is designed to mitigate error propagation through its comprehensive dual contextual learning, the detrimental impact caused by an initial incorrect 5-mer context progressively diminishes after approximately 4 subsequent bases (**Response Fig. 1b**). The corresponding results and discussion have been included in the revised manuscript (Section “**Effectiveness of modeling contextual dependencies**” and Fig. S7).



Response Figure 1. The impact of error-predicted 5-mer context on the prediction of the subsequent bases. a, Comparison of the error rate for the next predicted base given a correct or erroneous predicted 5-mer context. b, Error rate of d -th subsequent base given an incorrectly predicted 5-mer context.

Comment 2: The datasets used in this study are either RNA001 or RNA002 DRS reads. However, both RNA001 and RNA002 are now considered outdated, replaced by the RNA004 kit. There are already publicly available RNA004 data now. It is necessary to test Coral using RNA004 data. – Following this, I suggest that the authors develop the Coral tool to support POD5 format.

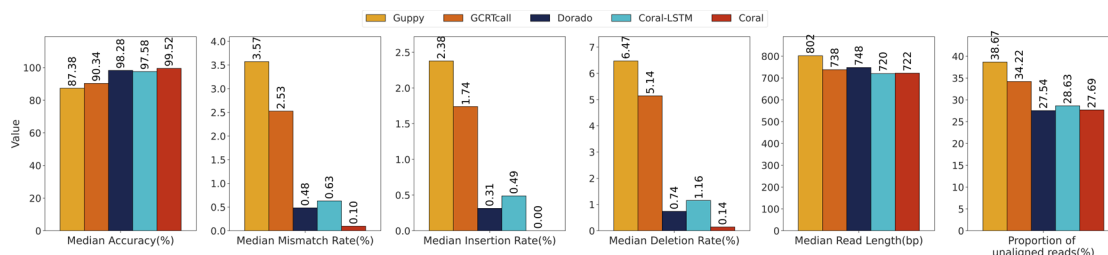
Response: We are grateful for this constructive recommendation. We have evaluated the performance of Coral on RNA004 data, and have extended Coral to natively support the POD5 file format. The previous version of Coral with LSTM-based decoder, denoted as Coral-LSTM, is also included in the comparison to showcase the improvement of the new version. Full details are provided below.

The RNA004 kit represents a substantial advance over RNA001/002 chemistries, featuring a new E8.2 motor protein and R10.4.1 nanopores operating at an elevated sampling rate of 4 kHz (vs. 3 kHz for RNA001/002). This upgrade yields faster strand translocation, enhanced signal resolution, and increased throughput. These changes fundamentally alter the raw signal landscape, rendering models optimized for older chemistries suboptimal. Accordingly, we trained a dedicated Coral model for RNA004 data to maintain optimal basecalling accuracy.

To develop the RNA004-specific model, we leveraged publicly available Universal Human Reference RNA data (accession ERX12369345) generated using R10.4.1 flow

cells. Reads were first basecalled using the Dorado 'rna004_130bps_sup@v5.1.0' model, aligned to the human transcriptome reference, and filtered to high-quality signal chunks (length 12,000 points, alignment accuracy $\geq 98\%$) for training. Model performance was subsequently evaluated on an independent HEK293T RNA004 run from the Singapore Nanopore Expression Project (Sample ID: SGNex_Hek293T_directRNA_replicate5_run1, accessed on [2024-11-21] at registry.opendata.aws/sg-nex-data). We compared the basecalling accuracy of Coral and other state-of-the-art basecallers including Dorado, Guppy, and GCRTcall. Note that RODAN was excluded as it does not support RNA004 chemistry. On this benchmark dataset, Coral achieved a median read accuracy of 99.52 %, surpassing Dorado (98.28 %) and also outperforming both Guppy and GCRTcall (**Response Fig. 2**). Coral also exhibited the lowest median mismatch (0.10 %), insertion (0.00 %) and deletion (0.14%) rates, while maintaining read length and the proportion of unaligned reads comparable to Dorado. The metrics demonstrated Coral's robust adaptation to RNA004-specific signal characteristics. The RNA004-specific results are now included in the revised manuscript (Results section, “**Basecalling accuracy**”, Fig. 2g).

In parallel, we extended Coral's I/O layer to stream data directly from POD5 archives, complementing the existing FAST5 support. The updated software release and documentation reflect this change.



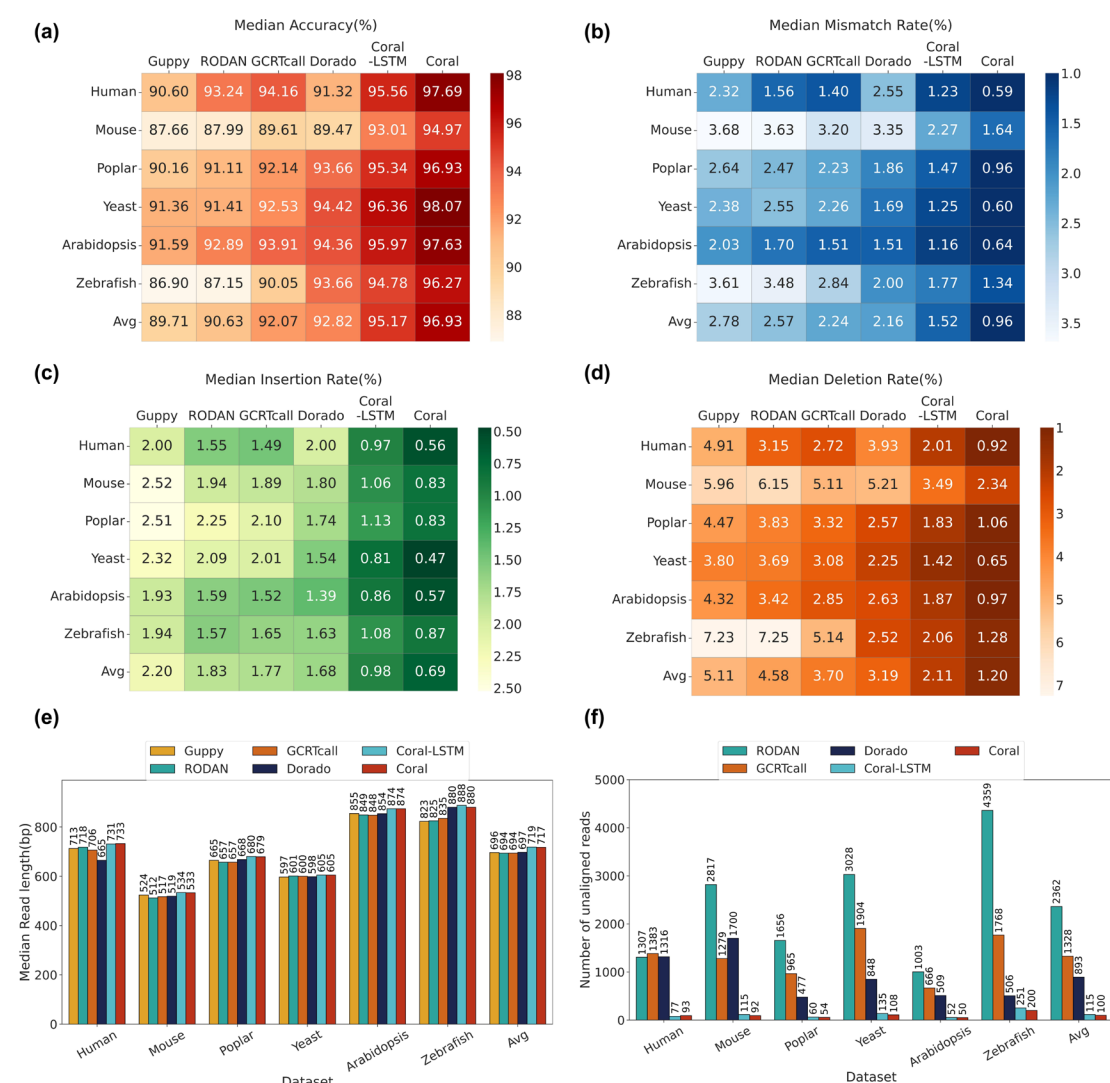
Response Figure 2. Median accuracy/mismatch/insertion/deletion/read length and unaligned reads proportion of Guppy/GCRTcall/Dorado/Coral-LSTM/Coral in the HEK293T RNA004 sample.

Comment 3: *The authors compared Coral with Guppy and RODAN. However, ONT has already released another basecaller Dorado for practical use. Also, a third-party basecaller GCRTcall was released and published recently (<https://doi.org/10.3389/fgene.2024.1443532>).*

- *The authors should compare Coral with Dorado and GCRTcall using the datasets used in this study.*
- *Dorado, Guppy, and GCRTcall all support RNA004 reads. A comparison on RNA004 data should also be performed between Coral and all other methods.*

Response: Thank you for the suggestions. We have incorporated the latest version of Dorado (v0.8.3) and GCRTcall (v1) in all comparison studies in this revision. For the RNA001/002 chemistry, all basecallers were evaluated on six independent test datasets. Note that the previous version of Coral with LSTM-based decoder, denoted as Coral-LSTM, was also included in the comparison to showcase the improvement of the new version. **Response Fig. 3** shows that Coral-LSTM and Coral outperformed all other basecallers across the evaluation metrics. Notably, in the human NA12878 sample dataset, Coral reaches 97.69% median identity; the same trend holds for yeast and *Arabidopsis*, where gains over competing basecallers exceed 2 percentage points. Across all test datasets, Coral achieved an average median read accuracy of approximately 97%, representing a notable advancement in basecalling accuracy for RNA001/002 chemistry. Error-rate decompositions confirm that Coral records the lowest median mismatch (0.96%), insertion (0.69%) and deletion (1.20%) rates, while simultaneously producing the longest aligned reads and the fewest unaligned reads demonstrating that improved basecalling accuracy can facilitate better alignment to the reference sequence. These results have been incorporated in the revised manuscript (Section “**Basecalling accuracy**”, Fig. 2a-f).

For the RNA004 chemistry, a dedicated Coral-LSTM and Coral model was trained on Universal Human Reference RNA sequenced with R10.4.1 pores and validated on an independent SG-NEx HEK293T run. As detailed in our response to Comment 2 (Fig. 2g; Response Fig. 2), Coral achieves a median accuracy of 99.52%, exceeding Dorado (98.28%), Guppy (87.38%) and GCRTcall (90.34%).



Response Figure 3. Basecalling performance of the basecallers on RNA001/002 test datasets. a-d, Heatmap of median accuracy/mismatch/insertion/deletion of six basecallers in the test datasets. Grids with the darker colors are better. The 'Avg' represents the average value across the datasets. e-f, Bar plot of median read length and the number of unaligned reads in the test datasets. Unaligned reads for Guppy are not available as the RNA001/002 test datasets include only reads that were successfully basecalled by Guppy and aligned to the reference sequence.

Comment 4: *As we all know, basecalling of nanopore reads is a very computationally intensive task, so computational efficiency is an important consideration in the development of basecalling tools. However, I didn't find any description about the computational efficiency of Coral.*

- Does Coral run faster and need less memory than other methods? The author should compare the time and memory usage of different tools.
- The authors demonstrated that the auto-regressive decoder in Coral outperforms the CTC-decoder in decoding accuracy. As CTC enables parallel decoding, the authors may also compare the two approaches in terms of computational efficiency for further investigation.

Response: We fully agree that computational efficiency is a critical aspect for nanopore basecalling tools. To improve the computational efficiency of Coral, we introduced architectural changes to Coral by replacing the original LSTM-based decoder with a Transformer decoder of efficient cross-attention for alignment and standard beam search for decoding, to reduce the decoding time. We also developed the Coral-FAST mode, which predicts k consecutive bases per auto-regressive step. We provided both a theoretical complexity analysis (**Response Table 1**) and empirical benchmarks (**Response Figs. 4 and 5**) comparing Coral with the existing basecallers in terms of decoding time, memory usage, and basecalling accuracy. These results have also been discussed in the “**Discussion**” section of the revised manuscript.

Comparison with existing CTC-based basecallers

All CTC-based models—Guppy, RODAN, GCRTcall, Dorado, and Coral-CTC (i.e., Coral with CTC decoder)—share the same theoretical decoding complexity of $O(T * d * V)$, where T is the length of signal features, d is the dimension size, and V is the alphabet size. This is reflected in our benchmarks (**Response Figs. 4 and 5**), where the decoding time of the basecallers fall within a comparable range. However, Coral-CTC consistently outperforms the others in terms of basecalling accuracy—1–5% higher (e.g., Coral-CTC: 94.49% vs Guppy: 89.52%, RODAN: 90.64%, Dorado: 93.46% on A549 RNA002 data)—at a similar time and memory cost. We attribute this to the encoder in Coral, which provides enhanced feature extraction compared to other models.

Auto-regressive decoding vs. CTC

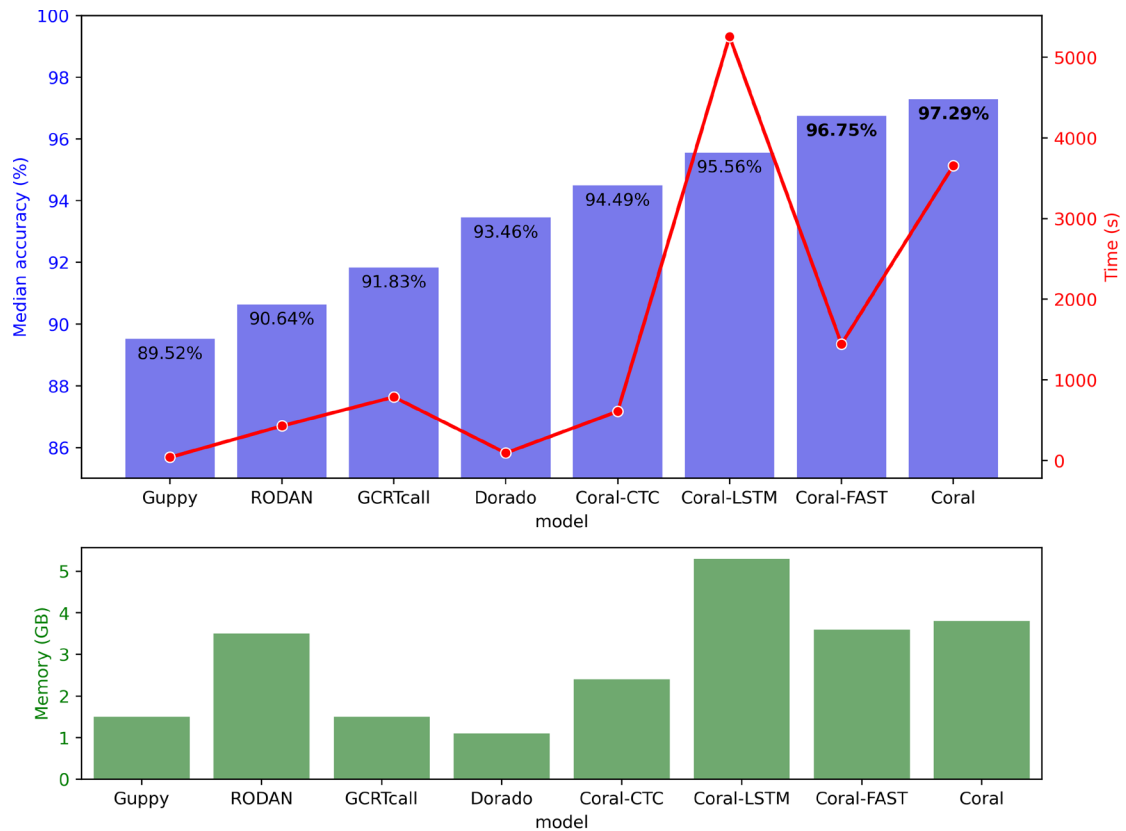
Compared to Coral-CTC, the auto-regressive decoding variants—Coral-LSTM (i.e., Coral with LSTM-based decoder), Coral, and Coral-FAST—all leverage a dual-context decoder that integrates long-range signal- and sequence-level dependencies, resulting in substantial accuracy improvements. Notably, Coral achieved a basecalling accuracy of 97.29% on RNA002 and 99.44% on RNA004 datasets. However, as shown in **Response Table 1**, the computational complexity of these models scales with the sequence length L , leading to a marked increase in decoding time and memory consumption (e.g., ~4000 seconds for Coral, as shown in **Response Fig. 5**). This overhead primarily arises from the long length of ONT reads and the inherent sequential limitations of auto-regressive models. The complexity of Coral-FAST is theoretically reduced to $O(\frac{L}{k} * N * d * d)$. As a result, Coral-FAST achieves a nearly 4× speed-up over Coral-LSTM while further improving accuracy (from 95.56% to 96.75% on RNA002). Compared to Coral, Coral-FAST also demonstrates a 4× speed-up with a negligible reduction in median accuracy (~0.03%) on RNA004, demonstrating a strong trade-off between accuracy and efficiency.

In conclusion, while Coral’s auto-regressive variants incur higher computational cost, they deliver substantial accuracy improvements. Moreover, the progressive architectural refinements—particularly Coral-FAST—strike a better balance between decoding performance and resource efficiency.

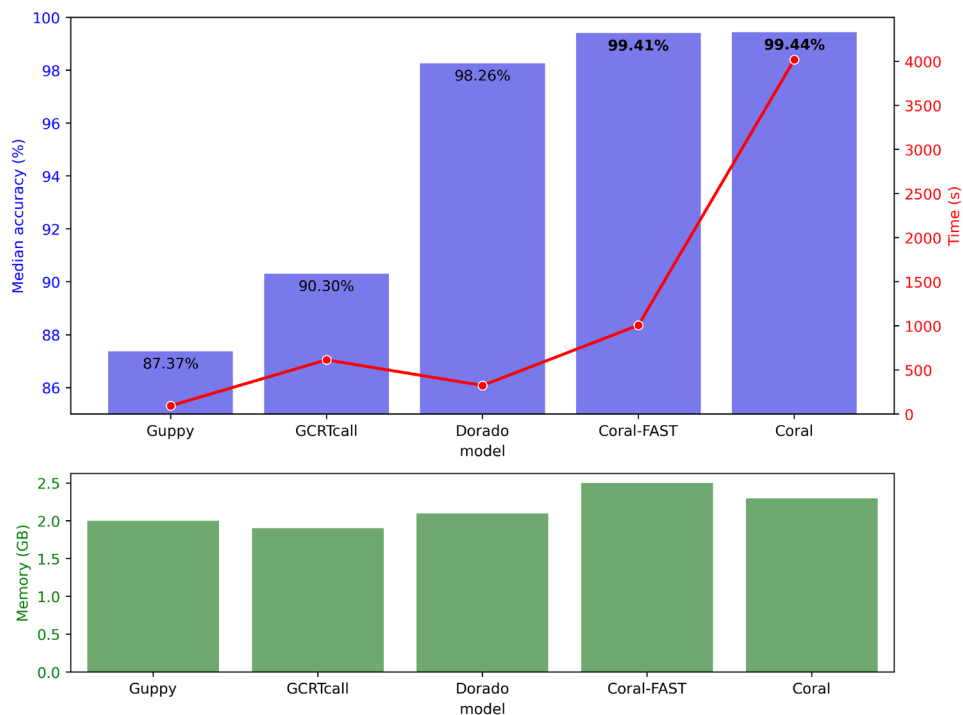
Response Table 1. Theoretical decoding complexity of various basecallers.

Basecaller	Decoder	Complexity
Guppy	CTC	$O(T * d * V)$
RODAN	CTC	$O(T * d * V)$
GCRTcall	CTC	$O(T * d * V)$
Dorado	CTC	$O(T * d * V)$
Coral-CTC	CTC	$O(T * d * V)$
Coral-LSTM	LSTM (dynamic programming and beam search)	$O(L * T * B * d * V)$
Coral	Transformer (beam search, predict single base per auto-regressive step)	$O(L * N * d * d)$
Coral-FAST	Transformer (beam search, predict k bases per auto-regressive step)	$O(\frac{L}{k} * N * d * d)$

(T is length of signal features; d is dimension size; V is alphabet size; L is length of output sequence; B is beam size; N represent the number of transformer decoder layers; $k=5$ for RNA001/002 kits and $k=6$ for RNA004 kit).



Response Figure 4. Median basecalling accuracy (blue bars, left axis) and decoding time (red points, right axis) for each model on sampled reads from A549 RNA002 data. Memory usage is shown below (green bars).



Response Figure 5. Basecalling accuracy and runtime on sampled reads from long-read RNA004 data across different methods. Memory usage is shown below (green bars).

Comment 5: *Regrading the tool, I suggest the authors output some statistics in the log/stderr of Coral, such as the number of reads that processed, successfully basecalled, and failed.*

Response: Thank you for pointing this out. The new version of Coral now supports displaying progress bars and logging information.

Comment 6: *In Figure 4, the font in panel(e) is inconsistent with other panels.*

Response: Thank you for pointing this out. We have revised Fig. 4 in the manuscript to fix this issue.

Reviewer #2

Comment 1: *The authors benched marked Coral against Guppy and RODAN, unfortunately due to the rapid speed of advancements in the nanopore field, Nanopore now has a more recent base caller Dorado. It would be important for the authors to include Dorado in their benchmarking to show that Coral is still the best.*

Response: Many thanks for the valuable suggestion. We have included Dorado (v0.8.3) in all benchmarking analyses presented in the revised manuscript.

For the RNA001/002 chemistry, we expanded our training corpus from four to eight species to enhance model generalizability. Subsequently, we re-evaluated all basecallers, including Dorado, on six independent test datasets. As depicted in **Response Fig. 3**, Coral outperformed other basecallers across various evaluation metrics. Notably, in the human NA12878 sample dataset, Coral achieved a median read identity of 97.69%, surpassing Dorado's performance. Similar trends were observed in yeast and Arabidopsis datasets, where Coral's accuracy exceeded that of Dorado by more than 2 percentage points. Across all test datasets, Coral attained an average median read accuracy of approximately 97%, representing a significant advancement in basecalling accuracy for RNA001/002 chemistry (**Response Fig. 3**).

Furthermore, error-rate decompositions revealed that Coral recorded the lowest median mismatch (0.96%), insertion (0.69%), and deletion (1.20%) rates among all evaluated basecallers. It also produced the longest aligned reads and the fewest unaligned reads, indicating that improved basecalling accuracy facilitates better alignment to reference sequences.

These comprehensive results have been incorporated into the revised manuscript, specifically in the section titled "**Basecalling accuracy**" and illustrated in Fig. 2a–f.

Comment 2: *It is unclear whether the sequencing in this manuscript is done using the direct RNA sequencing 002 kit or the 004 kit. Nanopore has shown improvements in base calling accuracy for direct RNA sequencing based on the 004 kit vs the 002 kit. The authors would need to show compatibility with 004 kit and whether their base-calling for the reads from the 004 kit are also more accurate.*

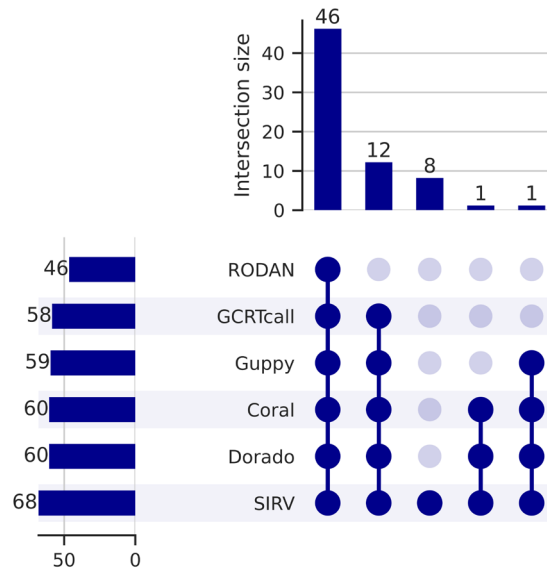
Response: We appreciate your insightful comment regarding the compatibility and performance of Coral with the latest Oxford Nanopore Technologies (ONT) Direct RNA Sequencing Kit (RNA004). As suggested, we have conducted a comprehensive evaluation of Coral's performance on RNA004 data and have incorporated these findings into the revised manuscript (Results section, "**Basecalling accuracy**", Fig. 2g). The RNA004 kit represents a significant advancement over its predecessors (RNA001/002), featuring a new E8.2 motor protein and R10.4.1 nanopores operating at an elevated sampling rate of 4 kHz (compared to 3 kHz for RNA001/002). This upgrade yields faster strand translocation, enhanced signal resolution, and increased throughput, fundamentally altering the raw signal landscape and necessitating tailored basecalling models for optimal performance. To adapt Coral for RNA004 chemistry, we trained a dedicated model using publicly available Universal Human Reference RNA data (accession ERX12369345) generated with R10.4.1 flow cells. Reads were initially basecalled using Dorado's 'rna004_130bps_sup@v5.1.0' model, aligned to the human transcriptome reference, and high-quality signal chunks (length 12,000 points, alignment accuracy $\geq 98\%$) were extracted for training.

We evaluated the performance of Coral on an independent HEK293T RNA004 dataset from the Singapore Nanopore Expression Project (Sample ID: SGNex_Hek293T_directRNA_replicate5_run1, accessed on 2024-11-21 at registry.opendata.aws/sg-nex-data). Comparative analysis included state-of-the-art basecallers such as Dorado, Guppy, and GCRTcall. Note that RODAN was excluded due to its lack of support for RNA004 chemistry. On this benchmark dataset, Coral achieved a median read accuracy of 99.52 %, surpassing Dorado (98.28 %) and markedly outperforming both Guppy and GCRTcall (**Response Fig. 2**). Coral also exhibited the lowest median mismatch (0.10 %), insertion (0.00 %) and deletion (0.14 %) rates, while maintaining read length and the proportion of unaligned reads comparable to Dorado. These metrics demonstrated Coral's robust adaptation to RNA004-specific signal characteristics and support its superior basecalling fidelity.

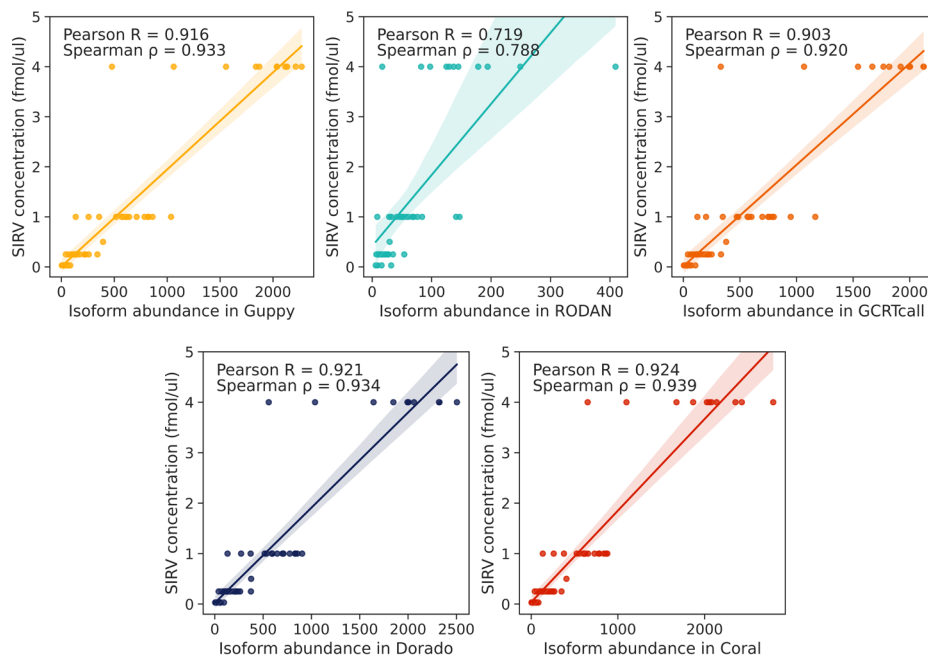
Comment 3: *The authors used annotated isoforms from NA12878 as benchmark – while this is useful, it is difficult to know the absolute truth in this dataset. It might be more useful to benchmark on a synthetic RNA pool (such as using SIRV synthetic RNA pools) whereby we know the truth- to determine the ability to detect transcript isoforms and low abundance transcripts. Using this synthetic RNA pool, the authors could also determine the abundance level cutoffs that enable effective detection by the three methods.*

Response: We sincerely appreciate your valuable suggestions and have carried out a dedicated SIRV benchmark to provide an absolute, ground-truth assessment of Coral. We analyzed the publicly available HEK293T dataset spiked with 25 ng of SIRV-Set 1-E2 RNA (Lexogen, lot no. 001418), which comprises 68 artificial transcripts with known sequences and precisely defined concentrations spanning two orders of magnitude. This dataset offers an ideal benchmark for evaluating transcript identification and quantification accuracy. The analysis involved basecalling raw signal data with Coral and other benchmarked basecallers (Dorado, Guppy, RODAN, GCRTcall), followed by alignment of the resulting basecalled reads to the SIRV isoform reference sequences using *minimap2* in splice-aware mode. The detected transcripts were then compared with the known SIRV transcripts, and we assessed the correlation between the estimated transcript abundances and the known input concentrations (fmol/ul).

Our results highlight the superior performance of Coral in both transcript detection and quantification accuracy. Coral and Dorado recalled the greatest number of transcripts (**Response Fig. 6**), with Coral achieving the highest concordance with ground-truth abundances (**Response Fig. 7**). The results are also integrated in the main text (Section “**Transcript isoform discovery and quantification**”, Fig. 4d) and the Supplementary Materials (Fig. S13, Fig. S14).



Response Figure 6. The number of transcripts discovered by Coral, Guppy, RODAN, Dorado, and GCRTcall compared with the SIRV ground truth. Upset plot displays the intersection size of SIRV transcripts discovered by the specific basecallers indicated by linked filled dots below each bar. The total number of SIRV transcripts identified by different basecallers is shown in the left-hand horizontal bars.

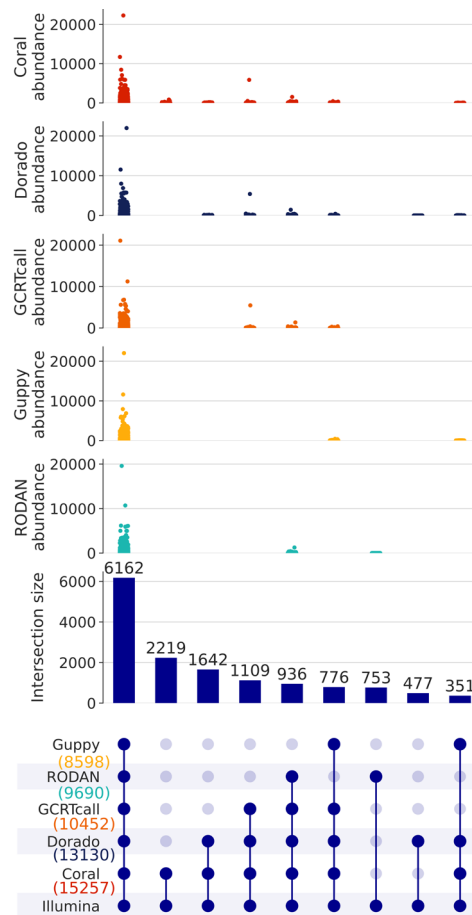


Response Figure 7. The correlation between estimated isoform abundances obtained from the various basecallers (Coral, Guppy, RODAN, GCRTcall, Dorado) and the true SIRV concentration levels.

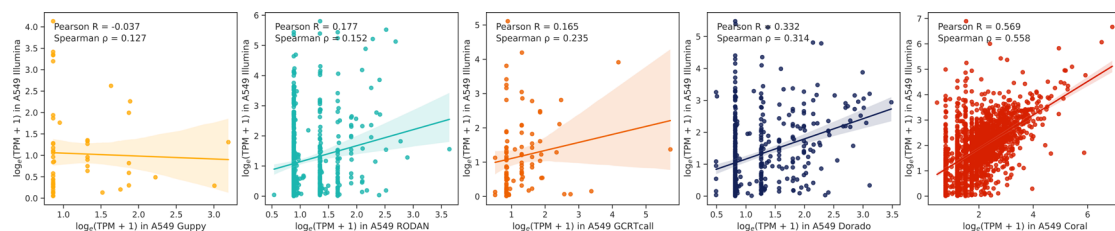
Comment 4: *For the newly identified isoforms (for e.g., RPL14-201)- can the authors do some experimental validation, such as RT-PCR, followed by sanger sequencing to determine whether this new isoform indeed exists? I have the same comment for fusion genes.*

Response: We are grateful for this important suggestion. Because our study is based entirely on publicly available datasets, we do not have access to the original biological material required for new wet-lab assays. Nevertheless, we have carried out two complementary in-silico and orthogonal validation analyses that leverage independent short-read and PCR-validated resources to provide rigorous evidence for the authenticity of Coral-specific isoforms and fusion genes.

Isoform validation with matched Illumina RNA-seq (A549 dataset). We compared isoforms discovered and quantified from ONT reads (basecalled by Coral, Dorado, Guppy, RODAN, GCRTcall) with deep-coverage Illumina short-read RNA-seq from the same A549 samples. Coral identified 15,257 isoforms, including 2,219 isoforms uniquely detected by Coral but absent from other long-read basecallers. Critically, all 2,219 Coral-specific isoforms are also supported by Illumina reads, whereas only 54 Guppy-unique, 99 GCRTcall-unique, 753 RODAN-unique and 477 Dorado-unique isoforms passed the same Illumina evidence filter (**Response Fig. 8**). For Coral-specific isoforms corroborated by Illumina, the correlation between long-read and short-read expression estimates is the strongest among all basecallers (Pearson $R = 0.569$, Spearman $\rho = 0.558$), indicating reliable abundance estimation down to low-expression levels (**Response Fig. 9**). These results demonstrate that the additional isoforms revealed exclusively by Coral are well supported by an independent Illumina sequencing modality. And the additional validation and results are also integrated in our revised manuscript Section “**Transcript isoform discovery and quantification**” (Fig. 4c) and the Supplementary Materials (Fig. S10, Fig. S11).



Response Figure 8. Overlap analysis of detected isoforms across basecallers and Illumina RNA-seq in the A549 dataset. Upset plot (bottom) displays the intersection size of annotated transcripts discovered by Illumina and the specific basecallers indicated by linked filled dots below each bar. Intersection sets smaller than 300 elements are not displayed. The total number of annotated transcripts identified by different basecallers is shown in parentheses, with each method distinguished by a unique color. Strip plot (upper) shows the abundance distribution of each basecaller across different annotated transcript sets.

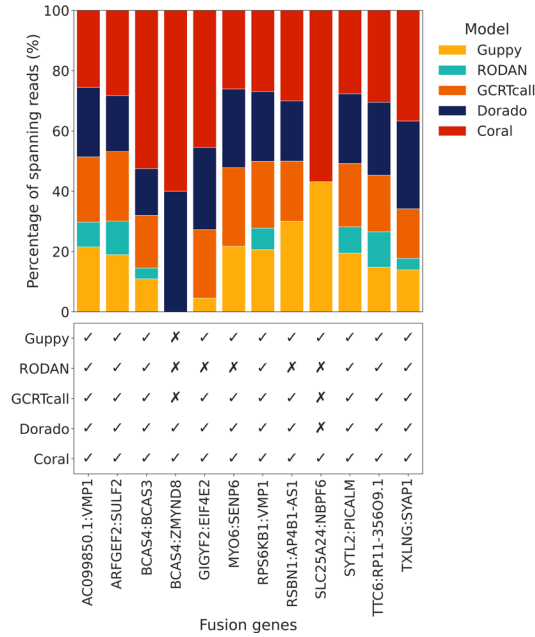


Response Figure 9. Correlation analysis of transcript abundance estimates between ONT and Illumina RNA-seq in the A549 dataset. Scatter plot shows the correlation between transcript abundance estimates obtained from ONT long-read data (using ESPRESSO) basecalled by Coral, Guppy, RODAN, Dorado, and GCRTcall and those from Illumina RNA-seq reads (using Salmon), focusing on transcripts uniquely detected by each basecaller but also identified in the Illumina data from the A549 dataset.

Fusion-gene validation with an RT-PCR/Sanger-confirmed MCF-7 dataset. We analyzed direct-RNA ONT data from the MCF-7 breast-cancer cell line, for which 12 fusion genes have been experimentally confirmed by RT-PCR and Sanger sequencing^[1]. Reads basecalled by each method were processed with JAFFAL¹ for fusion discovery. Notably reads basecalled by Coral achieved the highest median read identity (94.78%, Table S10 in supplementary), providing a high-quality input for downstream fusion detection. When compared the JAFFAL results to previously validated fusion events in the Mitelman database, Coral identified 44 known fusion genes, exceeding Guppy's 29, RODAN's 20, GCRTcall's 33, and Dorado's 36. For further validation against experimentally confirmed events, we utilized findings from a recent publication^[1]. This study experimentally validated 12 specific fusion genes within the same **MCF-7** sample using RT-PCR followed by Sanger sequencing. These 12 PCR-confirmed fusion genes served as a high-confidence ground-truth set for our evaluation. As shown in **Response Fig. 10**, Coral successfully identified all 12 (100%) of the experimentally validated fusion genes. This outperformed other basecallers: Guppy and Dorado both detected 11 out of 12 (91.7%), while RODAN detected the fewest. RODAN's lower detection rate may be due to its significantly shorter read lengths (Table S10 in supplementary). Coral also provides the largest proportion of spanning reads aligning across these validated fusion gene breakpoints, supporting robust breakpoint evidence. These analyses are also added to the revision in Section “**Transcript isoform discovery and quantification**” (Fig. 5c, Fig. S16). These data confirm that Coral's improved base-level accuracy translates into superior detection sensitivity for bona-fide fusion transcripts.

Nevertheless, we fully agree that orthogonal wet-lab confirmation remains the gold standard. For forthcoming work, we are partnering with wet-lab groups to perform validation on newly discovered barley and human isoforms identified by Coral. We hope this comprehensive validation strategy satisfies the reviewer's request and further underlines Coral's ability to identify genuine, low-abundance transcript isoforms and fusion events.

¹ Davidson, N. M. et al. JAFFAL: detecting fusion genes with long-read transcriptome sequencing [J]. Genome Biol, 2022, 23(10).



Response Figure 10. The detection status and proportion of spanning reads of 12 specific MCF-7 cell fusion genes (validated by RT-PCR and Sanger sequencing) using different basecallers.

Reviewer #3

Comment 1: *The authors compare Coral with Guppy which is deprecated by Oxford Nanopore. To show usefulness of their model they should compare Coral with Dorado. Also, there are new nanopores and sequence kit SQK-RNA004, which are not analyzed in this paper*

Response: Thank you for pointing this out. In this revised manuscript we have benchmarked Coral against Dorado as well as other state-of-the-art and third-party basecallers on both RNA001/002 and RNA004 datasets.

For the RNA001/002 chemistry, we broadened training corpus from four to eight species, then re-evaluated all basecallers on six independent test datasets. **Response Fig. 3** shows that Coral outperformed other basecallers across evaluation metrics. Notably, in the human NA12878 sample dataset, Coral reaches 97.69% median identity; the same trend holds for yeast and *Arabidopsis*, where gains over competing callers exceed 2 percentage points. Across all test datasets, Coral achieved an average median read accuracy of approximately 97%, representing a notable advancement in basecalling accuracy for RNA001/002 chemistry. Error-rate decompositions confirm that Coral records the lowest median mismatch (0.96%), insertion (0.69%) and deletion

(1.20%) rates, while simultaneously producing the longest aligned reads and the fewest unaligned reads demonstrating that improved basecalling accuracy can facilitate better alignment to the reference sequence. These results were also incorporated in the revised manuscript (Section “**Basecalling accuracy**”, Fig. 2a-f).

For the RNA004 chemistry, a dedicated Coral model was trained on Universal Human Reference RNA sequenced with R10.4.1 pores and validated on an independent SG-NEx HEK293T run. We compared Coral with other state-of-the-art basecallers including Dorado, Guppy, and GCRTcall. Note that RODAN was excluded as it does not support RNA004 chemistry. On this benchmark dataset, Coral achieved a median read accuracy of 99.52 %, surpassing Dorado (98.28 %) and markedly outperforming both Guppy and GCRTcall (**Response Fig. 2**). Coral also exhibited the lowest median mismatch (0.10 %), insertion (0.00 %) and deletion (0.14 %) rates, while maintaining read length and the proportion of unaligned reads comparable to Dorado. These metrics demonstrated Coral’s robust adaptation to RNA004-specific signal characteristics and support its superior basecalling fidelity. These RNA004-specific results are now included in the revised manuscript (Results section, “**Basecalling accuracy**”, Fig. 2g).

***Comment 2:** The encoder-decoder architecture with latent alignment variables likely introduces computational overhead compared to traditional CTC-based basecallers. While the paper mentions dynamic programming and beam search optimizations, a more detailed discussion of runtime performance and memory consumption would be beneficial.*

Response: We appreciate your insightful observation regarding the computational overhead introduced by Coral's encoder-decoder architecture with latent alignment variables (Coral-LSTM). To improve the computational efficiency of Coral, we introduced architectural changes to Coral by replacing the original LSTM-based decoder with a Transformer decoder with efficient cross-attention for alignment and standard beam search for decoding, to reduce the decoding time, and developed the Coral-FAST mode, which predicts k consecutive bases per auto-regressive step. We provide both a theoretical complexity analysis (**Response Table 1**) and empirical benchmarks (**Response Figs. 4 and 5**) comparing Coral with existing basecallers in terms of decoding time, memory usage, and basecalling accuracy. We have also incorporated this analysis into the “**Discussion**” section of the revised manuscript.

Comparison with existing CTC-based basecallers

All CTC-based models—Guppy, RODAN, GCRTcall, Dorado, and Coral-CTC (i.e., Coral with CTC decoder)—share the same theoretical decoding complexity of $O(T * \dots)$

$d * V$), where T is the length of signal features, d is the dimension size, and V is the alphabet size. This is reflected in our benchmarks (**Response Figs. 4 and 5**), where the decoding time of the basecallers fall within a comparable range. However, Coral-CTC consistently outperforms the others in terms of basecalling accuracy—1–5% higher (e.g., Coral-CTC: 94.49% vs Guppy: 89.52%, RODAN: 90.64%, Dorado: 93.46% on A549 RNA002 data)—at a similar time and memory cost. We attribute this to the encoder in Coral, which provides enhanced feature extraction compared to other models.

Auto-regressive decoding vs. CTC

Compared to Coral-CTC, our auto-regressive decoding variants—Coral-LSTM (i.e., Coral with LSTM-based decoder), Coral, and Coral-FAST—all leverage a dual-context decoder that integrates long-range signal- and sequence-level dependencies, resulting in substantial accuracy improvements. Notably, Coral achieved a basecalling accuracy of 97.29% on RNA002 and 99.44% on RNA004 datasets. However, as shown in **Response Table 1**, the computational complexity of these models scales with the sequence length L , leading to a marked increase in decoding time and memory consumption (e.g., ~4000 seconds for Coral, as shown in **Response Fig. 5**). This overhead primarily arises from the long length of ONT reads and the inherent sequential limitations of auto-regressive models. The complexity of Coral-FAST is theoretically reduced to $O(\frac{L}{k} * N * d * d)$. As a result, Coral-FAST achieves a nearly 4× speed-up over Coral-LSTM while further improving accuracy (from 95.56% to 96.75% on RNA002). Compared to Coral, Coral-FAST also demonstrates a 4× speed-up with a negligible reduction in median accuracy (~0.03%) on RNA004, demonstrating a strong trade-off between accuracy and efficiency.

In conclusion, while Coral’s auto-regressive variants incur higher computational cost, they deliver substantial accuracy improvements. Moreover, the progressive architectural refinements—particularly Coral-FAST—strike a better balance between decoding performance and resource efficiency.

Comment 3: *Ablation study should be standard for each deep learning model. The authors should perform an ablation study to better understand the relevance of each component to the model.*

Response: Thank you for the suggestions. We have now performed an ablation study for the Coral architecture, and the results are presented in the revised manuscript in Section “**Effectiveness of modeling contextual dependencies**” and Tables S4, S5, S6 and S7 in the Supplementary Material. The study investigates the impact of key

components, including:

First, different decoder architectures (Transformer, LSTM, and CTC) were evaluated while keeping the CNN-Transformer encoder fixed (**Response Table 2**). The Transformer decoder and LSTM decoder consider both signal and sequence context dependencies, while the CTC decoder considers only the context dependencies at the signal level. The transformer decoder outperforms both LSTM and CTC decoders, which highlights the benefit of the Transformer architecture for decoding.

Second, variations in the encoder design were tested with a fixed Transformer decoder. The CNN-Transformer encoder demonstrated superior performance compared to a CNN-LSTM encoder (**Response Table 3**), validating the choice of a fully Transformer-based attention mechanism in the encoder.

Third, the effect of symbol granularity was investigated by comparing a model predicting a single base per step (Coral) with one predicting five consecutive bases per step (Coral-FAST). While the Coral-FAST model is designed for higher throughput, the Coral model yielded slightly better accuracy (**Response Table 4**).

Finally, the impact of decoder depth was examined by varying the number of decoder layers (3, 6, 9, and 12 layers, corresponding to 1, 2, 3, and 4 decoder blocks, respectively). Performance generally improved with increasing layers, with the 9-layer configuration achieving the highest accuracy across multiple datasets. Further increasing to 12 layers resulted in either comparable or slightly reduced accuracy, suggesting that performance may reach a saturation point (**Response Table 5**). These studies collectively affirm the design choices of the Coral architecture.

Comment 4: *Figure 1a it should read “Raw signal” instead “Raw singal”*

Response: Thank you for pointing this out. We have corrected it in this revision.

Comment 5: *The authors use old python version 3.7.11 which is not supported on newer Mac processors and it would take significant amount of time for a workaround. I suggest authors to update the python version or provide a container image.*

Response: Thank you for the suggestion. The new version of Coral now supports Python 3.10. We also provide the docker container image to facilitate rapid software installation.

Response Table 2. Basecalling performance of Coral decoder variants on test datasets.

Dataset	Reads Count	Decoder network	Read Accuracy (%)		Mismatch (%)		Insertion (%)		Deletion (%)		Read Length (bp)		Unaligned Reads
Human	100000	Transformer	96.93	97.69	0.84	0.59	0.84	0.56	1.4	0.92	991	733	93
		LSTM	94.84	95.56	1.45	1.23	1.3	0.97	2.42	2.01	988	731	77
		CTC	94.28	95.03	1.28	1.09	0.88	0.72	3.55	3.03	967	717	226
Mouse	100000	Transformer	94.14	94.97	1.89	1.64	1.02	0.83	2.94	2.34	657	533	92
		LSTM	92.38	93.01	2.48	2.27	1.26	1.06	3.87	3.49	657	534	115
		CTC	91.46	92.13	2.25	2.06	0.93	0.81	5.37	4.87	638	518	369
Yeast	100000	Transformer	96.92	98.07	1.02	0.6	0.79	0.47	1.27	0.65	855	605	108
		LSTM	95.44	96.36	1.63	1.25	1.1	0.81	1.83	1.42	855	605	135
		CTC	94.67	95.6	1.49	1.15	0.89	0.72	2.95	2.39	842	597	355
Arabidopsis	100000	Transformer	96.76	97.63	0.92	0.64	0.9	0.57	1.42	0.97	984	874	50
		LSTM	95.07	95.97	1.45	1.16	1.22	0.86	2.27	1.87	983	874	52
		CTC	94.39	95.3	1.26	1.01	0.88	0.64	3.47	2.92	961	852	166
Poplar	100000	Transformer	95.79	96.93	1.31	0.96	1.34	0.83	1.55	1.06	808	679	54
		LSTM	94.28	95.34	1.81	1.47	1.66	1.13	2.25	1.83	809	680	60
		CTC	93.49	94.59	1.69	1.38	1.25	0.86	3.57	2.98	789	662	201
Zebrafish	100000	Transformer	95.44	96.27	1.65	1.34	1.2	0.87	1.72	1.28	1013	880	200
		LSTM	93.98	94.78	2.05	1.77	1.51	1.08	2.46	2.06	1019	888	251
		CTC	93.18	93.96	1.96	1.69	1.03	0.77	3.82	3.35	988	858	385

The best results are marked as bold text. The left and right side of the columns “Read Accuracy”, “Mismatch”, “Insertion”, “Deletion”, and “Read Length” represent the mean and median values respectively.

Response Table 3. Basecalling performance of Coral encoder-variants on test datasets.

Dataset	Reads Count	Encoder network	Read Accuracy (%)		Mismatch (%)		Insertion (%)		Deletion (%)		Read Length (bp)		Unaligned Reads
Human	100000	CNN-Transformer	96.93	97.69	0.84	0.59	0.84	0.56	1.4	0.92	991	733	93
		CNN-LSTM	95.38	96.49	0.97	0.78	1.19	0.94	2.46	1.48	970	714	330
Mouse	100000	CNN-Transformer	94.14	94.97	1.89	1.64	1.02	0.83	2.94	2.34	657	533	92
		CNN-LSTM	92.83	93.76	2.02	1.83	1.33	1.12	3.82	2.98	636	519	459
Yeast	100000	CNN-Transformer	96.92	98.07	1.02	0.6	0.79	0.47	1.27	0.65	855	605	108
		CNN-LSTM	95.55	96.89	1.15	0.82	1.2	0.92	2.1	1.13	836	591	500
Arabidopsis	100000	CNN-Transformer	96.76	97.63	0.92	0.64	0.9	0.57	1.42	0.97	984	874	50
		CNN-LSTM	95.53	96.59	1.03	0.8	1.23	0.92	2.21	1.45	963	853	248
Poplar	100000	CNN-Transformer	95.79	96.93	1.31	0.96	1.34	0.83	1.55	1.06	808	679	54
		CNN-LSTM	94.27	95.65	1.44	1.17	1.76	1.3	2.54	1.58	787	658	352
Zebrafish	100000	CNN-Transformer	95.44	96.27	1.65	1.34	1.2	0.87	1.72	1.28	1013	880	200
		CNN-LSTM	93.69	94.9	1.84	1.57	1.66	1.35	2.81	1.87	988	855	585

The best results are marked as bold text. The left and right side of the columns “Read Accuracy”, “Mismatch”, “Insertion”, “Deletion”, and “Read Length” represent the mean and median values respectively.

Response Table 4. Basecalling performance of Coral (predict single base per step) and Coral-FAST (predict 5 bases per step) on test datasets.

Dataset	Reads Count	Model	Read Accuracy (%)		Mismatch (%)		Insertion (%)		Deletion (%)		Read Length (bp)		Unaligned Reads
Human	100000	Coral	96.93	97.69	0.84	0.59	0.84	0.56	1.4	0.92	991	733	93
		Coral-FAST	96.18	97.23	0.97	0.77	0.82	0.59	2.03	1.14	980	724	160
Mouse	100000	Coral	94.14	94.97	1.89	1.64	1.02	0.83	2.94	2.34	657	533	92
		Coral-FAST	93.71	94.61	2.05	1.84	1.03	0.85	3.21	2.46	651	529	175
Yeast	100000	Coral	96.92	98.07	1.02	0.6	0.79	0.47	1.27	0.65	855	605	108
		Coral-FAST	96.48	97.7	1.17	0.8	0.8	0.51	1.55	0.75	848	601	153
Arabidopsis	100000	Coral	96.76	97.63	0.92	0.64	0.9	0.57	1.42	0.97	984	874	50
		Coral-FAST	96.35	97.42	1	0.74	0.89	0.58	1.76	1.05	976	866	68
Poplar	100000	Coral	95.79	96.93	1.31	0.96	1.34	0.83	1.55	1.06	808	679	54
		Coral-FAST	95.33	96.69	1.36	1.05	1.3	0.82	2	1.13	800	671	104
Zebrafish	100000	Coral	95.44	96.27	1.65	1.34	1.2	0.87	1.72	1.28	1013	880	200
		Coral-FAST	95.19	96.18	1.67	1.38	1.11	0.79	2.03	1.33	1009	877	241

The best results are marked as bold text. The left and right side of the columns “Read Accuracy”, “Mismatch”, “Insertion”, “Deletion”, and “Read Length” represent the mean and median values respectively.

Response Table 5. Basecalling performance of Coral using different decoder layers on test datasets.

Dataset	Reads Count	Decoder layers	Read Accuracy (%)		Mismatch (%)		Insertion (%)		Deletion (%)		Read Length (bp)		Unaligned Reads
Human	100000	3	96.43	97.29	0.94	0.72	0.94	0.66	1.68	1.11	987	730	125
		6	96.69	97.44	0.91	0.68	0.93	0.65	1.48	1.02	991	733	95
		9	97	97.8	0.78	0.54	0.86	0.54	1.36	0.89	991	733	86
		12	96.93	97.69	0.84	0.59	0.84	0.56	1.4	0.92	991	733	93
Mouse	100000	3	92.11	93.97	1.89	1.67	1.03	0.84	4.97	2.99	619	508	469
		6	93.97	94.79	1.97	1.72	1.12	0.94	2.94	2.34	656	532	115
		9	94.4	95.19	1.81	1.56	1.03	0.83	2.76	2.21	658	534	105
		12	94.14	94.97	1.89	1.64	1.02	0.83	2.94	2.34	657	533	92
Yeast	100000	3	96.43	97.65	1.12	0.74	0.88	0.59	1.57	0.8	850	603	148
		6	96.75	97.84	1.08	0.68	0.89	0.56	1.27	0.71	856	606	110
		9	97.12	98.24	0.94	0.51	0.81	0.46	1.13	0.58	857	606	109
		12	96.92	98.07	1.02	0.6	0.79	0.47	1.27	0.65	855	605	108
Arabidopsis	100000	3	96.35	97.36	0.97	0.7	0.96	0.63	1.73	1.11	976	866	62
		6	96.63	97.49	0.96	0.69	0.98	0.65	1.42	1.01	984	874	48
		9	96.9	97.76	0.87	0.59	0.93	0.57	1.3	0.9	985	875	46
		12	96.76	97.63	0.92	0.64	0.9	0.57	1.42	0.97	984	874	50
Poplar	100000	3	95.44	96.68	1.35	1.01	1.37	0.88	1.85	1.17	803	674	60
		6	95.67	96.79	1.36	1.01	1.42	0.91	1.56	1.08	809	679	50
		9	95.93	97.06	1.25	0.88	1.35	0.82	1.47	1.01	809	680	41
		12	95.79	96.93	1.31	0.96	1.34	0.83	1.55	1.06	808	679	54
Zebrafish	100000	3	95.31	96.18	1.66	1.36	1.15	0.83	1.88	1.38	1011	878	242
		6	95.42	96.25	1.66	1.35	1.19	0.86	1.74	1.3	1013	881	207
		9	95.58	96.43	1.56	1.25	1.17	0.84	1.68	1.26	1013	881	192
		12	95.44	96.27	1.65	1.34	1.2	0.87	1.72	1.28	1013	880	200

1. The best results are marked as bold text.
2. The left and right side of the columns “Read Accuracy”, “Mismatch”, “Insertion”, “Deletion”, and “Read Length” represent the mean and median values respectively.

Response of Manuscript NCOMMS-24-71768A

We sincerely thank the reviewers for the thorough evaluation of our manuscript and for providing insightful and constructive comments. Your feedback has been invaluable in helping us improve the clarity and rigor of the paper. We have carefully addressed each point raised, revised the manuscript accordingly, and believe that the changes have significantly strengthened the work.

Reviewer #1

Comment 1: *I thank the authors for their hard work. Most of my concerns have been addressed. I have one remaining concern regarding the computational efficiency of Coral. Fig.S9 indicates that Coral/Coral-FAST requires 2-8 times more processing time than Dorado. I suggest the authors further elaborate in the Discussion section on the specific benefits that users can gain from utilizing Coral over Dorado.*

Response: Many thanks for the valuable suggestion. We acknowledge that Coral currently requires more processing time than Dorado. This represents a deliberate design trade-off that prioritizes accuracy over raw computational speed. The primary benefit for users choosing Coral is the ability to train or fine-tune the basecalling model on bespoke datasets. This capability is particularly crucial for novel research applications, such as sequencing non-standard sample types or accurately identifying modified bases, where a proprietary basecaller like Dorado may be suboptimal. Furthermore, Coral's open-source framework dismantles the "black box" nature of commercial basecallers, empowering the scientific community to scrutinize, innovate, and extend the basecalling architecture itself. Therefore, while Dorado is positioned for high-throughput production sequencing, Coral serves the distinct need for a customizable tool that can be tailored to push the boundaries of nanopore technology. Looking ahead, we are actively exploring methods to accelerate Coral's inference speed. Moreover, the implementation of Coral can be further optimized in specific application scenarios, i.e., model quantization, knowledge distillation, and hardware acceleration can be applied to significantly reduce the computational overhead. We have incorporated this into the “**Discussion**” section of the revised manuscript to provide users with a clear understanding of Coral's value proposition.

Reviewer #3

Comment 1: *The authors evaluate the proposed models on multiple RNA001/002 datasets and a human RNA004 dataset. While the evaluation sets correspond to different cell lines, and the benefits of having a model trained on as much data as possible are obvious, a comprehensive argumentation of the model's generalization ability needs to include a leave-one-out approach. The authors should evaluate the*

model on data that was not seen by the model during the training in any way. This would correspond to removing one species from the RNA001/002 training dataset, and evaluating the model on that species, as well as removing one chromosome from the RNA004 training dataset and evaluating the model on that chromosome. Such an analysis would show that the model does not memorize reference-specific details and can be used on previously unseen species without the risk of collapsing.

Response: Thank you very much for the valuable suggestions. We have conducted the leave-one-out cross-validation (LOOCV) experiments to evaluate the generalization capability of the proposed model. Specifically, on the RNA001/002 datasets, we re-trained Coral and other retrainable third-party basecallers, i.e., RODAN and GCRTcall, in six runs, each time excluding one species from the training set and evaluating performance on that species' test reads. For the RNA004 dataset, we re-trained the models after excluding reads from the representative chromosome 22 in the training data and evaluated them on chr22 reads. Note that Guppy and Dorado do not offer a training pipeline, we cannot perform LOOCV evaluations with these two basecallers.

As shown in **Response Table 1**, across the six LOOCV runs on the RNA001/002 datasets, Coral achieved an average median read accuracy of 95.19%, approximately 1.74% decrease compared to the original Coral model. Nevertheless, Coral consistently outperformed the other two basecallers. In the LOOCV with RNA004 dataset, Coral achieved a median accuracy of 97.78%, comparable to Dorado and markedly higher than RODAN (92.44%) and GCRTcall (94.15%) (**Response Table 2**). These results demonstrate only a slight degradation in accuracy when species- or chromosome-specific data are excluded from training, indicating that Coral effectively learns a robust mapping between raw signal and sequence rather than merely memorizing reference-specific patterns. The LOOCV accuracy of Coral is also comparable or superior to that of Guppy and Dorado without LOOCV. The corresponding results and discussions have been included in the revised manuscript and supplementary materials (Section “**Basecalling accuracy**”, **Table S4**, and **Table S5**).

Response Table 1. Basecalling performance of leave-one-out cross-validation on RNA001/002 test datasets.

Dataset (Read Count)	Basecallers	Read Accuracy(%)		Mismatch(%)		Insertion(%)		Deletion(%)		Read Length (bp)		Unaligned Reads
		mean	median	mean	median	mean	median	mean	median	mean	median	
Human (100000)	RODAN	90.45	91.11	2.32	2.09	1.84	1.63	5.39	4.85	847	681	4063
	GCRTcall	91.68	92.69	1.8	1.56	1.87	1.42	4.65	4.07	973	722	1445
	Coral	94.8	95.55	1.5	1.28	1.23	1.01	2.46	2	987	730	117
Mouse (100000)	RODAN	87.78	88.12	3.19	3.05	1.62	1.44	7.4	7.11	498	450	4813
	GCRTcall	89.54	90.19	2.69	2.51	1.67	1.49	6.09	5.6	643	522	911
	Coral	92.69	93.46	2.45	2.23	1.26	1.09	3.6	3.03	652	529	154
Yeast	RODAN	91.15	91.9	2.51	2.21	2.08	1.89	4.25	3.81	652	553	3465

(100000)	GCRTcall	91.94	92.97	2.22	1.87	1.98	1.77	3.86	3.24	847	601	1071
	Coral	94.65	95.83	1.91	1.46	1.36	1.05	2.08	1.57	853	604	273
Arabidops is (100000)	RODAN	91.6	92.29	2.01	1.81	1.85	1.59	4.54	4.13	723	667	1907
	GCRTcall	92.17	93.24	1.74	1.48	1.91	1.5	4.17	3.59	969	859	565
	Coral	95.21	96.15	1.38	1.09	1.02	0.74	2.4	1.9	979	870	65
Poplar (100000)	RODAN	90.87	91.64	2.49	2.24	2.25	1.95	4.38	4	617	559	3025
	GCRTcall	90.85	91.94	2.44	2.11	2.41	2.01	4.3	3.72	797	669	1168
	Coral	94.48	95.58	1.75	1.41	1.56	1.1	2.2	1.73	805	676	70
Zebrafish (100000)	RODAN	88.7	89.27	3.16	2.95	1.75	1.59	6.4	5.92	759	696	2798
	GCRTcall	88.93	89.6	2.96	2.72	1.8	1.54	6.31	5.83	977	848	1537
	Coral	93.7	94.55	2.25	1.95	1.52	1.2	2.53	2.07	1008	875	348

The best results are marked as bold text.

Response Table 2. Basecalling performance of leave-one-out evaluation on chr22 reads of HEK293T RNA004 sample.

Dataset (Read Count)	Basecallers	Read Accuracy(%)		Mismatch(%)		Insertion(%)		Deletion(%)		Read Length (bp)		Unaligned Reads
		mean	median	mean	median	mean	median	mean	median	mean	median	
HEK293T RNA004 chr22 reads (233936)	RODAN	91.31	92.44	1.94	1.72	1.32	1.09	5.44	4.55	1074	875	8327
	GCRTcall	92.86	94.15	1.48	1.27	1.32	0.93	4.34	3.49	1094	880	3545
	Coral	96.54	97.78	0.93	0.70	0.70	0.36	1.83	1.03	1099	869	1750

The best results are marked as bold text.

Comment 2: *The authors support the proposed encoder-decoder architecture by performing a set of ablation experiments. However, they do not provide many details of the performed ablation experiments. How is the Transformer-decoder replaced by the CTC-decoder? How does the architecture of the obtained model compare to Dorado, which incorporates Transformer architecture and CTC loss? If Coral's CTC decoder architecture corresponds closely to Dorado, what brings the improvement compared with Dorado; is it the size of the model? How does the size of the ablated model compare to the original Coral model? Are there differences in that aspect that would affect the accuracy difference between the original and ablated model? The proposed encoder architecture includes a 12-fold down-sampling, does that impose issues to CTC decoding?*

Response: Many thanks for these insightful questions regarding the ablation experiments. To improve the clarity, we have expanded on the methodology and results in the revised manuscript. The detailed architecture and parameter size of ablated models are provided in **Response Table 3**.

In the CTC ablation experiment (Coral-CTC), the autoregressive Transformer decoder was replaced with a single linear projection layer that maps encoder output features to

nucleotide logits, and the model was subsequently trained with the CTC loss while the encoder architecture remained unchanged.

When comparing Coral-CTC and Dorado, it is important to note that Dorado does not employ a single model, but rather utilizes distinct architectures optimized for different sequencing kits. Dorado uses an LSTM-encoder with a CTC-decoder (Dorado-RNA002) for the RNA001/002 data, but a Transformer-encoder with a CTC-decoder (Dorado-RNA004) for the RNA004 data. Coral-CTC is roughly equivalent to Dorado-RNA004, since they have the same architecture and a similar model size. The performance gain observed with Coral-CTC over Dorado-RNA002 on the RNA001/002 datasets is therefore attributable to the combined effects of its more powerful Transformer-based encoder and the associated increase in model parameters. (**Response Figure 1**).

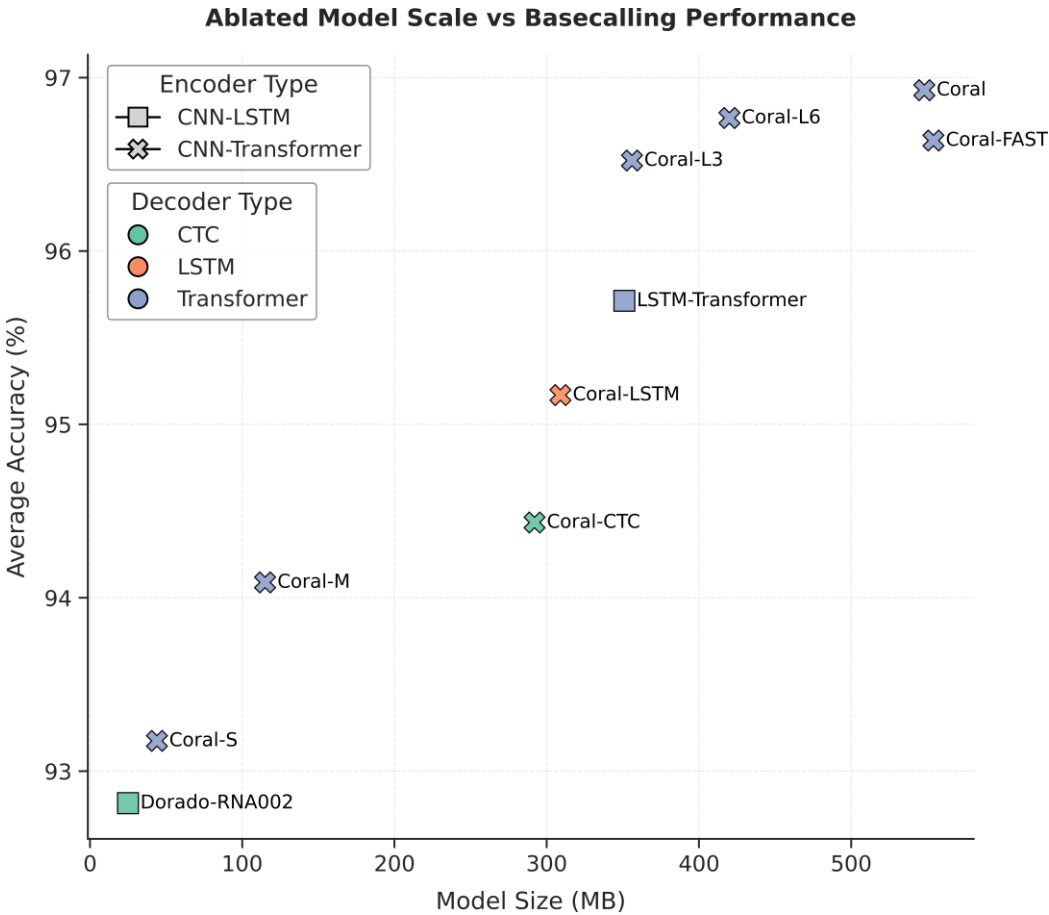
Our ablation studies were designed to systematically disentangle the effects of model architecture and scale (**Response Table 3**). The results, summarized in **Response Figure 1**, demonstrate that performance gains are driven by both factors. First, the performance of Coral scales with model size, as demonstrated by the accuracy gains from Coral-S (44 MB), Coral-M (115 MB), to the larger Coral-Lx variants (356–548 MB, where $x = 3, 6, 12$ represents the number of decoder layers). When the model size reaches a certain scale, performance begins to saturate, likely due to limitations in training data and the inherent complexity of the task—a trend consistent with neural scaling laws. Second, architecture remains a critical determinant when comparing models at a similar parameter scale. For instance, the superior performance of Coral-L3 (356MB) over Coral-CTC (292 MB) and Coral-LSTM (309 MB) affirms the advantage of the autoregressive Transformer decoder, while the performance gap between Coral-L3 and LSTM-Transformer (351MB) highlights the effectiveness of the CNN-Transformer encoder. Together, these comparisons provide a clear justification for our adoption of a full Transformer encoder-decoder architecture.

The 12-fold down-sampling of our encoder architecture risks creating a signal feature sequence too short for stable CTC alignment, as CTC requires the input to be longer than the target nucleotide sequence. To solve this, we added an up-sampling module before the decoder to preserve sufficient temporal resolution for robust decoding by both CTC and our auto-regressive decoder.

All corresponding results and detailed explanations have been provided in the revised manuscript and supplementary materials (Section “**Effectiveness of modeling contextual dependencies**”, Tables S6, S7, S8, and S10, and Fig. S10).

Response Table 3. Architecture and parameter size of ablated models used in RNA001/002 test datasets.

Ablated model	Encoder type	Decoder type	Encoder layers	Decoder layers	Feature dimension	Param size(MB)
LSTM-Transformer	CNN-LSTM	Transformer	5	12	512	351
Coral-CTC	CNN-Transformer	CTC	18	1	512	292
Coral-LSTM	CNN-Transformer	LSTM	18	1	512	309
Coral-S	CNN-Transformer	Transformer	3	2	256	44
Coral-M	CNN-Transformer	Transformer	1	6	384	115
Coral-L3	CNN-Transformer	Transformer	18	3	512	356
Coral-L6	CNN-Transformer	Transformer	18	6	512	420
Coral (i.e., Coral-L12)	CNN-Transformer	Transformer	18	12	512	548
Coral-FAST	CNN-Transformer	Transformer	18	12	512	554



Response Figure 1. Basecalling performance of Coral ablated models with different size on RNA001/002 test datasets. ‘Coral-S’ represents a small-scale variant of the Coral model. ‘Coral-M’ represents a medium-scale variant of the Coral model. ‘Coral-L3’, ‘Coral-L6’ and ‘Coral’ represent large-scale variants of the Coral model with 3/6/12 decoder layers. ‘Coral-CTC’ and ‘Coral-LSTM’ are two Coral decoder variants that replace the Transformer decoder with ‘CTC’ and

‘LSTM’. ‘LSTM-Transformer’ is Coral encoder variant that replace the CNN-Transformer encoder with CNN-LSTM encoder. The basecalling performance is measured using average median accuracy across six RNA001/002 test datasets.

Comment 3: *The authors do not comment on the overall size of the proposed Coral model compared with other architectures. This information would benefit the understanding of all advancements proposed by Coral's architecture.*

Response: Thanks for the precious suggestion. The results of the ablation studies (**Response Table 3** and **Response Figure 1**) demonstrate that performance gains are driven by both model architecture and scale. While accuracy improves with model size for Coral’s Transformer encoder-decoder architecture (e.g., Coral-S vs. Coral-M vs. Coral-L), in line with neural scaling laws, architecture remains a critical determinant. This is evident when comparing models of a similar size: the superior performance of Coral-L3 over Coral-CTC and Coral-LSTM affirms the advantage of the autoregressive Transformer decoder. Similarly, comparing Coral-L3 with LSTM-Transformer highlights the effectiveness of the Transformer encoder. Taken together, these comparisons provide a clear justification for our adoption of a full Transformer encoder-decoder architecture with a larger parameter scale to function optimally. Coral possesses the largest model scale compared to other basecallers (**Response Table 4**). This increased scale reflects the demands of its more expressive architecture, which contributes to the state-of-the-art accuracy observed in our results. We have added this discussion into the revised manuscript and supplementary materials to better contextualize the benefits of our approach (Section “**Effectiveness of modeling contextual dependencies**”, **Tables S6** and **S11**, and **Fig. S10**).

Response Table 4. Architecture and size of basecallers

Basecaller	Encoder	Decoder	Param size (MB)
Guppy	CNN-LSTM	CTC	11
RODAN	CNN	CTC	41
GCRTcall	CNN-Transformer	CTC	349
Dorado	CNN-LSTM (RNA002); CNN-Transformer (RNA004)	CTC	25 (RNA002); 300 (RNA004)
Coral	CNN-Transformer	Transformer (predict single base per auto-regressive step)	548 (RNA002); 612 (RNA004)
Coral-FAST	CNN-Transformer	Transformer (predict k bases per auto- regressive step)	554 (RNA002); 634 (RNA004)

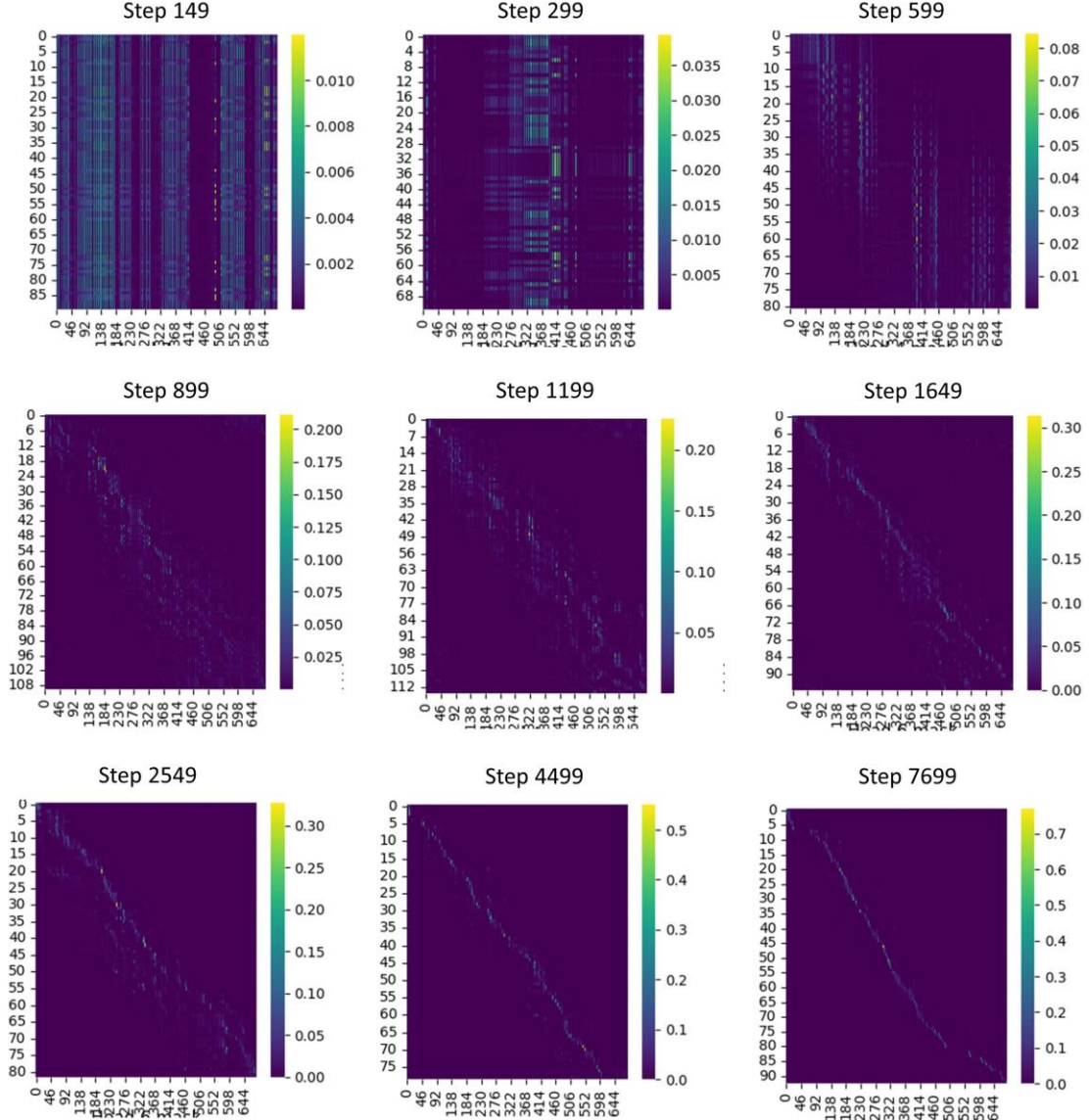
Comment 4: *The authors propose the architecture where the input is 12-fold down-sampled before being passed to the Transformer encoder, and up-sampled at the output of the Transformer. Justification for this approach, along with the details of the up-sampling module, would be beneficial for understanding the architecture.*

Response: Many thanks for pointing this out. In the encoder architecture of Coral, a 12-fold down-sampling is firstly applied to the raw signal to efficiently extract local contextual features and reduce the sequence length, thereby lowering the computational complexity of the subsequent self-attention operations within the Transformer encoder. Since CTC and Transformer decoders require an output sequence from the encoder that is longer than the target ribonucleotide sequence to ensure a stable and accurate signal-to-sequence alignment, a 12-fold down-sampling risks shortening the sequence excessively. To mitigate this, an up-sampling module is incorporated after the Transformer encoder. This module effectively doubles the sequence length, resulting in an overall 6-fold down-sampling relative to the raw signal, which preserves the necessary temporal resolution for robust decoding. Specifically, the up-sampling is achieved through a linear projection. The feature sequence from the encoder $h \in R^{T \times d}$ (T is sequence length and d is feature dimension), is transformed using a learnable matrix $U \in R^{d \times 2d}$ to produce an expanded sequence $h' = hU \in R^{T \times 2d}$. This output is then reshaped and interpreted as a sequence of length 2T with the same feature dimension d, denoted as $h' \in R^{2T \times d}$. This architectural detail and its justification have been added to the “**Methods**” section of our revised manuscript for greater clarity.

Comment 5: *The proposed regularization loss FRE aims to penalize non-monotonic alignments by comparing the consecutive expected alignment positions β . Can the authors comment on the cases where values obtained for β seem to be monotonic but are actually averaged values of distant and unrelated positions in the signal?*

Response: Thank you for this insightful question regarding the robustness of the expected alignment position calculation. The concern that β could be skewed by averaging over distant, unrelated signal positions is valid in theory. However, in practice, the nature of the attention mechanism during training mitigates this issue. The expected alignment position β_i for a symbol y_i is calculated as a weighted average over all signal positions: $\beta_i = \sum_{j=1}^T \alpha_c(i, j) * j$, where the attention weight $\alpha_c(i, j)$ represents the learned probability that symbol y_i aligns to the signal at position j . While this is an average across all positions, the model quickly learns to assign meaningful weights only to relevant parts of the signal. As shown in **Response Figure 2**, as training progresses, the cross-attention matrix α_c of final cross-attention layer (closest to the output logits) becomes sharply focused and nearly diagonal. The probabilities for distant and unrelated signal positions become vanishingly small (close to zero). Consequently, their contribution to the weighted average is negligible, and the calculated value of β accurately reflects the true, localized alignment position.

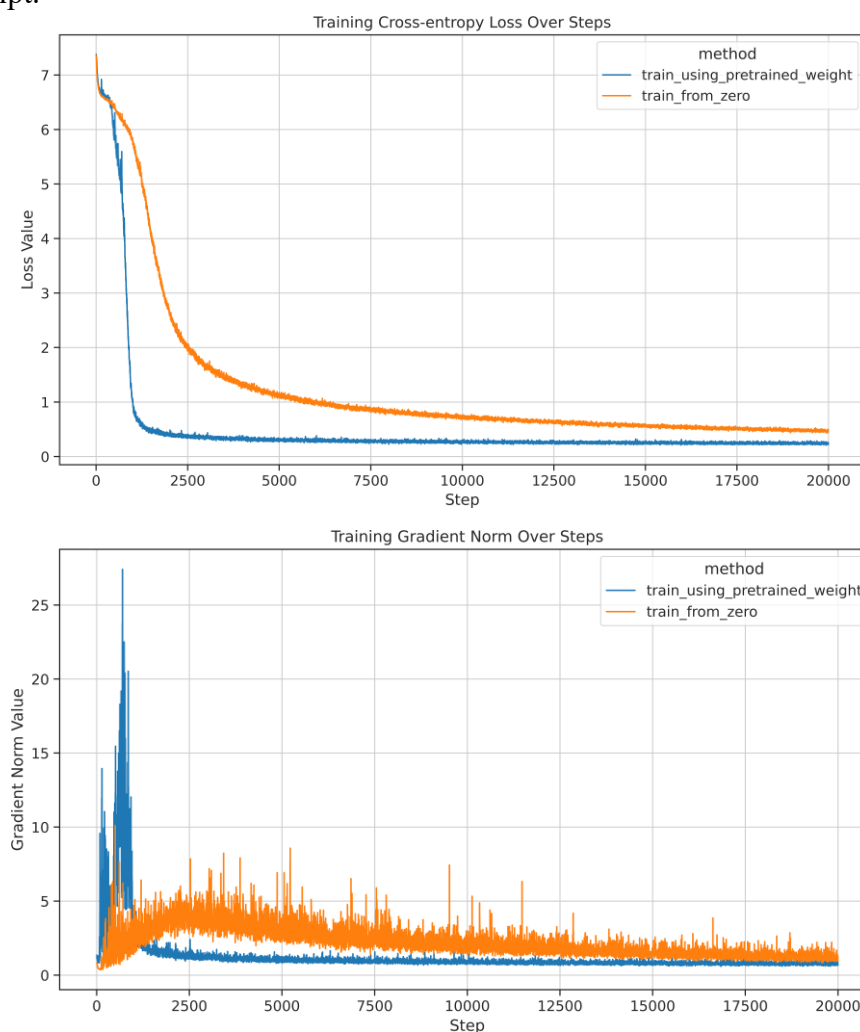
Therefore, the regularization loss $\mathcal{F}_{RE}$ effectively further penalizes non-monotonicity without being confounded by spurious alignments. A more detailed explanation of the computation of the expected alignment position has been added to the “**Methods**” section and the supplementary **Fig. S21**.



Response Figure 2. Visualization of the cross-attention alignment matrix α_c of final cross-attention layer at different training steps. The matrix shows the alignment probability between the ribonucleotide sequence (y-axis) and signal representations (x-axis). As training progresses from Step 149 to Step 7699, the attention becomes increasingly focused and diagonally aligned.

Comment 6: *It would be beneficial if the authors could comment and support the design choice for training the Coral-FAST model using part of the weights trained with the original Coral approach. Have you tried training from zero? How do the pretrained weights help the training, given the objective shift?*

Response: Many thanks for the suggestion. The decision was primarily driven by the need to ensure training stability. The Coral-FAST model has a significantly larger output vocabulary than the standard Coral model because it predicts k-mers (e.g., 5-mers or 6-mers) instead of single bases. This expanded vocabulary makes the model more challenging to train from scratch. Our experiments confirmed this: when training from a random initialization, the gradient of the loss function exhibited significant fluctuations. In contrast, initializing Coral-FAST with weights from a pre-trained Coral model provided a much more stable and effective starting point for optimization, leading to a smoother gradient norm and faster convergence of the loss value (**Response Figure 3**). The reason for training of Coral-Fast with pretrained model weights has been added to the “**Methods**” section and the supplementary **Fig. S22** of the revised manuscript.



Response Figure 3. Comparison of training cross-entropy loss (top) and gradient norm (below) for Coral-FAST with and without pre-trained weight initialization over training steps.

Comment 7: *It would be important to mention that experimental validation of new isoforms and fusion genes is necessary.*

Response: Thank you very much for the valuable suggestion, and we wholeheartedly agree. As our study was based entirely on publicly available data, we did not have access to the original biological material required for wet-lab assays. To rigorously validate our findings, we therefore performed two complementary in-silico and orthogonal analyses. We validated Coral-specific isoforms using matched Illumina short-read RNA-seq data from the same samples, confirming that our novel discoveries are well-supported by this independent sequencing modality. We also validated our fusion gene detection using findings from a recent publication that experimentally confirmed 12 specific fusion genes within the same MCF-7 sample with RT-PCR and Sanger sequencing. Future work will be dedicated to collaborating with experimental partners to perform targeted validation of high-confidence transcripts and apply Coral to new datasets where biological samples are available, creating a direct pipeline from computational discovery to experimental verification. We have incorporated this into the “**Discussion**” section of the revised manuscript to provide a discussion of the current study's limitation.

Reviewer #4

Comment 1: *The installation and running instructions on Github are well written. The conda installation and running the model were seamless. The output was well formatted. I recommend integrating support for running the model on multiple GPUs. Given the complexity and amount of data that we usually have for basecalling, this would make the model more practically applicable.*

Response: Thank you very much for the recommendation. We have updated the code of Coral and now the new version supports multiple GPU inference.

Response of Manuscript NCOMMS-24-71768B

We sincerely thank the reviewers for the thorough evaluation of our manuscript and for providing insightful and constructive comments. Your feedback has been invaluable in helping us improve the clarity and rigor of the paper. We have carefully addressed each point raised, revised the manuscript accordingly, and believe that the changes have significantly strengthened the work.

Reviewer #3

Comment 1: *It would be beneficial if the authors would include the performance of Guppy/Dorado on the same subset of reads for the leave-one-out generalization evaluation for comparison. We understand that these models are not retrained for generalization evaluation, but this would provide the information on the exact comparison.*

Response: Thank you very much for the valuable suggestion. In this revision, we have now included the basecalling performance of Guppy and Dorado in the leave-one-out cross-validation (LOOCV) experiments, providing a direct baseline for comparison (**Response Tables 1 and 2**).

As shown in **Response Table 1**, across the six leave-one-species-out runs on the RNA001/002 datasets, the LOOCV-trained Coral model consistently shows superiority to all other basecallers, including Guppy and Dorado. In the leave-one-chromosome-out experiment of the RNA004 dataset (**Response Table 2**), Coral achieved a median accuracy of 97.78%, which is better than the other retrainable models RODAN (92.44%) and GCRTcall (94.15%), and remains highly comparable to the non-LOOCV-trained Dorado model (98.13%). Since the RNA004 dataset is derived from a universal human reference, the official Dorado model is supposed to be optimally trained for this type of data. The results demonstrate that Coral could effectively learn a robust mapping between raw signal and sequence rather than merely memorizing reference-specific patterns, highlighting its strong generalization capabilities. The corresponding results and discussions have been incorporated into the revised manuscript and supplementary materials (Section “**Basecalling accuracy**”, **Table S4 and S5**).

Response Table 1. Basecalling performance of leave-one-species-out cross-validation on RNA001/002 test datasets.

Dataset (Read Count)	Basecallers	Read Accuracy (%)		Mismatch(%)		Insertion(%)		Deletion(%)		Read Length (bp)		Unaligned Reads
		mean	median	mean	median	mean	median	mean	median	mean	median	
Human (100000)	Guppy*	89.88	90.6	2.56	2.32	2.21	2	5.35	4.91	964	713	N/A
	Dorado*	90.77	91.32	2.76	2.55	2.17	2	4.3	3.93	900	665	1316
	RODAN	90.45	91.11	2.32	2.09	1.84	1.63	5.39	4.85	847	681	4063
	GCRTcall	91.68	92.69	1.8	1.56	1.87	1.42	4.65	4.07	973	722	1445
	Coral	<u>94.8</u>	<u>95.55</u>	<u>1.5</u>	<u>1.28</u>	<u>1.23</u>	<u>1.01</u>	<u>2.46</u>	<u>2</u>	<u>987</u>	<u>730</u>	<u>117</u>
Mouse (100000)	Guppy*	87.17	87.66	3.88	3.68	2.66	2.52	6.29	5.96	647	524	N/A
	Dorado*	88.84	89.47	3.59	3.35	1.96	1.8	5.61	5.21	639	519	1700
	RODAN	87.78	88.12	3.19	3.05	1.62	1.44	7.4	7.11	498	450	4813
	GCRTcall	89.54	90.19	2.69	2.51	1.67	1.49	6.09	5.6	643	522	911
	Coral	<u>92.69</u>	<u>93.46</u>	<u>2.45</u>	<u>2.23</u>	<u>1.26</u>	<u>1.09</u>	<u>3.6</u>	<u>3.03</u>	<u>652</u>	<u>529</u>	<u>154</u>
Yeast (100000)	Guppy*	90.51	91.36	2.78	2.38	2.51	2.32	4.2	3.8	842	597	N/A
	Dorado*	93.21	94.42	2.21	1.69	1.8	1.54	2.78	2.25	842	598	848
	RODAN	91.15	91.9	2.51	2.21	2.08	1.89	4.25	3.81	652	553	3465
	GCRTcall	91.94	92.97	2.22	1.87	1.98	1.77	3.86	3.24	847	601	1071
	Coral	<u>94.65</u>	<u>95.83</u>	<u>1.91</u>	<u>1.46</u>	<u>1.36</u>	<u>1.05</u>	<u>2.08</u>	<u>1.57</u>	<u>853</u>	<u>604</u>	<u>273</u>
Arabidopsis (100000)	Guppy*	90.73	91.59	2.32	2.03	2.18	1.93	4.77	4.32	964	855	N/A
	Dorado*	93.36	94.36	1.86	1.51	1.69	1.39	3.09	2.63	966	854	509
	RODAN	91.6	92.29	2.01	1.81	1.85	1.59	4.54	4.13	723	667	1907
	GCRTcall	92.17	93.24	1.74	1.48	1.91	1.5	4.17	3.59	969	859	565
	Coral	<u>95.21</u>	<u>96.15</u>	<u>1.38</u>	<u>1.09</u>	<u>1.02</u>	<u>0.74</u>	<u>2.4</u>	<u>1.9</u>	<u>979</u>	<u>870</u>	<u>65</u>
Poplar (100000)	Guppy*	89.26	90.16	2.95	2.64	2.87	2.51	4.91	4.47	792	665	N/A
	Dorado*	92.51	93.66	2.26	1.86	2.17	1.74	3.07	2.57	795	668	477
	RODAN	90.87	91.64	2.49	2.24	2.25	1.95	4.38	4	617	559	3025
	GCRTcall	90.85	91.94	2.44	2.11	2.41	2.01	4.3	3.72	797	669	1168
	Coral	<u>94.48</u>	<u>95.58</u>	<u>1.75</u>	<u>1.41</u>	<u>1.56</u>	<u>1.1</u>	<u>2.2</u>	<u>1.73</u>	<u>805</u>	<u>676</u>	<u>70</u>
Zebrafish (100000)	Guppy*	86.45	86.9	3.86	3.61	2.18	1.94	7.51	7.23	950	823	N/A
	Dorado*	92.81	93.66	2.31	2	1.93	1.63	2.94	2.52	1011	880	506
	RODAN	88.7	89.27	3.16	2.95	1.75	1.59	6.4	5.92	759	696	2798
	GCRTcall	88.93	89.6	2.96	2.72	1.8	1.54	6.31	5.83	977	848	1537
	Coral	<u>93.7</u>	<u>94.55</u>	<u>2.25</u>	<u>1.95</u>	<u>1.52</u>	<u>1.2</u>	<u>2.53</u>	<u>2.07</u>	<u>1008</u>	<u>875</u>	<u>348</u>

* Since there are no training pipelines available for Guppy and Dorado, the corresponding original pretrained official models were used. LOOCV was not applicable to these two models. There is no guarantee that data identical to the test set, or data from the same or closely related species, was unseen during the training process.

The best results among all methods are marked as bold text.

The best LOOCV results (excluding Guppy and Dorado) are underlined.

“N/A” on unaligned reads for Guppy in the RNA001/002 test datasets, which include only the reads successfully basecalled by Guppy and aligned to the reference sequence.

Response Table 2. Basecalling performance of leave-one-chromosome-out evaluation on the chr22 reads of HEK293T RNA004 sample.

Dataset (Read Count)	Basecallers	Read Accuracy (%)		Mismatch(%)		Insertion(%)		Deletion(%)		Read Length (bp)		Unaligned Reads
		mean	median	mean	median	mean	median	mean	median	mean	median	
HEK293T RNA004 chr22 reads (233936)	Guppy*	86.43	87.02	3.81	3.68	2.57	2.37	7.20	6.75	1130	966	26408
	Dorado*	96.77	98.13	0.76	0.55	0.98	0.36	1.49	0.79	1122	898	1747
	RODAN	91.31	92.44	1.94	1.72	1.32	1.09	5.44	4.55	1074	875	8327
	GCRTcall	92.86	94.15	1.48	1.27	1.32	0.93	4.34	3.49	1094	<u>880</u>	3545
	Coral	<u>96.54</u>	<u>97.78</u>	<u>0.93</u>	<u>0.70</u>	<u>0.70</u>	<u>0.36</u>	<u>1.83</u>	<u>1.03</u>	<u>1099</u>	869	<u>1750</u>

* Since there are no training pipelines available for Guppy and Dorado, the corresponding original pretrained official models were used. LOOCV was not applicable to these two models. There is no guarantee that data identical to the test set, or data from the same or closely related species, was unseen during the training process.

The best results among all methods are marked as bold text.

The best LOOCV results (excluding Guppy and Dorado) are underlined.

Comment 2: *Response Table 3 and Figure 1 show the relation between the model size and performance for RNA002 chemistry, where we see that the full Coral model is ~20 times larger in size, and only Coral-S can compare in size with the comparable performance. On the contrary, we see no such comparison for RNA004, where the size of the Dorado model is far more comparable to that of Coral. We believe those comparisons would be beneficial.*

Response: Thank you for the precious suggestion. We have now performed the ablation study examining performance relative to model size on the HEK293T RNA004 dataset. The architectures, parameter sizes, and performance of the ablated models are provided in **Response Tables 3 and 4**. The results were also visualized in **Response Figure 1**.

To have a fair comparison with Dorado-RNA004 in terms of model size, we configured a new Coral variant namely Coral-ML to have a parameter size of 308MB, which is very close to that of Dorado-RNA004 model (300MB). Coral-ML achieved a median accuracy of 98.71% and outperformed Dorado-RNA004 (98.25%) at the same parameter scale. The new results together with the previous RNA001/002 results confirm the advantages of the Coral architecture and the performance of the model scales up with the model size (from Coral-S up to Coral).

The new results have been provided and discussed in the revised manuscript and supplementary materials (Section “**Effectiveness of modeling contextual dependencies**”, **Table S11**, **Table S12**, and **Fig S11**).

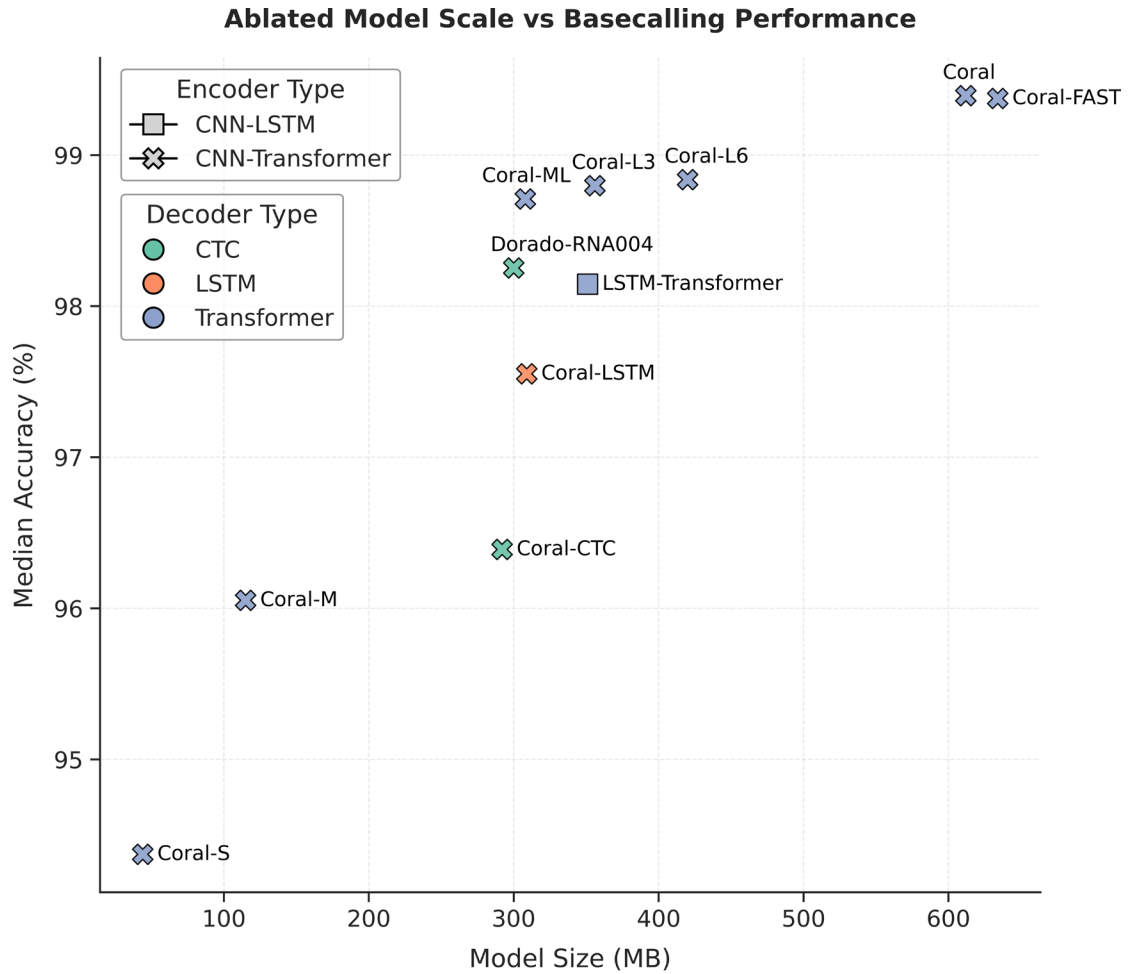
Response Table 3. Architecture and parameter size of ablated models used in RNA004 test dataset.

Ablated model	Encoder type	Decoder type	Encoder layers	Decoder layers	Feature dimension	Param size(MB)
LSTM-Transformer	CNN-LSTM	Transformer	5	12	512	351
Coral-CTC	CNN-Transformer	CTC	18	1	512	292
Coral-LSTM	CNN-Transformer	LSTM	18	1	512	309
Coral-S	CNN-Transformer	Transformer	3	2	256	44
Coral-M	CNN-Transformer	Transformer	1	6	384	115
Coral-ML	CNN-Transformer	Transformer	15	3	512	308
Coral-L3	CNN-Transformer	Transformer	18	3	512	356
Coral-L6	CNN-Transformer	Transformer	18	6	512	420
Coral (i.e., Coral-L15)	CNN-Transformer	Transformer	18	15	512	612
Coral-FAST	CNN-Transformer	Transformer	18	15	512	634

Response Table 4. Basecalling performance of Coral ablated models on sampled reads from HEK293T RNA004 dataset.

Dataset (Read Count)	Model	Read Accuracy (%)		Mismatch (%)		Insertion (%)		Deletion (%)		Read Length (bp)		Unaligned Reads
		mean	median	mean	median	mean	median	mean	median	mean	median	
HEK293T RNA004 sampled reads (623781)	Coral-CTC	95.32	96.39	0.83	0.63	0.61	0.41	3.25	2.32	1025	759	176111
	Coral-LSTM	96.28	97.55	0.90	0.64	0.81	0.50	2.01	1.17	1044	772	171153
	LSTM-Transformer	96.05	98.15	0.74	0.38	1.02	0.33	2.18	0.74	1015	735	173998
	Coral-S	93.38	94.37	1.57	1.28	1.65	1.23	3.40	2.66	1026	754	181249
	Coral-M	94.44	96.05	1.47	1.02	1.26	0.79	2.84	1.77	1030	756	178516
	Coral-ML	97.33	98.71	0.63	0.31	0.46	0.13	1.58	0.61	1045	775	171036
	Coral-L3	97.35	98.80	0.65	0.33	0.60	0.19	1.41	0.49	1050	777	170695
	Coral-L6	97.45	98.84	0.59	0.29	0.50	0.14	1.46	0.48	1046	775	170323
	Coral-FAST	97.75	99.37	0.46	0.15	0.38	0.00	1.41	0.21	1045	774	170233
	Coral	97.74	99.39	0.39	0.06	0.35	0.00	1.52	0.33	1047	776	169982

The best results are marked as bold text.



Response Figure 1. Basecalling performance of Coral ablated models with different size on HEK293T RNA004 dataset. ‘Coral-S’ represents a small-scale variant of the Coral model. ‘Coral-M’ represents a medium-scale variant of the Coral model. ‘Coral-ML’ represents a medium-large-scale variant of the Coral model. ‘Coral-L3’, ‘Coral-L6’ and ‘Coral’ represent large-scale variants of the Coral model with 3/6/15 decoder layers. ‘Coral-CTC’ and ‘Coral-LSTM’ are two Coral decoder variants that replace the Transformer decoder with ‘CTC’ and ‘LSTM’. ‘LSTM-Transformer’ is Coral encoder variant that replace the CNN-Transformer encoder with CNN-LSTM encoder. The basecalling performance is measured using median accuracy on 623k sampled reads from HEK293T RNA004 data.

Comment 3: *The authors motivate the down-then up-sampling approach by the need to preserve sufficient temporal resolution for robust decoding, with the up-sampling done along the embedding dimension rather than the temporal dimension. It should be discussed why a smaller scale down-sampling would not be a good choice compared with this approach, and how exactly this up-sampling approach helps preserve sufficient temporal resolution.*

Response: We thank the reviewer for this insightful question. The encoder architecture of Coral is intentionally designed to balance the Transformer's computational cost with the decoder's requirement for high temporal resolution. To validate this choice, we have trained an ablated Coral-L3 model by removing the up-sampling step and directly applying 6-fold down-sampling followed by Transformer layers in the encoder. The ablated Coral-L3 was compared to the standard Coral-L3 model on the RNA004 dataset. As shown in **Response Table 5**, the standard Coral-L3 attained the same median accuracy while requiring substantially fewer computational resources in terms of GFLOPs and GPU memory usage. The results indicate that a 12-fold down-sampling is sufficient for Transformer-based contextual modeling and is considerably more efficient.

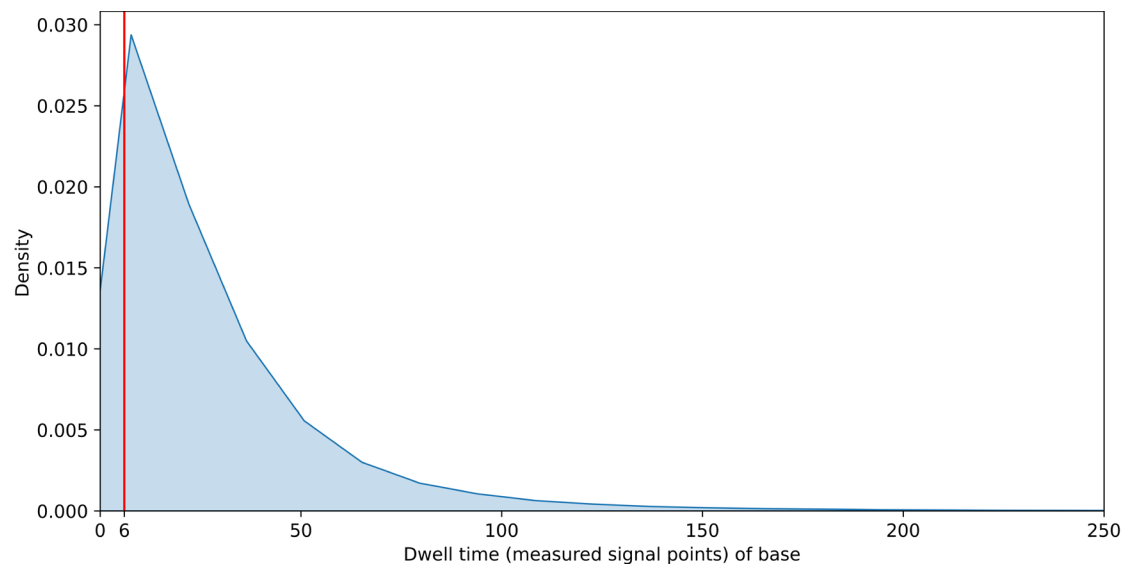
The up-sampling applied after the Transformer encoder restores an effective 6-fold down-sampling resolution. The design ensures that, during training, the signal feature sequence remains longer than the target nucleotide sequence—a necessary condition for stable alignment in both CTC and Transformer decoder. In addition, the 6-fold resolution provides finer granularity for decoding. As shown in **Response Figure 2**, this resolution adequately covers the dwell time of most signal events.

All corresponding results and detailed explanations have been provided in the revised manuscript and supplementary materials (Section “**Methods**”, **Table S13**, and **Fig S22**).

Response Table 5. Basecalling performance of Coral-L3 and Coral-L3 without up-sampling in the encoder on sampled reads from HEK293T RNA004 data.

Dataset (Read Count)	Model (Aligned reads)	Read Accuracy (%)		Mismatch(%)		Insertion(%)		Deletion(%)		Read Length (bp)		GFLOPS	GPU Memory Usage (MB)
		mean	median	mean	median	mean	median	mean	median	mean	median		
HEK293T RNA004 sampled reads (830885)	Coral-L3 (596880)	97.40	98.81	0.64	0.32	0.58	0.18	1.37	0.47	1000	726	131	1526
	Coral-L3 (w/o up-sampling) (595795)	97.21	98.81	0.60	0.24	0.50	0.11	1.69	0.58	993	722	524	2780

The best results are marked as bold text. GFLOPS (Giga Floating-point Operations Per Second) denotes the computational cost of the matrix-multiplication operations in the Transformer encoder layers under a batch size of 64 and a feature dimension of 512. GPU memory usage is reported for the operations performed within the Transformer encoder.



Response Figure 2. Distribution of dwell times (measured signal points per base) calculated from one million bases in the HEK293T RNA004 dataset. Dwell times were obtained by aligning the ribonucleotide sequence to the raw electrical signal using the ont-remora tool.