nature portfolio

Peer Review File

Empowering Low-Crosstalk, Dynamic-Decision Random Access of DNA Storage via 384-Multiplexed Nanopore Signatures

Corresponding Author: Professor Yi Li

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)
This paper describes a technique for designing barcodes whose current signatures are distinguishable during Oxford Nanopore-based sequencing. The authors aim to use these barcodes for selective sequencing of DNA strands, by identifying barcodes in real-time and prematurely halting the sequencing of strands without the barcode(s) of interest. The contributions in this paper are twofold: 1) a computational workflow for designing the barcode sequences ("SUSTag") and 2) a deep learning framework ("ORCtrL") to identify them based on raw Oxford Nanopore sequencing data (i.e., the current fluctuations observed). The authors designed a set of 96 barcodes, which they compared experimentally against two existing sets of 96 barcodes. They also designed a set of 384 barcodes that they used to demonstrate selective sequencing from a set of DNA strands encoding text and image data.

To design their SUSTag barcodes, the authors used standard tools to simulate the current signatures of short (8- or 9-nt) oligos, applied a distance metric based on Bhattacharyya distance, and used clustering to filter out a subset of the most distinct oligos. These oligos were then concatenated, analyzed, and filtered again twice more to generate a set of either 96 or 384 oligos, each 34-nt in length. The deep learning tool ORCtrL used to identify barcodes during sequencing applied a convolutional neural net (CNN) followed by a long short-term memory (LSTM) network. The output of the LSTM network is subsequently fed into three components: a selector network to determine if the signal corresponds to a valid barcode, a classifier network to identify the barcode, and an auxiliary classifier (role unclear).

Overall, the barcode design and identification systems appear to be novel and interesting, and to improve upon existing designs. However, I had difficulty interpreting and evaluating their methods and several of their claims. For example, the authors mention several times the importance of transfer learning, but (unless I misunderstood their claim) I could not find a comparison against a control without its use. In addition, they appear to have used some type of precision-recall score to provide an overall measure of how well the barcodes are identified (Figure 2e), but (to my knowledge) there should be separate precision/recall scores for every class (i.e., barcode) in a multiclass classification system. Additional (not exhaustive) examples are included in the comments below.

Significant revision could improve the clarity of this work, and its potential future impact.

General comments:

1) I did not find the description of the sequence design algorithm in the methods section sufficient. For example, the authors stated that clustering was used to select the top N barcode designs at each iteration, without stating how clusters were used to do this. In addition, the role of the auxiliary module was unclear to me.

2) Several claims are made without clear quantitative support. For example, in Figure 1, the authors claim that "off-diagonal values are suppressed", and visually this appears to be the case. However, this should be supported by some quantitative metric. A clearly stated metric would also clarify some specific claims. For example, on page 7, the authors state "ONT96 has more cross-talk for distanced sequences, while Porcupine 96 holds more cross-talks for neighbor ones", but it is unclear in either case what the additional cross-talk is relative to.

3) The authors claim that the transfer learning was important for the performance of the barcode classifier, but I could not find a comparison to a model without pre-training. In addition, I found the use of the term "transfer learning" questionable because it involved only fine-tuning the network parameters to the additional background present in the DNA data storage context. My understanding is that transfer learning typically refers to improving performance of a network using another network trained on a related but distinct task.

4) It was not clear to me if the barcode identification testing was performed in real-time during sequencing, which would be necessary to perform selective sequencing. If not, this should be clearly stated, and the authors should make some statements regarding the speed of the barcode classifier and whether this is compatible with the time constraints needed for selective sequencing.

5) I believe the paper would benefit from significant editing for grammar and readability. There were several key phrases that were difficult to understand. For example, on page 5,
a. "leveraging the noise properties as important as the squiggled levels…"
b. "our BD map calculation of all sequence candidates is expected to improve the robustness to outliers and biases, thus contributing to a confusion matrix…"
c. "Linclust is adopted as the incremental design strategy with linearly time starting with shorter sequences, since brutal-forcedly computing BD for all 4^34 candidate sequences is unaffordable"

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
Summary
The authors propose computational methods related to the use of DNA sequence-based tags for DNA technologies that use Nanopore sequencing. The first is a method for the design of a library of tag sequences with low cross-talk. The second is an neural network architecture for recognition of the tags in signal-space. An experiment is run in which tags designed with the algorithm are compared to two existing sets of tags, with all three tags concentenated in sequence prior to a dummy payload. The sequenced data is classified with three different neural network architectures to provide and compared to ground truth sequences assigned with a high accuracy basecaller. A second experiment is then run, in which a four-times larger set of tags is designed and used to investigate the ability to fine tune the model initially trained to recognize the 96-member set of tags for classification of the 384-member set. Finally, on-demand random access of a section of a file from the tagged pool is attempted.

Comments
- The paper is in general written for an audience highly familiar with the specific space of DNA sequenced-based tags for Nanopore applications. For a general purpose journal, it would be helpful to include more context around the performance targets necessary to achieve the applications described in the introduction, a discussion of why existing methods fall short and how they have been addressed, etc. I would also like to see the discussion section spend a little more time on the utility enabled by the developed methods and what additional advances are required to enable use in the applications from the introduction.

- The two developed methods read as relatively small adaptations of existing methods. This isn't necessarily disqualifying - if the small changes push performance over a critical threshold for applicability then that is a great contribution - but per the comment above I am unable to judge whether this is the case or not.

- Classification accuracy is used as a key metric throughout the manuscript. It is generally not clear what basis (per strand, per 96/384-mer, per timestep in signal space, something else?) it is being calculated on.

- If the classification accuracy is calculated on a per strand basis, are the Porcupine 96 tags unduly disadvantaged in the comparative experiment due to their position between the other two tags? That is, their precission may be suppressed due to cross-talk with the preceeding tag.

- The introduction speaks to the utilization of these tags in real-time applications. Given this, it would be useful to present the relative inference times for the various models alongside their classification performance.

- The introduction states: "A systematic benchmark of SUSTag demonstrated impressive classification performance with an F1-score of 99.69%, expanding the classification from 96 to 384 classes." The Discussion section has the F1-score of SUSTag with the 384 class library at 99.05%. Seems like one of these numbers is incorrect?

- The description of the SUSTag algorithm in the Results section is very difficult to follow. The description in the methods section is much better.

- The description of the transfer learning experiments are also difficult to follow. Here, I find the description in the methods

section inadequate to replicate the results as well.

- Many times, comparisons are made between models with and without transfer learning. This can be interpretted as base model versus fine-tuned or as trained from scratch versus fine-tuned. I believe the former is the intended interpretation, but it would be good to clarify the language to remove ambiguity.

- I do not understand the random access experiments as described in the results, and they do not have a methods section to fall back to for details.

(Remarks on code availability)
I did not review the code for consistency with the methods in the manuscript, as the description in the manuscript is in many places inadequate.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The authors have addressed my comments in detail so I'm happy to endorse publication.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
Thanks for the extensive updates! The manuscript looks good to me from here.

(Remarks on code availability)
I took a ~15 min look over the code. It primarily covers the "ORCtrL" portion of the paper, in that there were not directions for recreating the SUSTag set or for creating a new set with the same methodology. The top-level README is pretty well done for the ORCtrl part. I did not try to train/use a new model, but the ML portion looks pretty standard. I generally think that someone interested in using the ORCtrl model in their work could do so from this repo.

Empowering Low-Crosstalk, Dynamic-Decision Random Access of DNA Storage via 384-Multiplexed Nanopore Signatures

Response to reviewers

Manuscript ID NCOMMS-25-12743-T

Reviewer's comments in *Italic*

Author's response in Blue

Revised Manuscripts snippets in Red

Reviewer #1 (Remarks to the Author):

This paper describes a technique for designing barcodes whose current signatures are distinguishable during Oxford Nanopore-based sequencing. The authors aim to use these barcodes for selective sequencing of DNA strands, by identifying barcodes in real-time and prematurely halting the sequencing of strands without the barcode(s) of interest. The contributions in this paper are twofold: 1) a computational workflow for designing the barcode sequences ("SUSTag") and 2) a deep learning framework ("ORCtrL") to identify them based on raw Oxford Nanopore sequencing data (i.e., the current fluctuations observed). The authors designed a set of 96 barcodes, which they compared experimentally against two existing sets of 96 barcodes. They also designed a set of 384 barcodes that they used to demonstrate selective sequencing from a set of DNA strands encoding text and image data.

To design their SUSTag barcodes, the authors used standard tools to simulate the current signatures of short (8- or 9-nt) oligos, applied a distance metric based on Bhattacharyya distance, and used clustering to filter out a subset of the most distinct

oligos. These oligos were then concatenated, analyzed, and filtered again twice more to generate a set of either 96 or 384 oligos, each 34-nt in length. The deep learning tool ORCtrL used to identify barcodes during sequencing applied a convolutional neural net (CNN) followed by a long short-term memory (LSTM) network. The output of the LSTM network is subsequently fed into three components: a selector network to determine if the signal corresponds to a valid barcode, a classifier network to identify the barcode, and an auxiliary classifier (role unclear).

Overall, the barcode design and identification systems appear to be novel and interesting, and to improve upon existing designs. However, I had difficulty interpreting and evaluating their methods and several of their claims. For example, the authors mention several times the importance of transfer learning, but (unless I misunderstood their claim) I could not find a comparison against a control without its use. In addition, they appear to have used some type of precision-recall score to provide an overall measure of how well the barcodes are identified (Figure 2e), but (to my knowledge) there should be separate precision/recall scores for every class (i.e., barcode) in a multiclass classification system. Additional (not exhaustive) examples are included in the comments below.

Significant revision could improve the clarity of this work, and its potential future impact.

We thank the reviewer for the positive recognition for the novelty of our work. We have thoroughly revised the manuscript to address all concerns to improve the quality of this work.

General comments:

1) I did not find the description of the sequence design algorithm in the methods section sufficient. For example, the authors stated that clustering was used to select the top N barcode designs at each iteration, without stating how clusters were used to do this. In addition, the role of the auxiliary module was unclear to me.

Reply (1-1):

We sincerely appreciate the constructive feedback regarding the sequence design algorithm, which has significantly enhanced the clarity of our work.

To address the barcode selection process, we have included a detailed, step-by-step algorithmic description in the Methods section. Firstly, the core idea is to use the Linclust-based incremental clustering algorithm [1] to group sequence candidates based on the pre-computed Bhattacharyya distances between their simulated nanopore signatures. Secondly, to select the top N designs, we iteratively adjust the dissimilarity threshold until the number of resulting clusters is just above N. Lastly, we select one representative sequence from each cluster, followed by choosing the N representatives that are most distant from one another to form our final candidate set.

The auxiliary module is active during the training phase but is excluded from the testing phase. This is inspired by Geifman and El-Yaniv's SelectiveNet [2], where two output modules (prediction and auxiliary) are trained to compete with each other to vote for consistent classifications (Fig. R1), which is expected to play a specific role for the training part and consequently contributes to enhance the model performance.

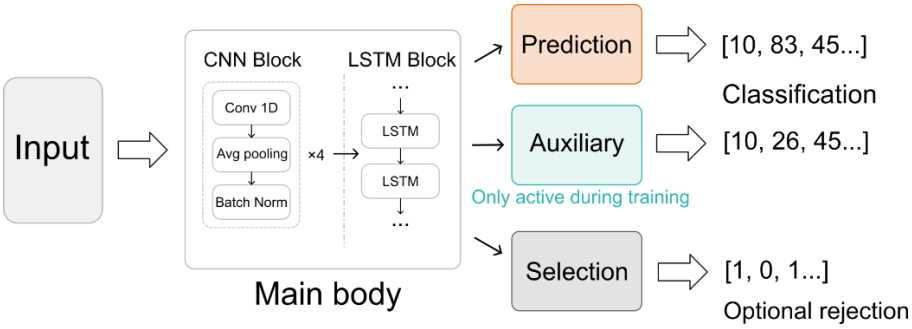

Fig. R1 | The model consists of three output modules during training procedure.

To evaluate the significance of the auxiliary module, we conducted an ablation study on the ORCtrL model. The results are summarized in Table R1 and Figure R2. The inclusion of the auxiliary module during training improves the model's performance, particularly after 20 epochs. Thus, we believe that this module helps to mitigate overfitting by preventing the model from becoming overly specialized to any subset of the training data.

Table R1 | Ablation study of the auxiliary module of ORCtrL model.

ORCtrL (**without** auxiliary module)			ORCtrL (**with** auxiliary module)		
Epochs	Weighted precision	Weighted recall	Epochs	Weighted precision	Weighted recall
10	**0.9214**	**0.9080**	10	0.9215	0.8979
40	0.9794	0.9785	**40**	**0.9833**	**0.9824**
70	0.9843	0.9838	**70**	**0.9898**	**0.9896**
100	0.9882	0.9878	**100**	**0.9917**	**0.9913**

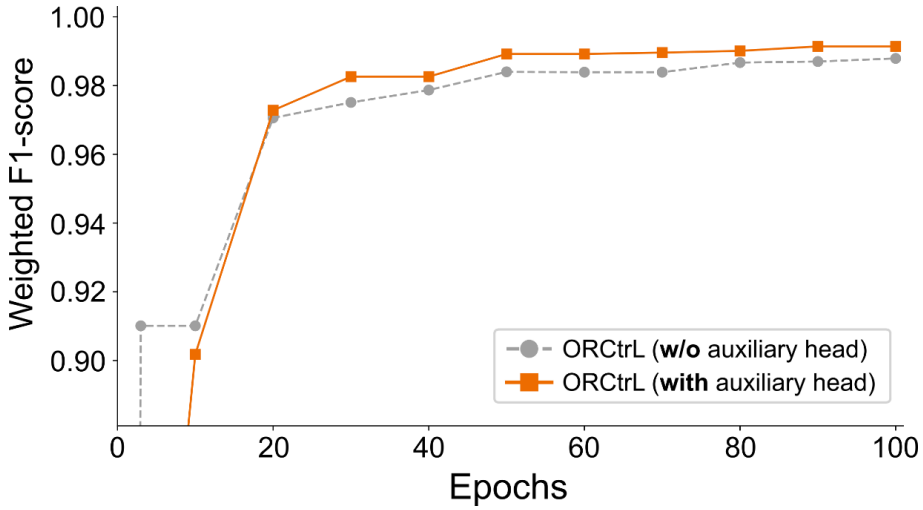

Fig. R2 | Weighted F1-score changes with the growth of training epochs for the ablation study of the auxiliary head.

References:

[1] Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. Nat. Commun. 9, 2542 (2018).

[2] Geifman, Y., & El-Yaniv, R. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. Proceedings of the 36th International Conference on Machine Learning (ICML). 97, 2151–2159 (2019).

Action (1-1):

In the Methods section:

"…Third, an incremental clustering algorithm[40] was used to distill top 400 sequences for both 8-mers and 9-mers as representative for the second round. The selection process involved adjusting the inter-cluster distance threshold, which defines the dissimilarity required to form a new cluster based on pre-computed Bhattacharyya distances. Since the number of clusters is inversely related to this threshold, we iteratively decreased its value until the total number of resulting clusters first met or exceeded our target (N=400). At this stage, a representative sequence was chosen from each of the ≥N clusters. To obtain the final set, we then selected the N representative sequences from this group that were most distant from one another, ensuring the final set has the lowest possible crosstalk potential."

"The auxiliary classifier, another fully connected layer, aids in training by outputting predictions of classes similar to those of the classifier. This optional-reject architecture is inspired by SelectiveNet[44], which enables a model to abstain from prediction on low-confidence inputs. The auxiliary module plays a role in this framework as it ensures the main feature-extracting backbone learns robust features from all training examples, not just those selected for the final prediction. We present ~~a~~ diagrams of this model and the comparative CNN and CNN-LSTM models ~~the model~~, including ~~its~~ their precise parameters, in Supplemental Table ~~S7~~ S6-8."

In the References section:

"44. Geifman, Y., & El-Yaniv, R. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. Proceedings of the 36th International Conference on Machine Learning (ICML). 97, 2151 – 2159 (2019)."

2) Several claims are made without clear quantitative support. For example, in Figure 1, the authors claim that "off-diagonal values are suppressed", and visually this appears to be the case. However, this should be supported by some quantitative metric. A clearly stated metric would also clarify some specific claims. For example, on page 7, the authors state "ONT96 has more cross-talk for distanced sequences, while Porcupine 96 holds more cross-talks for neighbor ones", but it is unclear in either case what the additional cross-talk is relative to.

Reply (1-2):

We thank the reviewer for the insightful feedback, which has prompted us to provide stronger quantitative support for our claims.

For the **"off-diagonal values are suppressed"**, we have used the equation 1 (E1) to calculate the suppression rate:

$$Suppression \quad rate = 1 - \frac{\sum_{1 \le i < j \le N} I(M_{ij} < \tau)}{N(N-1)/2} \qquad (E1)$$

where M is the pairwise distance matrix for an N-plex tag set, and $I(\cdot)$ is the indicator function. The threshold τ is determined based on the color-bar value used in Figure 1c and 1d, specifically 4.05 for Euclidean distance matrix and 50.00 for Bhattacharyya distance matrix.

Table R2 | Comparison of DNA tags' suppression rates.

Tag design	Distance method	Suppression rate
SUSTag 96	Euclidean	**99.87%**
	Bhattacharyya	**99.01%**
Porcupine 96	Euclidean	99.50%
	Bhattacharyya	97.39%
ONT 96	Euclidean	49.21%
	Bhattacharyya	71.14%

The results in Table R2 clearly support our claim, showing that our SUSTag 96 achieves a suppression rate of over 99% with both distance methods, a significant improvement over the 49% and 71% rate of the ONT 96.

For ***"ONT96 has more cross-talk for distanced sequences, while Porcupine 96 holds more cross-talks for neighbor ones"***, we have calculated two metrics, Near-diagonal and Far-from-diagonal crosstalk ratio, as follow:

$$R_{near} = \frac{\sum_{1\le|i-j|\le5} C_{ij}}{\sum_{i\neq j} C_{ij}} \qquad and \qquad R_{far} = \frac{\sum_{|i-j|>5} C_{ij}}{\sum_{i\neq j} C_{ij}} \qquad (E2)$$

where C is derived from the $N\times N$ normalized confusion matrix (N=96), and each element C_{ij} represents the proportion of samples belonging to the ground truth class i that were predicted as class j. Here we set $1 \le |i-j| \le 5$ as the Near-diagonal sequences.

Table R3 | Prediction crosstalk ratios for ONT96, Porcupine 96 and SUSTag 96.

Tag design	Near-diagonal crosstalk ratio	Far-from-diagonal crosstalk ratio
ONT 96	15.11%	84.89%
SUSTag 96	15.71%	84.29%
Porcupine 96	**55.77%**	**44.23%**
Even crosstalk	10.20%	89.80%

As quantified in Table R3, Porcupine 96 shows a significantly higher Near-diagonal crosstalk ratio (55.77%). In contrast, ONT 96 and our SUSTag 96 are substantially closer to an even crosstalk distribution, suggesting their misclassification errors are more randomly spread.

Action (1-2):

We have added the above-mentioned quantitative analysis results in Supplementary Information Tables S1, and S10 and added the description to the manuscript.

In the Results section:

"The off-diagonal values for both Porcupine 96 and SUSTag 96 in Fig. 1c are suppressed compared with ONT 96 (Quantitative percentile data are provided in the Table S1)."

"…yielding F1-scores of 0.9925, 0.9891 and 0.9969 in the 1-96 classification. ~~ONT 96 has more cross-talk for distanced sequences, while Porcuopine 96 holds more cross-talks for neighbor ones.~~ Compared with SUStag 96 and ONT 96, Porcupine 96 presented higher near-diagonal crosstalk, as quantitatively detailed in Supplementary Table S10."

3) The authors claim that the transfer learning was important for the performance of the barcode classifier, but I could not find a comparison to a model without pre-training. In addition, I found the use of the term "transfer learning" questionable because it involved only fine-tuning the network parameters to the additional background present in the DNA data storage context. My understanding is that transfer learning typically refers to improving performance of a network using another network trained on a related but distinct task.

Reply (1-3):

We greatly appreciate the reviewer's thoughtful inquiries on our methodology and associated claims.

For the "***comparison to a model without pre-training***", we now train the ORCtrL model from scratch on the DNA storage dataset. The results together with and the pre-trained + fine-tuned ORCtrL model ones are shown as below (Table R4, R5 and Figure R3):

Table R4 | Performance of ORCtrL model trained from scratch on DNA storage dataset.

Training data amount	Test precision	Test recall	Test F1-score
100,000	0.8984	0.7848	0.8293
300,000	0.9311	0.7973	0.8536
500,000	0.9561	0.8887	0.9197

Table R5 | Performance of ORCtrL model trained on SUSTag 384 dataset and fine-tuned by DNA storage dataset.

Training data amount	Test precision	Test recall	Test F1-score
10,000	0.9353	0.9193	0.9266
100,000	**0.9571**	**0.9118**	**0.9330**
200,000	**0.9591**	**0.9243**	**0.9405**

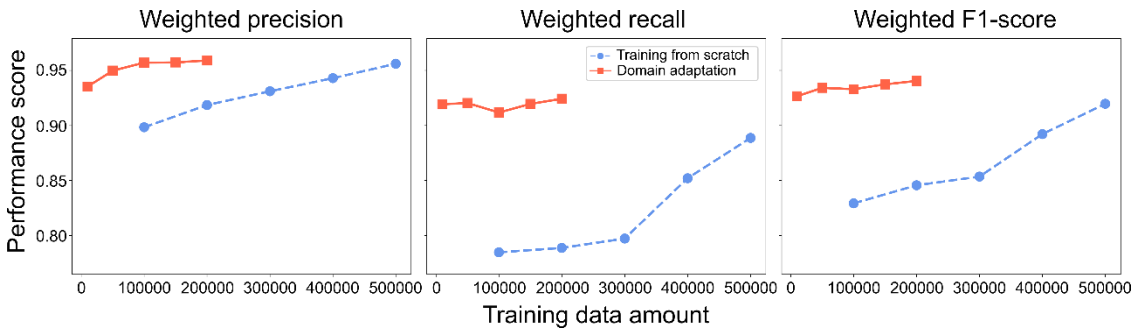

Fig. R3 | Performance of ORCtrL model trained with domain adaptation (Red) vs. trained from scratch (Blue).

Training DNA storage data from scratch proved to be inefficient. As shown in Table R5, fine-tuning with just 100,000 reads from the target domain surpasses the performance of a model trained from scratch on 500,000 reads (Table R4). The better performance can be achieved within one fifth of the data amount. Therefore, the domain adaption becomes compulsory for new DNA libraries, how to manage this with reasonable resources was a major bottleneck for practical applications. We believe our high-quality barcode design and wisely use of the datasets could be one of the key advances of our work, as it drastically minimizes the experimental cost and time.

For "***the definition of transfer learning***", we do agree with the reviewer's point that three terminologies ("transfer learning", "domain adaptation" and "fine-tuning") were mixed used in our manuscript:

1) Transfer learning is the overall strategy of applying knowledge gained from a source task/domain to a target task/domain.

2) Domain adaptation is a specific and common type of transfer learning where the task remains the same (in our study, DNA tag classification), although the model must adapt to a shift in the input data's distribution between the source domain and the target domain.

3) Fine-tuning is the practical method we employ to perform domain adaptation.

We believe "domain adaptation" is a decent description of our core challenge in this work.

Therefore, we confine ourselves to using domain adaptation' solely. We have revised and doubly checked the entire manuscript to improve clarity.

Action (1-3):

We have revised the Abstract, Introduction, Results, and Discussion sections, as well as the relevant figure captions.

In the Abstract section:

"…an optional-reject CNN-LSTM deep learning model, ~~and a~~ inspired by transfer-learning ~~pipeline~~ (ORCtrL)…"

"…~~Transfer learning with~~ Domain adaptation using only 120 minutes of new sequencing data (~200k reads) boosts the ~~pre-trained model's F1-score~~ model performance from 87% to over 94%, ..."

In the Introduction section:

"…Secondly, we introduce ORCtrL: an optional-reject module ~~and a~~ inspired by transfer learning ~~framework~~ to maximize adaptability to the complex and unstable nature of nanopore sequencing signals within domain adaptation…"

"Furthermore, by using SUSTag as address bits for DNA storage and adapting the ORCtrL to the new data domain, ~~combined with transfer learning techniques,~~ we enabled PCR-free random access…"

In the Results section:

"~~Transfer learning~~ Domain adaptation enables SUSTag 384 adapted for DNA storage datasets"

"We further analyze the detailed ~~transfer learning~~ performance of domain adaptation between precision and recall, as shown in Fig. 3e."

"With quantified performance of ~~transfer learning~~ model adaptation steps, we demonstrate the random-access capability of our ORCtrL model on #1-#51 over all 384

sequences. Figure 3f and 3g show the performance without and with ~~transfer learning~~ domain adaptation as the targeting regions."

In the Discussion section:

"This framework combines innovations in tag design, deep learning classification, and ~~transfer learning~~ domain adaptation to enable a higher rate of information gain…"

"We address optional rejection and compulsory ~~transfer learning~~ domain adaptation as the key for enabling our SUSTag-ORCtrL."

4) It was not clear to me if the barcode identification testing was performed in real-time during sequencing, which would be necessary to perform selective sequencing. If not, this should be clearly stated, and the authors should make some statements regarding the speed of the barcode classifier and whether this is compatible with the time constraints needed for selective sequencing.

Reply (1-4):

We thank the reviewer for raising this crucial question regarding the real-time implementation of our system.

We confirm that our system is compatible with the time constraints needed for real-time selective sequencing. To address this, we have conducted a real-time adaptive sampling demo experiment on a MinION device.

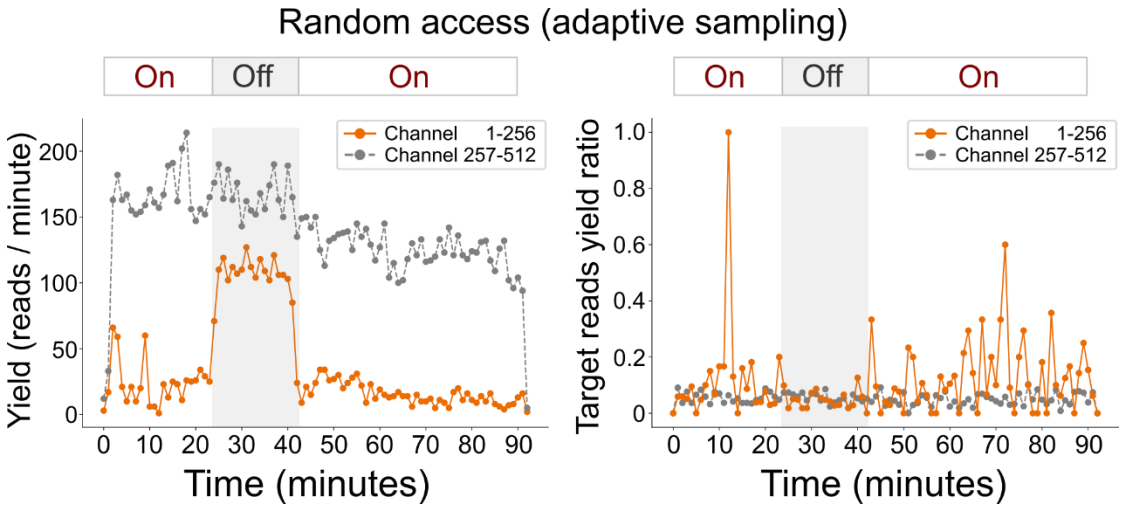

Fig. R4 | Comparison of cumulative read yields over time between adaptive sampling channels and control channels.

We have connected our system to a MinION sequencer via the MinKNOW API, with the results presented in the Fig. R4. During the experiment, we ran adaptive sampling exclusively on an experimental group (channels 1-256, orange), while a blank control group (channels 257-512, grey) sequenced without intervention. When adaptive sampling was active ('On'), the experimental group exhibited significantly suppressed read throughput (left panel) and a corresponding enrichment of on-target reads from a ~12% baseline to over 30% (right panel). Moreover, to provide a direct intra-group

comparison, the random-access function was temporarily disabled ('Off'). During this period, both metrics from the experimental group reverted to the levels of the control, providing clear supports of the system's real-time ability and functionality.

Table R6 | Average inference time analysis for different classification methods.

Batch size	ORCtrL	Porcupine CNN	Readfish
64	6 ms	5 ms	197 ms
128	11 ms	9 ms	394 ms
256	20 ms	18 ms	771 ms
512	39 ms	41 ms	1536 ms

The feasibility of this real-time performance is further supported by our model's high inference speed. As shown in our analysis (Table R6), ORCtrL requires only 6 ms to process a batch of 64 reads, making it substantially faster than the basecalling-and-mapping pipeline used by Readfish (197 ms) and meeting the latency requirements for effective real-time decision-making.

While the above demo experiment confirmed our system's real-time functionality, the DNA storage random access analysis in Figure 4 was conducted as an offline simulation. We chose this simulated approach because the DNA storage library's extremely short fragments (~253 nt) can challenge the physical hardware's response time in a live run. The simulation allowed us to create an ideal, zero-latency environment to fairly benchmark the performance of the ORCtrL model against other methods, separating software performance from hardware limitations.

Action (1-4):

We have added Fig. R4 as Fig. 4g and revised the Results section accordingly.

In the Result section:

"~~Figure 4 shows on-demand random access on the same dataset.~~ To evaluate the potential of our system for on-demand data retrieval, we performed a simulated random-access experiment on the DNA storage dataset. Using adaptive nanopore sequencing, "t was officially approved by the Ministry of Education in April 2012. SUSTech bears the responsibility for exploring and developing a modern university system with Chinese characteristics to serve as a model for cultivating innovative talents. SUSTech" indexed from #29 to #42 as the target sequences for further evaluation…"

"In our simulation, the inference time for the ORCtrL pipeline was merely 6 ms for a batch of 64 reads, outperforming the Readfish pipeline (197 ms) and confirming it meets the time constraints for real-time adaptive sampling (Supplementary Table S13). To further validate this, we conducted a live 'Read Until' demonstration on a MinION sequencer (Fig. 4g). We applied adaptive sampling to a subset of channels (1-256) simultaneously suppressed read throughput and enriched on-target sequences from a ~12% baseline to over 30%, relative to control channels (257-512). Moreover, temporarily disabling the random-access function caused reads yielding to revert to control levels. This provides direct evidence of the SUSTag-ORCtrL system's dynamic control over the sequencing process and confirms its feasibility for live selective sequencing."

5) I believe the paper would benefit from significant editing for grammar and readability. There were several key phrases that were difficult to understand. For example, on page 5,

a. "leveraging the noise properties as important as the squiggled levels…"

b. "our BD map calculation of all sequence candidates is expected to improve the robustness to outliers and biases, thus contributing to a confusion matrix…"

c. "Linclust is adopted as the incremental design strategy with linearly time starting with shorter sequences, since brutal-forcedly computing BD for all 4^34 candidate sequences is unaffordable"

Reply (1-5):

We extend our gratitude to the reviewer for their meticulous examination. We have undertaken a comprehensive revision through the manuscript with a focus on improving sentence structure, clarity of explanations, and overall readability.

Action (1-5):

We have made changes to the corresponding sentences in the manuscript.

In the Result section:

"Figure 1b illustrates SUSTag, our 34-nt nanopore-based DNA tag, design framework. The framework integrates dynamic time warping (DTW)[19,38] with Bhattacharyya distance (BD)[39] and an incremental clustering algorithm based on Linclust[40]. The features are as follows: ~~Figure 1b shows our SUSTag – a nanopore-based DNA tag design framework – with a length of 34 nt that employ dynamic time warping (DTW)[38,19] with Bhattacharyya distance[39] (BD) and Linclust-based incremental clustering algorithm[40].~~"

"…1) ~~Leveraging the noise properties as important as the squiggled levels, o~~Our BD-DTW, which treats noise properties (signal variance) as being as informative as the mean signal levels (squiggles), is used to estimate the simulated inter-signal distances

between ~~every two sequence candidates~~ all pairs of sequence candidates…”

“…2) Unlike Euclidean distance (ED), the BD ~~our BD map calculation of all sequence candidates~~ is expected to improve the robustness to outliers and biases[41] within the nanopore signals. The resulting BD map, which contains all pairwise inter-tag distances, effectively forms a distance matrix. This matrix is depicted in a confusion-style format (Fig. 1b, middle panel) to visually represent the predicted distinguishability between tags. ~~thus contributing to a confusion matrix, depicted in the middle panel~~…”

“...3) ~~Linclust is adopted as the incremental design strategy with linearly time starting with shorter sequences, since brutal-forcedly computing BD for all 4^34 candidate sequences is unaffordable.~~ A brute-force computation of BD for all 434 candidate 34-mer sequences is computationally prohibitive. Therefore, we have adopted an incremental design strategy. This strategy evaluates shorter sequences and utilizes Linclust, an algorithm that scales linearly with the number of sequences, prior to extending to the full 34-nt tag length…”

Besides, we have revised the Abstract, Results, and Methods sections, as well as the relevant figure captions to enhance the quality of our manuscript.

In the Abstract section:

“…However, ~~most of~~ contemporary methods for targeted retrieval using PCR amplification or bead-based extraction, ~~which~~ rely on Watson-Crick base pairing and pre-defined primers, ~~lack~~ limiting dynamic decision-making ~~capabilities~~ whilst sequencing. We introduce SUSTag-ORCtrL ~~that uses~~, a nanopore-based system enables real-time, PCR-free random access to DNA-stored data by directly classifying raw ionic current signatures of ~~DNA tags for demultiplexing into~~ 96 or 384 ~~classes.~~plex DNA molecular tags. Our ~~system, involving~~ framework combines SUSTag, a Bhattacharyya distance and incremental clustering enhanced molecular tag design (SUSTag~~),~~) to minimize crosstalk, with an optional-reject CNN-LSTM deep learning model~~, and a~~ inspired by transfer-learning ~~pipeline~~ (ORCtrL), ~~minimizes crosstalk for random access.~~designed to enhance adaptability to signal variability. SUSTag-ORCtrL achieves

an intra-class weighted F1-score of 99.69% for 96-plex classification and 99.05% for 384-plex classification, surpassing existing molecular ~~tags. Transfer learning with 120~~tagging systems. Domain adaptation using only120 minutes of new sequencing data (~200k reads) boosts the ~~pre-trained model's F1-score~~model performance from 87% to over 94%, ~~enabling significant crosstalk cancellation and~~ achieving complete recovery of target data ~~recovery for selective DNA data access~~ with minimal crosstalk in 10 minutes to 3 hours. ~~Our nanopore-oriented~~ This system provides ~~real-time,PCR-free randoma~~ scalable, low-latency solution for versatile DNA data access and ~~has potential~~ holds promise for genomic and transcriptomic disease screening and the DNA-of-things."

In the Introduction section:

"A systematic benchmark of SUSTag demonstrated impressive classification performance with a F1-score (weighted averages; Methods) of 99.69%..."

In the Methods section:

We have added a subsection called "Evaluation Metrics" to explain how to calculate the overall performance score.

"Evaluation metrics included the overall accuracy on different test datasets, recall and precision for binary classification of whether a sequence is a barcode, and the F1-score for intra-barcode category classification. For multiclass classification tasks, the reported F1-score, precision, and recall are the weighted averages of the per-class scores, providing a balanced overall assessment of model performance across all classes. In this paper, all references to precision, recall, and F1-score denote their weighted-average counterparts, unless explicitly stated otherwise."

Also, we have changed the axis labels of Fig. 2e and Fig. 3c-3e, and the annotations of the gray data points in Figure 3e, as follows:

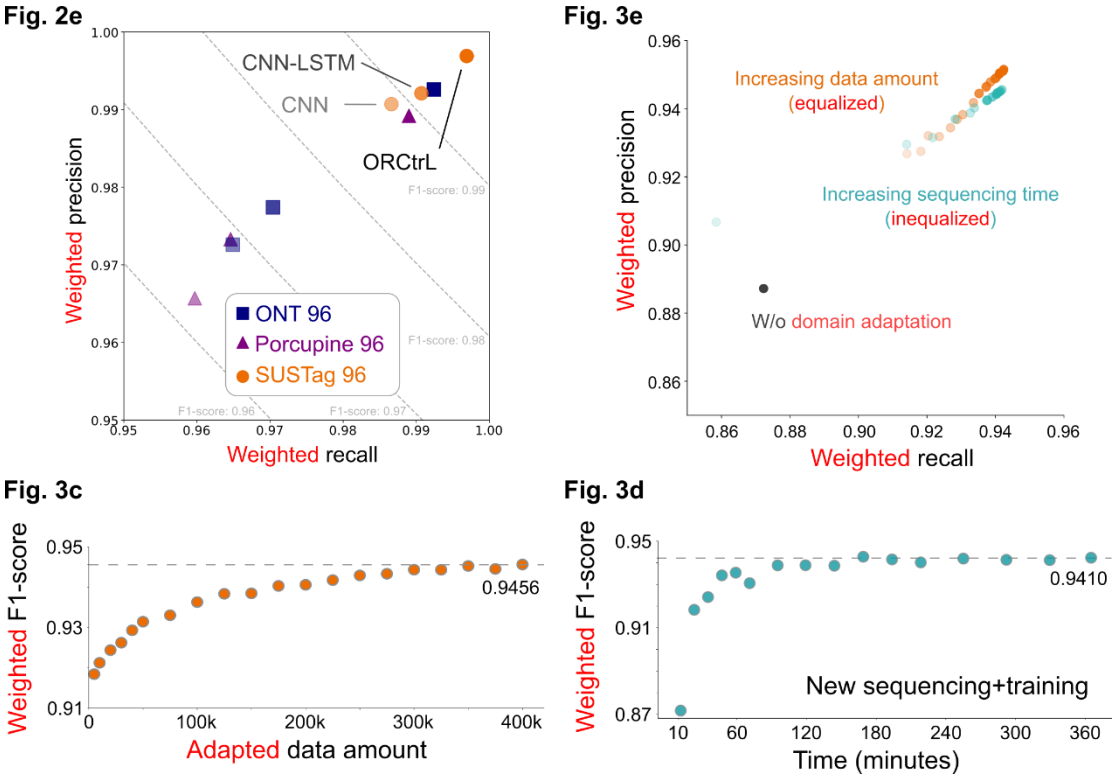

Fig. R5 | Revised Fig. 2e and Fig. 3c-3e.

Reviewer #2 (Remarks to the Author):

Summary

The authors propose computational methods related to the use of DNA sequence-based tags for DNA technologies that use Nanopore sequencing. The first is a method for the design of a library of tag sequences with low cross-talk. The second is an neural network architecture for recognition of the tags in signal-space. An experiment is run in which tags designed with the algorithm are compared to two existing sets of tags, with all three tags concentenated in sequence prior to a dummy payload. The sequenced data is classified with three different neural network architectures to provide and compared to ground truth sequences assigned with a high accuracy basecaller. A second experiment is then run, in which a four-times larger set of tags is designed and used to investigate the ability to fine tune the model initially trained to recognize the 96-member set of tags for classification of the 384-member set. Finally, on-demand random access of a section of a file from the tagged pool is attempted.

Comments

- The paper is in general written for an audience highly familiar with the specific space of DNA sequenced-based tags for Nanopore applications. For a general purpose journal, it would be helpful to include more context around the performance targets necessary to achieve the applications described in the introduction, a discussion of why existing methods fall short and how they have been addressed, etc. I would also like to see the discussion section spend a little more time on the utility enabled by the developed methods and what additional advances are required to enable use in the applications from the introduction.

Reply (2-1):

We sincerely appreciate the reviewer's valuable suggestions, which have vastly enhanced the manuscript's accessibility and broader scientific impact.

In the revised Introduction (Page 3), we have incorporated a detailed discussion of the performance benchmarks typically required to achieve the described applications.

We also have expanded the Discussion section in Page 18-19, as suggested, to further elaborate on the practical utility and advantages enabled by SUSTag-ORCtrL. We have also discussed what advances are needed to move the system proposed in Page 20 to practical applications.

Action (2-1):

We have revised the Introduction and Discussion sections.

In the Introduction section:

"…With ongoing advancements in stability and accuracy of nanopore sequencing[23], adaptive sampling is attracting growing interest in its potential for random access to the information stored in DNA[24]. Realizing these potential demands exceptional performance, including the ability to classify a DNA strand and make a keep-or-reject decision in hundreds of milliseconds (due to the sequencing speed of ~420 bp/s) with an accuracy high enough (e.g., F1-score > 0.95) to ensure data retrieval purity."

"…However, the complexity of raw nanopore signal signatures presents both challenges and opportunities for achieving effective adaptive sampling. A primary challenge for DNA tags is signal crosstalk, where the current signatures of numerous different barcodes are too similar to be reliably distinguished, resulting in unsatisfactory enrichment purity and efficiency~~, especially when dealing with the specific requirements of DNA tags.~~ Recent approaches work…"

In the Discussion section:

"…thereby satisfying the low-crosstalk demand of random access in DNA storage. This robustness conferred by domain adaptation is particularly critical in a real-time setting, where experimental conditions such as pore health and background noise can drift over a multi-hour run. An adaptable model is essential for maintaining stable and reliable

target selection throughout the experiment. ~~Transfer learning allows the model to adapt to new current signatures, effectively bridging the gap between the source and target domains.~~"

"…explore the full potential of this molecular tagging system. ~~Moreover, exploring more open-set recognition methods could identify universally applicable classification models and transfer learning strategies.~~ While we have now demonstrated the feasibility of random access, engineering developments are still required for widespread adoption. These include further minimizing the software latency between data streaming and classification, optimizing the model architecture (e.g., through quantization or pruning) for faster inference, and scaling the real-time pipeline to handle the data deluge from high-throughput sequencers like PromethION. Furthermore, the ultimate test will be applying this system to complex native biological samples, where performance must be maintained against a noisy and variable background. Overcoming these challenges will be key to unlocking the full potential of dynamic nanopore sequencing for the wide range of applications."

- The two developed methods read as relatively small adaptations of existing methods. This isn't necessarily disqualifying - if the small changes push performance over a critical threshold for applicability then that is a great contribution - but per the comment above I am unable to judge whether this is the case or not.

Reply (2-2):

We appreciate the reviewer's insightful perspective. To quantitatively justify our integrated SUSTag-ORCtrL system, we have performed a comprehensive ablation study.

This study demonstrates the value of each component by starting with a baseline CNN and sequentially adding the LSTM block, the optional rejection module, and finally, domain adaptation. We evaluated each configuration on the SUSTag 384 mixed barcode dataset, and the results are summarized below:

Table R7 | Ablation study of ORCtrL model components.

Model Configuration	Weighted F1-Score	Misidentified reads	Reads recovery
CNN	79.49%	19,693	78.22%
CNN-LSTM	90.99%	8,940	89.99%
CNN-LSTM-Optional Rejection	93.70%	4,109	67.53%
ORCtrL (adapted by 200,000 reads)	**95.87%**	**3,383**	**90.66%**

Each model component contributes to a substantial improvement in performance. The LSTM block halves the number of misidentified reads and boosts the F1-score by over 11%. The optional rejection module reduces the number of misidentified reads by half and further elevates the F1-score to 93.70%. This enhancement in precision comes at

the expected cost of a lower read recovery rate, confirming the module's effectiveness at filtering ambiguous signals. The final model, with domain adaptation, achieves the highest performance.

To connect these architectural improvements to the "***critical threshold for applicability,***" we evaluated these models on the task of targeted decoding the DNA storage data (Fig. R6).

Fig. R6 | Decoding results of random-access data obtained using the ablated models with 1 hour of sequencing data.

While all models could decode the target region, their ability to suppress crosstalk from non-targets varied significantly. The baseline CNN achieved an access purity of only 35.90%, which improved to 77.78% with the LSTM module. Crucially, the addition of the optional rejection module boosted access purity to 93.33%. Only the full ORCtrL pipeline achieved the 100% access purity required for error-free data retrieval.

- Classification accuracy is used as a key metric throughout the manuscript. It is generally not clear what basis (per strand, per 96/384-mer, per timestep in signal space, something else?) it is being calculated on.

Reply (2-3):

Thanks for highlighting the need to explicitly state the basis on which our classification metrics are calculated.

All classification performance metrics reported in the manuscript – including accuracy, F1-score, precision, and recall – are computed on a ***per-read*** (or ***per-strand***) basis.

In addition, we have added the calculation formulas for access purity and target decode ratio involved in Figure 4d, 4e as follows:

$$Access\ purity = \frac{Number\ of\ correctly\ accepted\ target\ reads}{Total\ number\ of\ accepted\ reads} \quad (1)$$

$$Target\ decode\ ratio = \frac{Number\ of\ successfully\ decoded\ target\ sequences}{Total\ number\ of\ target\ sequences} \quad (2)$$

To ensure the clarity, we have revised the manuscript, particularly in the "Methods" section under "Evaluation metrics" that all reported classification metrics were calculated on a per-read basis.

Action (2-3):

In **Action (1-0)**, we added the Evaluation metrics subsection in the Methods section. Here we add a description based on the revision.

"…and the F1-score for intra-barcode category classification. All performance metrics are calculated on a per-read basis, where each full read is treated as a single data point for classification. For multiclass classification tasks…"

Also, we have supplemented the definitions of Access purity and Target decode ratio in the Evaluation metrics:

"For the random-access experiments, we introduced two additional metrics. 1. Access Purity: This metric measures the precision of the selection process. It is calculated as:

$$Access\ purity = \frac{Number\ of\ correctly\ accepted\ target\ reads}{Total\ number\ of\ accepted\ reads}$$

2. Target Decode Ratio: This metric measures the completeness of data retrieval at a given time point. It is calculated as:

$$Target\ decode\ ratio = \frac{Number\ of\ successfully\ decoded\ target\ sequences}{Total\ number\ of\ target\ sequences}$$"

- If the classification accuracy is calculated on a per strand basis, are the Porcupine 96 tags unduly disadvantaged in the comparative experiment due to their position between the other two tags? That is, their precission may be suppressed due to cross-talk with the preceeding tag.

Reply (2-4):

We appreciate the reviewer's concern about potential advantages of our work over the Porcupine 96 tags. To address this, we have implemented a masking strategy during dataset preparation precisely to mitigate such an effect. (Figure R7)

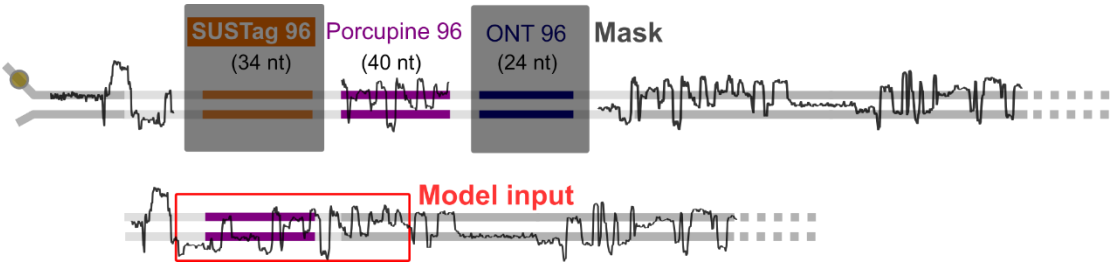

Fig. R7 | Scheme of masking during preprocessing of mixed barcode datasets.

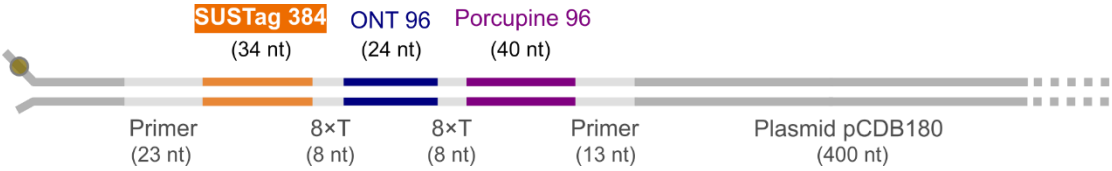

Fig. R8 | Sequence structure containing SUSTag 384, ONT 96 and Porcupine 96.

In the constructs containing SUSTag 384, the Porcupine 96 barcode is in the end position (Figure R8). This design enables a direct comparison of the performance of Porcupine 96 tags across two distinct experimental contexts. We have performed an analysis comparing the classification performance with and without the signal masking procedure mentioned in our Methods (Table R8), which we believe the positional hindrances can be reasonably ruled out.

Table R8 strongly supports that our masking strategy is both necessary and effective. When the masking strategy is applied, the performance of the ORCtrL model is consistently high (~0.99) and, importantly, is independent of the tag's position within

the construct. Conversely, when masking is removed, we observe a noticeable drop in performance, particularly when the tag is in the more challenging middle position.

Table R8 | Effect of positional context and signal masking on Porcupine 96 performance

Positional context	Masking applied?	Precision	Recall
SUSTag 96 | **Porcupine** | ONT	**yes**	**0.9893**	**0.9890**
SUSTag 96 | **Porcupine** | ONT	no	0.9654	0.9633
SUSTag 384 | ONT | **Porcupine**	**yes**	**0.9897**	**0.9895**
SUSTag 384 | ONT | **Porcupine**	no	0.9702	0.9697

Action (2-4):

To complement Figure 2a, we have added Supplementary Figure S5 (same as Fig. R8) to provide another schematic of the sequence structure used for the SUSTag 384 experimental constructs.

In the Results section:

"Figure 2a shows the assembled SUSTag 96, Porcupine 96, and ONT 96 into a single sequence (see Supplementary Fig. S5 for the SUSTag 384 construct) …"

- The introduction speaks to the utilization of these tags in real-time applications. Given this, it would be useful to present the relative inference times for the various models alongside their classification performance.

Reply (2-5):

Many thanks for the suggestion. Here we have profiled the inference speed in Table R7. The speed of the Porcupine CNN is nearly identical to our ORCtrL model, which is expected, as both are deep learning models with significantly less network complexity than a full basecaller. This advantage becomes clear when compared to the basecalling-based Readfish pipeline, which is substantially slower, requiring about 33 times more processing time than our direct-signal approach. Consequently, the highly competitive inference times achieved by SUSTag-ORCtrL underscore its practical utility in resource-constrained real-time applications.

Table R6 | Average inference time analysis for different classification methods.

Batch size	ORCtrL (ms)	Porcupine (ms)	Readfish (ms)
64	6	5	197
128	11	9	394
256	20	18	771
512	39	41	1,536

Action (2-5):

We have added Table R6 in the Supplementary Information as Supplementary Table S13.

"…showcasing neither missing of targeted addresses nor readout of unwanted tags. In our simulation, the inference time for the ORCtrL pipeline was merely 6 ms for a batch of 64 read segments, demonstrating a substantial speed advantage over the Readfish pipeline (197 ms) and confirming it meets the time constraints for real-time operation (Supplementary Table S13) ..."

- The introduction states: "A systematic benchmark of SUSTag demonstrated impressive classification performance with an F1-score of 99.69%, expanding the classification from 96 to 384 classes." The Discussion section has the F1-score of SUSTag with the 384 class library at 99.05%. Seems like one of these numbers is incorrect?

Reply (2-6):

We thank the reviewer for pointing this out. There are two values throughout the work:

1) The F1-score of 99.69% mentioned in the introduction and abstract refers specifically to the performance of our SUSTag 96 using the ORCtrL model, as detailed in the Results section and Figure 2c.
2) The F1-score of 99.05%, reported in Table 2 of the Discussion section, corresponds to the performance of the ORCtrL model applied to the SUSTag 384, which then incorporates domain adaptation as described in the Results section focusing on the DNA storage application.

We have revised the sentence in the Introduction to explicitly distinguish between the two F1-scores, thereby eliminating any potential ambiguity.

Action (2-6):

We have revised the Introduction section.

In the Introduction section:

"…A systematic benchmark of SUSTag demonstrated impressive classification performance with a weighted F1-score of 99.69%~~,~~ for 96-plex classification. Our work further expanded it to 384-plex classification, achieving a weighted F1-score of 99.05% on the mixed barcode dataset. ~~expanding the classification from 96 to 384 classes.~~"

- The description of the SUSTag algorithm in the Results section is very difficult to follow. The description in the methods section is much better.

Reply (2-7):

We thank the reviewer for this feedback. To address this issue, we have heavily reconstructed the description accompanying Figure 1b in the Results section.

To provide a comprehensive overview of the key principles and components underpinning our design framework, we have included a new version of introduction to the use of Bhattacharyya distance with DTW, its advantages over Euclidean distance in this context. The necessity of an incremental, Linclust-based strategy to manage computational complexity is also provided as they would reduce the barrier to understand Figure 1 comprehensively.

Action (2-7):

We have intensively rewritten the description accompanying Figure 1b in the Results section, similar to Action (1-5).

In the Result section:

"Figure 1b illustrates SUSTag, our 34-nt nanopore-based DNA tag, design framework. The framework integrates dynamic time warping (DTW)[19,38] with Bhattacharyya distance (BD)[39] and an incremental clustering algorithm based on Linclust[40]. The features are as follows: ~~Figure 1b shows our SUSTag – a nanopore-based DNA tag design framework – with a length of 34 nt that employ dynamic time warping (DTW)[38,19] with Bhattacharyya distance[39] (BD) and Linclust-based incremental clustering algorithm[40].~~"

"…1) ~~Leveraging the noise properties as important as the squiggled levels, o~~Our BD-DTW, which treats noise properties (signal variance) as being as informative as the mean signal levels (squiggles), is used to estimate the simulated inter-signal distances between ~~every two sequence candidates~~ all pairs of sequence candidates…"

"…2) Unlike Euclidean distance (ED), the BD ~~our BD map calculation of all sequence candidates~~ is expected to improve the robustness to outliers and biases[41] within the nanopore signals. The resulting BD map, which contains all pairwise inter-tag distances, effectively forms a distance matrix. This matrix is depicted in a confusion-style format (Fig. 1b, middle panel) to visually represent the predicted distinguishability between tags. ~~thus contributing to a confusion matrix, depicted in the middle panel~~…"

"...3) ~~Linclust is adopted as the incremental design strategy with linearly time starting with shorter sequences, since brutal-forcedly computing BD for all 4^34 candidate sequences is unaffordable.~~ A brute-force computation of BD for all 434 candidate 34-mer sequences is computationally prohibitive. Therefore, we adopted an incremental design strategy. This strategy begins by evaluating shorter sequences and employs Linclust, an algorithm that scales linearly with the number of sequences, before extending to the full 34-nt tag length…"

- The description of the transfer learning experiments are also difficult to follow. Here, I find the description in the methods section inadequate to replicate the results as well.

Reply (2-8):

We appreciate the reviewer's feedback. To ensure the experimental description is easy to follow and reproduce, we have largely revised the Result section and the Methods session. For the consistency of the terminologies, we will confine ourselves to use 'Domain adaption' for clarity.

The description in the Results section has been expanded to precisely define the model architecture, the datasets and training pipeline that correspond to Figures 3c-3e. We now explicitly define the source domain (mixed barcode dataset) and target domain (DNA storage dataset), detailing the performance drop caused by the domain shift and the subsequent recovery via fine-tuning.

For the concerns on reproducing our results, we have provided a detailed session in the methods under "Data collection and model training." We have added a subsection specifically outlining our domain adaptation protocol, which includes: a) The specific pre-trained model utilized as the starting point, b) The preparation of the DNA storage dataset subsets that were used for the fine-tuning process, c) The fine-tuning strategy itself, covering aspects such as which model layers were updated, the learning rates employed, batch sizes, the number of training epochs for fine-tuning, and any specific stopping criteria or regularization methods used.

Action (2-8):

We have revised the description of transfer learning in Figure 3 in the Results section to clarify that how to achieve domain adaptation:

"~~The model's capacity for precise predictions can be compromised by inconsistent current signal patterns from the non-SUSTag portions of DNA storage sequences. This results in the decrease in F1-score to 87.69% when directly applying a pre-trained~~

~~model to a novel dataset, demonstrating the importance of transfer learning.~~

The ORCtrL model, pre-trained on our mixed barcode dataset (the source domain), showed high classification performance. However, its capacity for precise predictions was compromised when applied directly to the DNA storage dataset (the target domain), where the barcodes are flanked by variable payload sequences. This domain shift resulted in a significant decrease in the F1-score to 87.69%. To bridge this gap, we employed a fine-tuning strategy, a common transfer learning method for domain adaptation. This process involves making small updates to the pre-trained model's parameters using a small subset of data from the target domain, enabling it to learn the new signal characteristics (see Methods for full details).”

We have added a subsection in the Methods section to illustrate the fine-tuning process for domain adaptation:

“Fine-tuning protocol for domain adaptation. To adapt the pre-trained model to the DNA storage dataset, we performed a fine-tuning procedure. The specifics of this protocol are as follows:

Pre-trained Model. The starting point was the ORCtrL model that had been pre-trained on the balanced SUSTag 384 mixed-barcode dataset for 30 epochs, as described above.

Fine-tuning Datasets. Subsets of the DNA storage dataset were used for fine-tuning. For the experiments shown in Fig. 3c, this involved randomly sampling and balancing reads across all 384 classes to create datasets of varying sizes (from 20k to 400k reads). For the experiments in Fig. 3d, all reads collected within a specific sequencing duration (10 to 360 minutes) were used directly without balancing.

Fine-tuning Strategy & Hyperparameters. During fine-tuning, the initial CNN feature extraction blocks of the model were frozen to preserve the learned low-level features, while the LSTM blocks and the final fully connected layers were made trainable. We utilized the Adam optimizer with a reduced learning rate of 1e-4 to make precise adjustments to the model weights. The model was fine-tuned with a batch size of 512

for 10 epochs on each respective dataset subset. No specific early stopping criteria were used for the fine-tuning process.”

- Many times, comparisons are made between models with and without transfer learning. This can be interpreted as base model versus fine-tuned or as trained from scratch versus fine-tuned. I believe the former is the intended interpretation, but it would be good to clarify the language to remove ambiguity.

Reply (2-9):

We thank the reviewer for raising this issue. To ensure the consistency and clarity over the manuscript, we will solely use "domain adaptation" instead of "fine-tuning" or "transfer learning" as it becomes a decent reflection of our core challenge.

Action (2-9):

We have revised the annotations of the subfigures in Figure 3, highlighting the changes in red.

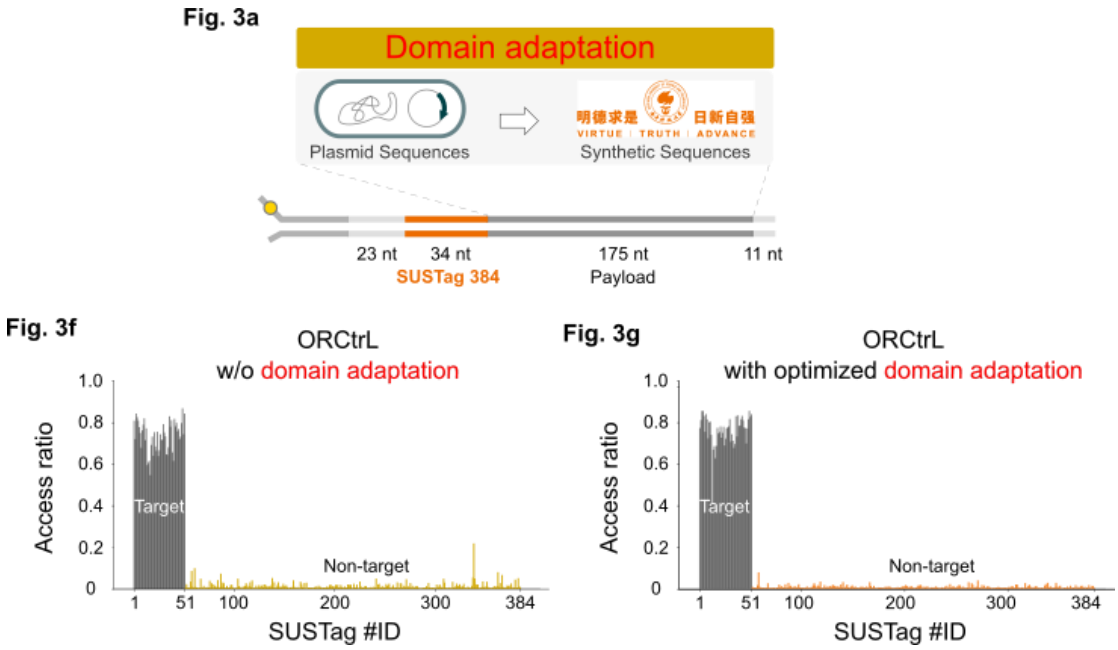

Fig. R9 | Revised Fig. 3a, Fig. 3f and Fig. 3g.

We have replaced all the related instances of "transfer learning" in the manuscript with "domain adaptation" to ensure terminological consistency. The specific changes are in **Reply (1-3)**.

- I do not understand the random access experiments as described in the results, and they do not have a methods section to fall back to for details.

Reply (2-10):

We appreciate the feedback of the missing methods for the random-access experiments.

To reduce the readers' barrier, we have inserted a brief introduction at the beginning of "SUSTag 384 demonstrates rapid and low crosstalk PCR-free random access for DNA storage" subsection of the results. This description outlines the setup of the DNA storage random access experiments, explaining that we aimed to selectively retrieve specific text data (addressed by our SUSTag 384 barcodes) from a larger pool of DNA-encoded data.

Furthermore, we have added a new, detailed subsection to the Methods. This new subsection, titled "DNA storage random access experimental process and evaluation," provides a comprehensive description of how these experiments were conducted and analyzed.

Action (2-10):

We have revised the Results and Methods sections.

In the Results section:

"To evaluate the potential of our system for on-demand data retrieval, we performed a simulated random-access experiment on the DNA storage dataset. ~~Figure 4 shows on-demand random access on the same dataset.~~ Using adaptive nanopore sequencing, "*t was officially approved by the Ministry of Education in April 2012. SUSTech bears the responsibility for exploring and developing a modern university system with Chinese characteristics to serve as a model for cultivating innovative talents. SUSTech*" indexed from #29 to #42 as the target sequences for further evaluation. In this scenario, the SUSTag 384 serve as addresses for distinct blocks of data encoded. The objective was to selectively retrieve and decode a specific subset of text data (addressed by indices #29-#42) from the complete pool of DNA strands, while rejecting all non-target data.

The performance of our fine-tuned ORCtrL model was benchmarked against two other approaches: the basecalling + mapping method, Readfish, and another signal-based method, the Porcupine CNN model. Figure 4a shows the readout examples…"

In the Methods section, we have added a new subsection to provide a description of how these random access experiments were conducted and analyzed:

"**DNA storage random access experimental process.** To validate the random access capabilities of SUSTag-ORCtrL and benchmark it against other methods, we conducted an offline analysis simulating a real-time selective sequencing process.

ORCtrL-based Random Access: The raw signal data (FAST5 files) from the DNA storage sequencing run were processed chronologically. For each read, our fine-tuned ORCtrL model was used to classify its SUSTag barcode signature. A decision was then simulated: if the classified barcode was within the target range (e.g., #29-#42), the read was "accepted"; otherwise, it was "rejected". The collection of "accepted" reads at different time points (e.g., 10, 30, 60 min) was then passed to the CHN decoder to assess if the targeted text data could be successfully recovered.

Readfish and Porcupine CNN Setup: For the Readfish comparison, a reference file containing all 384 barcode sequences was created. The basecalling function of Guppy was used, and the resulting sequences were mapped to the reference using minimap2. Reads mapping to a target barcode were "accepted". For the Porcupine CNN model comparison, the same raw signal data was processed through the Porcupine CNN classifier to identify barcodes, and the same "accept/reject" logic was applied based on its classification output."

Reviewer #2 (Remarks on code availability):

I did not review the code for consistency with the methods in the manuscript, as the description in the manuscript is in many places inadequate.

Reply (2-11):

In cooperation with above manuscript revisions, we have also updated our GitHub repository. All the codes have been rigorously refined for clarity, and we have now included a runnable demo to further facilitate the understanding and verification of our work.

We believe the manuscript has been considerably improved upon the reviewers' insightful comments, and that it is suitable for Nature Communications.