nature communications

Article

https://doi.org/10.1038/s41467-025-64293-2

# Empowering low-crosstalk, dynamic-decision random access of DNA storage via 384-multiplexed nanopore signatures

Received: 17 February 2025

Accepted: 15 September 2025

Published online: 17 October 2025

Check for updates

Junyao Li[1], Xuyang Zhao[1], Qingyuan Fan[1], Yanping Long[2], Ronghui Liu[1], Jixian Zhai[2], Qing Pan[3] & Yi Li[1]

On-demand access to information encoded in nucleotides lies at the heart of DNA/RNA applications. However, contemporary methods for targeted retrieval using PCR amplification or bead-based extraction, rely on Watson-Crick base pairing and pre-defined primers, limiting dynamic decision-making whilst sequencing. We introduce SUSTag-ORCtrL, a nanopore-based system enables real-time, PCR-free random access to DNA-stored data by directly classifying raw ionic current signatures of 96 or 384-plex DNA molecular tags. Our 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 inspired by TRansfer-Learning (ORCtrL), 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 tagging systems. Domain adaptation using only120 minutes of new sequencing data ( ~ 200k reads) boosts the model performance from 87% to over 94%, achieving complete recovery of target data with minimal crosstalk in 10 min to 3 h. This system provides a scalable, low-latency solution for versatile DNA data access and holds promise for genomic and transcriptomic disease screening and the DNA-of-things.

Multiplexed DNA molecular tags enable unique identifications to differentiate species or samples, widely utilized in both biomedical and electrical technologies[1,2]. These DNA tags are of paramount importance. For example, they enable single-nucleus barcoding for multimodal spatial genomics[3] and nanoliter droplets to capture single-cell heterogeneity and gene regulatory linkages[4]. Furthermore, quantitatively detecting blood biomarkers to advance clinical diagnostics beyond the single biomarker model has been demonstrated using DNA-barcoded probes[5]. Beyond these applications, DNA tags emerge as a growing alternative to conventional electronic sensing and computing methods. For instance, Koch et al. have assessed the ecotoxicity of surface and underground tracing of natural waterflows using encapsulated DNA tags[6]. Recognized as fingerprints for the hybridization process, DNA barcodes were proposed[7] and demonstrated[8] for encrypted, content-based DNA similarity search. DNA tags can also act as programmable elements, directing reactions and implementing logic functions[9] and DNA-based programmable gate arrays[10]. Thus, accurate and on-demand manipulation of DNA tags is the highway to success in DNA-of-things[11], content-based similarity search[8], information processing[12] as well as data storage[13].

Targeted DNA capture, also known as target enrichment, has become the key for realizing the full potential of DNA tags[14]. In the era

[1]School of Microelectronics, MOE Engineering Research Center of Integrated Circuits for Next Generation Communications, Southern University of Science and Technology, Shenzhen, China. [2]Department of Biology, School of Life Sciences, Southern University of Science and Technology, Shenzhen, China. [3]College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. e-mail: liy37@sustech.edu.cn

**Table 1 | Reported performance of methods using ONT native barcodes**

| Year | Methods | Multiplex | Reported precision/ accuracy | Reported recall | Reported F1-score |
|---|---|---|---|---|---|
| 2018 | DeepBinner | 12 | 98.5% | 94.3% | N.A. |
| 2022 | Readfish | 96 | N.A. | N.A. | >90% |
| 2024 | SeqTagger | 96 | 99% | 95% | N.A. |
| | This work (ORCtrL) | 96 | 99.26% | 99.24% | 99.25% |

**Table 2 | Reported performance of methods using custom-designed DNA tags**

| Year | Methods & tags | Multiplex | Reported precision/ accuracy | Reported recall | Reported F1-score |
|---|---|---|---|---|---|
| 2020 | DeePlexiCon | 4 | 99.4% | 98.9% | N.A. |
| 2020 | Porcupine | 96 | 96.9% | N.A. | N.A. |
| 2024 | WarpDemuX | 12 | 97.3% | 97.5% | N.A. |
| | This work (ORCtrL +SUSTag) | 384 | 99.09% | 99.02% | 99.05% |

of next-generation sequencing (NGS), researchers typically employed targeted PCR amplification with specific primers or hybridization-based methods to selectively sequence desired regions[15–17]. Nanopore sequencing devices offer a different approach, overcoming the limitations of Watson-Crick base pairing-dependent target enrichment strategies inherent in NGS platforms[18], particularly for sequencing long DNA samples that contain DNA tags. Selective sequencing of DNA molecules can be processed by reversing the voltage to reject unwanted reads during active sequencing, a technique known as adaptive sampling or Read Until[19]. which is ideal when working with multiplexed DNA tags. Moreover, the real-time nature of nanopore sequencing facilitates dynamic decision-making during downstream bioinformatic processing (see, for example, refs. 20–22). With ongoing advancements in stability and accuracy of nanopore sequencing[23], adaptive sampling is gaining growing interest in its potential for random access to the information stored in DNA[24]. Achieving these goals requires 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 sufficient accuracy (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. Recent approaches work either in base space or directly in signal space. Base-space approaches, often considered the gold standard due to their higher accuracy, rely on mapping basecalled sequences to a reference. For instance, Readfish[25] employs minimap2[26] for aligning basecalled and reference sequences, enabling targeted sequencing of gigabase-sized genomes. ReadBouncer[27] utilizes a mapping algorithm based on Interleaved Bloom Filters, achieving comparable performance. However, these methods are inherently limited by the basecaller performance. The two-step process of basecalling followed by mapping hinders rapid response and consumes substantial computational resources. In contrast, signal-space approaches prioritize computational efficiency. UNCALLED[28] and Sigmap[29] segment raw current signals into events and perform alignment within the signal domain, while SquiggleNet[30] employs a deep learning model for direct signal classification. Nevertheless, the shorter sequence signals of DNA tags are not well-suited to these methods primarily designed for genomic sequences. Effective adaptive sampling for DNA tags requires a more integrated, closed-loop design approach that accounts for tag signatures from the outset.

Nanopore-based DNA tag designs are in their infant stage, reported with inconsistent performance and methods. On one hand, deep-learning methods were applied onto commercially available barcodes from ONT with increasing accuracy and/or recall rates. Wick et al. demultiplexed 12 ONT native barcodes by DeepBinner[31], which employs a convolutional neural network (CNN) architecture, achieving 98.5% precision and 94.3% recall. The demultiplexing of 96 native ONT barcodes were further evaluated using the Readfish pipeline by Payne et al.[32], demonstrating an F1-score greater than 0.9 (Details of reported performance are summarized in Table 1). On the other hand, custom-designed tags have gained increasing attention. Novoa group showed that DeePlexiCon demultiplexed four 20 bp barcodes for direct RNA nanopore sequencing (DRS) with up to 99.4% accuracy and 98.9% recall[33]. WarpDemuX designed 12-plex 18 bp DNA barcodes for DRS, utilizing support vector machines (SVM) for identification, and reported 97% accuracy, demonstrated on enriching of low abundance viral RNA[34]. Porcupine expanded the demultiplexing scale, proposing a 96-plex 40 bp DNA tag system with 96.9% accuracy on a test dataset[35] (Details of reported performance are summarized in Table 2). However, a systematic comparison is missing and there is still plenty room for DNA tags to improve classification confidence and yield when increasing the total number of barcodes[36].

Here, we propose and benchmark SUSTag - an end-to-end 384-plex molecular tagging system designed for random access of DNA data, to unleash the limitations of dynamic decisions, classification confidence and yield. Our work encompasses two key contributions to enhance DNA tag capabilities in nanopore sequencing. Firstly, we present SUSTag that leverages Bhattacharyya distance for the evaluation of DNA tags, along with incremental clustering strategy to optimize the nonlinear mapping of base sequences to nanopore signatures. Secondly, we introduce ORCtrL: an optional-reject module inspired by transfer learning to maximize adaptability to the complex and unstable nature of nanopore sequencing signals within domain adaptation. A systematic benchmark of SUSTag demonstrated impressive classification performance with an F1-score (weighted averages; Methods) of 99.69% for 96-plex classification. Our work further expanded it to 384-plex classification, achieving a F1-score of 99.05% on the mixed barcode dataset. Furthermore, by using SUSTag as address bits for DNA storage and adapting the ORCtrL to the new data domain, we enabled PCR-free random access to DNA-stored data based on nanopore sequencing.

## Results

### SUSTag design suppresses nanopore current similarity using Bhattacharyya distance and incremental clustering

Figure 1a illustrates the scenario for random-access of stored DNA data using nanopore sequencing. Known a priori, these orange DNA tags as address bits are assembled with information bits, ahead of being read via nanopore sequencing. Correct readout of these tags enables selective addressing and retrieval of information stored within the DNA at will, such as the eyes and nose of a 'smile face'. However, either the targeted address bits or the information bits could be problematic upon inherently high error rates of nanopore sequencing[37], illustrated by an accident example of a 'crying face'.

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: 1) Our BD-DTW, which treats noise properties (signal variance) as being

https://doi.org/10.1038/s41467-025-64293-2

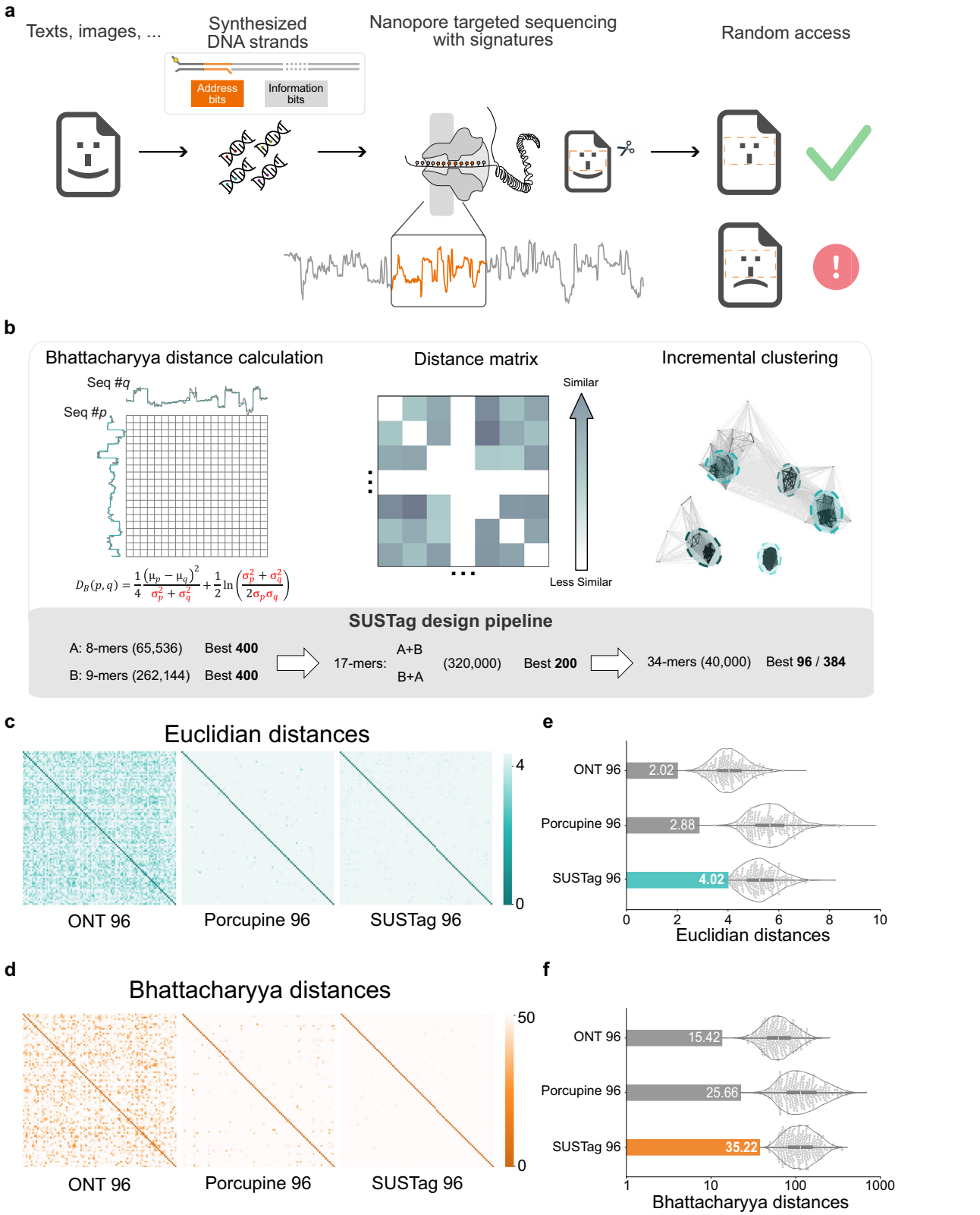



**Fig. 1 | Design of low-crosstalk molecular tags featured with nanopore signatures as addressing information for DNA storage. a** Schematics of random access using nanopore sequencing for digital contents stored in DNA. Orange current fluctuations represent signatures for specific addressing bits. **b** De novo design method for proposed SUSTag including maximized Bhattacharyya distance, distance confusion matrix and incremental clustering. **c** In silico performance evaluation for ONT 96, Porcupine 96 and SUSTag 96 using Euclidean distance matrices. **d** In silico performance evaluation for ONT 96, Porcupine 96 and SUSTag 96 using Bhattacharyya distance matrices. Statistical comparison of in silico Euclidean (**e**) and Bhattacharyya (**f**) distances. For each 96-plex set, the data distribution is shown as a violin plot, derived from $N = 200$ randomly sampled pairs of distinct barcodes. The inner box plot indicates the median (center line), interquartile range (IQR, box bounds), and whiskers (1.5×IQR). The separate bar chart and value highlight the observed minimal distance.

as informative as the mean signal levels (squiggles), is used to estimate the simulated inter-signal distances between all pairs of sequence candidates, forming a BD map in the left panel. 2) Unlike Euclidean distance (ED), the BD 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. 3) A brute-force computation of BD for all $4^{34}$ candidate 34-mer sequences is computationally prohibitive. Therefore, we have adopted an incremental design strategy. The strategy evaluates shorter sequences and employs Linclust, an algorithm that scales linearly with the number of sequences, before extending to the full 34-nt tag length. Initially, we exhaustively estimate BD of simulated current signals for all 8-mer and 9-mer sequences, distilling the best 400 tags from each set. Then, 17-mer tags are concatenated and evaluated with the best 200 tags, which are subsequently concatenated to form 34-nt candidate tags for SUSTag 96 and SUSTag 384. (See Methods for details).

The performance of our BD-DTW designed **SUSTag 96** is evaluated in silico in Figs. 1c-f, in comparison with 96 ONT native barcodes (so-called **ONT 96**) and Doroschak et al.[35] (**Porcupine 96**). Figure 1c, e shows the Euclidean distance matrices and the distribution of correlated Euclidean distances. 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). In Fig. 1e, the minimal E-distance of 4.02 is higher than those of 2.02 and 2.88 for ONT 96 and Porcupine 96, even though our SUSTag 96 is not intentionally optimized for the dissimilarity of mean values. This is likely due to the less noisy feature of ionic currents. Furthermore, Fig. 1d, f shows the in-silico results of Bhattacharyya distances, where SUSTag 96 holds an off-diagonal diminished confusion matrix and the minimal B-distance of 35.22, beating 15.42 from ONT 96 and 25.66 from Porcupine 96, significantly. This indicates that SUSTag 96 should in theory have a lower crosstalk rate and higher classification accuracy potential.

## High demultiplexing accuracy and yield require deep-learning network with optional-rejections

To benchmark the low crosstalk characteristics of our BD-DTW based design method, Fig. 2a shows the assembled SUSTag 96, Porcupine 96, and ONT 96 into a single sequence (see Supplementary Fig. S3 for the SUSTag 384 construct), where barcodes from each design are separated by eight poly(thymine) bases for avoiding the overlapping of sequencing current signatures. Figure 2c illustrates the confusion matrices for ONT 96 (blue), Porcupine 96 (purple), and SUSTag 96 (orange), yielding F1-scores of 0.9925, 0.9891 and 0.9969 in the 1-96 classification. Compared with SUStag 96 and ONT 96, Porcupine 96 presented higher near-diagonal crosstalk, as detailed in Supplementary Table S10.

Figure 2b shows our ORCtrL architecture, where a four-layer CNN aims to extract features from the sequencing signals and LSTM blocks are used to learn signal dependencies over longer time scales. The capability of optional rejection is enabled via a selection module, which returns binary (yes or no) results that mask the prediction unit or not. This allows us to process unwanted barcodes or the sequences without barcodes, which are all considered as #0 class. Figure 2d summarize the performance for well-established deep-learning methods: the CNN model following Porcupine's architecture (CNN-Porcupine), the modified CNN model and the modified CNN-LSTM model. Only CNN Porcupine omits the #0 class for unclassified data. The corresponding overall F1-scores were 0.9543, 0.9852 and 0.9892, respectively. Due to the involvement of the unclassified #0 label, the performances of the CNN and CNN-LSTM methods are significantly better than that of CNN-Porcupine. Moreover, the inclusion of the LSTM module enhances the model's performance in distinguishing the #0 label, which can be observed in the first column of the confusion matrices in Fig. 2d. Supplementary Table S2 shows the improved performance to 0.9974 when the selection, conventional prediction as well as auxiliary modules are functional. This also suggests the improved the model's precision may be raised by the cost of a slight loss in data coverage.

The performance of three types of tag designs together with three deep-learning models is depicted in Fig. 2e. Specifically, the precisions are found all better than the recall rates. Our SUSTag 96 led by precisions of 99.69%, 99.21%, and 99.07% and recall of 99.69%, 99.07%, and 98.66% across these three models. Furthermore, the ORCtrL model outperformed the conventional CNN model by up to 2.64% in F1-score for the same barcodes (Supplementary Table S3), shedding light towards practical use.

## Domain adaptation enables SUSTag 384 adapted for DNA storage datasets

We turn our focus onto synthetic DNA storage sequences from mixed-barcode sequences, shown in Fig. 3a, where SUSTag as 34 nt address bits (orange) are expanded from 96 to 384. SUSTech's introductory text data and the university's emblem and motto image data are arranged neighboring to our address bit using the CHN codec system[42] (Supplementary Table S12). To confirm the transition from 96 to 384 barcodes, Fig. 3b presents the crosstalk performances for SUSTag 384, in comparison with SUSTag 96 and the other two 96 tags. The distributions of crosstalk ratios were nearly alike for SUSTag 96 and 384, thereby illustrating the consistency in our sequence design between SUSTag 384 and SUSTag 96.

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 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).

Figure 3c, d illustrates the performance of ORCtrL, measured by F1-score, when adapting to this new data domain. The F1-score exhibited a logarithmic increase from 91.84% to 94.56% in Fig. 3c, achieving a plateau of optimal performance with data size of 300,000 reads using filtered and balanced DNA storage datasets. Figure 3d presents the results of fine-tuning using unprocessed DNA storage data accumulated over different sequencing durations, reflecting a more realistic application scenario. A similar logarithmic trend was observed, with the F1-score rising from 87.17% to 94.28% using DNA storage data accumulated over different sequencing durations, reaching a stable performance in 60 min of sequencing. Notably, the maximum F1-score achieved with the unprocessed dataset was 0.28% lower than that obtained with the pre-processed dataset. This discrepancy likely stems from the uneven intra-class read coverage present in the nanopore sequencing.

We further analyze the detailed performance of domain adaptation between precision and recall, as shown in Fig. 3e. The pre-trained model, without fine-tuning, exhibited a precision of 88.72% and a recall of 87.23%. For the pre-processed datasets of varying sizes, a clear linear relationship between precision and recall is observed, represented by the orange points. Similarly, the green points, corresponding to the unprocessed datasets, also demonstrate a linear trend. The higher precision observed for the pre-processed datasets is expected and highlights the effectiveness of the optional-reject functionality within ORCtrL. This suggests that the model is more precisely assigning SUSTag classifications when trained on cleaner, balanced data, leading

https://doi.org/10.1038/s41467-025-64293-2

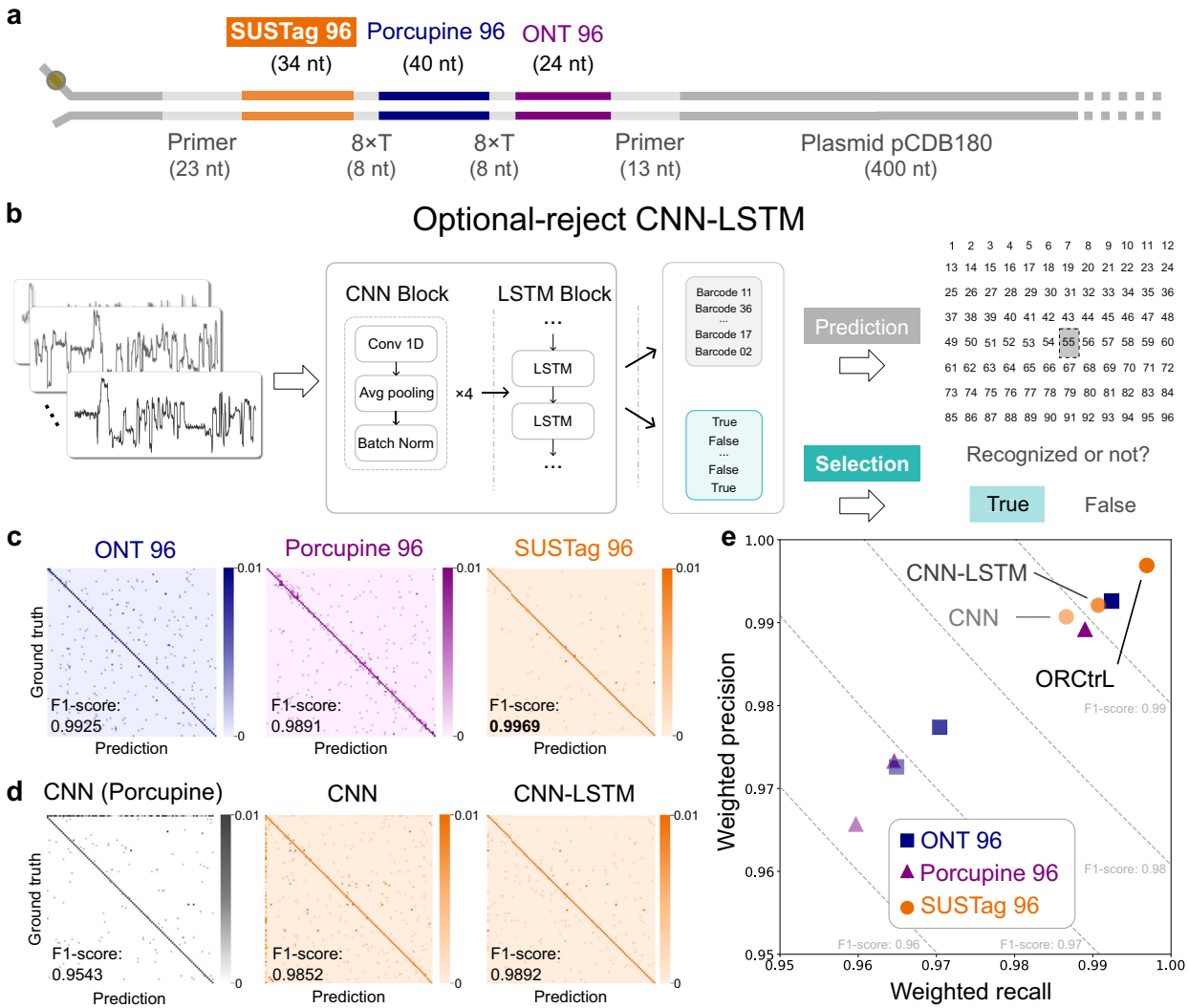



**Fig. 2 | Experimental benchmarking of tag designs and deep-learning classifiers. a** Sequence structure containing SUSTag and other tags (ONT 96 and Porcupine 96) as well as primer sequences and an arbitrary portion of plasmid pCDB180. **b** Schematic diagram of the optional-reject CNN-LSTM network architecture. The model uses a CNN-LSTM network as the backbone, followed by multiple output heads forming a predictionand selection structure. **c** Experimental confusion matrices for classifying different barcode designs (ONT, Porcupine and SUSTag 96) using the ORCtrL model. The ORCtrL model was trained with a target coverage of 0.95. **d** Experimental confusion matrices for classifying SUSTag 96 using three deep-learning methods: they are CNN (Porcupine), CNN and CNN-LSTM. **e** Precision vs. Recall for classifying three types of barcode designs using three different deep-learning models. The ORCtrL model was trained with a target coverage of 0.95. Orange, Navy Blue and Purple stand for SUSTag, ONT and Porcupine designs. Source data are provided as a Source Data file.

to a higher proportion of true positives among the predicted positives, thereby boosting precision.

With quantified performance of model adaptation steps, we demonstrate the random-access capability of our ORCtrL model on #1-#51 over all 384 sequences. Figure 3f, g shows the performance without and with domain adaptation as the targeting regions. When directly applied the pre-trained model, the access ratio of non-targeting sequences yields 2.53% ± 2.31%, while the targeting region was 74.30% ± 7.11%. For the domain-adapted model, the access ratio of non-targeting sequences drops to 0.82% ± 0.71%, whereas the targeting region stays at 76.44% ± 6.83%.

### SUSTag 384 demonstrates rapid and low crosstalk PCR-free random access for DNA storage

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 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 domain adapted 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 10 and 60 min using Readfish methods, which converts raw data by Guppy basecaller and then maps to a reference with minimap2. Although no other crosstalk sentences came out, there were a few targeted sentences missing. Figure 4b displays the readout examples using the Porcupine CNN model. In this case, the targeted regiond together with quite a lot unwanted sentences were decoded both in 10 and 60 min. Figure 4c depicts the decoded texts in 10, 30, 60

https://doi.org/10.1038/s41467-025-64293-2

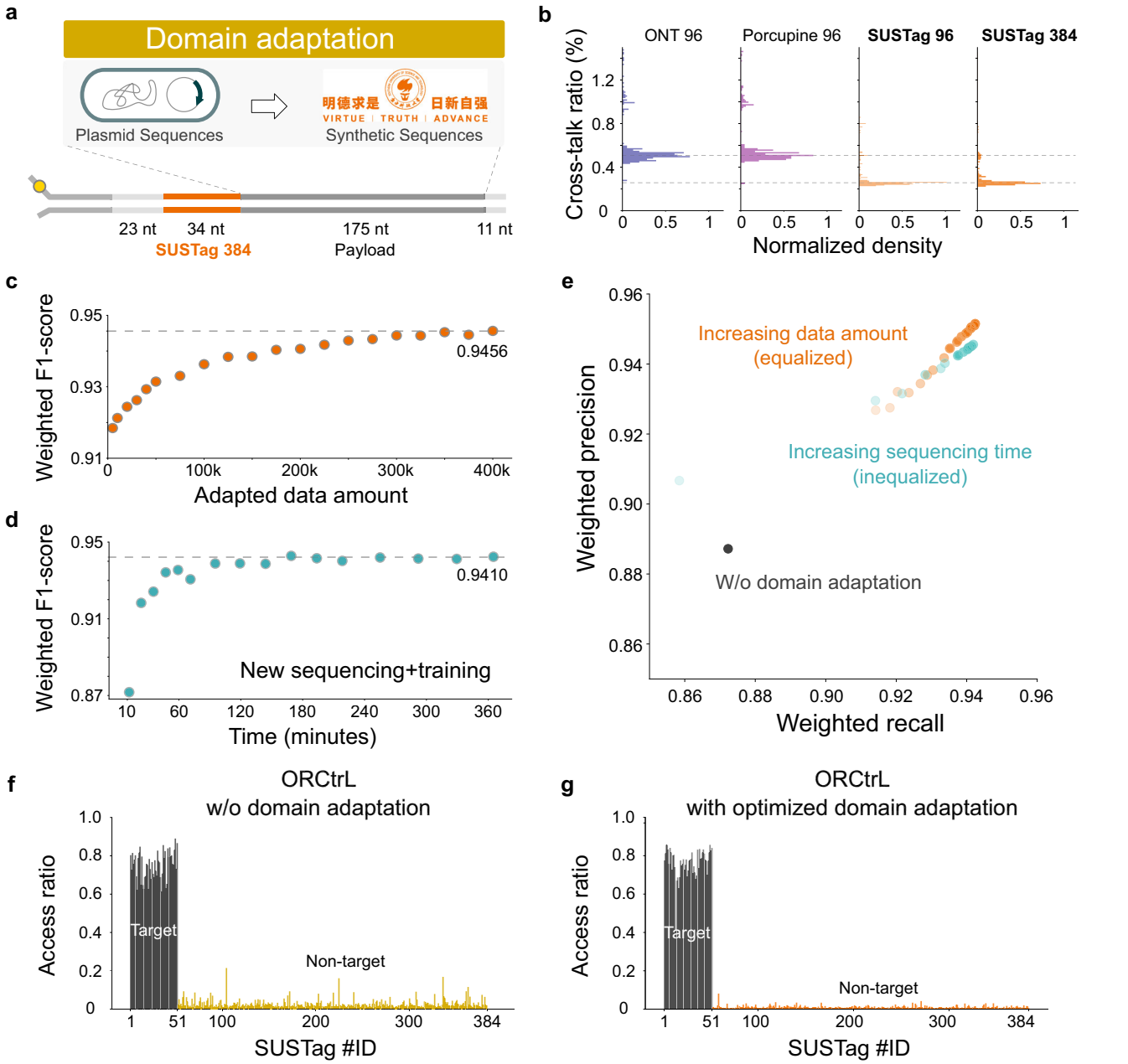
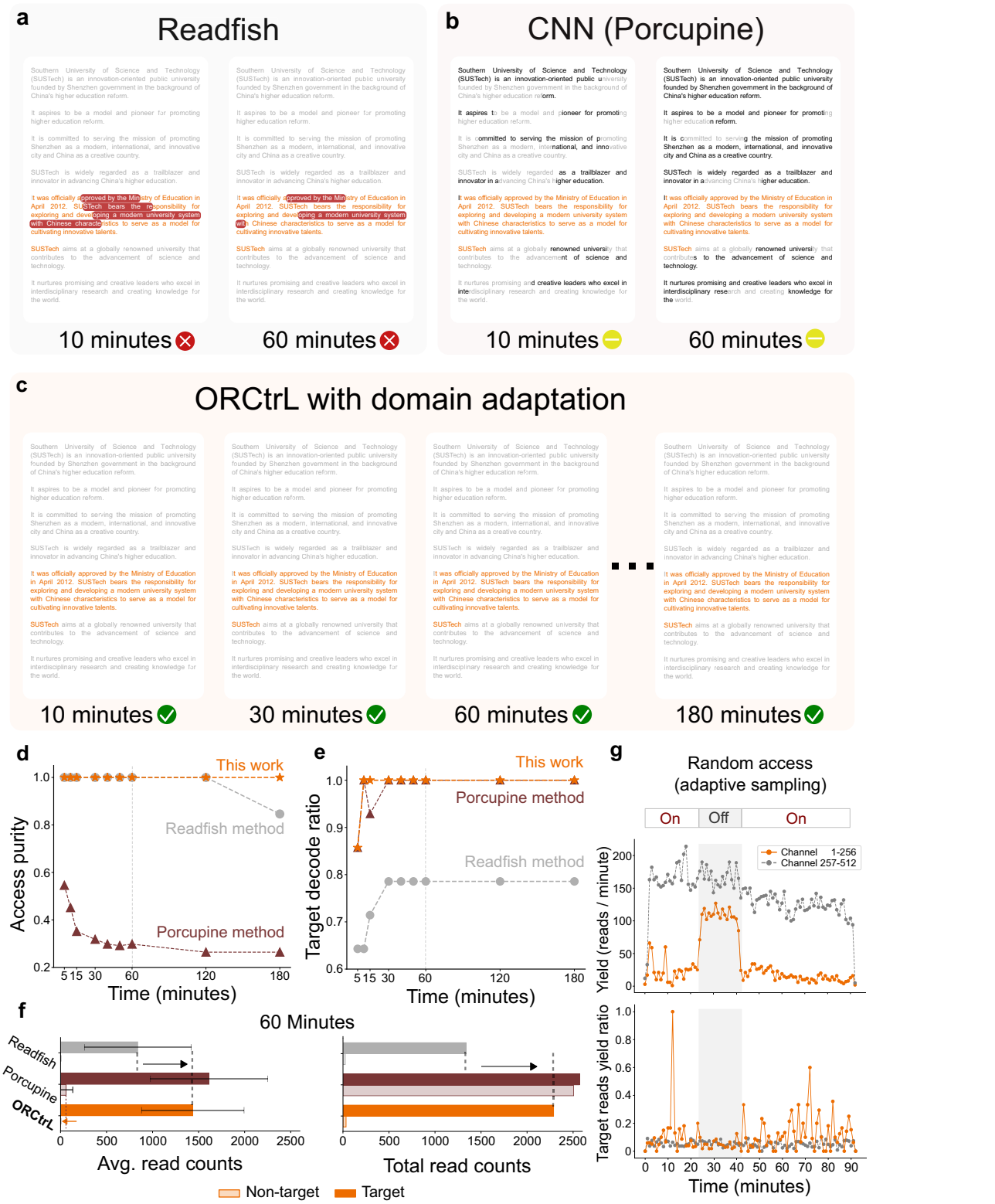



**Fig. 3 | Domain adaptation for DNA storage application. a** Illustration of address bits (barcodes) for DNA information storage. **b** Statistical chart of error patterns for four barcode classifications. The crosstalk of SUSTag 96 and SUSTag 384 is significantly lower than that of ONT 96 and Porcupine 96, with similar error patterns observed. **c** Weighted F1-score as a function of data amount from new domain. **d** Weighted F1-score as a function of total running time (sequencing plus training) for new domain data. **e** Scatter plot of both equalized data volume evolved (orange) and equalized sequencing time evolved (cyan) effect of different fine-tuning datasets on model performance. The gray points show the performance of the model without domain adaptation as a control. **f** Access ratios of 384 address bits for the ORCtrL model without domain adaptation. **g** Access ratios of the identical 384 address bits after domain adaptation. Source data are provided as a Source Data file. Source data are provided as a Source Data file.

and 180 min using domain adapted ORCtrL, where correct readouts can be observed on all the four conditions, showcasing neither missing of targeted addresses nor readout of unwanted tags.

To further characterize the random-access performance, we evaluated the decision precision (denoted as access purity) and target decode ratio (Decoded target sequences/Total target sequences) for decoded strands and plotted them as a function of sequencing time in Fig. 4d, e. Here, the access purity could serve as an indicator for enriching target sequences while suppressing non-target sequences. Our ORCtrL method in Fig. 4d does achieve 100% access purity ranging from 5 min to 180 min, whereas Readfish shows quite competitive 100% access purity in two hours. Meanwhile, Porcupine CNN exhibits a decay of access purity from 60% to 25%, which can be attributed to more non-targeted readouts, eventually forming texts. In Fig. 4e, ORCtrL can sustain a 100% target sequence decoding rate once the sequencing duration exceeds 10 min. Porcupine CNN model could catch up on this point in 30 min. However, the Readfish method reaches a plateau in 80% in 30 min. Figure 4f illustrates total number of reads and the average number of reads between targeted and non-targeted regions for 60 min. ORCtrL showed approximately 1.7×–1.8×

https://doi.org/10.1038/s41467-025-64293-2




total and average read counts compared to the Readfish method for the target region, while maintaining the same extremely low level of non-target region data. Compared to the Porcupine CNN method, ORCtrL only lost about 10.92% in average coverage of the target sequences, while obtaining as low as 0.78x average coverage for non-target sequences, demonstrating significant crosstalk suppression during on-demand data accessing.

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

**Fig. 4 | Adaptive nanopore sequencing based random access in DNA storage.** **a** Decoding results of PCR-free random-access data obtained using the Readfish method with 10 min and 1 h of sequencing data. **b** Decoding results of PCR-free random-access data obtained using the Porcupine method with 10 min and 1 h of sequencing data. **c** Decoding results of random-accessed data obtained using the proposed method with 10 min, 30 min, 1 h, and 3 h of sequencing data. **d** Access purity changes over time for the three methods. **e** Target sequences decoded ratio changes over time for the three methods. **f** Sequencing data enrichment performance of the three methods, derived from a 60-minute sequencing data. In the left panel, bars represent the mean read counts per barcode, with error bars indicating the standard deviation (SD). The right panel shows the total number of reads classified for each method. The statistics were calculated across distinct barcode categories within each group. $N = 14$ for the target group (barcodes #29 to #42) and $N = 370$ for the non-target group. **g** Comparison of cumulative read yields over time between adaptive sampling channels and control channels. Source data are provided as a Source Data file.

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.

## Discussion

Our system, SUSTag-ORCtrL, introduces an integrated framework for real-time, PCR-free random access to DNA-encoded information using nanopore sequencing. It tackles the core challenges of efficient DNA tag demultiplexing and signal classification, offering a significant improvement in addressing both accuracy and demultiplexing limitations, as further benchmarked in Tables 1 and 2. This framework combines innovations in tag design, deep learning classification, and domain adaptation to enable a higher rate of information gain during targeted DNA retrieval, leading to both reduced crosstalk and improved data recovery. By addressing fundamental constraints in existing methods, SUSTag-ORCtrL represents a significant step towards more practical and efficient applications in areas like DNA data storage and targeted sequencing.

Our benchmarking of ORCtrL, using the same ONT native 96 barcodes as DeepBinner[31] and ReadFish[32], revealed its leading performance in both precision (99.26 % vs 98.5% and ~90%) and recall (99.24% vs 94.3% and ~90%). In addition, Supplementary Fig. S1 displays our classification result of the Porcupine 96 using a CNN architecture consistent with Doroschak et al.[35], achieving a test-set accuracy of 95.97%, which is in a reasonable agreement with their published value. It is worth noting that these results underscore the importance of both including an unclassified label (termed label #0 in this work) during deep learning training and implementing an optional-reject module for effective demultiplexing.

We address optional rejection and compulsory domain adaptation as the key for enabling our SUSTag-ORCtrL. The incorporation of the optional rejection mechanism improves the precision of the CNN-LSTM model from Q21 (99.21%) to Q25 (99.69%), as illustrated in Fig. 2d. This enhancement can be attributed to the selection module within the network effectively rejecting ambiguous sequences. Furthermore, we demonstrate the necessity for adaptation to new data domains. When the pre-trained model is directly applied to the DNA storage data, its precision drops significantly to Q9 (88.72%). However, appropriate domain adaption recovers performance, raising the precision back to Q13 (95.23%), 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.

Less than 300 nt sequences can be correctly handled using our end-to-end ORCtrL system, which fills the blank of ~360nt the state-of-the-art pipelines that require basecalling-and-mapping-to-a-reference[32]. We achieved random access in DNA storage data with sequencing length N50 of 283. Furthermore, if higher precision is needed for other DNA storage systems, ORCtrL offers the flexibility to adjust the optional rejection module. This allows the coverage parameter to be tuned during training, with lower coverage typically resulting in higher precision and, consequently, reduced crosstalk (Supplementary Table S2). While deep learning architectures are constantly evolving, we anticipate that the optional rejection module will remain a core component. Future enhancements could involve incorporating more sophisticated networks to further improve accuracy or streamlining the network for faster processing speeds, depending on the specific computational resources and performance demands.

Notably, the SUSTag design methodology currently does not impose any limits on the arbitrary number of tags or pore model, resulting for the nonlinear projection from ionic current signals to base sequences. The scalability of SUSTag is facilitated by our dynamic incremental clustering strategy, which allows for customization of the final design based on individual requirements. As tools for simulating sequencing signals from R10 pore model mature, our design method can be seamlessly migrated from R9.4.1 flow cells. For instance, seq2squiggle[43] has recently demonstrated remarkable similarity to real data on the R10.4.1 flow cell, providing a powerful tool for simulating nanopore sequencing signals with the latest generation of flow cells. We plan to extend the SUSTag to the R10 flow cell to accommodate the unique current signal characteristics of newer nanopore chemistries and explore the full potential of this molecular tagging system. 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.

## Methods

### SUSTag design algorithm

We started from using the scrappie tool (v1.4.2) from Oxford Nanopore Technology (ONT) to simulate the electrical signatures from DNA sequences, which aims to produce the differences on electrical signals as large as possible. These DNA sequences (also termed as barcodes or tags) are designed as the following three steps: First, simulated current signals of $4^8 = 65{,}536$ and $4^9 = 262{,}144$ combinations were exhaustively generated based on the known k-mer table with experimental mean current μ and standard deviation σ for k-mers (k = 5, scrappie squiggle). Second, Dynamic Time Warping (DTW) algorithm was applied to calculate pairwise distances between the squiggled signals, resulting in a distance matrix. The pairwise distance can be calculated by either Euclidean distance or Bhattacharyya distance. The formula for calculating the Euclidean distance $D_E$ and Bhattacharyya distance $D_B$ between signal $p \sim \mathcal{N}(\mu_p, \sigma_p^2)$ and

signal $q \sim \mathcal{N}\left(\mu_q, \sigma_q^2\right)$ is given by:

$$D_E(p,q) = \left(\mu_p - \mu_q\right)^2 \tag{1}$$

$$D_B(p,q) = \frac{1}{4}\frac{\left(\mu_p - \mu_q\right)^2}{\sigma_p^2 + \sigma_q^2} + \frac{1}{2}\ln\left(\frac{\sigma_p^2 + \sigma_q^2}{2\sigma_p\sigma_q}\right) \tag{2}$$

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 is as follows: We systematically adjusted the minimum inter-cluster distance threshold, which defines the dissimilarity required to form a new cluster based on the pre-computed Bhattacharyya distances. The number of final clusters is inversely related to this threshold. We iterated this threshold value in descending order until the total number of resulting clusters was just greater than or equal to our target number ($N = 400$). At this stage, a representative sequence was chosen from each of the ≥400 clusters. To obtain the final set, we then selected the 400 representative sequences from this group that were most distant from one another, ensuring the final set has the lowest possible crosstalk potential.

Next, we concatenated these 8-mers and 9-mers for 17-mers sequences, achieving a total of 320,000 possibilities. Again, we re-simulated the current signatures, re-squiggled and re-calculated the distances, and distilled top 200 strands 17-mer sequences as representative.

These 17-mers were then paired and concatenated to form a total of 39,800 strands of 34-mer sequences. Following the same distilling process, we obtained a set of 500 SUSTag sequences. A mutation process was adapted to the neighboring four based between 8-mers and 9-mers, further improving the distances for designed sequences.

Additionally, the final barcode design should satisfy the following three constraints: (1) The GC content of the barcode is between 30% and 70%. (2) The maximum length of any homopolymer within the barcode is less than 3. (3) The sequences avoid the BsaI restriction endonuclease site in 5′ GGTCTC to accommodate subsequent experiments. Among these, 100 sequences were selected as 96 classification barcodes (with 4 sequences as redundancy), and the remaining 400 sequences were designated as 384 classification barcodes (with 16 sequences as redundancy).

### Synthesized barcode sequences

To fairly compare with the state-of-the-art, we devised an arrangement combining SUSTag (96 + 384 barcodes), ONT native barcodes (96 barcodes, https://nanoporetech.com/document/ligation-sequencing-amplicons-native-barcoding-v14-sqk-nbd114-96), and Porcupine (96 barcodes) to create mixed sequences of 150 nt in length. The specific arrangement method is as follows: The first 400 mixed sequences are arranged in the order of SUSTag 384 - ONT barcode - Porcupine; the subsequent 100 mixed sequences follow the order of SUSTag 96 - Porcupine - ONT barcode. Furthermore, mixed sequences are indexed starting from 0, denoted by $i$. The sequence of SUSTag is arranged consecutively, the indexing for the ONT barcode is determined by $i \bmod 400 \bmod 96$, and the indexing for Porcupine is calculated using $(i \bmod 400 + \lfloor i \bmod 400/96 \rfloor) \bmod 96$.

To delineate different barcodes, an 8-mer spacer sequence 5′ TTTTTTTT is used, and the end of each mixed barcode is uniformly marked with 5′ AAAAAAAA to signify the end of the barcode and the subsequent plasmid spacer. Detailed SUSTag96 and SUSTag384 sequences are provided in Supplementary Table S4, S5.

### Classification models

We employed a hybrid model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) as the main backbone, subsequently integrating a Selective Net to determine whether the model's input is a barcode signal. The data fed into the network first passes through a CNN block, which comprises four layers. Each of these CNN layers possesses an identical structure, including a 1D convolutional layer with ReLU activation, average pooling, and batch normalization. Following this, the signal features extracted by the CNN are fed into a standard two-layer LSTM. Finally, the backbone model connects to three components: a classifier, a selector, and an auxiliary classifier. The classifier, a fully connected layer, categorizes the extracted features, yielding predictions of the classes. The selector, a multilayer perceptron, forecasts whether the classification result of the current sample should be rejected, outputting a value between 0 and 1 that represents the probability of rejection. 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 head 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 diagrams of this model and the comparative CNN and CNN-LSTM models, including their precise parameters, in Supplemental Table S7-9.

### Experimental methods

**Mixed barcode sequences preparation.** We synthesized the mixed barcode sequences (150 nt in length, 500 in classification) through Twist Biosciences. After receiving the synthetic product as lyophilized DNA (110 ng), we dissolved it in 20 μL of nuclease-free water and stored it at −20 °C. To generate an insertion spacer, we amplified an arbitrary 400 nt segment of the plasmid pCDB180 (https://www.addgene.org/80677/) by PCR, adding BsaI restriction sites to the ends of the amplified product using a primer design identical to that used in the Porcupine. We also added BsaI sites to the reverse primer of the mixed barcode sequences. Primer sequences are provided in Supplementary Table S6. After amplifying the barcode and plasmid sequences separately by PCR, we performed digestion using the NEB BsaI-HF® v2 protocol, followed by rapid ligation of the mixed barcode sequences to the plasmid sequences using T4 DNA Ligase (Beyotime D7006). For sequencing preparation, we followed the ONT SQK-PCS109 protocol starting from the *'Selecting for full-length transcripts by PCR'* step, as the sample preparation was already completed. Sequencing was conducted using the ONT EXP-FLP001 protocol on an R9.4.1 MinION flowcell. We chose FAST5 as the file save format in MinKNOW (v24.02.8), disabling real-time basecalling, with independent basecalling performed by the Guppy basecaller (v6.0.0) super accuracy model.

**DNA storage sequences preparation.** We stored a text file and an image using the four-letter setup of our Composite Hedges Nanopores codec system[42], encoded a total of 487 sequences, with the SUSTag 384 barcode added at the front as the address bits for data storage. The text and image file with encoded sequences are provided in Supplementary Fig. S2 and Supplementary Table S11. Similarly, we sent these sequences to Twist Biosciences for synthesis. Upon receiving the synthetic product as lyophilized DNA (110 ng), we dissolved it in 20 μL of nuclease-free water. We then used 1 μL of this solution for PCR amplification. The PCR products underwent end preparation where dA-tails were added to their ends and rapid ligation with ONT AMX-F adapters following the Vazyme ND608 protocol. The resulting products were purified and then loaded for sequencing following ONT SQK-LSK110 protocol. Specifically, we began with the *'Adapter Ligation and Clean-up'* section of the protocol, skipping the adapter ligation itself as it had already been conducted. The sequencing was performed on an R9.4.1 MinION flowcell, with MinKNOW settings and basecalling configurations maintained identical to those used for the mixed sequences experiment.

### 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 domain-adapted 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.

### Data collection and model training

For the mixed barcode sequences, we performed a total of 37.8 h of sequencing using MinION, resulting in approximately 118 GB of FAST5 data, including about 11.76 million reads. After basecalling, we mapped the translated results with the mixed barcode sequences as references using minimap2 (v2.24). We annotated the reads based on the mapping results to create datasets. Specifically, we prepared four datasets labeled according to ONT barcode, Porcupine, SUSTag96, and SUSTag384. To ensure data balance, we limited the number of reads for each barcode category to 4000, and additionally, we included reads not mapped to the reference as class #0, which constituted 10% of the dataset. After dataset preparation, we randomly split 10% of the data as the test set, with the remaining 90% used for the training set. In terms of data preprocessing, we identified the positions of three barcode sequences (SUSTag96/SUSTag384, ONT barcode and Porcupine) in each read using mapping results, and applied masking to the unlabeled barcode sequences to prevent them from influencing model training. Finally, we trimmed the first 3000 data points to compile the final dataset.

We constructed our model using PyTorch (v2.3.1) and trained it on a workstation (CPU: AMD 5950X, GPU: NVIDIA RTX 3090, RAM: 128GB). We trained the model for 30 epochs using the Adam optimizer, with a batch size of 512, and utilized 10% of the training set as a validation set. 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.

### 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 min) 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.

### Evaluation metrics

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. 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, 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.

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

$$\text{Access purity} = \frac{\text{Number of correctly accepted target reads}}{\text{Total number of accepted reads}} \quad (3)$$

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

$$\text{Target decode ratio} = \frac{\text{Number of successfully decoded target sequences}}{\text{Total number of target sequences}} \quad (4)$$

### Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

## Data availability

The sequencing FAST5 data that used to train the model and the model.pth files have been deposited in the Zenodo database under https://doi.org/10.5281/zenodo.17112794 [https://doi.org/10.5281/zenodo.17112794]. Full sequences for this work can be found in Supplementary Information. All other data described in this work are available in the main text, supplied in the supplementary materials, or can be reproduced using deposited datasets and GitHub code. Source data are provided with this paper.

## Code availability

The complete codebase for this study, along with a short guide to the code, is available on GitHub at https://github.com/Taltalite/sustag_orctrl and is also available on Zenodo under the https://doi.org/10.5281/zenodo.17115519.

## References

1. Mullard, A. DNA tags help the hunt for drugs. *Nature* **530**, 367–369 (2016).
2. Kuzdraliński, A. et al. Unlocking the potential of DNA-based tagging: current market solutions and expanding horizons. *Nat. Commun.* **14**, 6052 (2023).

3. Russell, A. J. C. et al. Slide-tags enables single-nucleus barcoding for multimodal spatial genomics. *Nature* **625**, 101–109 (2024).
4. Klein, A. M. et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187–1201 (2015).
5. Koch, C. et al. Nanopore sequencing of DNA-barcoded probes for highly multiplexed detection of microRNA, proteins and small biomarkers. *Nat. Nanotechnol.* **18**, 1483–1491 (2023).
6. Koch, J., Doswald, S., Mikutis, G., Stark, W. J. & Grass, R. N. Ecotoxicological assessment of DNA-tagged silica particles for environmental tracing. *Environ. Sci. Technol.* **55**, 6867–6875 (2021).
7. Xu, G., Li, H., Ren, H., Lin, X. & Shen, X. DNA similarity search with access control over encrypted cloud data. *IEEE Trans. Cloud Comput.* **10**, 1233–1252 (2022).
8. Bee, C. et al. Molecular-level similarity search brings computing to DNA data storage. *Nat. Commun.* **12**, 4764 (2021).
9. Shah, S. et al. Using strand displacing polymerase to program chemical reaction networks. *J. Am. Chem. Soc.* **142**, 9587–9593 (2020).
10. Lv, H. et al. DNA-based programmable gate arrays for general-purpose DNA computing. *Nature* **622**, 292–300 (2023).
11. Koch, J. et al. A DNA-of-things storage architecture to create materials with embedded memory. *Nat. Biotechnol.* **38**, 39–43 (2020).
12. Meiser, L. C. et al. Synthetic DNA applications in information technology. *Nat. Commun.* **13**, 352 (2022).
13. Organick, L. et al. Random access in large-scale DNA data storage. *Nat. Biotechnol.* **36**, 242–248 (2018).
14. Kozarewa, I., Armisen, J., Gardner, A. F., Slatko, B. E. & Hendrickson, C. L. Overview of target enrichment strategies. *Curr. Protoc. Mol. Biol.* **2015**, 1–23 (2015).
15. Xie, N. G. et al. Designing highly multiplex PCR primer sets with Simulated Annealing Design using Dimer Likelihood Estimation (SADDLE). *Nat. Commun.* **13**, 1881 (2022).
16. Zhang, J. X. et al. A deep learning model for predicting next-generation sequencing depth from DNA sequence. *Nat. Commun.* **12**, 4387 (2021).
17. Singh, R. R. Target enrichment approaches for next-generation sequencing applications in oncology. *Diagnostics* **12**, 1539 (2022).
18. Deamer, D., Akeson, M. & Branton, D. Three decades of nanopore sequencing. *Nat. Biotechnol.* **34**, 518–524 (2016).
19. Loose, M., Malla, S. & Stout, M. Real-time selective sequencing using nanopore technology. *Nat. Methods* **13**, 751–754 (2016).
20. Martin, S. et al. Nanopore adaptive sampling: a tool for enrichment of low abundance species in metagenomic samples. *Genome Biol.* **23**, 11 (2022).
21. Weilguny, L. et al. Dynamic, adaptive sampling during nanopore sequencing using Bayesian experimental design. *Nat. Biotechnol.* **41**, 1018–1025 (2023).
22. Sauerborn, E. et al. Detection of hidden antibiotic resistance through real-time genomics. *Nat. Commun.* **15**, 5494 (2024).
23. Sereika, M. et al. Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nat. Methods* **19**, 823–826 (2022).
24. Sokolovskii, R., Ramirez-Garcia, R. & Heinis, T. Adaptive sampling in nanopore sequencing for PCR-free random access in DNA data storage. *bioRxiv* 2024.11.05.622081 https://doi.org/10.1101/2024.11.05.622081 (2024).
25. Payne, A. et al. Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nat. Biotechnol.* **39**, 442–450 (2021).
26. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
27. Ulrich, J. U., Lutfi, A., Rutzen, K. & Renard, B. Y. ReadBouncer: precise and scalable adaptive sampling for nanopore sequencing. *Bioinformatics* **38**, I153–I160 (2022).
28. Kovaka, S., Fan, Y., Ni, B., Timp, W. & Schatz, M. C. Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. *Nat. Biotechnol.* **39**, 431–441 (2021).
29. Zhang, H. et al. Real-time mapping of nanopore raw signals. *Bioinformatics* **37**, I477–I483 (2021).
30. Bao, Y. et al. SquiggleNet: real-time, direct classification of nanopore signals. *Genome Biol.* **22**, 298 (2021).
31. Wick, R. R., Judd, L. M. & Holt, K. E. Deepbinner: demultiplexing barcoded Oxford Nanopore reads with deep convolutional neural networks. *PLOS Comput. Biol.* **14**, e1006583 (2018).
32. Payne, A. et al. Barcode aware adaptive sampling for GridION and PromethION Oxford Nanopore sequencers. *bioRxiv* 2021.12.01.470722 (2022).
33. Smith, M. A. et al. Molecular barcoding of native RNAs using nanopore sequencing and deep learning. *Genome Res.* **30**, 1345–1353 (2020).
34. van der Toorn, W., Bohn, P. & Liu-Wei, W. Demultiplexing and barcode-specific adaptive sampling for nanopore direct RNA sequencing. *Nat. Commun.* **16**, 3742 (2025).
35. Doroschak, K. et al. Rapid and robust assembly and decoding of molecular tags with DNA-based nanopore signatures. *Nat. Commun.* **11**, 5454 (2020).
36. Gehring, J., Hwee Park, J., Chen, S., Thomson, M. & Pachter, L. Highly multiplexed single-cell RNA-seq by DNA oligonucleotide tagging of cellular proteins. *Nat. Biotechnol.* **38**, 35–38 (2020).
37. Logsdon, G. A., Vollger, M. R. & Eichler, E. E. Long-read human genome sequencing and its applications. *Nat. Rev. Genet.* **21**, 597–614 (2020).
38. Skutkova, H., Vitek, M., Sedlar, K. & Provaznik, I. Progressive alignment of genomic signals by multiple dynamic time warping. *J. Theor. Biol.* **385**, 20–30 (2015).
39. Bhattacharyya, A. On a Measure of Divergence between Two Multinomial Populations. *Sankhyā Indian J. Stat.* **7**, 401–406 (1946).
40. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).
41. D'Urso, P., De Giovanni, L. & Vitale, V. Robust DTW-based entropy fuzzy clustering of time series. *Ann. Oper. Res.* https://doi.org/10.1007/s10479-023-05720-9 (2023).
42. Zhao, X. et al. Composite Hedges Nanopores codec system for rapid and portable DNA data readout with high INDEL-Correction. *Nat. Commun.* **15**, 9395 (2024).
43. Beslic, D. et al. End-to-end simulation of nanopore sequencing signals with feed-forward transformers. *Bioinformatics* **41**, btae744 (2025).
44. Geifman, Y. & El-Yaniv, R. SelectiveNet: a deep neural network with an integrated reject option. *Proc. 36th Int. Conf. Mach. Learn.* **97**, 2151–2159 (2019).


## Acknowledgements

This work was supported by the National Key Research and Development Program of China (no. 2022YFF1203400), National Natural Science Foundation of China (no. 62571226, 62171211, 32371526, 32100021 and 32371372), Science and Technology Innovation Commission of Shenzhen (JCYJ20220814170440001, JCYJ20220818100218039, JCYJ20220530113013030 and JCYJ20230807092459028), NSQKJJ under grant K21799109 and K21799116, and Zhejiang Provincial Collaborative Innovation Center for High-end Digital Intelligence Diagnosis and Treatment Equipment and Center for Computational Science and Engineering at Southern University of Science and Technology. We acknowledge Mr. Qiang Ji's preliminary in silico trials of this project and his testing on Oxford Nanopore MinION sequencer.

## Author contributions

J.Y.L. and Y.L. conceived and designed the experimental procedures. J.Y.L. and Y.L. performed the SUSTag design. R.H.L., Y.P.L., and J.Y.L.

designed the primers for the mixed barcode DNA sequences. R.H.L., Y.P.L., and J.Y.L. also designed the primers for the DNA storage sequences. X.Y.Z. and Q.Y.F. developed the decoding algorithm for the DNA storage data. X.Y.Z., Q.Y.F., and J.Y.L. performed nanopore base-calling and associated quality control. J.Y.L. constructed the datasets, trained the deep learning models and conducted real-time adaptive sequencing experiments. J.Y.L., Q.P. and Y.L. prepared the figures and tables. J.Y.L., Q.P., and Y.L. drafted the manuscript. Y.L. and J.X.Z. supervised the study. All authors read, revised and approved the final manuscript.

## Competing interests
The authors declare no competing interests.

## Additional information
**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-64293-2.

**Correspondence** and requests for materials should be addressed to Yi Li.

**Peer review information** *Nature Communications* thanks Jake Smith and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.