Robust Multi-read Reconstruction from Noisy Clusters Using Deep Neural Network for DNA Storage

Yun Qin[a], Fei Zhu[a,*], Bo Xi[a], Lifu Song[b,c]

[a]*Center for Applied Mathematics, Tianjin University, Tianjin, China.*
[b]*Systems Biology Center, Key Laboratory of Engineering Biology for Low-carbon Manufacturing, Tianjin Institute of Industrial Biotechnology, Chinese Academy of Sciences, Tianjin, China.*
[c]*Haihe Laboratory of Synthetic Biology, Tianjin, China*

Abstract

DNA holds immense potential as an emerging data storage medium. However, the recovery of information in DNA storage systems faces challenges posed by various errors, including IDS errors, strand breaks, and rearrangements, inevitably introduced during synthesis, amplification, sequencing, and storage processes. Sequence reconstruction, crucial for decoding, involves inferring the DNA reference from a cluster of erroneous copies. While most methods assume equal contributions from all reads within a cluster as noisy copies of the same reference, they often overlook the existence of contaminated sequences caused by DNA breaks, rearrangements, or mis-clustering reads. To address this issue, we propose RobuSeqNet, a robust multi-read reconstruction neural network specifically designed to robustly reconstruct multiple reads, accommodating noisy clusters with strand breakage, rearrangements, and mis-clustered strands. Leveraging the attention mechanism and an elaborate network design, RobuSeqNet exhibits resilience to highly-noisy clusters and effectively deals with in-strand IDS errors. The effectiveness and robustness of the proposed method are validated on three representative next-generation sequencing datasets. Results demonstrate that RobuSeqNet maintains high sequence reconstruction success rates of 99.74%, 99.58%, and 96.44% across three datasets, even in the presence of noisy clusters containing up to 20% contaminated sequences, outperforming known sequence

*Corresponding author.
Email address: fei.zhu@tju.edu.cn (Fei Zhu)

reconstruction models. Additionally, in scenarios without contaminated sequences, it exhibits comparable performance to existing models, achieving success rates of 99.88%, 99.82%, and 97.68% across the three datasets.

Keywords:
DNA storage, Sequence reconstruction, Robust method, Attention mechanism, Deep neural network.

---

1. Introduction

In the contemporary era, the proliferation of information has resulted in the creation of vast datasets, presenting formidable challenges to conventional storage systems like mobile hard disks, USB flash memory, and integrated circuits. The utilization of these storage media gives rise to inherent issues, encompassing inadequate storage longevity, elevated energy consumption, and environmental pollution [1]. Meanwhile, the Deoxyribonucleic Acid (DNA) molecule has emerged as a promising storage medium, attributed to its theoretically high storage density and prolonged storage term, aligning with the necessity of accommodating vast amounts of data [2, 3, 4]. As illustrated in Figure 1, the workflow of a DNA storage system mainly consists of five stages: encoding, synthesis, storage, sequencing, and decoding.

In the context of DNA storage, the process typically involves initially encoding a binary stream into alphabet strings {A, T, C, G}, followed by the chemical synthesis of short DNA oligos, referred to as *references*. Subsequently, these references are then stored either *in vitro* or *in vivo*. To read the information using next-generation sequencing, the references should be extracted from a large, unordered collection of error-prone *reads*. This necessity arises due to the inherent introduction of insertion-deletion-substitution (IDS) errors during both DNA strand synthesis and sequencing in DNA storage. The error rate ranges from 1%-2% in mainstream next-generation sequencing to up to 10% for Nanopore sequencers [5]. During sequencing by Polymerase Chain Reaction (PCR), each reference outputs an uncertain number of noisy copies, and the reads corresponding to different references are gathered without ordering [2, 6]. In the decoding process, *clustering* is typically employed on the sequencing file to group noisy reads originating from the same reference into clusters [7]. Thereafter, the focus of this paper, *multi-read reconstruction*, is conducted to deduce the original reference from a cluster of noisy reads [8].

During the past ten years, a lot of research has been devoted to the sequence reconstruction problem in DNA storage. Broadly, these endeavors fall into four categories: the consensus methods of Bitwise Majority Alignment (BMA) [9, 10, 11, 8], the statistical inference methods [12, 13, 14], the probability backtracking methods [15, 16], and the recent deep learning ones [17, 18, 19]. The BMA and its variations are elaborated for IDS channels and applied to DNA storage systems in [9, 10, 8]. They perform position-to-position alignment among multiple reads and implement a majority voting strategy. BMA-based techniques prove to be effective, particularly for datasets characterized by low IDS error rates. However, their performance is found to be less satisfactory when applied to datasets with higher error rates.

The second category is based on statistical inference, where at each position of the sequence, the maximum a posterior (MAP) probability of all the possible input symbols are estimated and compared [12, 13, 14]. In [12], marker codes are inserted into LDPC codes at fixed intervals for error correction, and the decoder is based on a forward and backward (FB) algorithm In [13], a drift vector is introduced to model the insertion/deletion errors in each received word, and a factor graph is derived for joint probability estimation. Concatenated codes are considered in [14], whose inner codes and channels are modeled as joint Hidden Markov Models (HMM) and the BCJR inference is derived. The so-called Trellis BMA marries BMA with BCJR decoding and achieves a linear complexity in the number of traces [20]. However, due to the computational overhead, the feasible reads number per cluster can hardly exceed ten when applying these methods in practical DNA storage systems.

The probability backtracking method aims to refine the decoding results by incorporating probability information and likelihood rules. Two prominent encoding-decoding methods in this domain are HEDGES [15] and SPIDER-WEB [16]. In HEDGES [15], the encoding is based on plaintext auto-key, hashing each bit with strand ID, bit index, and several previous bits to correct errors. The corresponding decoding performs a greedy search on an expanding hypothesis tree and finally backtracks and outputs the best hypothesis. SPIDER-WEB [16] initially encodes DNA sequences based on a graph-based encoding technique. Following decoding, it employs a path-based error correction to rectify the DNA sequences. However, the limitation is that decoding can only be compatible with the associated encoding scheme.

With the emergence of deep learning, a few lately works have attempted

to exploit deep neural networks (DNN) to address the multi-read reconstruction [17], as well as single read reconstruction [18] in DNA storage systems. Similar in spirit of this work, the central concept involves training a DNN model with robust error correction capabilities, enabling it to map a cluster of noisy reads to the corresponding DNA reference. Given that this work is specifically focused on multi-read reconstruction using DNN, relevant studies [17, 18, 19] will be introduced in detail in Section 2.

It is noteworthy that in all the aforementioned methods, each strand within a cluster contributes equally to the reconstruction of the reference strand. This assumption is reasonable only when dealing with the clusters consisting of noisy copies originating from the same reference. However, achieving perfect clustering is not always feasible due to the inherent characteristics of current DNA storage systems. On one hand, as sequencing exhibit bias towards strands with specific properties, existing perfect clustering methods (*e.g.*, [7, 21, 22]) have the risk of losing references rarely sequenced [17, 23], potentially leading to inaccurate cluster assignments. On the other hand, in cases where the sequencing file includes contaminated reads resulting from DNA breaks and rearrangements, as detailed subsequently, clustering algorithms struggle to form clusters that align appropriately with the underlying DNA references.

In practical DNA storage, system stability and robustness are threatened by the presence of contaminated sequences that arise at various stages. Unlike noisy reads, which deviate from their references by only a few IDS errors, the *contaminated sequences*, as referred to in this paper, exhibit a considerably greater edit distance from the original DNA references. Several factors contribute to the occurrence of contaminated sequences. Firstly, in long-term storage and under certain conditions, DNA strands become vulnerable to degradation, resulting in strand breaks and loss [23]. In addition, unspecific amplification inevitably causes frequent DNA breaks and rearrangements, with oligos are fragmented and rejoined to new ones, a phenomenon investigated by one of the authors (Song *et* al. in [24]), as shown in Figure 2. Furthermore, contaminated sequences encompass not only various forms of damage but also the complementary strands of the references produced during sequencing [25]. Lastly, considering the security issues in DNA storage, intentionally introduced contaminated DNA molecules carrying false information are utilized for data encryption in studies such as [26, 27, 28]. Clearly, the presence of contaminated sequences further complicates the already challenging reconstruction problem [23, 24].

This paper proposes RobuSeqNet, a robust multi-read reconstruction method based on DNN by differentiating the sequence quality and reliability within the cluster, in the context of sequence reconstruction for DNA storage. To our knowledge, no method currently exists to differentiate sequence quality and reliability within clusters, specifically within the context of the DNA storage sequence reconstruction problem. This represents the primary advantage of our proposed method over existing approaches. Note that the proposed method differs from encoding-decoding systems like HEDGES [15] and SPIDER-WEB [16] as it is a general approach to multi-sequence reconstruction, independent of encoding methods. To ensure resilience to noisy clusters with contaminated sequences, including DNA breaks, rearrangements, and noisy reads with IDS errors, our proposed model strategically utilizes the attention mechanism and the Conformer block. The main contributions are as follows:

- **Integration of sequence quality through attention mechanism.** This innovative approach to multi-read reconstruction prioritizes sequence reliability within the cluster. Leveraging an attention module, each strand is scored based on its sequence quality, enabling varying degrees of contribution to the reconstruction process. This dynamic allocation of attention allows for effective suppression of the influence of diverse types of contaminated sequences.

- **Error correction capacity of IDS errors within clusters.** The proposed model realizes the error correction of IDS errors within the cluster. The Conformer-Encoder demonstrates robust feature extraction capabilities, intelligently integrating local features from the convolutional layers and global features from the attention module. The resulting features are high-level and representative, such that the underlying reference of the noisy cluster can be well recovered by a single-layer long short-term memory (LSTM) decoder.

- **Sequence reconstruction network accommodating varying cluster sizes.** The network is trained directly from clusters of different sizes, rather than summing up the reads within a cluster to form a structured input format [17]. Thereby, it is compatible with the input cluster of varying sizes at the testing stage.

- **Network with less parameters.** The proposed neural network has a small structure ($\approx$ 2.5 M parameters) with good generalization ability.

This helps to mitigate the overfitting issue caused by the shortage of training data, when using DNN to address the sequence reconstruction problem in DNA storage.

Figure 1: Overview of the DNA storage system.

Figure 2: Illustration of strands breaks and rearrangements in DNA data storage [24] and the proposed RobuSeqNet for dealing with them in sequence reconstruction.

The rest of this paper is organized as follows. The related work is reviewed in Section 2. In Section 3, we present the proposed multi-read reconstruction model. Experimental results and analysis are given in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

We succinctly review several deep learning-based sequence reconstruction methods in DNA storage. The most relevant literature to this paper is the so-called DNAformer, a scalable and robust solution for the DNA sequence reconstruction recently proposed in [17]. The model is based on DNN, and is well adapted to imperfect but fast clustering of copies. Leveraging convolution, Xception, and Transformer, the model demonstrates a good ability to correct IDS errors, particularly substitutions, within the cluster. Despite sharing dissimilar network designs, our approach distinguishes itself from DNAformer in the following aspects.

1. The input of DNAformer is formed by the element-wise sum of multiple copies, implying equal importance of each sequence within a cluster to

the reconstruction. However, it neglects to account for differences in sequence quality arising from the presence of contaminated sequences. On the contrary, our method scores every sequence within the cluster, and accordingly, the strands contribute to the reconstruction at different levels.

2. To overcome the shortage of training data, DNAformer applies the Synthetic Data Generator (SDG) [29] to generate sufficient DNA sequences for training the model, with the sequence error rates estimated by SOLQC [30]. Alternatively, our method circumvents this issue by designing a small but efficient network, which can be trained with much fewer labeled samples.

Nahum *et al.* [18] developed a single-read reconstruction model for DNA-based storage systems with the goal of comprehending error patterns from only a single sequence through a global, context-aware approach. The model uses an encoder-decoder Transformer architecture composed of two paired BERT models. The error correction is regarded as a self-supervised sequence-to-sequence task, and the network is trained using synthetic sequences generated by SDG [29].

In the context of Nanopore sequencing data recovery, Lv *et al.* [19] introduced an integration method that combines Viterbi error correction with a recurrent neural network (RNN). This approach involves reconstructing the reference directly from multiple records of the raw signal, bypassing the inference from highly noisy basecalled reads. Notably, this innovative strategy achieves a threefold reduction in reading cost when compared to prior methodologies.

3. The Proposed Model

The multi-read reconstruction problem in DNA storage presents significant challenges due to the complexity of decoding multiple erroneous sequences accurately, especially in the presence of DNA breakage and rearrangements. Overcoming this challenge is crucial for maximizing data storage capacity and ensuring reliable retrieval in DNA-based storage systems, thus paving the way for scalable and durable long-term data storage. In this section, we formulate the multi-read reconstruction problem mathematically and then describe the proposed robust multi-read reconstruction network (RobuSeqNet) in detail.

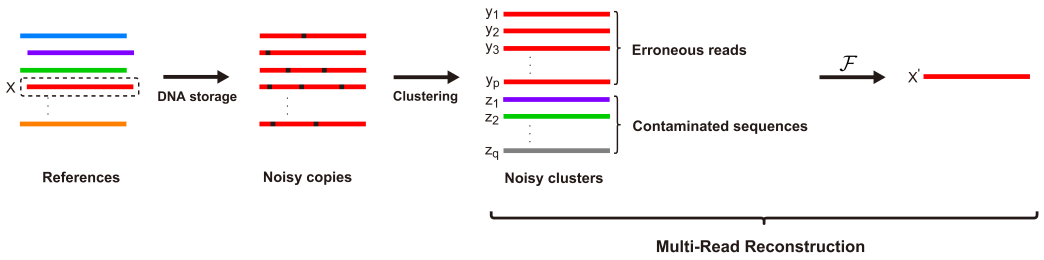


Figure 3: Illustration of the multi-read reconstruction problem defined in this paper. Binary files are encoded as DNA references. Multi-read reconstruction starts from a noisy cluster containing the erroneous copies originating from the original reference and the contaminated sequences occurring at different stages of DNA storage. The proposed reconstruction method aims at finding a mapping (characterized by a neural network), that minimizes the distance between the cluster and the original reference.

3.1. Problem Statement

Use $\Sigma = \{A, C, G, T\}$ to represent four DNA nucleotides. Let $\mathcal{C} \in (\Sigma^*)^N$ be a noisy cluster, which contains p erroneous reads $y_1, y_2, ..., y_p$ originating from the same reference $x \in \Sigma^L$, and q contaminated sequences $z_1, z_2, ..., z_q$ introduced at various stages of the DNA storage process, with $N = p + q$. Based on this assumption, the DNA multi-read reconstruction algorithm is a mapping:

$$\mathcal{F} : \mathcal{C} \rightarrow \Sigma^L,$$

which receives N sequences and outputs $\hat{x}$, an estimate of x, as shown in Figure 3. In this work, we focus on deploying a DNN model to find such a mapping $\mathcal{F}$ that the distance between $\hat{x}$ and x can be minimized.

3.2. Model Overview

Our objective is to address the sequence reconstruction problem as defined in Section 3.1 through the application of deep learning techniques. The proposed RobuSeqNet, illustrated in Figure 4, is built upon an encoder-decoder architecture and consists of three key components: the Attention Module, Conformer-Encoder, and LSTM-Decoder. Located at the forefront of our model, the Attention Module evaluates the quality of each sequence within the input cluster, generating a high-level feature weighted by an average score. This implementation incorporates an attention mechanism [31] to address the impact of contaminated sequences, dynamically allocating weights

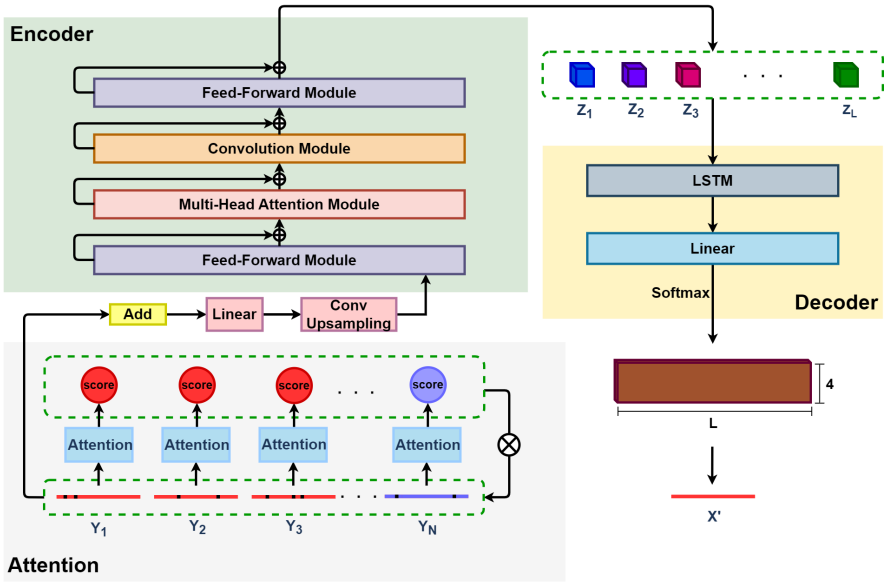


Figure 4: Model architecture. The proposed RobuSeqNet is composed of Attention Module, Conformer-Encoder, and LSTM-Decoder, which correspond to the three colored regions in the figure, respectively.

to different segments of the input. This non-uniform weighting strategy empowers the model to selectively prioritize the most relevant information, enhancing its adaptability and effectiveness for the given task. The Conformer-Encoder is expected to understand the IDS error patterns within a cluster, taking into account its powerful feature extraction ability. It interactively combines the local features extracted by the convolution with the global features generated by the attention module. The decoder is a single-layer LSTM, which outputs the predicted reference of the input cluster. Next, we present the sequence embedding and three model components in detail, as well as the loss function.

3.2.1. Sequence Embedding

The model input is a cluster of a non-fixed number of DNA sequences with varying lengths. Before being fed to the network, each sequence is represented by the one-hot encoding to a prefixed, uniform length L, where zeros are padded to short strands. Here, L represents the maximum sequence length in the dataset. In this way, every sequence is converted to a matrix of size $4 \times L$, each column being a one-hot vector indicating the corresponding base at that index position.

3.2.2. Attention Module

As illustrated in Figure 5, the attention module consists of the convolutional layer followed by an attention mechanism [31]. For every strand feature, we perform two successive 1D convolution operations with kernel sizes of 3 and 5 to model the position shifts from synchronization errors, while reducing the number of feature channels from 4 to 2 and finally to 1. The resulting one-dimensional vectors are scored by the attention mechanism in a similar way as in [32].

Let $y_i \in \mathcal{R}^{L\times 4}$ be the input feature of the i-th strand in cluster, and $\widetilde{y}_i \in \mathcal{R}^L$ be the corresponding vector after convolution. The attention mechanism is applied as

$$e_i = v^T f(W\widetilde{y}_i + b) + k. \tag{1}$$

Here, the linear transformation with parameters W and b serves to project the vector into a lower-dimensional space, thereby reducing the parameter count of the network. The nonlinear activation function f introduces nonlinearity to the model, allowing it to capture complex relationships within the data. Additionally, v and k are learnable parameters employed for scaling and shifting the attention scores, respectively, enabling the model to dynamically adjust the importance of each sequence within the cluster during the attention calculation process.

After the nonlinear activation layer f, the feature is transformed to a sequence-wise attention score e_i via a linear layer (parameterized by v and k). By applying the softmax function, the scalar e_i is normalized over all the strands within the cluster as

$$\alpha_i = \frac{\exp(e_i)}{\sum_{i=1}^{N}\exp(e_i)}, \tag{2}$$

where N is the cluster size, and α_i is the final attention score of the i-th sequence. Obviously, the attention score reflects the importance of each strand within the cluster. As a result, the weight-averaged feature for the given cluster becomes

$$y = \sum_{i=1}^{N}\alpha_i y_i. \tag{3}$$

Here, every sequence contributes to the representation differently according to sequence quality, with the importance of high-quality reads amplified and the effect of low-scored strands suppressed automatically. After the attention

module, the linear layer and convolution upsampling is applied to represent the feature (3) in an enlarged feature space. As a 4-dimensional representation is not enough to characterize a position in the sequence, we expand the feature dimension from $L \times 4$ to $L \times 128$ in the experiments.

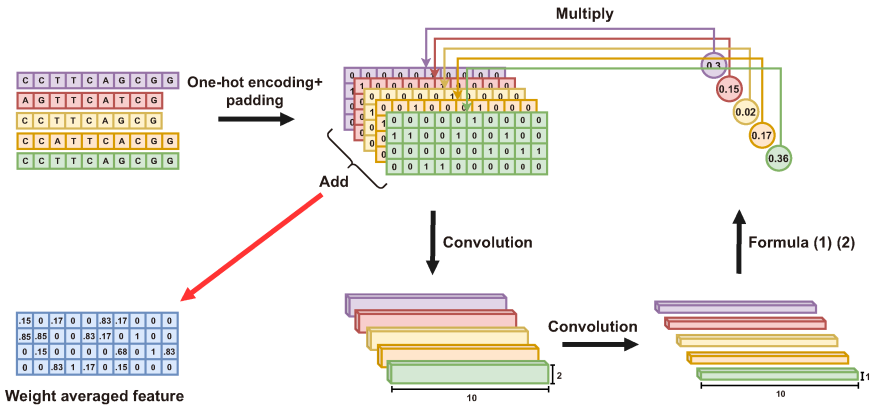


Figure 5: Attention Module. Each noisy copy is converted to a two-dimensional matrix by one-hot encoding and zero-padding. After convolution, each feature is transformed into a one-dimensional vector, and is fed to the attention mechanism to estimate a scalar score. Finally, the weight-averaged feature for the given cluster is generated.

3.2.3. Conformer-Encoder

Concerning the encoder, we adopt the convolution-augmented Transformer (Conformer), which is proposed for speech recognition and outperformed the CNN and Transformer-based models with SOTA results [33]. The Conformer combines convolution and self-attention to effectively capture both local features and global dependencies in sequential data. The convolutional module learns relative-offset-based local interactions caused by deletion or insertion errors. The utilization of self-attention layers is intended to address the global dependencies within sequences, dynamically focusing on different positions to better model long-term dependencies in the sequence. As shown in Figure 4, the Conformer-Encoder consists of multi-head self-attention layers and convolution layers sandwiched between two feed-forward modules with shortcut connections, where layer normalization is always applied at the junction of two modules.

The multi-head attention module (MHSA) [31] is computed by scaled-dot

product with

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^{\top}}{\sqrt{d_k}})V, \tag{4}$$

where Q, K, V denote the input matrix and d_k is the scaling factor that equal to the dimension of Q and K.

In this work, we employ $h = 8$ parallel attention heads, namely the concatenation of h scaled-dot product attention results, yielding

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_\text{h})W, \tag{5}$$

$$\text{head}_\text{i} = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V), \tag{6}$$

where $X \in \mathcal{R}^{L\times d}$ is the input to MHSA, $W_i^Q, W_i^K, W_i^V \in \mathcal{R}^{d\times d_k}$. $W \in \mathcal{R}^{hd_k\times d}$ maps the concatenated feature back to the original dimension d. In practice, we have $d = 128$ and $d_k = d/h = 16$.

As for the convolution module (Conv), we perform two pointwise convolutions and a 1-D depthwise convolution with kernal size 32 to capture local correlations among sequence positions. Each feed-forward module (FFN) has two linear layers, which firstly double and then restore the original feature dimension.

Mathematically, for x input to Conformer-Encoder, the output y is:

$$x^{'} = x + \frac{1}{2}\text{FFN}(x), \tag{7}$$

$$x^{''} = x^{'} + \text{MHSA}(x^{'}), \tag{8}$$

$$x^{'''} = x^{''} + \text{Conv}(x^{''}), \tag{9}$$

$$y = x^{'''} + \frac{1}{2}\text{FFN}(x^{'''}). \tag{10}$$

3.2.4. LSTM-Decoder

As an advanced variant of RNN, LSTM can model long-range dependencies well for chronological data [34]. Due to processes such as encoding, synthesis, and sequencing, sequences in DNA storage commonly exhibit a complex long-range dependency structure. The memory units and gating mechanisms of LSTM enable it to more effectively capture and leverage these long-term dependencies, thereby enhancing its modeling capability for DNA

sequences. In practice, we employ a single-layer LSTM decoder. Although simple, it is sufficient for the reconstruction task, owing to the powerful feature extraction ability of the Conformer-Encoder. The decoder reduces the feature dimension back to 4, outputting for each position the estimated probabilities for each base.

The proposed RobuSeqNet model is trained using a cross-entropy loss function defined as

$$\mathcal{L} = -\sum_{l=1}^{L} x_l \log f(y_l), \tag{11}$$

where L is the sequence length, y_l is one-hot label vector indicating the base category for the l-th position, and $f(x_l)$ represents the predicted probability vector by the proposed neural network. The cross-entropy loss function is widely used in classification tasks. During the evaluation of the model's performance on the testing set, our primary metric of assessment was the success rate, as defined in Equation (12).

Table 1: Data description.

Dataset	Erlich *et al.*[35]	Organick *et al.*[36]	Chandak *et al.*[37]
Num. of reference	72000	607150	11710
Sequence length	152	150	150
Synthesis	Twist Bioscience	Twist Bioscience	CustomArray
Sequencing	Ilumina miSeq	Ilumina NextSeq	Illumina iSeq
Num. of reference aligned to reads	72000	596669	11710
Missing clusters	0	10481	0
Num. of reads aligned to reference	13328870	14486345	1065117

Table 2: Statistics of the training and testing set.

	Dataset	Erlich *et al.* [35]	Organick *et al.* [36]	Chandak *et al.* [37]
Training	**Cluster number**	36000	296317	5857
	Cluster size	5-30	5-30	5-30
	Num. of reads	628875	5587728	101643
Testing	**Cluster number**	36000	296325	5853
	Cluster size	5-30	5-30	5-30
	Num. of reads	630945	5586351	102744

4. Experimental Results

4.1. Data Preparation and Training details

We use three well-known datasets for DNA-based storage provided in Erlich et al.[35], Organick et al.[36], and Chandak et al.[37]. Dataset descriptions are given in Table 1. These three datasets exhibit distinct data scales and varying error rates, thereby serving as representative instances of complex and diverse DNA storage scenarios in reality. Each dataset comprises two key files for sequence reconstruction, one containing the disordered collection of the noisy reads and the other recording all the original references. As no ground-truth clusters are available in practice, we first apply Burrows-Wheeler-Alignment Tool (BWA) [38] on both files, and take the sequence alignment results as perfect clusters, where each read is matched to its closest reference.

In the experiments, we set the cluster scale to be 5$\sim$30, a modest range for the sequence reconstruction task, by randomly picking reads from each previously-obtained cluster. This approach is justified by the large number of reads in the original sequencing files, leading to the typically substantial size of clusters and considerable information redundancy (e.g., the average number of copies per reference is approximately 185 in dataset Erlich *et al.*[35], Table 1).

To further intensify the challenge of multi-read reconstruction and simulate scenarios involving contamination sequences, we introduce a certain proportion of such sequences into each cluster. These contaminated sequences are produced from: 1) DNA fragments: Simulating DNA breakages and rearrangements encountered in DNA storage and PCR amplification-based DNA strand replication, as analyzed in [24]; 2) Misclustered reads: Occurring when clustering is imperfect and sequences are assigned to the wrong cluster; 3) Reverse complementary strands: As sequencing process generates reverse complementary sequences for DNA, strands in opposite orders may end up in the same cluster [25]; 4) Random sequences: Introduced to simulate intentionally added false information in [26]. To demonstrate the effectiveness of the proposed model under different contamination levels, we inject contaminated sequences into each cluster, with equal probability for every candidate reason. The contamination level is defined as the proportion of injected contaminated sequences relative to the cluster size, calculated by dividing the number of contaminated sequences by the total number of sequences in a cluster. It is distinguished from the error rate, which is specific to the DNA storage dataset and is typically quantified as the IDS error rate estimated using tools such as SOLQC [30].

More precisely, on each of the three datasets, five simulations with contamination levels ranging within the set $\{0\%, 5\%, 10\%, 15\%, 20\%\}$ are performed by using the proposed method as well as the comparative sequence reconstruction approaches. Here, 0% denotes the case without extra added contaminated sequences, i.e., the clusters are composed of the reads from the original sequencing file.

Table 2 reports the training and testing set on three datasets. For each experiment, the proportion of training data to test data is set to 1:1. In the training phase, the dataset is initially sorted based on the cluster sizes. Clusters with equal size are grouped, forming mini batches for training. Shuffling is performed both between and within groups at the beginning of each epoch to ensure the randomness of the training dataset. This approach enables our network to handle clusters of different sizes effectively. The training and testing are performed on a single NVIDIA GeForce RTX 2080 Ti GPU. We set the batch size to 64 and the initial learning rate to 0.005. The Adam optimizer is applied with parameter values $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The coefficient of L_2 regularization is chosen as $1e-4$ to prevent overfitting.

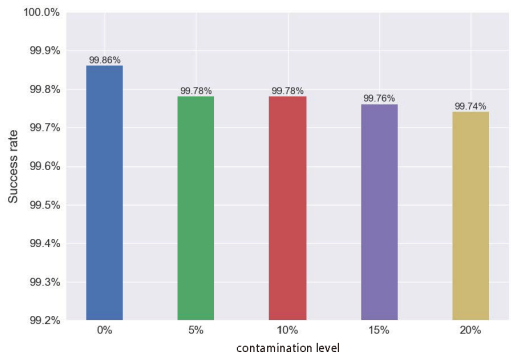
4.2. Evaluation metric and Comparative methods

The multi-read reconstruction performance is evaluated by the success rate, defined by

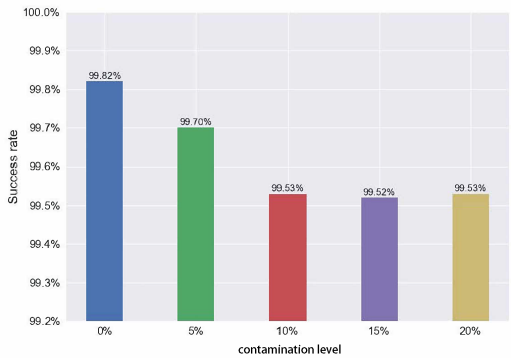
$$\text{success rate} = \frac{\#\{ \text{ predicted } \ \text{sequence} = \text{input } \ \text{reference } \}}{\#\{ \text{ input } \ \text{reference } \}}. \tag{12}$$

In this formula, a sequence will contribute to the success rate only if it is perfectly reconstructed without error at every index position. We apply the success rate as it is the most widely used in DNA sequence reconstruction tasks [17, 18]. This indicator is more stringent compared to other distance-based metrics, such as edit error rate or Hamming error rate, as only perfectly reconstructed sequences are counted. In real DNA storage scenarios, successfully reconstructed sequences convey specific information fragments. Therefore, the success rate effectively reflects the overall effectiveness of sequence reconstruction for a dataset, providing a stringent evaluation criterion.

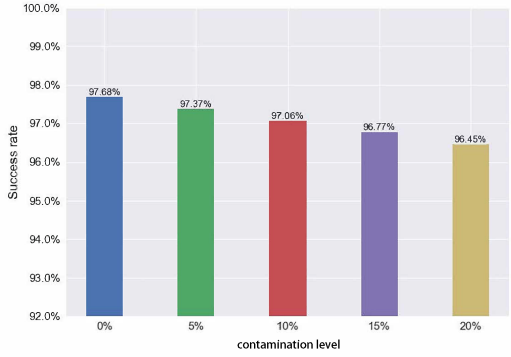
The effectiveness of the proposed method is assessed by benchmarking it against three SOTA sequence reconstruction methods. We specifically chose these comparison methods because they are recognized as SOTA solutions in the multi-read sequence reconstruction domain. Notably, like our proposed



(a) Erlich *et al.*[35]

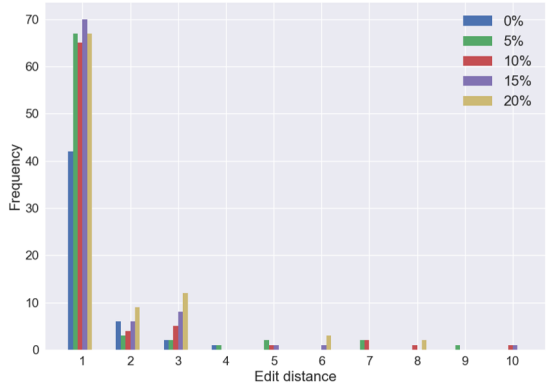


(b) Organick *et al.*[36]

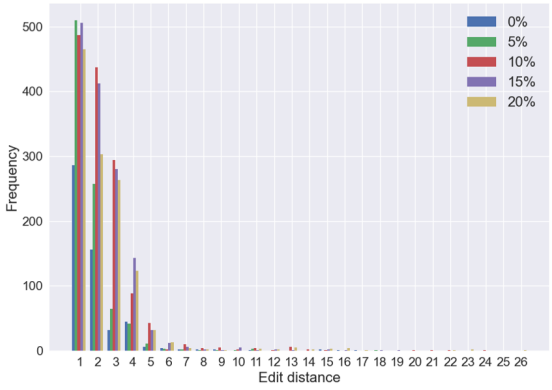


(c) Chandak *et al.*[37]

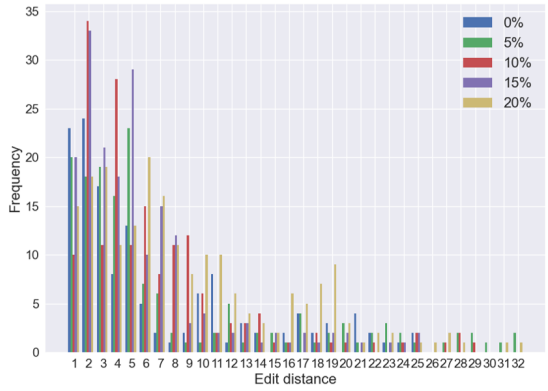
Figure 6: Changes of reconstruction success with respect to the contamination level ranging from 0% to 20% using the proposed RobuSeqNet, on three datasets.



(a) Erlich *et al.*[35]



(b) Organick *et al.*[36]



(c) Chandak *et al.*[37]

Figure 7: Frequency histograms of the edit distance measured between the wrong prediction and the corresponding cluster reference.

approach, these methods are designed to operate independently of the encoding scheme applied. This characteristic allows them to be directly applied to noisy read clusters generated from the sequencing file without considering the specific encoding method used in the dataset.

- **Iterative Reconstruction** [8]: This algorithm uses multiple methods to revise strands from clusters and return the candidate sequence most likely to be the original reference. The error vectors majority algorithm is used to correct insertion and substitution errors, while the pattern-path algorithm is applied to correct deletion errors.

- **BMA divider** [8]: This BMA-based algorithm divides the received clusters into three sub-clusters by their lengths. The majority voting is applied to the sequences of the correct length. Then deletion and insertion error corrections are performed on the sub-clusters with shorter and larger sequence lengths, respectively.

- **BMA Lookahead** [9]: This is an improved algorithm of the BMA method. For sequences whose current symbol does not match the majority of symbols, a "prior window" looking at the next two (or more) symbols is used.

4.3. Results anaysis

We report the reconstruction success rates of the proposed RobuSeqNet at different contamination levels $\{0\%, 5\%, 10\%, 15\%, 20\%\}$ on the testing set for all the three datasets, as shown in Figure 6. The horizontal axis of Figure 6 represents the contamination level and the vertical axis represents the success rate. For Erlich *et al.*[35], the success rates reach 99.86%, 99.78%, 99.78%, 99.76%, and 99.74%, corresponding to 51, 78, 79, 87, and 93 wrong predictions out of 36000 clusters. The success rates are 99.82%, 99.70%, 99.53%, 99.52%, and 99.58% with 540, 897, 1391, 1408, and 1230 wrong predictions out of 296325 clusters for Organick *et al.* [36]. On the third dataset Chandak *et al.*[37], the numbers are 97.68%, 97.37%, 97.06%, 96.77%, and 96.45% with 136, 154, 172, 189, and 208 wrong predictions out of 5853 testing clusters. On all three datasets, the performance of RobuSeqNet remains stable with only a slight decrease in success rate, even when the proportion of contaminated strands reaches 20%. Notice that the success rates are relatively low on the third dataset at all the contamination levels. It is due to the higher IDS error rates, as well as the mismatch in sequence lengths.

These results have significant implications for real-world DNA storage systems. They demonstrate the robustness and stability of the proposed RobuSeqNet model across varying levels of contamination. This resilience is crucial for real-world DNA storage systems, where challenges like IDS errors and contaminated sequences, including DNA breaks and rearrangements, are unavoidable due to factors such as synthesis errors and storage conditions. By enabling accurate information recovery, the resilience in multi-read reconstruction enhances the reliability of DNA storage systems.

4.3.1. Wrong prediction analysis

The frequency histograms of the edit distance, as shown in 7, illustrate the differences between the incorrectly predicted sequences and their corresponding cluster references. The analysis indicates that a majority of erroneous predictions have a relatively small edit distance to their original references. This observation suggests that the proposed model accurately predicts most sequence positions, even when the cluster reconstruction is not entirely perfect. Future modeling efforts should prioritize sequences that, despite not being successfully reconstructed, have a low edit distance from the reference sequence.

4.3.2. Inference time comparison

Once well-trained on a specified dataset, RobuSeqNet can be directly used for sequence reconstruction on that same dataset. Table 3 reports the inference time of our method compared to three other methods, with each batch containing 64 noisy clusters for reconstruction. While the inference time of RobuSeqNet is slightly higher than that of BMA divider [8] and BMA Lookahead [9], it is significantly lower than that of Iterative Reconstruction [8].

Table 3: Inference time comparison to SOTA methods

	RobuSeqNet	Iterative Reconstruction [8]	BMA divider [8]	BMA Lookahead [9]
Inference time	1s/batch	100s/batch	0.45s/batch	0.41s/batch

4.3.3. Impact of cluster size

We investigate how the smallest cluster size k in a dataset affects the reconstruction success rate, under varying contamination conditions. The

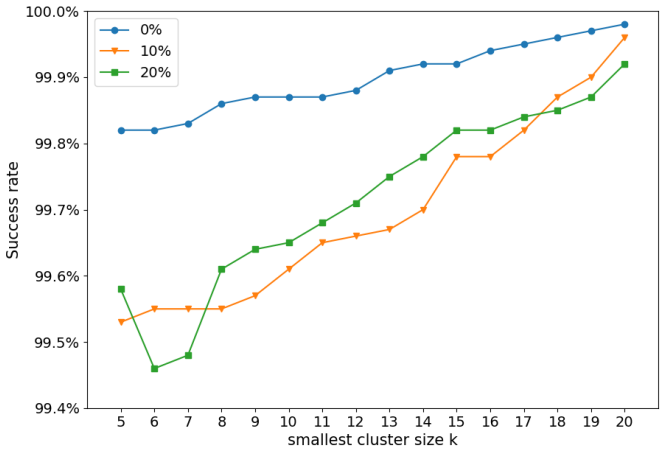


Figure 8: Changes of the success rate in terms of the smallest cluster size k, namely the cluster size of the dataset ranges from k to 30, on dataset Organick *et al.*[36].

results are given in Figure 8. As observed, for all contamination rates, the success rate increases with the increase of the smallest cluster size k in a dataset. This is attributed to the fact that the more sequences there are in a cluster, the greater the variety of error types the model can learn, thereby enhancing the modeling capability. With $k = 20$, the success rates achieves 99.98%, 99.96%, and 99.89% under the contamination levels 0%, 10% and 20%, respectively.

4.4. Comparative study

Figure 9 reports the success rate obtained on all three datasets at varying contamination levels, by using the proposed RobuSeqNet, and other three sequence reconstruction strategies.

It is worth noting that the proposed RobuSeqNet consistently outperforms the alternatives on the third dataset Chandak *et al.* [37], across all contamination levels. This is attributed to the dataset's characteristic of containing reads longer than the original reference, which leads to a higher IDS error rate. The Conformer-Encoder module in RobuSeqNet effectively captures these IDS error patterns, making it resilient to position shifts within the strands. Notably, the BMA divider [8] performs inadequately on this dataset.

In uncontaminated datasets, our method demonstrates comparable reconstruction results to SOTA methods. More precisely, on the first two datasets, the success rates by RobuSeqNet are slightly lower than that of Iterative Reconstruction [8] and BMA Lookahead [9], but slightly higher than in BMA divider [8].

The advantages of the proposed RobuSeqNet become increasingly evident as the proportion of contaminated sequences gradually augments in the dataset. For example, when the contamination proportion is increased to 10% in Erlich *et al.* [35] data, the performance decreases in RobuSeqNet, Iterative Reconstruction[8], BMA divider [8], and BMA Lookahead [9] are 0.1%, 0.15%, 0.37% and 0.22%, respectively. Compared to its counterparts, the proposed method is least affected by cluster contamination. With a 10% contamination rate, RobuSeqNet is second only to Iterative Reconstruction [8] by a marginal 0.06% on Erlich *et al.* [35] dataset and trails BMA Lookahead [9] by a similarly slight 0.07% on the second dataset. When the contamination proportion reaches 15%, RobuSeqNet surpasses all other methods in terms of success rates across all three datasets. With further increases in contamination, these advantages over other methods in terms of success rates become even more pronounced.

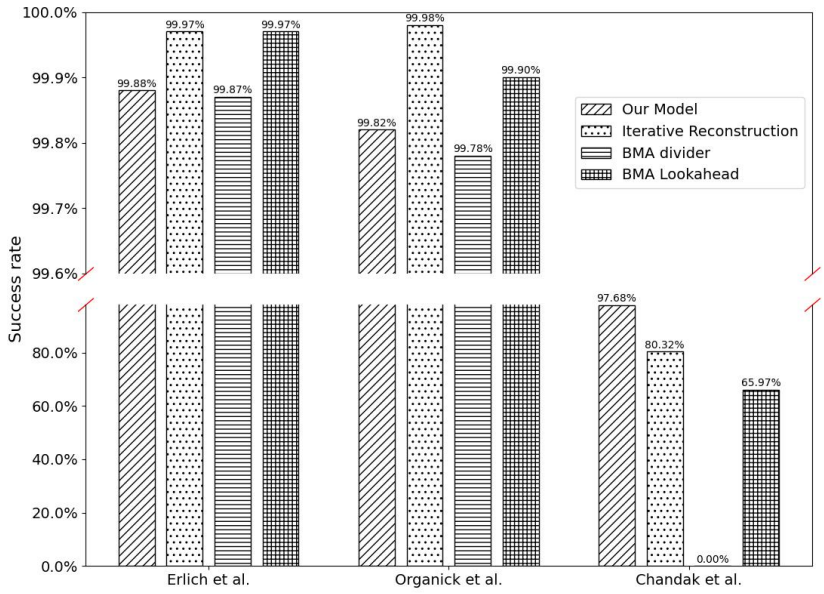
The results show RobuSeqNet's effectiveness in addressing various forms of contamination, such as strand breaks, rearrangements, and IDS errors within clusters, particularly in scenarios with high contamination levels. These findings imply that RobuSeqNet has the potential to enhance data retrieval and information recovery in real-world DNA storage scenarios by effectively tackling the challenges posed by contaminated sequences and IDS errors within clusters.

4.4.1. Robustness analysis

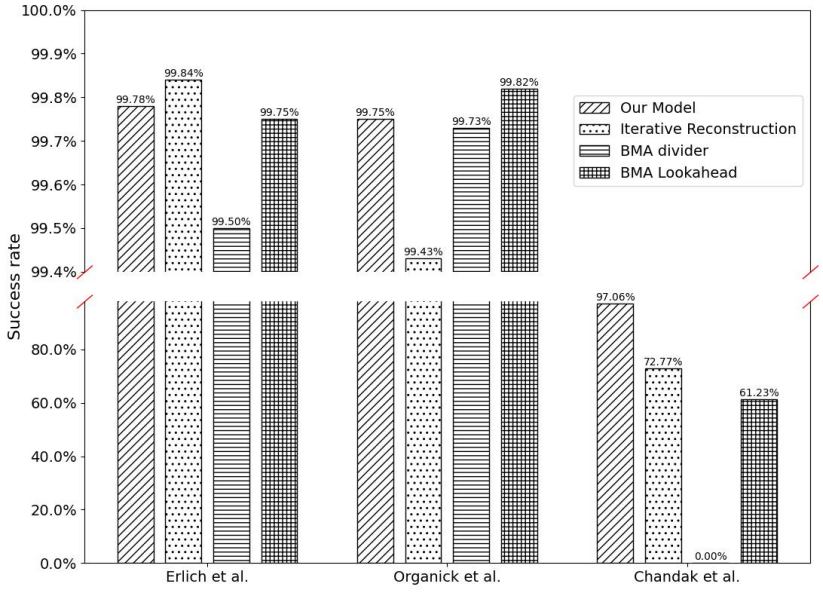
To assess the robustness of RobuSeqNet against three other models, we designed a composite score to measure the model's resilience to varying levels of contamination, primarily by referring to the formula for model robustness proposed in [39]. The specific formula we employed is

$$\text{Robustness Score} = \frac{1}{t}\sum_{s=1}^{t} RR_{s,c}^{f}, \tag{13}$$

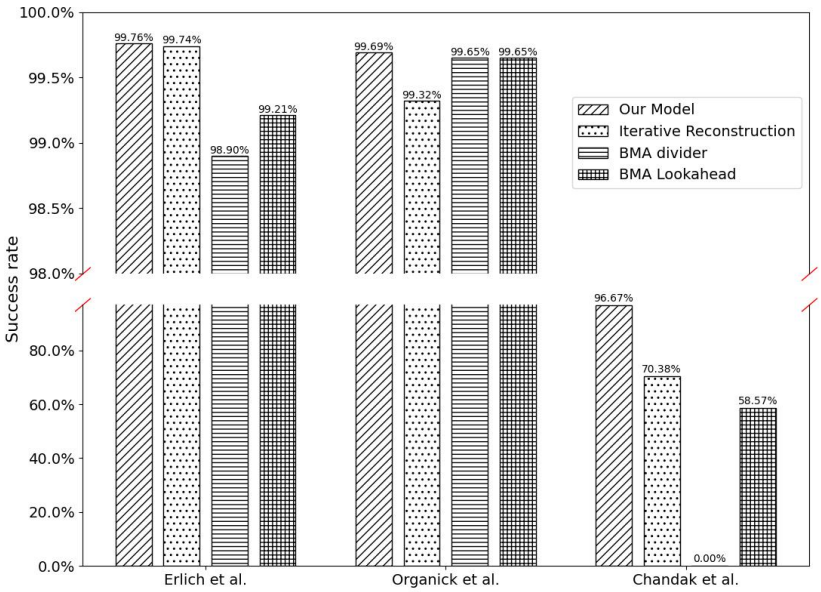
where f denotes the model, c indicates the dataset, s represents different contamination levels, and $RR_{s,c}^{f}$ refers to the reconstruction rate for a specific



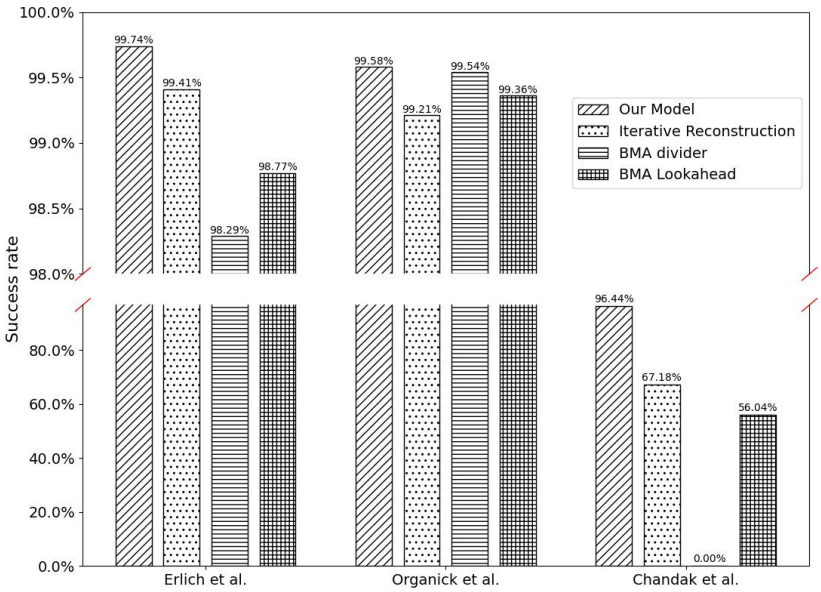
(a) Contamination level: 0%



(b) Contamination level: 10%



(c) Contamination level: 15%



(d) Contamination level: 20%

Figure 9: Comparison of success rate using the proposed RobuSeqNet, iterative reconstruction [8], BMA divider [8] and BMA Lookahead [9], on three datasets at contamination levels 0%, 10%, 15% and 20%.

contamination level, dataset, and model. It is noteworthy that the metric in [39] typically measures the average value of corruption errors of a model across different corruption levels. However, in our study, we adapted this metric to reflect the average reconstruction rate, aligning it more closely with the objectives of multi-sequence reconstruction in DNA storage.

The robustness score represents the average performance of each model across varying levels of contamination within individual datasets. As depicted in Table 4, these scores offer a holistic evaluation of each method's ability to handle contaminated sequences within specific datasets. Remarkably, our model consistently outperforms the other methods across all three datasets, indicating its superior resilience in real-world scenarios.

Table 4: Model robustness score.

Method	**Robustness score**		
	Erlich *et al.* [35]	**Organick *et al.* [36]**	**Chandak *et al.* [37]**
Our model	**99.79%**	**99.71%**	**96.96%**
Iterative Reconstruction	99.74%	99.49%	72.66%
BMA divider	99.14%	99.68%	-
BMA Lookahead	99.43%	99.68%	60.45%

4.5. Ablation Study

Table 5: Ablation Study.

Model	**Success rate**								
	Erlich *et al.* [35]			**Organick *et al.* [36]**			**Chandak *et al.* [37]**		
	0%	**10%**	**20%**	**0%**	**10%**	**20%**	**0%**	**10%**	**20%**
RobuSeqNet	**99.86%**	**99.78%**	**99.74%**	**99.82%**	**99.53%**	**99.58%**	**97.68%**	**97.06%**	**96.44%**
-Atten.	70.72%	63.76%	43.45%	80.65%	72.49%	57.53%	67.89%	65.98%	62.79%
-Atten.+Norm.	99.21%	98.56%	96.34%	99.81%	99.12%	97.51%	**97.68%**	95.33%	90.12%
-FFN-Conv	99.57%	99.32%	99.00%	99.15%	99.12%	98.47%	97.45%	96.78%	96.28%

The ablation study is designed to demonstrate the necessity of the Attention Module and the effectiveness of the Conformer-Encoder. To this end, we first remove the attention mechanism in attention module and directly feed the model with the summation of all the input sequences within a cluster. As shown in Table 5, the resulting model performs poorly on all datasets with varying contamination levels. This may be attributed to significant differences in the scales of the input data. The removal of the attention

mechanism is equivalent to omitting the normalization operation, leading to training instability and hindering the model's ability to learn meaningful representations.

We further impose an equal, normalized weight on every input strand. As seen from Figure 5, the resulting reconstruction performance is always inferior to our proposed model, especially when severe contamination is present in the dataset. The gaps in success rate between the two models are up to 3.4%, 2.07%, and 6.32% on three datasets when the contamination rate reaches 20%. In the case without additional contamination sequences, this model achieves similar results compared to our model.

Finally, we remove the feed-forward module and the convolution module from the Conformer block to simulate a Transformer-Encoder that relies entirely on the multi-head attention mechanism, while the encoder in Transformer [31] is composed of a multi-head attention module and a feed-forward module consisting of two linear layers, each performing residual concatenation and layer normalization. As shown in Table 5, the performance of the latter model is satisfactory but inferior to our model, demonstrating the effectiveness of the proposed Conformer-Encoder. When compared to the removal of other modules, the decrease in model performance was least noticeable when the feed-forward and convolution modules were removed. This suggests that in future model designs, alternatives for these specific components could be explored, considering their relatively minor influence during ablative analysis.

5. Conclusion

In this paper, we proposed a DNN-based multi-read reconstruction model for DNA storage, which is robust to noisy reads with IDS errors, and more importantly resilient to the contaminated sequences introduced during the DNA storage process. The proposed network has an encoder-decoder architecture with three pivotal components. The Attention Module suppresses the effect of contaminated sequences on the reconstruction, by automatically scoring the strands within the cluster and generating a representative, weight-averaged feature for subsequent tasks. The Conformer-Encoder has a sandwich structure and tackles most of the IDS errors within a cluster thanks to its advanced feature extraction capacity. The single-layer LSTM-decoder finally predicts the reference DNA of the input cluster. We prove the effectiveness and robustness of the proposed RobuSeqNet on three next-

generation sequencing datasets through a series of comparative experiments, where different levels of contamination caused by various factors during the process of DNA storage are simulated. The ablation study is also provided to verify the necessity of the attention mechanism and the Conformer block in the proposed model. Future works will focus on adapting the proposed sequence reconstruction model to the Nanopore sequencing data with higher error rates. On the other hand, due to the limited availability of real DNA storage data, transfer learning strategies and data augmentation can be employed to mitigate potential overfitting issues caused by the scarcity of training data for deep neural networks.

Code Availability

The proposed RobuSeqNet was implemented using Python and Pytorch. The source code is available at: `https://github.com/qinyunnn/RobuSeqNet`.

Funding

This work was supported by the National Key Research and Development Program of China (No. 2020YFA0712100) and by major program of Haihe Laboratory of Synthetic Biology (TSBICIP-CXRC-072).

References

[1] K. Goda, M. Kitsuregawa, The history of storage systems, Proceedings of the IEEE 100 (Special Centennial Issue) (2012) 1433–1440.

[2] V. Zhirnov, R. M. Zadegan, G. S. Sandhu, G. M. Church, W. L. Hughes, Nucleic acid memory, Nature materials 15 (4) (2016) 366–370.

[3] L. Ceze, J. Nivala, K. Strauss, Molecular digital data storage using dna, Nature Reviews Genetics 20 (8) (2019) 456–466.

[4] A. Rasool, J. Hong, Q. Jiang, H. Chen, Q. Qu, Bo-dna: Biologically optimized encoding model for a highly-reliable dna data storage, Computers in Biology and Medicine 165 (2023) 107404.

[5] Y. Dong, F. Sun, Z. Ping, Q. Ouyang, L. Qian, Dna storage: research landscape and future prospects, National Science Review 7 (6) (2020) 1092–1107.

[6] L. C. Meiser, P. L. Antkowiak, J. Koch, W. D. Chen, A. X. Kohll, W. J. Stark, R. Heckel, R. N. Grass, Reading and writing digital data in dna, Nature Protocols 15 (1) (2020) 86–101.

[7] C. Rashtchian, K. Makarychev, M. Racz, S. Ang, D. Jevdjic, S. Yekhanin, L. Ceze, K. Strauss, Clustering billions of reads for dna data storage, Advances in Neural Information Processing Systems 30 (2017).

[8] O. Sabary, A. Yucovich, G. Shapira, E. Yaakobi, Reconstruction algorithms for dna-storage systems, bioRxiv (2020).

[9] P. S. Gopalan, S. Yekhanin, S. D. Ang, N. Jojic, M. Racz, K. Strauss, L. Ceze, Trace reconstruction from noisy polynucleotide sequencer reads, uS Patent App. 15/536,115 (Jul. 26 2018).

[10] S. M. Yekhanin, M. Z. Racz, Trace reconstruction from reads with indeterminant errors, uS Patent App. 16/105,349 (Feb. 20 2020).

[11] S. R. Srinivasavaradhan, M. Du, S. Diggavi, C. Fragouli, Symbolwise map for multiple deletion channels, in: 2019 IEEE International Symposium on Information Theory (ISIT), IEEE, 2019, pp. 181–185.

[12] R. Shibata, G. Hosoya, H. Yashima, Fixed-symbols-based synchronization for insertion/deletion/substitution channels, in: 2016 International Symposium on Information Theory and Its Applications (ISITA), IEEE, 2016, pp. 686–690.

[13] R. Sakogawa, H. Kaneko, Symbolwise map estimation for multiple-trace insertion/deletion/substitution channels, in: 2020 IEEE International Symposium on Information Theory (ISIT), IEEE, 2020, pp. 781–785.

[14] A. Lenz, I. Maarouf, L. Welter, A. Wachter-Zeh, E. Rosnes, A. G. i Amat, Concatenated codes for recovery from multiple reads of dna sequences, in: 2020 IEEE Information Theory Workshop (ITW), IEEE, 2021, pp. 1–5.

[15] W. H. Press, J. A. Hawkins, S. K. Jones Jr, J. M. Schaub, I. J. Finkelstein, Hedges error-correcting code for dna storage corrects indels and allows sequence constraints, Proceedings of the National Academy of Sciences 117 (31) (2020) 18489–18496.

[16] H. Zhang, Z. Lan, W. Zhang, X. Xu, Z. Ping, Y. Zhang, Y. Shen, Spiderweb enables stable, repairable, and encryptible algorithms under arbitrary local biochemical constraints in dna-based storagearxiv preprint, arXiv preprint arXiv:2204.02855 (2022).

[17] D. Bar-Lev, I. Orr, O. Sabary, T. Etzion, E. Yaakobi, Deep dna storage: Scalable and robust dna storage via coding theory and deep learning, arXiv preprint arXiv:2109.00031 (2021).

[18] Y. Nahum, E. Ben-Tolila, L. Anavy, Single-read reconstruction for dna data storage using transformers, arXiv preprint arXiv:2109.05478 (2021).

[19] X. Lv, Z. Chen, Y. Lu, Y. Yang, An end-to-end oxford nanopore basecaller using convolution-augmented transformer, in: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2020, pp. 337–342.

[20] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, S. Yekhanin, Trellis bma: Coded trace reconstruction on ids channels for dna storage, in: 2021 IEEE International Symposium on Information Theory (ISIT), IEEE, 2021, pp. 2453–2458.

[21] E. Zorita, P. Cusco, G. J. Filion, Starcode: sequence clustering based on all-pairs search, Bioinformatics 31 (12) (2015) 1913–1919.

[22] G. Qu, Z. Yan, H. Wu, Clover: tree structure-based efficient dna clustering for dna-based data storage, Briefings in Bioinformatics (2022).

[23] K. Matange, J. M. Tuck, A. J. Keung, Dna stability: a central design consideration for dna data storage systems, Nature communications 12 (1) (2021) 1–9.

[24] L. Song, F. Geng, Z.-Y. Gong, X. Chen, J. Tang, C. Gong, L. Zhou, R. Xia, M.-Z. Han, J.-Y. Xu, B.-Z. Li, Y.-J. Yuan, Robust data storage in dna by de bruijn graph-based de novo strand assembly, Nature Communications 13 (1) (2022) 5361. doi:10.1038/s41467-022-33046-w.

[25] V. Mallet, J.-P. Vert, Reverse-complement equivariant networks for dna sequences, Advances in Neural Information Processing Systems 34 (2021) 13511–13523.

[26] J. Kim, J. H. Bae, M. Baym, D. Y. Zhang, Metastable hybridization-based dna information storage to allow rapid and permanent erasure, Nature communications 11 (1) (2020) 1–8.

[27] I. Shomorony, R. Heckel, Dna-based storage: Models and fundamental limits, IEEE Transactions on Information Theory 67 (6) (2021) 3675–3689.

[28] P. K. Vippathalla, N. Kashyap, The secure storage capacity of a dna wiretap channel model, arXiv preprint arXiv:2201.05995 (2022).

[29] O. S. E. Y. Gadi Chaykin, Nili Furman, Dna storage similator (2021). URL `https://github.com/oyerush/DNASimulator`

[30] O. Sabary, Y. Orlev, R. Shafir, L. Anavy, E. Yaakobi, Z. Yakhini, Solqc: Synthetic oligo library quality control tool, Bioinformatics 37 (5) (2020) 720–722. `doi:10.1093/bioinformatics/btaa740`.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[32] B. Desplanques, J. Thienpondt, K. Demuynck, Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification, in: Proc. Interspeech 2020, 2020, pp. 3830–3834. `doi:10.21437/Interspeech.2020-2650`.

[33] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al., Conformer: Convolution-augmented transformer for speech recognition, arXiv preprint arXiv:2005.08100 (2020).

[34] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey, IEEE Transactions on Neural Networks and Learning Systems 28 (10) (2017) 2222–2232. `doi:10.1109/TNNLS.2016.2582924`.

[35] Y. Erlich, D. Zielinski, Dna fountain enables a robust and efficient storage architecture, science 355 (6328) (2017) 950–954.

[36] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, et al., Random access in large-scale dna data storage, Nature biotechnology 36 (3) (2018) 242–248.

[37] S. Chandak, K. Tatwawadi, B. Lau, J. Mardia, M. Kubit, J. Neu, P. Griffin, M. Wootters, T. Weissman, H. Ji, Improved read/write cost tradeoff in dna-based data storage using ldpc codes, in: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2019, pp. 147–156.

[38] H. Li, R. Durbin, Fast and accurate short read alignment with burrows–wheeler transform, bioinformatics 25 (14) (2009) 1754–1760.

[39] J. Guo, W. Bao, J. Wang, Y. Ma, X. Gao, G. Xiao, A. Liu, J. Dong, X. Liu, W. Wu, A comprehensive evaluation framework for deep model robustness, Pattern Recognition 137 (2023) 109308.