TransDNA: A Deep Transfer Learning Network for Sequence Reconstruction in DNA-Based Data Storage

Yun Qin, Fei Zhu, *Member*, *IEEE*, Bo Xi, Yuping Duan, *Member*,*IEEE*

Abstract—DNA is a promising storage medium, offering advantages in high density, long durability, and low maintenance cost. However, information recovery in DNA storage systems is challenged by errors arising during synthesis, amplification, and sequencing phases. A key challenge in decoding is sequence reconstruction, which involves recovering the original reference sequence from a set of noisy copies. While recent research has explored deep learning-based methods for this task, the high cost of synthesis and sequencing results in a limited availability of training samples. To overcome this challenge, we propose TransDNA, a deep transfer learning network specifically designed for sequence reconstruction in DNA storage. It consists of an encoder, a domain-specific decoder, and a domain-invariant feature extractor, with alternating domain alignment and domain-specific reconstruction mechanisms. By transferring knowledge from a larger source dataset, TransDNA significantly enhances the reconstruction success rate on two target datasets from real DNA storage experiments, outperforming the base model without transfer learning and several comparative methods. Notably, TransDNA surpasses the SDG method in both reconstruction success rate and training efficiency. These results demonstrate the effectiveness of TransDNA as the first transfer learning approach applied to the DNA sequence reconstruction task. The source code is available at: https://github.com/qinyunnn/TransDNA.

Index Terms—DNA storage, Sequence reconstruction, Deep transfer learning, Domain adaptation, MMD loss.

1 Introduction

DNA has emerged as a promising storage medium, offering advantages in terms of storage density, maintenance costs, and durability over traditional storage medium [1], [2]. Recent advancements in synthesis, sequencing technologies, and interdisciplinary research in biology and information technology have significantly contributed to the progress in DNA storage [3]. Existing studies have investigated the robustness, scalability, and feasibility of the complete workflow for DNA storage in the laboratory settings [4], [5], [6], [7], [8], [9]. As illustrated in Fig.1, a typical workflow of DNA storage system involves five steps. Binary information is first encoded into DNA sequences (encoding), which are then synthesized into DNA molecules (synthesis). These DNA molecules are stored either in vitro or in vivo (storage). Next, the stored DNA is sequenced by PCR (sequencing), and finally, the sequences are converted back into binary information (decoding) [2], [7].

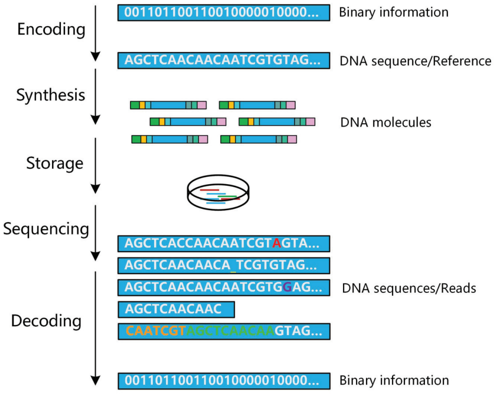


Fig. 1: Workflow of DNA storage system.

In a DNA-based data storage system, the process of encoding converts the original information into DNA sequences, referred to as *references*. Each reference undergoes amplification by synthesis and polymerase chain reaction (PCR), resulting in the generation of multiple replicated copies, termed *reads*, which are prone to errors. Errors present in the reads primarily arise from insertion, deletion and substitution (IDS) of a few bases during different phases of DNA storage system, including synthesis, storage, and sequencing. Additionally, long-term preservation and PCR-based strand copying may lead to DNA breaks and rearrangements [9]. To retrieve information from the sequencing file, the decoding process typically involves two steps: clustering and sequence reconstruction. A clustering algorithm [10], [11] is utilized to group reads originating from the same reference into clusters. Subsequently, the focus of this work lies in sequence reconstruction, which aims to infer the reference from a set of noisy reads [12].

Recently, the sequence reconstruction problem in DNA

This work was supported by the National Key Research and Development Program of China under Grant 2020YFA0712100 and the Emerging Frontiers Cultivation Program of Tianjin University Interdisciplinary Center. (Corresponding author: Fei Zhu)

Y. Qin, F. Zhu and B. Xi are with the Center for Applied Mathematics, Tianjin University, Tianjin, 300072, China (E-mail: qinyunnn@tju.edu.cn; fei.zhu@tju.edu.cn; bo_xi@tju.edu.cn).

F. Zhu is also with the State Key Laboratory of Synthetic Biology, Tianjin University, Tianjin, 300072, China.

Y. Duan is with School of Mathematical Sciences, Beijing Normal University, Beijing, 100875, China (E-mail: doveduan@gmail.com).

storage has emerged as a prominent area of research [12], [13], [14]. Bitwise Majority Alignment (BMA) has gained significant attention [13], [14]. Typically, the BMA-based methods align multiple reads and then applies a majority voting strategy to determine the symbol at each index. While highly effective for clusters with low IDS rates, these methods exhibit sensitivity to cluster size. When the number of reads is small, it is difficult to restore the sequence perfectly by majority voting. More recently, algorithms proposed in [12] decode the original reference by globally analyzing the cluster of reads and applying dynamic programming techniques, which are used to address the shortest common supersequence and longest common subsequence problems. Another category of reconstruction methods rely on statistical inference [15], [16], [17]. These methods assume an IDS channel associated with a particular DNA storage system. In this context, a single reference is assumed to pass through the channel several times, independently and repeatedly, generating multiple noisy observations. The decoding algorithms are derived by comparing the a posteriori probabilities (APPs) of all possible symbols at each index position across all observed reads [15], [16]. Nevertheless, these methods suffer a high computational burden, particularly when dealing with large clusters comprising more than ten reads. Moreover, the channel parameters employed for decoding is inaccurate, as the actual IDS error rates associated to the DNA storage system are inaccessible.

The advent of deep neural networks (DNN), exemplified by Transformer [18], has exhibited the capability to effectively learn and extract semantic information from DNA sequences. Several studies have been devoted to deep learning-based algorithms for sequence reconstruction. Bar-Lev *et al.* [19] introduced a scalable and robust sequence reconstruction approach that employs fast pseudo-clustering and combines convolution and Transformer blocks for effective error correction. Nahum *et al.* [20] investigated a DNN-based algorithm for the single-read reconstruction, utilizing an encoder-decoder architecture composed of multiple Transformer blocks. More recently, we proposed in [21] a robust multi-read reconstruction model that considers sequence reliability within clusters. This method leverages an attention mechanism to score the sequences within clusters and employs Conformer blocks to correct IDS errors.

Current DNN models for sequence reconstruction are typically large-scale, necessitating a substantial amount of samples as the foundation for model training. However, accessing DNA storage data is costly and time-consuming, constrained by the expenses associated with synthesis and sequencing, as well as experimental limitations. Synthesizing a single base currently costs around 10^{-3} dollars, and storing 1 TB of data amounts to approximately 1 billion dollars [22]. Insufficient training samples may result in overfitting issues during network training.

In response to this challenge, Bar-Lev *et al.* [19] pioneered the use of Synthetic Data Generator (SDG) [23] for training the sequence reconstruction network. By injecting the statistical IDS error rates into a reference sequence, SDG generates multiple noisy copies for the sequence. These generated sequences, along with the reference are then utilized to form the labeled training clusters. Experimental results demonstrate that the synthesized data generated by SDG can effectively replace real experimental data in model training. This is attributed to the capability of SDG to infinitely generate data conforming to a specific error pattern based on the provided parameters. By far, SDG stands as the exclusive solution for addressing the issue of limited training samples in sequence reconstruction networks.

An important question arises: *Is it possible to directly employ publicly available DNA storage datasets for training sequence reconstruction models?* Transfer learning is a potential strategy. It exploits the correlation between data or tasks, involves transferring knowledge from the source domain to facilitate more effective model training in the target domain [24]. Several studies have explored pre-trained DNA language models with transfer learning for different sequence analysis tasks [25], [26], [27]. For instance, Ji *et al.* [25] developed DNABERT, a pre-trained bidirectional encoder model based on BERT, to acquire global and transferable understanding of genomic DNA sequences. Luo *et al.* [26] proposed iEnhancer-BERT, a pre-trained DNA language model comprising a BERT layer for feature extraction and a CNN layer for classification task. These works have enhanced the accuracy and efficiency of sequence analysis by leveraging pre-trained models fine-tuned for specific tasks.

However, as of now, transfer learning remains unexplored in the context of addressing the challenge of limited training samples in DNA storage-based sequence reconstruction tasks. The challenge is two-fold. Firstly, there is variation in the lengths of references and reads obtained from different experiments, attributed to the specific encoding methods designed. This necessitates an adaptive sequence reconstruction model that can accommodate the varying input and output lengths. Secondly, specific experimental datasets exhibit distinct error rates due to factors such as the instruments used for synthesis and sequencing, as well as storage conditions. The resulting distribution discrepancy between the source and target domains poses additional challenges for transfer learning.

In this paper, we propose TransDNA, the first transfer learning-based model specifically designed to address the sample scarcity problem in the sequence reconstruction task for DNA storage. TransDNA offers an innovative alternative to synthetic training data strategy using SDG [19], effectively tackling the challenge of limited labeled samples in sequence reconstruction. The major contributions are as follows:

- **Exploration of transfer learning for small sample sequence reconstruction:** This is the first application of the transfer learning to address the challenge of limited training samples in DNA storage-based sequence reconstruction. By leveraging knowledge from source datasets, TransDNA demonstrates its effectiveness in enhancing sequence reconstruction performance in target datasets despite limited data.
- **Positive knowledge transfer by domain adaptation:** TransDNA adopts the concept of domain adaptation by employing Maximum Mean Discrepancy (MMD) loss to align the source and target domain by a domain-invariant feature extractor. This efficiently mitigates the distribution discrepancy between the domains, enhancing the overall performance of the model.

- **Error correction capacity for IDS errors and adaptive input/output lengths:** The proposed model effectively corrects IDS errors due to the robust feature extraction capabilities of the Conformer block within the encoder. Additionally, the model can handle varying input and output lengths through the use of an autoregressive LSTM in the decoder. This flexibility allows for adaptive customization to different source and target datasets, even those from distinct experiments.

2 Methods

2.1 Problem formulation

Use $\Sigma = \{A, C, G, T\}$ to denote the four DNA nucleotides. Assume a reference $x \in \Sigma^L$ of length L pass through a DNA storage channel, generating t noisy copies $y = \{Y_1, Y_2, ..., Y_t\} \in \mathcal{C}$, where each $Y_i \in \Sigma^{L_i}$ represents a copy of varying length. Sequence reconstruction aims at finding a mapping

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

such that $d(x, \mathcal{F}(y))$ is minimized, where $d(\cdot, \cdot)$ represents a distance metric, *e.g.*, edit distance or Hamming distance.

Now, consider the task of enhancing sequence reconstruction on a small DNA storage dataset, referred to as the target domain $\mathcal{D}_T$, by leveraging knowledge from a larger DNA storage dataset, known as the source domain $\mathcal{D}_S$, through transfer learning. Let P_S and P_T denote the distributions of the source and target domains, respectively, where $P_S \neq P_T$. Consider a set of n_S labeled samples $(y_j^S, x_j^S)_{j=1}^{n_S}$ from the source domain $\mathcal{D}_S$ and n_T labeled samples $(y_j^T, x_j^T)_{j=1}^{n_T}$ from the target domain $\mathcal{D}_T$, with $n_S \gg n_T$.

The main goal of sequence reconstruction is to minimize the expected value of the loss function on the target domain, expressed by

$$R_T(\mathcal{F}) = \mathbb{E}_{(y,x)\sim P_T(y,x)}[\mathcal{L}(x, \mathcal{F}^*(y))], \tag{1}$$

with the estimation form given by

$$\frac{1}{n_T}\sum_{j=1}^{n_T}[\mathcal{L}(x_j^T, \mathcal{F}^*(y_j^T))]. \tag{2}$$

Here, $\mathcal{L}(\cdot)$ is the loss function based on some distance metric that measures the inconsistency between the actual reference sequence and the one predicted by the improved mapping function $\mathcal{F}^*$ after transfer learning [28].

2.2 Model overview

TransDNA is designed to address the training sample scarcity issue in the sequence reconstruction task for DNA storage, leveraging advanced techniques in transfer learning and domain adaptation. As depicted in Fig.2, the framework of TransDNA consists of three main components:

1) Encoder: The Conformer-based encoder plays a pivotal role in extracting high-level features from sequence clusters. By employing a hybrid approach combining convolutions and self-attention mechanisms, it effectively captures IDS error patterns and models both local and global dependencies within clusters.
2) Domain-specific Decoders: Utilizing autoregressive Long Short-Term Memory (LSTM) networks, the domain-specific decoders are capable to handle variable-length sequences and model long-term dependencies in sequential DNA data. They operate independently in both the source and target domains.
3) Domain-invariant Feature Extractor: Composed of 1D covolution layers, the domain-invariant feature extractor primarily aligns the encoder outputs from both domains, avoiding negative transfer effects stemming from distribution discrepency.

As illustrated in Algorithm 1, TransDNA genearlly alternates between two stages: In the first stage, domain alignment is performed to establish compatibility between two domains, where domain adaptation by Maximum Mean Discrepancy (MMD) loss is performed. The second stage focuses on domain-specific reconstruction, leveraging the aligned domains to achieve precise sequence reconstruction in each specifc domain.

2.3 Network structure

2.3.1 Data Preprocessing

The input to the model consists of a cluster of reads with an unfixed sequence number, and each read may have varying sequence lengths. Before fed to the network, each sequence in the cluster undergoes one-hot encoding according to different nucleotides and is then padded to a uniform length of k_i ($i = S$ or T), where shorter sequences are zero-padded at the end to match the specified length. The resulting sequences are subsequently summed across the index positions. Consequently, the input to the model is a matrix with dimensions $4 \times k_i$, where each column is a 1-D vector denoting the confidence of the corresponding base at that particular index position. The reference sequence, which has a pre-fixed length of L_i, is encoded as a $4 \times L_i$ matrix using one-hot encoding. Accounting for the possibility of insertions in the sequence, we set $k_i \geq L_i + 1$. As sequences whose lengths differed from the reference length by more than 5 were excluded from analysis, we also have $k_i \leq L_i + 5$.

Upon inputting the data into the model, convolutional up-sampling is performed using kernels of diverse sizes. This step transforms the input into a feature matrix with an expanded feature dimension of $64 \times k_i$. The utilization of convolution kernels of various sizes enables the extraction of features at multiple scales.

2.3.2 Encoder (E)

As shown in Fig.2 (b), we employ a single Conformer block without the post layer norm as the encoder E to achieve a shared representation across all domains. The Conformer [29], originally designed for speech recognition tasks, combines convolutional layers with self-attention to efficiently capture both local details and global dependencies within sequence data, surpassing Transformer and CNN counterparts in speech-related tasks. It features two macaron-like feed-forward layers with residual connections of weight 1/2, interleaved with Multi-Headed Self-Attention (MHSA) and convolution operations.

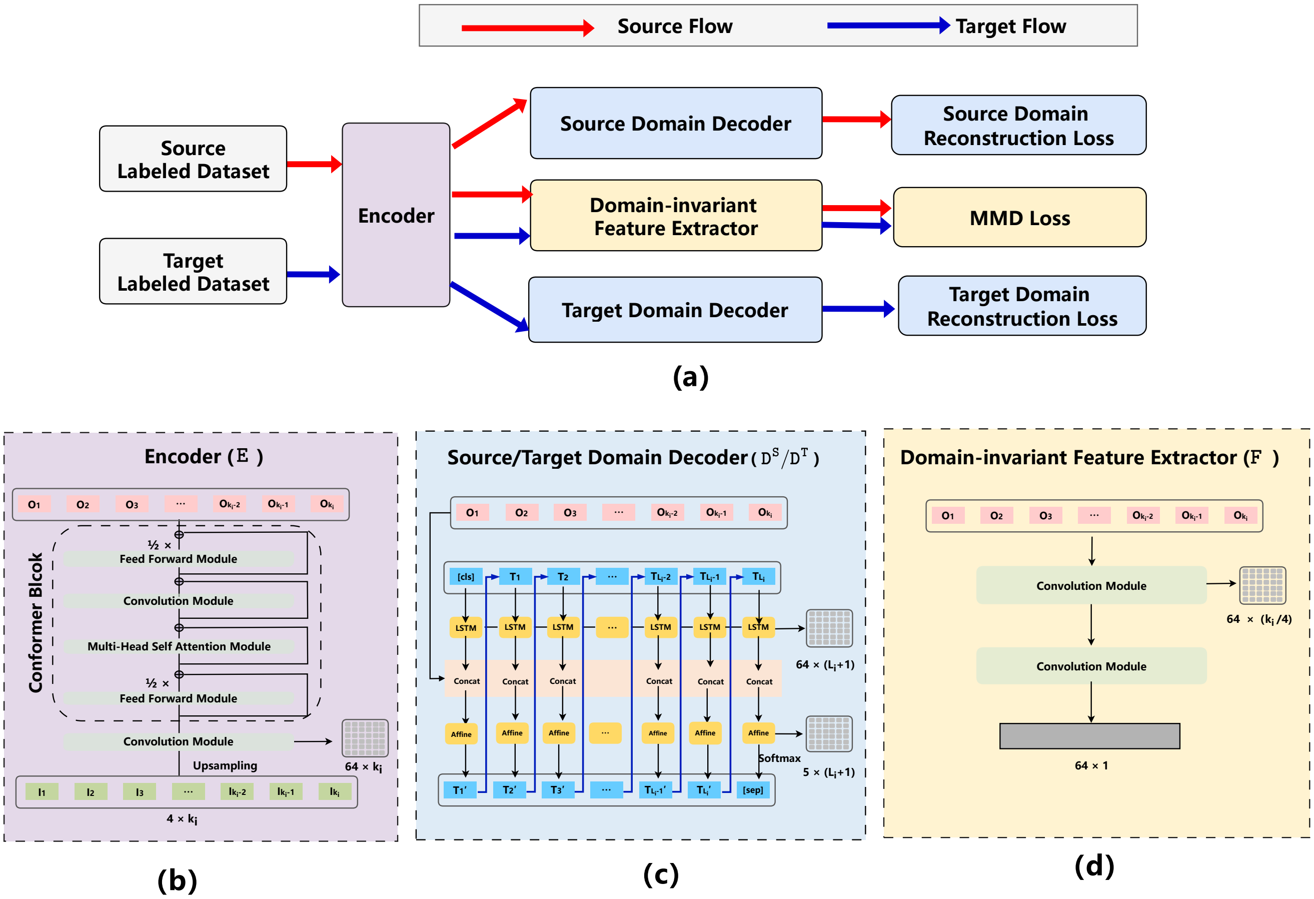

Source Flow
Target Flow
Source Labeled Dataset
Target Labeled Dataset
Encoder
Source Domain Decoder
Domain-invariant Feature Extractor
Target Domain Decoder
Source Domain Reconstruction Loss
MMD Loss
Target Domain Reconstruction Loss
(a)
Encoder (E)
Conformer Blcok
O1 O2 O3 ··· Ok_i-2 Ok_i-1 Ok_i
½ × Feed Forward Module
Convolution Module
Multi-Head Self Attention Module
½ × Feed Forward Module
Convolution Module
Upsampling
64 × k_i
I1 I2 I3 ··· Ik_i-2 Ik_i-1 Ik_i
4 × k_i
(b)
Source/Target Domain Decoder (D^S/D^T)
O1 O2 O3 ··· Ok_i-2 Ok_i-1 Ok_i
[cls] T1 T2 ··· TL_i-2 TL_i-1 TL_i
LSTM
Concat
Affine
64 × (L_i+1)
Softmax
5 × (L_i+1)
T1' T2' T3' ··· TL_i-1' TL_i' [sep]
(c)
Domain-invariant Feature Extractor (F)
O1 O2 O3 ··· Ok_i-2 Ok_i-1 Ok_i
Convolution Module
64 × (k_i/4)
Convolution Module
64 × 1
(d)

Fig. 2: Overview of TransDNA. (**a**) Network structure of TransDNA. It consists of an encoder, a domain-invariant feature extractor, and two domain-specific decoders. (**b**) Conformer-based encoder. The input to TransDNA is a matrix of size $4 \times k_i$ ($i = S$ or T). After convolution up-sampling, the feature becomes a $64 \times k_i$ matrix, which is fed into the Conformer-encoder block. The encoder output is also a $64 \times k_i$ matrix. (**c**) Source/Target domain decoder. Each decoder incorporates $< cls >$ token (start of the sentence) and $< sep >$ token (sentence separation), with $< cls > = < sep >$. During training, the label is fed as a whole in the time direction. During inference, the $< cls >$ token serves as the initial input, and the output from each time step is used as the input for the subsequent step. (**d**) Domain-invariant feature extractor. This module aligns the encoder outputs from two domains to 64×1 vectors.

For sequence reconstruction tasks, we choose the Conformer block as the encoder for its powerful feature extraction capabilities and ability to model IDS patterns within sequences. The hybrid architecture of Conformer effectively captures both local and global features, essential for handling IDS errors. Convolutional layers extract local features and positional offsets, crucial for identifying insertion and deletion patterns. The MHSA with positional encoding ensures accurate retention of positional information across the network. Furthermore, the hierarchical structure in Conformer further enhances feature extraction across multiple scales. This comprehensive approach empowers the Conformer-based encoder to capture high-level features within clusters and effectively handle positional shifts caused by base insertions and deletions.

In the encoder, the MHSA employs the scaled-dot product as the basis for its calculations, as described in [18]:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{Q^\top K}{\sqrt{d_k}})V \tag{3}$$

where Q, K, V denote the query, key and value matrices, respectively, and d_k is the scaling factor that equal to the feature dimension of queries and keys. For the h-head attention mechanism, the feature vectors are transformed h times using (3) before concatenating them, expressed by

$$\text{MHSA}(Y) = W^\top \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_\text{h}), \tag{4}$$

$$\text{head}_\text{i} = \text{Attention}(W_i^Q Y, W_i^K Y, W_i^V Y) \tag{5}$$

where $Y \in \mathcal{R}^{d \times k_i}$ ($i = S$ or T), is the input to MHSA, $W_i^Q, W_i^K, W_i^V \in \mathcal{R}^{d_k \times d}$ for $i = 1, 2, ...h$. The matrix $W \in \mathcal{R}^{hd_k \times d}$ maps the concatenated feature back to the original dimension $d \times k_i$. In MHSA, the relative position encoding from Transformer-XL [30] is applied to model the relative positional relationships of bases in a sequence, following [29].

In practical implementation, we set $h = 8$, $d = 64$ and $d_k = d/h = 8$, while retaining most parameter settings from Conformer-S [29]. Specifically, we reduced the encoder dimension from $d = 144$ in Conformer-S to $d = 64$ in our encoder. This adjustment is based on the premise that a smaller dimension is sufficient for capturing the feature space of DNA sequences, given that the four nucleotide bases represent a less complex feature space compared to the original Automatic Speech Recognition (ASR) task.

In convolution module, we perform two pointwise convolutions and a 1-D depthwise convolution with kernel size of 16. In addition, each feed-forward module (FFN) is a fully connected network with two linear layers. Specifically, we fix the coefficients of the FFN layer to $1/2$, following the original Conformer paper [29]. A theoretical explanation for setting the half-step residual connection in the Transformer architectures is provided in Macron-Net [31] from the perspective of Ordinary Differential Equations (ODEs).

In general, for the feature $\widetilde{Y}$ (after convolution modules) fed to the conformer block, the corresponding output $Y_{conformer}$ is expressed as follows, considering the half-step residual connection:

$$Y^{'} = \widetilde{Y} + \frac{1}{2}\mathrm{FFN}(\widetilde{Y}) \tag{6}$$

$$Y^{''} = Y^{'} + \mathrm{MHSA}(Y^{'}) \tag{7}$$

$$Y^{'''} = Y^{''} + \mathrm{Conv}(Y^{''}) \tag{8}$$

$$Y_{conformer} = Y^{'''} + \frac{1}{2}\mathrm{FFN}(Y^{'''}) \tag{9}$$

2.3.3 Domain-specific Decoder (D^S and D^T)

The domain-specific decoder in our model comprises a source domain decoder $\mathtt{D}^{\mathtt{S}}$ and a target domain decoder $\mathtt{D}^{\mathtt{T}}$, sharing the same structure as depicted in Fig.2 (c). This design ensures consistent sequence reconstruction across domains. To capture contextual dependencies in sequential DNA data, we employ an autoregressive LSTM (Long Short-Term Memory) as the decoder. The reason for this choice is three-fold. Firstly, due to the variation in the length of references obtained from different experiments, the decoder must be capable of handling variable-length sequences. The autoregressive LSTM is known for the ability to generate sequences with continuous semantic information, making it well-suited for reconstructing references of varying lengths [32]. Secondly, in our transfer learning context with limited training samples, a simpler decoder with fewer parameters, such as LSTM, is more preferable over a Transformer decoder. This choice mitigates overfitting and improves generalization, particularly in the target domain where data is scarce. Lastly, our Conformer-based encoder is advantageous in feature extraction for DNA sequences, making a simple network like LSTM sufficient for effective decoding.

During training, the LSTM employs teacher forcing, namely the input at each time step is the true label from the ground-truth sequence, rather than the generated output. This facilitates faster convergence and enhances the accuracy of the generated sequences. During inference, the input at each time step is the output from the previous time step, and the model generates the prediction for the next time step based on this output, as shown by the arrows in Fig.2 (c). Regarding the impact of sequence length in transfer learning between different domains, the source and target domains share only the encoder while using separate decoders.

Specifically, the input to the decoder is a $5 \times (L_i + 1)$ matrix, where L_i is the reference length specific to that domain, and each column corresponds to the symbol at a specific index position. Here, we introduce a new symbol $< cls >$ and $< sep >$ to mark the start and end of the generated sequence, where $< cls > = < sep >$. After passing through the LSTM block, it produces an output matrix O_{lstm} with dimensions of $64 \times (L_i + 1)$. The output of the encoder, denoted as $O_{encoder}$, is truncated in length from $64 \times k_i$ to $64 \times (L_i + 1)$. Then, O_{lstm} and $O_{encoder}$ are concatenated into a $128 \times (L_i + 1)$ matrix, which is subsequently mapped to a $5 \times L_i$ matrix by a linear layer.

For each decoder $\mathtt{D}^i$ ($i = S$ or T), the corresponding loss function is defined as the expectation value of cross-entropy $\mathcal{L}_{CE}$, computed over the probability distribution of samples specific to that domain, expressed by

$$\mathcal{L}^i_{decoder} = \mathbb{E}_{(y,x)\sim P_i(y,x)}[\mathcal{L}_{CE}(x^i, \mathtt{D}^i(\mathtt{E}(y^i)))], \tag{10}$$

with $\mathcal{L}_{CE}$ being the cross-entropy loss given by

$$\mathcal{L}_{CE} = -\sum_{l=1}^{L_i} x^i_l \log \mathcal{F}(y^i_l), \tag{11}$$

with x^i_l being one-hot label vector recording the actual base category for the l-th position, while $\mathcal{F}(y^i_l)$ being the predicted probability vector for the read cluster y^i at the l-th position.

2.3.4 Domain-invariant Feature Extractor (F)

To address distribution discrepancies between source and target domains, such as those arising from variations in error rates, sequence lengths, or encoding methods, we apply domain adaptation technique by minimizing Maximum Mean Discrepancy (MMD) loss. This technique aligns the distributions across domains, which is crucial as significant gaps can hinder the extraction of common features and lead to negative transfer effects. The domain-invariant feature extractor $\mathtt{F}$, as depicted in Fig.2 (d), ensures consistency between outputs from both domains, thereby mitigating the risk of negative transfer.

Specifically, this module aligns the encoder outputs from two domains. It comprises two 1-D convolution layers that take the output of the encoder as input. The first convolution employs a kernel size of 3 to transform the output of encoder, namely $\mathtt{E}(y_j)$ into a $64 \times k_i/4$ matrix. Subsequently, the output is processed by a 1-D convolution with a kernel size of 5, which transforms it into a 64×1 vector.

To achieve domain alignment, we employ the well-known MMD as the metric to measure the the discrepancy between two domains in high-dimensional spaces [33], [34]. It utilizes the kernel method to map samples from both domains into the feature space and computes the differences between the mapped samples, quantifying the distribution discrepancy. Let $\phi(\cdot)$ be a nonlinear function mapping the samples from the input space to the reproducing kernel Hilbert space (RKHS) endowed by some kernel $\kappa(\cdot, \cdot)$. Mathematically, MMD is formulated as

$$\mathcal{MMD}_{\mathcal{H}}(P_S, P_T) = \|\mathbb{E}_S[\phi(y^S)] - \mathbb{E}_T[\phi(y^T)]\|^2_{\mathcal{H}}, \tag{12}$$

where $\|\cdot\|_{\mathcal{H}}$ is the associated norm.

An unbiased estimate of (12) computes the squared distance between the empirical kernel mean embeddings, given by

$$\mathcal{M}\hat{\mathcal{M}}\mathcal{D}_{\mathcal{H}}(P_S, P_T) = \left\| \frac{1}{n_S} \sum_{y_j^S \in \mathcal{D}_S} \phi(y_j^S) - \frac{1}{n_T} \sum_{y_j^T \in \mathcal{D}_T} \phi(y_j^T) \right\|^2_{\mathcal{H}}, \tag{13}$$

where the inner product can be evaluated by the kernel trick, with $\langle \phi(y^i), \phi(y^j) \rangle = \kappa(y^i, y^j)$, $(i, j = S$ or $T)$. Therefore, the MMD loss for domain alignment is given by

$$\mathcal{L}_{mmd} = \mathcal{M}\hat{\mathcal{M}}\mathcal{D}_{\mathcal{H}}(\mathtt{F}(\mathtt{E}(y^S)), \mathtt{F}(\mathtt{E}(y^T))), \tag{14}$$

where the commonly-used Gaussian kernel of the form $\kappa(y^S, y^T) = \exp\left(-\frac{\|y^S - y^T\|^2}{2\sigma^2}\right)$ is adopted.

2.4 Training process and loss function

To improve training efficiency, we have implemented a pre-training strategy aimed at accelerating convergence during model training. At the beginning of training, the encoder $\mathtt{E}$ and source-domain decoder $\mathtt{D}^{\mathtt{S}}$ are jointly pre-trained using the entire source domain dataset and then fine-tuned. Pre-training benefits downstream tasks by learning generalized features, reducing reliance on labeled data, and improving generalization. In contrast, the target domain decoder $\mathtt{D}^T$ and domain-invariant feature extractor $\mathtt{F}$ are trained from scratch with randomly initialized parameters.

Following pre-training, the source and target domain data are sequentially passed through the encoder $\mathtt{E}$ and domain-invariant feature extractor $\mathtt{F}$. The MMD loss in (14) is calculated to align the distributions of the source and target domains.

Subsequently, the reconstruction losses for both domains are independently evaluated. The source domain data is processed by the source-domain decoder $\mathtt{D}^{\mathtt{S}}$ to compute the source domain reconstruction loss, as defined in (10). Similarly, the target domain data is processed by the target-domain decoder $\mathtt{D}^{\mathtt{T}}$ to calculate the target domain reconstruction loss using (10).

As a result, the total loss function of TransDNA consists of three parts, namely the decoder losses for both the source and target domains, as well as the MMD loss between them. It is expressed by

$$\mathcal{L}_{total} = \mathcal{L}^S_{decoder} + \mathcal{L}^T_{decoder} + \alpha \times \mathcal{L}_{mmd} \tag{15}$$

where α is an adjustable parameter weighting the importance of MMD loss. In this study, we emprically set $\alpha = 0.5$ in all the experiments. An analysis of the parameter sensitivity of α is provided in Section 3.8.

We employ the Adam optimizer [35] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, and apply L_2 regularization with weight decay of $1e-4$. Dropout is applied after each convolution layer with a probability of 0.1. The initial learning rate is set to 0.001, and an exponentially decaying learning rate strategy is employed. Algorithm 1 outlines the training algorithm for TransDNA.

Algorithm 1 Training algorithm for TransDNA

Input: source labeled dataset $(y_j^S, x_j^S)_{j=1}^{n_S}$, target labeled dataset $(y_j^T, x_j^T)_{j=1}^{n_T}$, pre-trained encoder $\mathtt{E}$ and source-domain decoder $\mathtt{D}^{\mathtt{S}}$, randomly initialized target domain decoder $\mathtt{D}^{\mathtt{T}}$ and domain-invariant feature extractor $\mathtt{F}$.

Output: well-trained encoder $\mathtt{E}^*$, decoder $\mathtt{D}^{\mathtt{S}*}$ and $\mathtt{D}^{\mathtt{T}*}$, and domain-invariant feature extractor $\mathtt{F}^*$.

```
Give the number of training iterations Q
for q in 1:Q do
    Draw $m$ samples $(y_j^S, x_j^S)_{j=1}^{m}$ from the source dataset.
    Draw $n$ samples $(y_j^T, x_j^T)_{j=1}^{n}$ from the target dataset.
    Input the sampled data from both domains into the encoder in turn to obtain $\mathtt{E}(y_j^S)$ and $\mathtt{E}(y_j^T)$.
    Feed $\mathtt{E}(y_j^S)$ and $\mathtt{E}(y_j^T)$ into the domain-invariant feature extractor to obtain $\mathtt{F}(\mathtt{E}(y_j^S))$ and $\mathtt{F}(\mathtt{E}(y_j^T))$, respectively, and computes the MMD loss by (14).
    Feed $\mathtt{E}(y_j^S)$ and $\mathtt{E}(y_j^T)$ into the domain-specific decoder to obtain $\mathtt{D}^{\mathtt{S}}(\mathtt{E}(y_j^S))$ and $\mathtt{D}^{\mathtt{T}}(\mathtt{E}(y_j^T))$, respectively, and compute their corresponding reconstruction losses by (10).
    Update $\mathtt{E}$, $\mathtt{F}$, $\mathtt{D}^{\mathtt{S}}$ and $\mathtt{D}^{\mathtt{T}}$ by minimizing the total loss in (15).
end for
$\mathtt{E}^* = \mathtt{E}$, $\mathtt{F}^* = \mathtt{F}$, $\mathtt{D}^{\mathtt{S}*} = \mathtt{D}^{\mathtt{S}}$, $\mathtt{D}^{\mathtt{T}*} = \mathtt{D}^{\mathtt{T}}$.
```

3 Results

3.1 Data preparation

To evaluate the performance of TransDNA when data availability is limited, experiments were conducted using five publicly available DNA storage datasets of different sizes. TABLE 1 provides a description of the datasets used in this study. Three larger datasets, namely 'id20 [7]', 'P10_5_BDDP210000009 [9]' and 'PE_AYB [4]', were used as source datasets. Two smaller datasets, namely 'Sequencing_data_first_dimension [36]' and 'SRR9701379 [37]', were chosen as target datasets.

Each dataset consists of two files. The first file contains a list of references encoded using the original information, while the second file stores the sequencing outcomes in a disordered manner. To prepare labeled clusters for supervised learning, we paired each read with its most comparable reference using the Burrows-Wheeler-Alignment Tool (BWA) [38] on both files. For double-end sequencing results, sequences were merged using the Paired-End reAd mergeR (PEAR) [39]. Default parameters for both tools were employed to ensure consistency across experiments and minimize variability due to parameter tuning. It is noteworthy that a small fraction of reads that were excessively long or short, even after alignment, were removed from the analysis. Specifically, sequences differing from the reference length by more than 5 bases were excluded. Due to the inherent redundancy and the typical 1-2% error rate in next-generation sequencing, this exclusion minimally impacts overall performance and generalizability, as most reads fall within the acceptable range and contribute effectively to the analysis. For each source dataset, half of the clusters are used as the training set, and the other half are used as the testing set. Similarly, for each target dataset, half of the clusters are designated as the training set, with the remaining half reserved for the testing set.

3.2 Compared Methods

To evaluate the effectiveness of our transfer learning-based approach for small-sample sequence reconstruction tasks, we conducted a comparative analysis involving TransDNA and following five methods:

- **Base model without transfer learning:** We removed the domain-invariant feature extractor and one decoder from the TransDNA structure, resulting in an encoder-decoder model for sequence reconstruction. This base model serves as a benchmark for comparison.
- **Synthetic training data strategy using SDG [19]:** Bar-Lev *et al.* utilized SDG [23] to generate a synthetic dataset with 1.5M labeled clusters for training their network. Following [19], the training set for each target dataset was utilized to compute IDS error rates. References in quantities of 0.5M, 1M, and 1.5M, with sequence lengths matching those of the target dataset were then generated. These error rates and references were input into the SDG to generate synthetic training data in the form of labeled clusters. The size of each cluster is randomly determined, with a maximum number of up to 30, to simulate real scenarios in DNA storage.
- **Iterative Reconstruction [12]:** This algorithm begins by correcting insertion and substitution errors within a cluster using the error vector majority algorithm. It then addresses deletion errors through the pattern-path algorithm. If the cluster includes at least one sequence with the same length as the reference, the algorithm identifies and returns the sequence that minimizes the edit distance within the cluster. In cases where the cluster contains a sequence differing in length by one from the reference, the algorithm returns the most frequently occurring sequence.
- **BMA Lookahead [13]:** This is an enhancement to the BMA algorithm. For any sequence where the current symbol mismatches with the majority symbol, a look-ahead window is utilized to inspect the subsequent 2 (or more) symbols. Subsequently, the symbols within the look-ahead window are compared against the symbols elected through voting, determining their correctness.
- **RobuSeqNet [21]:** Our previous work, RobuSeqNet, is a deep learning-based multi-sequence reconstruction model designed for DNA storage. It is robust to noisy clusters with contaminated sequences resulting from DNA breakage and rearrangement, as well as noisy reads with IDS errors. The model features an encoder-decoder structure, and its attention module reduces the impact of contaminated sequences on reconstruction accuracy by automatically scoring the reads within a cluster.

We employed the commonly-used success rate [19], [21] as the evaluation metric for comparing the performance of different sequence reconstruction methods, defined by

$$\text{success rate} = \frac{\#\{\text{ predicted sequence} = \text{input reference }\}}{\#\{\text{ input reference }\}}. \tag{16}$$

This metric measures the proportion of sequences that are perfectly reconstructed without any errors at any position, out of all the references.

3.3 Results

The reconstruction performances of all compared methods are presented in Table 2. As observed, TransDNA surpasses all other methods on both target datasets, achieving a reconstruction success rate of over 98.2% for 'Sequencing_data_first_dimension [36]', and over 97.5% for 'SRR9701379 [37]' by transferring positive knowledge from each of the source datasets. Specifically, Iterative Reconstruction [12] and BMA Lookahead [13] exhibit inferior performance compared to the base model on target dataset 'Sequencing_data_first_dimension [36]'. However, on target dataset 'SRR9701379 [37]', Iterative Reconstruction [12] outperforms the base model by 1.42% in terms of success rate. RobuSeqNet [21], which is trained from scratch using only the training set in the target domain, outperforms the base model on both target domains but shows slightly inferior performance compared to TransDNA.

Compared to the base model trained using only the training set in the target domain, both TransDNA and its primary counterpart synthetic training data strategy using SDG [19] achieved improved success rates. Specifically, on the target dataset 'Sequencing_data_first_dimension [36]', TransDNA increased the success rate by approximately 1.2%, while the synthetic training data strategy [19] yielded an improvement of approximately 0.5%. Similarly, on the target dataset 'SRR9701379 [37]', TransDNA significantly improved the success rate by approximately 2.2%, while the synthetic training data strategy [19] demonstrated a modest improvement of approximately 0.6%.

Additionally, increasing the amount of training data generated by SDG from 0.5M to 1.5M resulted in only marginal improvements in the success rate for the synthetic training data strategy. This suggests that the performance gains of the SDG method reach saturation when the training data exceeds a certain quantity. The results indicate that TransDNA has a superior boosting effect on the two target datasets compared to the synthetic training data strategy using SDG. Furthermore, TransDNA offers a cost-effective and flexible alternative to SDG, requiring a smaller training set. For instance, the largest source dataset, 'id20 [7]', consists of hundreds of thousands of samples, which is significantly smaller than the millions of training samples required by the SDG method [19] to achieve satisfactory results.

Lastly, given that the target dataset 'Sequencing_data_first_dimension [36]' is approximately 2.76 times larger than 'SRR9701379 [37]', it is reasonable to infer that TransDNA demonstrates more significant improvements for smaller target datasets, as to be examined next in Section 3.4.

3.4 Effectiveness with varying target dataset size

To examine the impact of target dataset size on the advantage of TransDNA over the base model, we randomly sam-

TABLE 1: Description of the source and target datasets.

	Source			Target	
	id20 [7]	P10_5_BDDP210000009 [9]	PE_AYB [4]	Sequencing_data _first_dimension [36]	SRR9701379 [37]
Reference number	607150	210000	153335	11826	4355
Designed length	150	200	183	196	143
Synthesis	Twist Bioscience	Twist Bioscience	Agilent SurePrint	IDT	Microarray Synthesizer
Sequencing	Ilumina NextSeq	Ilumina	Illumina HighSeq	Ilumina MiSeq	MiniSeq
References aligned to reads	596669	209185	153331	11751	4355
Missing clusters	10481	815	4	75	0
Reads aligned to references	14486345	15256705	69510060	2687556	943113

TABLE 2: Comparison of sequence reconstruction performance.

Target	Source	Success rate							
		Base model	SDG (0.5M)	SDG (1M)	SDG (1.5M)	Iterative Reconstruction [12]	BMA Lookahead [13]	RobuSeqNet [21]	TransDNA
Sequencing_data _first_dimension [36]	id20 [7]	97.15%	97.61%	97.63%	97.61%	96.44%	96.54%	97.65%	**98.27%**
	P10_5_ BDDP210000009 [9]		97.65%	97.68%	97.68%				**98.26%**
	PE_AYB [4]		97.51%	97.43%	97.48%				**98.33%**
SRR9701379 [37]	id20 [7]	95.42%	96.01%	96.06%	96.20%	96.84%	93.37%	96.84%	**97.62%**
	P10_5_ BDDP210000009 [9]		95.97%	95.92%	96.15%				**97.62%**
	PE_AYB [4]		95.97%	96.01%	96.06%				**97.53%**

TABLE 3: Impact of target dataset size on the performance advantage of TransDNA over the base model.

Target	Sampling (%)	Target set size	Target train set size	Test set size	Base model	Success rate		
						id20 [7]	P10_5_BDDP210000009 [9]	PE_AYB [4]
SRR9701379 [37]	10%	435	216	219	89.95%	94.98%	94.98%	**95.43%**
	20%	871	443	428	90.65%	95.09%	**95.79%**	94.16%
	30%	1306	650	656	93.29%	**95.73%**	95.58%	**95.73%**
	40%	1742	875	867	91.79%	96.66%	96.67%	**97.12%**
	50%	2177	1093	1084	94.19%	**95.48%**	95.11%	**95.48%**
	60%	2613	1312	1301	93.85%	**96.77%**	96.69%	96.69%
	70%	3048	1529	1519	94.73%	**97.30%**	**97.30%**	96.97%
	80%	3484	1752	1732	95.03%	97.17%	97.11%	**97.34%**
	90%	3919	1965	1954	96.11%	**96.93%**	96.88%	**96.93%**
	100%	4355	2172	2183	95.42%	**97.62%**	**97.62%**	97.53%

ple a certain percentage of samples from the target dataset 'SRR9701379 [37]' to simulate smaller target datasets. TABLE 3 provides the sampling details and the corresponding test results. We observe that TransDNA demonstrates more pronounced improvements in success rate for smaller sample sizes. For example, when sampling 10% of the target training data, TransDNA achieves a 5.48% improvement in success rate compared to the base model. However, this improvement reduces to 2.2% when the network is trained using the full target training data. These findings confirm the effectiveness of TransDNA in enhancing sequence reconstruction especially for smaller datasets.

3.5 Ablation study on domain alignment

To demonstrate the necessity of the domain alignment phase, we conducted an ablation study by comparing the performance of TransDNA with and without the domain-invariant feature extractor F. We utilized three source datasets and two target datasets for this evaluation. As shown in TABLE 4, excluding this module resulted in a decrease in the success rate by approximately 0.1-0.2%. This demonstrates the crucial role of domain alignment in obtaining domain-invariant feature representations, which are essential for successful sequence reconstruction.

3.6 Wrong prediction analysis

Fig. 3 depicts the frequency histogram of the edit distances between erroneous predictions and their corresponding reference sequences. In this analysis, 'id20 [7]' was selected as the source dataset, and the results include both target datasets. As illustrated, the majority of erroneous predictions exhibit small edit distances from their original references, while a smaller proportion of erroneous predictions have larger edit distances. TABLE 5 provides distributions on the error types in incorrectly predicted sequences for the two target domains. In the target dataset 'Sequencing_data _first_dimension [36]', substitution errors were the most prevalent. In contrast, for the target dataset 'SRR9701379 [37]', the distribution of the three types of errors was more evenly spread.

TABLE 4: Ablation study on domain alignment.

Target	Source	Success rate	
		with F	w/o F
Sequencing_data_first_dimension [36]	id20 [7]	**98.27%**	98.16%
	P10_5_BDDP210000009 [9]	**98.26%**	98.13%
	PE_AYB [4]	**98.33%**	98.07%
SRR9701379 [37]	id20 [7]	**97.62%**	97.34%
	P10_5_BDDP210000009 [9]	**97.62%**	97.34%
	PE_AYB [4]	**97.53%**	97.30%

TABLE 5: Error type distribution for incorrect predictions using 'id20 [7]' as the source dataset.

Target	Insertion	Deletion	Substitution
Sequencing_data _first_dimension [36]	0.26	0.26	0.48
SRR9701379 [37]	0.35	0.35	0.30

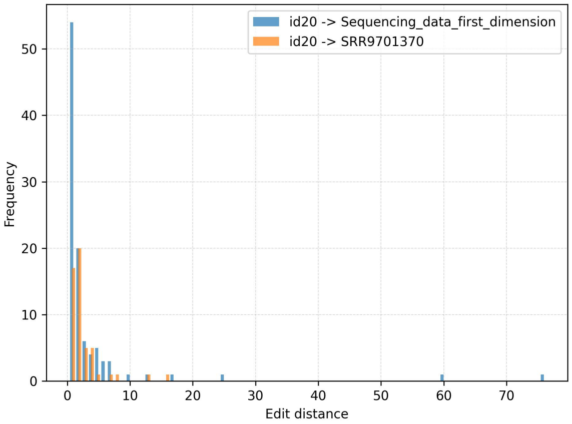


Fig. 3: Frequency histograms of the edit distances between erroneous predictions and their corresponding references.

3.7 Latent feature analysis

To validate the effectiveness of positive knowledge transfer, Fig. 4 visualizes the latent features of the encoder E for the target dataset 'SRR9701379 [37]' alongside each of the three source datasets using UMAP embeddings [40], both *before* (in (a)-(c)) and *after* (in (d)-(f)) transfer learning. The UMAP method [40] reduces the high-dimensional features extracted by the encoder to two dimensions for easier visualization.

Furthermore, we employ the inter-class distance to quantitatively evaluate the compactness of latent features between each pair of target-source datasets, *before* and *after* transfer learning. Given two classes of latent features $\mathcal{C}_S$ and $\mathcal{C}_T$ corresponding to source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$, respectively, the inter-class distance between them is defined by

$$d_{inter-class}(\mathcal{C}_S, \mathcal{C}_T) = \frac{1}{|\mathcal{C}_S| \cdot |\mathcal{C}_T|} \sum_{\forall x \in \mathcal{C}_S, \forall y \in \mathcal{C}_T} d(x, y), \quad (17)$$

where $|\mathcal{C}_i|$ denotes the number of samples in class $\mathcal{C}_i$, for $i = S, T$, and $d(\cdot, \cdot)$ represents the Euclidean distance between latent featrues x and y. These latent features, with dimensions 64×1, are output by the encoder E (of dimension $64 \times k_i$) followed by a pooling operation across sequence dimension k_i. TABLE 6 reports the inter-class distances between the target dataset 'SRR9701379 [37]' and three source datasets *before* and *after* transfer learning, as well as the relative reduction percentage (RRP).

Both visually and quantitatively, all three pairs of target-source datasets exhibit feature alignment after transfer learning, leading to more convergent feature distributions between the two domains. Specifically, before transfer learning, the target dataset is significantly dissimilar to the original representations of the source datasets 'id20 [7]' and 'P10_5_BDDP210000009 [9]', indicating a pronounced effect of distribution alignment achieved through transfer learning. On the other hand, as the original distributions of the target dataset 'SRR9701379 [37]' and the source dataset 'PE_AYB [4]' are similar, the feature alignment effect of transfer learning is less pronounced.

TABLE 6: Inter-class distances and relative reduction percentages (RRP) between the target dataset 'SRR9701379 [37]' and three source datasets, before and after transfer learning.

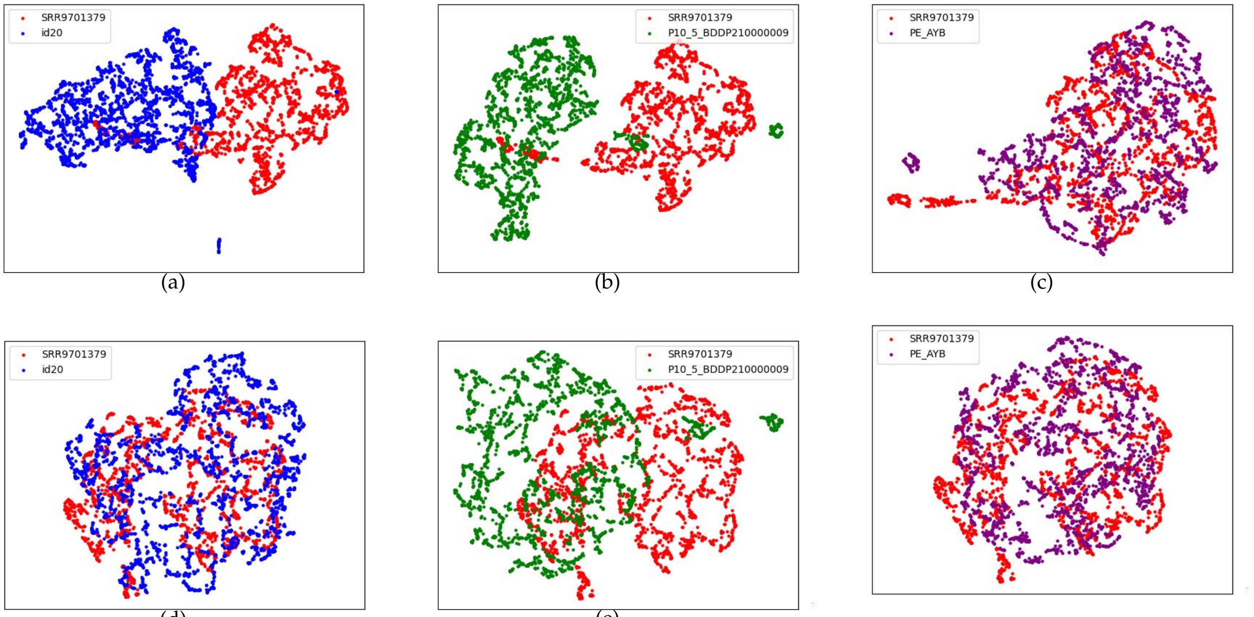
Source	Inter-class distance		
	Before	After	RRP
id20 [7]	7.96	2.38	70.1%
P10_5_BDDP210000009 [9]	20.93	13.40	36.0%
PE_AYB [4]	16.04	14.69	8.4%

3.8 Parameter sensitivity

We investigate the sensitivity of the parameter α in (15), which balances the trade-off between the reconsturction losses and the MMD loss in TransDNA. Candidate value set $\alpha = \{0.01, 0.05, 0.1, 0.5, 1, 1.5, 2\}$ is considered. The source dataset chosen is 'id20 [7]', and the target datasets are 'Sequencing_data_first_dimension [36]' and 'SRR9701379 [37]'. As illustrated in Fig. 5, the success rate on both target datasets initially increases slightly before decreasing, with a peak around $\alpha = 0.5$. Therefore, we empirically set $\alpha = 0.5$ in our experiments.

4 Discussion

In this study, we proposed TransDNA, a transfer learning-based sequence reconstruction model for DNA storage. By utilizing the principles of transfer learning, we alleviated the issue of training sample scarcity in deep learning-based sequence reconstruction algorithms, providing an alternative



(a) (b) (c)

(d) (e) (f)

Fig. 4: Latent features visualization using UMAP embeddings, *before* and *after* transfer learning. Sampled from target dataset ‘SRR9701379 [37]’ (red) and each source dataset, namely ‘id20 [7]’ (blue), ‘P10_5_BDDP210000009 [9]’ (green), and ‘PE_AYB [4]’ (purple).

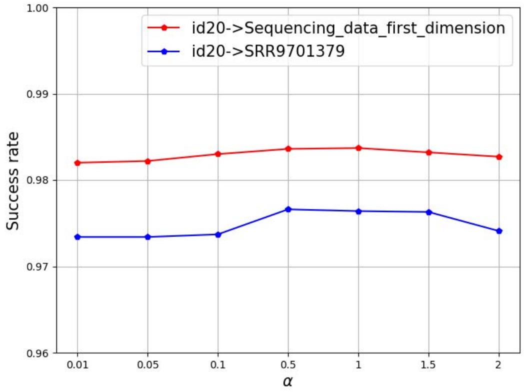


Fig. 5: Effect of the parameter α on the performance of TransDNA. Success rates for transferring the source dataset ‘id20 [7]’ to target datasets ’Sequencing_data_first_dimension [36]’ (red) and ‘SRR9701379 [37]’ (blue) are shown.

solution to synthetic training data strategy using SDG [19], particularly in situations where training data is limited.

TransDNA leveraged the power of transfer learning and domain adaptation to align the data distributions between the source and target domains. By minimizing the MMD loss between domain-invariant features extracted from the two domains, TransDNA effectively mitigated the distribution discrepancy and facilitated positive knowledge transfer from the source domain to the target domain.

In addition to the domain-invariant feature extractor, the model included a Conformer block and an autoregressive LSTM, which enhanced its robustness and adaptability. The Conformer block adeptly captured position shifts caused by IDS errors, while the autoregressive LSTM enabled the generation of sequences of varying lengths.

Experiments with five public datasets showed that TransDNA outperformed several state-of-the-art sequence reconstruction methods, including the synthetic training data strategy using SDG [19]. Specifically, compared to a base model without transfer learning, TransDNA improved the sequence reconstruction success rate by up to 1.2% and 2.2% on two respective target datasets. These results highlight the effectiveness of transfer learning as an effective approach for mitigating the challenges of limited training samples in DNA sequence reconstruction tasks.

However, TransDNA has two limitations: it necessitates retraining with each new source-target dataset pair due to the need for model adaptation for optimal performance on novel data, and it is currently confined to single-source domain transfer learning. Future work will aim to extend TransDNA to multi-source transfer learning, integrating knowledge from multiple source datasets to improve both performance and robustness, and to more effectively tackle the challenges of information recovery in DNA storage. Additionally, the transfer learning techniques in TransDNA have potential applications beyond DNA sequence reconstruction, such as in quality control, variant detection, and mutation prediction in DNA sequencing. Applying these techniques is expected to enhance the efficiency and accuracy of these genomic tasks using large-scale sequencing datasets.

5 Conclusion

In conclusion, TransDNA presents a promising solution for sequence reconstruction in DNA storage by leveraging transfer learning. It not only addresses the scarcity of training data but also exhibits error correction capabilities for IDS errors. Our findings demonstrate the potential of transfer learning strategies to enhance the performance of deep learning models in DNA sequence reconstruction, particularly in scenarios with limited training samples.

References

[1] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.

[2] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using dna," *Nature Reviews Genetics*, vol. 20, no. 8, pp. 456–466, 2019.

[3] Y. Dong, F. Sun, Z. Ping, Q. Ouyang, and L. Qian, "Dna storage: research landscape and future prospects," *National Science Review*, vol. 7, no. 6, pp. 1092–1107, 2020.

[4] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[5] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on dna in silica with error-correcting codes," *Angewandte Chemie. International Edition*, vol. 54, no. 8, pp. 2552–2555, 2015.

[6] Y. Erlich and D. Zielinski, "Dna fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, 2017.

[7] 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*, vol. 36, no. 3, pp. 242–248, 2018.

[8] W. H. Press, J. A. Hawkins, S. K. Jones Jr, J. M. Schaub, and I. J. Finkelstein, "Hedges error-correcting code for dna storage corrects indels and allows sequence constraints," *Proceedings of the National Academy of Sciences*, vol. 117, no. 31, pp. 18 489–18 496, 2020.

[9] L. Song, F. Geng, Z.-Y. Gong, X. Chen, J. Tang, C. Gong, L. Zhou, R. Xia, M.-Z. Han, J.-Y. Xu *et al.*, "Robust data storage in dna by de bruijn graph-based de novo strand assembly," *Nature Communications*, vol. 13, no. 1, p. 5361, 2022.

[10] C. Rashtchian, K. Makarychev, M. Rácz, S. D. Ang, D. Jevdjic, S. Yekhanin, L. Ceze, and K. Strauss, "Clustering billions of reads for dna data storage," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., 2017, p. 3362–3373.

[11] J. Jeong, S.-J. Park, J.-W. Kim, J.-S. No, H. H. Jeon, J. W. Lee, A. No, S. Kim, and H. Park, "Cooperative sequence clustering and decoding for dna storage system with fountain codes," *Bioinformatics*, vol. 37, no. 19, pp. 3136–3143, 2021.

[12] O. Sabary, A. Yucovich, G. Shapira, and E. Yaakobi, "Reconstruction algorithms for dna-storage systems," *Scientific Reports*, vol. 14, no. 1, p. 1951, 2024.

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

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

[15] R. Sakogawa and 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.

[16] A. Lenz, I. Maarouf, L. Welter, A. Wachter-Zeh, E. Rosnes, and 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.

[17] R. Shibata, G. Hosoya, and 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.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., 2017, p. 6000–6010.

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

[20] Y. Nahum, E. Ben-Tolila, and L. Anavy, "Single-read reconstruction for dna data storage using transformers," *arXiv preprint arXiv:2109.05478*, 2021.

[21] Y. Qin, F. Zhu, B. Xi, and L. Song, "Robust multi-read reconstruction from noisy clusters using deep neural network for dna storage," *Computational and Structural Biotechnology Journal*, 2024.

[22] V. Zhirnov, "Semiconductor synthetic biology roadmap," *Semiconductor Research Corporation Durham*, 2018.

[23] G. Chaykin, N. Furman, O. Sabary, and E. Yaakovi, "Dna storage similator," 2021. [Online]. Available: https://github.com/oyerush/DNASimulator

[24] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[25] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, 2021.

[26] H. Luo, C. Chen, W. Shan, P. Ding, and L. Luo, "ienhancer-bert: A novel transfer learning architecture based on dna-language model for identifying enhancers and their strength," in *Intelligent Computing Theories and Application*. Cham: Springer, 2022, pp. 153–165.

[27] H. Iuchi, T. Matsutani, K. Yamada, N. Iwano, S. Sumi, S. Hosoda, S. Zhao, T. Fukunaga, and M. Hamada, "Representation learning applications in biological sequence analysis," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 3198–3208, 2021.

[28] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*. Cham: Springer, 2021, pp. 877–894.

[29] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proceedings of Interspeech 2020*. ISCA, 2020, pp. 5036–5040.

[30] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 2978–2988.

[31] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," *arXiv preprint arXiv:1906.02762*, 2019.

[32] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[33] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[34] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*. AAAI Press, 2019.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

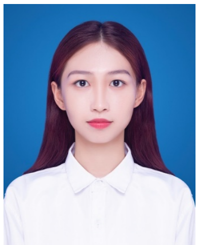
[36] C. Pan, S. K. Tabatabaei, S. H. Tabatabaei Yazdi, A. G. Hernandez, C. M. Schroeder, and O. Milenkovic, "Rewritable two-dimensional dna-based data storage with machine learning reconstruction," *Nature Communications*, vol. 13, no. 1, p. 2984, 2022.

[37] Y. Choi, H. J. Bae, A. C. Lee, H. Choi, D. Lee, T. Ryu, J. Hyun, S. Kim, H. Kim, S.-H. Song *et al.*, "Dna micro-disks for the management of dna-based data storage with index and write-once–read-many (worm) memory features," *Advanced Materials*, vol. 32, no. 37, p. 2001249, 2020.

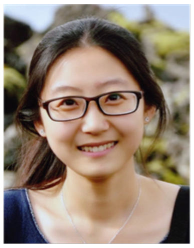
[38] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[39] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis, "Pear: a fast and accurate illumina paired-end read merger," *Bioinformatics*, vol. 30, no. 5, pp. 614–620, 2014.

[40] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

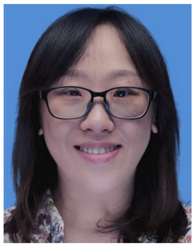


Yun Qin Yun Qin received the B.S. degree in Statistics from Xi'an Polytechnic University, Xi'an, China, in 2019. She is currently working toward the M.S. degree at Tianjin University, Tianjin, China. Her main interests include DNA storage and deep learning.



Fei Zhu Fei Zhu received the B.S. degree in mathematics and applied mathematics and in economics from Xi'an Jiaotong University, Xi'an, China, in 2011. She received the M.S. and the Ph.D. degrees in systems optimization and security from the University of Technology of Troyes (UTT), Troyes, France, in 2013 and 2016, respectively. She is currently an associate professor with the Center for Applied Mathematics, Tianjin University, Tianjin, China. Her research interests include nonlinear signal processing, hyperspectral image processing and DNA storage.

Bo Xi Bo Xi received the B.S. degree in mathematics and applied mathematics from Shanxi Normal University, Linfen, China, in 2020. She is currently working toward the M.S. degree at Tianjin University, Tianjin, China. Her main interests include multiple sequence reconstruction in DNA storage.



Yuping Duan Yuping Duan is a professor at School of Mathematical Sciences, Beijing Normal University, Beijing, China. She received her Ph.D. degree from the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore in 2012. From January 2012 to Sep- tember 2015, she worked as a research scientist at Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. Her research interests include mathematical image processing, inverse problems and machine learning.