

Learning Structurally Stabilized Representations for Lossless DNA Storage

Ben Cao^{1,2,4,*}, Xue Li^{3,*}, Tiantian He^{4,†}, Bin Wang³, Shihua Zhou³, Xiaohu Wu⁵, Qiang Zhang^{1,2,3,†}

¹School of Computer Science and Technology, Dalian University of Technology, Dalian, China.

²Key Laboratory of Social Computing and Cognitive Intelligence (Dalian University of Technology), Ministry of Education.

³Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, China.

⁴ Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore.

⁵ Beijing University of Posts and Telecommunications, China.

he.tiantian@a-star.edu.sg, zhangq@dlut.edu.cn

Abstract

This paper presents Reed-Solomon coded single-stranded representation learning (RSRL), a novel end-to-end model for learning representations for lossless DNA data storage. In contrast to existing learning-based methods, RSRL is inspired by both error-correction codec and structural biology. Specifically, RSRL first learns the representations for the subsequent storage from the binary data transformed by the Reed-Solomon codec (RS code). Then, the representations are masked by an RS-code-informed mask to focus on correcting the burst errors occurring in the learning process. The synergy of RS masks and graph attention enables active error localization, breaking through the limitations of traditional passive error correction. With the decoded representations with error corrections, a novel biologically stabilized loss is formulated to regularize the data representations to possess stable single-stranded structures. By incorporating these novel strategies, RSRL can learn highly durable, dense, and lossless representations for subsequent storage tasks in DNA sequences. The proposed RSRL has been compared with a number of baselines in real-world tasks of multi-type data storage. The experimental results obtained demonstrate that RSRL can store diverse types of data with much higher information density and durability, but much lower error rates.

1 Introduction

DNA storage has become one of the most promising technical solutions for coping with data explosion (Ping et al. 2022). Compared with conventional storage techniques, DNA storage utilizes DNA molecules as a storage medium to read and write data (Nguyen et al. 2021).

Although benefiting from the emergence of modern information and biotechnology (Carmean et al. 2019), DNA storage still suffers from the critical bottlenecks of cost and latency compared with electromagnetic storage media. Recently, several models for DNA coding and decoding have been developed to learn compact data representations that can improve the base utilization while reducing latency (Cao et al. 2025). These approaches can be categorized into

two classes, i.e., coding-theory-based and learning-based approaches. Methods based on coding theory (Grass et al. 2015; Goldman et al. 2013; Anavy et al. 2019; Nguyen et al. 2021) are designed to strictly follow certain coding systems. Thus, they have high storage capacity ratios. However, these approaches are computationally demanding when dealing with large-scale data (Zhou et al. 2024; Bi et al. 2025a). In contrast, learning-based approaches adopt heuristic search algorithms (Lochel et al. 2021) or neural networks (He, Ong, and Bai 2021; He et al. 2024; Bi et al. 2025b) to acquire an optimized encode/decoder that can write/read data stored in DNA. Although effective to some extent, these learning-based approaches always suffer from limited base utilization. Besides, they lack sufficient biological constraints during training, which can compromise data integrity.

In this paper, we hypothesize that the coalescence of contemporary learning models and stable traits of biomolecular structures in DNA can overcome the previously mentioned challenges confronted by existing learning-based approaches for DNA storage. To this end, we present Reed-Solomon coded single-stranded representation learning (RSRL), a novel model for learning representations for lossless DNA storage of multi-type data. To develop RSRL, we make the following two technical contributions. Firstly, inspired by the Reed-Solomon codec, we propose a novel data preprocessing and use the Fourier basis function and k-mer graph structure jointly capture the periodic stability characteristics of DNA for representation learning for DNA storage. Specifically, the representations are learned from binary data coded based on the Reed-Solomon codec. Then, the representations are masked by an RS-code-informed Mask to focus on correcting the burst errors occurring in the learning process. Secondly, with the decoded data representations with error corrections, we propose a novel biologically stabilized loss that regularizes the data representations to possess stable single-stranded structures. With the mentioned techniques incorporated into the training process, the data representations learned for the subsequent writing to DNA are highly durable, dense, and lossless. In our experiments, the proposed RSRL has been compared with several strong baselines in real-world storage tasks of diverse data types. The results demonstrate that RSRL achieves a

*Co-first author, these authors contributed equally

†Corresponding author

notable reduction in learning complexity, with an 18% increase in net information density and an 11% improvement in thermodynamic performance. Additionally, our approach reduces coding and decoding delays by more than two orders of magnitude, showcasing a significant advancement in the field of DNA storage. The main contributions of this paper are summarized as follows:

- We have verified that consideration of stable traits of biomolecular structures and error-correction codec can significantly improve representation learning for DNA storage.
- To the best of our knowledge, we initialize the first attempt to formulate the loss function that can guide the proposed RSRL to learn representations for the subsequent DNA storage tasks that possess stable single-stranded structures. Such formulated loss enables RSRL to learn representations for durable, dense, and lossless DNA storage.
- The proposed RSRL has been compared with baselines in various real-world storage tasks. Experimental results show that RSRL can overcome the shortcomings of existing learning-based approaches, demonstrating that it is an effective and promising method for real-world DNA storage tasks.

2 Related work

Conventional DNA storage and coding Due to intrinsically bearing life information as a natural storage medium, DNA has become the most competitive alternative to silicon-based storage (Limbachiya, Gupta, and Aggarwal 2022; Wu et al. 2025; Liu et al. 2025). DNA storage can be divided into three main phases, including data writing (Rasool et al. 2023; Grass et al. 2015; Bögels et al. 2023), preservation, and reading (Cao et al. 2024; Organick et al. 2018; Bögels et al. 2023). Efficient and robust coding schemes not only improve coding efficiency but also ensure data integrity. However, due to the uncontrollability of biomolecules (Zhang et al. 2023) and the inherent errors during DNA synthesis and sequencing, the coding rate ($Coding_rate = information_bits/coding_bits$) was still distance from the theoretical upper limit. Existing approaches to DNA storage coding can be divided into coding theory-based and learning-based (Nguyen et al. 2021). Huffman coding is the earliest framework of coding theory that is used in DNA coding for storage purposes (Goldman et al. 2013; Li et al. 2025). It can completely avoid consecutive base repetitions. Subsequently, the Galois field and DNA codon wheel are combined for coding to avoid consecutive bases greater than three. Recently, Reed-Solomon coding (Blawat et al. 2016), prefix-synchronized coding (Yazdi et al. 2015), DNA Fountain (Erlich and Zielinski 2017), the ying-yang code (Ping et al. 2022), and Repeat-Accumulate coding (Wang et al. 2023) have also been considered for empirical DNA storage systems.

Learning-based DNA storage On the other hand, the main idea of learning-based methods is to compress data vectors using neural networks and then encode the compressed vectors for storage (Franzese et al. 2021; Guo and

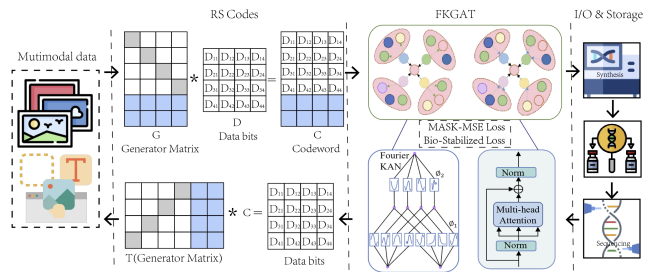


Figure 1: Overview of the proposed RSRL.

Qi 2021). One of the representatives is DNA-QLC (Zheng et al. 2024), which adopts over ten layers of CNNs to extract hidden information from images and encode it into DNA sequences using Levenshtein code. Recently, biological constraints have been considered when building learning-based models for DNA storage. For example, homopolymer and GC content have been used to build the loss functions (Wu et al. 2023), leveraging which a CNN-based encoder-decoder can learn image representations for DNA storage. Though effective, these learning-based approaches are either computationally demanding or only applicable for storing data that are tolerable for information loss (e.g., images). Differing from both traditional and learning-based approaches, the proposed RSRL considers coding theory and biologically stabilized structures when learning data representations. Regardless of data types, this novel strategy enables RSRL to learn highly durable, dense, and lossless representations for DNA storage.

3 The proposed method

In this section, we introduce RSRL, a novel end-to-end model for learning representations for lossless DNA storage (Fig. 1). RS codes is firstly used to process diverse types of data into a redundant binary data stream to correct errors during the learning process.

FKGAT Network

We introduce the Fourier-Kolmogorov Graph Attention Network (FKGAT), which replaces traditional MLP components with Fourier-Kolmogorov-Arnold Networks (FKAN) to reduce learnable parameters while enhancing relationship learning in DNA fragment graphs. Given a graph $G = (V, E, \mathbf{X})$ where V is the node set (DNA fragments), E is the edge set (fragment connections), and $\mathbf{X} \in \mathbb{R}^{N \times D}$ are initial node features (N nodes with D -dimensional features), FKGAT employs two separate FKAN modules: one for node feature transformation and one for attention score computation.

Node Feature Transformation For each node i , we transform its raw feature vector $\mathbf{x}_i \in \mathbb{R}^D$ to a latent representation $\mathbf{h}_i = \phi_F^{\text{node}} \in \mathbb{R}^{d_{\text{out}}}$ using an FKAN module ϕ_F^{node} :

$$\phi_F^{\text{node}}(\mathbf{x})_j = \sum_{i=1}^d \sum_{k=1}^g (\cos(kx_i) \cdot a_{ijk}^{\text{node}} + \sin(kx_i) \cdot b_{ijk}^{\text{node}}). \quad (1)$$

Here, d is the input dimension, $j = 1, \dots, d_{\text{out}}, d_{\text{out}}$ is the output dimension, g is a frequency grid hyperparameter (empirically set to $g = 2$), and $a_{ijk}^{\text{node}}, b_{ijk}^{\text{node}} \in \mathbb{R}$ are learnable Fourier coefficients specific to the node transformation.

Attention Score Computation For a pair of nodes i and j connected by an edge with feature vector $\mathbf{e}_{ij} \in \mathbb{R}^m$, we compute an attention score using another FKAN module ϕ_F^{attn} . This module takes the concatenation of the transformed node features and the edge feature and outputs a scalar:

$$\text{atn}_{ij} = \phi_F^{\text{attn}}(\mathbf{h}_i \parallel \mathbf{h}_j \parallel \mathbf{e}_{ij}) \quad (2)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\text{atn}_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\text{atn}_{ik}))} \quad (3)$$

Here, $\phi_F^{\text{attn}}: \mathbb{R}^{2d_{\text{out}}+m} \rightarrow \mathbb{R}$ is defined as:

$$\phi_F^{\text{attn}}(\mathbf{z}) = \sum_{i=1}^{d_{\text{in}}^{\text{attn}}} \sum_{k=1}^g (\cos(kz_i) \cdot a_{ik}^{\text{attn}} + \sin(kz_i) \cdot b_{ik}^{\text{attn}}), \quad (4)$$

where $\mathbf{z} \in \mathbb{R}^{d_{\text{in}}^{\text{attn}}}$ with $d_{\text{in}}^{\text{attn}} = 2d_{\text{out}} + m$, and $a_{ik}^{\text{attn}}, b_{ik}^{\text{attn}} \in \mathbb{R}$ are learnable parameters for the attention module.

Multi-Head Attention with Residual Connection We use K parallel attention heads. For each head k , the updated node feature is computed by aggregating the neighborhood information and adding a residual connection:

$$\mathbf{h}_i'^{(k)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{h}_j + \mathbf{h}_i. \quad (5)$$

Note that each head has its own attention parameters (i.e., independent $\phi_F^{\text{attn},k}$ for each head).

The outputs of all heads are concatenated, and then a residual connection with a projection matrix is applied to handle dimension mismatch:

$$\mathbf{h}_i' = \left(\parallel_{k=1}^K \mathbf{h}_i'^{(k)} \right) + \mathbf{W} \mathbf{h}_i, \quad (6)$$

where $\mathbf{W} \in \mathbb{R}^{K \cdot d_{\text{out}} \times d_{\text{out}}}$ is a learnable projection matrix.

Edge Feature Update For each edge (i, j) , we update its feature by first projecting the original edge feature \mathbf{e}_{ij} to each head's space and then combining with the updated node features from that head:

$$\mathbf{e}_{ij}' = \parallel_{k=1}^K \left(\mathbf{h}_i'^{(k)} + \mathbf{h}_j'^{(k)} + \mathbf{e}_{ij}^{(k)} \right) \quad (7)$$

Representation learning and codec

For any file of size W , it is first converted to a binary matrix $\mathbf{X} \in \mathbb{R}^{M \times 48}$, where $M = \lceil W/48 \rceil$, and then row-wise encoded using a RS code RS(64, 48) in $\text{GF}(2^8)$ to generate an error-corrected binary stream $\mathbf{C} \in \mathbb{R}^{M \times 64}$.

To construct graph-structured data: (1)**K-mer Segmentation**: Divide \mathbf{C} into 4-bit chunks (each $\in \{A, C, G, T\}$). (2)**Node Definition**: Generate 4-mers ($k_{\text{mer}} = 4$) by sliding a window of size 4 with step 1. Each 4-mer (16-bit vector) becomes a node $v_i \in V$ with feature $\mathbf{x}_i \in \mathbb{R}^{16}$. (3)**Edge**

Construction: Connect consecutive 4-mers with edges (implicit overlap k_{lap}), and add self-loops to all nodes.

Specifically, RS codes adopted in this paper are typically defined in Galois fields, given a finite field F and polynomial ring $F[x]$, where n and k satisfy $1 \leq k \leq n \leq |F|$. n distinct elements selected from F , are denoted as $\{x_1, x_2, \dots, x_n\}$. The codeword \mathbf{C} is obtained by computing the values of the polynomials in $F[x]$ such that the order of each x_i in F is less than k :

$$\mathbf{C} = \{(f(x_1), \dots, f(x_n)) \mid f \in F[x], \deg(f) < k\}. \quad (8)$$

So \mathbf{C} is an $[n, k, n - k + 1]$ code, which is also a linear code in F of length n , dimension k , and minimum Hamming distance $n - k + 1$. Thus, any dimensionally matched binary matrix of the file to be stored can be RS encoded according to \mathbf{C} to get a binary data stream with error-correction redundancy. Then binary data stream is used as the input sequences $Y = \{y_1, y_2, \dots, y_n\}$, where y_i represents the data (representation) i in some GNN layer, the output representations that will be either fed into the loss functions or passed to next layers, are learned through the KAN (Liu et al. 2024) and self-attention mechanism. After training with input data that have been encoded by the RS codec, RSRL is able to learn the compressed representations of data, which are then encoded as sequences of nucleobase in DNA.

DNA sequences (transcoded representation) should be biostable, which is required to integrably preserve the information carried by the low-dimensional representations. To improve learning efficiency, complex encoding methods are not suitable for use during training. A widely accepted method for encoding representations to nucleobase is to directly map $00 - A, 01 - T, 11 - G, 10 - C$. However, this coding method is prone to homopolymers. Additional constraints on homopolymers are required by this coding method, thus increasing the complexity of the encoding. In this paper, we propose a novel block mapping strategy for representations for DNA storage encoding. Specifically, we view two bases as a community and encode the four-bit representation into two bases at once. This strategy can minimize the generation of homopolymers. Results presented in Section 4 validate its effectiveness.

Biologically stabilized loss functions

Existing loss functions adopted by previous learning-based models fail to guide a learning model to achieve lossless DNA storage as they do not consider factors of biological stability. Inspired by the single-stranded structure in RNA and RS codec, we propose to formulate biologically stabilized loss functions that can guide the learned representations to possess the stable structures like bio-molecules have, thus achieving highly durable, information-dense, and lossless storage in DNA.

Synergizing RS codes with MASK-MSE loss The primary purpose of data storage is to ensure the consistency of data reading and writing. Naturally, mean squared error (MSE) is a widely accepted loss function that quantifies the average deviation between the reconstructed binary data stream and the original data. However, conventional MSE

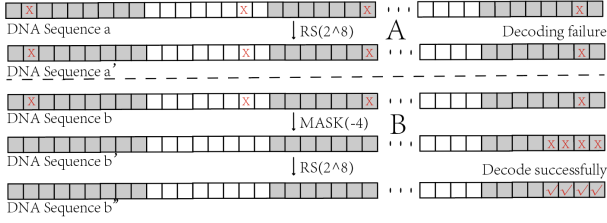


Figure 2: MASK-MSE loss Maximizes the potential of RS error correction codes.

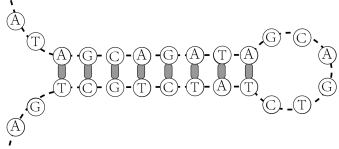


Figure 3: The Hair structure.

cannot fully address the errors caused by information loss during representation learning. Moreover, to ensure data integrity, RSRL incorporates RS codes as an error-correction measure that is good at handling burst errors (a series of adjacent errors). In contrast, random errors (random single-bit errors) generally exist in DNA storage process. Therefore, to fully utilize the error-correction capability of the RS codec, random errors in DNA storage have to be transformed into burst errors. To address these two issues, we introduce an additional mask operation to conventional MSE loss, and thus propose MASK-MSE loss based on RS codes, which guides the reduction of learning efficiency for a specific integer block of the current tensor during the learning process, concentrating errors within this block. Let Z denote the ground truth data matrix, \hat{Z} represents the predicted matrix (i.e., the output representation learned from Y), M and N denote the mask matrix, and the total number of elements in the matrix, i and j are the indices of the elements in the matrix. $M_{i,j}$ is the mask value at *position*(i, j). If $M_{i,j} = 1$, the loss is computed at the corresponding position. Otherwise, the loss regarding $Z_{i,j}$ is masked out:

$$\mathcal{L}_{\text{MASK-MSE}} = \frac{1}{N} \sum_{i,j} M_{i,j} \cdot (Z_{i,j} - \hat{Z}_{i,j})^2. \quad (9)$$

As backpropagation is deactivated when $Z_{i,j}$ is masked out, the RS codec will dominate the encoding process of $Z_{i,j}$, thus correcting potential errors during training. In this paper, we set the mask size of each representation as 8, which is the same as the block size of the RS codes adopted by the proposed RSRL. With the proposed MASK-MSE loss, dispersed random errors are transformed into burst errors, which RS is adept at handling (Fig. 2).

Learning single-stranded representations It is known that DNA and RNA with stable structures can generally carry genetic information with minimized errors in transcription and translation (Leppek et al. 2022). In this paper, we introduce single-stranded loss functions to endow the learned representations with the previously mentioned

properties of stable structures. Here, we formulate a single-stranded loss by mainly considering GC content and hairpin structure. The proposed single-stranded loss will be computed based on the DNA sequences (i.e., sequences of bases constituted by A, T, G, and C) that are transcoded from the learned representations (i.e., sequence l is \hat{Z}_l in \hat{Z} after transcoding). Let $\mathcal{G}(l)$ and $\mathcal{H}(l)$ denote the GC content and hairpin structure of the sequence l , \mathcal{G}^* and \mathcal{H}^* denote the target values of $\mathcal{G}(l)$ and $\mathcal{H}(l)$, which are 50% and 0, respectively. Therefore, our goal is to formulate a loss that can minimize the difference between $\mathcal{G}(l)$ and \mathcal{G}^* , and that between $\mathcal{H}(l)$ and \mathcal{H}^* .

It is known that keeping the bases evenly distributed is beneficial to the stability of the DNA sequence (Ping et al. 2022). Thus, we consider GC content as one of the learning objectives in the proposed single-stranded loss and aim to minimize $\mathcal{G}(l)$ to \mathcal{G}^* , i.e., 50%. Accordingly, the GC content of each DNA sequence transcribed from the corresponding representation is computed as the following:

$$\mathcal{G}(l) = \frac{|G| + |C|}{|G| + |C| + |A| + |T|} \times 100\%, \quad (10)$$

where $|\cdot|$ is defined as the sum of the number of bases in each DNA sequence.

A hairpin structure forms a hairpin-like shape in which two base pairs are bonded together by hydrogen bonds to form a loop (Fig. 3), which increases the error rate in reading and replicating DNA storage data (Li et al. 2020). Therefore, we aim to form a loss that can minimize the hairpin structure in learned representations. Hairpin structures have two important parameters, i.e., the minimum stem region S_{min} and the minimum ring region R_{min} . In calculating the probability of forming a hairpin structure of different sizes at each position of the sequence, the first consideration is the result of forming a hairpin at position i with R ring and S stem region. A hairpin structure is considered to be formed if more than half of $l_{i-s} \cdots l_i$ and $l_{i+r} \cdots l_{i+r+s}$ are hybridized. Accordingly, for each sequence transcribed from the learned representation, the number of existing hairpin structures can be computed as follows:

$$\mathcal{H}(l) = \sum_{s=S_{min}}^{(L-R_{min})/2} \sum_{r=R_{min}}^{L-2s} \sum_{i=1}^{L-2s-r} T \left(\sum_{j=1}^s bp(l_{s+i-j}, l_{s+i+r+j-1}), \frac{s}{2} \right), \quad (11)$$

where s is the stem length, S_{min} is the set minimum stem length. r is the ring length, R_{min} is the set minimum ring length, and L denotes the length of the DNA sequence. $T(\cdot)$ is the threshold function. Only when $m > n$, $T(m, n) = m$, otherwise = 0. The threshold is generally taken as 0.5, i.e., only when the consecutive matches reach half of the stem region is regarded as the generation of the hairpin structure. $bp(j, k)$ function indicates the number of bases complementary to each other at positions j and k in the DNA sequence (1 for complementary and 0 otherwise). The ranges of values of the relevant parameters are as follows. The maximum

length of the stem region $(L - R_{min})/2$ is obtained at R_{min} , and the maximum length of the loop region $L - 2 * S_{min}$ is obtained at S_{min} . i denotes the index at the beginning of the DNA sequence, where i starts at one at the minimum, and the maximum could be up to $L - 2s - r$. Given Eqs. (10) and (11), the proposed single-stranded loss function can be formulated as the following:

$$\mathcal{L}_{BC} = \frac{1}{m} \sum_{l=1}^m d(\mathcal{G}(l), \mathcal{G}^*)^2 + \frac{\beta}{m} \sum_{l=1}^m d(\mathcal{H}(l), \mathcal{H}^*)^2, \quad (12)$$

where $d(\cdot)$ is the Euclidean distance between two items. Accordingly, the biologically stabilized loss function of RSRL for lossless DNA storage is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MASK-MSE}} + \alpha \mathcal{L}_{BC} \quad (13)$$

With the loss function shown above, the proposed RSRL can learn representations with stable structures also possessed by biomolecules. Besides, the representations for data to store are learned by RSRL in an end-to-end manner and achieve lossless DNA storage.

4 Experimental evaluation

In this section, we validate the effectiveness of the proposed RSRL by comparing it with strong baselines on real-world tasks of data storage. Besides, the unique properties of the proposed approach are also revealed by ablation studies.

Compared baselines

We compare RSRL with nine strong baseline approaches, which can be divided into two categories according to the used coding methods. Church (Church, Gao, and Kosuri 2012), Goldman (Goldman et al. 2013), Grass (Grass et al. 2015), Blawat (Blawat et al. 2016), DNA Fountain (Erlich and Zielinski 2017), Yin-Yang (Ping et al. 2022), and HL-DNA (Li et al. 2022) are coding theory-based DNA storage methods. DJSCC (Wu et al. 2023) and DNA-QLC (Zheng et al. 2024) are learning-based DNA storage approaches.

DNA storage tasks and experimental settings

DNA storage tasks Due to the cost of DNA storage, current baselines are often experimented at KB/MB data volume levels (Erlich and Zielinski 2017; Ping et al. 2022). Following the data volume settings of previous studies, we evaluate the storage performance of all approaches using five files of diverse types, including images, PDFs, and text files. For fair comparisons, all the experiments are conducted at the binary data level. Thus, the file type has basically no effect on the performance, except in the case of lossless reading and writing. In the main content of this paper, we report the results regarding DNA storage for image files.

Experimental settings To fulfill the task of multi-type DNA storage, the proposed RSRL performs $RS(64, 48)$ to pre-coding in the $GF(2^8)$ field. The input dimension of the RS encoder is $M * 48$. The input files are first converted to matrix form, and the output dimension is $M * 64$ after being coded by RS. After reshaping the dimension of the file matrix to $N * 32 * 64$ vector, it serves as input to a FKGAT

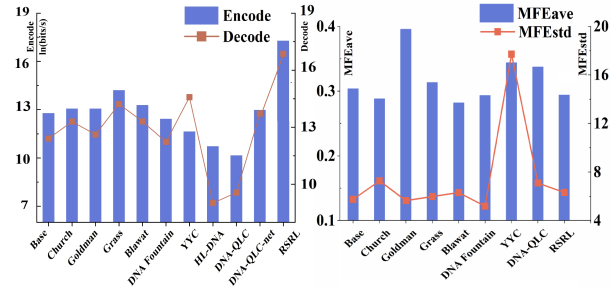


Figure 4: Comparison of Encoding Speed between RSRL and other baselines(left), and comparison of mean and standard deviation of MFE between RSRL and other baselines(right).

with two layers and four heads, which will learn representations for the subsequent DNA storage tasks. Hyperparameters α and β in biologically stabilized loss functions are set to 16.67 and 0.058, respectively, determined through cross-validation. As for the settings of compared baselines, we use the ones recommended in previous studies.

Evaluation metrics

In our experiments, data consistency can be directly evaluated by checking whether the learning and DNA encoding process is lossless. Data read/write efficiency is evaluated by encoding methods, net information density (NID, $Net_Information_Density = information_bits / (coding_base + ECC_base)$), error rates, and coding speed. As for the metrics of stability, we use minimum free energy (MFE) and melting temperature (T_m) to evaluate all approaches in our experiment. These evaluation metrics can comprehensively reveal the performances of all approaches.

Comprehensive analysis of DNA storage performance

We compare the overall performance of the DNA storage obtained by RSRL and other advanced approaches. As the table shows, RSRL demonstrates a significant advantage in NID (net information density) compared to lossless coding theory-based methods. Compared to Goldman, RSRL achieved an 18% improvement in Net information density. Although learning-based approaches like DNA-QLC may obtain a higher net information density, they are not applicable for data storage as their representation learning is not lossless. Besides, DNA-QLC and DJSCC are computationally demanding as they stack many convolution layers for learning data representations. The proposed RSRL is the only learning-based model that can efficiently learn lossless representations with desirable net information density. And RSRL is the only learning-based model applicable for storing diverse types of data in DNA.

Regarding data loss, both DJSCC and DNA-QLC use convolution neural networks (CNNs) to compress the input image. It is known that CNNs may result in information loss. Therefore, in Table 1, the lossless ratios for DJSCC

Method	Coding strategy	ECS	NID	GC (%)	HL	NP	File type	Lossless	Model complexity	Capacity
Church	Direct mapping	-	0.94	39–61	3	×	All Type	1.00	$O(n * m)$	-
Goldman	Ternary Huffman	Repetite	1.48	39–60	1	×	All Type	1.00	$O(n * (m + \log_m))$	-
Grass	Galois +Rotation	RS	1.56	36–62	3	×	All Type	1.00	$O(n * m + \log_m)$	-
Blawat	Segment mapping	RS	1.40	24–60	3	×	All Type	1.00	$O(n * m)$	-
DNAF	DNA Fountain	Fountain	1.23	39–62	4	×	All Type	1.00	$O(n * m)$	-
Yin-Yang	Yin-yang	RS	1.36	40–60	4	×	All Type	1.00	$O(n * m)$	-
HL-DNA	Quater-mapping	Barrier	1.85	51	''AA''	✓	Image	0.896	$O(n * m)$	-
DJSCC	CNN	-	-	50.0±5.0	~5	×	Image	0.841	1.76/0.24	5.4E4
DNA-QLC	Conv+VAE	LC	2.90	50.0±0.0	2	×	Image	0.926	1812/13.5	3.5E5
RSRL	FKGAT	RS&MASK	1.75	50.0±0.3	~3	✓	All Type	1.00	0.098/0.19	4.9E4

Table 1: Performance comparisons of approaches to coding image files into DNA

and DNA-QLC are much lower than 1.00, which are 0.841 and 0.926, respectively. In contrast, the proposed RSRL achieves lossless storage of data, as shown in Table 1 (1.00 for the lossless ratio). Obtaining such results is mainly because RSRL is the first learning-based approach to DNA storage incorporated with Reed-Solomon (RS) coding as an error correction strategy. Moreover, we further propose the MASK-MSE loss based on RS coding, which converts random errors that are difficult for RS to handle into burst errors (Fig. 2B), thereby maximizing the error correction potential of RS codes.

From Table 1, we also observe that RSRL supports high NID of all file types while keeping GC content balanced and pair-free, demonstrating that the proposed RSRL can make a good balance between the performance of DNA storage and indicators of biological stability. This is because RSRL additionally adopts the proposed single-stranded loss functions based on structural biology, achieving constraint satisfaction through a learning approach and overcoming the limitations of previous learning-based approaches (e.g., DJSCC) in terms of the number and accuracy of constraints.

Encoding speed directly impacts read-write latency in DNA storage. The comparisons of encoding speed between RSRL and other baselines are depicted in Fig. 4, where the proposed RSRL demonstrates the highest speed of encoding data for DNA storage. RSRL can encode more data per unit of time than other baselines because it adopts a lightweight network structure. In our experiments, we additionally design DNA-QLC-net, which is a variant of DNA-QLC and only records the time cost by the neural network. As the figure shows, DNA-QLC-net is still much slower than the proposed RSRL due to its complex network structure.

Thermodynamic comparisons

Thermodynamic changes can better reflect the essence of biochemical reactions, consistently interweaving with biochemical reactions, thus more directly manifesting the stability and performance of DNA sequences for storage purposes. In DNA storage, DNA sequences can be evaluated based on thermodynamic properties such as free energy, melting temperature, and GC content. So, we compare RSRL with other baseline methods in terms of minimum Gibbs free energy, melting temperature, GC content, and local GC content. The results are presented in Figs. 4-6.

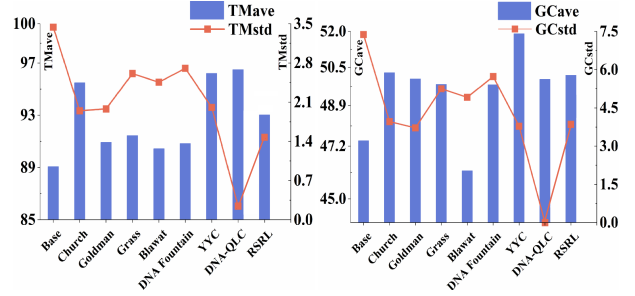


Figure 5: Comparison of mean and standard deviation of Tm between RSRL and other baselines (left), and comparison of mean and standard deviation of GC content between RSRL and other baselines (right).

Minimum free energy Energy changes directly reflect the intrinsic variations in biochemical reactions, thus indicating the stability of DNA sequences for data storage. In our experiment, we use Gibbs standard free energy (ΔG), a widely accepted indicator to reflect energy changes (Wang et al. 2018). In Fig. 4, the corresponding results of minimum Gibbs free energy are depicted. As shown in Fig. 4, RSRL exhibits a smaller MFEave compared to Goldman, Grass, Yin-Yang, and DNA-QLC. A lower MFE indicates that the DNA sequences encoded by RSRL are more stable. While the MFEave of RSRL is similar to Church and Blawat, its MFEstd is more advantageous, indicating that the quality of the DNA sequences encoded by RSRL tends to be stable, with less influence from outliers. Particularly compared to DNA-QLC (the learning-based method), both MFEave and MFEstd obtained by RSRL are lower by over 11%, signifying the superior performance of the network model and biologically stabilized properties adopted by RSRL.

Melting temperature and GC content Throughout the entire process of DNA storage, factors such as DNA assembly, PCR amplification, and storage stability necessitate that the melting temperature (Tm) (Li et al. 2020) and GC content of oligonucleotides exhibit minimal deviation. Therefore, we compare the Tm and GC content of RSRL with other baseline methods. When Tm fell within the range of (85-95), particular attention is given to the standard deviation of Tm (Tmstd). Smaller Tmstds ensure smooth data reading and writing processes in DNA storage. In Fig. 5,

		Errors bits=1	Coding rate
RS codes	-	6.32E-4, 1	0.875
LDPC-d.c_30	d.v=3	1.9E-3, 5E3	0.9133
	d.v=4	1.9E-3, 5E3	0.8866
	d.v=5	3.9E-3, 5E2	0.86
LDPC-d.c_12	d.v=6	0.5416	
RSRL	-	2.69E-5, 1	0.875

Table 2: Comparisons of decoding time of different error bits and coding rates obtained by LDPC, RS, and RSRL

Method	Reconstruct rate	MFEave	Tmave
No GC	100%	-28.8	91.37
No pair	100%	-26.7	92.22
No GC&pair	100%	-24.7	91.03
No MASK	93.42%	-18.72	85.86
No GC&p&M	98.76%	-20.66	88.90
RSRL	100%	-34.49	94.49

Table 3: Ablation study of RSRL

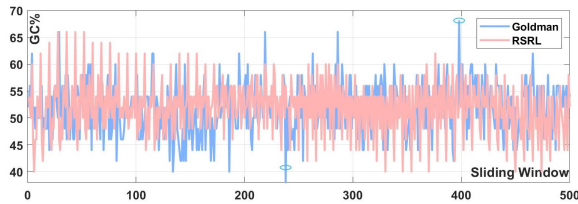


Figure 6: Distribution of local GC content under a sliding window.

the Tmstd of RSRL is shown to be significantly lower than that of other coding theory-based schemes. The Tmstd of RSRL is higher than that of DNA-QLC, which may result from DNA-QLC’s time-consuming hard coding after completing the learning of representations, limiting the GC content to 50%, as shown in Fig. 5. However, directly setting GC content as 50% does not give rise to a significant boost of performance, compared to 48-50%, which has a negligible effect in DNA storage (Yang et al. 2020; Ping et al. 2022). So, the optimal value of GC content is generally considered as around 50%. In Fig. 5, we compare the average (GCave) and variance (GCstd) of GC content, showing that all methods maintain GC content within the range of 46-52%, essentially meeting the GC content constraint and validating the effectiveness of the encoding strategy.

The GC content is calculated by considering the entire DNA sequence as the smallest unit in Fig. 5, which might overlook the impact of local GC content (Cao et al. 2022). Therefore, we additionally analyze local GC content, comparing RSRL with the Goldman method, which is most similar to RSRL in terms of global GCave and GCstd. From Fig. 6, it is evident that the local GC content of the proposed RSRL is smoother. While local GC content of Goldman have more outliers, potentially resulting in lower off-machine quality in Illumina sequencing data, and affecting the consistency of DNA storage data reading and writing.

Comparisons with communication codes

In this subsection, we compare the performance of the RS error correction codes that are adopted by the proposed RSRL with that of other Error-Correcting Codes (ECC) that are also available for learning-based approaches to DNA storage. To this end, we select LDPC codes as the main comparison of ECC in communication codes. Specifically, the parameters d_c, d_v of LDPC are selected as 30, 12 and 3, 4, 5, 6. $d_c = 30$ is because this setting closely matches the code rate (0.875) of RS codes in RSRL. And $d_c, d_v = (12, 6)$ can maximize the error correction capability of LDPC. We mainly analyze the relationship between error correction capability and the coding bits. In Table 2, we list the comparative results with respect to the decoding time (S) and iterations obtained by LDPC and RSRL. Based on the results listed in the table, RS codes are more desirable for the proposed RSRL for conducting DNA storage tasks of multi-type data.

Ablation study

In this subsection, we conduct ablation studies to show the effect of each module in the proposed RSRL on DNA storage tasks. Specifically, we systematically analyze the effects of MASK-MSE and biologically stabilized loss functions on RSRL performance. The result shows that all key modules of the proposed RSRL. i.e., the MASK-MSE and biologically stabilized loss functions, play critical roles in lossless DNA storage, manifesting the rationality of the design of RSRL. Each RS codeword: $n = 64, k = 48$, thus $t = \lfloor (n-k)/2 \rfloor = 8$. For $p_{bit} = 1\%$, $p_{sym} = 1 - (1 - 0.01)^8 \approx 0.0773$. RS success probability:

$$P_{RS.succ} = \sum_{i=0}^t \binom{n}{i} p_{sym}^i (1 - p_{sym})^{n-i}. \quad (14)$$

This gives $P_{RS.succ} \approx 0.943$, matching the observed 93.42% (Table 3, No MASK). When $p_{bit} \rightarrow 10^{-4}$, $P_{RS.succ} \rightarrow 1$, explaining observed lossless decoding.

Conclusion

In this paper, we have proposed Reed-Solomon coded single-stranded representation learning (RSRL), a novel end-to-end approach to learning representations for DNA storage. Unlike existing learning-based approaches to DNA storage, RSRL incorporates an error-correction codec and stable biological structures into the process of learning representations for data storage. Representations learned by RSRL possess remarkable structural properties like biomolecules in biont and are, therefore, highly durable, dense, and lossless for subsequent storage tasks. The proposed RSRL has been compared with both coding theory and learning-based methods for DNA storage. The obtained experimental results demonstrate that RSRL can outperform prevalent approaches in the tasks of representation learning for diverse types of data. In the future, we will further improve the proposed RSRL by identifying more efficient strategies to incorporate error-correction codes into neural networks and formulating more efficient biologically informed loss functions for model training.

Acknowledgments

This work is supported by 111 Center (No. D23006), the National Natural Science Foundation of China (Nos. 62272079, 62572088, 62502063), the National Foreign Expert Project of China (No. D20240244), Natural Science Foundation of Liaoning Province (Nos. 2024-MS-212, 2024-BS-267), Scientific Research Project of Liaoning Provincial Department of Education (No. LJ222411258005), Liaoning Revitalization Talent Program (No. XLYC2403039), the Artificial Intelligence Innovation Development Plan Project of Liaoning Province (No. 2023JH26/10300025), Joint Plan of Liaoning Province Science and Technology Plan (Nos. 2024JH2/102600064, 2024-MSLH-009), the Dalian Outstanding Young Science and Technology Talent Support Program (No. 2022RJ08), Dalian Major Projects of Basic Research (No. 2023JJ11CG002), the Dalian Young Science and Technology Star Program (No. 2023RQ056), the Interdisciplinary Project of Dalian University (Nos. DLUXK-2024-YB-001, DLUXK-2025-FX-003, DLUXK-2025-QNLG-003, DLUXK-2024-QN-002).

References

- Anavy, L.; Vaknin, I.; Atar, O.; Amit, R.; and Yakhini, Z. 2019. Data storage in DNA with fewer synthesis cycles using composite DNA letters. *Nature Biotechnology*, 37(10): 1229–1236.
- Bi, F.; He, T.; Ong, Y.-S.; and Luo, X. 2025a. Discovering Spatiotemporal–Individual Coupled Features From Non-standard Tensors—A Novel Dynamic Graph Mixer Approach. *IEEE Transactions on Neural Networks and Learning Systems*, 1–1.
- Bi, F.; He, T.; Ong, Y.-S.; and Luo, X. 2025b. Graph Linear Convolution Pooling for Learning in Incomplete High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, 37(4): 1838–1852.
- Blawat, M.; Gaedke, K.; Huetter, I.; Chen, X.-M.; Turczyk, B.; Inverso, S.; Pruitt, B. W.; and Church, G. M. 2016. Forward error correction for DNA data storage. *Procedia Computer Science*, 80: 1011–1022.
- Bögels, B. W.; Nguyen, B. H.; Ward, D.; Gascoigne, L.; Schrijver, D. P.; Makri Pistikou, A.-M.; Joesaar, A.; Yang, S.; Voets, I. K.; Mulder, W. J.; et al. 2023. DNA storage in thermoresponsive microcapsules for repeated random multiplexed data access. *Nature Nanotechnology*, 1–10.
- Cao, B.; Zhang, X.; Cui, S.; and Zhang, Q. 2022. Adaptive coding for DNA storage with high storage density and low coverage. *NPJ systems biology and applications*, 8(1): 23.
- Cao, B.; Zhao, Y.; Xie, L.; Shao, Q.; Wang, K.; Wang, B.; Zhou, S.; and Zheng, P. 2025. DBSP: An End-to-end Pipeline for DNA Storage Data Reconstruction from DNA Sequencing. *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 1–1.
- Cao, B.; Zheng, Y.; Shao, Q.; Liu, Z.; Xie, L.; Zhao, Y.; Wang, B.; Zhang, Q.; and Wei, X. 2024. Efficient data reconstruction: The bottleneck of large-scale application of DNA storage. *Cell Reports*, 43(4).
- Carmean, D.; Ceze, L.; Seelig, G.; Stewart, K.; Strauss, K.; and Willsey, M. 2019. DNA Data Storage and Hybrid Molecular–Electronic Computing. *Proceedings of the IEEE*, 107(1): 63–72.
- Church, G. M.; Gao, Y.; and Kosuri, S. 2012. Next-generation digital information storage in DNA. *Science*, 337(6102): 1628.
- Erlich, Y.; and Zielinski, D. 2017. DNA Fountain enables a robust and efficient storage architecture. *Science*, 355(6328): 950–953.
- Franzese, G.; Yan, Y.; Serra, G.; D’Onofrio, I.; Appuswamy, R.; and Michiardi, P. 2021. Generative dna: Representation learning for dna-based approximate image storage. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 01–05. IEEE.
- Goldman, N. M.; Bertone, P.; Chen, S.; Dessimoz, C.; Leproust, E. M.; Sipos, B.; and Birney, E. 2013. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435): 77–80.
- Grass, R. N.; Heckel, R.; Puddu, M.; Paunescu, D.; and Stark, W. J. 2015. Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes. *Angewandte Chemie*, 54(8): 2552–2555.
- Guo, A. J.; and Qi, H. 2021. Using Artificial Neural Networks to Model Errors in Biochemical Manipulation of DNA Molecules. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1.
- He, T.; Liu, Y.; Ong, Y.-S.; Wu, X.; and Luo, X. 2024. Polarized message-passing in graph neural networks. *Artificial Intelligence*, 331: 104129.
- He, T.; Ong, Y. S.; and Bai, L. 2021. Learning conjoint attentions for graph neural nets. *Advances in Neural Information Processing Systems*, 34: 2641–2653.
- Leppek, K.; Byeon, G. W.; Kladwang, W.; Wayment-Steele, H. K.; Kerr, C. H.; Xu, A. F.; Kim, D. S.; Topkar, V. V.; Choe, C.; Rothschild, D.; et al. 2022. Combinatorial optimization of mRNA structure, stability, and translation for RNA-based therapeutics. *Nature communications*, 13(1): 1536.
- Li, X.; Cao, B.; Wang, J.; Meng, X.; Wang, S.; Huang, Y.; Petretto, E.; and Song, T. 2025. Predicting mutation-disease associations through protein interactions via deep learning. *IEEE Journal of Biomedical and Health Informatics*.
- Li, X.; Wang, B.; Lv, H.; Yin, Q.; Zhang, Q.; and Wei, X. 2020. Constraining DNA sequences with a triplet-bases unpaired. *IEEE transactions on nanobioscience*, 19(2): 299–307.
- Li, Y.; Du, D. H.; Ou, L.; and Li, B. 2022. HL-DNA: A Hybrid Lossy/Lossless Encoding Scheme to Enhance DNA Storage Density and Robustness for Images. In *2022 IEEE 40th International Conference on Computer Design (ICCD)*, 434–442. IEEE.
- Limbachiya, D.; Gupta, M. K.; and Aggarwal, V. 2022. 10 Years of Natural Data Storage. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 8(4): 263–275.

- Liu, Z.; Cao, B.; Shao, Q.; Zheng, Y.; Wang, B.; Zhou, S.; and Zheng, P. 2025. Family of Mutually Uncorrelated Codes for DNA Storage Address Design. *IEEE Transactions on NanoBioscience*.
- Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halver-son, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.
- Lochel, H. F.; Welzel, M.; Hattab, G.; Hauschild, A. C.; and Heider, D. 2021. Fractal construction of constrained code words for DNA storage systems. *Nucleic Acids Research*, e30.
- Nguyen, T. T.; Cai, K.; Immink, K. A. S.; and Kiah, H. M. 2021. Capacity-approaching constrained codes with error correction for DNA-based data storage. *IEEE Transactions on Information Theory*, 67(8): 5602–5613.
- Organick, L.; Ang, S. D.; Chen, Y.; Lopez, R.; Yekhanin, S.; Makarychev, K.; Racz, M. Z.; Kamath, G. M.; Gopalan, P.; and Nguyen, B. H. 2018. Random access in large-scale DNA data storage. *Nature Biotechnology*, 36(3): 242–248.
- Ping, Z.; Chen, S.; Zhou, G.; Huang, X.; Zhu, S. J.; Zhang, H.; Lee, H. H.; Lan, Z.; Cui, J.; Chen, T.; et al. 2022. Towards practical and robust DNA-based data archiving using the yin–yang codec system. *Nature Computational Science*, 2(4): 234–242.
- Rasool, A.; Hong, J.; Jiang, Q.; Chen, H.; and Qu, Q. 2023. BO-DNA: Biologically optimized encoding model for a highly-reliable DNA data storage. *Computers in Biology and Medicine*, 165: 107404.
- Ross, M. G.; Russ, C.; Costello, M.; Hollinger, A.; Lennon, N. J.; Hegarty, R.; Nusbaum, C.; and Jaffe, D. B. 2013. Characterizing and measuring bias in sequence data. *Genome biology*, 14: 1–20.
- SantaLucia Jr, J. 1998a. A unified view of polymer, dumb-bell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95(4): 1460–1465.
- SantaLucia Jr, J. 1998b. A unified view of polymer, dumb-bell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95(4): 1460–1465.
- Wang, J.; Mbah, C. F.; Przybilla, T.; Apeleo Zubiri, B.; Spiecker, E.; Engel, M.; and Vogel, N. 2018. Magic number colloidal clusters as minimum free energy structures. *Nature communications*, 9(1): 5259.
- Wang, Y.; Noor-A-Rahim, M.; Gunawan, E.; Guan, Y. L.; and Poh, C. L. 2023. Modelling, characterization of data-dependent and process-dependent errors in DNA data storage. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Wu, J.; Wang, P.; Zheng, Y.; Wang, B.; Zhang, Q.; and Zheng, P. 2025. Stable DNA Storage Encoding Scheme Based on Repeating Substring Tree. *IEEE Transactions on Computational Biology and Bioinformatics*, 1–10.
- Wu, W.; Xiang, L.; Liu, Q.; and Yang, K. 2023. Deep Joint Source-Channel Coding for DNA Image Storage: A Novel Approach with Enhanced Error Resilience and Biological Constraint Optimization. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*.
- Yang, D.; Liu, W.; Deng, X.; Xie, W.; Chen, H.; Zhong, Z.; and Ma, J. 2020. Gc-content dependence of elastic and over-stretching properties of DNA: RNA hybrid duplexes. *Biophysical Journal*, 119(4): 852–861.
- Yazdi, S.; Yuan, Y. B.; Ma, J.; Zhao, H. M.; and Milenkovic, O. 2015. A Rewritable, Random-Access DNA-Based Storage System. *Scientific Reports*, 5: 14138.
- Zhang, X.; Liu, Y.; Wang, B.; Zhou, S.; Shi, P.; Cao, B.; Zheng, Y.; Zhang, Q.; and Kirilov Kasabov, N. 2023. Biomolecule-Driven Two-Factor Authentication Strategy for Access Control of Molecular Devices. *ACS nano*, 17(18): 18178–18189.
- Zheng, Y.; Cao, B.; Zhang, X.; Cui, S.; Wang, B.; and Zhang, Q. 2024. DNA-QLC: an efficient and reliable image encoding scheme for DNA storage. *BMC genomics*, 25(1): 266.
- Zhou, H.; He, T.; Ong, Y.-S.; Cong, G.; and Chen, Q. 2024. Differentiable clustering for graph attention. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 3751–3764.

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.

For each question (that applies), replace the “Type your response here” text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims  
are included} {(yes/partial/no)}  
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims  
are included} {(yes/partial/no)}  
yes
```

Please make sure to:

- Replace **ONLY** the “Type your response here” text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this `.tex` file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference’s website to see if you will be asked to provide this checklist with your paper or separately.

1. General Paper Structure

- 1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) **yes**
- 1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) **yes**
- 1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) **yes**

2. Theoretical Contributions

- 2.1. Does this paper make theoretical contributions? (yes/no) **yes**

If yes, please address the following points:

- 2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) **yes**
- 2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) **yes**

- 2.4. Proofs of all novel claims are included (yes/partial/no) **yes**
- 2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) **yes**
- 2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) **yes**
- 2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) **NA**
- 2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) **NA**

3. Dataset Usage

- 3.1. Does this paper rely on one or more datasets? (yes/no) **yes**

If yes, please address the following points:

- 3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) **yes**
- 3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) **yes**
- 3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) **yes**
- 3.5. All datasets drawn from the existing literature (potentially including authors’ own previously published work) are accompanied by appropriate citations (yes/no/NA) **yes**
- 3.6. All datasets drawn from the existing literature (potentially including authors’ own previously published work) are publicly available (yes/partial/no/NA) **yes**
- 3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying (yes/partial/no/NA) **NA**

4. Computational Experiments

- 4.1. Does this paper include computational experiments? (yes/no) **yes**

If yes, please address the following points:

- 4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) **yes**
- 4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) **no**

- 4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) **no**
- 4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) **yes**
- 4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) **yes**
- 4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) **NA**
- 4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) **yes**
- 4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) **yes**
- 4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) **yes**
- 4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) **yes**
- 4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) **yes**
- 4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) **yes**

More details on DNA data storage

The International Data Corporation (IDC) predicts that the capacity of the global data circle will increase to 175 zettabytes by 2025. Current storage systems are faced with high cost and huge energy consumption. In contrast, DNA is a highly parallel, low-cost storage medium with great storage potential. Different from traditional storing medium that are replaced every few years, DNA is very stable in decades or centuries, and can be easily replicated and stored based on methods of molecular biology.

There are five steps in DNA data storage, including encoding information into DNA codewords (encoding), synthesizing DNA from the sequence (writing), storage, DNA sequencing (reading), and decoding. For encoding, fountain, rotation, and Huffman code are commonly used methods. Methods that are based on modern AI techniques have also been proposed for encoding purposes. Popular techniques for DNA synthesis can then be used to synthesize the sequences storing real-world data in DNA. After that, DNA molecules of real-world data can be stored in test tubes or in the form of dry powder for a very long period. To read the data from DNA, prevalent sequencing methods, e.g., Illumina sequencing and nanopore sequencing can be used. After processing the sequenced DNA with clustering and assembly techniques, the original data can finally be recovered by decoding, which is completed inversely by the previously used encoding method.

Currently, the main bottlenecks existing in DNA storage are cost and read/write latency. DNA storage is costly due to DNA synthesis and sequencing. Issues of read/write latency in DNA storage are mainly due to codec and corresponding biotechnologies. It is seen that efficient and robust codec algorithms can reduce not only the cost by improving the encoding rate but also the read-write latency by reducing errors in DNA storage. In this paper, we propose a novel learning-based model that can significantly reduce the read and write delay from the codec stage. Besides, the proposed model provides an economical solution to DNA storage by reducing the error rate and improving the encoding rate.

More details on the baselines

Based on the coding method, the baselines can generally divided into two categories, i.e., coding theory based on learning based approaches. Methods based on coding theory have the advantage of predictable results and complete proof of theory.

- Church (Church, Gao, and Kosuri 2012): This method proposes encoding one binary bit per base (A or C for 0, G or T for 1) to encode bitstreams directly into DNA sequences.
- Goldman (Goldman et al. 2013): This approach uses the Huffman trinomial tree to analyze binary files to be transcoded based on the frequency of occurrence of individual bytes. The binary sequences (0/1) are converted to the corresponding ternary sequences (0/1/2), which are subsequently mapped to the corresponding DNA sequences according to the ternary mapping

model.

- Grass (Grass et al. 2015): This approach combines the Galois field with the DNA codon wheel style base mapping rules to propose a coding algorithm that avoids the length of a single base being greater than three.
- Blawat (Blawat et al. 2016): This method uses the byte as the basic unit of base conversion and maps eight bits of information into five nucleotides. The first six bits are fixed conversion portions mapped to nucleotides A, C, G, and T. The last two bits are optional conversion portions. This design limits the maximum length of the homopolymer to three.
- DNA Fountain (Erlich and Zielinski 2017): This approach preprocesses binary information into a series of non-overlapping fragments, randomly selects a variable number of sequence fragments for heterogeneous operation based on Luby Transform, and appends a fixed-length seed to form a droplet.
- Yin-Yang (Ping et al. 2022) : This method provides a dynamic combinatorial coding scheme that combines two independent coding rules (called "yin" and "yang") into a single binary sequence, thus compressing two bits into a single nucleotide.
- HL-DNA (Li et al. 2022): This approach proposes a hybrid lossy and lossless image storage scheme implemented by quaternary mapping. HL-DNA uses about 300 nt as the length of DNA strands and four extra nucleotides, and it is a hybrid lossy/lossless encoding scheme.

Recently, attention has been paid to learning-based DNA storage. However, due to the information loss inherent in neural network learning, current learning-based DNA storage schemes are lossy and can only be used for storing multimedia data such as images.

- DJSCC (Wu et al. 2023): This approaches uses a convolutional neural network for the DNA encoding and decoding process and reconstructs the loss function by optimizing for GC content and homopolymer constraints. DJSCC used input size (32, 32, 3), $R(nt/pixel)$ is 1/8, 1) Encoder: $CV(32, 1, 3)$, BN , $ReLU$; $CV(64, 32, 3)$, BN , $ReLU$; $CV(63, 128, 3)$; $FC(32)$.2 Decoder: $FC(1152)$, $DC(128, 64)$, BN , $ReLU$; $DC(64, 32)$, BN , $ReLU$, $DC(32, 1)$, Sigmoid.
- DNA-QLC (Zheng et al. 2024): This approach uses the quantized ResNet VAE (QRes-VAE) model and LC for image compression. It differs from VAE-QC in that it also designed an error correction module to improve the storage system's robustness. DNA-QLC used the QRes-VAE, and based on ResNet VAE, all parameters are consistent with the original text.

As for the settings of RSRL, the model hyperparameters

were fixed with latent dimension 4, hidden dimension 4, and 4 attention heads. Training employed Adam optimization ($lr = 0.009$, $weight_decay = 5e-4$). The experimental environment for this work is RTX3090, 256G RAM, i7 9700k.

More analysis and discussions on the performance of DNA storage

Here, we further analyze the results in Table 1. Because higher biological stability of DNA sequences can reduce the probability of errors occurring during the storage process, this section mainly analyzes the GC content, homopolymers, and hairpin structure of the encoding results.

GC content and homopolymers Most storage schemes can meet the GC content constraint. However, the fluctuations (large deviations) of the GC content obtained by Grass and Blawat are evident. Previous studies have shown that deviations of the GC content have a significant effect on the melting temperature (Ross et al. 2013). Thus, the PCR yield and sequencing accuracy of these two approaches are reduced. In contrast, RSRL, DNA-QLC, and HL-DNA perform robustly when evaluated by GC content (50%), demonstrating that these approaches may cause fewer errors during DNA sequencing. Homopolymers during synthesis can cause difficulties, while during sequencing, they may cause gaps such as *AAAA* to be misread as *AAA*, affecting data consistency. However, overly strict homopolymers could affect base utilization since the total number of base combinations is fixed. RSRL limited the number of homopolymers to around three, achieving a good balance between base utilization and data consistency in reading and writing.

Hairpin structure A hairpin structure is a distinctive secondary structure in DNA, where two base pairs are held together by hydrogen bonds (Li et al. 2020). The hairpin structure can increase the error rate when reading and replicating in DNA storage, thus influencing the performance. It can be seen from Table 1 that RSRL is one of the few approaches to DNA storage, considering the impact of hairpin structures. Although DJSCC can satisfy GC content and homopolymer constraints through learning, its deviations of GC content and homopolymer limitations are higher than those of RSRL. DNA-QLC attempts to force the learned representations to satisfy the biological constraints by post-processing. Thus, it is not an end-to-end learning approach. Moreover, existing learning approaches like DJSCC and DNA-QLC, can only store multimedia data, e.g., image data, due to the information loss during model training. Compared with existing learning-based approaches, the proposed RSRL is the first end-to-end model that considers stable properties of biological structures. Thus, RSRL performs much better than existing learning-based methods regarding diverse metrics of DNA storage.

Additional results of data encoding

Although any data type is binary at input to the model, RSRL has no bias to the data type, that is, the data type has minimal influence on the encoding result. However, in order to further

illustrate the applicability of RSRL to multi-modal data, we also verified this point through experiments, as shown in Tables 4-7. The evaluation metrics are the same as the main text, and the results show that RSRL is unbiased in terms of data types. In particular, the unit of encoding and decoding time is bits/s.

Details on the metrics of DNA thermodynamics

Minimum free energy

The Gibbs Free Energy contains two important thermodynamic parameters, entropy change and enthalpy change: $\Delta G = \Delta H - T\Delta S$, where T is the temperature. The minimum Gibbs free energy is the minimum value of the standard free energy of all possible secondary structures in the DNA sequence. Secondary structures with lower Gibbs free energy are more stable. Therefore, the minimum free energy (MFE) could be used to assess the quality of DNA sequences. Let $\Delta G(s, s')$ denote the Gibbs free energy value of DNA sequence s , where s' is its complementary strand, which could be calculated using the PairFold method (SantaLucia Jr 1998a):

$$MFE = \min\{\Delta G(u, v), \Delta G(u, v'), \Delta G(u', v')\}, \quad (15)$$

where $\Delta G(u, v)$, $\Delta G(u, v')$, and $\Delta G(u', v')$ are the Gibbs free energy between u and v , and that between their complements, respectively. Given the the number of sequences and their lengths, the average MFE (MFEave) can be computed as the following:

$$MFE_{ave} = \frac{1}{n} \sum_{i=1}^n \frac{MFE_i}{L_i}, \quad (16)$$

where n is the number of DNA sequences, and L_i is the length of DNA sequence i . Accordingly, we are able to obtain the standard deviation of MFE of each approach (MFEstd).

Melting temperature

In this paper, the melting temperature (T_m) is calculated as the following:

$$T_m = \Delta H^\circ / (\Delta S^\circ + R \ln C_T) - 273.15, \quad (17)$$

where R is the gas constant (1.987 cal/K·mol), C_T is the total oligonucleotide strand concentration, ΔH° and ΔS° can be obtained by search the Unified oligonucleotide table (SantaLucia Jr 1998b). The T_m represented the temperature required for a DNA sequence to transition from a double-stranded structure to a single-stranded structure. When T_m falls within the range of (85-95), we should pay more attention to the standard deviation of T_m (T_mstd), as stable T_m values indicate orderly progression of assembly and PCR processes. Then, a smoother data reading and writing process in DNA storage can be achieved.

Method	NID	MFEave	MFEstd	Tmave	Tmstd	GCave	GCstd	Encoding speed	Decoding time
Grass	1.56	-31.25	6.37	91.55	2.78	49.65	5.87	1.32E6	2.00E6
Blawat	1.40	-23.39	6.58	90.65	1.61	46.47	5.42	6.73E7	4.00E7
DNA Fountain	1.23	-23.31	5.32	90.55	2.77	49.82	5.83	1.56E5	1.83E5
DNA-QLC	-	-	-	-	-	-	-	-	-
RSRL	1.75	-34.49	6.42	94.49	1.61	50.42	4.03	6.73E7	4.00E7

Table 4: Results of PDF data encoding.

Method	NID	MFEave	MFEstd	Tmave	Tmstd	GCave	GCstd	Encoding speed	Decoding time
Grass	1.56	-25.70	5.98	91.43	2.60	49.79	5.25	1.48E6	1.49E6
Blawat	1.40	-38.77	7.4	96.57	1.74	58.15	3.59	5.86E5	6.0E5
DNA Fountain	1.23	-26.64	5.98	92.19	2.54	49.57	5.44	8.49E3	3.45E4
DNA-QLC	-	-	-	-	-	-	-	-	-
RSRL	1.75	-32.07	5.67	94.54	2.33	51.28	6.47	6.45E7	4.11E7

Table 5: Results of text data encoding.

Network structure of RSRL and other learning-based approaches

In this section, we compare the network structure of the proposed RSRL with other learning-based approaches, including DJSCC (Wu et al. 2023) and DNA-QLC (Zheng et al. 2024). Regarding computational complexity, we mainly consider Total Madd and Total Flops, which represent the number of multiply-accumulate and floating-point operations. Regarding space complexity, we mainly consider the total memory used by each model (Total Memory) and the total number of parameters of each model (Total Params). The comparative results have been listed in Table 8. As the table shows, the proposed RSRL achieves much lower than any other two approaches, demonstrating it is very efficient in learning representations for DNA storage. Together with the performance results of DNA storage, we conclude that a lightweight learning model with proper biological inspirations (e.g., the biologically stabilized loss proposed in this paper) can still perform robustly in diverse DNA storage tasks.

End-to-end training

Data Input and Graph Construction

For any file of size W (in bits), it is first converted to a binary matrix $\mathbf{X} \in \mathbb{R}^{M \times 48}$, where $M = \lceil W/48 \rceil$. This matrix is then row-wise encoded using a Reed-Solomon code RS(64, 48) in $\text{GF}(2^8)$ to generate an error-corrected binary stream $\mathbf{C} \in \mathbb{R}^{M \times 64}$.

To construct graph-structured data:

- 0.1 **K-mer Segmentation:** Divide \mathbf{C} into 4-bit chunks (each representing a nucleotide $\in \{A, C, G, T\}$)
- 0.2 **Node Definition:** Generate 4-mers ($k_{\text{mer}} = 4$) by sliding a window of size 4 with step 1. Each 4-mer (16-bit vector) becomes a node $v_i \in V$ with feature $\mathbf{x}_i \in \mathbb{R}^{16}$.
- 0.3 **Edge Construction:**

- Connect consecutive 4-mers with edges (implicit overlap k_{lap})
- Add self-loops to all nodes

The resulting graph $G = (V, E, \mathbf{X})$ has node features $\mathbf{X} \in \mathbb{R}^{N \times 16}$ ($N = \text{number of 4-mers}$) and edge features $\mathbf{e}_{ij} \in \mathbb{R}^2$.

This graph is processed by the FKGAT network to extract low-dimensional representations $\mathbf{Z} \in \mathbb{R}^{N \times 4}$, which are optimized via a hybrid loss function $\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{RS}}$ and encoded to DNA sequences via binary-to-base mapping (Table 10).

Codec

The proposed RSRL framework implements an end-to-end training pipeline that seamlessly integrates error correction coding with deep representation learning. The RSRL framework implements error correction through these stages:

Encoder During encoding, input files undergo a multi-stage transformation: first converted to binary streams, then partitioned into 6-byte chunks ($M \times 48$ bits) for RS(64,48) encoding in $\text{GF}(2^8)$, resulting in $M \times 64$ -bit codewords. These codewords are restructured into $N \times 32 \times 64$ tensors and processed through a graph construction module that represents 4-bit k-mers ($k = 4$) as nodes in a graph structure. The core learning component features a hybrid GAT-FKAN encoder with 2 graph attention layers ($\text{hidden_dim} = 16$), learning compact 4-dimensional latent representations. A strategic masking mechanism suppresses attention weights for the final 8 bits to localize potential errors. The normalized representations are finally transcoded to DNA sequences using our binary-to-base mapping scheme by Table 10.

Decoder For the decoder, the input is a DNA sequence, inverted to a binary matrix by Table 10. The structure of the decoder is similar to that of the encoder, except that a masking mechanism and encoder-decoder attention are used to make the decoder pay more attention to the different parts of the input sequence. The raw tensor vector output from

Method	NID	MFEave	MFEstd	Tmave	Tmstd	GCave	GCstd	Encoding speed	Decoding time
Grass	1.56	-29.09	5.42	92.86	1.93	51.37	3.73	1.43E6	1.42E6
Blawat	1.40	-24.25	6.85	90.43	2.45	45.35	5.77	1.22E6	1.69E6
DNA Fountain	1.23	-34.49	6.42	94.49	1.61	46.42	4.29	6.10E5	4.03E7
DNA-QLC	2.90	-69.19	7.09	96.47	0.24	50	0	2.58E4	1.43E4
RSRL	1.75	-33.92	4.06	94.97	2.20	51.57	4.89	6.30E7	3.93E7

Table 6: Results on image data encoding.

		Errors bits=1	Errors bits=2	Errors bits=3	Errors bits=4	Coding rate
RS codes	-	6.32E-4, 1	-	-	-	0.875
	d_v=3	1.9E-3, 5E3	-	-	-	0.9133
LDPC-d.c_30	d_v=4	1.9E-3, 5E3	2E-3, 5E4	-	-	0.8866
	d_v=5	3.9E-3, 5E2	2.9E-3, 5E2	1.38E2, 5E5	-	0.86
LDPC-d.c_12	d_v=6	6.0E-1, 5E2	6.0E-1, 5E2	6.0E1, 5E5	6.0E1, 5E5	0.5416
RSRL	-	2.69E-5, 1	2.69E-5, 1	2.69E-5, 1	2.69E-5, 1	0.875

Table 7: Comparisons of decoding time of different error bits and coding rates obtained by LDPC, RS, and RSRL

Model	Input size	Total params	Total memory (MB)	Total Madd(M)	Total Flops(M)
DJSCC	(32,32,3)	54921	0.24	7.47M	1.76M
DNA-QLC	(64,64,1)	3538944	13.5	906M	1812M
RSRL	(32,32,64)	49152	0.1875	0.049M	0.098M

Table 8: Comparison of the number and complexity of model parameters.

Method	Reconstruction rate	Failure rate	MFEave	MFEstd	Tmave	Tmstd	GCave	GCstd
No GC	100%	0	-28.8	7.65	91.37	3.35	48.66	6.91
No pair	100%	0	-26.7	6.24	92.22	2.97	50.77	6.15
No GC&Pair	100%	0	-24.7	4.24	91.03	2.58	48.33	5.76
No MASK	93.42%	9.6%	-18.72	4.32	85.86	2.20	38.22	3.92
No GC&P&MASK	98.76%	2.4%	-20.66	3.66	88.90	1.74	44	3.84
RSRL	100%	0	-34.49	6.42	94.49	1.61	50.42	4.03

Table 9: Ablation study of RSRL

	00	01	10	11
00	AT	AG	AC	AA
01	TA	TC	TG	TT
10	GG	GA	GT	GC
11	CC	CT	CA	CG

Table 10: Binary streams coded as base sequences

the linear decoder is converted into hexadecimal characters and fed into the RS decoder for decoding. Due to the Mask operation at the time of encoding, the errors are concentrated in the last 8 bits of the tensor so that RS codes can correct all errors in the last 8 bits. Finally, the RS decoded hexadecimal characters are converted to a binary stream to complete the reconstruction of the original file.