scientific reports

Check for updates

OPEN

Reliable image reconstruction from DNA storage through marker-guided decoding and candidate filtering

Monika Kurgonaitė[1,2]✉, Agnė Paulauskaitė[1,2], Birutė Frercks[3] & Andrius Kriščiūnas[1]

DNA has emerged as a promising medium for long-term digital data storage because of its exceptional information density and durability. However, reliable data recovery remains challenging for Oxford Nanopore sequencing, where insertion, deletion, and substitution errors can disrupt sequence synchronization and increase ambiguity during decoding. In this study, we present a proof-of-concept framework for DNA-based RGB image storage and reconstruction that integrates marker-aware codebook construction, consensus-based marker-guided decoding from a single sequence, and AI-assisted image restoration. The framework systematically evaluates how different codebook construction strategies influence downstream decoding, candidate ambiguity, and final image reconstruction. Under simulated nanopore sequencing conditions, all constrained codebooks achieved high payload recovery (> 91%), but substantial differences were observed in their ability to preserve the correct payload among decoded candidate sequences. The codebook combining sequence-space discrimination, biochemical quality control, and structural compatibility produced the highest candidate-set quality and, when coupled with Random Forest-based reliability filtering, achieved the best image reconstruction performance, with a peak signal-to-noise ratio of 29.82 dB, a structural similarity index of 0.9309, and 80.19% exact pixel recovery. These findings demonstrate that payload recovery alone is insufficient to evaluate DNA storage performance, as downstream candidate ambiguity directly impacts the quality of reconstructed data. The proposed framework provides an end-to-end approach to investigating the relationship among codebook design, decoding robustness, and image reconstruction, thereby supporting the development of more reliable nanopore-based DNA data storage systems.

Keywords DNA data storage, Marker-guided decoding, Codebook design, Image reconstruction, Nanopore sequencing

The exponential growth of digital data has intensified the search for alternative storage technologies that surpass the physical density, durability, and energy efficiency of conventional mainstream data storage mediums. Deoxyribonucleic acid (DNA), the primary information-bearing molecule in biology, has emerged as a compelling candidate for long-term archival storage due to its exceptionally high storage density per unit volume, longevity, and technological stability. In theory, DNA can store up to 2 bits per nucleotide, corresponding to approximately 455 exabytes per gram of single-stranded DNA[1]. Moreover, DNA can remain readable for centuries to millennia under appropriate storage conditions, far exceeding the lifespan of magnetic, optical, or solid-state media[2]. Unlike traditional digital storage systems, which depend on rapidly evolving hardware standards, DNA benefits from its central role in biology, ensuring that sequencing technologies will remain available and actively developed. These properties make DNA particularly attractive for cold data and deep archival applications, where write-once, read-rarely paradigms dominate[3–5].

A typical DNA data storage system consists of five principal stages: digital encoding, DNA synthesis, physical storage, DNA sequencing, and computational data reconstruction[3–6]. During the encoding stage, binary information is transcoded into nucleotide sequences using the four-letter DNA alphabet (A, C, G, and T), while

[1]Department of Applied Informatics, Faculty of Informatics, Kaunas University of Technology, Kaunas 51368, Lithuania. [2]Artificial Intelligence Excellence Centre, Kaunas University of Technology, Kaunas 51423, Lithuania. [3]Institute of Biological Systems and Genetic Research, Lithuanian University of Health Science, Kaunas 44307, Lithuania. ✉email: monika.kurgonaite@ktu.lt

simultaneously satisfying biochemical constraints necessary for reliable DNA synthesis and sequencing[4,7–9]. The encoded DNA molecules are subsequently synthesized, stored, and later recovered through high-throughput sequencing technologies. Because DNA synthesis and sequencing introduce substitutions, insertions, deletions, strand loss, and coverage variability, the sequenced reads generally cannot be decoded directly. Instead, sequencing reads are first reconstructed into representative DNA sequences before original information is recovered computationally[3–5]. Consequently, the accuracy of the entire storage pipeline depends not only on sequencing performance but also the interaction between sequence design, synchronization strategy, consensus reconstruction, and downstream decoding[3,4,8–11].

Most experimental DNA data storage systems have been developed and evaluated using Illumina sequencing, where relatively low error rates and a predominantly substitution-based error profile enable reliable read clustering, consensus reconstruction, and subsequent decoding[3–5,7,8]. Consequently, many widely adopted encoding strategies, including DNA Fountain and the Yin-Yang codec, were developed under assumptions that are well suited for short-read sequencing technologies[7,8]. In contrast, Oxford Nanopore sequencing offers several practical advantages for DNA data storage, including long-read capability, portable instrumentation, real-time sequencing, and continuously improving throughput[12–15]. These characteristics make nanopore sequencing an attractive platform for future decentralized and large-scale DNA storage applications. However, compared with Illumina sequencing, nanopore data typically contain substantially higher insertion and deletion rates together with sequence-context-dependent errors, resulting in synchronization loss that complicates downstream sequence reconstruction and decoding[11,16–18]. As a result, encoding and decoding strategies designed for substitution-dominated sequencing technologies cannot be directly transferred to nanopore-based storage systems without accounting for the distinct characteristics of indel-rich sequencing channels[11,18,19].

Recent research has proposed numerous approaches to improve robustness of DNA data storage under challenging sequencing conditions. These include constrained encoding schemes that enforce biochemical sequence properties, error-correcting codes capable of handling insertion and deletion errors, synchronization aware coding strategies, and nanopore-specific sequencing and decoding workflows[8,10,11,19–22]. HEDGES, for example, provides error correction for insertions, deletions, and substitutions while supporting user-defined sequence constraints such as homopolymer and GC-content control[20]. DNA-Aeon similarly combines constraint adherence and error correction through flexible arithmetic coding[23]. More recently, DNA-ELMR introduced an end-to-end Transformer-based approach in which GC-content, homopolymers, hairpin structures, and error-prone motifs are incorporated into a differentiable multi-constraint optimization framework[24]. Its complementary-matrix convolution approach further reduces the computational complexity of hairpin constraint evaluation, enabling biological constraints to be incorporated directly into model optimization[24].

For image data specifically, recent approaches have additionally combined DNA coding with image compression and machine learning. DNA-QLC combines variational-autoencoder-based image compression with Levenshtein coding for sequence error correction, thereby addressing both coding density and image reconstruction[25]. INNSE uses an invertible neural network to obtain lower-dimensional image representation, followed by constrained codebook generation and sequence-level self-correction[26]. These approaches demonstrate that image-specific representations can improve storage efficiency and reconstruction robustness. However, neural-network-based image representations can also introduce information loss, as acknowledged for INNSE, making reconstruction quality an important application-level measure in addition to sequence-level recovery[26].

While recent approaches have increasingly focused on error correction, image representation, or the joint optimization of biological constraints during encoding, the effect of codebook design itself on ambiguity during downstream decoding and final image reconstruction remains less explored. In practical DNA data storage workflows, sequencing reads are typically combined into a consensus sequence before downstream decoding is performed. Although consensus reconstruction substantially reduces random sequencing errors, residual insertions, deletions, and substitutions may remain, particularly in nanopore sequencing data[11,19]. These residual errors can disrupt synchronization between encoded information blocks, increase ambiguity during payload assignment and ultimately reduce decoding accuracy[10,25–28]. Importantly, successful sequence recovery does not necessarily guarantee that the correct encoded value remains uniquely identifiable when residual errors produce multiple plausible codeword candidates. For image data, such ambiguity can propagate into incorrect pixel assignments and therefore affect reconstruction quality. Consequently, robust recovery depends not only on consensus accuracy but also on sequence organization, codebook properties, synchronization, and the ability of the decoding algorithm to correctly distinguish among plausible encoded values.

In this work, we developed an end-to-end DNA data storage framework for RGB image encoding and reconstruction designed for nanopore sequencing. The proposed framework integrates marker-aware DNA codebook construction, consensus-based marker-guided decoding with resynchronization, candidate-assisted RGB assignment, and machine-learning-based reliability filtering. Rather than introducing synchronization markers or constrained coding as standalone concepts, the framework investigates how codebook construction influences downstream decoding and application-level reconstruction. Codebooks constructed using different combinations of sequence-space discrimination, biochemical quality control, and structural compatibility are systematically compared under identical simulated nanopore sequencing conditions. Their effects are evaluated at multiple levels, including sequencing and consensus accuracy, payload recovery, candidate ambiguity, and final image reconstruction quality. This enables the relationship between DNA sequence design and application-level recovery to be examined directly and provides a framework for evaluating DNA storage systems beyond sequence recovery alone.

Methodology

Framework overview

The proposed framework for DNA-based RGB image storage and reconstruction is illustrated in Fig. 1. First, a constrained DNA codebook and a corresponding set of synchronization markers are generated. An RGB image is then encoded as an ordered sequence of alternating marker and payload segments. The encoded sequence is then subjected to nanopore-like read simulation, after which the simulated reads are combined into a consensus sequence. The consensus sequence is decoded using a marker-guided algorithm that identifies markers, extracts payloads, and restores synchronization when disrupted by sequencing errors. Finally, the recovered payloads are mapped back to RGB values to reconstruct the image, with unresolved pixels handled by image restoration procedures.

Constrained DNA codebook generation

Overall construction workflow

A marker-aware DNA codebook was constructed to support robust encoding and decoding under insertion, deletion, and substitution errors. The construction process consisted of two stages. First, a reserved pool of marker sequences was generated for synchronization, indexing, and sequence parsing. Marker candidates were accepted only if they satisfied marker-specific biochemical constraints and minimum pairwise Hamming-distance requirements.

After the marker pool was established, payload codewords were generated while preserving compatibility with the reserved markers. Candidate payloads were evaluated using three complementarity constraint groups: sequence-space discrimination to reduce local codeword similarity, biochemical quality control to ensure suitable sequence composition, and structural compatibility to prevent marker interference and undesirable terminal interactions. Together, these constraints were designed to improve decoding robustness while maintaining a sufficiently large payload codebook. This procedure is summarized schematically in Fig. 2 and detailed below.

Marker pool construction

Marker sequences were generated before payload codeword construction and reserved for downstream synchronization, indexing, or parsing. Each marker candidate was evaluated using marker-specific biochemical constraints and strict pairwise Hamming-distance separation from previously accepted markers:

$$d_H(m_i, m_j) \geq d_{marker}$$

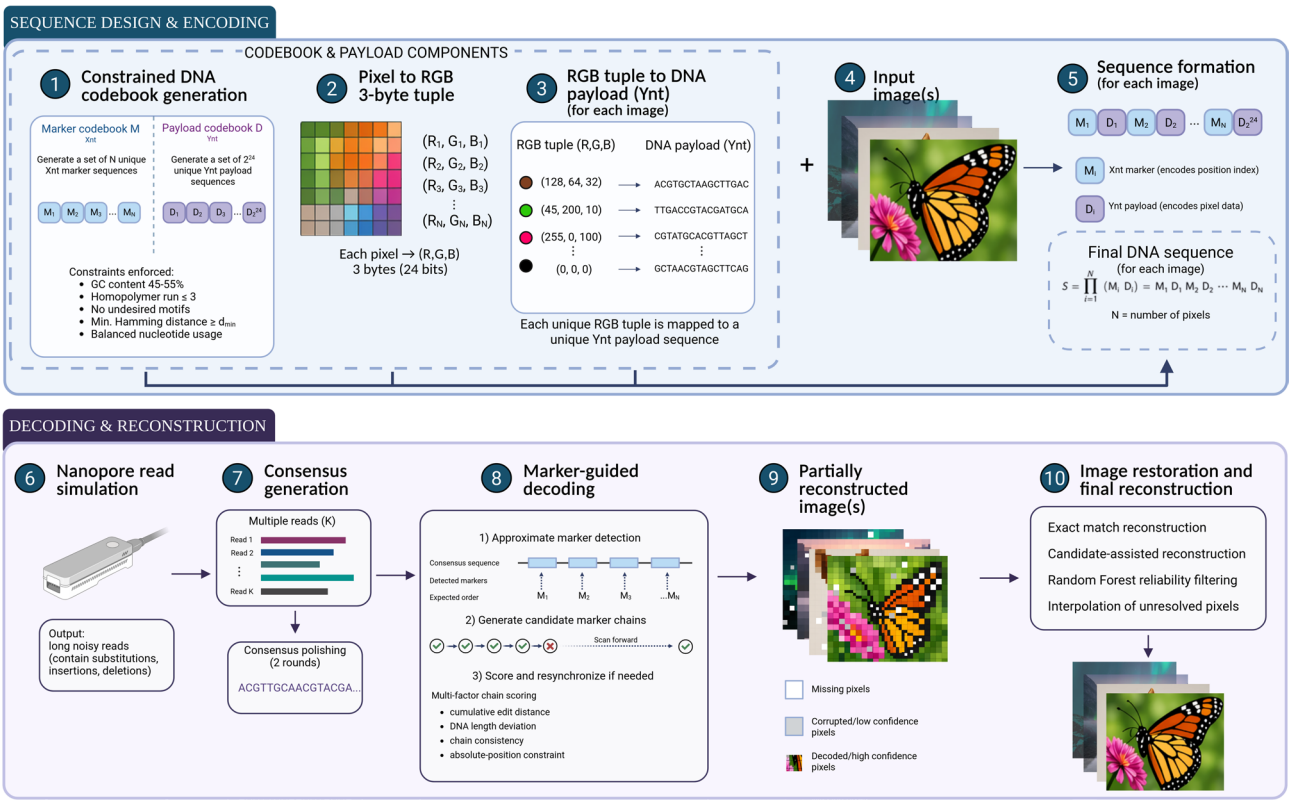


Fig. 1. Overview of the proposed DNA image storage and reconstruction framework. The framework comprises sequence design and encoding (steps 1–5) followed by decoding and image reconstruction (steps 6–10).

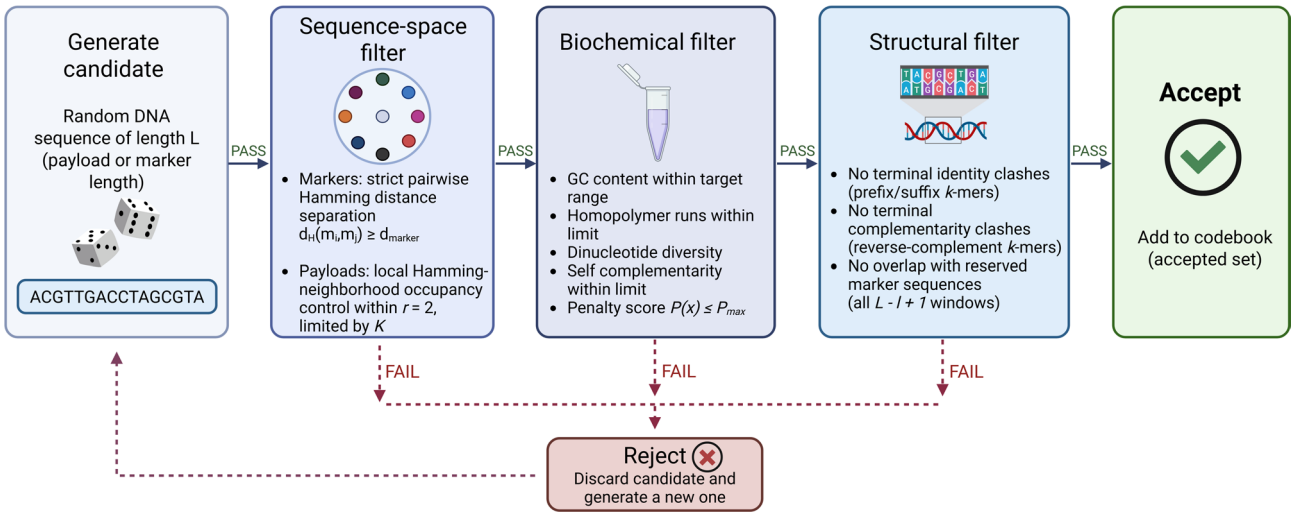


Fig. 2. Schematic overview of marker and payload codebook construction. Candidate marker and payload sequences are generated randomly and evaluated sequentially using sequence-space, biochemical, and structural compatibility constraints.

where m_i and m_j denote marker sequences and d_{marker} is the minimum allowed Hamming distance between any pair of markers.

The biochemical quality of each candidate was assessed using the hard-filter and penalty-based scoring procedure described in Sect. 2.2.4, with thresholds adapted to the shorter marker length. For each marker position, multiple candidate sequences were generated and evaluated, and the lowest-scoring valid candidate was retained. The best-of selection strategy improved the overall quality of the marker set while preserving the required pairwise sequence separation. Once the required number of markers had been generated, the marker pool was fixed and used during payload codeword construction.

Payload local density control

Each candidate payload codeword was evaluated using a distance-aware acceptance criterion designed to limit local clustering within the Hamming space. Let C denote the set of previously accepted payload codewords. For a newly generated candidate x, the number of accepted codewords within a predefined Hamming radius was calculated as:

$$N_r(x) = \{ y \in C : d_H(x, y) \leq r,$$

where d_H denotes the Hamming distance and r is the monitored neighborhood radius. A candidate was rejected when the local neighborhood occupancy exceeded the permitted limit K, $N_r(x) > K$. In this study, the monitored Hamming radius was set to $r=2$. This criterion does not impose a strict minimum pairwise Hamming distance between payload codewords; instead, it limits the number of previously accepted codewords within this local neighborhood to K.

This rule discourages the formation of densely populated regions in the sequence space while avoiding computationally expensive exhaustive pairwise distance calculations. Reducing the number of closely related payload codewords decreases sequence ambiguity following insertion, deletion, and substitution errors, thereby improving subsequent decoding.

The occupancy limit K was adjusted dynamically during codebook construction rather than being fixed a priori. Construction began with an initial value of K, and candidate acceptance statistics were monitored in predefined blocks. After each block, the ratio of neighborhood-based rejections to accepted codewords was evaluated. If this ratio exceeded the predefined rejection threshold, the occupancy limit was increased incrementally:

$$K \leftarrow K + 1 \; if \; R_{neighbor} > \alpha A,$$

where R_{neighbor} is the number of candidates rejected because of excessive local neighborhood occupancy, A is the number of accepted codewords within the same block, and α is the rejection-to-acceptance threshold. This adaptive strategy prevents codebook generation from stalling as the available sequence space becomes progressively saturated while continuing to discourage excessive local clustering.

Biochemical quality control

Biochemical quality control was applied to both marker and payload candidates using the same two-stage evaluation procedure, with thresholds and penalty weights adjusted according to sequence length. The procedure consisted of an initial hard-filtering step followed by penalty-based scoring.

For each candidate sequence x, sequence-level characteristics were evaluated, including GC content, maximum homopolymer length, dinucleotide diversity, maximum contiguous self-complementarity match length, and similarity to the reverse complement. Candidates exceeding predefined thresholds for homopolymer length, GC content, or contiguous self-complementarity were rejected immediately.

Candidates passing the hard-filtering stage were assigned a biochemical penalty score calculated as:

$$P(x) = P_{GC} + P_{run} + P_{dinuc} + P_{selfcomp} + P_{RC},$$

where P_{GC} penalizes deviations from the target GC content, P_{run} penalizes homopolymer lengths approaching the rejection threshold, P_{dinuc} penalizes reduced dinucleotide diversity, $P_{selfcomp}$ penalizes contiguous self-complementarity regions, and P_{RC} penalizes excessive similarity to the reverse complement. A candidate was accepted only if $P(x) \leq P_{max}$.

This two-stage evaluation combines strict exclusion of biologically unsuitable sequences with a weighted assessment of sequence quality, enabling preference for candidates that satisfy multiple biochemical criteria simultaneously while avoiding computationally intensive thermodynamic modelling or secondary-structure prediction.

Structural compatibility constraints

Structural compatibility constraints were applied to maintain separation between the reserved marker pool and the payload codeword space while reducing undesirable interactions between payload termini. Because the marker pool was generated before payload codeword construction, each payload candidate could be evaluated against a fixed set or reserved marker sequences.

Each payload candidate was first screened for exact marker occurrences. For a payload sequence of length L and marker length l, the sequence contains $L - l + 1$ overlapping marker-length windows. Each window is compared with a reserved marker pool, and the candidate was rejected if any window exactly matched a marker sequence. This constraint prevents reserved marker sequences from occurring within payload codewords while avoiding unnecessary exclusion of near-marker variants.

To reduce unintended interactions at codeword junctions, terminal compatibility constraints were additionally applied. As an additional junction-specific heuristic, candidates with homopolymeric two-base prefixes and suffixes were rejected, and short terminal k-mers from accepted payload codewords were stored for subsequent comparisons. A candidate was rejected if its terminal k-mers produced an identity clash with those of previously accepted codewords, for example, when its prefix matched a stored suffix or its suffix matched a stored prefix:

$$prefix_k(x) \neq suffix_k(y),\ suffix_k(x) \neq prefix_k(y),$$

where $y \in C$ denotes the set of previously accepted payload codewords.

Reverse-complement terminal interactions were evaluated using a separate k-mer length. Reverse-complement representations of accepted prefixes and suffixes were stored, and candidates were rejected when their terminal k-mers matched these reverse-complement patterns. Together, these constraints reduce the likelihood of unintended sequence interactions at payload junctions while preserving compatibility with the reserved marker set.

Image encoding and nanopore read simulation

Following codebook construction, digital RGB images were encoded into DNA sequences using the generated marker-aware codebook. Each pixel was represented by its corresponding payload codeword, while marker sequences were interleaved between consecutive payloads to preserve positional information and enable synchronization during decoding.

The encoded DNA sequences were subjected to nanopore read simulation to generate sequencing reads containing insertion, deletion, and substitution errors characteristic of Oxford Nanopore sequencing. Read simulation was performed using DeepSimulator (v1.5), which reproduces the error profile of nanopore sequencing and generates realistic noisy reads[29].

Because the proposed decoding framework operates on a single reconstructed sequence rather than individual sequencing reads, the simulated reads were first combined into a consensus sequence using a reference-free consensus workflow. For each encoded image, 10 simulated reads were used for consensus reconstruction. An initial longest draft sequence was selected from the simulated reads, and the remaining reads were iteratively aligned and incorporated into a consensus sequence using Racon (v1.4.3)[30]. The resulting consensus sequence served as the input for the marker-guided decoding algorithm.

Sequence decoding

Image decoding was performed using a marker-guided algorithm designed to recover encoded payloads from consensus sequences containing substitutions, insertions, and deletions and its schematic overview is depicted in Fig. 3. The reconstructed sequence was represented as an ordered series of marker-payload blocks:

$$R \approx M_i P_i M_{i+1} P_{i+1} \ldots M_j P_j,$$

where M_i denotes the marker associated with pixel i and P_i is the corresponding payload codeword. Each block consisted of a marker followed by a payload of predefined nominal length. Rather than assuming fixed block

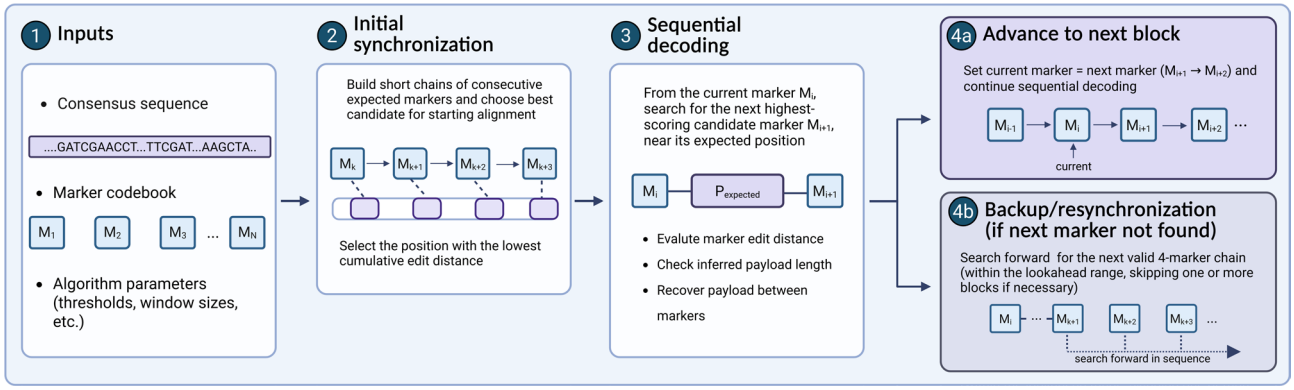


Fig. 3. Workflow of the marker-guided decoding algorithm. Initial synchronization is established by identifying the best-scoring marker chain. Payloads are recovered sequentially between consecutive markers. When synchronization is lost, downstream marker chains are searched to restore synchronization and continue decoding.

boundaries, payloads were identified through approximate marker matching, allowing synchronization to be maintained despite sequencing errors.

Marker detection was performed using bounded Levenshtein edit distance. A marker M_i was accepted at a candidate position p when:

$$d_{edit}\left(M_i R\left[p : p+1\right]\right) \leq \tau\,,$$

where l denoted the length of the compared sequence segment and τ is the permitted edit-distance threshold. Allowing the comparison length to vary around the nominal marker length enabled robust marker detection in the presence of insertion and deletion errors.

Decoding was initiated by identifying an ordered chain of consecutive expected markers near the beginning of the reconstructed sequence. Candidate marker chains were evaluated according to their cumulative edit distance and consistency with the expected marker positions. Chains exhibiting substantial disagreement between the observed marker locations and their expected positions were rejected to avoid incorrect synchronization. The Hamming-distance constraint is applied only during marker-set construction and does not guarantee equivalent separation under Levenshtein distance. Therefore, tolerance to insertion and deletion errors is handled during decoding through ordered marker expectations, positional consistency, and multi-marker chain validation rather than through a strict Levenshtein-distance separation requirement during marker generation.

Following successful synchronization, payloads were recovered sequentially without permitting skipped marker positions. For each marker M_i, the algorithm searched for the immediately following expected marker M_{i+1} near its predicted location:

$$p_{expected} = p\left(M_i\right) + L_{block},$$

where L_{block} denotes the nominal marker-payload block length.

Candidate transitions were evaluated using marker edit distance, inferred payload length, and consistency with a short downstream marker chain. Once the next marker was identified, the intervening sequence was extracted as the decoded payload:

$$\hat{P}_i = R\left[e\left(M_i\right) : s\left(M_{i+1}\right)\right],$$

where $e(M_i)$ and $s(M_{i+1})$ denote the end position of the current marker and the start position of subsequent marker, respectively. Payloads outside the predefined acceptable length range were discarded.

If the expected marker could not be identified, the decoder attempted to restore synchronization by searching downstream for a new chain of consecutive expected markers using predicted sequence coordinates based on the nominal block spacing and a local positional window. Candidate chains were evaluated using the same edit-distance criterion as during initial synchronization together with positional consistency. When a valid chain was identified, decoding resumed from the first recovered marker. Otherwise, synchronization was considered lost and decoding was terminated.

For each recovered payload, an internal confidence score was calculated using marker similarity, payload length agreement, and transitional consistency. The decoder produced one output record for every pixel position, including unresolved positions. Successfully decoded payloads were reported together with their marker identities and sequence coordinates, whereas unresolved positions were explicitly labelled as missing. These records were subsequently used to evaluate decoding performance and support image reconstruction.

Image restoration

Following sequence decoding, recovered payloads were mapped to RGB values using the marker-aware DNA codebook. Because sequencing errors and decoding ambiguity may result in missing, corrupted, or incorrectly assigned payloads, an image restoration step was applied before final reconstruction. Three restoration strategies with increasing complexity were evaluated: exact match reconstruction, candidate-assisted reconstruction, and machine-learning-assisted restoration.

In the exact-match reconstruction strategy, only payloads that exactly matched a codebook entry were assigned RGB values. A decoded payload was considered valid only when its Levenshtein distance to a codeword was zero. Payloads without an exact match were classified as unresolved.

After all exact matches had been assigned to their corresponding pixel positions, unresolved pixels were reconstructed by interpolation from neighboring recovered pixels. This conservative strategy served as the baseline restoration approach because it did not attempt to infer values from ambiguous or partially corrupted payloads.

In the candidate-assisted reconstruction strategy, payloads without an exact codebook match were further evaluated using near-codeword candidates. For each unresolved payload, the codebook was searched for candidate codewords within a predefined Levenshtein-distance threshold, and each candidate was mapped to its corresponding RGB value.

The final assignment was selected using local color consistency. Candidate RGB values were compared with neighboring reconstructed pixels, and the candidate minimizing local color discrepancy was selected. If no suitable candidate was identified or insufficient neighboring information was available, the pixel remained unresolved.

Following candidate selection, any remaining unresolved pixels were reconstructed by interpolation. The strategy extends exact-match reconstruction by exploiting information from partially corrupted payloads while reducing ambiguous assignments through local image context.

The AI-based restoration strategy introduced an additional reliability assessment before the final interpolation step. A Random Forest classifier was used to identify pixel assignments that were likely to be incorrect, including both erroneous exact matches and incorrect candidate-assisted assignments.

The classifier used features describing the assigned RGB value, alternative candidate colors, neighboring pixel values, and local color-difference measures to estimate the reliability of each reconstructed pixel.

Assignments classified as reliable were retained. Assignments classified as unreliable were replaced with a more locally consistent candidate when available or marked as unresolved if no reliable alternative could be identified. Remaining unresolved pixels were subsequently reconstructed by interpolation.

This strategy extends candidate-assisted reconstruction by filtering unreliable pixel assignments before interpolation, allowing both incorrect exact matches and ambiguous candidate selections to be corrected.

Framework evaluation

The proposed framework was evaluated by comparing DNA codebooks generated using different combinations of sequence-space discrimination, biochemical quality control, and structural compatibility constraints. A random codebook served as the baseline for comparison. All codebooks were evaluated using the same image encoding procedure, nanopore read simulation settings, consensus reconstruction workflow, marker-guided decoding algorithm, and candidate-search parameters to ensure that observed differences originated solely from the codebook construction strategy.

For each codebook, RGB images were encoded into marker-payload DNA sequences, simulated as nanopore reads, reconstructed into consensus sequences, and decoded using the proposed marker-guided decoder. Decoder performance was assessed using the sequencing error rate, read identity, proportion of recovered pixels, and the percentage of images with more than 90% recovered pixels. For codebooks employing sequence-space discrimination, the final adaptive occupancy limit (K) was also recorded.

Candidate-set quality was evaluated by determining whether the original payload codeword remained within the candidate set after decoding. For each decoded payload, the original codeword was classified as an exact match (Levenshtein distance 0), retained within the candidate set at Levenshtein 1 or 2, or absent from the candidate set.

Based on this analysis, the marker-aware codebook generated using the complete constraint set was selected for image reconstruction experiments. Three restoration strategies (exact-match reconstruction, candidate-assisted reconstruction, and machine-learning-based reliability filtering) were subsequently compared using mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and exact pixel recovery (Table 1).

As an external image-level benchmark, the proposed framework was additionally evaluated using the 24-image set from Kodak Lossless True Color Image Suite (PCD0992)[31]. Each image was processed using the

Evaluation stage	Reported metrics
Codebook construction	Final adaptive occupancy limit (K) value, where applicable
Sequence decoding	Mean sequencing error rate; mean read identity; recovered pixels, mean ± SD; images with > 90% payload recovery
Candidate-set analysis	Original payload codeword retained at Levenshtein distance 0, 1, or 2; original codeword absent from the candidate set
Image reconstruction	MSE; PSNR; SSIM; exact pixel recovery (%)

Table 1. Evaluation stages and performance metrics used to assess the proposed framework.

same encoding, nanopore simulation, consensus reconstruction, decoding, and RF-filtered reconstruction workflow as the primary evaluation. Reconstruction performance was quantified using PSNR and SSIM and compared with published image results for DNA-QLC and INNSE on the same dataset.

Results

Experimental setup

To evaluate the proposed marker-aware codebook construction strategy and image reconstruction framework, experiments were performed using the first 1,000 natural RGB images from the ImageNet dataset[32]. The images were divided into non-overlapping subsets of 500 training, 250 validation, and 250 test images. The training subset was used to generate the data required for training the Random Forest reliability classifiers, the validation subset was used during model development and parameter evaluation, and the final 250 images were reserved exclusively for testing the complete reconstruction pipeline. Images were represented as 24-bit RGB pixel values.

For nanopore read simulation, 10 reads were generated per encoded sequence, with a mean read length corresponding to the full encoded sequence length and a random seed = 1, providing approximately 10x nominal sequence coverage. For reference-free consensus reconstruction, the reads were aligned using minimap2 (v2.22) with the *map-ont* preset, and the consensus was polished using two successive rounds of Racon (v1.4.3)[30,33] .

The target payload codebook therefore comprised the complete 24-bit RGB color space, corresponding to 2^24 distinct color values. Under an ideal quaternary DNA representation, where each nucleotide stores 2 bits of information, a 24-bit RGB value has a theoretical minimum representation of 12 nt. In the proposed framework, each RGB value was represented by 16-nt payload codeword together with an 8-nt synchronization marker, corresponding to an effective storage density of 1 bit/nt. The increased sequence length provided sufficient sequence-space redundancy to accommodate biochemical quality control, sequence-space discrimination, and structural compatibility constraints while maintaining robust decoding under nanopore sequencing errors.

Before payload codeword generation, a pool of 100 marker sequences (8 nt) was generated using marker-specific biochemical constraints, strict pairwise Hamming-distance separation ($d_{marker} = 3$), and penalty-based candidate selection. Payload candidates were subsequently evaluated using local Hamming-neighborhood occupancy control with a monitored radius of $r = 2$, and an initial adaptive occupancy limit K set to 4.

Biochemical quality control applied hard constraints on GC content, homopolymer length, and contiguous self-complementarity, followed by penalty-based scoring ($P(x) \leq 25$). Payload candidates were required to contain a GC count between 6 and 10 nucleotides, a maximum homopolymer length of 3 nt, and contiguous self-complementarity below 7 nt before penalty-based evaluation. Structural compatibility was enforced through exact marker exclusion together with terminal identity checks ($k_{id} = 4$), reverse-complement-aware checks ($k_{comp} = 5$), and terminal self-clash rejection. For each encoded image, 10 simulated nanopore reads were used for reference-free consensus reconstruction.

Sequence decoding, initial synchronization, downstream validation, and resynchronization were performed using four-marker chains. The edit distance thresholds were $\tau_{init} = 2$, $\tau_{resync} = 3$, $\tau_{fallback} = 4$. Marker search windows of 10 nt and 16 nt were used during sequential decoding and resynchronization, respectively. Resynchronization considered up to 24 downstream marker positions and skipped marker positions were not permitted during sequential decoding.

Sequence decoding performance and candidate-set analysis

To investigate how different codebook construction strategies influence downstream decoding, each codebook was evaluated under identical nanopore sequencing conditions using the proposed marker-guided decoding framework. Decoded performance was first assessed according to the proportion of successfully recovered payloads, after which candidate-set quality was evaluated by determining whether the original payload codeword remained identifiable following decoding.

Table 2 summarizes decoder performance across all codebook-generation strategies. Overall, decoded recovery remained high regardless the applied constraint combination, with mean payload recovery exceeding 91% in every scenario. Codebooks incorporating biochemical quality control achieved the highest recovery rates, with the biochemical-only and sequence-space plus biochemical strategies recovering 94.13% and 94.10% of payload, respectively. The marker-aware codebook generated using the complete constraint set achieved a comparable recovery rate of 93.61% and retained more than 90% of pixel payloads in 95.9% of images, indicating that the combined constraint set maintained high decoder-level recovery.

The adaptive occupancy threshold remained at $K = 4$ for the sequence-space-only and sequence-space+biochemical codebooks. In contrast, K increased from 4 to 15 for the sequence-space+structural codebook and from 4 to 16 for the full-constraints codebook, with the first increases occurring after 7.16 and 6.58 million accepted codewords, respectively. Across 20 million sampled codeword pairs per codebook, the final pairwise Hamming-distance distributions remained centered near 12 (mean 11.96–12.00; SD 1.73–1.78). Very close pairs were rare in all cases, with $P(d_H \leq 2)$ ranging from 5×10^{-8} to 1.6×10^{-6}, although structurally constrained codebooks showed a somewhat higher close-pair tail. Thus, adaptive relaxation enabled completion of the more restrictive codebooks without substantially altering the overall distance distribution. Although decoder recovery was similar across all constrained codebooks, successful payload recovery alone does not indicate whether the correct payload codeword can subsequently be identified. Therefore, the quality of the candidate sets generated after decoding was evaluated separately.

Figure 4 summarizes the availability of the original payload codeword among candidate sequences following decoding. Candidate sets were classified according to whether the original codeword was recovered as an exact match (Levenshtein distance 0), retained within the candidate set at distances 1 or 2, or absent from the candidate set.

Scenario	Adaptive occupancy limit (K)	Mean sequencing error (%)	Mean read identity (%)	Payload recovery, mean ± SD (%)	Images with >90% recovered pixels (%)
Random baseline (random_unique_baseline_cb)	4	5.27 ± 1.26	94.81 ± 1.22	92.73 ± 3.29	83.9
Structural compatibility (structural_only_cb)	4	5.16 ± 1.32	94.92 ± 1.28	91.45 ± 5.78	65.4
Biochemical quality control (bio_only_cb)	4	5.20 ± 1.25	94.88 ± 1.21	94.13 ± 4.94	98.5
Biochemical + structural (bio_structural_cb)	4	5.31 ± 1.28	94.78 ± 1.24	93.28 ± 3.89	91.6
Sequence-space discrimination (sequence_space_only_cb)	4	5.35 ± 1.33	94.74 ± 1.29	93.08 ± 7.63	94.2
Sequence-space + structural (sequence_space_structural_cb)	15	5.27 ± 1.24	94.81 ± 1.20	92.42 ± 3.10	78.1
Sequence-space + biochemical (sequence_space_bio_cb)	4	5.07 ± 1.28	95.02 ± 1.24	94.10 ± 6.45	97.8
Full constraints (full_constraints_cb)	16	5.03 ± 1.26	95.05 ± 1.22	93.61 ± 5.70	95.9

Table 2. Sequence decoding performance for different codebook construction strategies. Values are reported as mean ± standard deviation across 1,000 simulated images. The adaptive occupancy limit is reported only for codebook employing sequence-space discrimination.

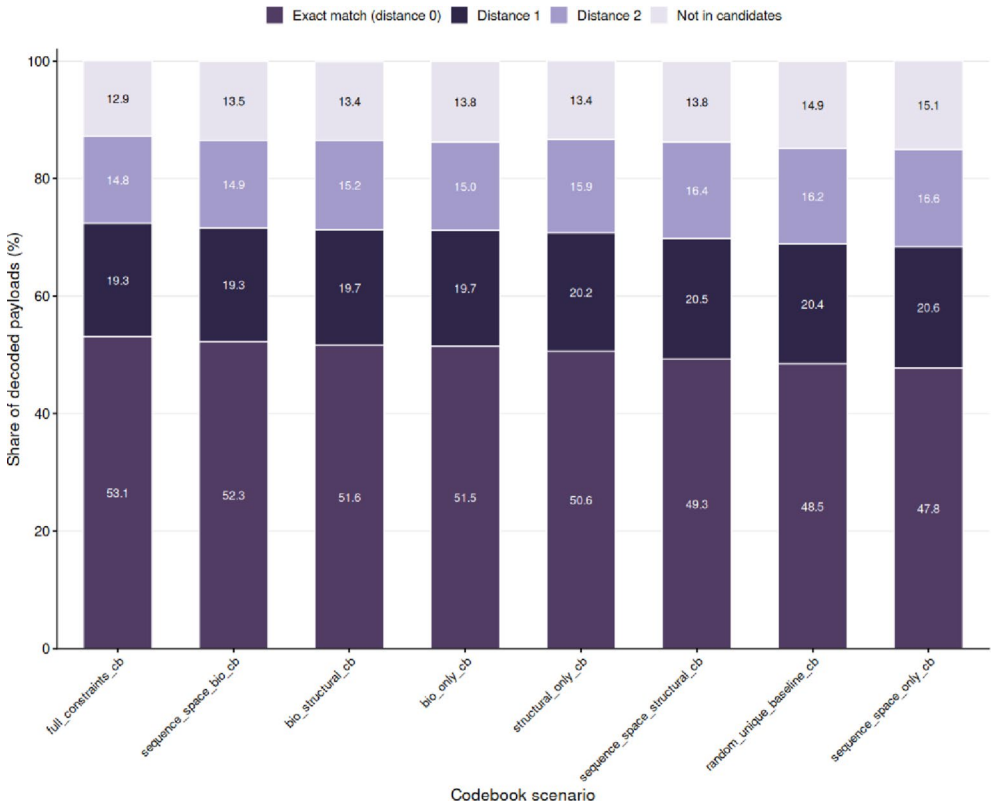


Fig. 4. Candidate-set quality across different codebook construction strategies. Decoded payloads were classified according to whether the original payload was recovered as an exact match (Levenshtein distance 0), retained within the candidate set at Levenshtein distances 1 or 2, or absent from the candidate set.

The marker-aware codebook generated using the complete constraint set produced the highest proportion of exact matches (53.06%) while also yielding the lowest proportion of cases in which the original payload was absent from the candidate set (12.89%). Although biochemical quality control alone resulted in a slightly higher decoder recovery rate, the complete constraint set more effectively preserved the correct payload among the candidate sequences generated after decoding.

These findings demonstrate that decoder recovery alone is insufficient for evaluating codebook quality. Codebooks with comparable payload recovery rates can differ substantially in their ability to preserve the correct payload within the candidate set, directly influencing the subsequent image reconstruction stage.

The full-constraints scenario produced the highest proportion of exact candidate matches, with 53.06% of decoded payloads retaining the original RGB codeword at Levenshtein distance 0. It also showed the lowest proportion of cases in which the original codeword was absent from the candidate set, at 12.89%. These findings indicate that the full-constraints scenario provided the strongest overall candidate-set quality, which was the intended objective of the proposed codebook construction. Although biochemical quality control alone produced a slightly higher recovered-pixel proportion, the full-constraints codebook more frequently preserved the correct RGB codeword as an exact candidate after simulated sequencing errors. This result is consistent with the combined action of sequence-space discrimination, biochemical filtering, and structural compatibility constraints, which reduce competing sequence patterns and improve the probability that the original codeword remains identifiable during candidate-assisted reconstruction.

False exact matches were additionally quantified across all available decoded images. Among payloads for which an exact candidate was identified, the false exact-match rate ranged from 0.185% to 1.063% across codebooks and was 0.968% for the full-constraints codebook. When all candidates within $d_L \leq$ 2were retained, 96.56–97.05% of non-empty candidate sets contained multiple candidates, including 97.04% for the full-constraints codebook.

Image reconstruction results

Because the marker-aware codebook generated using the complete constraint set provided the highest candidate-set quality, it was selected for the image reconstruction experiments. The reconstructed payloads obtained after marker-guided decoding were subsequently converted to RGB values using the generated codebook, and three restoration strategies were evaluated to determine how candidate ambiguity influences the final image quality.

Illustrative decoding example

Figure 5 illustrates representative decoding outcomes obtained using the marker-aware codebook. Following marker-guided decoding, recovered payloads were compared with the original payload sequence and the corresponding codebook entries within the selected Levenshtein-distance threshold. Depending on the sequencing errors present, the original payload could be recovered as an exact match, retained among the candidate sequences, or excluded from the candidate set.

Figure 5a shows an example interval of a decoded sequence, whereas Fig. 5b presents the corresponding candidate RGB assignments. Exact payload recovery resulted in a unique RGB assignment, while payloads recovered with one or two sequence differences produced multiple candidate colors. In some cases, the original payload was absent from the candidate set entirely, preventing direct reconstruction and requiring subsequent image restoration. These examples illustrate how candidate ambiguity increases as sequencing errors accumulate, motivating the subsequent image restoration procedures. A maximum Levenshtein distance of 2 was retained because the correct RGB codeword was sometimes recovered only at distance 2. Restricting the search to distance 1 would reduce the number of candidates but could exclude the correct codeword while still retaining incorrect alternatives, as illustrated in Fig. 5.

Machine-learning-assisted reliability filtering

Exact-match and candidate-assisted reconstruction both recovered substantially more pixels, but neither approach could guarantee that the assigned RGB value was actually correct. Sequencing errors and ambiguous candidates sometimes led to pixel values that clashed with their surroundings, visibly degrading the reconstructed image. To catch these problematic assignments before interpolation, we trained a Random Forest reliability filter on decoded images produced with the full-constraint codebook.

Separate classifiers were developed for exact-match and candidate-assisted assignments. The models were trained using balanced datasets comprising 100,000 and 200,000 labelled examples, respectively, generated from 500 reconstructed training images. Performance was evaluated using two independent sets of 250 images each for validation and testing. Both classifiers employed 300 decision trees with a maximum depth of 18, a minimum leaf size of 5, balanced class weights, bootstrap sampling, and out-of-bag (OOB) evaluation. In addition to OOB evaluation, classifier performance was assessed on the independent test sets. The exact-assignment classifier achieved an accuracy of 0.9547, a macro F1-score of 0.9533, and a ROC-AUC of 0.9919. The candidate-assignment classifier achieved an accuracy of 0.9534, a macro F1-score of 0.9533, and a ROC-AUC of 0.9909.

Figure 6 summarizes the training results. Both classifiers reached stable OOB performance before 300 trees, indicating that further increases in model complexity provided little additional benefit. The candidate-assignment classifier reached a final OOB score of 0.9496, outperforming the exact-assignment classifier at 0.8867. Feature-importance analysis showed that reliability decisions were driven by combination of local image-context features and candidate-assignment characteristics rather than sequence distance alone.

During image reconstruction, exact assignments were evaluated first, followed by candidate-assisted assignments. Pixel assignments with predicted reliability below 0.55 were rejected and treated as unresolved before interpolation. This additional reliability assessment reduced propagation of locally inconsistent RGB assignments and formed the basis for the quantitative reconstruction comparison presented in the following section. A fixed reliability threshold of 0.55 was used as a conservative probability cutoff for both classifiers rather than being optimized on the test set. Assignments with predicted probability of correctness below this threshold were rejected and treated as unresolved before subsequent candidate selection or interpolation. This additional reliability assessment reduced propagation of locally inconsistent RGB assignments and formed the basis for the quantitative reconstruction comparison presented in the following section.

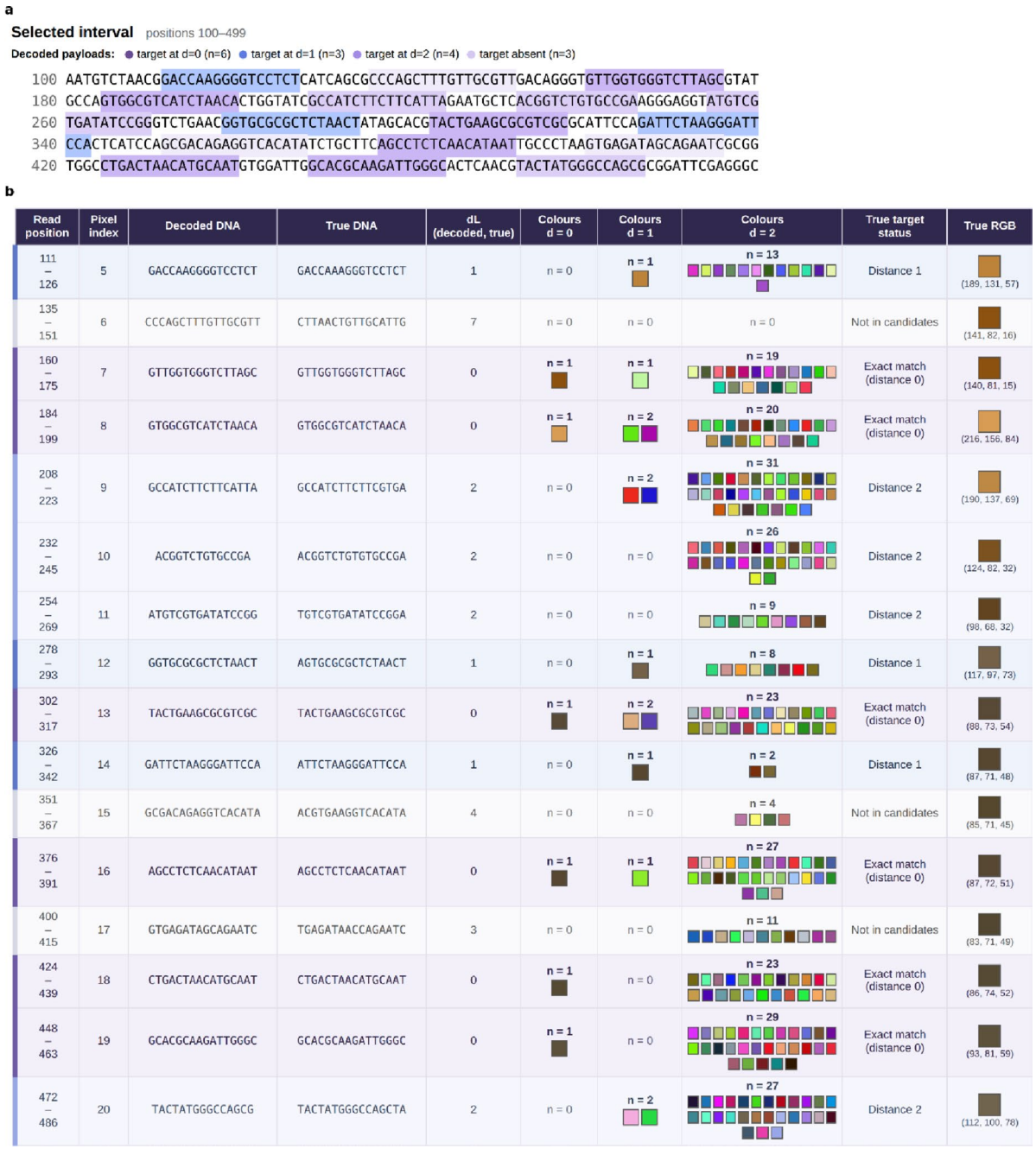


Fig. 5. Illustrative example of marker-guided decoding and candidate generation: (**a**) Segment of a decoded consensus sequence showing exact matches, near matches, and absent payloads. (**b**) Representative decoded payloads with candidate RGB values, classified by candidate-set status.

Image reconstruction performance

The three restoration strategies were evaluated using 250 independently reconstructed test images generated with the full-constraints codebook. Reconstruction quality was assessed using mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and exact pixel recovery (Table 3). Representative reconstruction examples are shown in Fig. 7.

Candidate-assisted reconstruction substantially increased exact pixel recovery from 53.17% to 79.08%, demonstrating that the candidate searches successfully assigned many previously unresolved pixels. However, this increase did not translate into improved reconstruction quality. As shown in Fig. 7c, selecting the most likely candidate without an additional reliability assessment introduced numerous incorrect RGB assignments, resulting in widespread color artefacts across all reconstructed images. Consequently, candidate-assisted reconstruction produced the highest reconstruction error (MSE = 634.55 ± 298.79), the lowest PSNR (20.51 ± 1.90 dB), and the lowest SSIM (0.5194 ± 0.1319) among the evaluated methods.

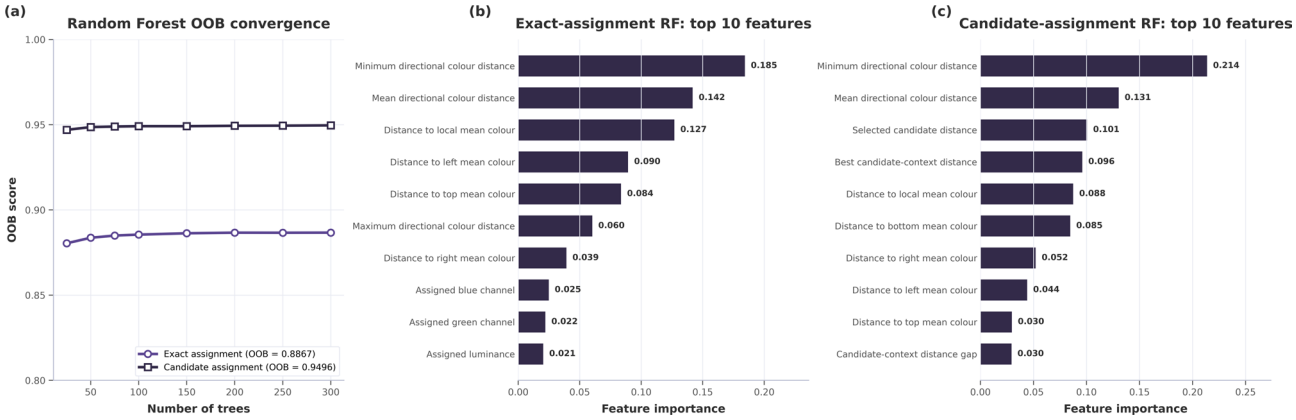


Fig. 6. RF training diagnostics for assignment-reliability filtering: (**a**) Out-of-bag performance vs. number of trees for both classifiers. (**b**) Top 10 features, exact-assignment classifier. (**c**) Top 10 features, candidate-assignment classifier.

Method	Images	MSE	PSNR (dB)	SSIM	Exact pixel recovery (%)
Exact match + interpolation	250	166.05 ± 110.80	26.83 ± 2.82	0.8630 ± 0.0530	53.17 ± 9.69
Candidate-assisted + interpolation	250	634.55 ± 298.79	20.51 ± 1.90	0.5194 ± 0.1319	79.08 ± 6.92
RF-filtered + interpolation	250	124.52 ± 122.48	29.82 ± 5.58	0.9309 ± 0.0538	80.19 ± 6.92

Table 3. Quantitative comparison of image restoration strategies using the full-constraints codebook. Values are reported as mean ± standard deviation across 250 test images.

Applying Random Forest reliability filtering substantially improved reconstruction quality while preserving the high level of direct pixel recovery. Compared with candidate-assisted reconstruction, RF filtering reduced the MSE by 80.4% (634.55 to 124.52), increased PSNR by 9.31 dB (20.51 to 29.82 dB), improved SSIM from 0.5194 to 0.9309, and slightly increased exact pixel recovery from 79.08% to 80.19%. The corresponding reconstructions (Fig. 7d) show that removing unreliable candidate assignments effectively suppresses the color artifacts observed in candidate-assisted reconstruction while preserving image structure and fine visual details.

Compared with the exact-match baseline (Fig. 7b), RF filtering further reduced the reconstruction error by approximately 25.0%, increased PSNR by 2.99 dB, improved SSIM from 0.8630 to 0.9309, and increased exact pixel recovery by 27.02% points. Across all five representative images in Fig. 7, RF-filtered reconstruction consistently produced images that were visually closest to the originals (Fig. 7a), supporting the quantitative improvements observed in Table 3. Together, these results demonstrate that incorporating candidate information improves image reconstruction only when unreliable assignments are identified and removed before interpolation.

Comparison with existing DNA image storage approaches

To provide an external comparison with existing DNA image storage approaches, the proposed framework was additionally evaluated using the 24-image Kodak dataset, for which image-level reconstruction results have been reported for DNA-QLC and INNSE. All 24 Kodak images were processed using the same DNA encoding, nanopore read simulation, consensus reconstruction, marker-guided decoding, and RF-filtered reconstruction workflow used in the primary experiment. The proposed method was evaluated under the standard DeepSimulator-based nanopore-like conditions used in this study, corresponding to an observed sequencing error rate of approximately 5%.

Across the 24 Kodak images, the proposed method successfully reconstructed all images and achieved a mean PSNR of 29.93 ÷ 3.19 dB and mean SSIM of 0.9406 ÷ 0.0321. For comparison, the published INNSE evaluation of the same dataset at a 1% simulated error rate yielded a mean PSNR of 31.53 ÷ 3.26 dB and SSIM of 0.884 ÷ 0.048 across 24 images. DNA-QLC results reported 26.53 ÷ 9.33 dB and 0.754 ÷ 0.22, respectively, across the 22 successfully reconstructed images, while two images were reported as unsuccessful. These results are shown in Table 4 and several reconstructed Kodak dataset images are shown in Fig. 8.

The proposed method therefore achieved a mean PSNR intermediate between the published DNA-QLC and INNSE results, while achieving the highest mean SSIM and successfully reconstructing all 24 images. However, these values should not be interpreted as a direct performance ranking because the experimental conditions differ. DNA-QLC and INNSE used learned image representations and were evaluated at a 1% controlled simulated error rate, whereas the proposed framework directly encodes RGB values and was evaluated under nanopore-like sequencing conditions with an observed error rate of ~ 5%. The comparison nevertheless demonstrates that the proposed marker-guided framework maintains high image reconstruction quality on an established image dataset.

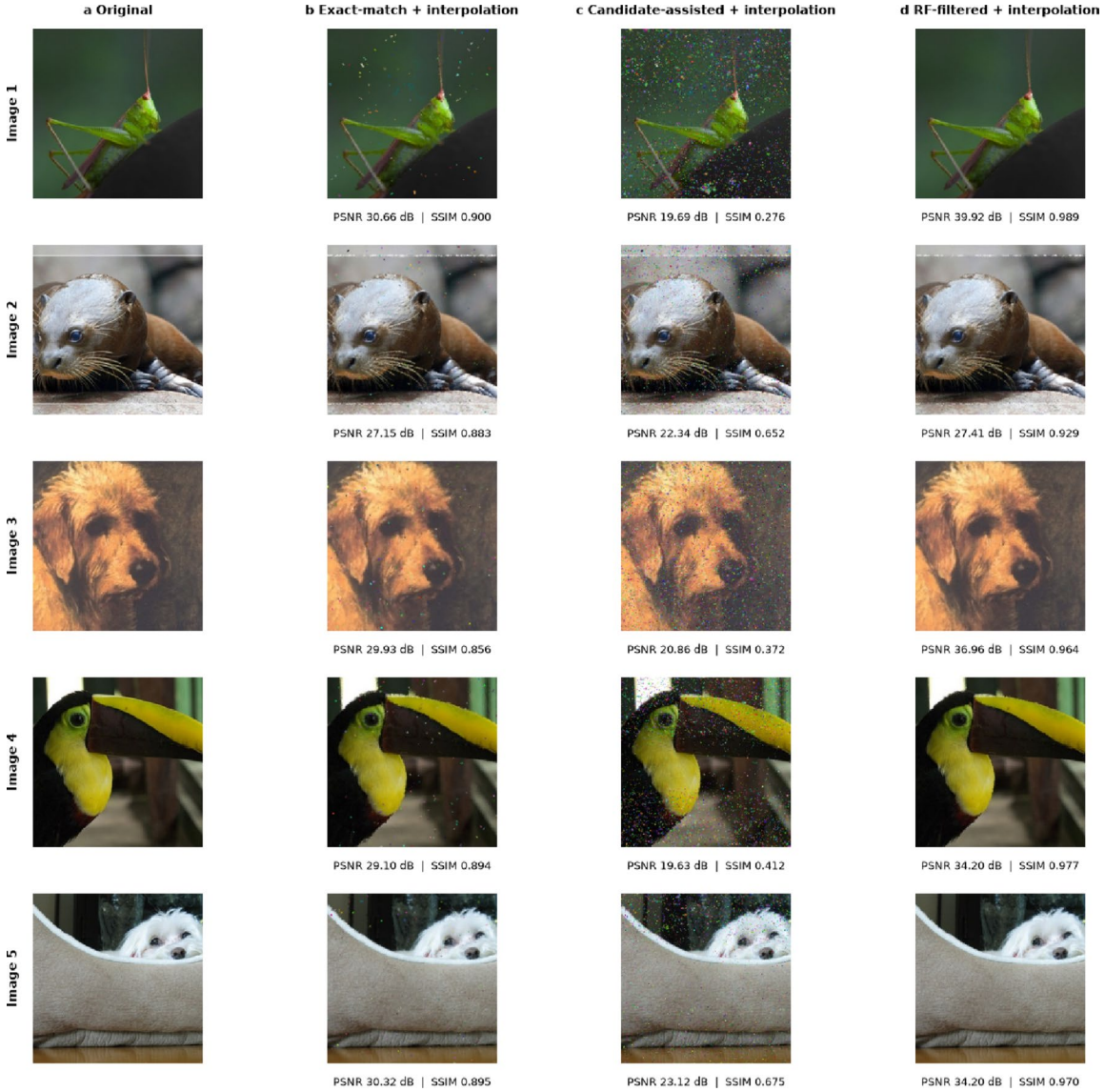


Fig. 7. Representative image reconstructions obtained with the full-constraints codebook: (**a**) Original image. (**b**) Exact-match reconstruction with interpolation. (**c**) Candidate-assisted reconstruction with interpolation. (d) Candidate-assisted reconstruction with RF reliability filtering and interpolation. PSNR and SSIM reported for each image.

Method	Image representation	Error model	Error rate	Successful reconstructions	PSNR (dB)	SSIM
DNA-QLC	QRes-VAE compressed	Simulated S/D/I	1%	22/24	26.53 ± 9.33[a]	0.754 ± 0.22[a]
INNSE	Neural network based	Simulated S/D/I	1%	24/24	31.53 ± 3.26	0.884 ± 0.048
Proposed method	Raw RGB pixels	DeepSimulator + consensus from 10 reads	~ 5%[b]	24/24	29.93 ± 3.19	0.9406 ± 0.0321

Table 4. Contextual comparison of image reconstruction performance on the 24-image Kodak dataset. [a]Mean ± SD calculated from the 22 successfully reconstructed images reported for DNA-QLC; Kodak 19 and 24 were reported as N/A. [b]Observed sequencing error of approximately 5% in the DeepSimulator-based nanopore evaluation; unlike DNA-QLC and INNSE, this is not a directly injected 1% S/D/I condition.

Discussion

The framework combines marker-aware codebook construction, marker-guided sequence decoding with resynchronization, candidate-based RGB assignment, and machine-learning-assisted reliability filtering. The central finding of this study is that high payload recovery alone does not guarantee accurate image reconstruction. Decoded payloads may remain ambiguous or be assigned to incorrect RGB values even when sequencing-level

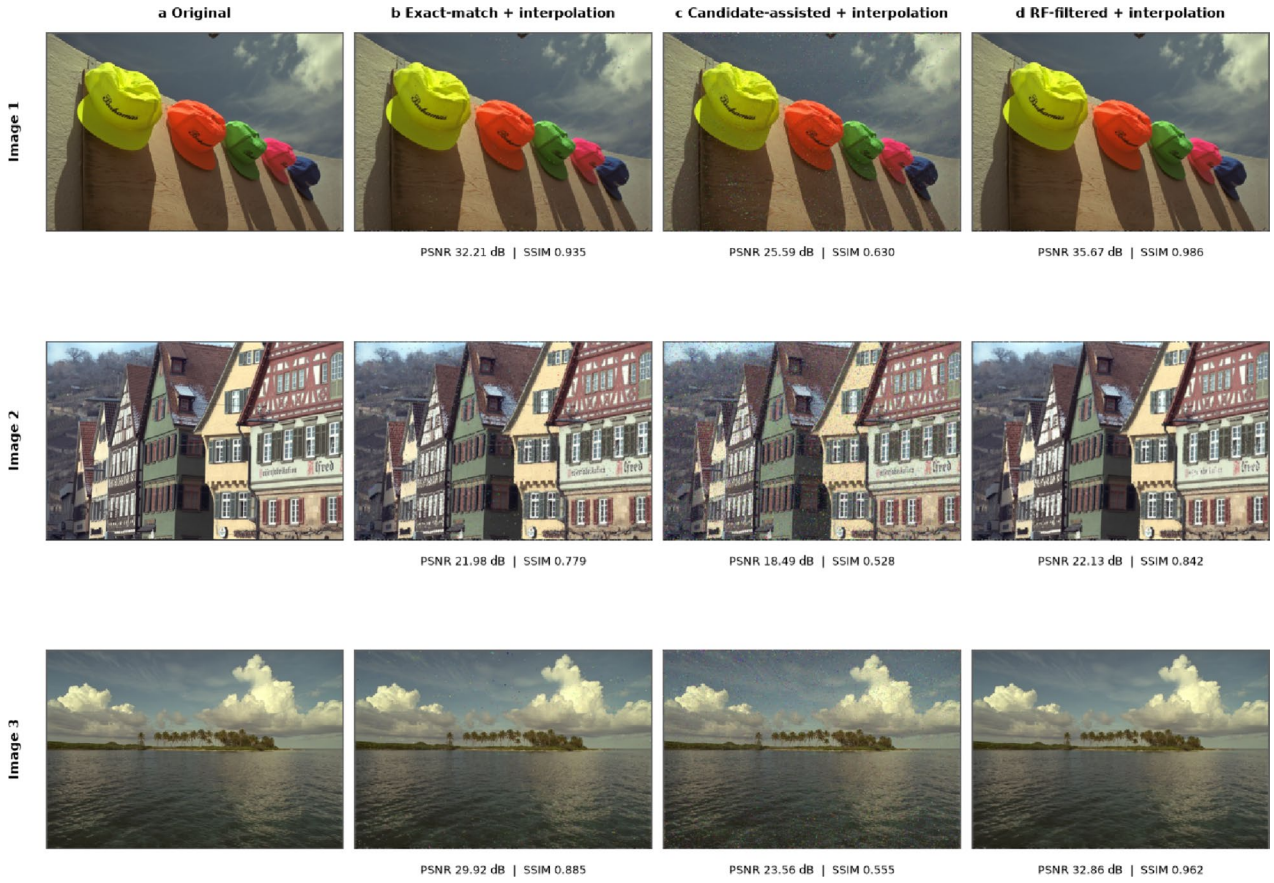


Fig. 8. Representative image reconstruction results from the Kodak dataset. Original images (**a**) are compared with exact-match + interpolation (**b**), candidate-assisted + interpolation (**c**), and RF-filtered + interpolation (**d**). PSNR and SSIM values are shown for each reconstructed image.

recovery is high. This distinction between decoder recovery and reconstruction quality was consistent across all evaluated codebook construction strategies and highlights the importance of considering downstream data reconstruction when assessing DNA storage performance.

Although robust sequence reconstruction and error correction remain central objectives in DNA data storage, most evaluation approaches focus primarily on sequence- or byte-level recovery. These metrics quantify decoding performance but do not capture what happens once recovered payloads are mapped to application-level data such as images. Our results show that residual ambiguity in decoded payloads can propagate into incorrect RGB assignments, reducing reconstruction quality even when payload recovery remains high. This highlights the importance of evaluating downstream candidate ambiguity alongside conventional decoding metrics, particularly for image-based DNA storage applications.

Constrained coding has also become an important strategy for improving DNA storage reliability. DNA-Aeon, for example, supports user-defined GC content, homopolymer limits, motif avoidance, and custom codebooks while correcting substitutions, insertions, deletions, and strand loss[23]. More recently, Whritenour et al. proposed constrained coding specifically for error mitigation in nanopore-based DNA data storage[22]. Unlike recent approaches that jointly optimize biological constraints through learned or differentiable objective functions, the present framework applies explicit rule-based constraint combinations and evaluates their downstream effects on marker-guided decoding, candidate ambiguity, and image reconstruction[24]. The eight codebook strategies were evaluated primarily as an ablation analysis of how different combinations of sequence-space, biological, and structural constraints affect decoding and candidate-set quality. The full-constraints codebook showed the strongest overall candidate quality and was therefore selected for subsequent image-reconstruction experiments.

The results show that decoder recovery and reconstruction quality are not equivalent. Although biochemical quality control produced the highest mean payload recovery, the full-constraints codebook produced the strongest candidate-set quality, with the highest proportion of exact original-codeword matches and the lowest proportion of absent original codewords. This distinction is important because candidate-assisted reconstruction increased exact pixel recovery but degraded PSNR and SSIM when incorrect candidates were accepted. Reliability filtering reversed this effect: the RF-filtered strategy achieved the best overall reconstruction performance, with MSE = 124.52 ± 122.48, PSNR = 29.82 ± 5.58 dB, SSIM = 0.9309 ± 0.0538, and 80.19% ± 6.92% exact pixel recovery.

Recent learning-based approaches have also applied machine learning directly to DNA sequence reconstruction. For example, Li et al. proposed a multi-scale attention and contrastive-learning framework for reconstructing DNA sequences from noisy sequencing reads under high-error conditions[34]. This operates at a different stage from the machine-learning component used in the present framework. Here, Random Forest classification is not used to reconstruct the DNA sequence itself; instead, it is applied after consensus reconstruction and marker-guided decoding to assess the reliability of RGB assignments and reject potentially incorrect pixel value before image interpolation.

Evaluation on the Kodak dataset provided additional context relative to existing DNA image storage approaches. Across all 24 images the proposed method achieved a mean PSNR of 29.93 ± 3.19 dB and SSIM of 0.9406 ± 0.0321 under nanopore-like sequencing conditions with an observed error rate of approximately 5%. Published results for INNSE and DNA-QLC on the same dataset were obtained under a lower 1% simulated error rate and using different image representations. The proposed method therefore cannot be directly ranked against these approaches; nevertheless, its consistent reconstruction of all 24 images supports the robustness of the marker-guided reconstruction strategy on an established image benchmark.

The computational cost of the proposed framework is concentrated primarily in codebook construction. Because the complete 24-bit RGB codebook is generated once and subsequently reused, the step does not need to be repeated for each stored image. During codebook construction, local Hamming-neighborhood evaluation is used instead of exhaustive pairwise comparison against the complete codebook, reducing computational burden as the codebook grows. Once the codebook is available, image encoding is performed through direct RGB-to-codeword lookup and therefore scales linearly with the number of pixels. In the current representation, each pixel requires 24 nt, corresponding to 1,204,224 nt for 224×224 RGB image. Decoding additionally requires the codebook and marker set as side information. Using a compact 2-bit-per-nucleotide representation, the complete 2^24-entry codebook of 16-nt payload sequences requires approximately 64 MiB, while the 100 8-nt markers require approximately 200 bytes. This information is fixed for a given codebook and can be reused across all images encoded with that codebook, so its relative overhead decreases as the amount of stored data increases.

A limitation of the current framework is that images are represented as raw RGB pixel values rather than as compressed image files such as PNG or JPEG. This allows reconstruction errors to remain localized at pixel level, whereas errors in compressed image data may affect file decoding more broadly and may prevent correct reconstruction of the image. Therefore, extending the proposed framework to compressed image formats would require an appropriate encoding and error-protection strategy for the compressed data representation.

A further limitation is that the framework was evaluated using simulated nanopore sequencing rather than experimentally generated reads. DeepSimulator enabled all codebook strategies to be evaluated under consistent sequencing conditions, but simulated reads cannot fully reproduce the variability introduced by DNA synthesis, storage, and amplification.

Overall, these findings suggest that DNA storage systems, especially those intended for nanopore readout, should be evaluated not only by sequence recovery or error rate, but also by the reliability of downstream data assignment. For image-based payloads, visually and quantitatively accurate reconstruction depends on whether the correct codeword remains distinguishable after sequencing and decoding. The proposed framework therefore complements existing reconstruction and constrained-coding approaches by adding an image-level evaluation layer that links sequence design, candidate ambiguity, and final reconstruction quality.

Conclusions

This study presents a proof-of-concept framework for DNA-based RGB image storage and reconstruction that combines marker-aware codebook construction, marker-guided sequence decoding, candidate-assisted image reconstruction, and machine-learning-based reliability filtering. The results demonstrate that codebook design influences not only payload recovery but also directly affects reconstruction performance. The full-constraints codebook was selected for final image reconstruction based on its candidate-set quality, and Random Forest reliability filtering substantially improved reconstruction quality by removing unreliable pixel assignments.

The proposed framework was evaluated using simulated nanopore sequencing data and RGB images of fixed size under controlled conditions. Consequently, the findings should be validated using experimentally generated nanopore reads and assessed for larger images and additional data types. Future work will also focus on alternative encoding configurations to further optimize the trade-off between storage density, decoding robustness, and reconstruction quality.

Data availability

The source code, generated DNA codebooks, and supporting files required to reproduce the computational workflow are publicly available in Zenodo at DOI:10.5281/zenodo.23021414.

Received: 16 July 2026; Accepted: 25 September 2026
Published online: 03 November 2026

References

1. Church, G. M., Gao, Y. & Kosuri, S. Next-Generation Digital Information Storage in DNA. *Science* **337**, 1628–1628 (2012).
2. Goldman, N. et al. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **494**, 77–80 (2013).
3. Bornholt, J. et al. A DNA-based archival storage system. *ACM SIGPLAN Notices*. **51**, 637–649 (2016).
4. Organick, L. et al. Random access in large-scale DNA data storage. *Nat. Biotechnol.* **36**, 242–248 (2018).
5. Meiser, L. C. et al. Reading and writing digital data in DNA. *Nat. Protoc.* **15**, 86–101 (2020).

6. Ceze, L., Nivala, J. & Strauss, K. Molecular digital data storage using DNA. *Nat. Rev. Genet.* **20**, 456–466 (2019).
7. Erlich, Y. & Zielinski, D. D. N. A. Fountain Enables a Robust and Efficient Storage Architecture. *Science* **35**, 950–954 (2017).
8. Ping, Z. et al. Towards practical and robust DNA-based data archiving using the yin–yang codec system. *Nat. Comput. Sci.* **2**, 234–242 (2022).
9. Antkowiak, P. L. et al. Low cost DNA data storage using photolithographic synthesis and advanced information reconstruction and error correction. *Nat. Commun.* **11**, 5345 (2020).
10. Yazdi, S. M. H. T., Gabrys, R. & Milenkovic, O. Portable and Error-Free DNA-Based Data Storage. *Sci. Rep.* **7**, 5011 (2017).
11. Lopez, R., Chen, Y. J. & Ang, D. DNA assembly for nanopore data storage readout. *Nat. Commun.* **10**, 2933 (2019).
12. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338–345 (2018).
13. Zhou, Y. et al. Advances and Challenges in Solid-State Nanopores for DNA Sequencing. *Langmuir* **41**, 5736–5761 (2025).
14. Amarasinghe, S. L. et al. Opportunities and challenges in long-read sequencing data analysis. *Genome Biol.* **21**, 30 (2020).
15. Chen, X. et al. Addressing the Challenges of Solid-State Nanopores: Strategies for Performance Enhancement. *Int. J. Mol. Sci.* **27**, 2536 (2026).
16. Magi, A., Giusti, B. & Tattini, L. Characterization of MinION nanopore data for resequencing analyses. *Brief. Bioinform.* **18**, 940–953 (2017).
17. Delahaye, C. & Nicolas, J. Sequencing DNA with nanopores: Troubles and biases. *PLoS One* **16**, (2021).
18. Santos, R. et al. Investigating the Performance of Oxford Nanopore Long-Read Sequencing with Respect to Illumina Microarrays and Short-Read Sequencing. *Int. J. Mol. Sci.* **26**, 4492 (2025).
19. Volkel, K. D. et al. Nanopore decoding with speed and versatility for data storage. *Bioinformatics* **41**, (2025).
20. Press, W. H. et al. HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints. *Proc. Natl. Acad. Sci.* **117**, 18489–18496 (2020).
21. Liu, B., Wang, F., Fan, C. & Li, Q. Data Readout Techniques for DNA-Based Information Storage. *Adv Mater* **37**, (2025).
22. Whritenour, K., Civelek, M. & Farnoud, F. Constrained coding for error mitigation in nanopore-based DNA data storage. *Sci. Rep.* **15**, 33899 (2025).
23. Welzel, M. et al. DNA-Aeon provides flexible arithmetic coding for constraint adherence and error correction in DNA storage. *Nat. Commun.* **14**, 628 (2023).
24. Liu, X. et al. An end-to-end DNA storage coding method based on a low-complexity multiple biological constraints loss and RL-inspired differentiable solver. *Expert Syst. App.* **315**, 131726 (2026).
25. Zheng, Y. et al. DNA-QLC: an efficient and reliable image encoding scheme for DNA storage. *BMC Genom.* **25**, 266 (2024).
26. Zheng, Y. et al. INNSE: Invertible neural network-based DNA image storage with self-correction encoding. *Comput. Struct. Biotechnol. J.* **27**, 2492 (2025).
27. Beslic, D. et al. End-to-end simulation of nanopore sequencing signals with feed-forward transformers. *Bioinformatics* **41**, (2024).
28. Xie, S. et al. NanoSimFormer: an end-to-end Transformer-based nanopore signal simulator with basecaller guidance. *Bioinformatics* **42**, (2026).
29. Li, Y. et al. DeepSimulator1.5: a more powerful, quicker and lighter simulator for Nanopore sequencing. *Bioinformatics* **36**, 2578–2580 (2020).
30. Vaser, R. et al. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res.* **27**, 737–746 (2017).
31. Eastman Kodak Company. *Kodak Lossless True Color Image Suite (PhotoCD PCD0992)*. (1993). https://r0k.us/graphics/kodak/
32. Deng, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. (2009).
33. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
34. Li, X. et al. Highly biased DNA sequence reconstruction in DNA storage with multi-scale attention mechanism and contrast learning. *Synth. Syst. Biotechnol.* **12**, 422–432 (2026).

Author contributions

MK, AP, AK, BF conceived the study, designed the methodology, developed the software. MK and AK performed the computational experiments, analyzed the data, and prepared the figures. AK supervised the research and contributed to the conceptualization of the study. All authors wrote the manuscript.

Funding

This research was co-funded by the European Union under Horizon Europe programme grant agreement No. 101059903, by the European Union funds for the period 2021–2027, and by the state budget of the Republic of Lithuania financial agreement Nr. 10-042-P-0001.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2026