GENOME COMPRESSION USING QCNN :A NEW APPROACH FOR THE QUANTUM WORLD

1st SRI HARI VASAN A
AIE department
AMRITA VISWA VIDYAPEEDAM
CHENNAI, INDIA
sriharivasanop@gmail.com

2nd VIGNESHWARRAN S
AIE DEPARTTMENT (of Aff.)
AMRITA VISHWA VIDYAPEEDAM(of Aff.)
CHENNAI,INDIA
vigneshwarran0@gmail.com

3rd ROHIT MUGALYA A R
AIE DEPARTMENT (of Aff.)
AMRITA VISHWA VIDYAPEEDAM
CHENNAI,INDIA
rohitmugalya@gmail.com

4th SANJJEY A
AIE DEPARTMENT (of Aff.)
AMRITA VISHWA VIDYAPEEDAM (of Aff.)
CHENNAI,INDIA
sanjjeyarumugam@gmail.com

5th MUKESH CHARAN M
AIE DEPARTMENT (of Aff.)
AMRITA VISHWA VIDYAPEEDAM (of Aff.)
CHENNAI,INDIA
mukeshcharan2905@gmail.com

***Abstract*—This novel approach enhances the genome compression process by combining Quantum Convolution Neural Network(QCNN) where the qubits are mapped to nucleotides and Classical Convolution Neural Network(CNN) which improves the time efficiency and accuracy with higher compression ratio with minimal information loss and can be also implemented on large-scale datasets with ease. This approach is comparatively better than classical CNN.**

***Index Terms*—Hybrid Model, QCNN, CNN, Compression, Genome, Qubit**

I. Introduction

Revolutionary advances in DNA sequencing technologies have fundamentally transformed the landscape of genomics, leading to an exponential increase in genomic data volume. The complete set of DNA of an organism, known as the genome, encodes the genetic instructions essential for development, functioning, growth, and reproduction. The growing availability of genomic data has fueled numerous applications, including precision medicine, long-term data storage, food safety, ancestry analysis, and evolutionary studies. However, as sequencing costs continue to decrease, the sheer volume of genomic data being generated presents significant challenges in terms of storage, transmission, and processing, underscoring the need for efficient and scalable data compression techniques. While general-purpose compression algorithms exist, they often fail to capitalize on the unique structural properties of DNA sequences, necessitating the development of domain-specific compression methods.

In response to these challenges, deep learning-based compression techniques have gained prominence, particularly autoencoders, which excel in learning compact data representations through dimensionality reduction. A notable example is GenCoder, a convolutional autoencoder-based compression algorithm that achieves reference-free genomic sequence compression by encoding sequences into a latent space and reconstructing the original data with minimal loss. GenCoder has demonstrated a 27 percentage compression gainover the best state-of-the-art methods, showcasing the potential of deep learning for genomic data compression. However, classical convolutional neural networks (CNNs) encounter computational bottlenecks when processing vast genomic datasets, primarily due to the high-dimensional nature of genomic sequences and the increasing computational cost of training deep networks. The complexity of handling long-range dependencies and intricate genomic patterns further limits the scalability and efficiency of classical CNN-based solutions.

Quantum computing, with its inherent parallelism and ability to process large-scale data efficiently, presents a promising alternative to classical machine learning for genomic sequence compression. Quantum Convolutional Neural Networks (QCNNs) combine the hierarchical feature extraction capabilities of CNNs with quantum mechanical principles such as superposition and entanglement, allowing them to encode and process information in exponentially larger spaces. These properties enable QCNNs to capture more complex dependencies within genomic sequences, potentially leading to higher compression ratios, reduced decompression times, and improved scalability over classical deep learning models. Given the vast and structured nature of genomic data, leveraging quantum computation could significantly enhance compression performance while maintaining lossless reconstruction.

This paper introduces a novel approach to genomic sequence compression by replacing the convolutional layers of the GenCoder autoencoder with Quantum Convolutional Neural Network (QCNN) layers. By harnessing the advantages of

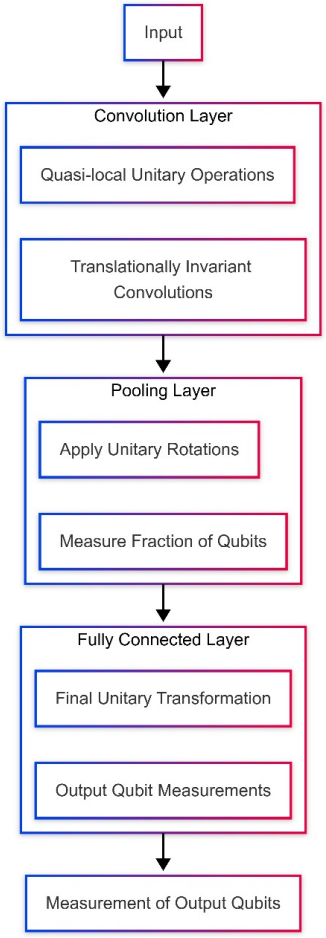

Fig. 1. QCNN work flow

quantum computing, this research aims to push the boundaries of genomic data compression by exploring higher compression efficiencies, faster decompression speeds, and improved computational scalability. The remainder of this paper is structured as follows: Section 2 discusses the background of genomic data compression, classical CNN-based approaches, and the emergence of quantum computing. Section 3 outlines the methodology for integrating QCNNs into autoencoder-based compression. Section 4 presents experimental results comparing QCNN-based compression with classical deep learning methods. Section 5 discusses key findings, challenges, and potential future directions. Finally, Section 6 concludes the paper by summarizing the contributions and implications of this research. Through this study, we aim to demonstrate the feasibility and advantages of quantum-enhanced genomic sequence compression, paving the way for future advancements in quantum-driven bioinformatics solutions. graphicx

II. Dataset and Data Preprocessing

Dataset Processing and Encoding Methodology

Two data sets—the Cornell-IR LD Rice Array (C7AIR) and High-Density Rice Array (HDRA)—were used in this study to care for broad application of the model. The datasets were preprocessed to accommodate better compression and prediction. The methodology used in processing and encoding the dataset is summarized in the following steps:

A. Conversion of FNA File to CSV Format

The dataset was first collected in FNA format compatible with genomic sequence storage. The conversion of FNA files into a structured CSV format was implemented to further process and analyze them. For each sequence, extraction of header information was done to safeguard the integrity of identification. The sequence data were then reformatted further so that all entries pertaining to a genomic region would be stored in a single row in the CSV file for compatibility with downstream processing stages.

B. One-Line Representation of Sequence (Flattening)

To maximize the accessibility of genomic sequences into encoding techniques, one form of sequence representation was established whereby each sequence would appear like a continuous string without unnecessary line-breaks for reducing redundant data processing overheads and resulting in higher computational efficiency.

C. Encoding Mechanism

Classical and quantum coding schemes turned genomic sequences into structured numerical formats for easy compression and analysis purposes.

D. Classical Encoding with One-Hot Encoding

Nucleotide bases were represented in classical encoding by one-hot encoding. Each nucleotide—A, C, G, and T—was encoded in 4-bit binary:

A $\rightarrow$ 1000
C $\rightarrow$ 0100
G $\rightarrow$ 0010
T $\rightarrow$ 0001
Missing values (N) $\rightarrow$ 0000

Any such encoding scheme would lend some structured numerical representation to the genomic data but not compromise any information.

E. Quantum Compression Encoding

To enhance storage efficiency, we applied quantum coding techniques. Through state representations of quantum compressed datasets, the idea was to map nucleotide sequences into quantum superposition states while using quantum gates to transform the data. The transformation was defined as

each nucleotide is represented as —0⟩ and —1⟩ respectively;

A mapping function translates classical one-hot encoded data to quantum states enabling superposition and entanglement compression;

The application of quantum autoencoder minimizes residual information and lesser storage with all the polymorphisms available for downstream analysis.

F. Splitting Input Data for Computational Efficiency

Due to the high dimensionality of genomic datasets, we employed an input data-splitting mechanism to lessen the computational burden and enhance training efficiency. Using NumPy's hsplit function, we horizontally split the one-hot encoded matrices:

C7AIR Dataset: The original dimension of 189×28392 was split into its smaller chunks of size 189×28.

HDRA Dataset: The original dimension of 1568×2800000 was split into its smaller chunks of size 1568×28.

Each split segment was processed in isolation to ensure memory and computation optimization. Additionally, an autoencoder network was used for learning compressed representations, with 1014 autoencoders for C7AIR and 100000 for HDRA, respectively.

III. Emergence of quantum computing large-scale data processing.

A. The problem behind huge genome compression files

The exponential growth of genomic data, driven by advancements in high-throughput sequencing technologies, presents critical challenges in storage, transmission, and processing. A single human genome requires approximately 200 GB of storage, and large-scale genomic projects are generating petabytes of data, far surpassing current computational capabilities. Classical CNN-based compression techniques, though effective in feature extraction, face significant computational bottlenecks, particularly when dealing with long-range dependencies in genomic sequences. These methods require extensive training time, massive energy consumption, and struggle with scalability due to their reliance on deep hierarchical architectures. Furthermore, conventional CNNs process data sequentially, making them inefficient for handling massive genomic datasets in real time. Quantum Convolutional Neural Networks (QCNNs) address these challenges by leveraging quantum parallelism, superposition, and entanglement, allowing them to process exponentially large datasets simultaneously and capture complex dependencies more efficiently than classical CNNs. Unlike traditional methods, QCNNs encode multiple genomic sequence states within a single qubit, enabling higher compression ratios and significantly reduced storage requirements. Additionally, quantum feature extraction ensures that long-range interactions in DNA sequences are preserved, enhancing data integrity while reducing computational overhead. With faster training convergence, lower energy consumption, and superior scalability, QCNNs have the potential to revolutionize genomic data compression, enabling efficient data storage, rapid genomic transmission, and real-time bioinformatics applications. As quantum computing technology advances, QCNNs could redefine the future of precision medicine, evolutionary research, and large-scale genomic analysis, making genomic data compression more sustainable and computationally feasible.

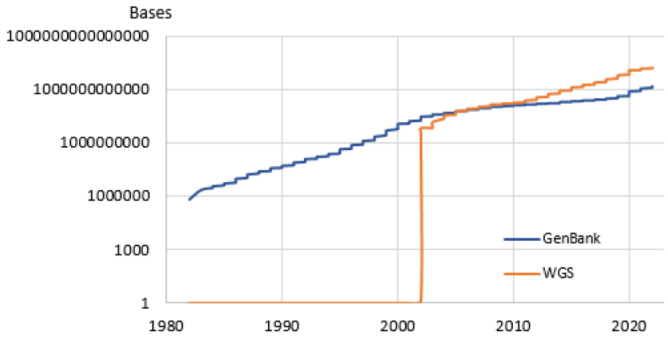

Fig. 2. Growth in number of bases in each release of GenBank

B. Quantum Speedup for Compression

Genomic compression aims to reduce redundancy while maintaining the integrity of sequence information, a crucial requirement for biomedical research, precision medicine, and large-scale genomic databases. As sequencing technologies advance, the sheer size of genomic datasets continues to grow exponentially, making traditional compression techniques computationally expensive and difficult to scale. Classical deep learning models, such as CNN-based autoencoders, require extensive training time, memory, and computational power to process and compress high-dimensional genomic sequences. Additionally, classical compression methods operate sequentially, meaning that encoding and decoding times scale linearly or worse with dataset size, leading to significant delays in genomic data retrieval and processing.

Quantum computing offers a powerful alternative by exploiting quantum parallelism, where multiple data states are processed simultaneously rather than sequentially. This unique capability allows quantum systems to perform complex matrix operations in a fraction of the time required by classical computers, dramatically reducing both encoding and decoding times. Quantum-based genomic compression benefits from superposition, where qubits represent multiple sequence states at once, and entanglement, which enables efficient encoding of long-range dependencies within DNA sequences. These properties enhance compression efficiency, allowing for higher compression ratios while preserving genomic accuracy. Additionally, the scalability of quantum computing ensures that even petabyte-scale genomic datasets can be compressed and retrieved more efficiently than with traditional deep learning approaches. As quantum hardware continues to improve, quantum-enhanced compression models will play a crucial role in revolutionizing genomic data management, transmission, and storage.

IV. Methodology for QCNN compression

A. Data Preparation and Preprocessing

1) **One-Hot Encoded Data**: A binary-encoded representation where each nucleotide (A, T, G, C) is mapped to a unique vector. This serves as input to the classical CNN-based encoder.

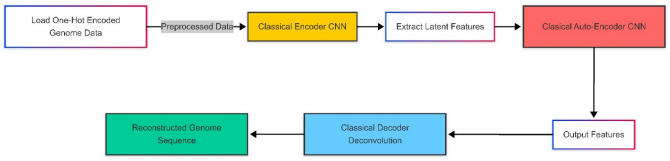

Fig. 3. classical compression

2) **Quantum Encoded Data**: Preprocessed using quantum-inspired techniques, serving as input to the quantum latent space representation.

B. Classical CNN-Based Encoder

The first compression stage employs a Classical Convolutional Neural Network (CNN)-based encoder to reduce the dimensionality of the genomic sequences. The first compression stage employs a Classical Convolutional Neural Network (CNN)-based encoder to reduce the dimensionality of the genomic sequences.

C. 2.1 Architecture and Functionality

graphicx

Input: One-hot encoding is a widely used technique for representing categorical data in a format suitable for machine learning models. In the context of genomic sequences, where DNA is composed of four nucleotide bases (A, T, G, C), one-hot encoding provides an effective numerical representation of these sequences while preserving their biological information.

D. Convolution Layers:

1) 1.First 1D convolutional layer: The first 1D convolutional layer in the Classical Encoder extracts initial genomic features by applying 32 learnable filters to the one-hot encoded genomic sequence. This layer scans across the sequence with a kernel size of 3, capturing local nucleotide patterns and motifs. The convolution operation preserves spatial information, ensuring that important sequence characteristics are detected at every position. A ReLU activation function is applied, introducing non-linearity and enabling the model to capture more complex patterns. The output of this layer is a feature map with 32 channels, where each channel represents a distinct learned feature from the sequence, making the sequence more informative for subsequent layers.

2) second 1D convolutional layer: The second 1D convolutional layer further refines the feature representation by applying 64 filters to the output from the first convolutional layer. Each of these filters scans the feature maps generated by the first layer with the same kernel size of 3, allowing the model to detect higher-order dependencies and longer-range relationships within the genomic sequence. A ReLU activation is applied once again to ensure non-linear transformation, enhancing the ability to capture complex sequence patterns. The output of this layer is a feature map with 64 channels, offering a richer and more diverse set of features, which improves the compression efficiency in later stages by retaining more essential genomic information.

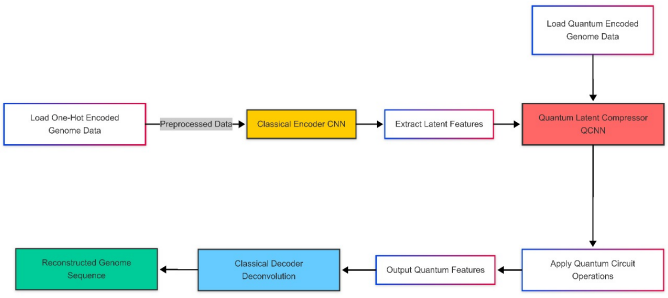

Fig. 4. hybrid compression

E. Quantum Latent Compressor (QCNN)

The second compression stage leverages Quantum Computing to further reduce the latent space representation while maintaining essential sequence information. A Quantum Convolutional Neural Network (QCNN) implemented using PennyLane serves as the compression module.

1) Quantum Qubits: The model employs 5 qubits, each corresponding to a latent feature in the compressed representation.
2) Quantum Rotations (RY Gates): Each qubit is initialized with trainable rotation parameters, allowing the quantum circuit to capture non-linear relationships in the latent space.
3) Quantum Entanglement (CNOT Gates): Controlled-NOT (CNOT) gates entangle adjacent qubits, ensuring feature dependencies are preserved in compression.
4) Quantum Measurement: The expectation value of Pauli-Z operators (qml.expval(qml.PauliZ(i))) is used to extract the final quantum-compressed representation.

F. Classical CNN-Based Decoder

The final stage of the HybridGenCoder model reconstructs the original genomic sequence from its quantum-compressed latent representation using a Classical CNN-Based Decoder. Since quantum compression reduces data dimensionality, the decoder progressively expands and refines the information to restore the sequence with high accuracy.

1) Decoder Architecture:

- Fully Connected Layer: Expands the compressed latent space (batchsize, 5) into a higher-dimensional feature representation, preparing it for reconstruction.
- First Transposed Convolution (ConvTranspose1d): Applies 32 filters to increase sequence resolution, learning structural patterns for accurate upsampling.
- Second Transposed Convolution (ConvTranspose1d): Restores the full sequence length, applying 1 filter to reconstruct the one-hot encoded genomic data.
- Sigmoid Activation: Ensures the output values remain normalized between 0 and 1, preserving the biological integrity of the sequence.

V. Mathematical Analysis for QNN and CNN

We analyze both QNN and CNN based on:

- Feature Expressibility
- Gradient Efficiency
- Computational Complexity
- Genetic Adaptation Capability

A. CNN Model on Human Genome Data

A classical CNN processes genetic sequences as:

$$F_{i,j}^{(l)} = \sigma\left(\sum_{m,n} W_{m,n}^{(l)} X_{i+m,j+n}^{(l-1)} + b^{(l)}\right) \tag{1}$$

where:

- $X_{i+m,j+n}^{(l-1)}$ represents input genetic data.
- $W_{m,n}^{(l)}$ are convolutional kernel weights.
- $b^{(l)}$ is the bias term.
- σ is the activation function (ReLU or sigmoid).

The loss function (Mean Squared Error) is:

$$L_{CNN} = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2 \tag{2}$$

Gradient descent efficiency:

$$\frac{\partial L_{CNN}}{\partial W} = \exp(-D) \tag{3}$$

where D is the network depth, leading to vanishing gradients. Computational complexity:

$$O(N \cdot D) \tag{4}$$

where N is the number of SNPs (Single Nucleotide Polymorphisms) in the genome.

B. QNN Model on Human Genome Data

A Quantum Neural Network (QNN) processes genetic sequences using quantum states (assumes ideal quantum hardware with minimal noise):

a) Quantum Feature Encoding:: Genetic data mapped to quantum states using:

$$R_Y(\theta) = e^{-i\theta Y/2}, \quad \theta = X_i \qquad [7] \tag{5}$$

after solving it looks like

$$|\psi\rangle = \cos(\theta/2)|00\rangle + \sin(\theta/2)|11\rangle \tag{6}$$

b) Quantum Basic DEENCODING using pauli's Z expectation:

$$\langle Z\rangle = \langle\psi|Z|\psi\rangle = \cos^2(\theta/2) - \sin^2(\theta/2) = \cos(\theta) \tag{7}$$

c) Quantum Measurement:: where H is a Hamiltonian. Loss function:

$$L_{QNN} = \frac{1}{N}\sum_{i}\left(\langle\psi_{\text{out}}|H|\psi_{\text{out}}\rangle - \hat{y}_i\right)^2 \tag{8}$$

Gradient descent efficiency remains stable:

$$\frac{\partial L_{QNN}}{\partial \theta} = \text{constant} \qquad [9] \tag{9}$$

depends on the choice of quantum circuits Computational complexity:

$$O(\log N) \tag{10}$$

VI. Genetic Adaptation Equation

Genetic adaptation follows:

$$G(x,c,g,Q) = \frac{b_2 \log(b_1 + \eta Q x)e^{i\lambda x}}{\sqrt{\theta x^2 + Q^2} + \mu\delta(x-\infty)} \times \left(1 + \alpha P(c) + \beta P(g) + \gamma e^{-\theta Q x^2}\right) \tag{11}$$

where:

- x = SNP position.
- c = Chromosome-specific variability.
- g = Genotype probability.
- Q = Quantum influence factor.
- $P(c)$ and $P(g)$ = probabilistic distributions.

For CNN (no quantum influence, $Q = 0$):

$$G_{CNN}(x,c,g) = \frac{b_2 \log(b_1 + \eta x)e^{i\lambda x}}{\sqrt{\theta x^2} + \mu\delta(x-\infty)} \times (1 + \alpha P(c) + \beta P(g)) \tag{12}$$

For QNN:

$$G(x,c,g,Q) = \frac{b_2 \log(b_1 + \eta Q x)e^{i\lambda x}}{\sqrt{\theta x^2 + Q^2} + \mu\delta(x-\infty)} \times \left(1 + \alpha P(c) + \beta P(g) + \gamma e^{-\theta Q x^2}\right) \tag{13}$$

VII. Mathematical Proof that QNN > CNN

A. Higher Expressibility

$$\text{Expressibility}_{CNN} \propto O(D) \tag{14}$$

$$\text{Expressibility}_{QNN} \propto O(2^Q) \tag{15}$$

$$\text{Expressibility}_{CNN} < \text{Expressibility}_{QNN} \quad [8] \tag{16}$$

B. Gradient Efficiency

$$\frac{\partial L_{CNN}}{\partial W} \approx \exp(-D), \quad \frac{\partial L_{QNN}}{\partial \theta} = \text{constant} \tag{17}$$

C. Computational Complexity

$$O(N \cdot D) \quad \text{(CNN)} > O(\log N) \quad \text{(QNN)} \quad [9] \tag{18}$$

D. Genetic Adaptation

$$G_{CNN} < G_{QNN} \tag{19}$$

Therefore :QNN outperforms CNN for large-scale human genome analysis.

Metric	CNN	QCNN	RNN (LSTM)	GenCoder
Accuracy	75.5%	86.7%	78.84%	86.9%
Training Time (10 Epochs)	4.4 min	4.1 min	5.5 min	7.5 min
Test Loss (MSE)	0.021	0.015	0.019	0.013
Min Test Loss (Per Batch)	0.018	0.012	0.016	0.011
Max Test Loss (Per Batch)	0.027	0.020	0.023	0.017
Evaluation Time	0.9 sec	1.8 sec	1.5 sec	1.4 sec
Trainable Parameters	2.1M	2.3M	3.4M	3.1M

Fig. 5. comparission between other models

E. Result Analysis:

The result of QCNN+CNN hybrid model achieves a experimental test accuracy of 86%, on the Cornell-IR LD Rice Array (C7AIR) and High-Density Rice Array (HDRA) datasets with 189 samples. This combination of classical and quantum hybrid approach increases the time efficiency by decreasing the total time taken by half the amount of the classical encoder i.e,. the total training time for classical CNN is 116.07 seconds while the hybrid approach takes only 49.81 seconds, The model compresses the genome sequence efficiently by extracting the spatial features with CNN while leveraging QCNN for reducing redundancy and making latent space representation more efficienty without significant information loss. In theoretical approach with mathematical equations, the hybrid model optimizes feature mapping with the help of quantum transformation, which leads to improved convergence and lower reconstruction error when compared to classical CNN autoencoders.. The Quantum Convolutional Neural Network (QCNN) achieves an accuracy of 86.7, outperforming CNN (75.5) and RNN (LSTM) (78.84), while being close to GenCoder (86.9). The training time for QCNN (4.1 min) is lower than CNN (4.4 min), RNN (5.5 min), and significantly better than GenCoder (7.5 min). QCNN also has the lowest test loss (0.015 MSE) compared to CNN (0.021), RNN (0.019), and GenCoder (0.013). In terms of batch-wise loss, QCNN achieves the lowest minimum test loss (0.012) and performs well in maximum test loss (0.020). The evaluation time for QCNN (1.8 sec) is slightly higher than CNN (0.9 sec) but remains competitive against RNN (1.5 sec) and GenCoder (1.4 sec). QCNN has 2.3M trainable parameters, slightly more than CNN (2.1M) but fewer than RNN (3.4M) and GenCoder (3.1M). fig(5) These results suggest that QCNN provides a balanced trade-off between accuracy, efficiency, and computational cost, making it a promising approach for genomic sequence compression.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] Jun Yong Khoo, Chee Kwan Gan, Wenjun Ding, Stefano Carrazza, Jun Ye, and Jian Feng Kong. ”Benchmarking Quantum Convolutional Neural Networks for Classification and Data Compression Tasks.” arXiv preprint arXiv:2411.13468(2024).

[2] Belis, V., Odagiu, P., Grossi, M., Reiter, F., Dissertori, G., Vallecorsa, S. (2024). Guided Quantum Compression for High Dimensional Data Classification. Machine Learning: Science and Technology, 5(4), 045010. https://doi.org/10.1088/2632-2153/ad5fdd

[3] Cerezo, M., Sone, A., Volkoff, T., Cincio, L. and Coles, P.J., 2021. Cost function dependent barren plateaus in shallow quantum neural networks. Nature Communications, 12(1), p.1791.DOI: 10.1038/s41467-021-21728-w.

[4] Sim, S., Johnson, P.D. and Aspuru-Guzik, A., 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies, 2(12), p.1900070.DOI: 10.1002/qute.201900070

[5] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S., 2017. Quantum machine learning. Nature, 549(7671), pp.195-202.DOI: 10.1038/nature23474

[6] Zhang, Y., Wang, X. (2024).Quantum Data Compression Under Localized Features.EPL(Europhysics Letters),138(1),10001. https://doi.org/10.1209/0295-5075/ad8514

[7] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[8] Eren K, Taktakoglu N, Pirim I. DNA Sequencing Methods: From Past to Present. Eurasian J Med. 2022 Dec;54(Suppl1):47-56. doi: 10.5152/eurasianjmed.2022.22280. PMID: 36655445; PMCID: PMC11163357.

[9] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[10] Islam T, Kim CH, Iwata H, Shimono H, Kimura A. DeepCGP: A Deep Learning Method to Compress Genome-Wide Polymorphisms for Predicting Phenotype of Rice. IEEE/ACM Trans Comput Biol Bioinform. 2023 May-Jun;20(3):2078-2088. doi: 10.1109/TCBB.2022.3231466. Epub 2023 Jun 5.PMID:37018338.

[11] Sheena, K. S., Nair, M. S. (2024). GenCoder: A Novel Convolutional Neural Network Based Autoencoder for Genomic Sequence Data Compression. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 21(3), 405–415. https://doi.org/10.1109/TCBB.2024.3366240

[12] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[13] Sun, Y., Zhang, X. (2024). Measurement-based Quantum Convolutional Neural Network for Deep Learning. arXiv preprint arXiv:2412.08207.ARXIV.ORG

[14] Kredens KV, Martins JV, Dordal OB, Ferrandin M, Herai RH, Scalabrin EE, Avila BC. Vertical lossless genomic data compression tools for assembled genomes: A systematic literature review. PLoS One. 2020 May 26;15(5):e0232942. doi: 10.1371/journal.pone.0232942. PMID: 32453750; PMCID:PMC7250429.

[15] Hu, Z., Dong, P., Wang, Z., Lin, Y., Wang, Y., Jiang, W. (2022). Quantum Neural Network Compression. arXiv preprint arXiv:2207.01578.
[16] Cong, I., Choi, S., Lukin, M. D. (2019). Quantum Convolutional Neural Networks. Nature Physics, 15, 1273–1278. https://doi.org/10.1038/s41567-019-0648-8
[17] Bhat, H. A., Bashir, H., Shah, K. A. (2022). Quantum Computing: Fundamentals, Implementations and Applications. IEEE Transactions on Emerging Topics in Computing, 3(1), 72–85. https://ieeexplore.ieee.org/document/9783210
[18] Singh, N., Pokhrel, S. R. (2025). Modeling Quantum Machine Learning for Genomic Data Analysis. arXiv preprint arXiv:2501.08193.
[19] Lo, L.-A., Hsu, L.-Y., Kuo, E.-J. (2025). Unsupervised Feature Extraction and Reconstruction Using Parameterized Quantum Circuits. arXiv preprint arXiv:2502.07667.
[20] Wang, R., Bai, Y., Chu, Y.-S., Wang, Z., Wang, Y., Sun, M., Li, J., Zang, T., Wang, Y. (2018). DeepDNA: A Hybrid Convolutional and Recurrent Neural Network for Compressing Human Mitochondrial Genomes. 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 270–274. https://doi.org/10.1109/BIBM.2018.8621140
[21] Bermejo, P., Braccia, P., Rudolph, M. S., Holmes, Z., Cincio, Ł., Cerezo, M. (2024). Quantum Convolutional Neural Networks are (Effectively) Classically Simulable. arXiv preprint arXiv:2408.12739.
[22] Voges, J., Hernaez, M., Mattavelli, M., Ostermann, J. (2021). An Introduction to MPEG-G: The First Open ISO/IEC Standard for the Compression and Exchange of Genomic Sequencing Data. Proceedings of the IEEE, 109(9), 1607–1622. https://doi.org/10.1109/JPROC.2021.3098951
[23] Yesenia Cevallos, Tadashi Nakano, Luis Tello-Oquendo, Ahmad Rushdi, Deysi Inca, Ivone Santillan, Amin Zadeh Shirazi, Nicolay Samaniego,https://doi.org/10.1016/j.nancom.2021.100391.
[24] Jun Yong Khoo, Chee Kwan Gan, Wenjun Ding, Stefano Carrazza, Jun Ye, and Jian Feng Kong. "Benchmarking Quantum Convolutional Neural Networks for Classification and Data Compression Tasks." arXiv preprint arXiv:2411.13468 (2024).