QeHDC: Hyperdimensional Computing based on Quantum-enhanced binding and SuperClass Construction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Hyperdimensional Computing (HDC) is a robust computational framework inspired by human cognition characterized by simple and efficient operations within high-dimensional vector spaces. Quantum-enhanced Hyperdimensional Computing (QeHDC) extends classical HDC by leveraging quantum mechanical properties to enhance computational efficiency. In this paper, we propose a novel Quantum HDC framework featuring a one-pass training method, leveraging sinusoidal and quantum encoding to project classical data into quantum amplitude states efficiently. Our framework introduces an innovative reference-state-based quantum binding operation realized via quantum circuits. Furthermore, we propose a density-matrix-based superclass generation strategy employing eigenvalue decomposition to extract critical quantum state features effectively, enabling a more accurate and robust class representation. Experimental evaluations conducted on standard benchmark datasets demonstrate our approach's superior performance, robustness to noise, and computational feasibility compared to traditional classical and existing quantum-enhanced approaches. The results highlight the practical benefits and potential of Quantum HDC for quantum-enhanced classification tasks and pave the way for future advancements in quantum-inspired computational paradigms.

1 Introduction

Hyperdimensional Computing [15], also known as Vector Symbolic Architectures [8], is a computational paradigm that leverages high-dimensional (typically larger than 10,000 dimensions) vectors for information representation, processing, and learning. Inspired by cognitive science and neuroscience principles, HDC aims to mimic how the human brain encodes, stores, and retrieves information by operating in high-dimensional space. Instead of using traditional low-dimensional numerical representations, HDC uses high-dimensional vectors. Each vector represents a concept in a distributed way.

Information is processed through algebraic operations. These include bundling (element-wise addition with normalization), binding (component-wise multiplication or XOR), and permutation (reordering of dimensions). These operations allow HDC to efficiently perform classification, pattern recognition, associative memory, and temporal sequence modeling tasks. One of the key advantages of HDC is its robustness to noise, as high-dimensional representations inherently provide redundancy and fault tolerance, making it well suited for real-world applications with noisy data. In addition, HDC is computationally efficient because it relies primarily on vector arithmetic rather than complex matrix operations, making it well suited for low-power, real-time, and edge computing scenarios such as embedded systems, IoT devices, and neuromorphic computing platforms. Unlike traditional deep learning methods that require extensive training using backpropagation and gradient descent, HDC

can learn through a fast, one-time encoding and updating mechanism, which makes it particularly effective for applications with limited training data.

Although HDC has demonstrated substantial advantages, it still faces serious computing and storage bottlenecks with the increase in data complexity and scale. Therefore, using quantum enhancement to break through the traditional architecture and mechanism of HDC is the motivation for this study.

QeHDC breaks through traditional architectural limitations via three core mechanisms. **Dimensional Compression:** Quantum states utilize the geometric properties of the Bloch sphere to represent traditional vector space. By using only log_2N qubits, they achieve the same representation capability as classical high-dimension spaces while enabling exponential dimensional compression. **Computational Architecture:** Quantum Circuit replaces traditional $O(N^2)$ complexity matrix operations with quantum gate operations based on unitary evolution. Through superposition-based parallel processing and entanglement gates, these circuits facilitate efficient feature interactions. **Representation transformation:** Quantum system overcomes the constraints of classical linear algebra by constructing nonlinear entangled states using quantum circuits. By leveraging quantum interference effects, they achieve superlinear coupling between features, enhancing the interaction dynamics beyond classical approaches.

Based on the above challenges and motivations, our proposed Quantum-enhanced hyperdimensional computing mainly achieves the following contributions.

- First, we introduce a **two-step hybrid encoder**. It initially maps the original data features into a high-dimensional space using random projection and trigonometric transformations. Then, it expresses the high-dimensional vectors in Bloch space using only a few qubits. This implicitly reduces the dimension of data representation. The proposed encoding method lowers the computational resources overhead required for data processing while making data encoding more suitable for quantum computing implementation.

- Second, we develop an innovative **Quantum Binding** operation called **Referencec-State-Based Quanutm Binding**. Traditional HDC binding operations often rely on simple XOR, circular convolution, or Hadamard product. In contrast, our quantum binding and bundling approaches leverage a predefined quantum reference estate as a binding anchor. By designing specialized quantum circuits, such as the controlled rotation gate (CRX) and the entanglement gate (CX), our method achieves a more efficient data association and representation. This reference-state-based binding not only enhances the expressiveness of high-dimensional vectors but also provides a natural operation mode that is well-suited for quantum hardware implementation.

- Third, we propose a Density-Matrix-Based template state construction method, known as **Density-Matrix-Based Superclass Generation**. This method decomposes the density matrix of quantum states and selects key feature vectors to construct a representative template state (SuperClass) for each class. By capturing core features from the data more accurately, this approach improves the model's robustness and classification accuracy.

- Additionally, we built a complete classification experimental verification system based on IBM quantum computing framework Qiskit. First, we complete the classification experimental modeling in high-fidelity quantum noise environment in Qiskit Aer Simulator. We then carry out further confirmatory experiments on a small sample of physical qubits (QPUs) on IBM processors. It is worth noting that despite the essential differences in the calculation principle and noise environment, we still get the corresponding results. This fully proves that this scheme has a unique cross-platform adaptability, which can not only validate and optimize the algorithm in the existing classical computing system but also seamlessly connect with the quantum computing unit in the NISQ era to realize prototype verification.

Through these innovations, our QeHDC approach overcomes the efficiency bottleneck of traditional HDC when processing large and complex data. It provides a new idea and a feasible solution for the practical application of quantum computing in machine learning.

The remaining paper is organized as follows. Section 2 presents the background, including HDC-related work and quantum computing basics. Section 3 introduces the overview framework of our proposed method and detailed designs of each component. Section 4 shows the experiments set-up and data in detail. Section 5 provides the results on classification tasks and the comprehensive

evaluation of our approach. The final section 6 concludes the paper and envisions the direction of future work.

2 Backgroud

2.1 Hyperdimensional Computing

HDC is originally conceived in the 1990s as cognitive models that depend on very high dimensionality and randomness [15]. It is inspired by the way the human brain represents and processes information. HDC has the advantages of robustness, fault tolerance, and efficient parallel computing by using high dimensional random vectors to represent and process information [3]. In recent years, with the rapid development of artificial intelligence and computing hardware, HDC has regained widespread attention from researchers. In 2017, Rahimi et. al. proposed HDC for machine learning tasks, demonstrating its high efficiency and low complexity in simple text classification [22]. Subsequently, Hersche et al. evaluated a variety of HDC coding strategies in the motor imagination classification of EEG, demonstrating high energy efficiency on embedded devices[25].

In recent years, many innovative HDC methods have emerged. AdaptHD [12] can dynamically adapt to data changes and effectively deal with non-stationary data environments. The real-time incremental learning framework built by OnlineHD [10] can support continuous updating of streaming data tasks. In terms of representation optimization, CompHD [21] adopts a composite coding strategy to enhance feature interaction, LeHDC [6] optimizes high-dimensional space mapping through the learnable embedding mechanism, and NeuraldHD [30] deeply integrates a neural network architecture to achieve joint representation optimization. In the field of computational efficiency optimization, the dynamic coding technology designed by DistHD [28] not only improves accuracy but also strengthens noise immunity. QuantHD [11] adopts low bandwidth representation to significantly reduce the storage overhead on the premise of ensuring performance, while SparseHD [13] reduces the computational complexity based on sparse coding theory. For specific application scenarios, BioHD [29] developed a biometric coding protocol to enhance the sensitivity of genome analysis. The graph structure coding operator constructed by GraphHD [23] highly improved the accuracy of node classification and SecureHD [4] ensured the privacy of medical data processing through the homomorphic encryption protocol. In terms of application, HDC has been widely used in a variety of complex tasks in recent years, including biological signal analysis [7, 24], natural language processing [16], robot perception and navigation [19, 17], and remote sensing image processing [20].

In summary, HDC is gradually becoming an important research direction in the field of artificial intelligence due to its computational simplicity, robustness, and compatibility with emerging computing architectures, especially in the context of quantum computing convergence, which has a broader prospect.

2.2 Quantum Computing Basics

Quantum computing is a computational model based on the principles of quantum mechanics, which uses qubits as the basic computing unit. Unlike classical bits, which exist strictly in either 0 or 1, qubits can both be in state $|0\rangle$ and $|1\rangle$. Qubits can exist in a **superposition**[22] of both states, represented mathematically as:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \text{, where } \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1 \tag{1}$$

α and β are the amplitudes of the complete complex probabilities, indicating the probabilities of obtaining $|0\rangle$ and $|1\rangle$, respectively, when measuring. Due to the existence of superposition states, quantum computing can simultaneously process multiple computation paths in the most exponential level of state space, thus increasing computing power.

Non-local correlations can be formed between multiple qubits through entanglement, where the measurement of one qubit instantaneously affects the state of the other one, even if they are far apart. For example, the **Bell State**[1] of two qubits is one of the most basic entangled states:

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2}$$

3

Entanglement is particularly important in high-dimensional computing because it allows data to form more complex relationships between multi-qubit systems, improving the ability to represent information. Quantum computing manipulates qubits through a series of quantum gates, which are unitary transformations that guarantee the reversibility of quantum information. Typical quantum gates include the Hadamard gate (H), Pauli-X (bit-flip) gate, rotation gates (RX, RY, RZ), and controlled gates such as controlled-NOT (CNOT or CX gate), which flip the state of a target qubit conditioned on the state of a control qubit. For example, the widely used H gate generates superposition states from computational basis states:

$$H\left|0\right\rangle = \frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}, \quad H\left|1\right\rangle = \frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}} \tag{3}$$

Moreover, practical quantum systems typically cannot be described completely by pure states due to noise and environmental interaction. Thus, introduce the concept of **density matrices**. A density matrix[27] is a generalized mathematical tool used to represent mixed states (statistical ensembles) of a quantum system:

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|, \quad \sum_i p_i = 1 \tag{4}$$

where p_i represents the probability that the quantum system exists in the state ψ_i. Density matrices are crucial for quantum computing tasks involving statistical modeling and noisy environments, providing a powerful way to encode complex statistical information beyond pure-state representation. In this paper, the density matrix method is used to construct the Superclass vector, and the core information of the class is extracted by feature decomposition to improve the classification model's generalization ability. Finally, information encoded in quantum states is obtained through the operation of **measurement**. Measurement collapses the quantum superposition into one of the classical basis states with probabilities determined by the amplitude squared of each basis state. Specifically, measuring a qubit $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ yields outcomes with probabilities:

$$P(|0\rangle) = |\alpha|^2, \quad P(|1\rangle) = |\beta|^2 \tag{5}$$

In our study, measurement is used in the classification decision process of quantum high-dimensional computation. The data features of hyperdimensional representation are extracted by measurement, and the final pattern recognition task is performed using the classical computational framework. In summary, quantum computing provides the most new information representation and computation framework; through the superposition of quantum states, entanglement, statistical representation of density matrices, operation of quantum circuits, and measurement mechanisms, it can enhance the feature representation of hyper-dimensional computing. Based on these quantum properties, a novel Quantum-enhanced Hyperdimensional Computing (QeHDC) method is proposed to encode, bind, and classify high-dimensional data more efficiently.

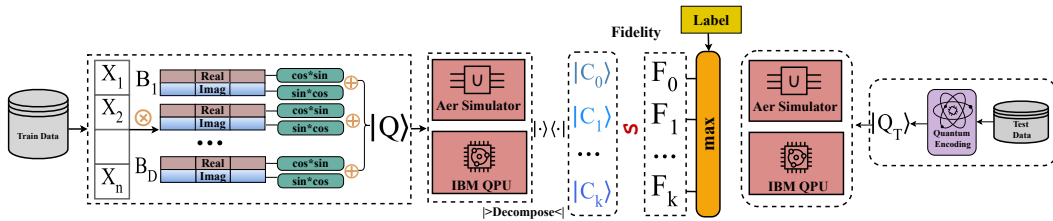## 3 Quantum-enhanced Hyperdimensional Computing

Figure 1: Method

In this paper, we propose a Quantum-enhanced Hyperdimensional Computing (QeHDC) system that leverages quantum computational properties to enhance the efficiency and expressiveness of classical HDC. The system comprises four core modules. First, a 2-step classical-quantum hybrid encoder transforms classical data into quantum states. Then, a Quantum HD Operator performs HDC operations like binding and construction using quantum circuits. Superclass construction builds a density matrix over bound states and extracts features via eigen-decomposition. Finally, classification

4

is achieved by comparing input states with learned superclasses using quantum state fidelity. We will introduce each of these parts in detail below. Before that, we define a system input sample as:

$$D = (X_i, y_i)_{i=1}^{N} \tag{6}$$

where $X_i \in \mathbb{R}^d$ is the feature vector of dimension d and y_i is the class label, typically encoded as a categorical variable.

3.1 2-step Hybrid Encoder

The core of the coding process is to efficiently map classical eigenvectors to quantum states while preserving the structural information of the input data. The first step is a classical transformation. Given a classical input vector $X \in \mathbb{R}^d$, we perform a random projection using a randomly initialized weight matrix $W \in \mathbb{R}^{n*d}$, where $n = 2^{N_{\mathbf{qubits}}}$ represents the target quantum dimension. The transformation is computed as:

$$Z = WX \tag{7}$$

To introduce nonlinearity and phase variation in the coding process, we introduce a bias term $B \in \mathbb{R}^n$ whose values are uniformly sampled from the interval $[0, 2\pi]$ forming a transformed projection:

$$Z' = Z + B \tag{8}$$

To convert the projected values into **quantum-compatible** complex-valued presentations, we Then, the real part and imaginary part are mapped separately using the sinusoidal transformations:

$$\Phi_{\mathrm{re}} = cos(Z')sin(Z), \quad \Phi_{\mathrm{im}} = sin(Z')cos(Z) \tag{9}$$

This formulation which we called **Cross-Multiplicative** achieves the same uniform phase angle distribution while using only single weight matrix, making it the most efficient approach (See Appendix A for explanation) The final quantum state representation is given by:

$$\Psi = \frac{\Phi_{\mathrm{re}} + i\Phi_{\mathrm{im}}}{||\Phi_{\mathrm{re}} + i\Phi_{\mathrm{im}}||} \tag{10}$$

Compared to other encoding schemes, this approach ensures a well-distributed representation in the complex space, enhancing separability and stability in quantum computation. Further theoretical analysis, including its impact on data distribution, is provided in Appendix A.

3.2 QeHDC Operation

Quantum operations form the core of our method by enabling transformations and interactions between encoded quantum states. Once classical data is converted into quantum representations via Cross-Multiplicative Encoding, a set of quantum operations is applied to enhance, bind, and extract relevant information. The goal of these operations is to exploit quantum properties such as superposition and entanglement to improve computational efficiency and feature representation. We focus on the **binding operation**, which is critical in representing relationships between encoded quantum states. Unlike classical models that rely on matrix multiplications or convolutional operations, our approach leverages quantum state interactions to achieve a compact and efficient representation of information.

In the Quantum hyperdimensional system, direct implementation of classical HDC binding operation causes an exponential increase in the required qubits (See Appendix B). To address this, we propose a specifically compressed binding method, employing a quantum circuit for efficient information binding and compression. Consider the encoded quantum state for a single input sample as :

$$|\Phi_{\mathrm{in}}\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle, \quad \sum_i |\alpha_i|^2 = 1 \tag{11}$$

Also define a predefined reference quantum state as:

$$|\Phi_{\mathrm{ref}}\rangle = \sum_{j=0}^{2^N-1} \beta_j |j\rangle, \quad \sum_j |\beta_j|^2 = 1 \tag{12}$$

We initialize these two states into two separate quantum registers, each containing n qubits, resulting in the initial state:

$$|\Psi_{\text{init}}\rangle = |\Phi_{\text{in}}\rangle \otimes |\Phi_{\text{ref}}\rangle \tag{13}$$

Next, we apply Controlled-RX (CRX) gates between each pair of corresponding qubits (q_i, q_{i+n}) with rotation angle $\pi/4$:

$$CRX(\pi/4) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-i\frac{\pi}{8}X} \tag{14}$$

At this point, the state evolves into:

$$|\Psi'\rangle = \prod_{i=0}^{n-1} CRX(\pi/4)|\Psi_{\text{init}}\rangle \tag{15}$$

Subsequently, apply single-qubit RY rotations on each qubit in the reference register for additional nonlinear compression:

$$RY(\pi/4) = e^{-i\frac{\pi}{8}Y} \tag{16}$$

The state then becomes:

$$|\Psi''\rangle = \prod_{i=0}^{n-1} RY_i(\pi/4)|\Psi'\rangle \tag{17}$$

To further enhance the binding and compression effect, we add chain entanglement operations (CX gates) within the reference register. The final highly entangled and compressed quantum representation is:

$$|\Psi_{\text{comp}}\rangle = \prod_{\{q_a,q_b\}\in Q} CX_{q_a,q_b}|\Psi''\rangle \tag{18}$$

Where $Q = \{n, n+1, ..., 2n-1\}$. For all samples within the same class c (assuming a total of m samples), we perform the above binding and compression process individually, obtaining:

$$\{|\Psi_{\text{comp}}^{(1)}\rangle, |\Psi_{\text{comp}}^{(2)}\rangle, ..., |\Psi_{\text{comp}}^{(m)}\rangle\} \tag{19}$$

The binding circuit described above is a fully entangled structure, meaning any two qubits in the latter half are directly entangled. The circuit for the example with 4 qubits is shown in Fig. 3.
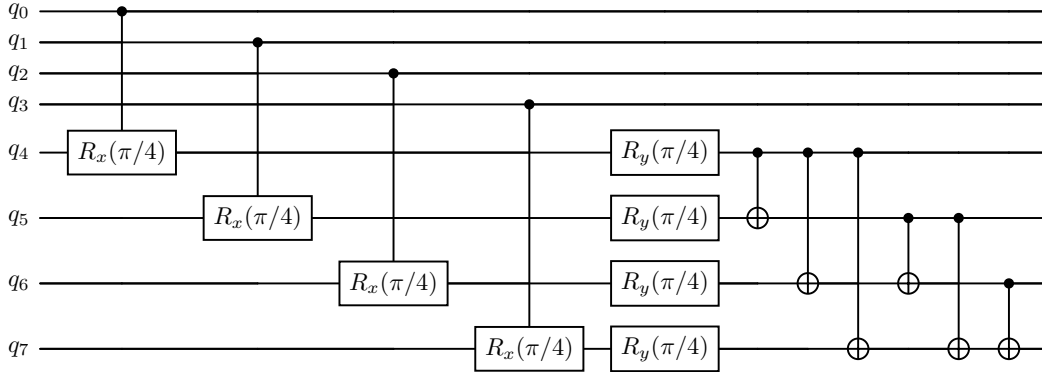

Figure 2: Quantum Bind Circuit

We also employed some structures with lower complexity. Please refer to Appendix C. Through these detailed steps, we achieve quantum state binding and compression at both sample and class level, significantly avoiding the exponential qubit resource increase caused by the traditional classical HDC method. This approach holds substantial potential and expansion possibilities in future practical applications. In our proposed compressed binding method, we implement quantum state binding compression using specifically designed quantum circuits. In our practical study, we employ two approaches for realizing and verifying our quantum bound states:

- Ideal Quantum State simulation via the Qiskit Aer Simulator (no measurement required)

- Physical experiments on IBM's Quantum Processing Unit (QPU), requiring measurement and quantum state tomography for reconstruction.

6

It should be noted that using the ideal state vector from the Aer simulator applies only to theoretical analysis and algorithm validation. On real quantum devices (e.g., IBM QPU), measurement and quantum state tomography reconstruction are essential steps. Our research encompasses both ideal simulations and real-device experiments to comprehensively demonstrate both the theoretical performance and the practical feasibility of our quantum binding algorithm.

3.3 SuperClass Template Generate

In our method, using the previously introduced compressed binding method, each class contains numerous samples represented by bound quantum states. Therefore, from multiple sample states with the same class, we must extract a representative and stable quantum state, termed the **SuperClass Template State**. Suppose a class c has m data samples, after obtaining a holistic representation for the class, we need to comprehensively capture the statistical characteristics of these sample states. The quantum density matrix provides the ideal mathematical tool for this purpose:

$$\rho_c = \frac{1}{m} \sum_{i=1}^{m} |\Psi_{\text{comp}}^i\rangle \langle \Psi_{\text{comp}}^i| \tag{20}$$

The density matrix ρ_c is essentially a statistical representation of all bound states in class c. It reflects information from all states and reveals intrinsic correlations among them. The density matrix has dimensions $2^{2n} \times 2^{2n}$. To address the computational challenge due to dimensionality, we employ two strategies based on matrix dimension. Directly construct the density matrix using standard methods and perform eigen-decomposition:

$$\rho_c |\phi_j\rangle = \lambda |\phi_j\rangle \tag{21}$$

We choose the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ as the representative state for the class:

$$|\Phi_c^{\text{template}}\rangle = |\phi_{\max}\rangle, \quad \rho_c |\phi_{\max}\rangle = \lambda |\phi_{\max}\rangle \tag{22}$$

Through the described SuperClass template state generation approach, we effectively extract representative and stable class-level quantum representations from multiple bound states.

3.4 Single-Pass Training

In our method, we use one-pass training. Thanks to the characteristics of the high-dimensional computing framework, the model can be constructed with only one data traversal, avoiding the computational cost and parameter tuning complexity caused by multiple iterative optimizations in traditional machine learning methods. This approach not only significantly reduces the occupation of quantum resources, but is more suitable for the current environment with limited quantum hardware resources, and effectively improves the actual deployment capability and generalization performance of the algorithm.

3.5 Classification

In the classification phase, for each unseen classical data sample X_{test}, we first encode it into a quantum state $|\Phi_{X_{\text{test}}}\rangle$ using the Cross-Multiplicative Encoding approach described previously. Subsequently, the encoded state is bound to the same predefined reference quantum state $|\Phi_{\text{ref}}\rangle$ via the compressed Binding process, obtaining a quantum bound state:

$$|\Psi_{\text{test}}\rangle = \mathcal{U}_{\text{bind}}\left(|\Phi_{x_{\text{test}}}\rangle, |\Phi_{\text{ref}}\rangle\right).$$

Classification is then performed by calculating the quantum state fidelity between the bound test state $|\Psi_{\text{test}}\rangle$ and each pre-computed SuperClass template state $|\Psi_{c_j}^{(\text{template})}\rangle$:

$$F\left(|\Psi_{\text{test}}\rangle, |\Psi_{c_j}^{(\text{template})}\rangle\right) = \left|\langle \Psi_{\text{test}}|\Psi_{c_j}^{(\text{template})}\rangle\right|^2.$$

The class prediction for the test sample is obtained by selecting the SuperClass template that yields the highest fidelity value:

$$c^* = \arg\max_{c_j} F\left(|\Psi_{\text{test}}\rangle, |\Psi_{c_j}^{(\text{template})}\rangle\right).$$

On ideal quantum simulators (e.g., Qiskit Aer), the fidelity is directly computed using state vectors, whereas on real quantum hardware (e.g., IBM QPUs), quantum state tomography is first performed to reconstruct the density matrix ρ_{test}. Subsequently, fidelity is calculated as:

$$F(\rho_{\text{test}}, |\Psi_{C_j}^{(\text{template})}\rangle) = \langle \Psi_{C_j}^{(\text{template})} | \rho_{\text{test}} | \Psi_{C_j}^{(\text{template})} \rangle.$$

This quantum fidelity-based classification approach naturally leverages quantum coherence and quantum state similarity, providing robust, noise-resistant classification performance that is particularly suitable for resource-constrained quantum computing environments.

4   Experiments

In this section, we evaluate the performance and effectiveness of our proposed QeHDC framework. We provide details on the datasets, quantum simulation, and hardware environments, as well as the parameter settings shared across all experiments.

4.1   Dataset

We evaluate our method on three widely used datasets: **ISOLET**[5], **MNIST**[18], and **UCI HAR**[2]. The quantitative details of the datasets are shown in Table 1.

Table 1: Quantitative details of the datasets

Dataset	# Features	# Classes	# Traing Samples	# Test Samples
MNIST	784	10	60000	10000
ISOLET	617	26	6238	1559
UCI HAR	561	6	7352	2947

The ISOLET dataset consists of spoken letter recordings, where each sample corresponds to one of the 26 English alphabet letters spoken by different speakers. Each instance is represented by 617 acoustic features. Due to its high-dimensional input and large class space, ISOLET presents a challenging scenario for scalable classification. The MNIST dataset is a benchmark image classification task

Table 2: Binary Classification Results (2 Classes) for Three Datasets by Method and Dimension

Method	ISOLET-2Classes			MNIST-2Classes			UCI HAR-2Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	72.78±8.12	76.11±5.11	76.39±4.37	78.50±8.90	90.12±2.04	94.47±0.92	60.15±1.64	67.63±2.12	70.70±2.79
AdaptHD	67.50±13.96	68.33±6.49	79.72±8.20	77.37±8.20	90.10±2.82	96.71±0.83	57.15±1.96	62.50±8.14	70.29±1.45
OnlineHD	**88.89±4.53**	95.28±0.39	98.89±1.04	**95.19±1.76**	**99.16±0.34**	99.67±0.12	**87.80±4.17**	88.38±4.15	90.90±1.03
NeuralHD	86.39±2.58	95.83±1.80	96.39±0.79	94.44±1.03	98.49±0.38	99.01±0.32	83.52±2.98	87.56±2.56	91.97±1.35
CompHD	55.83±5.14	68.89±1.57	75.83±4.76	72.37±12.39	87.42±7.31	90.76±2.61	57.05±5.21	68.05±2.05	64.84±4.27
SparseHD	66.94±1.42	61.94±4.43	70.83±4.25	68.31±17.31	72.58±2.37	66.49±9.15	57.33±2.83	62.84±6.95	68.39±4.71
QuantHD	55.00±3.60	75.56±6.85	74.17±1.36	70.50±19.59	83.04±0.60	82.81±6.94	55.33±5.70	61.67±12.14	67.56±6.21
LeHDC	59.44±5.15	56.39±1.04	65.00±5.31	51.06±5.21	61.61±1.58	67.30±11.97	52.36±2.04	55.05±0.83	53.43±2.39
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	86.35±1.11	**96.05±0.57**	**98.97±0.49**	89.99±0.62	95.61±1.34	**99.97±0.83**	87.59±1.36	**89.72±1.43**	**92.04±0.26**

composed of handwritten digits from 0 to 9. Each image is 28×28 pixels, flattened into a 784-dimensional vector. MNIST is widely used for evaluating pattern recognition and machine learning algorithms.

The UCI HAR (Human Activity Recognition) dataset contains 561-dimensional feature vectors extracted from smartphone sensor signals. The data represents six human activities (e.g., walking, standing, lying down) collected from multiple subjects. It is a compact but structured dataset that tests the model's ability to distinguish between temporal movement patterns. These datasets vary in feature dimension, number of categories, and sample complexity, providing different assessment environments. To investigate the extensibility of our model, we performed a classification task ranging from 2 classes to 10 classes, rather than using a complete set of classes (for example, all 26

letters in ISOLET or all 10 numbers in MNIST). We observed that the full class classification task led to a significant decline in performance, so we adopted a progressive evaluation strategy that better reflects the actual capabilities of current quantum systems.

4.2 Low-Dimensional HDC and Quantum Encoding

While our approach is inspired by traditional HDC, we intentionally work in a significantly lower dimension([16,32,64]) to reflect the limitations of current quantum hardware. In the classical HDC literature, extremely high sizes (up to 10,000) are often used to ensure robustness and orthogonality. In contrast, our QeHDC performs well in low dimensions using the expressiveness of quantum states. For fair comparison, all HDC baselines are also constrained to the same vector dimension. This highlights a key difference in the design. While traditional techniques rely on the force dimension, our approach seeks to preserve a quantum-efficient representation of core HDC principles.

4.3 Implementation Details

Experiments are conducted in the following environments:

- **Qiskit Aer Simulator:**[14] An ideal, noiseless quantum simulation environment used for baseline accuracy and fidelity evaluation. The majority of our experiments are conducted using the Aer simulator, as it allows scalable, noise-free validation of our method across different datasets and class counts.

- **IBM QPU:** [26]A small scale of experiments is deployed on real IBM QPU backends to assess the practical feasibility of our framework under hardware constraints. Quantum state tomography is used to reconstruct output states due to the non-availability of state vectors on real hardware. Basic error mitigation techniques are applied where applicable. For comparison, the same scale and data samples used on IBM devices are also evaluated using the Aer simulator to establish a controlled benchmark.

- **Classical Baselines**: For traditional HDC and classical parts of our methods, we use TorchHD [9] implementations. All classical experiments are run on a workstation equipped with Lujun's station and he do not know the detail, GPU acceleration is used for efficient batching in TorchHD. Dimensionality (16, 32, 64) is matched with the quantum model, and all baselines are evaluated on the same training/testing splits as the quantum system.

All quantum operations, including encoding, binding, and fidelity computation, are implemented using Qiskit. State preparation and circuit compilation are optimized for each backend. In real-device executions, we use 8192 shots and transpile circuits to the native gate set of the selected backend. Fidelity measurements are obtained by reconstructing the output density matrices via tomography. Hyperparameters, such as the number of qubits ($\{4,5,6\}$), repeats (5), and choice of reference state, are fixed across experiments unless specified in dedicated analysis sections.

5 Results and Analysis

In this section, we present the key findings from our experiments. We focus on evaluating classification accuracy and the contribution of each core component in the QeHDC framework. All results are organized by task complexity and dataset, followed by comparative studies to highlight the advantages and limitations of our approach.

5.1 Overall Performance

Table 2 provides a comprehensive comparison of our QeHDC framework against various classical hyperdimensional computing (HDC) baselines—such as VanillaHD, AdaptHD, OnlineHD, NeuralHD, CompHD, SparseHD, QuantHD, and LeHDC—across three benchmark datasets (ISOLET, MNIST, and UCI HAR) under different dimensional constraints (16D, 32D, and 64D). The dimensions in the classical methods correspond respectively to quantum states encoded by 4, 5, and 6 qubits in our approach.

From these results, our QeHDC consistently achieves robust and superior classification accuracy. Specifically, on the ISOLET binary classification task, our model demonstrates accuracy that increases

significantly with the dimension, reaching 98.97% at 6 qubits (64D), clearly outperforming all classical baselines except OnlineHD, with which it achieves comparable performance. In the MNIST binary classification scenario, our approach further excels, reaching an accuracy of 99.97% at 6 qubits—surpassing or matching the best classical results. Likewise, for the UCI HAR dataset, which presents complex temporal feature patterns, our QeHDC method attains an accuracy of 92.04% at 6 qubits, which notably exceeds most classical baselines.

Importantly, even at lower-dimensional settings , our method maintains competitive performance. This highlights the effective expressive capacity of our quantum-inspired encoding and compression-based binding mechanisms. Such robust performance under dimensional constraints is particularly relevant for near-term quantum computing applications, where resource availability (e.g., number of qubits) is significantly limited.

Beyond binary classification tasks, our method also demonstrates promising results in more complex multi-class classification settings, systematically evaluated from 3-class up to 10-class problems. These additional experiments confirm that the QeHDC framework consistently retains strong classification capability across increasing classification complexity and various datasets. Detailed multi-class classification results and further analysis can be found in the Appendix D.

Overall, these extensive experimental evaluations indicate that our quantum-based HDC approach not only competes favorably against classical HDC methods but also shows distinct advantages regarding efficiency, scalability, and robustness across different classification scenarios.

5.2 Limitation Analysis

(a) Performance on ISOLET Dataset   (b) Performance on UCI HAR
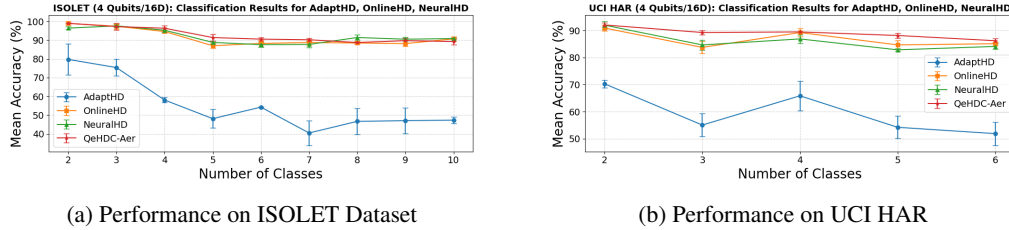
Figure 3: Mean classification accuracy of AdaptHD, OnlineHD, NeuralHD, and QeHDC-Aer on (a) ISOLET and (b) UCI HAR datasets with varying numbers of classes.

From the figures Fig. 3a and Fig. 3b, it is evident that our proposed Quantum Hyperdimensional Computing framework (QeHDC-Aer) consistently demonstrates stable and superior classification accuracy, around 90%, across varying classification complexities (from 2-class up to 10-class scenarios) on both ISOLET and UCI HAR datasets. In contrast, classical methods such as AdaptHD experience significant accuracy degradation and instability as the number of classes increases, highlighting the robustness of our approach under more complex classification tasks. Moreover, our method achieves performance comparable to, or better than, top-performing classical methods like OnlineHD and NeuralHD, especially notable on the UCI HAR dataset. Other classical baselines (e.g., VanillaHD, CompHD, SparseHD) fail to maintain satisfactory performance at such low-dimensional settings, indicating difficulty in handling complex tasks under limited dimensional constraints. These results clearly demonstrate the strong generalization capability and scalability of our QeHDC framework, emphasizing its advantage over classical HDC approaches in low-dimensional, resource-constrained quantum environments.

5.3 Aer Simulation and IBM QPU

To evaluate both the ideal performance and practical feasibility of our proposed QeHDC framework, we conduct experiments using the Qiskit Aer simulator and IBM Quantum Processing Units (QPUs). The Aer simulator provides a noise-free quantum environment with access to full quantum statevectors, allowing us to assess the expressiveness and fidelity of our model under ideal conditions, free from decoherence and gate imperfections. In contrast, IBM QPUs introduce realistic limitations such as noise, qubit connectivity constraints, and readout errors, offering a platform to validate the framework under physical quantum computing constraints.

Table 3: Small-batch Classification on UCI HAR (4 Qubits)

Method	# Qubits	# Train	# Test	Accuracy	Fidelity (to Aer Ideal)
QeHDC-Aer	4	50	20	0.80	0.98
QeHDC-QPU	4	50	20	0.75	0.87

On the Aer simulator, we observed high fidelity and classification accuracy for the quantum states generated after binding and template construction. These results validate the theoretical soundness of our encoding, binding, and template generation procedures. Specifically, the fidelity between the ideal post-binding statevector and the density matrix reconstructed via simulated tomography consistently exceeded 0.98, confirming the accuracy and stability of our tomography pipeline in a noise-free setting.

Quantum State Tomography (QST) plays a critical role in real-device testing where full access to quantum states is unavailable. By preparing and measuring the same quantum state across multiple bases and statistically reconstructing its density matrix, QST enables us to recover the output state from QPU measurements and perform fidelity-based classification. In our workflow, the test state is first encoded and the binding circuit is executed on the QPU. The resulting measurement data is then used to reconstruct a density matrix via QST, from which we extract a quantum representation that is compared with the stored SuperClass template to perform final classification.

Due to the significant cost and scaling complexity of QST and QPU, where the number of measurement circuits increases exponentially with the number of qubits. We adopted a small-batch testing strategy. Combined with device runtime constraints, this makes full-scale experiments infeasible on current hardware. As a representative case, we selected a small batch from the UCI HAR dataset (4-qubit setting), training on 50 samples and testing on 20. We redo this experiment for 5 times and get the average results in Table 3.

These results indicate that even under real noise, the quantum states reconstructed from the QPU retain sufficient structure to support meaningful classification, with fidelity and accuracy close to ideal conditions. Notably, the difference in fidelity between the QPU and simulator also quantifies the impact of hardware noise on the final inference performance.

The examples of our binding circuits, representative tomography circuits, and visual comparisons of ideal and reconstructed states are included in the Appendix E.

Overall, this experiment validates the expressiveness of our method under ideal conditions and its deployability on current meso-scale noisy quantum devices (NISQs), providing a practical basis and experimental experience for further expanding the QeHDC model on more powerful quantum hardware platforms in the future.

5.4 Ablation Study

In order to evaluate the effect of coding mechanism on the overall performance of QeHDC systems, we conduct an ablationon on Quantum encoder. The objective of this study is to analyze the effects of different coding strategies on the quality of class templates and the final classification accuracy.

6 Conclusion and Future Work

In this paper, a new framework Quantum-enhanced hyperdimensional computing (QeHDC) is proposed, which combines the basic idea of traditional hyperdimensional computing (HDC) with the ability to express quantum states. We design a classic-quantum two-step hybrid coding mechanism and introduce a reference state-based compression binding method to efficiently generate a compact high-dimensional quantum representation that can be used for classification tasks. At the same time, we construct a SuperClass template by means of density matrix decomposition, so that class abstraction can be realized in quantum representation space.

Unlike traditional HDC models, which typically rely on tens of thousands of dimensional vectors, our quantum scheme is able to operate in significantly low-dimensional (e.g. 64-dimensional) spaces,

and achieves competitive and even superior results on multiple classification tasks. Experiments on ISOLET, MNIST, and UCI HAR datasets demonstrate the effectiveness and robustness of the proposed method in simulated environments (Aer) and real devices (IBM QPU). Especially in the small batch test of QPU, the model can still obtain high quantum state fidelity and classification accuracy even if there is hardware noise, which indicates that the method has good usability and deployment potential.

In the future, we will promote our work from the following directions: On the one hand, we will expand this method to more complex task scenarios (such as sequence modeling, online learning), and explore the combination of quantum binding and class incremental update mechanisms. On the other hand, to further optimize the computational cost of quantum state chromatography, or to design a more suitable reasoning mechanism for the hardware structure to adapt to larger-scale quantum systems. In addition, the introduction of learnable quantum encoders and differentiable template generation processes is also expected to further enhance the integration of QeHDC with deep architectures such as neural networks. With the continuous development of quantum hardware, we believe that the framework proposed in this work will show its broad application prospects in larger-scale scenarios and provide a solid foundation for building the next generation of quantum cognitive computing systems.

References

[1] John-Mark A. Allen. Quantum superpositions cannot be epistemic. *Quantum Studies: Mathematics and Foundations*, 3:161–177, 2015.

[2] Erhan Bulbul, Aydin Cetin, and Ibrahim Alper Dogru. Human activity recognition using smartphones. In *Proceedings of 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6, 2018.

[3] Cheng-Yang Chang, Yu-Chuan Chuang, Chi-Tse Huang, and An-Yeu Wu. Recent progress and development of hyperdimensional computing (hdc) for edge intelligence. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 13(1):119–136, 2023.

[4] Rahul Chaurasia and Anirban Sengupta. Securehd: Designing low-cost reliable and security aware hardware accelerators during high-level synthesis for computationally intensive application frameworks. In *2024 IEEE International Symposium on Smart Electronic Systems (iSES)*, pages 380–383, 2024.

[5] Ron Cole and Mark Fanty. ISOLET. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C51G69.

[6] Shijin Duan, Yejia Liu, Shaolei Ren, and Xiaolin Xu. LeHDC: Learning-based hyperdimensional computing classifier. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 1111–1116, 2022.

[7] Manuel Eggimann, Abbas Rahimi, and Luca Benini. A 5 μw standard cell memory-based configurable hyperdimensional computing accelerator for always-on smart sensing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(10):4116–4128, 2021.

[8] Ross W Gayler. Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*, 2004.

[9] Mike Heddes, Igor Nunes, Pere Vergés, Denis Kleyko, Danny Abraham, Tony Givargis, Alexandru Nicolau, and Alexander Veidenbaum. Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures. *Journal of Machine Learning Research*, 24(255):1–10, 2023.

[10] Alejandro Hernández-Cano, Namiko Matsumoto, Eric Ping, and Mohsen Imani. Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system. In *Proceedings of 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 56–61, 2021.

[11] Mohsen Imani, Samuel Bosch, Sohum Datta, Sharadhi Ramakrishna, Sahand Salamat, Jan M Rabaey, and Tajana Rosing. QuantHD: A quantization framework for hyperdimensional computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 2268–2278, 2019.

[12] Mohsen Imani, Justin Morris, Samuel Bosch, Helen Shu, Giovanni De Micheli, and Tajana Rosing. Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing. In *Proceedings of 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, 2019.

[13] Mohsen Imani, Sahand Salamat, Behnam Khaleghi, Mohammad Samragh, Farinaz Koushanfar, and Tajana Rosing. SparseHD: Algorithm-hardware co-optimization for efficient high-dimensional computing. In *Proceedings of 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 190–198, 2019.

[14] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.

[15] P. Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1:139–159, 2009.

[16] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, Abbas Rahimi, and A. Sebastian. In-memory hyperdimensional computing. *Nature Electronics*, 3:327 – 337, 2019.

[17] Hyukjun Kwon, Kangwon Kim, Junyoung Lee, Hyun-Soo Lee, Jiseung Kim, Jinhyung Kim, Taehyung Kim, Yongnyeon Kim, Yang Ni, Mohsen Imani, Ilhong Suh, and Yeseong Kim. Brain-inspired hyperdimensional computing in the wild: Lightweight symbolic learning for sensorimotor controls of wheeled robots. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5176–5182, 2024.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 2278–2324, 1998.

[19] Alisha Menon, Anirudh Natarajan, Laura Isabel Galindez Olascoaga, Youbin Kim, Braeden C. Benedict, and J. Rabaey. On the role of hyperdimensional computing for behavioral prioritization in reactive robot navigation tasks. *2022 International Conference on Robotics and Automation (ICRA)*, pages 7335–7341, 2022.

[20] A. Mitrokhin, Peter Sutor, C. Fermüller, and Y. Aloimonos. Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Science Robotics*, 4, 2019.

[21] Justin Morris, Mohsen Imani, Samuel Bosch, Anthony Thomas, Helen Shu, and Tajana Rosing. CompHD: Efficient hyperdimensional computing using model compression. In *Proceedings of 2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2019.

[22] Fateme Rasti Najafabadi, Abbas Rahimi, Pentti Kanerva, and Jan M Rabaey. Hyperdimensional computing for text classification. In *Design, automation test in Europe conference exhibition (DATE), University Booth*, pages 1–1, 2016.

[23] Igor Nunes, Mike Heddes, Tony Givargis, Alexandru Nicolau, and Alex Veidenbaum. Graphhd: Efficient graph classification using hyperdimensional computing. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1485–1490. IEEE, 2022.

[24] U. Pale, T. Teijeiro, and David Atienza Alonso. Systematic assessment of hyperdimensional computing for epileptic seizure detection. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 6361–6367, 2021.

[25] Kaspar A Schindler and Abbas Rahimi. A primer on hyperdimensional computing for ieeg seizure detection. *Frontiers in neurology*, 12:701791, 2021.

[26] Andreas Sturm. Theory and implementation of the quantum approximate optimization algorithm: A comprehensible introduction and case study using qiskit and ibm quantum computers. 2023.

[27] G. Thekkadath, L. Giner, Y. Chalich, Matthew Horton, J. Banker, and Jeff S. Lundeen. Direct measurement of the density matrix of a quantum system. *Physical review letters*, 117 12:120401, 2016.

[28] Junyao Wang, Sitao Huang, and M. Imani. DistHD: A learner-aware dynamic encoding method for hyperdimensional classification. *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023.

[29] Zhuowen Zou, Hanning Chen, Prathyush P. Poduval, Yeseong Kim, Mahdi Imani, Elaheh Sadredini, Rosario Cammarota, and M. Imani. Biohd: an efficient genome sequence search platform using hyperdimensional memorization. *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022.

[30] Zhuowen Zou, Yeseong Kim, Farhad Imani, Haleh Alimohamadi, Rosario Cammarota, and Mohsen Imani. Scalable edge-based hyperdimensional learning system with brain-like neural adaptation. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

A Parameter Usage Comparison

In QeHDC, the choice of encoding strategy significantly affects the **phase angle distribution** in the complex plane. It directly impacts the separability and expressiveness of quantum states. This section compares and analyzes three sinusoidal transformation strategies: Single-Weight Cross-Multiplicative (our method), Dual-Weight Same-Product, and Single-Weight Same-Product. We compare these methods with respect to parameter efficiency, phase angle distribution, and normalization properties.

A.1 Phase Angle Concentration in Single-Weight Same-Product

In **Single-Weight Same-Product Encoding**, both the real and imaginary components share the **same weight matrix**, defined as:

$$\text{Re}(\Psi') = \cos(WX + B) \cdot \sin(WX),$$

$$\text{Im}(\Psi') = \cos(WX + B) \cdot \sin(WX)$$

Since the transformation for both components is identical, their ratio remains constant:

$$\frac{\text{Im}(\Psi')}{\text{Re}(\Psi')} = 1$$

This results in all encoded data points having the same phase angle. Consequently, the data distribution in the complex plane becomes **highly concentrated**, reducing the effectiveness of encoding.

A.2 Parameter Efficiency in Cross-Multiplicative Transformation

Single-Weight Cross-Multiplicative Encoding solves this issue by using a single weight matrix while applying different transformations to the real and imaginary components:

$$\text{Re}(\Phi) = \cos(WX + B) \cdot \sin(WX),$$

$$\text{Im}(\Phi) = \sin(WX + B) \cdot \cos(WX)$$

This ensures a uniform phase angle distribution while minimizing parameter usage.

In contrast, Dual-Weight Single-Product Encoding requires two independent weight matrices to achieve the same effect:

$$\text{Re}(\Psi'') = \cos(W_{\text{re}}X + B) \cdot \sin(W_{\text{re}}X),$$

$$\text{Im}(\Psi'') = \cos(W_{\text{im}}X + B) \cdot \sin(W_{\text{im}}X)$$

While this method also maintains a uniform phase angle distribution, it doubles the parameter count, increasing computational overhead.

B Compressed Binding

This appendix provides a detailed analysis demonstrating that traditional binding methods used in classical HDC, such as element-wise multiplication or circular convolution, lead to an exponential explosion in the number of required qubits when directly applied to quantum systems. Although theoretically feasible to use quantum state amplitude as classical high-dimensional vectors for binding, such an approach is practically infeasible in quantum hardware. Additionally, this approach has been explored in previous studies, but predominantly through classical binding in classical HDC, which is usually defined via element-wise multiplication or XOR operation simulations rather than quantum implementation. Consider 2 D-dimensional vectors:

$$A = [a_1, a_2, .., a_D], B = [b_1, b_2, ..., b_D]$$

Their binding result (element-wise multiplication) is expressed as:

$$C = A \circ B = [a_1 b_1, a_2 b_2, ..., a_D b_D]$$

Consider 2 D-dimensional vectors:

$$A = [a_1, 1_2, .., a_D], B = [b_1, b_2, ..., b_D]$$

15

Their binding result (element-wise multiplication) is expressed as:

$$C = A \circ B = [a_1 b_1, a_2 b_2, ..., a_D b_D]$$

This type of binding operation is efficient and easy to perform in classical system, but if it is implemented directly in a quantum system, it faces huge resource challenges.

In quantum system, a quantum state of n qubits represents a 2^n-dimensional complex vector:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha |i\rangle, \quad \sum |\alpha|^2 = 1$$

If we want to bind two quantum states using the traditional HDC binding method, we first need to simultaneously represent two full $D = 2^n$-dimensional quantum states in the quantum computer. Each stare requires n qubits and the joint representation of two states thus requires $2n$ qubits. If we bind k quantum states, the toatal number of qubits required scales as:

$$Q_{total} = k \cdot n$$

This implies that if we directly adopt traditional binding methods, adding each additional state linearly increases the required qubits. Given that each quantum state's dimensionality grows exponentially, total computational resources effectively explode exponentially. Theoretically, we could directly use quantum state amplitudes as classical high-dimensional vectors for binding. However, this approach does not genuinely leverage quantum computing advantages, as it simply treats quantum amplitudes as classical data. Hence, it deviates from our quantum algorithmic context, failing to harness quantum parallelism or the computational benefits of quantum superposition. It is worth noting that previous studies have explored this "amplitude-as-vector" binding approach. Nevertheless, these studies typically relied on classical simulations rather than actual implementations on real quantum hardware.

C Bind Circuits Alternatives

Below, we describe two alternative entanglement structures of bind Circuits with lower complexity than the fully-connected entanglement structure.

C.1 Linear (Chain) Entanglement

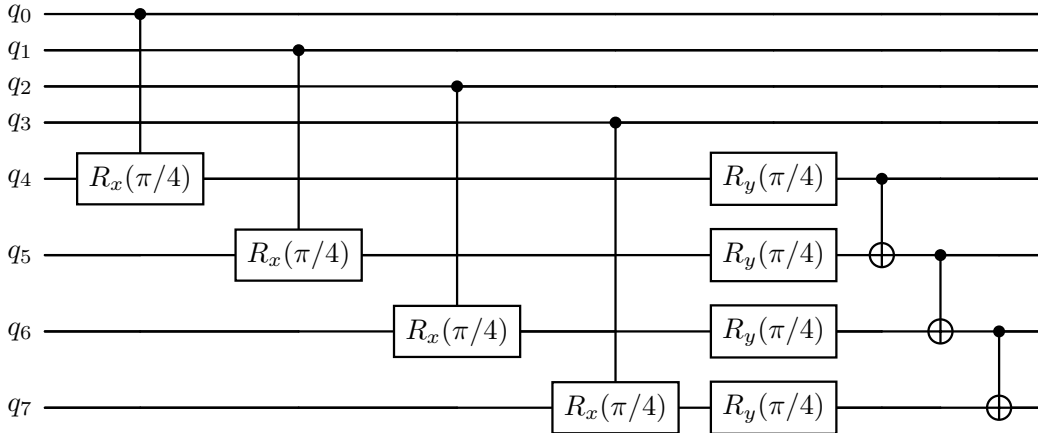The circuit for the example with four qubits is shown in the figure.

Figure 4: Linear Entanglement Binding Circuit with 4 qubits.

In this structure which, qubits in the second half (indexed from n to $2n - 1$) are entangled sequentially in a linear (chain-like) configuration. Each qubit is entangled directly with its immediate neighbor, resulting in a simpler, lower-complexity structure.

16

The **unitary operator** representing this structure is expressed as follows:

$$U_{\text{linear}} = U_{\text{chain}} \cdot U_{\text{param}} \tag{23}$$

where

$$U_{\text{chain}} = \prod_{i=n}^{2n-2} CX_{i,\,i+1}, \tag{24}$$

and

$$U_{\text{param}} = \prod_{i=0}^{n-1} \left[RY\left(\frac{\pi}{4}\right)_{n+i} \cdot CRX(\theta_i)_{i,\,n+i} \right]. \tag{25}$$

C.2 Ring (Circular) Entanglement

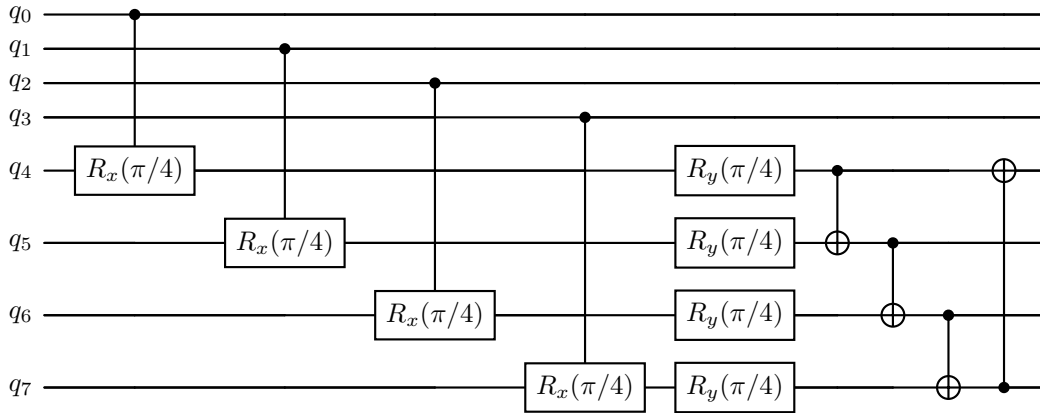The circuit for the example with four qubits is shown in the figure.

Figure 5: Ring Entanglement Binding Circuit with 4 qubits.

In this entanglement structure, the qubits in the latter half form a closed loop. Each qubit is entangled directly with its neighbor, including an additional connection between the first and last qubits to complete the ring. This structure is slightly more complex than linear entanglement but simpler than the fully connected entanglement.

The **unitary operator** for ring entanglement is expressed as:

$$U_{\text{ring}} = U_{\text{circle}} \cdot U_{\text{param}} \tag{26}$$

where

$$U_{\text{circle}} = \prod_{i=n}^{2n-1} CX_{i,\,((i-n+1) \mod n)+n}, \tag{27}$$

and U_{param} is as defined above.

Here, qubits indexed n to $2n - 1$ form a circular chain, ensuring each qubit is entangled with exactly two neighbors.

D Complete Results for Classification

In addition to the two-class evaluations presented in the main text, we extend our experiments to multi-class classification tasks with increasing numbers of classes. Specifically, we conduct classification from 3 to 10 classes on the ISOLET and MNIST datasets, and from 3 to 6 classes on the UCI HAR dataset.

These experiments follow the same QeHDC framework, using one pass training and fidelity-based inference under fixed qubit and dimension configurations. Class-balanced subsets are selected for each setting.

17

The goal is to evaluate the scalability and stability of our method under increasing classification complexity. The results, summarized in the Table 4 to 11, show that although accuracy naturally decreases as the number of classes increases, our method maintains strong relative performance compared to classical HDC baselines, especially under low-dimensional constraints.

E Tomography Circuits and Resconstructed State Comparison

This appendix illustrates the tomography process used in our quantum classification framework, including representative tomography circuits and a comparison between the ideal post-binding state and the reconstructed density matrix.

Fig. 6, Fig. 7, and Fig. 8 show three example circuits generated by the tomography module. These circuits correspond to different measurement bases required for full quantum state reconstruction. To emphasize the core structure, all circuits are presented in a cleaned form with the initialization components removed.

After executing all tomography circuits, we reconstruct the output density matrix and compare it with the ideal quantum statevector obtained from the Aer simulator. Fig. 9 and 10 show the 3D cityscape visualizations of the complex amplitudes. The real and imaginary parts of both the ideal and reconstructed states are shown side by side for comparison.

As seen in the figures, the overall structure and amplitude distribution between the ideal and reconstructed states are highly consistent, validating the effectiveness of our QST pipeline even under limited measurement noise. These results demonstrate that the reconstructed quantum states retain sufficient fidelity for classification, as further supported by the numerical fidelity scores in the main text.

Table 4: 3-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-3Classes			MNIST-3Classes			UCI HAR-3Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	56.85±6.04	65.37±2.73	71.67±5.90	54.90±7.66	76.18±4.68	83.34±2.26	47.85±2.16	54.29±1.29	54.63±1.98
AdaptHD	54.81±10.82	63.89±7.26	75.37±4.48	55.36±11.27	71.80±6.06	83.53±1.92	45.93±5.30	46.84±4.49	55.06±4.18
OnlineHD	86.85±5.45	94.81±1.89	97.41±1.39	74.96±8.72	83.15±2.88	94.53±0.41	65.66±7.46	77.34±5.70	83.75±2.21
NeuralHD	86.48±2.28	93.89±0.79	97.59±0.94	79.84±3.33	88.20±2.15	93.37±1.14	68.42±3.57	77.72±0.51	84.67±1.83
CompHD	52.78±7.74	63.33±2.36	68.15±4.12	49.66±3.36	59.93±1.56	75.29±1.62	45.06±1.18	46.07±1.85	53.06±3.37
SparseHD	51.48±5.00	61.48±7.61	73.33±7.86	44.46±12.26	43.00±4.77	64.65±17.43	41.14±3.05	46.43±3.78	51.43±3.60
QuantHD	52.96±8.95	58.89±3.63	71.30±2.14	64.07±5.01	67.10±5.01	67.92±1.87	38.02±8.45	44.97±3.41	50.54±5.82
LeHDC	46.67±7.07	48.15±2.73	53.89±9.50	44.55±5.35	53.21±5.02	57.91±11.07	37.66±0.65	38.62±2.94	42.78±2.03

Table 5: 4-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-4Classes			MNIST-4Classes			UCI HAR-4Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	43.75±6.49	59.03±6.33	65.83±0.59	47.17±2.89	62.33±3.42	75.81±2.35	55.75±4.70	62.74±2.83	69.36±1.56
AdaptHD	33.61±6.13	45.14±6.58	58.06±1.37	34.86±7.01	62.61±6.05	71.77±5.90	46.96±9.54	51.24±8.91	65.87±5.50
OnlineHD	79.44±2.60	84.17±1.89	94.58±0.68	61.32±3.67	78.00±2.44	87.53±2.04	72.10±2.94	81.82±2.17	89.24±1.55
NeuralHD	80.69±1.87	89.17±2.91	95.28±1.04	73.07±4.01	80.76±2.49	87.82±0.62	77.53±3.43	83.28±1.20	86.88±1.54
CompHD	39.44±0.71	46.53±2.60	59.72±5.40	37.90±2.92	52.08±1.22	64.29±1.58	42.55±11.07	56.59±1.59	61.87±5.04
SparseHD	42.08±3.25	41.94±4.82	61.67±5.31	35.54±4.85	37.40±4.10	38.35±10.15	39.94±3.58	50.66±2.50	56.37±4.04
QuantHD	34.17±5.62	52.50±3.25	55.00±2.91	43.78±0.69	43.47±1.50	50.85±7.99	32.14±6.62	52.48±2.35	60.21±6.60
LeHDC	32.64±3.99	40.83±2.57	38.33±7.27	32.79±5.02	38.92±8.12	50.01±5.80	43.91±5.97	40.47±3.11	47.46±1.29

Table 6: 5-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-5Classes			MNIST-5Classes			UCI HAR-5Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	34.33±2.13	54.89±4.12	56.11±5.65	44.54±4.14	58.66±3.18	71.17±3.59	47.25±3.31	51.81±2.70	59.97±3.75
AdaptHD	32.56±4.73	42.56±10.02	48.11±4.94	34.43±5.58	46.27±5.24	62.39±11.57	32.45±4.27	49.07±2.78	54.23±4.15
OnlineHD	62.22±3.45	78.44±0.79	87.00±1.25	53.30±10.33	76.50±2.72	86.98±0.93	62.14±4.12	75.19±3.11	84.67±1.73
NeuralHD	70.78±4.36	80.33±1.96	88.78±0.87	69.59±0.80	79.67±3.36	86.95±1.38	71.44±3.02	79.52±1.56	82.85±0.54
CompHD	31.22±4.01	36.89±4.31	46.67±0.72	38.79±6.67	48.76±3.28	57.24±0.80	41.96±4.42	45.39±0.50	54.54±2.33
SparseHD	23.78±3.50	38.78±0.57	44.78±2.78	28.71±2.32	30.85±3.36	35.12±3.32	33.87±9.18	38.87±4.20	47.40±0.76
QuantHD	24.22±5.97	34.67±4.23	50.11±5.12	29.78±8.10	36.56±6.67	53.29±1.56	34.34±1.32	43.62±5.63	44.48±3.34
LeHDC	25.67±2.42	31.33±2.33	32.56±4.24	32.54±3.07	30.31±3.60	30.73±5.82	27.62±5.46	33.89±4.39	38.38±1.14

Table 7: 6-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-6Classes			MNIST-6Classes			UCI HAR-6Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	38.70±3.46	53.06±3.15	55.46±2.93	44.58±3.79	47.49±5.53	57.42±2.79	36.91±4.12	46.08±5.89	57.26±0.48
AdaptHD	30.56±1.64	39.17±7.20	54.26±0.13	27.00±9.06	32.54±8.22	55.82±6.17	38.91±10.41	42.11±2.05	51.89±4.34
OnlineHD	63.52±3.10	80.93±3.67	88.24±1.82	51.21±3.14	67.49±1.42	80.39±1.85	68.47±2.94	76.96±3.53	85.11±0.67
NeuralHD	75.09±0.92	82.69±2.16	87.59±1.33	63.05±1.87	71.76±0.66	82.70±0.98	73.56±1.65	80.24±0.98	84.11±0.99
CompHD	29.44±4.47	41.67±3.20	52.50±2.40	30.46±1.68	40.80±5.94	51.21±4.03	32.47±4.98	41.42±7.17	51.95±3.16
SparseHD	23.98±4.81	36.76±2.92	50.74±1.98	23.76±4.26	24.99±5.00	27.75±5.64	30.84±1.15	35.69±3.96	41.78±4.06
QuantHD	29.81±3.19	40.19±2.87	46.39±4.74	31.76±10.03	36.68±3.68	39.92±6.45	28.06±2.35	40.48±3.19	38.89±0.99
LeHDC	23.61±3.60	28.15±4.93	34.35±4.67	25.86±4.28	30.90±2.35	32.76±4.65	21.67±2.43	31.21±3.91	34.02±6.37

Table 8: 7-Class Results (ISOLET and MNIST)

Method	ISOLET-7Classes			MNIST-7Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	36.75±7.88	47.94±2.59	54.21±3.77	34.69±7.28	43.94±1.91	53.18±1.64
AdaptHD	26.43±3.03	30.32±2.35	40.48±6.62	32.28±6.47	30.57±4.43	54.22±2.76
OnlineHD	67.14±1.47	80.71±1.85	88.73±1.63	50.60±3.36	63.90±1.21	80.04±1.12
NeuralHD	71.51±0.59	81.75±3.34	87.70±1.81	52.95±2.54	71.00±1.94	80.43±1.30
CompHD	28.49±4.89	30.32±0.74	45.95±2.21	26.21±2.02	36.59±0.51	45.29±2.88
SparseHD	24.92±2.65	31.35±3.57	41.59±2.63	22.16±2.90	19.84±2.22	25.70±4.52
QuantHD	30.16±5.63	37.54±5.35	44.05±4.60	25.84±4.11	31.91±6.93	39.29±6.00
LeHDC	20.48±3.41	22.14±2.44	29.44±1.29	22.17±4.49	24.20±2.38	34.02±6.17

Table 9: 8-Class Results (ISOLET and MNIST)

Method	ISOLET-8Classes			MNIST-8Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	33.33±3.96	40.49±3.45	57.85±3.47	30.77±1.49	43.19±3.33	58.51±1.72
AdaptHD	27.50±1.53	37.71±6.78	46.67±7.04	24.16±1.67	33.92±5.43	47.44±7.57
OnlineHD	65.42±5.35	80.90±1.37	88.47±0.39	42.01±3.21	60.49±1.76	77.44±1.54
NeuralHD	70.90±1.87	85.76±2.06	91.39±1.43	56.90±2.44	69.39±2.67	80.10±0.66
CompHD	25.35±0.55	32.85±0.94	42.57±4.53	24.54±3.71	32.65±2.69	42.71±0.97
SparseHD	17.50±4.36	30.00±4.34	35.14±3.09	22.64±5.33	19.76±4.34	26.02±1.65
QuantHD	18.61±4.03	30.62±2.86	39.79±1.84	16.57±6.14	37.69±0.89	39.40±2.17
LeHDC	16.32±1.90	19.93±1.82	28.19±2.77	16.70±4.37	23.32±1.14	29.05±3.52

Table 10: 9-Class Results (ISOLET and MNIST)

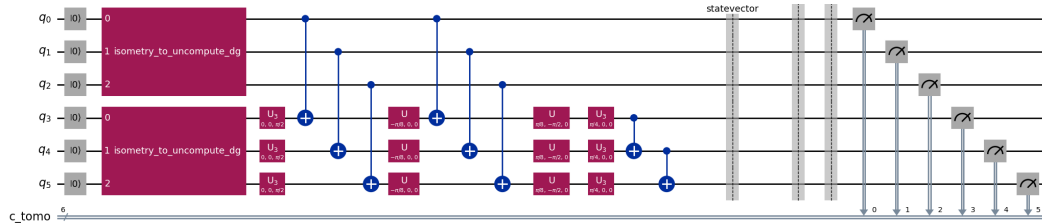| Method	ISOLET-9Classes			MNIST-9Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	29.44±4.92	41.98±1.83	57.16±2.60	27.87±0.53	39.08±4.12	52.22±2.58
AdaptHD	25.43±2.57	24.01±6.12	47.04±6.91	21.74±3.17	23.52±7.59	47.90±3.54
OnlineHD	62.78±4.74	84.51±0.76	88.15±1.46	37.52±4.24	54.04±2.83	71.56±1.28
NeuralHD	70.62±3.33	83.21±2.06	90.49±1.01	53.65±1.53	68.81±1.62	77.13±0.85
CompHD	26.91±1.44	30.43±1.57	41.54±5.18	23.36±1.98	29.93±2.70	38.34±1.48
SparseHD	17.35±4.61	24.26±0.69	37.41±4.45	15.79±2.40	18.57±2.11	21.57±3.18
QuantHD	20.37±2.72	26.42±3.29	38.02±3.97	20.80±2.57	29.95±0.28	36.52±4.86
LeHDC	20.06±4.52	22.65±4.83	31.60±1.29	19.33±0.34	18.30±0.39	25.10±4.19

Figure 6: Quantum State Tomography Circuit (Z bias)

Table 11: 10-Class Results (ISOLET and MNIST)

| Method	ISOLET-10Classes			MNIST-10Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	27.72±2.73	39.78±3.68	54.11±2.70	23.49±3.11	30.96±4.59	46.61±4.41
AdaptHD	16.28±1.95	25.11±2.91	47.33±1.78	16.42±4.03	21.82±4.04	36.27±4.52
OnlineHD	62.89±4.83	81.89±1.47	90.67±1.30	34.03±1.42	53.24±4.90	69.79±0.30
NeuralHD	68.67±3.42	85.61±1.80	90.83±0.49	47.04±1.52	63.92±1.89	75.73±0.36
DistHD	75.67±3.31	86.00±2.12	92.94±0.63	50.84±2.48	67.90±2.13	78.97±1.54
CompHD	21.39±4.47	32.44±3.85	35.22±1.61	20.25±0.67	26.88±1.51	36.88±2.37
SparseHD	13.94±1.59	22.06±2.79	36.11±3.94	14.07±1.66	14.53±0.92	23.50±3.23
QuantHD	21.61±2.09	28.33±4.90	39.39±6.28	21.17±2.65	25.27±1.64	30.93±2.74
LeHDC	14.78±5.18	17.17±4.25	22.33±1.91	15.43±3.16	17.43±4.05	26.50±2.60
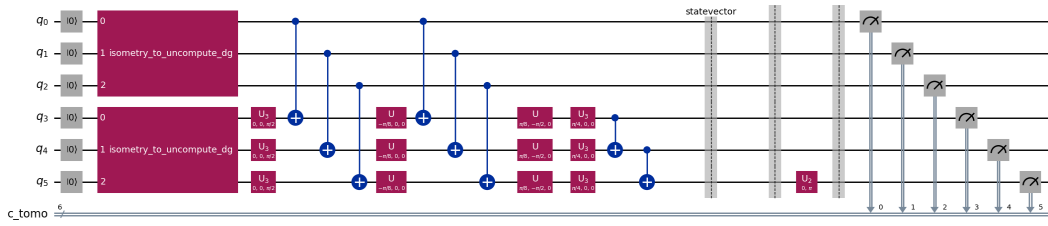
Figure 7: Quantum State Tomography Circuit (X bias)

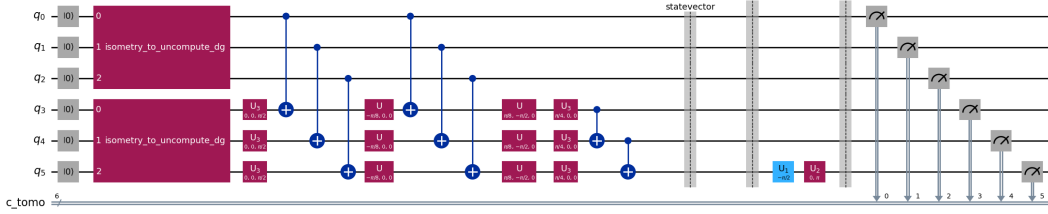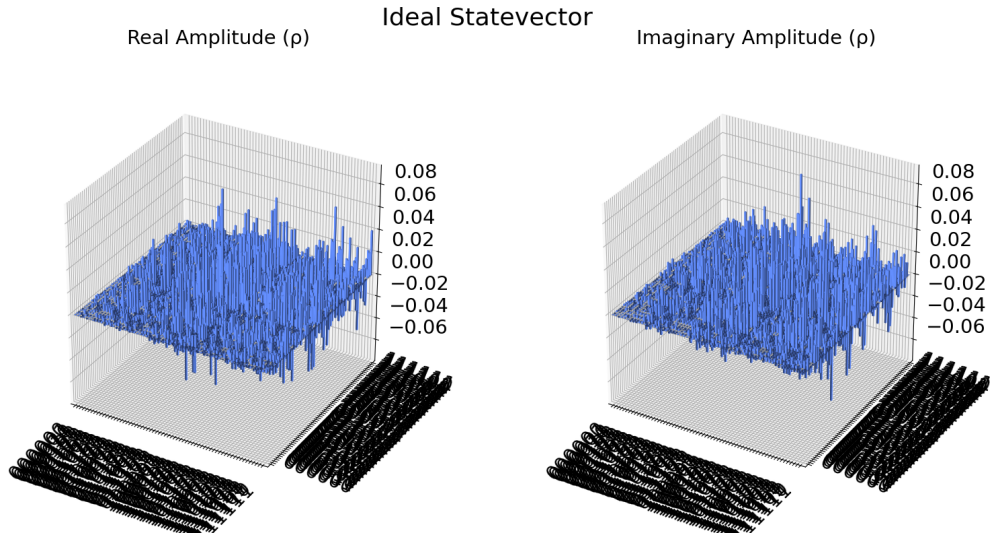Figure 8: Quantum State Tomography Circuit (Y bias)

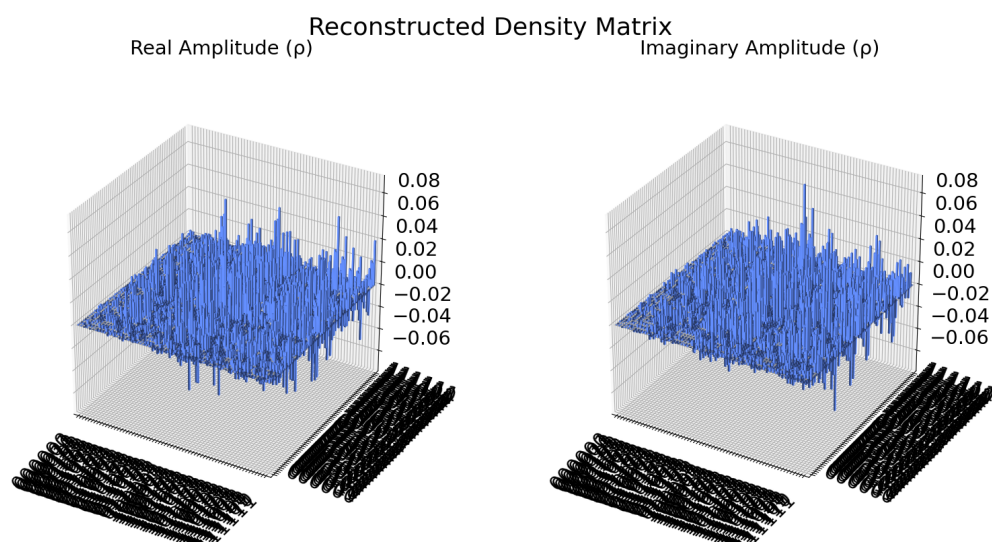Figure 9: Ideal Statevector (real and d Imaginary Components)

20

Reconstructed Density Matrix

Real Amplitude (ρ)   Imaginary Amplitude (ρ)

Figure 10: Reconstructed Density Matrix (Real and Imaginary Components)

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

24

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [TODO]

   Justification: [TODO]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [TODO]

   Justification: [TODO]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [TODO]

   Justification: [TODO]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [TODO]

Justification: [TODO]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.