QILLER: Quantum Incremental Learning for Lifelong Erosion Resilience in Variational Quantum Algorithms

Journal:	*IEEE Transactions on Neural Networks and Learning Systems*
Manuscript ID	TNNLS-2024-P-38194
Manuscript Type:	Regular Paper
Keywords:	Quantum Machine Learning, Representation Learning, Incremental Learning, Knowledge Distillation, Catastrophic Forgetting

QILLER: Quantum Incremental Learning for Lifelong Erosion Resilience in Variational Quantum Algorithms

***Abstract*—Catastrophic forgetting, also known as lifelong erosion resilience, is a phenomenon in machine learning, particularly in neural networks, where acquiring new information causes the model to forget previously learned knowledge. This issue extends to quantum machine learning, which, like its classical counterpart, is vulnerable to catastrophic forgetting, especially in classification tasks. Although quantum machine learning holds promise for quantum speedup in dynamic environments with large, continuous data streams, catastrophic forgetting remains a significant hurdle in continual learning scenarios. In this paper, we examine catastrophic forgetting within the framework of variational quantum learning, applying it to both quantum models and quantum data. Our findings reveal that state-of-the-art representation learning-based variational quantum models also experience this challenge in classification tasks. We propose a novel approach to address this issue that integrates representation learning, knowledge distillation, and an exemplar memory-based incremental learning model. This method mitigates catastrophic forgetting, achieving stable accuracy in representation learning-based variational quantum models and enhancing quantum speedup in continual learning contexts.**

***Index Terms*—Quantum Machine Learning, Representation Learning, Incremental Learning, Knowledge Distillation, Catastrophic Forgetting**

I. Introduction

The field of continual learning, also known as lifelong learning, draws inspiration from the capacity of humans and other animals to incrementally acquire new skills while retaining previously learned knowledge [1], [2]. This natural intelligence phenomenon serves as a model for enhancing the continual learning capabilities in artificial intelligence systems, which utilize continuous data streams to learn from new experiences while preserving previously acquired knowledge [3], [4], [5].

Incremental learning bridges the gap between natural and artificial intelligence in continual learning systems. This approach emphasizes the gradual acquisition of new information from continuous data streams [6]. Its capability to support real-time, large-scale data processing generates significant interest, particularly in the context of Big Data [7]. Incremental learning finds applications in smart cities, financial systems, and social media platforms [8].

An incremental learning algorithm generates a sequence of models $g_1, g_2, \ldots, g_k$ based on a defined sequence of training data $t_1, t_2, \ldots, t_k$. Each t_j represents labeled training data, defined as

$$t_j = (a_j, b_j) \in \mathbb{R}^m \times \{1, \ldots, D\}. \tag{1}$$

The model function is expressed as

$$g_j : \mathbb{R}^m \to \{1, \ldots, D\}, \tag{2}$$

which depends solely on the previous model g_{j-1} and the most recent q samples $t_j, \ldots, t_{j-q}$, where q is strictly limited [8]. Despite its advantages, incremental learning with deep neural networks faces several challenges. The model adapts progressively, meaning that g_{j+1} is developed from g_j without requiring full retraining. A critical aspect of incremental learning is preserving previously acquired knowledge to prevent catastrophic forgetting while retaining only a restricted set of q training examples. Catastrophic forgetting is a well-documented phenomenon in machine learning, particularly in neural networks, where learning new information leads to the loss of previously acquired knowledge [9].

The emerging field of Quantum Machine Learning (QML) [10] introduces innovative approaches to tackle these challenges by harnessing the unique properties of quantum computing, such as superposition, entanglement, and quantum parallelism. Variational quantum algorithms (VQAs), a cornerstone of QML, have emerged as a promising solution for enabling incremental learning. QML leverages quantum states to store and manipulate quantum data, which represents information encoded in quantum states. This capability enables QML to leverage the principles of quantum mechanics more effectively than classical methods, addressing key challenges in machine learning while improving model adaptability [11], [12]. VQAs employ quantum circuits to approximate machine learning models, offering significant advantages in scalability and computational efficiency [13]. Practical demonstrations further underscore their effectiveness across various domains, highlighting the transformative potential of QML in incremental learning [14]–[16].

The phenomenon of catastrophic forgetting has also posed a significant challenge in quantum machine learning, particularly in VQAs [17]–[19]. These algorithms rely on parametric quantum circuits, which operate like classical neural networks. However, like their classical counterparts, VQAs are typically designed for specific tasks, making it challenging to learn multiple tasks sequentially. Wenjie Jiang et al. [20] highlighted this issue by training a quantum classifier on two distinct tasks: recognizing quantum phases of matter and classifying handwritten digits. Initially, the classifier was trained to differentiate quantum phases using the Hamiltonian:

$$H = J_x \sum_{\langle i,j \rangle_x} \sigma_i^x \sigma_j^x + J_y \sum_{\langle i,j \rangle_y} \sigma_i^y \sigma_j^y - t \sum_{\langle i,j \rangle} c_i^\dagger c_j + \mu \sum_i n_i n_{i+1}, \tag{3}$$

where J_x and J_y represented the strengths of spin interactions in the x- and y-directions, respectively. The term t denoted the strength of particle movement between neighboring sites, while μ represented local interaction strength. The classifier utilized the Chern number C_1:

$$C_1 = \int_{\text{area}} F_{xy}(k)\, dk, \tag{4}$$

and Berry curvature $F_{xy}(\mathbf{k})$, which described the geometric properties of quantum states. These tools enabled the classifier to accurately distinguish between different quantum phases. However, when the same classifier was subsequently trained on the MNIST dataset for handwritten digit recognition, its performance on the original phase classification task declined. While it became better at identifying digits, it forgot how to classify quantum phases accurately. This phenomenon, known as catastrophic forgetting, illustrates the difficulty of retaining previously acquired knowledge while learning new tasks.

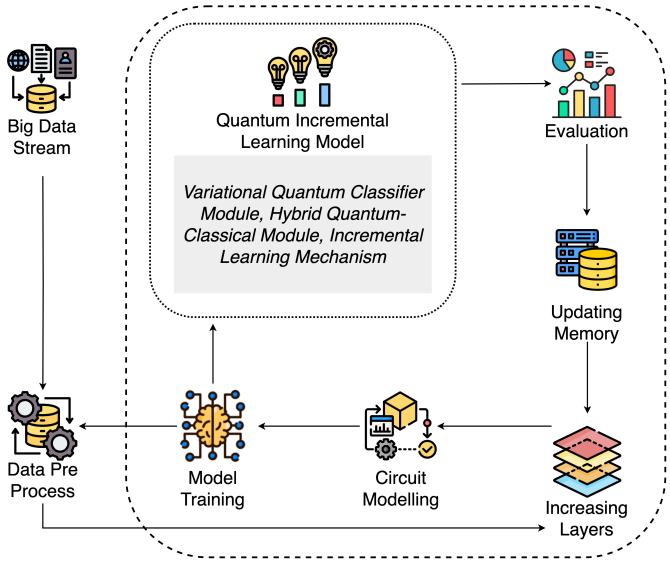

Fig. 1. Quantum Incremental Learning: Existing quantum incremental learning processes generally follow traditional machine learning methodologies but include two distinct features: the addition of more quantum circuit layers to enhance model capacity and the use of a memory mechanism that stores previous data, which grows incrementally over time.

An effective incremental deep learning approach handles a continuous data flow with randomly appearing classes, maintains high performance on old and new classes, uses a reasonable number of parameters and memory, and updates the classifier and feature representation together, managing infinite classes without losing accuracy or increasing parameters. Existing approaches [17], [18], [20], which follow the incremental training setup illustrated in Figure 1, fail to meet these requirements. These methods are often limited to theoretical analysis and specific tasks, with a greater emphasis on learning new tasks rather than expanding to new classes. Many methods employ Variational Quantum Classifiers (VQCs), a quantum classifier based on VQAs. However, these approaches typically increase the number of parameters or layers to generalize the model, resulting in a higher demand for qubits, quantum operations, quantum gates, and memory usage.

In this paper, we propose a decoupled representation learning approach as a feature extractor combined with a VQC as the classifier, utilizing an incremental learning strategy that enhances resilience against catastrophic forgetting in VQCs. This method is extendable to any classical or quantum feature extractor. Initially, the representation learning model, a quantum autoencoder, is trained using supervised contrastive learning loss to transform classical data into quantum data. Incremental learning is then performed using quantum features produced by the frozen autoencoder. These features are passed to the VQC, which optimizes a combination of cross-entropy loss and distillation loss—cross-entropy for learning new classes and distillation to retain knowledge of previously learned classes. We demonstrate the effectiveness of our approach by achieving state-of-the-art results for incremental learning on MNIST, FMNIST, KMNIST, and CIFAR-10. The rest of this manuscript is structured as follows: Section II provides a review of related works. Section III details the proposed architecture. Section III-G presents experimental results showcasing the effectiveness of our approach across various applications. Finally, Section IV concludes with a summary and outlines potential future research directions.

II. Related Works

Traditional incremental learning often relies on Support Vector Machines (SVMs), as demonstrated in works like [21] and [22]. These approaches incrementally and decrementally solve quadratic programming (QP) problems but risk missing potential support vectors if the candidate set is too small. LASVM [23], an efficient online SVM, reduces training time by considering only the current example as a potential support vector, though it provides an approximate solution. Despite promising results, these traditional algorithms are task-specific and struggle with the large datasets common today, leading to decreased performance. Our proposed model overcomes these limitations by combining representation learning with quantum advantages for high-dimensional data.

In deep learning, Stochastic Gradient Descent (SGD) with deep neural networks proves effective for large-scale learning [24], [25], showing success in [26] and [27]. However, these models perform poorly with low-dimensional data and are limited to linear class boundaries.

Knowledge distillation in incremental learning [28] is notable for its resilience against catastrophic forgetting. Li et al. [29] combine distillation loss with cross-entropy loss, reducing forgetting in cases where old and new samples are distinct. However, this approach falls short in high-dimensional learning scenarios, which our model addresses through representation learning.

Other methods, such as [30], freeze layers to restrict new feature learning, while Terekhov et al. [31] propose adding layers to learn new features, including new classes. These strategies degrade performance by increasing parameters, which is particularly problematic for VQCs on Noisy Intermediate-Scale Quantum (NISQ) devices with limited qubits.

The iCaRL method [32] addresses incremental learning by integrating a classifier with a feature representation model,

enabling the learning of new classes without forgetting old ones. It uses a memory-efficient strategy by retaining exemplars from previous classes, achieving competitive results in continual learning tasks. However, iCaRL's reliance on stored exemplars leads to scalability issues as the number of classes grows, and its performance degrades with limited memory.

In quantum incremental learning, various studies explore methods to reduce catastrophic forgetting in VQCs. Jiang et al. [20] highlight that quantum classifiers, like classical neural networks, suffer from forgetting when their parameters are updated during training on sequential tasks. The authors examine how task similarity affects forgetting, finding that tasks with permuted inputs and fundamentally different tasks degrade performance on previously learned tasks. They suggest a mitigation strategy that uses local geometrical information in the loss function landscape tailored to specific scenarios. However, our proposed method focuses on a generalized quantum incremental learning model that does not rely on specific variations of quantum classifier models.

Another study [33] introduces a hybrid quantum-classical approach for binary classification in an incremental learning context. This method combines VQCs with classical techniques to tackle the challenge of catastrophic forgetting. It aims to maintain classification accuracy over time while adapting to new data without losing previously acquired knowledge. Although this approach demonstrates promising results compared to classical methods, it primarily focuses on classical data. In contrast, our proposed model operates on quantum data, which is generated by transforming classical data through quantum models, offering a novel strategy to address catastrophic forgetting.

Other quantum-based strategies [17] that focus on exemplar memory and continual data uploading [34] fail to effectively address the problem of catastrophic forgetting and lack an effective incremental learning approach (see Section I). Consequently, research in QML for continual data settings often relies on transfer learning or training models from scratch. These methods are resource-intensive and yield poor results with limited quantum hardware resources. In contrast, our approach uses only a few samples from old classes and a minimal number of qubits to run circuits. This strategy overcomes the limitations of existing quantum incremental learning methods and outperforms those that use VQCs, as it optimizes both the computation model and data processing within a quantum framework.

III. Methodology

Our proposed quantum incremental learning model employs a decoupled architecture, consisting of a feature extractor and a classifier. The feature extractor is based on a quantum autoencoder, trained using a contrastive learning loss function. As illustrated in Figure 2, the classifier is a Variational Quantum Classifier (VQC) designed to handle multiple classes incrementally. Each VQC is responsible for learning a specific subset of classes during each iteration of the incremental training process. The classifier utilizes a combined loss function that incorporates cross-entropy and distillation loss to preserve the knowledge of previously learned classes.

After the initial training of the feature extractor on the first incremental data batch, the output logits from the feature extractor are used to train the initial VQC. During the incremental training process at each step, the loss function is calculated using both the old logits, generated by the previous VQCs and the new logits from the current VQC. This approach ensures that the model retains the knowledge acquired from earlier classes while learning new ones. We use a classical exemplar memory to store a sorted set of representative samples from previous classes, which are then used with the old VQCs to compute the old logits. This exemplar memory is updated after each incremental training step to maintain the most representative images (see Section III-B).

A. Quantum Model

Architecture. The decoupled quantum incremental learning model comprises two main components: the feature extractor and the classifiers. The feature extractor utilizes a Quantum Autoencoder (QAE) architecture [35], augmented with a quantum data augmentation module [36] as its head and a quantum projection head as its tail. This architecture transforms classical image data into quantum representations, producing a reduced-dimensional feature vector optimized using supervised contrastive loss [37].

$$\text{DA} = \bigotimes_{i=1}^{n} \left(SX \cdot R_Z \left(\frac{\pi}{2} \right) \right)^2 \cdot \prod_{k=1}^{n-1} \text{CNOT}(q_k, q_{k+1}) \tag{5}$$

The quantum data augmentation module (see Equation 5) encodes image pixel values into 6 qubits using $R_Z(\theta)$ gates, where θ corresponds to pixel values. Initially set to the $|0\rangle$ state, these qubits undergo transformations that prepare the data for dimensionality reduction by the quantum encoder. The encoder compresses an 8-qubit input state through alternating $R_Y(\theta)$ rotations and CX entanglements. 2 trash states are discarded in the bottleneck layer, leaving 6 latent qubits representing core features.

$$\text{QAE} = AmpEnc \cdot \text{Barrier} \cdot (H(q_{n+1}) \cdot \text{CSWAPs} \cdot H(q_{n+1})) \tag{6}$$

$$\text{AmpEnc} = R_Y(\theta) \cdot \text{CNOTs} \cdot R_Y(\theta') \tag{7}$$

A simple quantum neural network projection head outputs a 64-unit feature vector, reducing complexity during classification. All components—the quantum autoencoder, augmentation module, and projection head—are trained with supervised contrastive loss, defined as:

$$L_S(\omega) = -\sum_{\beta \in K} \log \left(\frac{\exp(Z_i \cdot Z_j / T)}{\sum_{n \in S} \exp(Z_i \cdot Z_n / T)} \right) \tag{8}$$

Here, Z_i and Z_j are encoded feature states, T is a temperature parameter, and the loss clusters similar data while separating dissimilar ones. Optimization is performed using the COBYLA algorithm [38], which refines objectives without

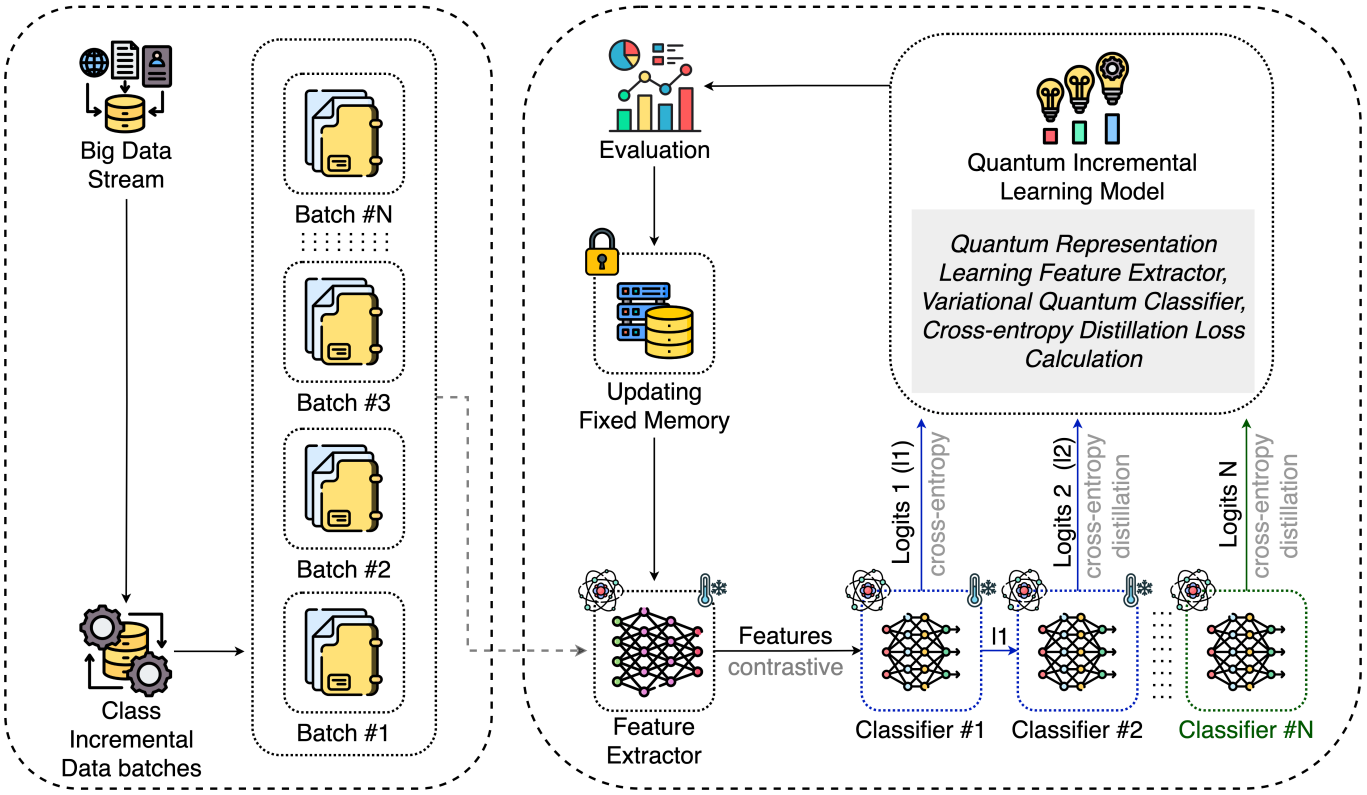

Fig. 2. Our Quantum Incremental Learning Model: Consists of a Quantum Autoencoder (QAE)-based feature extractor trained with supervised contrastive loss, frozen after initial training. A set of Variational Quantum Classifier (VQC) classifiers is employed, where the initial VQC is trained with cross-entropy loss. Subsequent VQCs incrementally learn new classes using logits from previous VQCs, leveraging cross-entropy and distillation losses, with support for a fixed memory size.

requiring gradients, effectively managing quantum parameter tuning and constraints.

The VQCs [39], [40] serve as the classification component in the incremental learning framework. Each VQC processes classical feature vectors from the pre-trained feature extractor, transforming them into quantum data to generate a probability distribution over the current set of classes. These classifiers are incrementally trained using a combination of cross-entropy and distillation losses. The VQCs utilize the SU2 variational circuit (see Equation 9), a hardware-efficient architecture employing 6 qubits to process the 64-dimensional feature vectors produced by the feature extractor. The initial VQC is trained using the cross-entropy loss function (see Equation 11). However, all VQCs minimize the loss using the COBYLA optimizer.

$$\text{SU2} = R_Y R_Z(\theta) \cdot \text{CNOTs}, \tag{9}$$

As shown in Figure 2, the feature extractor's architecture and weights remain frozen after initial training, allowing compatibility with any architecture. During incremental learning, distillation loss is computed using old VQC logits and exemplar memory samples. Previous VQCs, responsible for retaining earlier class knowledge, are discarded after training, leaving only the latest VQC for the model's final evaluation.

Cross-Entropy Distillation Loss. The cross-entropy distillation loss integrates two key components: a distillation loss [41], which helps retain knowledge from previously learned classes, and a multi-class cross-entropy loss, which enables the classification of newly introduced classes. The distillation loss is specifically applied to the earlier stages' Variational Quantum Classifiers (VQCs), which are trained on prior classes, while the multi-class cross-entropy loss is utilized across all VQCs' logits, encompassing the entire incremental learning process. According to [20], the formulation of the combined cross-entropy distillation loss function $L(\omega)$ is as follows:

$$L(\omega) = L_C(\omega) + \sum_{f=1}^{F} L_{Df}(\omega) \tag{10}$$

where $L_C(\omega)$ denotes the cross-entropy loss affecting both old and new classes, and $L_{Df}(\omega)$ represents the distillation loss linked to each VQC f. Here, F indicates the total count of VQCs relevant to the previously learned classes. The cross-entropy loss $L_C(\omega)$ is defined by:

$$L_C(\omega) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} p_{ij} \log q_{ij} \tag{11}$$

where p_{ij} is the ground truth for sample i in class j, q_{ij} is the VQC output for sample i in class j, and N and C represent the number of samples and classes, respectively. For the distillation loss $L_D(\omega)$, the expression is given by:

$$L_D(\omega) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij}^{dst} \log q_{ij}^{dst} \tag{12}$$

where p_{ij}^{dst} and q_{ij}^{dst} are modified versions of p_{ij} and q_{ij}, respectively, scaled by raising them to the power of $1/T$, with T being the distillation parameter. For $T = 1$, the loss is predominantly influenced by the class with the highest score. However, when $T > 1$, other classes contribute more significantly to the loss, which helps the network in learning finer distinctions between classes. Based on our studies, we set $T = 2$ throughout the research.

B. Classical Exemplar Memory

One of the critical components enabling knowledge distillation in our model is the classical exemplar memory, which stores new classes used to train the model at each iteration. In this study, we utilize an exemplar memory with a fixed capacity and a fixed number of samples. The proposed training dataset, as explained in Section III-C, contains the old classes from the previous $i-1$ iterations in the ith iteration of incremental training. For the first setup, when storing new classes, each class is allocated n samples, where n can be determined by $n = \lfloor \frac{K}{c} \rfloor$ with c representing the number of classes stored in memory and K indicating the total memory capacity. The second setup, which uses a fixed number of samples, simply stores a constant number of exemplars per class.

This exemplar memory addresses two main functions: selecting samples for each class before storing them in memory and updating the memory.

Selecting new samples. To select new samples, we use the herding selection method [42], which ranks the most representative samples from the training dataset. Herding selection involves creating an exemplar memory by choosing a subset of data points that best represent the distribution of the entire dataset. This process iteratively selects data points x_i that minimize the discrepancy between the empirical mean $\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ and the true mean $\mathbb{E}[\phi(x)]$, where $\phi(x)$ is a feature map. This selection method was chosen based on studies [43], [44], [45], and [46].

Updating memory. Updating the memory involves removing leftover samples. After applying the herding selection method, samples at the end of the list are discarded to ensure memory capacity constraints are maintained.

C. Dataset Selection and Preparation

Selection. This study utilizes four datasets: MNIST [47], FMNIST [48], KMNIST [49], and CIFAR10 [50], each containing ten classes. Due to quantum hardware constraints, only a subset of classes and samples from each dataset were selected for the experiments. Furthermore, images were resized to a 16×16 pixel resolution to accommodate these limitations. A random sampling method was employed for selecting classes and data samples.

- **Class Selection:** Each dataset contains ten classes. During the random selection process, n classes (where $1 < n < 10$) were initially selected and used for the experiments, with this procedure repeated five times. Each iteration involved discarding previously selected classes and choosing new ones. Due to quantum hardware constraints, not all possible class combinations were evaluated to prevent bias.
- **Data Sample Selection:** MNIST, FMNIST, and CIFAR10 each contain 60,000 samples, while KMNIST contains 70,000 samples. Given the high computational cost of processing all these samples on quantum hardware, a smaller, balanced subset was selected for the experiments. The selection process ensured that the class distribution remained balanced.

Preparation. Due to the lack of established benchmarks for class-incremental learning, we follow the standard approach [32], [51], which involves dividing the classes of a traditional multi-class dataset into incremental batches. The process begins by selecting a small subset of classes (at least two) and creating a batch with corresponding data and labels. This batch is then split into training and testing sets. In subsequent iterations, new classes are gradually introduced and combined with previously encountered ones. Random splitting is repeated to simulate a scenario where the model incrementally learns new classes over time. By maintaining both training and testing batches, this method enables the evaluation of the model's ability to retain prior knowledge while learning new concepts.

D. Incremental Learning

As illustrated in Figure 2, our quantum incremental learning approach consists of two main steps for each incremental iteration: the incremental step training process and exemplar memory updating.

Incremental Step Training Process. The quantum autoencoder feature extractor requires initial training using supervised contrastive loss. After this initial training stage, the feature extractor is frozen and not further trained in subsequent steps due to quantum resource limitations. The trained quantum autoencoder generates feature vectors for the classifier, which is a Variational Quantum Classifier (VQC). The VQC utilizes these feature vectors and is trained using the cross-entropy distillation loss. During this training, the VQC incorporates old logits, calculated using previous VQCs, for samples retrieved from the exemplar memory in the cross-entropy distillation loss. To manage quantum resource constraints, this study discards previous VQCs after training instead of maintaining them, while still leveraging their logits for knowledge distillation.

Exemplar Memory Updating. At the end of each incremental step, the exemplar memory is updated with new classes using the herding selection method. In this study, memory updating is performed in two different ways:

- **Dynamic Memory Capacity:** The exemplar memory capacity increases as new classes are added while maintaining a fixed number of samples per class.
- **Fixed Memory Capacity:** The total memory size remains fixed, which requires reducing the number of samples per class as new classes are introduced.

Algorithm 1 Our Quantum Incremental Learning Algorithm

Input: Data $\mathbf{X} \in \mathbb{R}^{N\times d}$, Labels $\mathbf{Y} \in \mathbb{R}^{N\times C}$, Parameters $\theta_{\text{FM}}, \theta_{\text{DA}}, \theta_{\text{AE}}, \theta_{\text{VQC}}$, Memory $\mathbf{M}$, Old VQCs $\{\theta_{\text{VQC},0}, \ldots, \theta_{\text{VQC},b-1}\}$
Output: Updated $\theta_{\text{VQC}}, \mathbf{M}$
Compute $\mathbf{Z}_0 = \text{FM}(\mathbf{X}_0; \theta_{\text{FM}})$
Augment: $\mathbf{Z}'_0 = \text{DA}(\mathbf{Z}_0; \theta_{\text{DA}})$
Encode: $\mathbf{Z}''_0 = \text{AE}(\mathbf{Z}'_0; \theta_{\text{AE}})$
Optimize: $\min_{\theta_{\text{AE}}} \mathcal{L}_{\text{SCL}}(\mathbf{Z}''_0, \mathbf{Y}_0)$
Predict: $\mathbf{Z}''_{p0} = \text{AE}(\text{DA}(\text{FM}(\mathbf{X}_0)))$
Train VQC: $\min_{\theta_{\text{VQC}}} \mathcal{L}_{\text{CE}}(\mathbf{Z}''_{p0}, \mathbf{Y}_0)$
Update old VQCs: $\theta_{\text{VQC},0} \leftarrow \theta_{\text{VQC}}$
for each batch $(\mathbf{X}_b, \mathbf{Y}_b)$, $b = 1 \ldots B$ **do**
  Initialize $\mathbf{Z}_{\text{mem}} = \emptyset$
  for $(\mathbf{M}_i, \mathbf{Y}_i) \in \mathbf{M}$ **do**
    Compute: $\mathbf{Z}_{\text{mem}} \cup \{\text{AE}(\text{DA}(\text{FM}(\mathbf{M}_i)))\}$
  end for
  Predict: $\mathbf{Z}''_{pb} = \text{AE}(\text{DA}(\text{FM}(\mathbf{X}_b)))$
  Initialize $\mathbf{L}_{\text{old}} = \emptyset$
  for $\theta_{\text{VQC},k} \in \{\theta_{\text{VQC},0}, \ldots, \theta_{\text{VQC},b-1}\}$ **do**
    Compute: $\mathbf{L}_{\text{old}} \cup \{\text{VQC}(\mathbf{Z}_{\text{mem}}; \theta_{\text{VQC},k})\}$
  end for
  Predict logits: $\mathbf{L}_{\text{new}} = \text{VQC}(\mathbf{Z}''_{pb}; \theta_{\text{VQC}})$
  Minimize: $\mathcal{L}_{\text{CDL}} = \mathcal{L}_{\text{CE}}(\mathbf{L}_{\text{new}}, \mathbf{Y}_b) + \alpha \sum_k \mathcal{L}_{\text{Distill}}(\mathbf{L}_{\text{old}})$
  Update $\mathbf{M} \leftarrow \text{update}(\mathbf{M}, (\mathbf{X}_b, \mathbf{Y}_b))$
  Update old VQCs: $\theta_{\text{VQC},b} \leftarrow \theta_{\text{VQC}}$
end for

E. Convergence Analysis

The convergence of the proposed quantum incremental learning model is demonstrated by analyzing the behavior of the combined loss function, Cross-Entropy Distillation loss, which consists of cross-entropy and distillation loss components. This section presents the underlying mathematical principles ensuring the model converges to a local minimum through the optimization of gradient descent. The goal is to show that, over time, the network's parameters stabilize, minimizing both the cross-entropy loss for new classes and the distillation loss for knowledge retention from previous classes.

The Cross-Entropy Distillation loss function is defined as a convex combination of the cross-entropy loss (L_C) and distillation loss (L_D), with a weighting parameter $\alpha \in [0, 1]$:

$$L(\omega) = \alpha L_C(\omega) + (1 - \alpha) L_D(\omega), \tag{13}$$

where ω represents the model parameters. The cross-entropy loss $L_C(\omega)$ captures the divergence between the true labels y and the predicted outputs $\hat{y}$ for newly introduced classes:

$$L_C(\omega) = -\sum_i y_i \log(\hat{y}_i). \tag{14}$$

The distillation loss $L_D(\omega)$ preserves information from previously learned classes by minimizing the divergence between the softened logits of the previous model y^o and the current model $\hat{y}^o$, with a temperature parameter T:

$$L_D(\omega) = -\sum_i \sigma\left(\frac{y_i^o}{T}\right) \log\left(\sigma\left(\frac{\hat{y}_i^o}{T}\right)\right), \tag{15}$$

where σ denotes the softmax function.

Gradient Descent and Assumptions. To ensure convergence, the model parameters are updated using gradient descent. The update rule is given by:

$$\omega_{t+1} = \omega_t - \eta \nabla L(\omega_t), \tag{16}$$

where η is the learning rate and $\nabla L(\omega_t)$ is the gradient of the loss function at iteration t. For the theoretical analysis of convergence, the following conditions are assumed:

1) **Differentiability:** Both $L_C(\omega)$ and $L_D(\omega)$ are differentiable with respect to the model parameters ω.
2) **Lipschitz Continuous Gradients:** The gradients $\nabla L(\omega)$, $\nabla L_C(\omega)$, and $\nabla L_D(\omega)$ are Lipschitz continuous, meaning there exists a constant $L > 0$ such that for any ω_1, ω_2,

$$\|\nabla L(\omega_1) - \nabla L(\omega_2)\| \leq L\|\omega_1 - \omega_2\|. \tag{17}$$

These assumptions align with standard convergence results in gradient-based optimization.

The behavior of the loss function under gradient descent is analyzed using Taylor's expansion. The loss at the next iteration can be approximated as:

$$\begin{aligned} L(\omega_{t+1}) \approx L(\omega_t) + \nabla L(\omega_t)^\top (\omega_{t+1} - \omega_t) \\ + \frac{1}{2}(\omega_{t+1} - \omega_t)^\top H(\omega_t)(\omega_{t+1} - \omega_t), \end{aligned} \tag{18}$$

where $H(\omega_t)$ is the Hessian matrix of the loss function at iteration t.

Substituting the gradient descent update rule into this expression gives:

$$\begin{aligned} L(\omega_{t+1}) \approx L(\omega_t) - \eta \|\nabla L(\omega_t)\|^2 \\ + \frac{1}{2}\eta^2 \nabla L(\omega_t)^\top H(\omega_t) \nabla L(\omega_t). \end{aligned} \tag{19}$$

The loss decreases at each iteration by selecting a sufficiently small learning rate η, provided the gradients are Lipschitz continuous. Using the descent lemma:

$$L(\omega_{t+1}) \leq L(\omega_t) - \eta \left(1 - \frac{1}{2}\eta L\right) \|\nabla L(\omega_t)\|^2. \tag{20}$$

This result guarantees that if $0 < \eta < \frac{2}{L}$, the loss function decreases as long as $\nabla L(\omega_t) \neq 0$. The gradient norm diminishes as iterations progress, leading to smaller updates to ω_t. Consequently, the sequence of model parameters $\{\omega_t\}$ converges to a critical point ω^*, where $\nabla L(\omega^*) = 0$, indicating a local minimum.

However, achieving practical convergence presents challenges due to the optimization complexities involved in balancing cross-entropy loss (L_C) and distillation loss (L_D), which necessitate careful tuning of the weighting parameter α. Difficulties in selecting an optimal learning rate can be mitigated by employing adaptive learning rate methods. Furthermore, quantum noise and hardware limitations in real-world quantum systems require the application of error mitigation techniques to ensure reliable training.

The convergence of the model under the Cross-Entropy Distillation Loss function is thus ensured by appropriate choices of the learning rate η, the weighting parameter α, and careful

consideration of the loss components. This balance enables effective incremental learning while mitigating catastrophic forgetting.

F. Space and Gate Complexity Analysis

In this theoretical analysis of space and gate complexity for the proposed quantum incremental learning model, based on the study by Cardoso et al. [52], we focus on the complexity of the incremental training process. The model incorporates a quantum autoencoder, quantum data augmentation, and a Variational Quantum Classifier (VQC). This analysis exclusively examines the space and gate complexity for each circuit execution. The primary focus is on key quantum circuit operations performed on IBM quantum systems [53] during a single incremental learning step, as outlined in Algorithm 1. Notably, the analysis does not account for quantum errors.

1) Space Complexity: The classical-to-quantum data transformation operation primarily determines the space complexity in Algorithm 1. Specifically, the data encoding circuit in Algorithm 1, line 3, is responsible for loading the training data into qubit amplitudes without employing a kernel transformation. Assuming a system with a Hilbert space dimension corresponding to an N-feature space, the preparation of qubits into a superposition state can be achieved using $O(\log_2(N))$ qubits.

2) Gate Complexity: The gate or time complexity of Algorithm 1 can be divided into three main steps: state preparation, circuit execution, and readout.

State Preparation: The cost of state preparation is determined by the circuit in Algorithm 1, line 3, which loads the training data into qubit amplitudes without applying a kernel transformation. As a result, the upper bound of quantum operations for state preparation is $O(N)$, corresponding to the preparation of an N-dimensional superposition [54].

Circuit Execution: For circuit execution, we analyze the qubit operations required to transform qubits from the previously prepared state within the quantum system. The cost of gate operations is influenced by the data augmentation circuit in Algorithm 1, line 4, the autoencoder circuit in Algorithm 1, line 5, and the Variational Quantum Classifier (VQC) circuit in Algorithm 1, line 8. The data augmentation circuit involves applying single-qubit gates (SX and $RZ(\theta)$) to each qubit, and potentially introducing entanglement through CNOT gates between adjacent qubits. The SX gate, which is the square root of the Pauli-X gate, performs a partial flip of the qubit state, and applying it twice is equivalent to the full Pauli-X gate. The $RZ(\theta)$ gate applies a rotation around the Z-axis of the Bloch sphere by an angle θ, modifying the phase of the qubit's state.

The SX gate transforms the qubit state $|\psi\rangle$ to $|\psi'\rangle = SX|\psi\rangle$. The $RZ(\theta)$ gate modifies the phase of the quantum state to $|\psi''\rangle = RZ(\theta)|\psi\rangle$. When both gates are applied sequentially, the resulting transformation is $|\psi_{\text{final}}\rangle = RZ(\theta)SX|\psi\rangle$. Assuming single-qubit operations are constant time $O(1)$ [55], the total cost is primarily determined by the CNOT gate operations. Considering widely used gate decomposition methods, such as QR Decomposition [56], which requires $O(N^2 \log_2^3(N))$ elementary operations, and cosine-sine decomposition (CSD) [57], which requires $O(N^2 - 2N)$ CNOT operations and $O(N^2)$ elementary single-qubit operations, the complexity of this circuit can be approximated to $O(N^2)$.

The quantum autoencoder circuit integrates a variational circuit with alternating layers of 8 single-qubit $RY(\theta)$ and entangling CX (CNOT) gates. The $RY(\theta)$ gate performs a rotation around the Y-axis of the Bloch sphere by an angle θ, while the CX gate, or Controlled-NOT (CNOT) gate, is a two-qubit entangling gate that flips the state of the target qubit if the control qubit is in the $|1\rangle$ state.

Applying $RY(\theta)$ to a qubit transforms its state to $|\psi'\rangle = RY(\theta)|\psi\rangle$. The CX gate operation can be described as $CX(|c\rangle \otimes |t\rangle) = |c\rangle \otimes X^c|t\rangle$, where $|c\rangle$ is the control qubit and $|t\rangle$ is the target qubit. If $c = 1$, the X (Pauli-X) gate is applied to the target qubit, flipping its state; if $c = 0$, the target qubit remains unchanged. Applying a single-qubit $RY(\theta)$ gate followed by an entangling CX gate results in the transformation $|\psi_{\text{final}}\rangle = CX\,(RY(\theta) \otimes I)\,|\psi\rangle$. This is followed by a swap test involving controlled swap gates. Assuming single-qubit operations take constant time $O(1)$, and considering 5 replicate layers, the total CNOT complexity is $O(5N^2 - 10N)$, which can be approximated as $O(N^2)$. Additionally, each single SWAP gate operation requires 3 CNOT operations, contributing $O(3N^2 - 6N)$ complexity, also approximated as $O(N^2)$.

The variational quantum classifier circuit alternates between applying single-qubit $RY(\theta)$ and $RZ(\theta)$ gates on individual qubits, followed by entangling operations using CX (CNOT) gates between adjacent qubits. The VQC's first layer applies $RY(\theta)$ and $RZ(\theta)$ gates in parallel to each qubit. The subsequent layer introduces CNOT gates between adjacent qubits. This process repeats for the remaining layers, alternating between single-qubit gates and entangling operations. Assuming that single-qubit operations require constant time $O(1)$, the total CNOT gate complexity for a VQC with 3 replicated layers is $O(3N^2 - 6N)$.

The incremental learning process described in Algorithm 1 (line 10) is executed m times. Consequently, the computational complexity of the VQC component must be scaled by m to account for the total complexity of the incremental training process. As this is a class-incremental learning approach, the number of iterations depends on the total number of classes and the number of classes processed in each iteration.

G. Experimental Setup

This experimental setup optimized performance across classical systems and quantum simulators. The classical systems were equipped with a 16 GiB NVIDIA T4 Tensor Core GPU to support machine learning inference and graphics-intensive tasks.

For the Fixed Memory quantum experiment detailed in Section III-I, the Amazon Braket IonQ device, Harmony [58], [59], was employed. Harmony is an 11-qubit quantum processor based on a universal gate-model ion-trap architecture. Its Universal Gate Set includes quantum gates such as Pauli-X, Y, Z, and CNOT, which are capable of approximating any

unitary transformation to the desired precision. In this ion-trap system, qubits are represented by the electronic states of trapped ions, which interact via the Coulomb force [60]. These ions are isolated and trapped to preserve quantum coherence, while laser beams manipulate their internal energy levels for single-qubit operations.

The IBM Qiskit library [61] was used for experiment implementation and execution. Qiskit circuits were converted for compatibility with the IonQ system and executed on the IonQ-Harmony device via the Amazon Braket Provider [62].

H. Evaluation on MNIST, KMNIST, and FMNIST

We conducted three categories of experiments using the MNIST, FMNIST, and KMNIST datasets.

In the first experiment, we employed a fixed classical exemplar memory with an imbalanced number of samples per class. In the second experiment, we evaluated methods under a setup where a fixed number of samples was retained for each previously encountered class. This configuration allowed the total memory size to grow incrementally as new classes were added to the classical exemplar memory unit. Given the absence of a standard benchmarking framework for quantum incremental learning, we compared our model, INC-VQC, against several baseline methods: quantum-adapted iCaRL (iCaRL-VQC) [32], quantum transfer learning (TL-VQC), and quantum supervised contrastive learning (SCL-VQC) [63]. In real-world scenarios, initial classes might be trained using different models, a consideration beyond the scope of this research. For consistency, in each benchmark method, the initial increment involving two classes was trained using the same quantum autoencoder with supervised contrastive loss and a variational quantum classifier (VQC) with cross-entropy loss. Lastly, as detailed in Section III-K, we conducted a sensitivity analysis to investigate the influence of various quantum components on classification accuracy in our approach.

I. Fixed Memory

We evaluated two experimental setups using the original class order, with incremental steps of 1 and 2 classes. These experiments were conducted with a fixed classical exemplar memory size, where the number of samples per class varied. To ensure a fair comparison across methods, the class order remained consistent throughout all evaluations.

The results are summarized in Tables I and II. These tables present data for increments of 2, 3, and 10 classes, where the number of classes added per increment is 1, as well as for increments of 2, 4, and 10 classes, where the number of classes added per increment is 2. Figure 3 provides a visual representation of the performance across incremental steps for 1 and 2 classes. The 'Upper-Bound' performance, depicted as a large cross (X) at the final step in Figures 3, 4, 5, 6, and 7, represents the accuracy of a non-incremental VQC trained on all classes and their corresponding training samples.

In both the 1-class-per-iteration and 2-class-per-iteration setups, our model (INC-VQC) demonstrated consistent and robust performance across the MNIST, FMNIST, and KMNIST datasets. In the 1-class-per-iteration setup, INC-VQC achieved

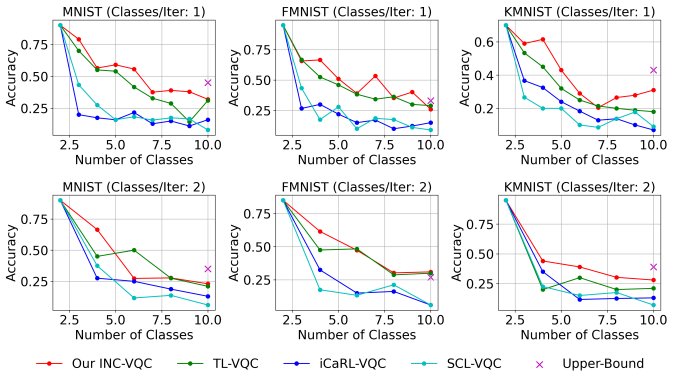

Fig. 3. Model accuracy on MNIST, FMNIST, and KMNIST under a fixed memory setting. The first row shows the accuracy of INC-VQC, SCL-VQC, iCaRL-VQC, and TL-VQC on the three datasets when the number of classes per incremental training step is 1. The second row presents the accuracy for the same datasets and methods when the number of classes per incremental training step is 2

90% accuracy for 2 classes and 32% accuracy for 10 classes on MNIST, significantly outperforming iCaRL-VQC (10%) and SCL-VQC (14%), which showed rapid performance degradation as the number of classes increased. On FMNIST and KMNIST, INC-VQC also outperformed competing methods, achieving 26% and 31% accuracy for 10 classes, respectively.

In the 2-classes-per-iteration setup, INC-VQC continued to maintain strong performance, achieving 90% accuracy for 2 classes and 23% accuracy for 10 classes on MNIST. On FMNIST and KMNIST, its performance stabilized at 31% and 28% accuracy for 10 classes, respectively, further showcasing its superiority over baseline methods.

TABLE I
TEST ACCURACIES WITH FIXED MEMORY AND 1 CLASS PER INCREMENT

Dataset	MNIST			FMNIST			KMNIST		
Classes	**2**	**3**	**10**	**2**	**3**	**10**	**2**	**3**	**10**
Test Accuracy (%)									
Our INC-VQC	90	79	32	95	66	26	70	59	31
TL-VQC	90	70	26	95	66	29	70	53	18
iCaRL-VQC	90	20	10	95	26	15	70	37	7
SCL-VQC	90	43	14	95	43	9	70	27	9

Our INC-VQC achieved accuracy comparable to TL-VQC, which retrained a single VQC at each incremental step. This performance underscores the effectiveness of INC-VQC's representation learning, driven by its use of a quantum autoencoder feature extractor and Variational Quantum Classifiers (VQCs) trained with cross-entropy distillation loss. The inclusion of a classical exemplar memory, which retained representative samples of previously learned classes, further contributed to preserving knowledge across incremental steps. While the Upper-Bound method achieved higher accuracy (e.g., 45% on MNIST for 10 classes in the 1-class-per-iteration setup), it relied on non-incremental access to all data, making it impractical for real-world incremental learning scenarios.

J. Fixed Number of Samples

Similar to the fixed memory experiment (see Section III-I), we evaluated two experimental splits using the original class

TABLE II
TEST ACCURACIES WITH FIXED MEMORY AND 2 CLASSES PER INCREMENT.

Dataset	MNIST			FMNIST			KMNIST		
Classes	**2**	**4**	**10**	**2**	**4**	**10**	**2**	**4**	**10**
Test Accuracy (%)									
Our INC-VQC	90	83	23	85	69	31	95	76	28
TL-VQC	90	80	21	85	54	30	95	51	21
iCaRL-VQC	90	30	10	85	21	12	95	34	12
SCL-VQC	90	57	14	85	17	13	95	24	15

order with incremental steps of 1 and 2 classes. In this setup, the experiments were conducted with a fixed number of samples per class, ensuring a constant sample count for each class. Unlike the fixed memory approach, the classical exemplar memory grew incrementally as new classes were added during each step.

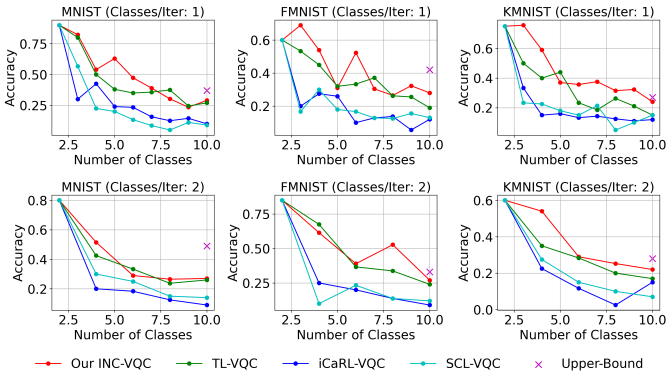

Fig. 4. Model accuracy on MNIST, FMNIST, and KMNIST under a fixed number of samples setting. The first row shows the accuracy of INC-VQC, SCL-VQC, iCaRL-VQC, and TL-VQC on the three datasets when the number of classes per incremental training step is 1. The second row presents the accuracy for the same datasets and methods when the number of classes per incremental training step is 2

In the fixed number of samples setup, our INC-VQC model demonstrated robust performance across all datasets with consistent accuracy improvements over other methods (see Figure 4. For the MNIST dataset, INC-VQC achieved 90% accuracy for 2 classes and maintained 29.0% for 10 classes in the 1-class-per-iteration setup, outperforming TL-VQC (27.0%), iCaRL-VQC (10.0%), and SCL-VQC (9.0%). In the 2-classes-per-iteration setup, INC-VQC achieved 80.0% accuracy for 2 classes and 27.0% for 10 classes, remaining superior to TL-VQC (26.0%), iCaRL-VQC (9.0%), and SCL-VQC (14.0%). Similar trends were observed across the FMNIST and KMNIST datasets in the 1-class-per-iteration setup. On FMNIST, INC-VQC consistently outperformed other methods, achieving 28.0% accuracy for 10 classes in the fixed sample setup, compared to TL-VQC (19.0%), iCaRL-VQC (12.0%), and SCL-VQC (13.0%). Similarly, on KMNIST, INC-VQC achieved 24.0% accuracy for 10 classes, outperforming TL-VQC (15.0%), iCaRL-VQC (12.0%), and SCL-VQC (15.0%). These performance trends were consistent in the 2-classes-per-iteration setup for both FMNIST and KMNIST datasets as well.

However, the accuracies of the fixed samples and fixed memory setups excel in different scenarios, as shown in Figure 3 and Figure 4, making it difficult to identify a universally superior setting. Given the limited size of the experimental datasets, drawing definitive conclusions about the relative performance of these two approaches is challenging. Furthermore, the observed differences in test accuracies may not translate to larger datasets or more complex tasks, emphasizing the need for further investigation across diverse conditions.

K. Sensitivity Analysis

In this study, we evaluated the impact of different components of our model on classification accuracy, with a specific focus on the classifier component. Specifically, we examined how variations in the structure of the variational circuits and the number of quantum gates within these circuits affected the performance of the Variational Quantum Classifier (VQC). The VQC utilized a variational circuit to iteratively optimize its parameters, enabling it to learn complex data patterns, analogous to weight optimization in classical neural networks. In this experiment, we examined how increasing the number of quantum gates in the variational circuit—analogous to adding hidden layers in a deep neural network—impacted the model's accuracy. The experiment was conducted using a fixed memory setting with a 1-class-per-iteration incremental learning setup.

As described in Equation 9, the SU2 variational circuit used in this study comprised layers of R_Y and R_Z rotation gates applied to each qubit, followed by $CNOT$ gates that created entanglement between adjacent qubits. The initial configuration of the rotational gates in the classifier consisted of four repetitions, which were progressively increased to ten repetitions to evaluate their impact on the model's performance.

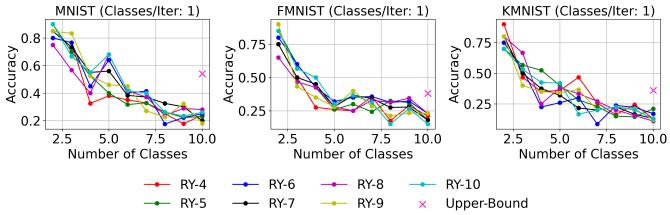

Fig. 5. Model Accuracy on MNIST, FMNIST, and KMNIST vs. Rotational Gates: This figure illustrates the accuracy of INC-VQC on the three datasets using an SU2 variational circuit in the VQC, configured with 4 to 10 layers of rotational gates. The results are presented for a dynamic memory setting, where the number of classes per incremental training step is set to 1

Our results demonstrated (see Figure 5) that while increasing the number of rotational gates improved accuracy for smaller class increments (1-class-per-increment), the improvements plateaued for larger class counts. On the MNIST dataset, increasing the repetitions from 4 to 10 resulted in a marginal accuracy increase for 10 classes, from 24% to 26%. A similar trend was observed for FMNIST, where accuracy for 10 classes improved slightly from 21% (4 repetitions) to 23% (10 repetitions). However, on KMNIST, the accuracy for 10 classes remained relatively stable, ranging from 13% to 13%. These results suggested that while additional rotational gates could help in learning more complex patterns, the benefits

diminished as the circuit depth increased. This diminishing return was likely due to noise and increased circuit complexity, which adversely affected performance in quantum systems.

Increasing the repetitions of R_Y rotation layers in our model's classifier resulted in minor accuracy improvements, likely due to the increased circuit complexity. To better understand this effect, we further explored the impact of various variational circuit architectures on the classifier's performance. Specifically, we evaluated the SU2 variational circuit architecture (see Equation 9) used throughout this study, alongside the Amplitude Encoding Circuit (AmpEnc/ AE) (see Equation 7) and the Local Interactions Circuit (TL) [64].

The TL circuit (see Equation 21) features alternating layers of R_Y and R_Z rotations, interleaved with $CNOT$ gates that create entanglement between adjacent qubits in a linear topology. This configuration can be expressed as:

$$\text{TL} = R_Y R_Z(\theta) \cdot \text{CNOTs} \cdot R_Y R_Z(\theta'), \tag{21}$$

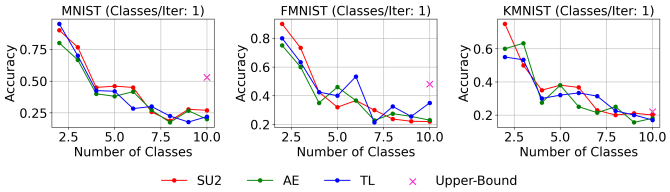

Fig. 6. Model Accuracy on MNIST, FMNIST, and KMNIST vs. Variational Circuit Architecture: This figure illustrates the accuracy of INC-VQC on the three datasets using SU2, Amplitude Encoding (AE), and TL variational circuits in the VQC. The results are presented for a dynamic memory setting, where the number of classes per incremental training step is set to 1.

Figure 6 revealed that the TL circuit slightly outperformed the SU2 and Amplitude Encoding circuits across most datasets. On MNIST, the TL circuit achieved 22% accuracy for 10 classes, compared to 27% for SU2 and 20% for Amplitude Encoding. Similarly, for FMNIST, the TL circuit achieved 35% for 10 classes, outperforming SU2 (22%) and Amplitude Encoding (23%). On KMNIST, the TL circuit achieved comparable performance, with 17% accuracy for 10 classes, slightly lower than SU2 (20%) and on par with Amplitude Encoding (18%). These results demonstrated that while the TL circuit provided consistent accuracy improvements for smaller class increments, its benefits diminished for larger class sizes.

L. Evaluation on CIFAR10

The CIFAR10 dataset presented a challenging benchmark for incremental learning due to its diverse image classes and inherent complexity. We evaluated our model using two incremental setups: 1-class-per-iteration and 2-classes-per-iteration. The experiments leveraged a fixed sample size per class, ensuring a consistent basis for comparison across methods (SCL-VQC, iCaRL-VQC, and TL-VQC).

In the 1-class-per-iteration setup, INC-VQC demonstrated (see Figure 7) superior accuracy compared to alternative methods, especially for lower numbers of classes. Our INC-VQC achieved 65% accuracy for 2 classes and 16% for 10 classes, outperforming TL-VQC (9%), iCaRL-VQC (7%),

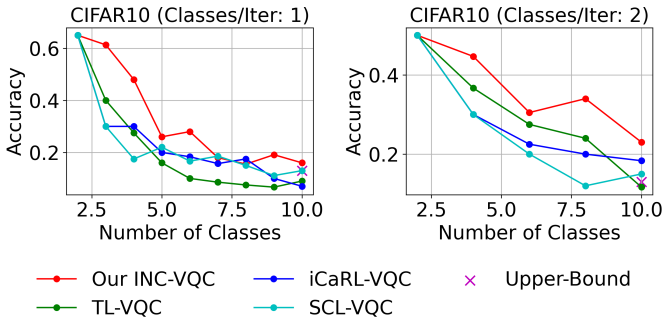

Fig. 7. Model Accuracy on CIFAR-10 under a Dynamic Memory Setting: The first plot displays the accuracy of four methods on CIFAR-10 when the number of classes per incremental training step is 1. The second plot shows the accuracy for the same methods and dataset when the number of classes per incremental training step is 2

and SCL-VQC (13%). Notably, the Upper-Bound method, which required non-incremental access to all data, achieved a comparable 13% for 10 classes, further validating INC-VQC's efficiency under incremental constraints. For the 2-classes-per-iteration setup, INC-VQC maintained a competitive edge, achieving 50% accuracy for 2 classes and 23% for 10 classes. In comparison, TL-VQC scored 11.67%, iCaRL-VQC 18.33%, and SCL-VQC 15% for 10 classes. The Upper-Bound method achieved 16.67% accuracy, underscoring the challenge of maintaining performance with limited incremental data.

IV. Conclusion

In this paper, we proposed a quantum incremental learning framework that effectively addresses the challenges of catastrophic forgetting and knowledge retention in VQCs. By integrating a decoupled representation learning model with supervised contrastive loss and a cross-entropy distillation loss, the approach demonstrates the capability to incrementally learn new classes while preserving knowledge of previously learned classes. The quantum autoencoder's efficient feature extraction and exemplar memory's representative sampling played a crucial role in achieving state-of-the-art performance across benchmarks.

Our experimental evaluation of MNIST, FMNIST, KMNIST, and CIFAR-10 datasets highlights the robustness of the proposed model (INC-VQC). It consistently outperformed baseline methods, including SCL-VQC, iCaRL-VQC, and TL-VQC, particularly in scenarios with fixed memory constraints or dynamically growing exemplar memory. Sensitivity analysis further validated the effectiveness of the selected variational circuit architecture and its rotational gate choices, demonstrating an optimal balance between accuracy and complexity.

The model's limitations on complex datasets like CIFAR-10 highlight challenges in scalability and quantum hardware constraints, emphasizing the need for improved circuit designs and hardware advancements. The proposed framework bridges classical and quantum machine learning, providing a scalable solution for class-incremental learning. Future directions include error mitigation, adaptive memory management, and extending to multi-task learning, enhancing real-world applicability.

REFERENCES

[1] D. Kudithipudi, M. Aguilar-Simon, J. Babb, M. Bazhenov, D. Blackiston, J. C. Bongard, A. P. Brna, S. C. Raja, N. Cheney, J. Clune, A. R. Daram, S. Fusi, P. Helfer, L. M. Kay, N. A. Ketz, Z. Kira, S. Kolouri, J. L. Krichmar, S. Kriegman, M. Levin, S. Madireddy, S. Manicka, A. Marjaninejad, B. L. McNaughton, R. Miikkulainen, Z. Navratilova, T. Pandit, A. Parker, P. K. Pilly, S. Risi, T. J. Sejnowski, A. Soltoggio, N. Soures, A. S. Tolias, D. Urbina-Meléndez, F. J. Valero-Cuevas, G. M. van de Ven, J. T. Vogelstein, F. Wang, R. Weiss, A. Yanguas-Gil, X. Zou, and H. T. Siegelmann, "Biological underpinnings for lifelong learning machines," *Nature Machine Intelligence*, vol. 4, pp. 196 – 210, 2022.

[2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[3] L. Wang, B. Lei, Q. Li, H. Su, J. Zhu, and Y. Zhong, "Triple-memory networks: A brain-inspired method for continual learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 1925–1934, 2022.

[4] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, no. 1, p. 4069, 2020.

[5] T.-C. Kao, K. Jensen, G. van de Ven, A. Bernacchia, and G. Hennequin, "Natural continual learning: success is a journey, not (just) a destination," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 28 067–28 079.

[6] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.

[7] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[8] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[9] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, 2016.

[11] C. Zhang, Z. Lu, L. Zhao, S. Xu, W. Li, K. Wang, J. Chen, Y. Wu, F. Jin, X. Zhu, Y. Gao, Z. Tan, Z. Cui, A. Zhang, N. Wang, Y. Zou, T. Li, F. Shen, J. Zhong, Z.-H. Bao, Z. Zhu, Z. Song, J. Deng, H. Dong, P. Zhang, W. Jiang, Z. Sun, P.-X. Shen, H.-P. Li, Q.-W. Guo, Z. Wang, J. Hao, H. Wang, D.-L. Deng, and C. Song, "Quantum continual learning on a programmable superconducting processor," *arXiv*, 2024.

[12] H. Xu and H. Situ, "Dynamic model structure adjustment to realize quantum continual learning based on quantum data," *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5280–5284, 2024.

[13] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[14] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, "Variational quantum algorithms for nonlinear problems," *Physical Review A*, vol. 101, no. 1, p. 010301, 2020.

[15] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, "Variational quantum algorithms for discovering hamiltonian spectra," *Physical Review A*, vol. 99, no. 6, p. 062304, 2019.

[16] M. Benedetti, M. Fiorentini, and M. Lubasch, "Hardware-efficient variational quantum algorithms for time evolution," *Physical Review Research*, vol. 3, no. 3, p. 033083, 2021.

[17] H. Situ, T. Lu, M. Pan, and L. Li, "Quantum continual learning of quantum data realizing knowledge backward transfer," *Physica A: Statistical Mechanics and its Applications*, vol. 620, p. 128779, 2023.

[18] H. Xu and H. Situ, "Dynamic model structure adjustment to realize quantum continual learning based on quantum data," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 5280–5284.

[19] Y. Kong, L. Liu, H. Chen, J. Kacprzyk, and D. Tao, "Overcoming catastrophic forgetting in continual learning by exploring eigenvalues of hessian matrix," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[20] W. Jiang, Z. Lu, and D.-L. Deng, "Quantum continual learning overcoming catastrophic forgetting," *Chinese Physics Letters*, vol. 39, no. 5, p. 050303, 2022.

[21] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, vol. 13. MIT Press, 2000.

[22] Y. Lu, K. Boukharouba, J. Boonært, A. Fleury, and S. Lecœuche, "Application of an incremental SVM algorithm for on-line human recognition from video surveillance using texture and color features," *Neurocomputing*, vol. 126, pp. 132–140, 2014.

[23] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, p. 1579–1619, Dec. 2005.

[24] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. Association for Computing Machinery, 2004, p. 116.

[25] P. Richtárik and M. Takác, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, vol. 156, pp. 433 – 484, 2012.

[26] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Good practice in large-scale learning for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 507–520, 2014.

[27] M. Sapienza, F. Cuzzolin, and P. H. Torr, "Learning discriminative space–time action parts from weakly labelled videos," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 30–47, 2014.

[28] U. Michieli and P. Zanuttigh, "Knowledge distillation for incremental learning in semantic segmentation," *Computer Vision and Image Understanding*, vol. 205, p. 103167, 2021.

[29] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 614–629.

[30] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," 2016.

[31] A. V. Terekhov, G. Montone, and J. K. O'Regan, "Knowledge transfer in deep block-modular neural networks," in *Proceedings of the 4th International Conference on Biomimetic and Biohybrid Systems - Volume 9222*, ser. Living Machines 2015. Springer-Verlag, 2015, pp. 268–279.

[32] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5533–5542.

[33] C. Loglisci, I. Diliso, and D. Malerba, "A hybrid quantum-classical framework for binary classification in online learning," *CEUR-WS*, 2023.

[34] M. Periyasamy, N. Meyer, C. Ufrecht, D. D. Scherer, A. Plinge, and C. Mutschler, "Incremental data-uploading for full-quantum classification," *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 31–37, 2022.

[35] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology*, vol. 2, no. 4, p. 045001, 2017.

[36] A. Chalumuri, R. Kune, S. Kannan, and B. S. Manoj, "Quantum–classical image processing for scene classification," *IEEE Sensors Letters*, vol. 6, no. 6, pp. 1–4, 2022.

[37] A. K. K. Don and I. Khalil, "Q-supcon: Quantum-enhanced supervised contrastive learning architecture within the representation learning framework," *ACM Transactions on Quantum Computing*, 2024.

[38] IBM, "COBYLA," 2023. [Online]. Available: https://docs.quantum.ibm.com/api/qiskit/0.26/qiskit.algorithms.optimizers.COBYLA

[39] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018.

[40] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.

[41] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[42] M. Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1121–1128.

[43] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Springer International Publishing, 2018, pp. 241–257.

[44] Y. Chen, T. Zang, Y. Zhang, Y. Zhou, L. Ouyang, and P. Yang, "Incremental learning for mobile encrypted traffic classification," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[45] E. Belouadah, A. Popescu, and I. Kanellos, "A comprehensive study of class incremental learning algorithms for visual tasks," *Neural Networks*, vol. 135, pp. 38–54, 2021.
[46] F. Mi, L. Kong, T. Lin, K. Yu, and B. Faltings, " Generalized Class Incremental Learning ," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, 2020, pp. 970–974.
[47] Li Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[48] "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017.
[49] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep learning for classical japanese literature," 2018.
[50] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009.
[51] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3420–3429.
[52] F. R. Cardoso, D. Y. Akamatsu, V. L. Campo Junior, E. I. Duzzioni, A. Jaramillo, and C. J. Villas-Boas, "Detailed account of complexity for implementation of circuit-based quantum algorithms," *Frontiers in Physics*, vol. 9, 2021.
[53] IBM, "IBM quantum," 2023. [Online]. Available: https://quantum-computing.ibm.com/
[54] IBM, "RawFeatureVector," 2023. [Online]. Available: https://docs.quantum.ibm.com/api/qiskit/0.19/qiskit.aqua.components.feature_maps.RawFeatureVector
[55] A. R. Brown and L. Susskind, "Complexity geometry of a single qubit," *Physical Review D*, vol. 100, no. 4, p. 046020, 2019.
[56] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins University Press, 2013.
[57] C. C. Paige and M. Wei, "History and generality of the CS decomposition," *Linear Algebra and its Applications*, vol. 208-209, pp. 303–326, 1994.
[58] Amazon, "Quantum cloud computing service - amazon braket - AWS." [Online]. Available: https://aws.amazon.com/braket/
[59] IonQ, "IonQ harmony," 2023. [Online]. Available: https://ionq.com/quantum-systems/harmony
[60] D. Kielpinski, C. R. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature*, vol. 417, pp. 709–711, 2002.
[61] M. Treinish, "Qiskit/qiskit-metapackage: Qiskit 0.44.0," 2023. [Online]. Available: https://zenodo.org/record/2573505
[62] Amazon, "Qiskit provider for amazon braket," 2022. [Online]. Available: https://aws.amazon.com/blogs/quantum-computing/introducing-the-qiskit-provider-for-amazon-braket/
[63] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *ArXiv*, vol. abs/2004.11362, 2020.
[64] IBM, "TwoLocal," 2024. [Online]. Available: https://docs.quantum.ibm.com/api/qiskit/docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.realamplitudes