

PAPER • OPEN ACCESS

# A hybrid quantum–classical neural network for learning transferable visual representation

To cite this article: Ruhan Wang *et al* 2023 *Quantum Sci. Technol.* **8** 045021

View the [article online](#) for updates and enhancements.

## You may also like

- [Key questions for the quantum machine learner to ask themselves](#)  
Nathan Wiebe
- [All-fiber low threshold tunable multimode Q-switched mode locked fiber laser using a few mode Er-doped fiber](#)  
Yunlong Fan, Peng Zhang, Yuzhu Ning et al.
- [Fast Scalar Quadratic Maximum Likelihood Estimators for the CMB \*B\*-mode Power Spectrum](#)  
Jiming Chen, Shamik Ghosh, Hao Liu et al.

# Quantum Science and Technology



## PAPER

# A hybrid quantum–classical neural network for learning transferable visual representation

### OPEN ACCESS

RECEIVED  
30 January 2023

REVISED  
26 July 2023



ACCEPTED FOR PUBLICATION  
18 August 2023

PUBLISHED  
11 September 2023

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



Ruhan Wang<sup>1</sup>, Philip Richerme<sup>2</sup>  and Fan Chen<sup>1,\*</sup> 

<sup>1</sup> Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, Indiana, United States of America

<sup>2</sup> Department of Physics, Indiana University, Bloomington, Indiana, United States of America

\* Author to whom any correspondence should be addressed.

E-mail: [fc7@iu.edu](mailto:fc7@iu.edu)

**Keywords:** hybrid quantum–classical algorithm, quantum neural network, variational quantum circuit, NISQ, contrastive language–image pre-training, zero-shot learning

## Abstract

State-of-the-art quantum machine learning (QML) algorithms fail to offer practical advantages over their notoriously powerful classical counterparts, due to the limited learning capabilities of QML algorithms, the constrained computational resources available on today's noisy intermediate-scale quantum (NISQ) devices, and the empirically designed circuit ansatz for QML models. In this work, we address these challenges by proposing a hybrid quantum–classical neural network (CaNN), which we call QCLIP, for Quantum Contrastive Language-Image Pre-Training. Rather than training a supervised QML model to predict human annotations, QCLIP focuses on more practical transferable visual representation learning, where the developed model can be generalized to work on unseen downstream datasets. QCLIP is implemented by using CaNNs to generate low-dimensional data feature embeddings followed by quantum neural networks to adapt and generalize the learned representation in the quantum Hilbert space. Experimental results show that the hybrid QCLIP model can be efficiently trained for representation learning. We evaluate the representation transfer capability of QCLIP against the classical Contrastive Language-Image Pre-Training model on various datasets. Simulation results and real-device results on NISQ IBM\_Auckland and quantum computer both show that the proposed QCLIP model outperforms the classical CLIP model in all test cases. As the field of QML on NISQ devices is continually evolving, we anticipate that this work will serve as a valuable foundation for future research and advancements in this promising area.

## 1. Introduction

The recent phenomenal investment and rapid development of quantum computing hardware have ushered in the noisy intermediate-scale quantum (NISQ) [1] era where quantum machines are expected to support  $50 \sim 100$  qubits (quantum bits) and around  $10^3$  quantum operations in the coherence time of the physical qubits. In table 1, we summarize the key features of two state-of-the-art quantum computers—IonQ Forte [2] launched in 2022 and IBM Heron [3] slated for 2023. As it shows, NISQ computers suffer from errors due to imperfect qubit control and external interference. Current error rates on NISQ devices greatly exceeds the  $10^{-15}$  error rate required for many quantum algorithms [4–12] to achieve computational advantages. Although fault-tolerant quantum computers are theoretically feasible by incorporating quantum error-correction protocols [13–15], their practical implementation with millions of physical qubits may take decades of research.

NISQ algorithms [16] exploiting error-prone qubits and imperfect quantum gates to solve classically challenging problems have recently been intensively studied in various disciplines [17–27], among which, quantum machine learning (QML) [25–27] has shown significant advantages over its classical counterpart in small-scale learning tasks [28–31]. With the power to access an exponentially large Hilbert space [32] and the ability to represent complex high-dimensional distributions [30], QML models are expected to revolutionize

**Table 1.** A summary on two state-of-the-art quantum computers (1Q-Gate: one-qubit gate; 2Q-Gate: two-qubit gate; SPAM: state preparation and measurement).

Machine	Technology	Qubits #	Coherence	Error Rate		
				1Q-Gate	2Q-Gate	SPAM
IonQ_Forte [2]	Trapped-Ions	32	$\sim 1$ s	0.02%	0.4%	0.5%
IBM_Heron <sup>a</sup> [3]	Superconducting	133	$< 40$ $\mu$ s	0.1 %	2.07%	1.42%

<sup>a</sup> IBM did not provide error rates for IBM\_Heron. We report the error rates for the v3 generation of 127-qubit IBM\_Eagle processor as an approximation.

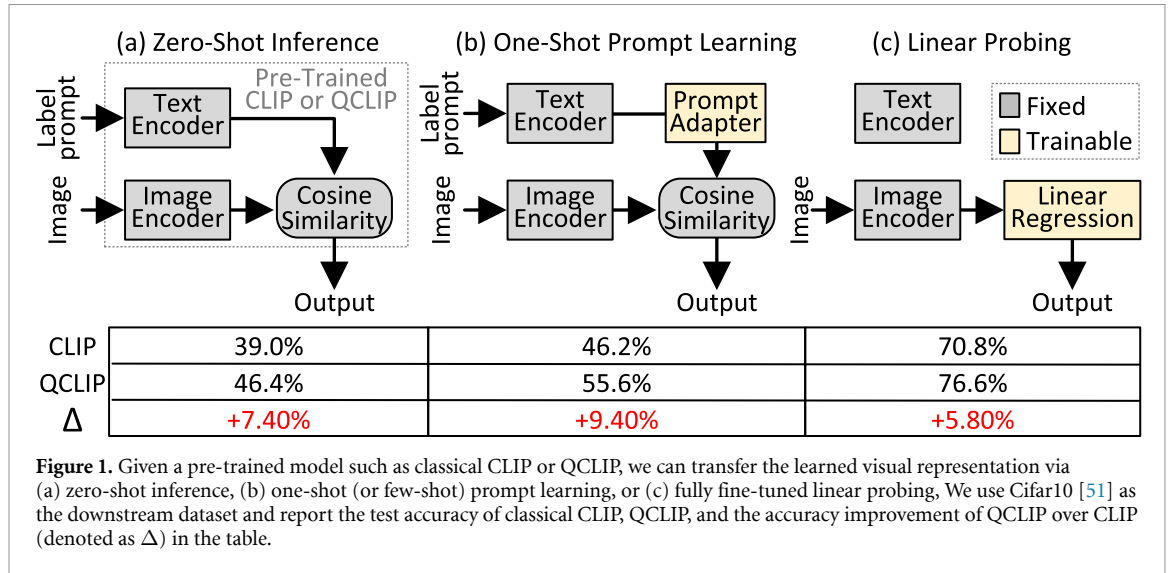
a wide range of applications including material discovery [33, 34], medical health [35, 36], and financial services [37, 38]. *Despite demonstrated advantages, state-of-the-art QML models have yet to solve practical problems due to the limited learning capabilities of QML algorithms, the constrained computational resources available on NISQ computers, and the empirically designed circuit ansatzes for QML models.*

**First**, most QML algorithms [27–29] focus on supervised classification by training models to predict class labels on test data that is generated from the same distribution as the training data. However, sufficient labeled training data for real-world tasks is usually unavailable [39, 40] or prohibitively expensive [41] to obtain. Moreover, representations learned from supervised QML are restricted to a set of ‘golden labels’, which greatly limits the generalization and transferability of the developed models on datasets that are generated from different distributions [42]. **Second**, NISQ computers suffer from limitations in terms of qubit number and coherence time. The input size for real-world datasets is normally millions of tensors with millions of entries each, however, current NISQ devices can only work with small-scale toy benchmarks with input sizes of  $2 \times 2$  or  $4 \times 4$  [43–46]. How to achieve quantum advantages in practical-scale problems with NISQ devices is of great research significance. **Third**, QML models are typically implemented as parameterized quantum circuits [43–50] consisting of a classical-to-quantum data encoder and repeated layers of a variational quantum circuit (VQC). The circuit architecture for the data encoder and the VQC ansatz are currently empirically designed or simply randomly assigned.

### 1.1. Our contributions

In this work, we address the aforementioned challenges by proposing a hybrid quantum–classical neural network (CaNN) architecture for learning transferable visual representation, which we call QCLIP, for Quantum Contrastive Language-Image Pre-training. Our main contributions can be summarized as follows:

- **A novel QML framework for learning transferable visual representation.** Instead of training a supervised QML model for predicting human annotations, we advance the flagship Contrastive Language-Image Pre-Training (CLIP) method [52] by proposing QCLIP, a quantum CLIP framework, which enjoys quantum-enhanced transferability and generalization only efficiently accessible on quantum computers. QCLIP combines limited NISQ resources and classical computing power to perform meaningful tasks, where CaNNs are used to generate low-dimensional data embeddings in classical feature space, while quantum neural networks (QuNNs) are exploited to enhance the model generalization in an exponentially large quantum Hilbert space (section 3.1).
- **Quantum encoding methods and QuNN circuit ansatzes specialized for transferable visual representation learning.** We investigate various encoding methods and circuit ansatzes in the proposed QCLIP framework and identify the optimal candidate circuit ansatz for each quantum component (section 3.2). We implement QCLIP on NISQ devices and carefully study how different training configurations affect final model performance. We provide a detailed training procedure for QCLIP (section 3.3).
- **High-performance visual representation transfer on NISQ devices.** We demonstrate that the hybrid QCLIP model can be successfully trained for representation learning (section 4.1). We evaluate the representation transferability of QCLIP using all mainstream methods including *zero-shot inference*, *one-shot prompt learning*, and *linear probing* and show that QCLIP outperforms the classical CLIP model on various datasets (section 4.2). A brief description of the experimental setup and numerical results are summarized in figure 1. We also provide experimental results on different training configurations (section 4.3) and NISQ IBM\_Auckland quantum computer (section 4.4). Our results show that the proposed QCLIP model outperforms the classical CLIP model in all test cases.



## 2. Background

### 2.1. Learning transferable visual representation

Supervised representation learning methods [42, 53–59] suffer from prohibitively expensive cost on labeled data preparation and poor representation transferability to downstream unseen datasets. Therefore, learning transferable visual representations is proposed and become a long-standing core problem in machine learning. Given a source domain  $\mathcal{D}_S$  with a source task  $\mathcal{T}_S$  and a target domain  $\mathcal{D}_T$  with a target task  $\mathcal{T}_T$ , the goal of transferable visual representation learning is to improve the target function  $f_T(\cdot)$  by reusing the representation learned from  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{D}_S \neq \mathcal{D}_T$  or  $\mathcal{T}_S \neq \mathcal{T}_T$ . Recent works [52, 60–72] encourage models to extract underlying explanatory factors hidden in the image by using unlabeled data in an unsupervised fashion, rather than just predicting human annotations. Provided the unlimited free raw data available on the Internet, this produces a model with better performance, and most importantly, the learned perception enables flexible representation transfer to downstream unseen datasets.

Among all prior arts, the CLIP method [52] has demonstrated state-of-the-art visual representation transfer performance. CLIP collects over 400 M (image, text) pairs and trains an image encoder and a text encoder jointly with a task-agnostic contrastive loss [66, 67]. It is worth mentioning that the text descriptions are often referred to as ‘prompt’ and their design is critical to CLIP performance. Once the training is complete, the quality of the visual representations learned by CLIP can be evaluated via different methods [73] including (1) *zero-shot inference* by directly generalizing the learned CLIP model to an unseen dataset; (2) *one-shot (or few-shot) prompt learning* by training a lightweight prompt adapter neural network [74–76] using one (or a few) training samples per class from the target dataset; or (3) *linear probing* which connects the pre-trained image encoder with a linear classifier [52, 66, 67] fully trained on a sufficiently large number of training data from the target domain. In general, *zero-shot inference* and *linear probing* respectively set the lower and upper bound on model transferability, while *one-shot (or few-shot) prompt learning* achieves intermediate performance because it considers a more practical scenario where the target dataset is neither completely inaccessible nor fully accessible.

### 2.2. QuNNs

As illustrated in figure 2, a standard QuNN begins with a classical-to-quantum encoder  $\mathbf{E}(\mathbf{x})$  that encodes a classical input vector  $\mathbf{x}$  into a  $N_Q$ -qubit quantum state  $|\mathbf{x}\rangle$  [77]:

$$\mathcal{E} : \mathbf{x} \rightarrow |\mathbf{x}\rangle = \mathbf{E}(\mathbf{x})|0\rangle^{\otimes N_Q} = \bigotimes_{j=1}^{N_Q} \mathbf{R}(x_j)|0\rangle \quad (1)$$

where  $\mathbf{R}$  denotes one-qubit gates  $\{\mathbf{R}_X, \mathbf{R}_Y, \mathbf{R}_Z\}$  or their combinations, commonly referred to as *angle* encoding. Note that in this work, we exclude the *amplitude* encoding method due to its high  $\mathcal{O}(2^{N_Q})$  circuit depth, making a QuNN more error-prone [44]. Instead, we focus on the *angle* encoding, which uses  $N_Q$

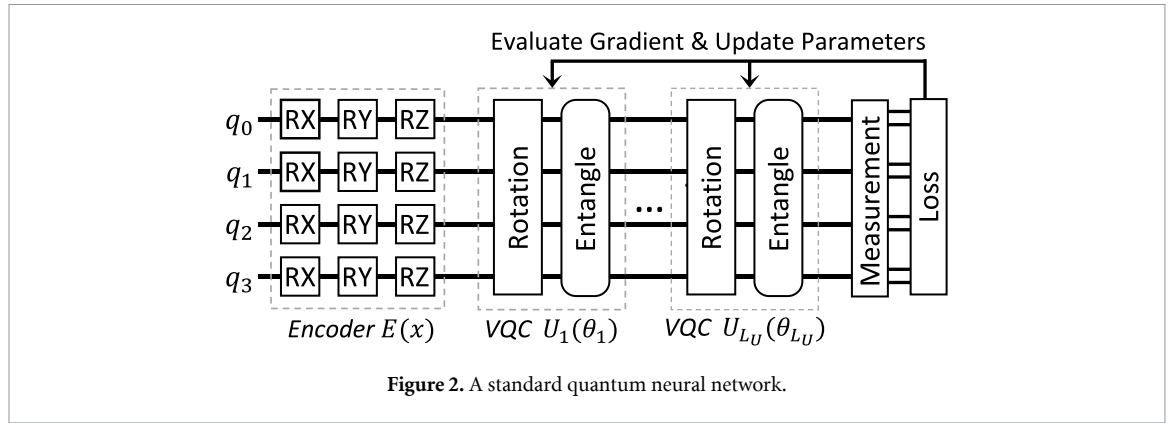


Figure 2. A standard quantum neural network.

qubits and a constant-depth quantum circuit to encode a  $N_Q$ -bit classical data. The generated  $|\mathbf{x}\rangle$  state is often referred to as a quantum input feature map and is manipulated by a subsequent VQC  $\mathbf{U}(\theta)$ :

$$\mathcal{U} : |\mathbf{x}\rangle \rightarrow |\mathbf{y}(\theta)\rangle = \mathbf{U}(\theta)|\mathbf{x}\rangle = \left( \prod_{k=1}^{L_U} \mathbf{U}_k(\theta_k) \right) |\mathbf{x}\rangle \quad (2)$$

where  $\mathbf{U}(\theta)$  is implemented as a concatenation of a VQC ansatz in repeated  $L_U$  layers, and  $\theta_k$  is a set of trainable variables for the  $k_{th}$  layer. As illustrated in figure 2, VQC ansatzes used in mainstream QML models [43–46] are normally constructed by single-qubit rotation gates followed by two-qubit entanglement gates. The final output results are obtained by quantum state measurement,  $\mathbf{M}$ , that maps the output quantum state  $|\mathbf{y}(\theta)\rangle$  to a classical vector  $\mathbf{y}(\theta)$ :

$$\mathcal{M} : |\mathbf{y}(\theta)\rangle \rightarrow \mathbf{y}(\theta) = \langle \mathbf{y}(\theta) | \mathbf{M}^\dagger \mathbf{M} | \mathbf{y}(\theta) \rangle. \quad (3)$$

By default, qubits are measured in the  $z$ -basis for implementation simplicity. Globally the full QuNN can be written as

$$\mathcal{Q} : \mathbf{Q} = \mathbf{M} \circ \mathbf{U}(\theta) \circ \mathbf{E}(\mathbf{x}). \quad (4)$$

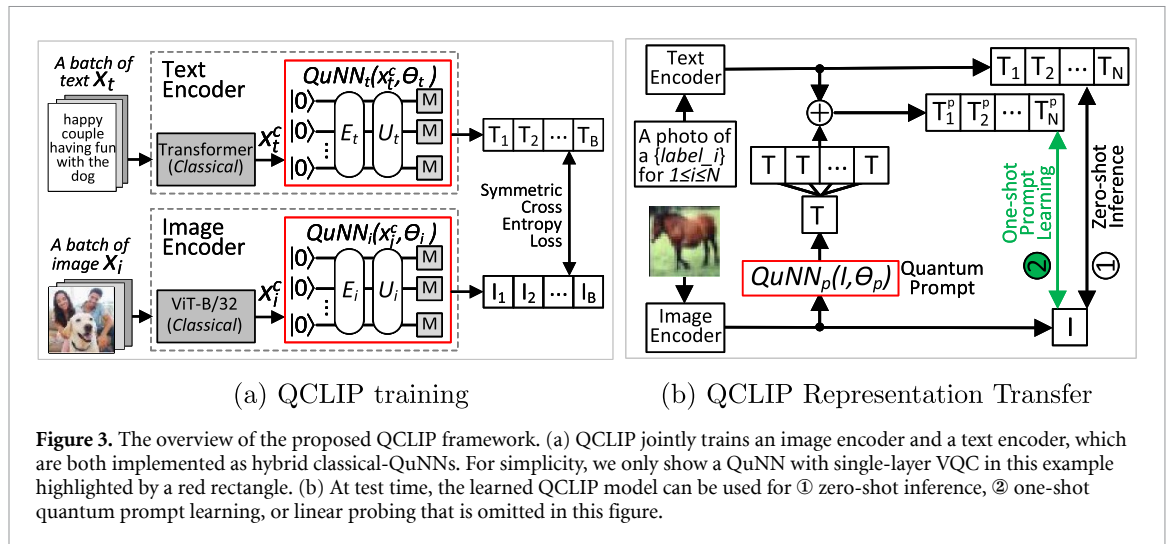
A QuNN model is evaluated by a pre-defined loss function  $\mathbf{L}(\cdot)$  and iteratively trained to obtain optimal parameters via hybrid quantum–classical gradient descent [78]:

$$\mathcal{L} : \mathbf{y}(\theta) \rightarrow \text{Loss} = \mathbf{L}(\mathbf{y}(\theta)) \quad (5)$$

$$\text{Update rule} : \theta_j^{t+1} = \theta_j^t - \eta \frac{\partial \mathbf{L}(\mathbf{y}(\theta))}{\partial \theta_j}. \quad (6)$$

### 2.2.1. Theoretical Insights

While QML theory is continually evolving and in its nascent stages, this work provides insights on the optimal quantum encoder and variational circuit ansatz designs (see appendix C) based on the current state of QML theory research. However, it is important to note that the field currently lacks a standardized consensus. As a result, the discussions presented may be subject to changes or even controversies as our understanding of QML progresses.



**Figure 3.** The overview of the proposed QCLIP framework. (a) QCLIP jointly trains an image encoder and a text encoder, which are both implemented as hybrid classical-QuNNs. For simplicity, we only show a QuNN with single-layer VQC in this example highlighted by a red rectangle. (b) At test time, the learned QCLIP model can be used for ① zero-shot inference, ② one-shot quantum prompt learning, or linear probing that is omitted in this figure.

### 3. Method

In this section, we present the details of the proposed hybrid quantum–CaNN architecture. In section 3.1, we describe the general QCLIP framework, introduce QCLIP representation transfer for *zero-shot inference*, *one-shot (or few-shot) quantum prompt learning*, and fully supervised *linear probing*. In section 3.2, we present the implementation of the QuNNs used in QCLIP. Finally, in section 3.3, we discuss the training approach of QCLIP.

#### 3.1. The QCLIP framework

At the core of QCLIP is to learn image representations by contrasting them with the text prompt of the images, the same as classical CLIP [52]. The idea of QCLIP is inspired by recent research advances in quantum-enhanced feature learning [32] through exploiting quantum mechanical superposition, entanglement, and interference principles. Instead of using purely QuNNs on small datasets as in [32], the proposed QCLIP architecture is implemented by combining classical and QuNNs in one framework, thus, QCLIP can leverage CaNNs for large dataset preprocessing while utilizing QuNNs for quantum-enhanced feature adaptation and generalization.

##### 3.1.1. QCLIP overview

As shown in figure 3(a), each high-dimensional input (image, text) pair  $(\mathbf{x}_i, \mathbf{x}_t)$  is first processed by CaNNs to generate compact low-dimensional data embedding in the classical feature space, and then QuNNs are utilized to further adapt the embeddings in an exponentially large quantum Hilbert space. Taking the hybrid image encoder network as an example, it utilizes a classical ViT-B/32 model [52] to produce a low-dimensional classical image embedding vector  $\mathbf{x}_i^c$  and then uses an QuNN,  $QuNN_i(\mathbf{x}_i^c, \theta_i)$ , to map  $\mathbf{x}_i^c$  to the quantum state space. A classical image embedding  $I$  is eventually generated via quantum measurements in the  $z$ -basis. Similarly, the hybrid text encoder is implemented as a classical 12-layer 512-wide text Transformer model with 8 attention heads [79] followed by an QuNN,  $QuNN_t(\mathbf{x}_t^c, \theta_t)$ , to generate a text embedding vector  $T$ . Note that  $I$  and  $T$  share a common dimensionality, specifically  $N_Q$ , which corresponds to the number of qubits utilized in the QuNNs. At the training time, QCLIP is optimized to predict the correct pairings of a batch (with a batch size  $B$ ) of  $(I_k, T_j)$  ( $0 \leq k, j < B$ ) pairs using symmetric cross-entropy loss. The ViT-B/32 and text Transformer models are particularly selected as classical feature extractors since they have demonstrated the best performance in classical CLIP models [52].

##### 3.1.2. QCLIP representation transfer

We evaluate the transfer capability of learned QCLIP visual representations using all mainstream evaluation methods introduced in section 2.1. Below we describe the detailed configuration for each method.

**Zero-Shot inference** assumes no access to the target dataset at all. Assuming the downstream dataset has  $N$  class names, we reuse the pre-trained QCLIP and compute the text embeddings,  $\{T_1, T_2, \dots, T_N\}$ , for each target class name, as denoted as ① in figure 3(b). A test image is processed by the image encoder to generate a feature embedding,  $I$ . The similarity between  $I$  and  $\{T_1, T_2, \dots, T_N\}$  is then calculated and normalized into a probability distribution via a softmax function. We identify the most probably (image, text) pair as the output prediction. Prior works [52] show that the transferability of the classical CLIP model is greatly

impacted by the input text that describes the image and found that using a text template improves performance. We follow the same text template engineering and ensembling schemes in [52].

**One-Shot (or Few-shot) prompt learning** targets a more practical scenario where one (or a few) training samples per class from target datasets are available at the test time. Various prompt learning algorithms [74–76] are recently proposed to alter the functionality of a pre-trained model across domains. However, none of these schemes can be directly applied to work with QuNNs. In this work, we introduce a quantum prompt learning algorithm.

As denoted as ② in figure 3(b), we design a domain prompt adapter,  $\text{QuNN}_p(I, \theta_p)$ , which is implemented as a parameterized QuNN. At the training time, the quantum prompt adapter takes the image vector  $I$  as input and generates a prompt  $T$  using one (or a few) unlabeled images  $x_i$  from the target training dataset.  $T$  has the same width as text embedding vectors and is added to all the original class embeddings to generate an adapted set of text pairing embeddings, denoted as  $\{T^p_1, T^p_2, \dots, T^p_N\}$ . At the test time, we utilize domain-adapted text embeddings  $\{T^p_1, T^p_2, \dots, T^p_N\}$  instead of the general QCLIP text embeddings  $\{T_1, T_2, \dots, T_N\}$  to compute the similarity between the input image and the predicted classes.

**Linear probing** assumes full access to the target training dataset. We adopt the established linear evaluation protocol [52, 66, 67] to test the visual representation transfer of QCLIP, where we freeze the QCLIP image encoder and only train a linear classification prediction layer on the output of the encoder network. The linear classifier is implemented as a logistic regression model and fully trained on target datasets for 1000 iterations. We then apply the whole network consisting of the QCLIP image encoder and the linear classifier head to the test data and report the classification accuracy.

### 3.1.3. QCLIP implementation on NISQ computers

The classical ViT-B/32 and text Transformer respectively map the original data pair to a 512-dimensional image/text feature vector [52], which is considered as a classical compact encoding of the input. Ideally, the CaNNs can pass these 512-dimensional vectors to the QuNNs for further processing, however, NISQ computers available now only have  $50 \sim 100$  qubits. Therefore, we follow the common practice [80, 81] by inserting a 512-to- $N_C$  fully-connected layer between the classical and quantum layers to compress the initial feature vectors to a  $N_C$ -dimensional vector that can be effectively encoded in a practically available  $N_Q$ -qubit quantum system. The relationship between  $N_C$  and  $N_Q$  is determined by the classical-to-quantum encoding methods. To investigate the impact of compressed feature dimensions on the final performance, we conducted a study of the accuracy achieved by QCLIP with different  $N_C$ , as reported in figure A1 in appendix A. The experimental results demonstrate that increasing  $N_C$  leads to improved accuracy and transferability of the QCLIP model.

In conclusion, with a fixed  $N_Q$  qubits on a quantum computer, the encoder is expected to enable a larger  $N_C$ , allowing for a more accurate input representation by preserving a greater amount of information from the classical input data. The default angle encoding, which uses  $N_Q$  qubits, can only encode  $N_Q$  features, motivating the development of a denser encoder to accommodate a larger  $N_C$  in this work. Furthermore, the performance improvement with the increasing  $N_C$  also indicates that advancements in technology and the availability of more qubits will lead to improvements in the implementation scale of QCLIP and its corresponding performance and transferability.

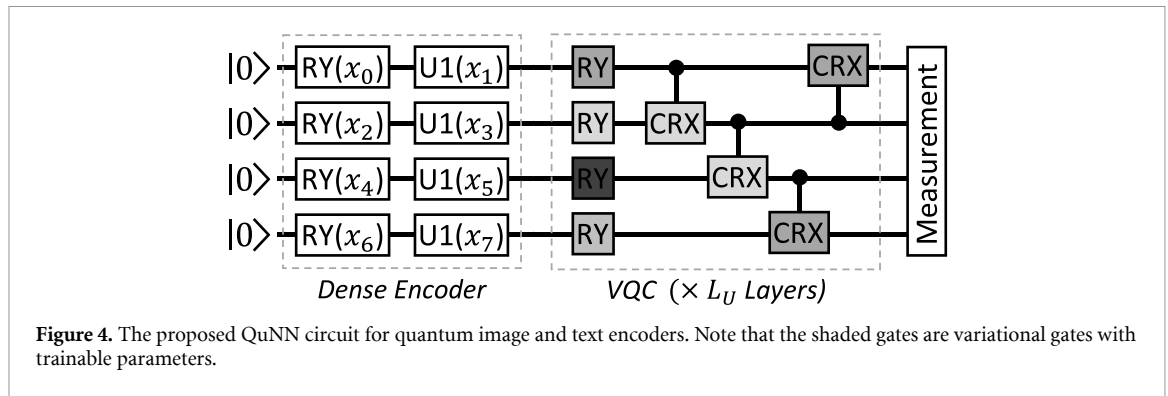
## 3.2. QuNNs

QuNNs used in QML models are currently empirically designed. In this work, we investigate various widely used encoding methods and VQC circuit ansatzes. Based on the performance evaluation, we identify the optimal QuNN circuits for each quantum component in the proposed QCLIP framework. We provide a full list of candidate quantum encoding methods and VQC ansatzes studied in this work respectively in appendices A and B.

### 3.2.1. Quantum image and text encoders

Figure 4 shows the QuNN circuits used in the text and image encoder networks. In this example, we consider a QuNN with only four qubits for simplicity. The number of qubits as well as the number of VQC layers (i.e.  $L_U$ ) in a generic QCLIP model can be adjusted to fit the problem of interest.

**Classical-to-quantum encoder** is essential for ensuring QML model accuracy, as it extracts and encodes relevant features from classical data into a quantum format, enabling subsequent processing in the quantum domain. However, the limited number of qubits in current quantum computers presents challenges in effectively embedding classical data, particularly with large-dimensional input datasets. In this work, we follow the generalized *dense angle encoding* [77] and present a dense classical-to-quantum encoder consisting of a layer of RY gates followed by a layer of U1 gates, as shown in figure 4. Given a classical  $N_C$ -dimensional input vector  $\mathbf{x} = (x_0, x_1, \dots, x_{N_C-1})$ , a quantum input feature map is generated by applying the encoding



circuits to the ground quantum state  $|0\rangle^{\otimes N_Q}$  of a  $N_Q$ -qubit system where  $N_c = 2N_Q$ , defining an encoder  $\mathbf{E}(\mathbf{x})$  given by (see detailed mathematical derivation in appendix A):

$$\mathbf{x} \rightarrow |\mathbf{x}\rangle = \mathbf{E}(\mathbf{x})|0\rangle^{\otimes N_Q} = \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} \cos\left(\frac{x_{2j-1}}{2}\right)|0\rangle + e^{i \cdot x_{2j}} \sin\left(\frac{x_{2j-1}}{2}\right)|1\rangle. \quad (7)$$

In contrast to the conventional encoding method represented by equation (1), which uses  $N_Q$  qubits to represent  $N_Q$  features, QCLIP leverages the relative phase degree of freedom along with the angles to embed  $2 \times$  more features using the same number of qubits. On top of this dense encoding method, we also explored *data re-uploading* [82] and *variational encoding* [83] to improve QuNN performance. Experimental results (see appendix C) show that these two methods achieve negligible accuracy improvement in a QCLIP model, which contradicts previous conclusions from QML models [82, 83] implemented purely by QuNNs. We interpret the main reason as that these two methods primarily provide nonlinearity to a linear QuNN, while in a QCLIP model, nonlinearity is already sufficiently provided by the earlier CaNNs in the framework. Considering the significant implementation and training overhead introduced by *data re-uploading* and *variational encoding*, we do not recommend using these two methods in QCLIP.

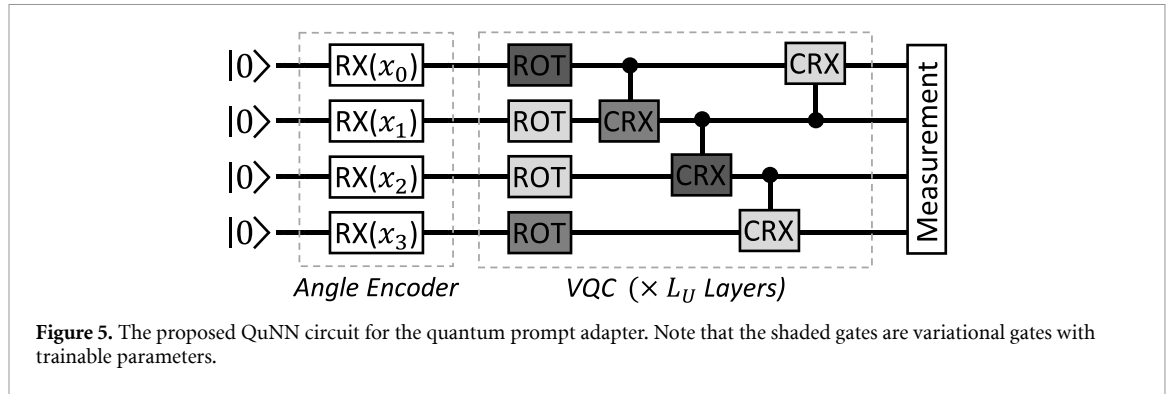
**VQC Ansatz** is constructed by parameterized single-qubit rotation gates followed by nearest-neighbor coupling of qubits using entanglement two-qubit gates in current QuNNs [43–46]. Such a circuit ansatz has demonstrated superior expressive capability in various applications. The logic behind such designs is that single-qubit rotations provide a way to parameterize circuits, while two-qubit gates provide entanglement between two target qubits. Early designs [25] utilize fixed two-qubit CNOT gates to force maximum entangling power, while recent research [43–45] explored trainable entanglement by replacing fixed CNOT gates with parameterized two-qubit gates such as CRX( $\theta$ ) [43], U3( $\theta, \phi, \lambda$ ) [45], or CROT( $\phi, \theta, \omega$ ) [44].

We run experiments with different VQC ansatzes (see details in appendix B) in the QCLIP architecture, results (see appendix C) show that a VQC using parameterized two-qubit CRX( $\theta$ ) gates leads to significant accuracy improvement compared to a baseline VQC with fixed two-qubit CNOT gates, demonstrating that adaptive and flexible entanglement rather than fixed maximal entanglement performs better for a QML algorithm, which is consistent with the conclusions in supervised QuNN models [43–45]. However, we find that further increasing the flexibility by replacing CRX( $\theta$ ) gates with U3( $\theta, \phi, \lambda$ ) and CROT( $\phi, \theta, \omega$ ) gates introduces significant hardware overhead and training complexity with no noticeable performance improvement. Therefore, we present the VQC circuit implemented with two-qubit CRX( $\theta$ ) in figure 4.

### 3.2.2. Quantum prompt adapter neural network

The quantum prompt adapter  $\text{QuNN}_p(I, \theta_p)$  takes an image vector  $I$  as input and generates a domain-adapted text vector  $T$ . In designing  $\text{QuNN}_p$  encoders, we chose the default *angle* encoding over the *dense* encoder for two main reasons. First, maintaining the output dimensionality of  $\text{QuNN}_p$  as the input vector  $I$  is required to ensure seamless integration with the subsequent components. Second, expanding the dimensionality of  $I$  by  $2 \times$  and using the *dense* encoder is possibly but considered impractical.  $I$  is already a compact representation learned by  $\text{QuNN}_i$ , and increasing its dimensionality would not provide significant benefits. Moreover, it could introduce unnecessary complexity without improving overall performance. Through experiments, we identified the optimal circuit structure shown in figure 5 consisting of a single layer of RX gates in the encoder and a VQC circuit employing two-qubit CRX( $\theta$ ) gates for qubit entanglement, following previous work [43].





### 3.3. Training of QCLIP

To fix the parameters in the 512-to- $N_C$  compression layer and the  $N_Q$ -qubit QuNNs used in the image and text encoders, we train the QCLIP model using CC3M [84] as a proxy dataset. The training goal is to predict which text *as a whole* is paired with which image. Specifically, given a batch of  $B$  input (images, text) pairs, QCLIP obtains respectively  $B$  image embedding vectors and  $B$  text embedding vectors. We denote  $(I_k, T_j)$  where  $k = j$  is a positive pair and a negative pair for  $k \neq j$ . We define a function that calculates loss using all these possible pairs and minimizes this function via stochastic gradient descent. Intuitively, if information can be successfully passed forward and backward in the hybrid architecture of QCLIP, the measured similarity between representations for positive pairs will decrease, while the distance between representations for negative pairs will increase.

#### 3.3.1. Loss function

We consider two widely used loss functions, namely, normalized temperature-scaled contrastive loss [66, 67] and symmetric cross-entropy loss [52, 85]. We optimize the loss over similarity scores. Experimental results show symmetric cross-entropy loss outperforms contrastive loss for the training of QCLIP. We provide the pseudocode of cross-entropy loss based QCLIP training in algorithm 1. We also provide details of the contrastive loss in appendix D for comparison.

---

#### Algorithm 1. Cross-Entropy Loss based QCLIP Training.

---

##### Input:

1. Batch size:  $B$ ,
2. Label:  $[1, 2, \dots, B]$ ,
3. Cross entropy loss:  $F_{loss} = -\frac{\sum_{i=1}^B l_i \cdot \log(p_i)}{B}$ , where  $l_i$  is the truth label and  $p_i$  is the *Softmax* probability for the  $i^{\text{th}}$  class.

##### Output:

1. Training loss,  $loss$
  - 1: Generate a batch of image embedding output vector  $[I_1, I_2, \dots, I_B]$ .
  - 2: Generate a batch of text embedding output vector  $[T_1, T_2, \dots, T_B]$ .
  - # Compute  $logits\_image = [l_{I_1}, l_{I_2}, l_{I_3}, \dots, l_{I_B}]$
  - 3: **for** ( $i = 1; i < B+1; i++$ ) **do**
  - 4:   **for** ( $t = 1; t < B+1; t++$ ) **do**
  - 5:      $l\_T_i = \frac{l_i \cdot T_t}{|I_i| |T_t|}$
  - 6:   **end for**
  - 7: **end for**
  - # Compute  $logits\_text = [l_{T_1}, l_{T_2}, l_{T_3}, \dots, l_{T_B}]$
  - 8: **for** ( $t = 1; t < B+1; t++$ ) **do**
  - 9:   **for** ( $i = 1; i < B+1; i++$ ) **do**
  - 10:      $l\_I_t = \frac{T_t \cdot I_i}{|T_t| |I_i|}$
  - 11:   **end for**
  - 12: **end for**
  - 13:  $loss\_image = F_{loss}(logits\_image, label)$ .
  - 14:  $loss\_text = F_{loss}(logits\_text, label)$ .
  - 15:  $loss = \frac{1}{2}(loss\_image + loss\_text)$ .
  - 16: **return**  $loss$ .
-

### 3.3.2. Training method

We implement the classical ViT-B/32 and text Transformer models in PyTorch [86]. We implement the QuNNs using PennyLane [87]. We use a mini-batch size of 128. We train the model for 75 iterations. We use Adam optimizer and set the learning rate to 0.001.

Among all the training hyperparameters, the initialization of parameters in QuNNs emerges as the most critical factor influencing the final performance of a QCLIP model. This is primarily due to the challenge of exponentially vanishing gradients concerning the quantum circuit depth and qubit number. For a deeper understanding, interested readers can refer to the theoretical discussion on the effect of parameter initialization on the trainability and performance of QML models provided in [88]. In this work, we study both uniform initialization and Gaussian initialization in QCLIP as detailed in appendix E. Inspired by classical Xavier initialization [89], we utilize the information of QuNN structures in the Gaussian initialization by defining  $\mathcal{N}(0, \sigma^2)$ , where  $\sigma = 1/\sqrt{N_Q}$ . Experimental results show that the Gaussian distribution demonstrates better performance in terms of accuracy, training stability, and convergence.

## 4. Results and analysis

In this section, we evaluate the effectiveness of the proposed QCLIP model. We follow the general QCLIP architecture and implement a practical design by setting  $N_C$ ,  $N_Q$ , and  $L_U$  respectively to 16, 8, and 2. We run numerical simulations and report results on representation learning in section 4.1. To compare QCLIP with classical CLIP, we create a baseline model by implementing classical CLIP in PyTorch. We follow the training approaches used in the original work [52, 81], with the only difference being the insertion of a 512-to- $N_C$  fully-connected layer in the image/text encoder. This modification is made to ensure a fair and equal comparison between QCLIP and classical CLIP models. In section 4.2, we evaluate the representation transferability of QCLIP and show that QCLIP outperforms the classical CLIP model on various datasets. Section 4.3 provides exploration results for different training configurations. We also implement a proof-of-concept QCLIP on NISQ IBM\_Auckland quantum computer and report its performance results in section 4.4.

### 4.1. Results on QCLIP representation learning

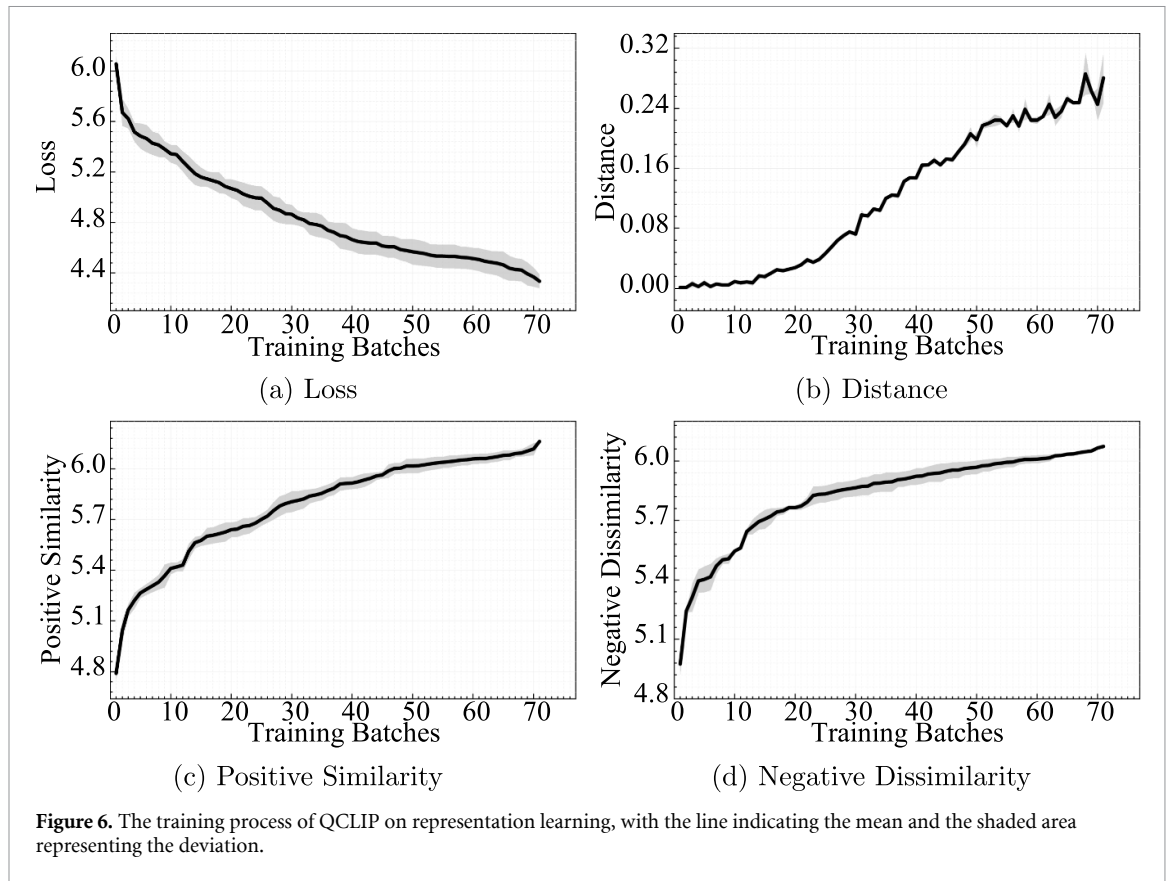
We first verify whether the proposed hybrid QCLIP model can be successfully trained for representation learning. To this end, we train the QCLIP model using CC3M [84] as a proxy dataset for 70 batches and record the training loss after each batch in figure 6(a). Results show that the loss decreases from 6.075 to 4.152 over the course of training, indicating that our model is able to learn. It is notable that the training time for QCLIP is significantly less than classical QCLIP models, which would typically take several hundred epochs [52]. By comparison, QCLIP is more compute-efficient, which allows us to reach higher overall performance within a limited computing budget.

We further quantitatively study the representation learning ability of QCLIP. We adopt the widely used Hilbert-Schmidt distance as the evaluation metric and report results on several key distances, following the approach taken in related work [52, 81]. Figures 6(b)–(d) respectively record the distance between positive and negative pairs (denoted as *Distance*), similarity within positive pairs (denoted as *Positive Similarity*), dissimilarity between negative pairs (denoted as *Negative Dissimilarity*). Throughout the training process, we observe that the measured similarity and dissimilarity undergo expected changes, indicating successful information propagation both forward and backward in the hybrid architecture of QCLIP. These quantitative results affirm that quantum components can effectively combine with classical resources to achieve meaningful and nontrivial representation learning tasks.

### 4.2. Results on QCLIP representation transfer

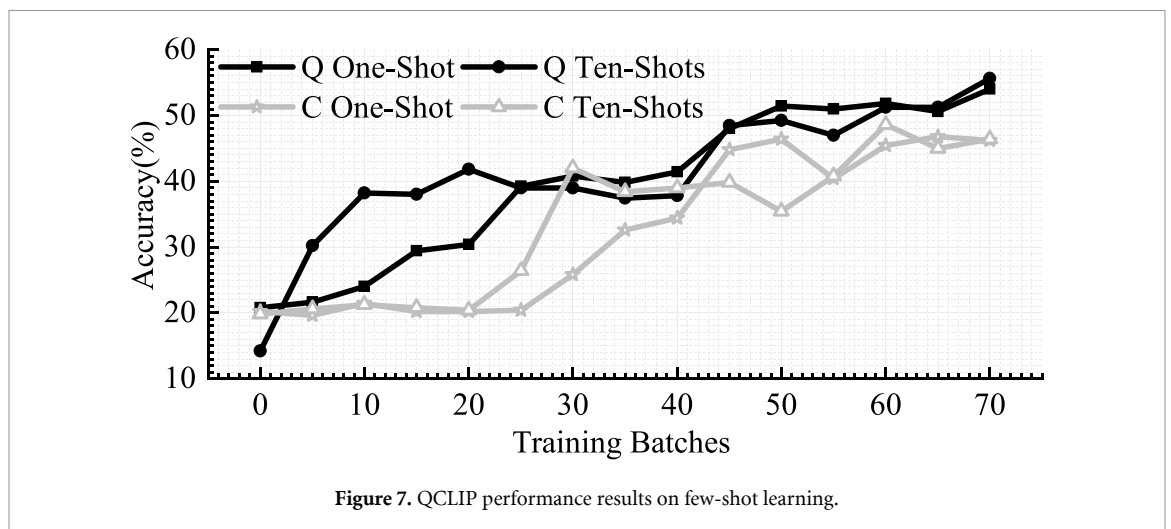
QCLIP is pre-trained to predict whether an image and a text prompt are paired together in a source dataset. This capability is then reused to perform *zero-shot inference*, *one-shot prompt learning*, and *linear-probing*, to study the representation transfer ability on downstream datasets. To demonstrate the robustness of QCLIP on various datasets with wide distributions, we evaluate QCLIP on four different target datasets including MNIST [90], Cifar10 [51], OxfordPet [91], and Food101 [92].

In table 2, we summarize the performance of QCLIP on each task and highlight the accuracy improvement (denoted as  $\Delta$ ) provided by QCLIP compared to classical baselines. The quantitative results show that QCLIP is robust on all tested datasets and outperforms classical CLIP on all tasks. While supervised *linear probing* exhibits the upper bound on model transferability, QCLIP has the lowest performance improvement over CLIP on this task. Notably, *one-shot prompt learning* benefits the most from QCLIP with a performance improvement up to +17.21% on the Food101 dataset.



**Table 2.** Performance comparison between QCLIP and CLIP on representation transfer.

Dataset	Zero-Shot			One-Shot			Linear probing		
	CLIP	QCLIP	$\Delta$	CLIP	QCLIP	$\Delta$	CLIP	QCLIP	$\Delta$
MNIST [90]	17.97%	20.32%	+2.35%	20.03%	30.51%	+10.48%	59.12%	62.05%	+2.93%
Cifar10 [51]	44.41%	46.40%	+1.99%	46.82%	55.62%	+8.80%	70.82%	76.63%	+5.81%
OxfordPet [91]	17.95%	27.12%	+9.17%	19.32%	33.93%	+14.61%	67.73%	68.26%	+0.53%
Food101 [92]	31.19%	37.97%	+6.78%	32.25%	49.46%	+17.21%	59.76%	64.02%	+4.26%



We further increase the shot number from one to ten for both classical CLIP and QCLIP and report the *few-shot* performance in figure 7. The performance of few-shot prompt learning shows negligible improvement when the shot number increases from one to ten, indicating that the accuracy of the small domain prompt generators rapidly saturated with just very few (i.e. one per class) training data.

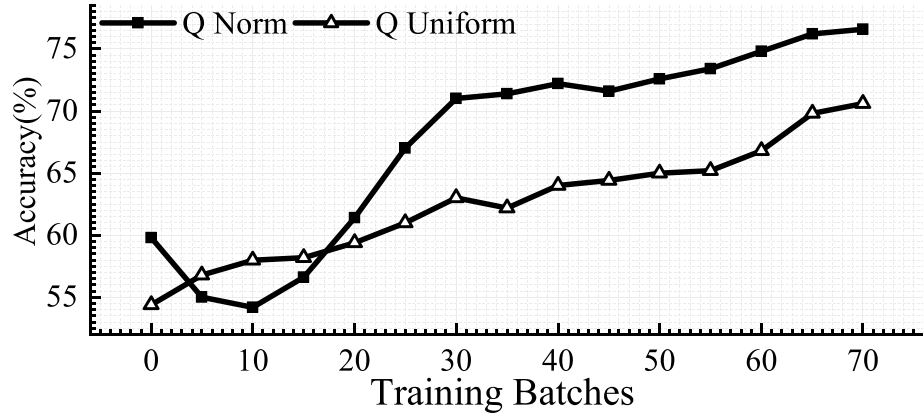


Figure 8. QCLIP performance using different initialization methods.

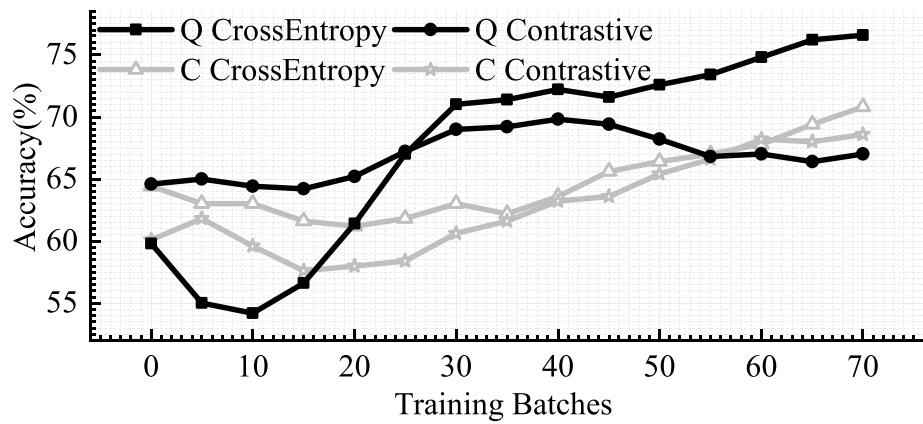


Figure 9. QCLIP performance using different loss functions.

#### 4.3. Results on different training configurations

As discussed in section 3.3, the QCLIP performance is greatly impacted by pre-defined loss functions and parameter initialization. Since *linear probing* represents an upper bound of QCLIP representation transferability, here we use it as a proxy task to explore the impact of different types of loss functions and parameter initialization methods.

Figure 8 compares the QCLIP model accuracy on *linear probing* by using normalized initialization (denoted as *Q Norm*) and uniform initialization (denoted as *Q Uniform*). Results show that *Q Uniform* performs better in the first several training runs, while *Q Norm* provides better (8.2% higher than *Q Uniform*) final accuracy. These results are consistent with the observation reported in a previous work [93]. Therefore, normalized initialization is adopted in QCLIP training.

Figure 9 reports the performance on *linear probing* for classical CLIP and QCLIP by using respectively contrastive loss and cross-entropy loss. In general, cross-entropy loss improves the performance of both classical and quantum models. For classical CLIP training, the cross-entropy loss (denoted as *C CrossEntropy*) provides a 2.2% accuracy improvement compared to the contrastive loss (denoted as *C Contrastive*). For QCLIP, a significant 9.6% accuracy improvement is achieved when replacing the contrastive loss (denoted as *Q Contrastive*) with the cross-entropy loss (denoted as *Q CrossEntropy*). Recent work on quantum self-supervised learning [81] directly employs the contrastive loss function for QuNN training, whereas in this work we identify the cross-entropy loss function as an optimal option and used it for QCLIP training.

#### 4.4. Results on NISQ devices

In addition to the numerical simulation results reported in previous sections, we also implement a proof-of-concept QCLIP on real NISQ devices and report its performance to demonstrate the effectiveness of QCLIP. We use the IBM\_Auckland quantum computer, which is a 27-qubit device with respective 0.022%, 1.164%, and 1.110% error rates for 1Q-Gate, 2Q-Gate, and SPAM. Compared with the state-of-the-art devices reported in table 1, IBM\_Auckland is a more practical NISQ device that is publicly

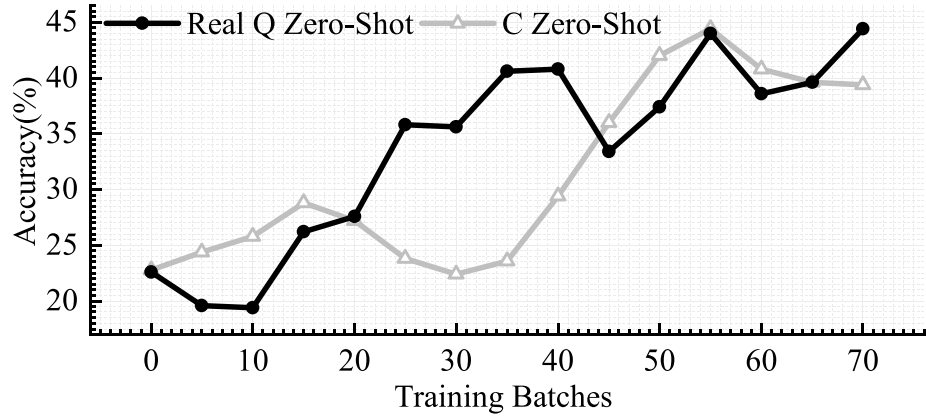


Figure 10. QCLIP performance on zero-shot inference using the IBM\_Auck1and quantum computer.

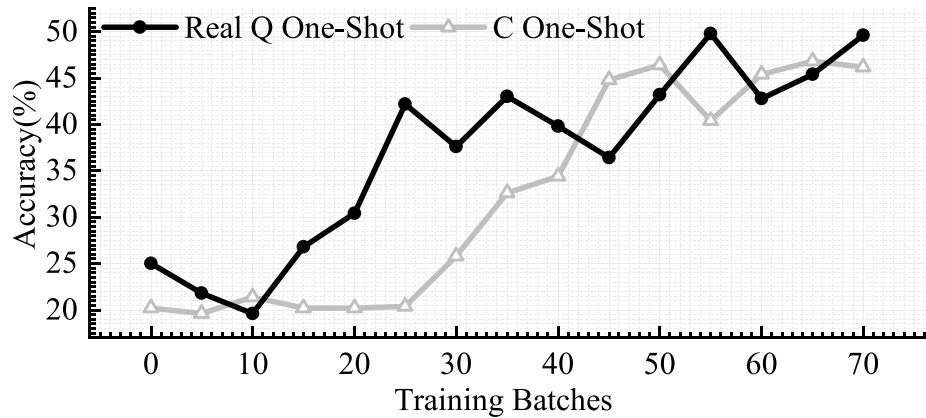


Figure 11. QCLIP performance on one-shot prompt learning using the IBM\_Auck1and quantum computer.

available to average users. We adopt the pre-trained QCLIP model and implemented it on IBM\_Auck1and using only 8 qubits.

We perform *zero-shot inference* and *one-shot prompt learning* on real devices and report the results respectively in figures 10 and 11. Note that we exclude the fully fine-tuned *linear probing* on real devices due to its long training latency. In general, the performance of QCLIP on real devices is decreased due to the noisy qubits and imperfect control and measurement. Specifically, the QCLIP accuracy on *zero-shot inference* drops from 46.4% to 44.4% (i.e. the final accuracy for *Real Q Zero-Shot* in figure 10), while the performance on *one-shot prompt learning* decreases from 55.6% to 49.6% (i.e. the final accuracy for *Real Q One-Shot* in figure 11). However, the classical CLIP model only achieves respectively an accuracy of 39.0% and 46.2% for *zero-shot inference* and *one-shot prompt learning*. Therefore a quantum advantage (up to 5.4%) on representation transferability is still reserved for real-device results.

## 5. Conclusion

Current QML models mainly focused on supervised classification tasks using down-sampled input data with a very small scale, i.e. labeled images with a  $4 \times 4$  or even  $2 \times 2$  size. Such models failed to solve practical problems and show limited generalization and transferability to unseen downstream datasets. In this work, we propose to advance the flagship CLIP method by proposing QCLIP, a quantum CLIP framework, to improve the performance of QML algorithms on transfer representation learning tasks. The key idea is to leverage the quantum-enhanced transferability and generalization only efficiently accessible on quantum computers. However, current quantum computers are all NISQ devices, which can only support  $50 \sim 100$  qubits and a limited number of quantum gate operations. In order to leverage the limited NISQ resources to perform meaningful tasks, QCLIP combines quantum computing resources with classical computing power in a hybrid quantum–classical fashion, where CaNNs are used to generate low-dimensional input embeddings in the classical feature space, and QuNNs are employed to enhance the model generalization in

the quantum Hilbert space. We survey the mainstream QuNN implementation and study how different encoding methods, variational circuit ansatzes, and training configurations affect the final performance of the QCLIP model. We present a dense encoding method in this work, and also identify the optimal quantum circuit for each quantum component in QCLIP.

We implement a small-scale QCLIP and demonstrate the proposed hybrid quantum–CaNN can be successfully trained for representation learning. We evaluate the transfer representation learning capability of QCLIP against the classical CLIP model using different datasets. Experimental results on numerical simulation and NISQ IBM\_Auckland quantum computer both show that QCLIP model outperforms the classical CLIP model in all test cases.

## Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

## Acknowledgments

This work was supported by the National Science Foundation CAREER Award (Grant No. CNS-2143120). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of grant agencies or their contractors.

## Appendix A. Quantum encoding methods

Here we provide a detailed mathematical derivation of equation (7). As illustrated in figure 4, the proposed dense encoding method in this work is implemented by applying a layer of  $RY(x_{2j-1})$  gates followed by a layer of  $U1(x_{2j})$  gates to the ground quantum state of a  $N_Q$ -qubit system, where  $\mathbf{x} = (x_0, x_1, \dots, x_{N_C-1})$  represents the classical  $N_C$ -dimensional input vector. The matrix representations for  $RY$  gate and  $U1$  gate are:

$$RY = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad U1 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}. \quad (\text{A.1})$$

The generated quantum input feature map is:

$$\mathbf{x} \rightarrow |\mathbf{x}\rangle = \mathbf{E}(\mathbf{x})|0\rangle^{\otimes N_Q} \quad (\text{A.2})$$

$$= \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} U1(x_{2j}) \cdot RY(x_{2j-1}) \cdot |0\rangle \quad (\text{A.3})$$

$$= \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} \begin{bmatrix} 1 & 0 \\ 0 & e^{i \cdot x_{2j}} \end{bmatrix} \cdot \begin{bmatrix} \cos \frac{x_{2j-1}}{2} & -\sin \frac{x_{2j-1}}{2} \\ \sin \frac{x_{2j-1}}{2} & \cos \frac{x_{2j-1}}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (\text{A.4})$$

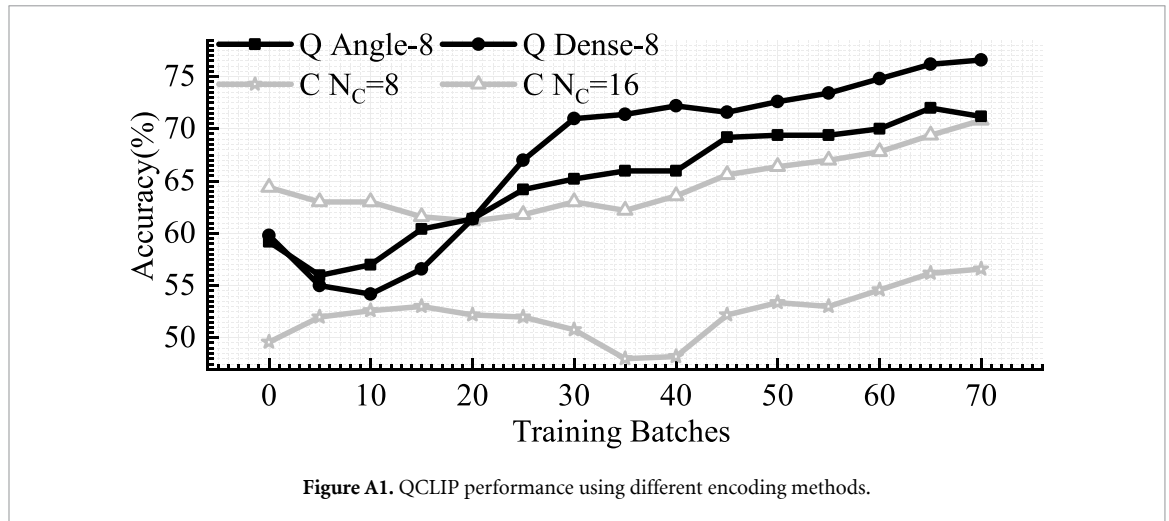
$$= \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} \begin{bmatrix} \cos \frac{x_{2j-1}}{2} \\ e^{i \cdot x_{2j}} \sin \frac{x_{2j-1}}{2} \end{bmatrix} \quad (\text{A.5})$$

$$= \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} \cos \left( \frac{x_{2j-1}}{2} \right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + e^{i \cdot x_{2j}} \sin \left( \frac{x_{2j-1}}{2} \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (\text{A.6})$$

$$= \bigotimes_{j=1}^{\lceil N_Q/2 \rceil} \cos \left( \frac{x_{2j-1}}{2} \right) |0\rangle + e^{i \cdot x_{2j}} \sin \left( \frac{x_{2j-1}}{2} \right) |1\rangle. \quad (\text{A.7})$$

Therefore, we obtain the encoder function same as shown in equation (7).

To evaluate the effectiveness of the proposed dense encoding method (equation (7)) against the baseline angle encoding method (equation (1)), we provide a performance comparison using *linear probing* as a proxy task. As shown in figure A1, we denote performance results using dense encoding and the baseline angle encoding respectively as *Q Angle-8* and *Q Dense-8*, where 8 denotes the total number of qubits for a fair comparison. We also provide performance results for classical CLIP using different 512-to- $N_C$  compression layers, denoted as  $C N_C = 8$  and  $C N_C = 16$ . We find that QCLIP with  $N_Q = 8$  matches the performance of a



classical CLIP with  $N_C = 16$  trained on the same dataset, indicating the improved representation learning capability enhanced by QuNNs. The proposed dense encoding provides an average of 5.4% accuracy improvement compared to the baseline angle encoding. Moreover, increasing the width of the QuNN (i.e.  $N_Q$ ) improves the QCLIP accuracy, demonstrating the scalability of our approach.

We also explored *data re-uploading* [82] and *variational encoding* [83], which are two recently proposed encoding techniques to improve QuNN performance. The key idea of *data re-uploading* is to repeatedly apply the classical-to-quantum encoder,  $E(\mathbf{x})$ , before each parameterized VQC ansatz,  $U_k(\theta_k)$ . *Variational encoding* proposes to introduce trainable parameters to a classical-to-quantum encoder by defining a variational encoder function,  $E(\mathbf{x}; \theta)$ , where the parameter set  $\theta$  is pre-trained to produce faithful quantum presentations in which data from different clusters are separated. We refer interested readers to [82, 83] for a more detailed explanation and demonstration.

## Appendix B. QuNN circuit ansatzes

In this work, we survey the recently proposed QuNN circuit ansatzes and identify four designs that have demonstrated state-of-the-art performance as shown in figure B1. We denote these four designs respectively as *C14* [43], *QMLP* [44], *DAC22* [45], and *DATE22* [46]. These four ansatzes all follow the general structure summarized in figure 2 with a single-qubit rotation layer followed by a two-qubit entanglement layer. Specifically, *DATE22* adopt the early designs [25] that utilize fixed two-qubit CNOT gates to force maximum entangling power, while *C14*, *QMLP* and *DAC22* explore to replace CNOT with trainable entanglement two-qubit gates.

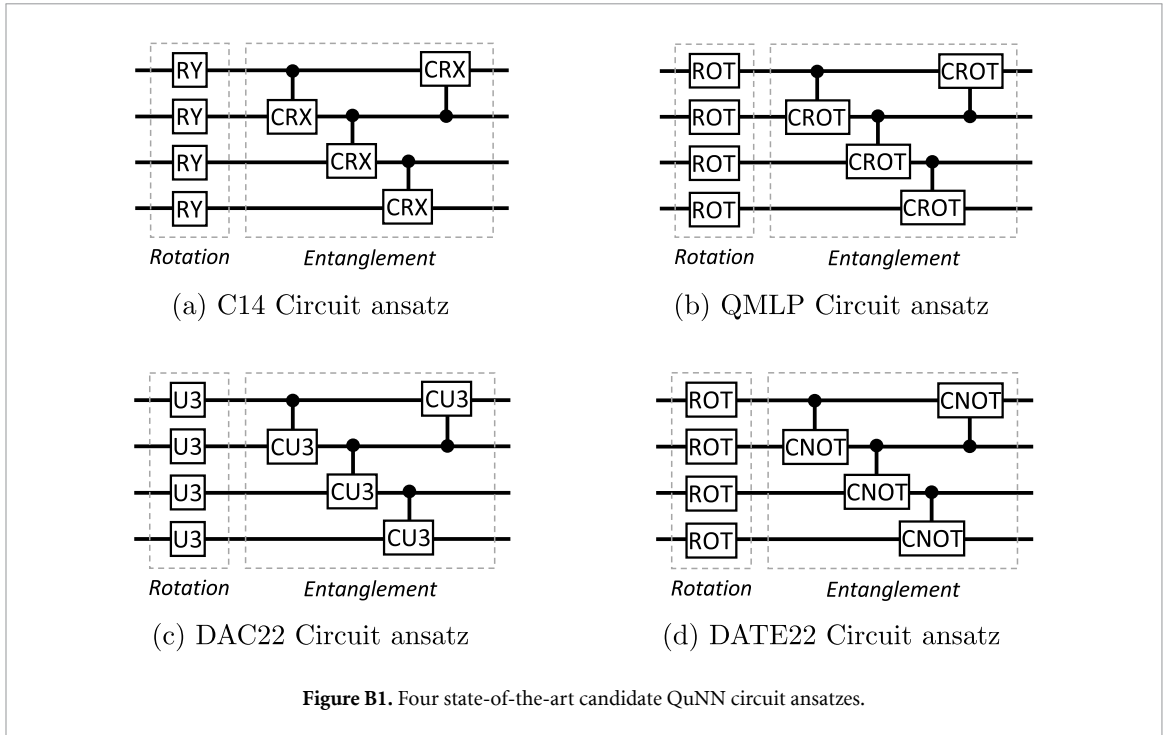
## Appendix C. Performance with different encoding methods and QuNNs

To investigate all the candidate quantum encoding methods (appendix A) and VQC ansatzes (appendix B) in the proposed QCLIP framework, we run various experiments using Cifar10 [51] as the downstream dataset. Based on the performance evaluation, we identify the optimal QuNN circuits for each quantum component in the proposed QCLIP framework, including the quantum image/text encoder neural networks and the quantum prompt adapter neural networks.

Taking the design of quantum image and text encoder as an example, we report the performance comparison for different circuit ansatz selections using *one-shot prompt learning*. As shown in table C1, we make two conclusions: (1) the proposed dense encoding method followed by the QuNN circuit ansatz in figure 4 achieves the best performance and thus is identified as the optimal QuNN implementation for the quantum image and text encoder neural networks. (2) *data re-uploading* [82] and *variational encoding* [83] achieves negligible accuracy improvement or even performance degradation in such a hybrid quantum–classical framework. Therefore, we do not recommend using these two methods in QCLIP.

We follow the similar approach described above and identify the optimal QuNN circuit for the quantum prompt adapter neural networks shown in figure 5.

**Theoretical insights.** Theoretical research [94] on data encoding interprets QML models as a Fourier-type sum, where the data encoding plays a crucial role in determining the functions the model can access and how these accessible functions can be combined. Consequently, the data encoding significantly influences the expressivity of the model. By applying this analysis, we propose that dense encoding

**Table C1.** QCLIP performance on one-shot prompt learning using different encoding methods and VQC circuit ansatzes.

Ansatz	Encoding Scheme		
	Dense	Reuploading	Variational
C14 [43]	<b>54.06%</b>	53.18%	53.31%
QMLP [44]	51.61%	<b>50.83%</b>	50.12%
DAC22 [45]	51.61%	51.42%	<b>52.25%</b>
DATE22 [46]	51.25%	<b>52.42%</b>	51.58%
QCLIP ( <b>This Work</b> )	55.62%	53.65%	53.42%

outperforms single-qubit rotation-based angle encoding, likely due to the two-layer RY-U1 gate in dense encoding, enabling access to frequency spectra with two frequencies, in contrast to angle encoding's single frequency. However, it is important to consider that increasing encoding density also leads to higher training complexity. Considering the problem set used in this work, we find the two-layer RY-U1 dense encoding to be the optimal choice for our QML models.

The theoretical analysis in VQC circuit ansatz [43] primarily explores the impact of circuit entanglement capacity on the expressivity of QML models. As of now, there is no universally agreed-upon optimal VQC design, and VQC circuits are typically empirically designed. However, there is a common consensus that adaptive and trainable entanglement capabilities can be beneficial for QML algorithms compared to fixed maximized entanglement provided by fixed CNOT gates.

## Appendix D. Loss functions

Here we formally define the contrastive loss [66, 67] that has also been explored in QCLIP training. Contrastive loss defines two loss functions named image-to-text contrastive loss, i.e.  $l_i^{(I \rightarrow T)}$ , and text-to-image contrastive loss, i.e.  $l_t^{(T \rightarrow I)}$ . The image-to-text contrastive loss for the  $i_{\text{th}}$  image and the text-to-image contrastive loss for the  $t_{\text{th}}$  text can be calculated by the following equations (D.1) and (D.2), where  $i = 1, 2, \dots, B$  labels the input image feature in a batch and  $\tau \in B^+$  represents a temperature parameter.

$$l_i^{(I \rightarrow T)} = -\log \frac{\exp(\langle I_i, T_t \rangle / \tau)}{\sum_{t=1}^B \exp(\langle I_i, T_t \rangle / \tau)} \quad (\text{D.1})$$

$$l_t^{(T \rightarrow I)} = -\log \frac{\exp(\langle T_t, I_i \rangle / \tau)}{\sum_{i=1}^B \exp(\langle T_t, I_i \rangle / \tau)}. \quad (\text{D.2})$$



The final training loss is defined as the weighted sum of the above two losses. For batch training, the averaged loss is calculated using the following equation (D.3), where  $\lambda \in [0, 1]$  is a scaling hyperparameter.

$$\text{loss} = \frac{1}{B} \sum_{p=1}^B (\lambda l_p^{(I \rightarrow T)} + (1 - \lambda) l_p^{(T \rightarrow I)}). \quad (\text{D.3})$$

## Appendix E. Parameter initialization

We study both *uniform* initialization and *Gaussian* initialization in QCLIP training. Below we provide details for each initialization method.

Uniform initialization generates the initial values for the trainable parameters from Uniform distribution. The general formulation is shown in equation (E.1) with a minimal value  $a$  and a maximal value  $b$ . In QCLIP, we set the minimal and maximal values respectively to 0 and  $\frac{\pi}{2}$ , as shown in equation (E.2).

$$f(x) = \begin{cases} \frac{1}{a-b}, & a < x < b \\ 0, & \text{else.} \end{cases} \quad (\text{E.1})$$

$$f(\text{weight}) = \begin{cases} \frac{2}{\pi}, & 0 < \text{weight} < \frac{\pi}{2} \\ 0, & \text{else.} \end{cases} \quad (\text{E.2})$$

Gaussian initialization generates initial values from a Gaussian distribution. We show the general formulation for Gaussian distribution in equation (E.3) with a mean value  $\mu$  and a standard deviation  $\sigma$ . Inspired by classical Xavier initialization [89], we utilize the information of QuNN structures and initialize parameters according to the network width  $N_Q$ . The Gaussian initialization used in QCLIP can be formalized by equation (E.4).

$$X \sim \mathcal{N}(\mu, \sigma^2), f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (\text{E.3})$$

$$\text{Weight} \sim \mathcal{N}(0, N_Q), f(\text{weight}) = \frac{1}{\sqrt{2\pi N_Q}} \exp\left(-\frac{\text{weight}^2}{2N_Q}\right). \quad (\text{E.4})$$

## ORCID iDs

Philip Richerme  <https://orcid.org/0000-0003-1799-7612>

Fan Chen  <https://orcid.org/0000-0001-7928-8331>

## References

- [1] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79
- [2] IonQ Forte (available at: <https://ionq.com/quantum-systems/forte/>)
- [3] IBM Quantum Heron (available at: <https://research.ibm.com/blog/ibm-quantum-roadmap-2025/>)
- [4] Shor P W 1999 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM Rev.* **41** 303–32
- [5] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. 28th Annual ACM Symp. on Theory of Computing* pp 212–9
- [6] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev. A* **86** 032324
- [7] Childs A M, Maslov D, Nam Y, Ross N J and Su Y 2018 Toward the first quantum simulation with quantum speedup *Proc. Natl Acad. Sci.* **115** 9456–61
- [8] Campbell E, Khurana A and Montanaro A 2019 Applying quantum algorithms to constraint satisfaction problems *Quantum* **3** 167
- [9] Kivlichan I D, Gidney C, Berry D W, Wiebe N, McClean J, Sun W, Jiang Z, Rubin N, Fowler A and Aspuru-Guzik A 2020 Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization *Quantum* **4** 296
- [10] Gidney C and Ekerå M 2021 How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits *Quantum* **5** 433
- [11] Lee J, Berry D W, Gidney C, Huggins W J, McClean J R, Wiebe N and Babbush R 2021 Even more efficient quantum computations of chemistry through tensor hypercontraction *PRX Quantum* **2** 030305
- [12] Lemieux J, Duclos-Cianci G, Sénéchal D and Poulin D 2021 Resource estimate for quantum many-body ground-state preparation on a quantum computer *Phys. Rev. A* **103** 052408
- [13] Shor P W 1995 Scheme for reducing decoherence in quantum computer memory *Phys. Rev. A* **52** R2493–6
- [14] Lidar D A and Brun T A 2013 *Quantum Error Correction* (Cambridge University Press)
- [15] Terhal B M 2015 Quantum error correction for quantum memories *Rev. Mod. Phys.* **87** 307–46

- [16] Bharti K, Cervera-Lierta A, Kyaw T H, Haug T, Alperin-Lea S, Anand A, Degroote M, Heimonen H, Kottmann J S and Menke T 2022 Noisy intermediate-scale quantum algorithms *Rev. Mod. Phys.* **94** 015004
- [17] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
- [18] Cao Y, Romero J, Olson J P, Degroote M, Johnson P D, Kieferová M, Kivlichan I D, Menke T, Peropadre B and Sawaya N P 2019 Quantum chemistry in the age of quantum computing *Chem. Rev.* **119** 10856–915
- [19] Endo S, Cai Z, Benjamin S C and Yuan X 2021 Hybrid quantum-classical algorithms and quantum error mitigation *J. Phys. Soc. Japan* **90** 032001
- [20] McArdle S, Endo S, Aspuru-Guzik A, Benjamin S C and Yuan X 2020 Quantum computational chemistry *Rev. Mod. Phys.* **92** 015003
- [21] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O’Brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 1–7
- [22] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6
- [23] Cerezo M, Sharma K, Arrasmith A and Coles P J 2020 Variational quantum state eigensolver (arXiv:2004.01372)
- [24] Huang H Y et al 2022 Quantum advantage in learning from experiments *Science* **376** 1182–6
- [25] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
- [26] Lloyd S, Mohseni M and Rebentrost P 2013 Quantum algorithms for supervised and unsupervised machine learning (arXiv:1307.0411)
- [27] Schuld M and Killoran N 2019 Quantum machine learning in feature hilbert spaces *Phys. Rev. Lett.* **122** 040504
- [28] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
- [29] Huang H Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 Power of data in quantum machine learning *Nat. Commun.* **12** 1–9
- [30] Lloyd S and Weedbrook C 2018 Quantum generative adversarial learning *Phys. Rev. Lett.* **121** 040502
- [31] Dallaire-Demers P L and Killoran N 2018 Quantum generative adversarial networks *Phys. Rev. A* **98** 012324
- [32] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
- [33] Xia R and Kais S 2018 Quantum machine learning for electronic structure calculations *Nat. Commun.* **9** 1–6
- [34] Choudhary K, Bercx M, Jiang J, Pachter R, Lamoen D and Tavazza F 2019 Accelerated discovery of efficient solar cell materials using quantum and machine-learning methods *Chem. Mater.* **31** 5900–8
- [35] Cao Y, Romero J and Aspuru-Guzik A 2018 Potential of quantum computing for drug discovery *IBM J. Res. Dev.* **62** 6–1
- [36] Amin J, Sharif M, Gul N, Kadry S and Chakraborty C 2022 Quantum machine learning architecture for covid-19 classification based on synthetic data generation using conditional adversarial neural network *Cogn. Comput.* **14** 1677–88
- [37] Alcazar J, Leyton-Ortega V and Perdomo-Ortiz A 2020 Classical versus quantum models in machine learning: insights from a finance application *Mach. Learn.: Sci. Technol.* **1** 035003
- [38] Coyle B, Henderson M, Le J C J, Kumar N, Paini M and Kashefi E 2021 Quantum versus classical generative modeling in finance *Quantum Sci. Technol.* **6** 024013
- [39] Parsons D F 2011 Possible medical and biomedical uses of quantum computing *Neuroquantology* **9** 596–600
- [40] Crawford S E, Shugayev R A, Paudel H P, Lu P, Syamlal M, Ohodnicki P R, Chorpene B, Gentry R and Duan Y 2021 Quantum sensing for energy applications: Review and perspective *Adv. Quantum Technol.* **4** 2100049
- [41] Focardi S, Fabozzi F J and Mazza D 2020 Quantum option pricing and quantum finance *J. Derivatives* **28** 79–98
- [42] Bengio Y, Courville A C and Vincent P 2013 Representation learning: a review and new perspectives *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1798–828
- [43] Sim S, Johnson P D and Aspuru-Guzik A 2019 Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms *Adv. Quantum Technol.* **2** 1900070
- [44] Chu C, Chia N, Jiang L and Chen F 2022 QMLP: an error-tolerant nonlinear quantum MLP architecture using parameterized two-qubit gates *ACM/IEEE Int. Symp. on Low Power Electronics and Design (ISLPED) (ACM)* vol 4 pp 1–4:6
- [45] Wang H, Gu J, Ding Y, Li Z, Chong F T, Pan D Z and Han S 2022 Quantumnat: quantum noise-aware training with noise injection, quantization and normalization *DAC ’22: 59th ACM/IEEE Design Automation Conf. (ACM) (San Francisco, California, USA, 10–14 July 2022)* pp 1–6
- [46] Patel T, Silver D and Tiwari D 2022 OPTIC: A practical quantum binary classifier for near-term quantum computers *Design, Automation & Test in Europe Conf. & Exhibition (DATE) (IEEE)* pp 334–9
- [47] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 Circuit-centric quantum classifiers *Phys. Rev. A* **101** 032308
- [48] Patel T, Silver D and Tiwari D 2022 Optic: a practical quantum binary classifier for near-term quantum computers *2022 Design, Automation & Test in Europe Conf. & Exhibition (DATE) (IEEE)* pp 334–9
- [49] Niu M Y, Zlokapa A, Broughton M, Boixo S, Mohseni M, Smelyanskiy V and Neven H 2022 Entangling quantum generative adversarial networks *Phys. Rev. Lett.* **128** 220505
- [50] Kübler J M, Arrasmith A, Cincio L and Coles P J 2020 An adaptive optimizer for measurement-frugal variational algorithms *Quantum* **4** 263
- [51] Krizhevsky A, Hinton G et al 2009 *Learning Multiple Layers of Features From Tiny Images* (Toronto, ON) (available at: [www.cs.toronto.ca/~kriz/learning-features-2009-TR.pdf](http://www.cs.toronto.ca/~kriz/learning-features-2009-TR.pdf))
- [52] Radford A et al 2021 Learning transferable visual models from natural language supervision *Int. Conf. on Machine Learning (ICML) (Proc. Machine Learning Research (PMLR)* vol 139) pp 8748–63
- [53] Zeiler M D and Fergus R 2014 Visualizing and understanding convolutional networks *European Conf. on Computer Vision (ECCV)* vol 8689 (Springer) pp 818–33
- [54] Razavian A S, Azizpour H, Sullivan J and Carlsson S 2014 CNN features off-the-shelf: an astounding baseline for recognition *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (IEEE Computer Society)* pp 512–9
- [55] LeCun Y, Bengio Y and Hinton G E 2015 Deep learning *Nature* **521** 436–44
- [56] Grover A and Leskovec J 2016 node2vec: scalable feature learning for networks *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (ACM)* pp 855–64
- [57] Chen X, Duan Y, Houthoofd R, Schulman J, Sutskever I and Abbeel P 2016 Infogan: interpretable representation learning by information maximizing generative adversarial nets *Advances in Neural Information Processing Systems (NIPS)* pp 2172–80

- [58] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D and Batra D 2017 Grad-cam: Visual explanations from deep networks via gradient-based localization *IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE Computer Society) pp 618–26
- [59] Hjelm R D, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A and Bengio Y 2019 Learning deep representations by mutual information estimation and maximization *Int. Conf. on Learning Representations (ICLR)* (OpenReview.net) (available at: <https://openreview.net/forum?id=Bklr3j0cKX>)
- [60] Kingma D P and Welling M 2014 Auto-encoding variational bayes *Int. Conf. on Learning Representations (ICLR)* ed Y Bengio and Y LeCun
- [61] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A C and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems (NIPS)* pp 2672–80
- [62] Doersch C, Gupta A and Efros A A 2015 Unsupervised visual representation learning by context prediction *IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE Computer Society) pp 1422–30
- [63] Zhang R, Isola P and Efros A A 2016 Colorful image colorization *European Conf. on Computer Vision (ECCV) (Lecture Notes in Computer Science vol 9907)* ed B Leibe, J Matas, N Sebe and M Welling (Springer) pp 649–66
- [64] Gidaris S, Singh P and Komodakis N 2018 Unsupervised representation learning by predicting image rotations *Int. Conf. on Learning Representations (ICLR)* (OpenReview.net)
- [65] Bachman P, Hjelm R D and Buchwalter W 2019 Learning representations by maximizing mutual information across views *Advances in Neural Information Processing Systems (NeurIPS)* pp 15509–19
- [66] Chen T, Kornblith S, Norouzi M and Hinton G E 2020 A simple framework for contrastive learning of visual representations *Proc. 37th Int. Conf. on Machine Learning (ICML) (Proc. Machine Learning Research (PMLR) vol 119)* pp 1597–607
- [67] Chen T, Kornblith S, Swersky K, Norouzi M and Hinton G E 2020 Big self-supervised models are strong semi-supervised learners *Advances in Neural Information Processing Systems (NeurIPS)*
- [68] Quattoni A, Collins M and Darrell T 2007 Learning visual representations using images with captions *Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society)
- [69] Srivastava N and Salakhutdinov R 2012 Multimodal learning with deep boltzmann machines *Annual Conf. on Neural Information Processing Systems (NIPS)* ed P L Bartlett, F C N Pereira, C J C Burges, L Bottou and K Q Weinberger pp 2231–9
- [70] Joulin A, van der Maaten L, Jabri A and Vasilache N 2016 Learning visual features from large weakly supervised data *European Conf. on Computer Vision (ECCV) (Lecture Notes in Computer Science vol 9911)* (Springer) pp 67–84
- [71] Li A, Jabri A, Joulin A and van der Maaten L 2017 Learning visual n-grams from web data *IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE Computer Society) pp 4193–202
- [72] Desai K and Johnson J 2021 Virtex: Learning visual representations from textual annotations *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Computer Vision Foundation/IEEE) pp 11162–73
- [73] Locatello F, Bauer S, Lucic M, Rätsch G, Gelly S, Schölkopf B and Bachem O 2020 A sober look at the unsupervised learning of disentangled representations and their evaluation *J. Mach. Learn. Res.* **21** 1–62
- [74] Elsayed G F, Goodfellow I and Sohl-Dickstein J 2018 Adversarial reprogramming of neural networks (arXiv:1806.11146)
- [75] Li X L and Liang P 2021 Prefix-tuning: optimizing continuous prompts for generation (arXiv:2101.00190)
- [76] Bahng H, Jahanian A, Sankaranarayanan S and Isola P 2022 Visual prompting: modifying pixel space to adapt pre-trained models (arXiv:2203.17274)
- [77] LaRose R and Coyle B 2020 Robust data encodings for quantum classifiers *CoRR* (arXiv:2003.01695)
- [78] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023
- [79] Radford A, Wu J, Child R, Luan D, Amodei D and Sutskever I 2019 *OpenAI blog* OpenAI (available at: [https://d4mucfpksywv.cloudfront.net/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf))
- [80] Mari A, Bromley T R, Izaac J A, Schuld M and Killoran N 2020 Transfer learning in hybrid classical-quantum neural networks *Quantum* **4** 340
- [81] Jaderberg B, Anderson L W, Xie W, Albanie S, Kiffner M and Jaksch D 2022 Quantum self-supervised learning *Quantum Sci. Technol.* **7** 035005
- [82] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E and Latorre J I 2020 Data re-uploading for a universal quantum classifier *Quantum* **4** 226
- [83] Chu C, Skipper G, Swamy M and Chen F 2022 IQGAN: robust quantum generative adversarial network for image synthesis on NISQ devices (arXiv:2210.16857)
- [84] Sharma P, Ding N, Goodman S and Soricut R 2018 Conceptual captions: a cleaned, hypernymed, image alt-text dataset for automatic image captioning *Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (Association for Computational Linguistics) pp 2556–65
- [85] Zhang Y, Jiang H, Miura Y, Manning C D and Langlotz C P 2020 Contrastive learning of medical visual representations from paired images and text *CoRR* (arXiv:2010.00747)
- [86] PyTorch PyTorch (available at: <https://pytorch.org/>)
- [87] Pennylane (available at: <https://pennylane.ai/>)
- [88] Zhang K, Hsieh M, Liu L and Tao D 2022 Gaussian initializations help deep variational quantum circuits escape from the barren plateau *CoRR* (arXiv:2203.09376)
- [89] Kumar S K 2017 On weight initialization in deep neural networks (arXiv:1704.08863)
- [90] LeCun Y 1998 The mnist database of handwritten digits (available at: <http://yann.lecun.com/exdb/mnist/>)
- [91] Parkhi O M, Vedaldi A, Zisserman A and Jawahar C V 2012 Cats and dogs 2012 *IEEE Conf. on Computer Vision and Pattern Recognition* pp 3498–505
- [92] Bossard L, Guillaumin M and Van Gool L 2014 Food-101—mining discriminative components with random forests *European Conference on Computer Vision* (Springer) pp 446–61
- [93] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parametrized quantum circuits *Quantum* **3** 214
- [94] Schuld M, Sweke R and Meyer J J 2021 Effect of data encoding on the expressive power of variational quantum-machine-learning models *Phys. Rev. A* **103** 032430