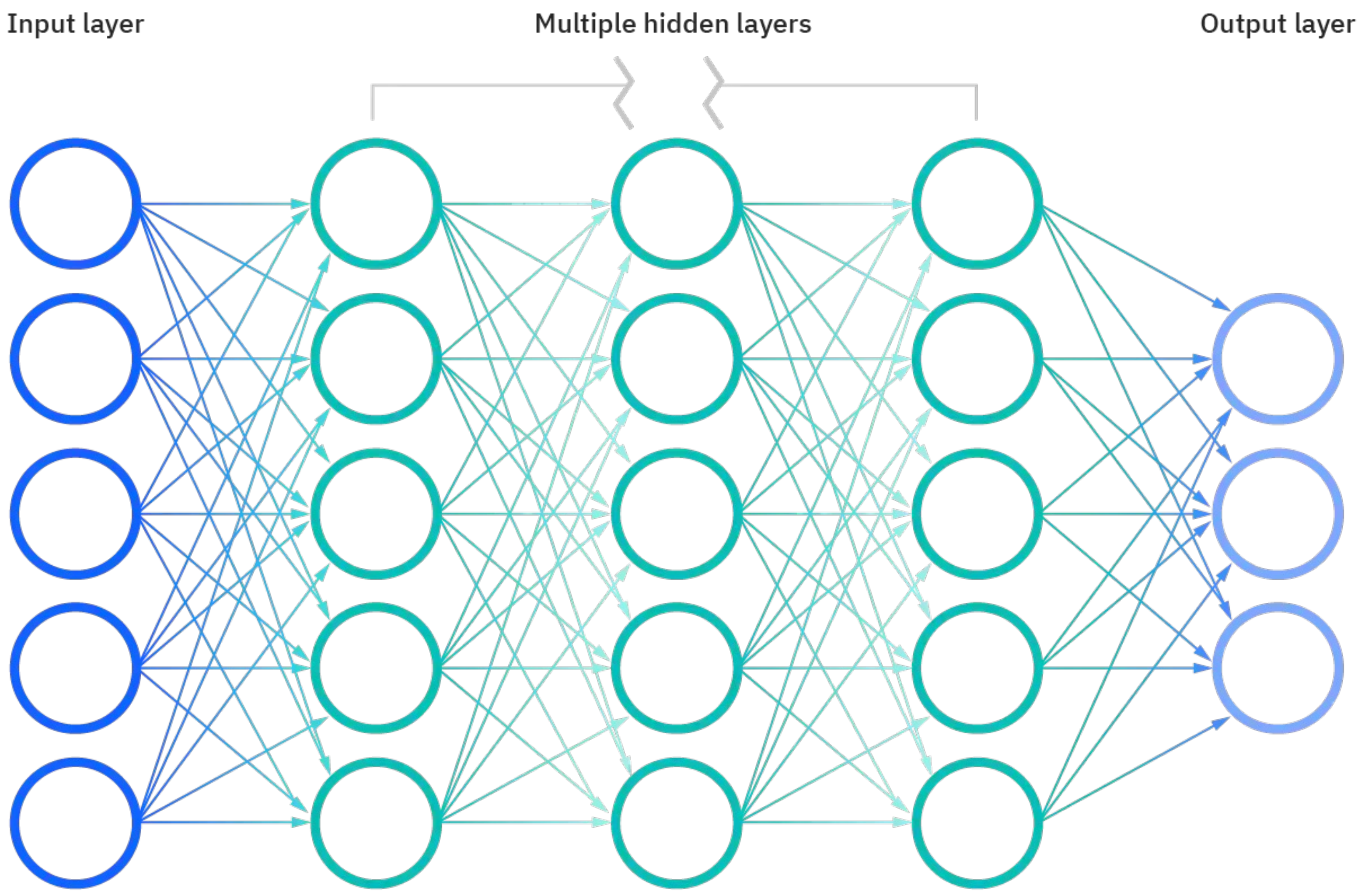
Quanvolutional neural network

14/2/2023

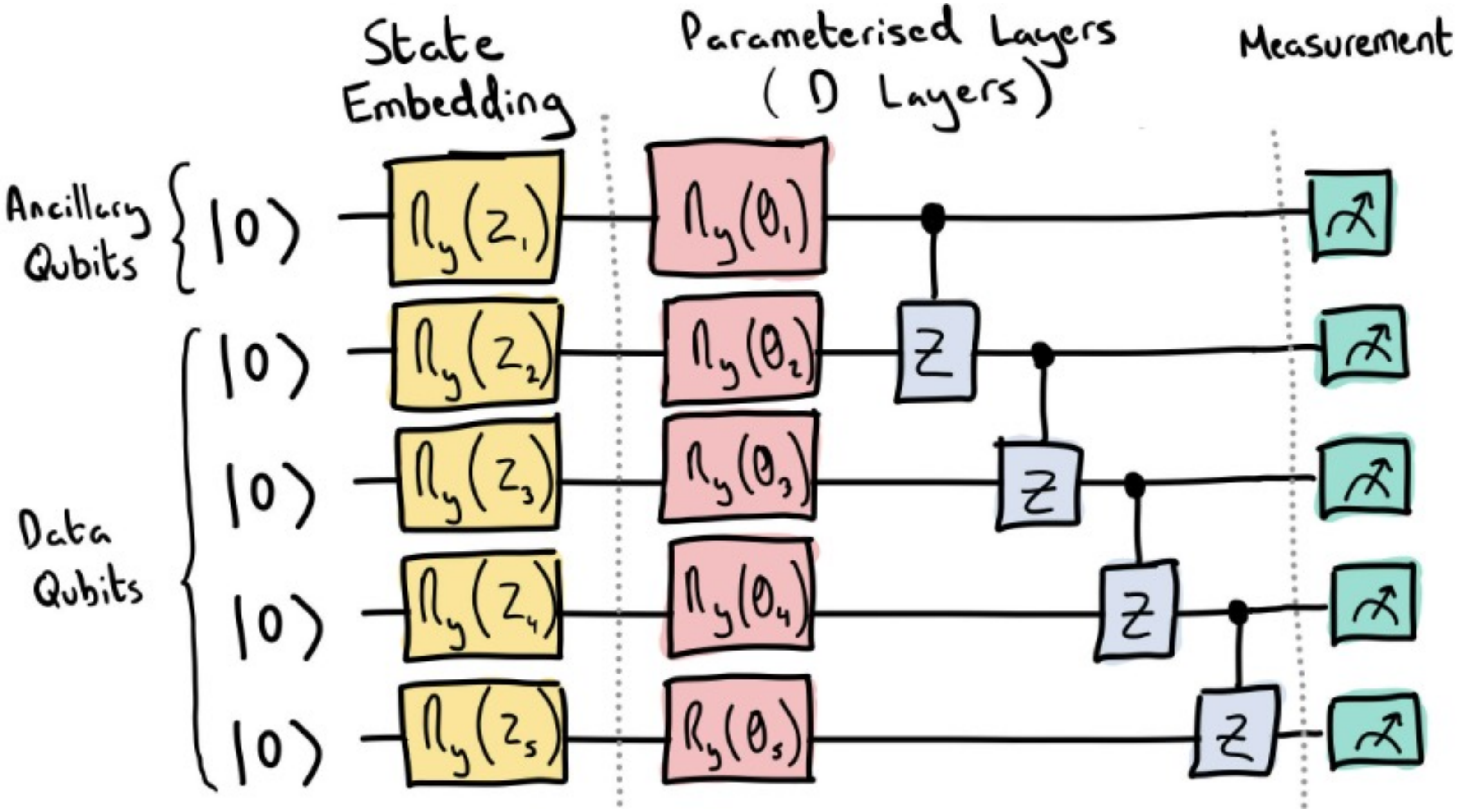
https://arxiv.org/abs/1904.04767

https://arxiv.org/abs/2108.00661.pdf

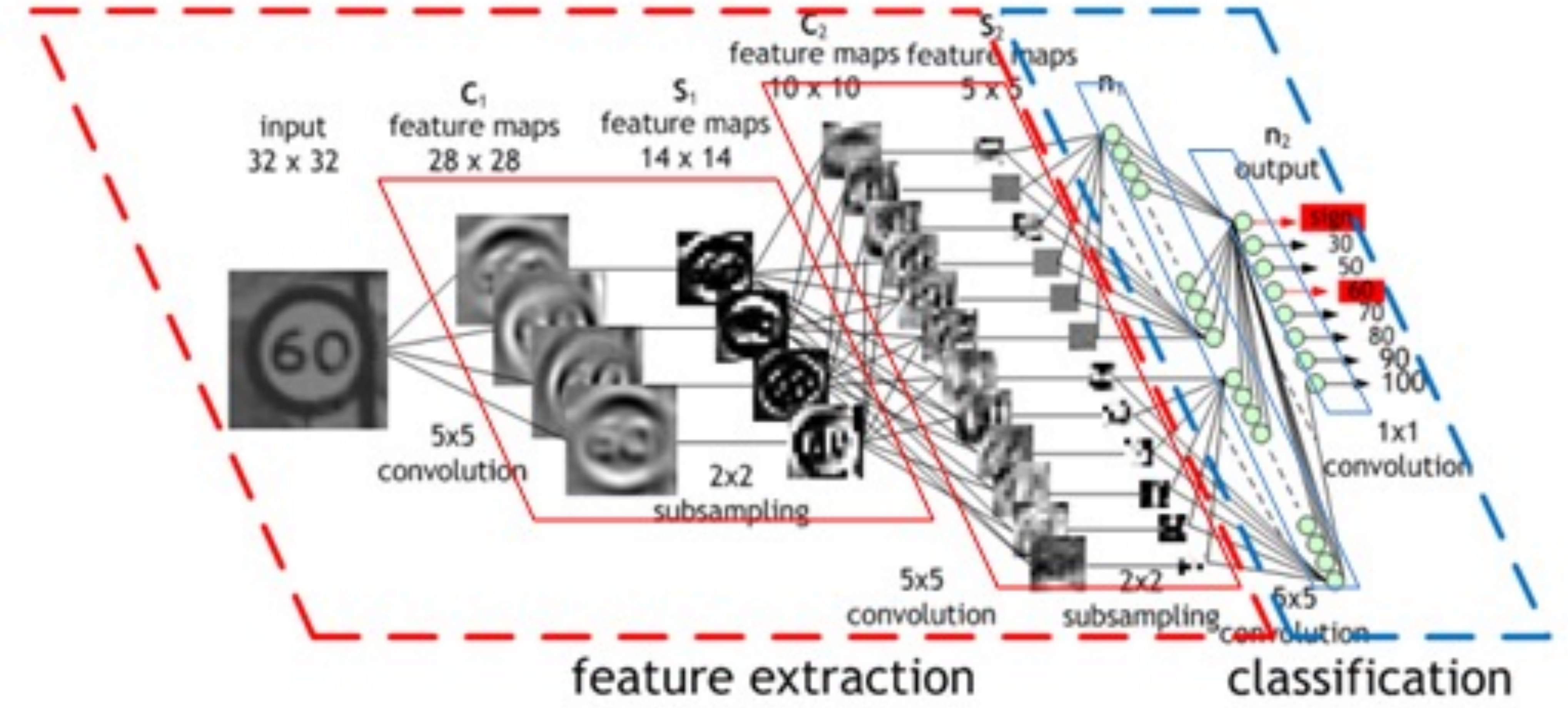
Neural network



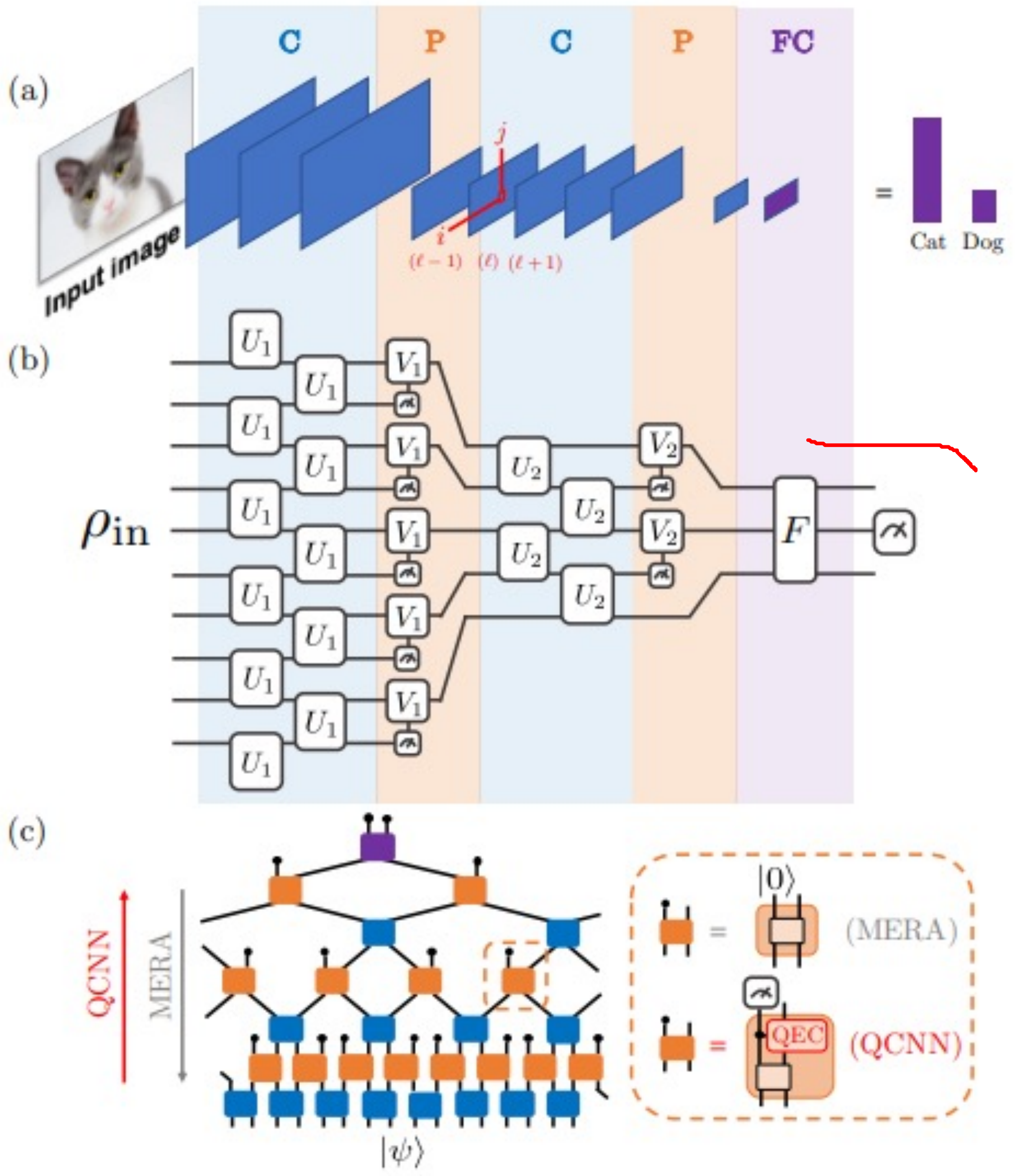
Quantum neural network



Convolutional neural network



Quantum convolutional neural network



Absence of Barren Plateaus in Quantum Convolutional Neural Networks, PRX, 11,041011 (2021)

Overall structural

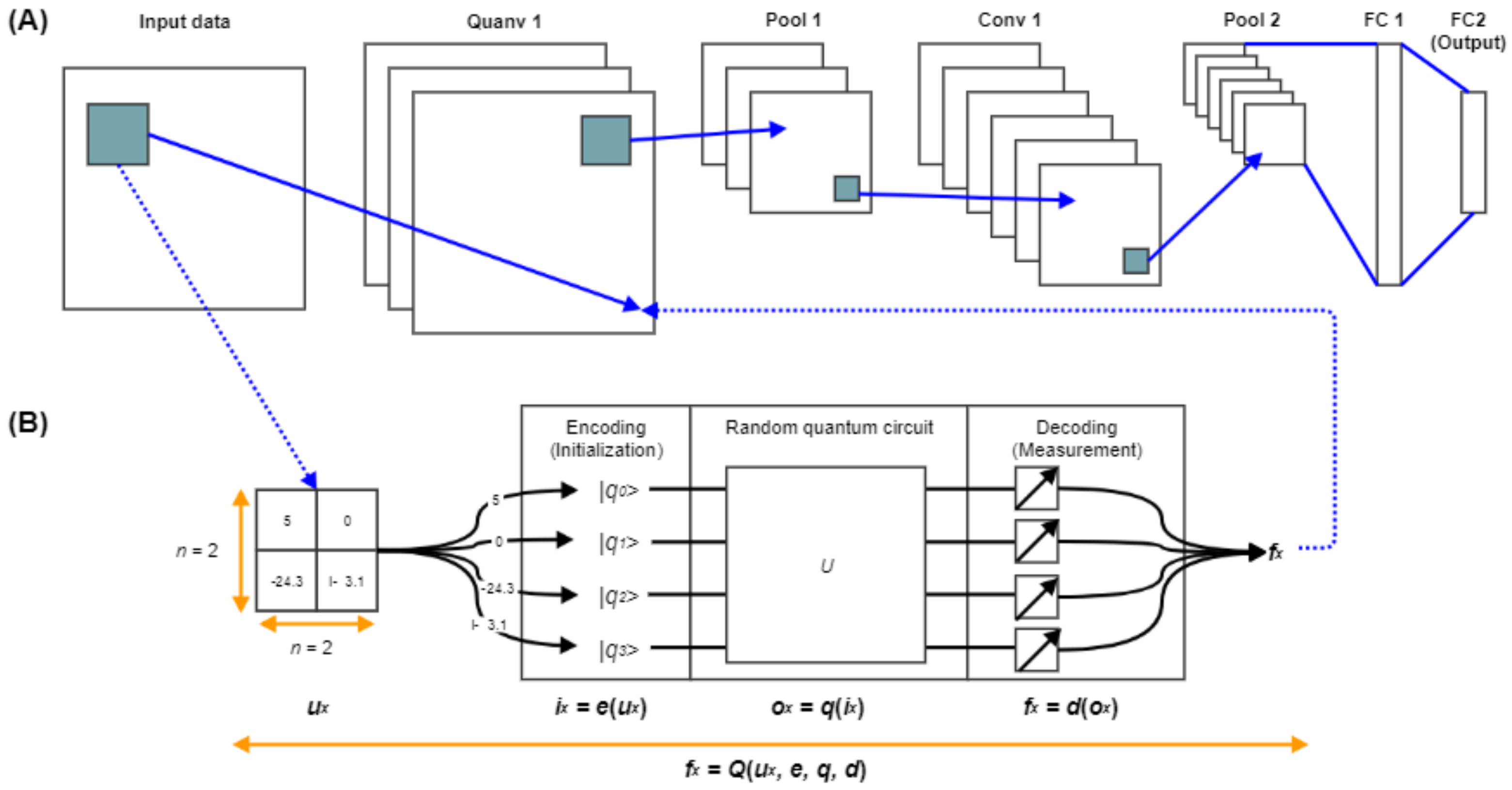
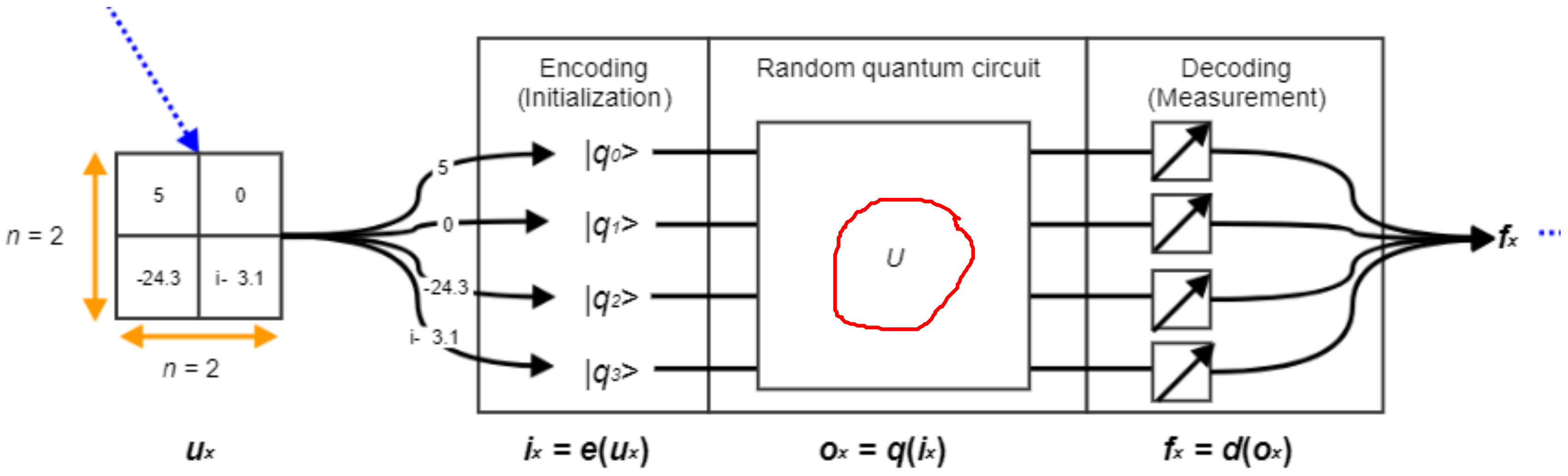


Fig. 1.: A. Simple example of a quanvolutional layer in a full network stack. The quanvolutional layer contains several quanvolutional filters (three in this example) that transform the input data into different output feature maps. B. An in-depth look at the processing of classical data into and out of the random quantum circuit in the quanvolutional filter.

Quanvolutional layer



Quanvolutional layer = $f(u_x, \boldsymbol{e}, \boldsymbol{q}, \boldsymbol{d})$: $\mathbb{R}^{nxn} \rightarrow \mathbb{R}$ where:

- u_x: patch
- e: encoder
- q: random quantum circuit
- d: decoder

Encoding & Decoding

$e: \mathbb{R}^{nxn} \rightarrow |\psi\rangle$

Amplitude encoding

- $\log N$ qubits but exponentially number of gates
- N qubit, linear number of gate

Threshold encoding: if pixel value is less than threshold t, the according qubits will be $|0\rangle$ and vice versa.

$d: |\psi\rangle \rightarrow \mathbb{R}$

Quantum circuit

The number of qubits is 9.

Chose $(0 \to 2n^2$) random 1 qubit gate.

$$X(\theta), Y(\theta), Z(\theta), U(\theta), P, T, H$$

θ is random in $[0,2\pi]$

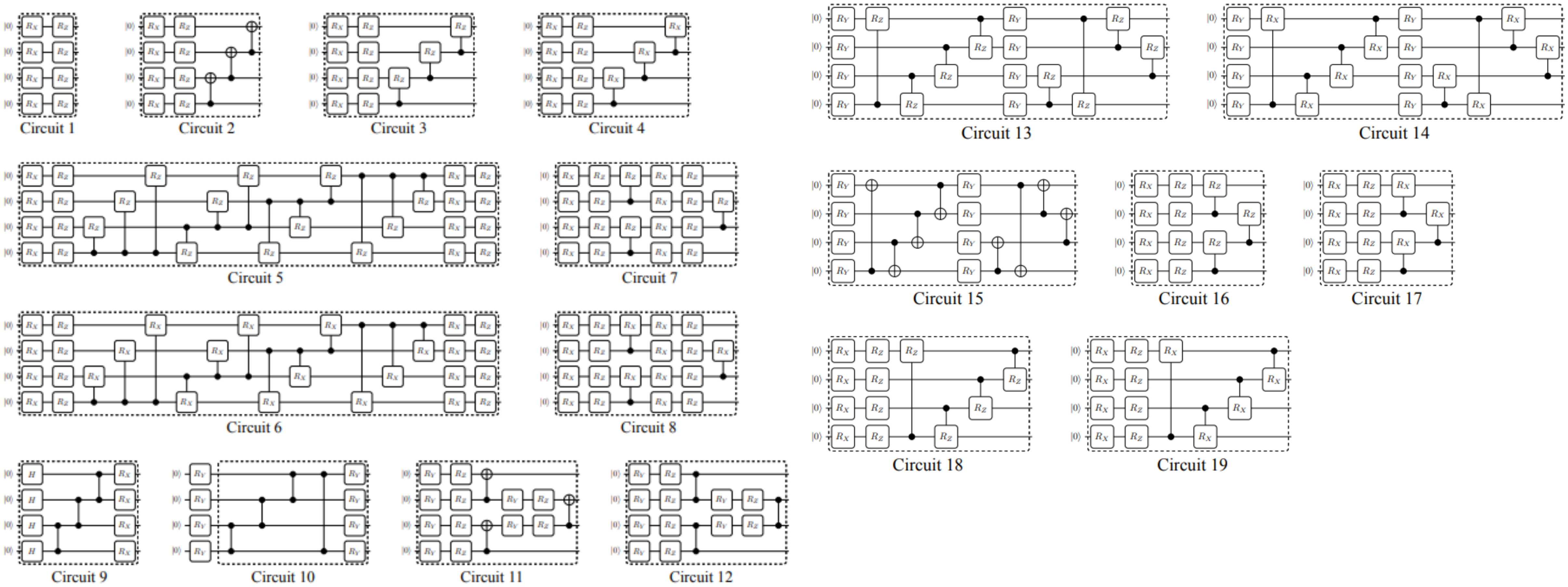
And random 2 qubits gate (to make entanglement)

→ The set of gates is suffled => One quanvolutional layer

Entangled properties in quantum filter acts as a compresser.

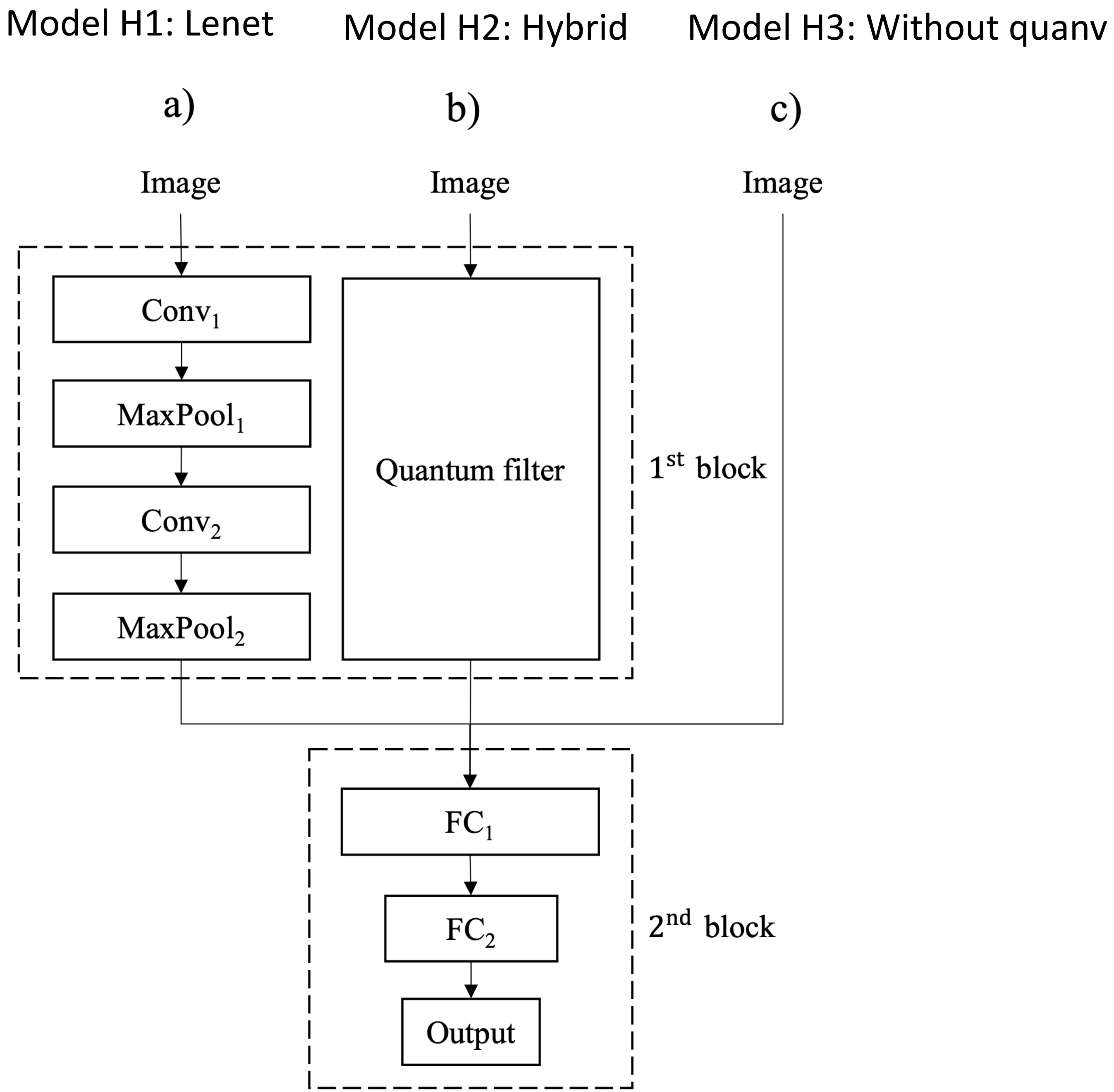
More entangle, less number of iteration.

Quantum circuit



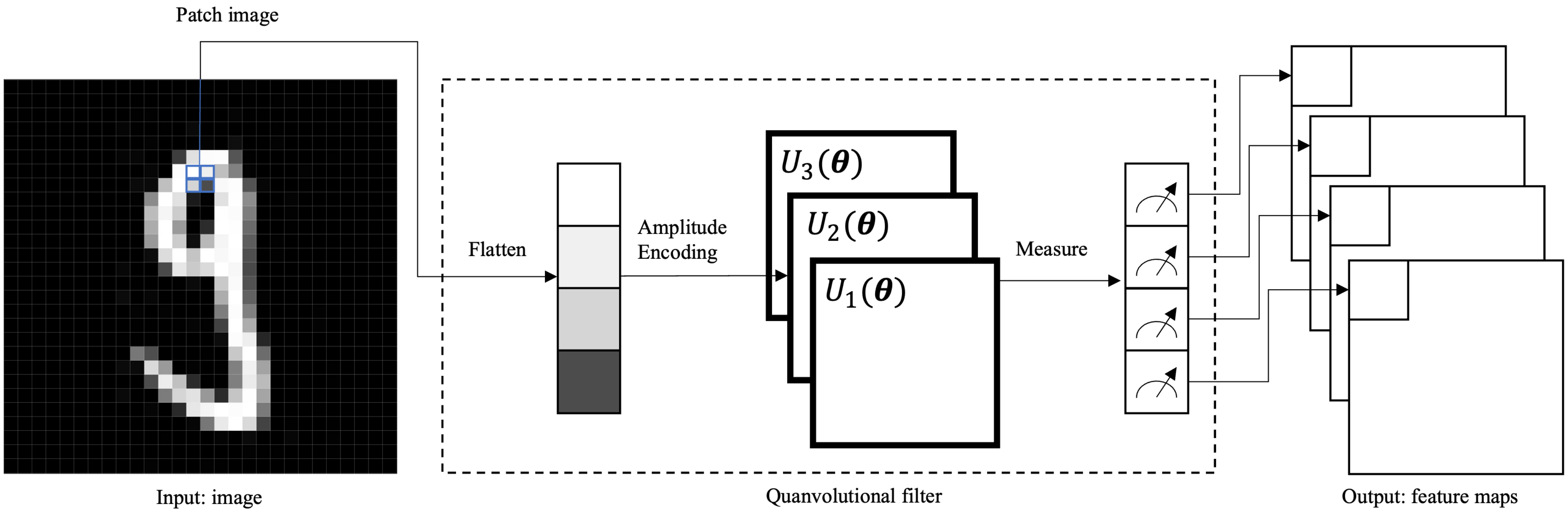
Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms

Model



Model H1: Lenet
Model H2: Hybrid
Model H3: Without quanv
a)
b)
c)
Image
Image
Image
Conv1
MaxPool1
Conv2
MaxPool2
Quantum filter
1st block
FC1
FC2
Output
2nd block

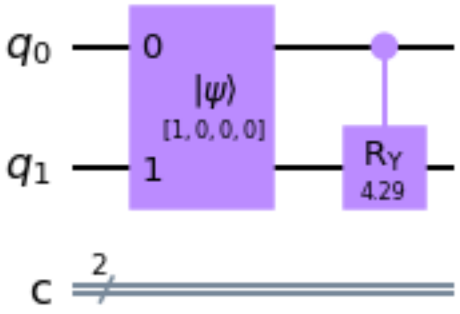
Scheme


Patch image
Flatten
Amplitude
Encoding
$U_3(\boldsymbol{\theta})$
$U_2(\boldsymbol{\theta})$
$U_1(\boldsymbol{\theta})$
Measure
Input: image
Quanvolutional filter
Output: feature maps

Quanv layer / filter

$$\begin{bmatrix}255\\0\\0\\0\end{bmatrix} \xrightarrow[\text{Normalize}]{} \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \xrightarrow[\text{Load}]{}$$

Random R_Y gate



2 x 2 filter

$$\xrightarrow[\text{Measure}]{} \begin{bmatrix}0.77\\0.61\\0\\0.71\end{bmatrix}$$

Probability vector

Circuit

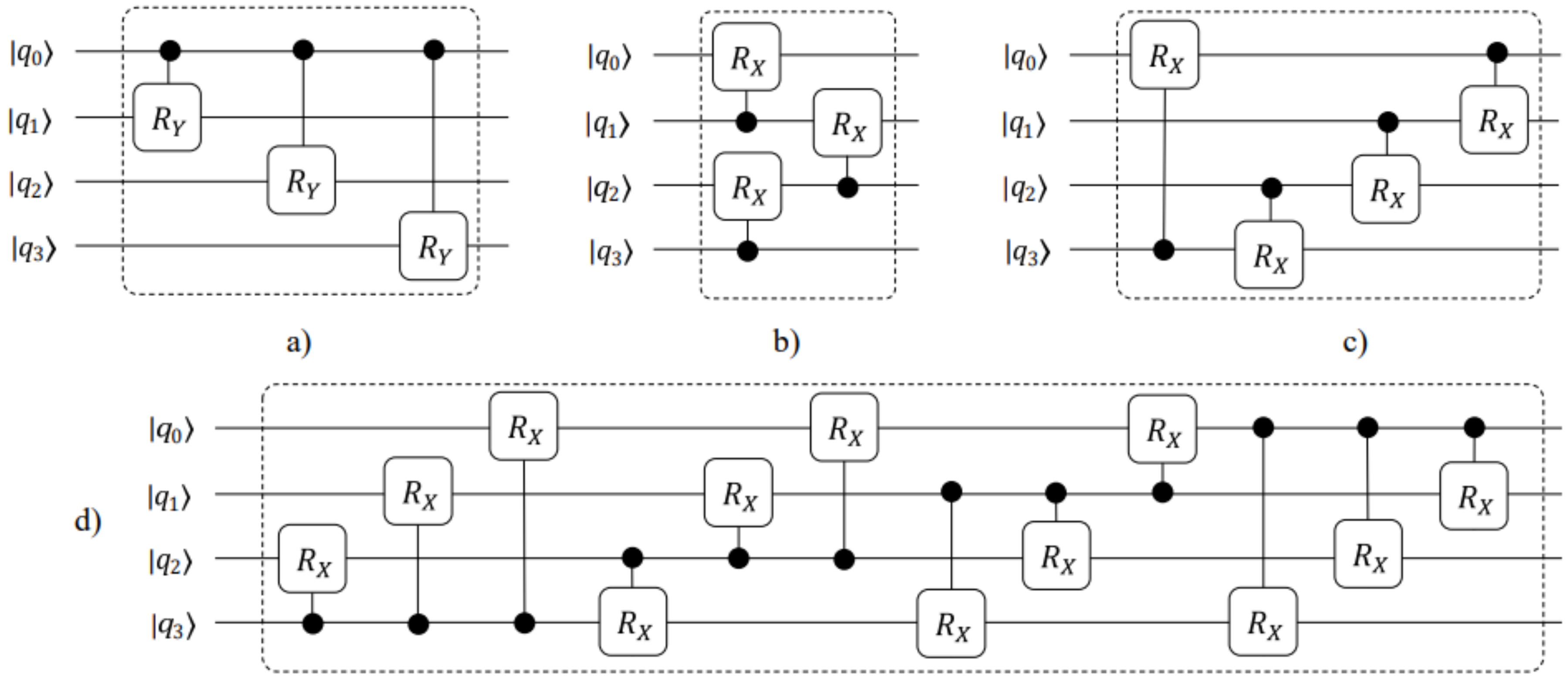
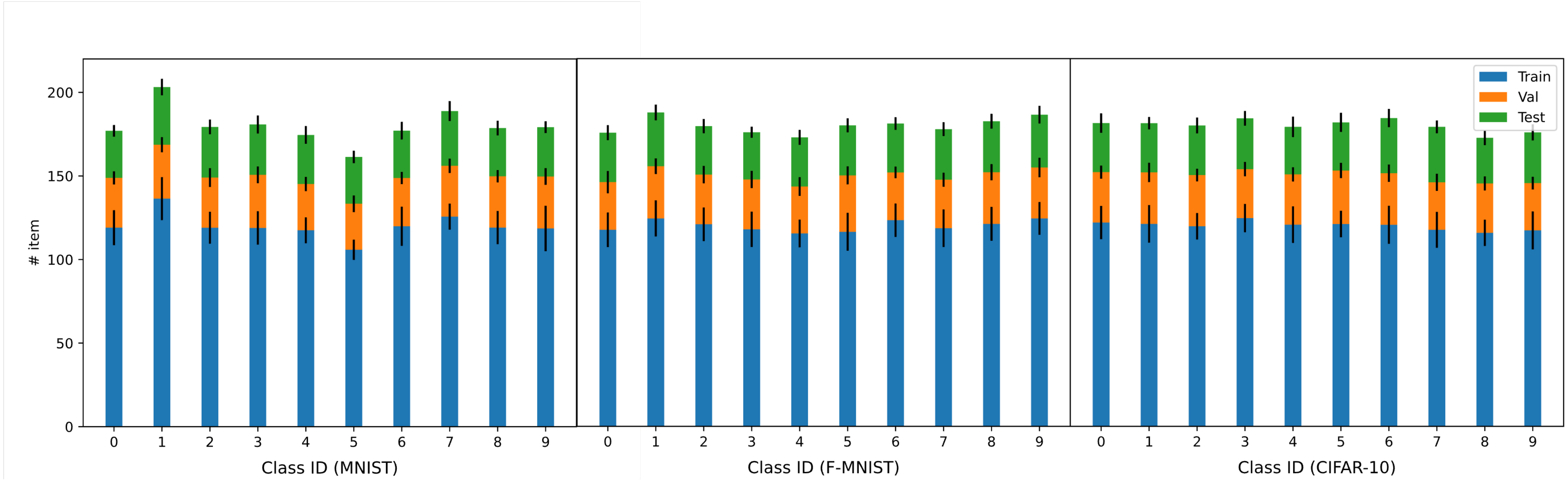


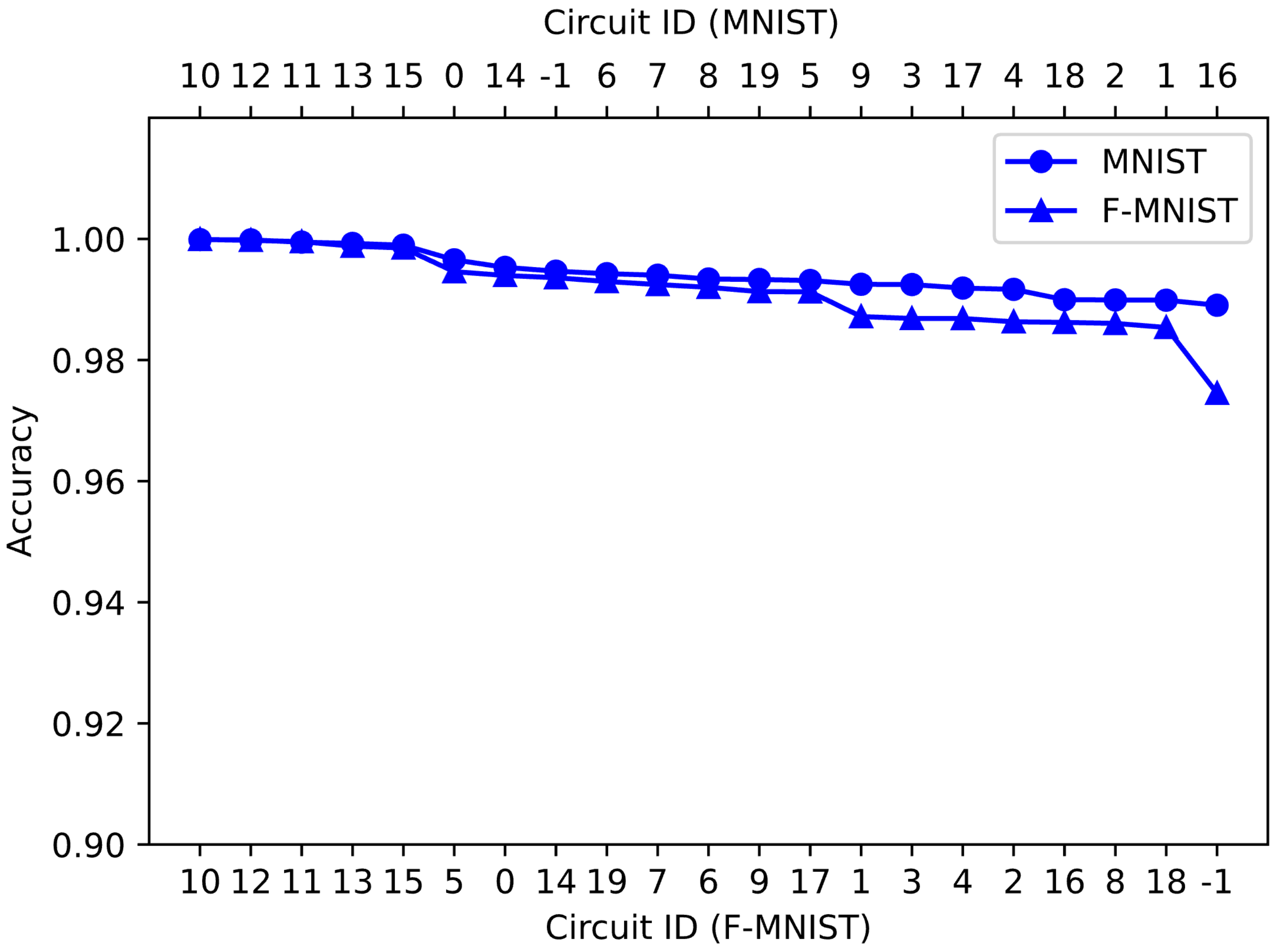
Fig. 2. Four considered configurations: (a) GS, the default circuit in all experiments (b) NN, the arrangement of a linear array of two-qubit operations (c) CB, the setup for a closed loop-forming array of qubits and (d) AA, fully connected graph structure of qubit.

Dataset

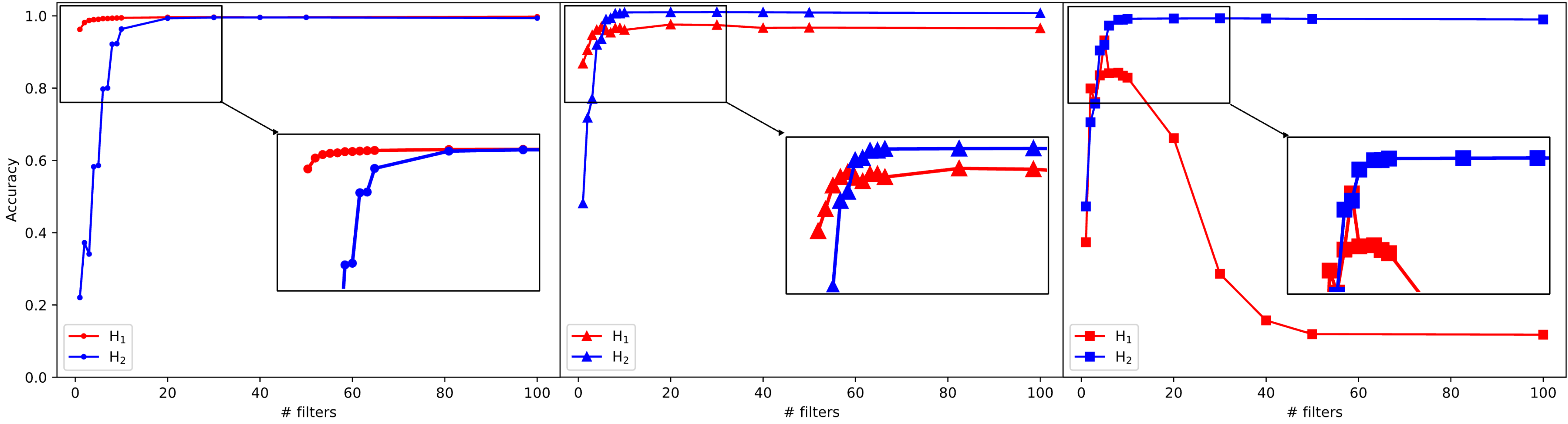
- 28x28 gray MNIST and FMNIST (60k train, 10k test)
- 32x32 gray CIFAR-10



1. Different between quanv filters (4x4, 4 filter)



2. Different between number of filter



Train and test on MNIST

Train and test on MNIST Fashion

Train and test on CIFAR-10

From 1 to 100 filters

3. Compare #. pram

Table 1. A comparison of top-accuracy between models on three datasets and total parameter on the first block, with filter size 2×2 and 4×4.

Model	MNIST	F-MNIST	CIFAR-10	# params
2×2				
H_1 (1 filter)	0.8548	0.9197	0.5608	10
H_1 (20 filters)	**0.9956**	**0.9869**	0.9843	1720
H_1 (50 filters)	0.9951	0.9848	**0.9876**	10300
H_2 (20 filters)	0.9550	0.9727	0.7559	5
H_3	0.9894	0.5264	0.1128	$\emptyset$
4×4				
H_1 (1 filter)	0.9598	0.8492	0.3748	34
H_1 (20 filters)	0.9950	0.9539	0.6636	6720
H_1 (50 filters)	**0.9960**	0.9482	0.1195	40900
H_2 (20 filters)	0.9932	**0.9960**	**0.8654**	6
H_3	0.9894	0.5264	0.1180	$\emptyset$

4. Filter size

Table 3. The accuracy on MNIST (left, ● and ●), F-MNIST (center, ▲ and ▲) and CIFAR-10 (left, ■ and ■) datasets with filter size from 2 to 5 and respect the number of the required qubit.

Filter size	H_1 (●)	H_2 (●)	H_1 (▲)	H_2 (▲)	H_1 (■)	H_2 (■)	# required qubit
2	**0.9943**	0.6972	**0.9893**	0.9278	**0.9772**	0.5903	2
3	0.9924	0.7525	0.9726	0.8337	0.9518	0.6233	4
4	0.9895	**0.9965**	0.9479	**0.9939**	0.8381	**0.8608**	4
5	0.9894	0.4123	0.9557	0.7099	0.5059	0.5779	5

Advanced of quanv filter

We increase the number of quanv filter by adding more random circuit.

One random circuit n qubit is equivalent to 2^n filter $2^n x 2^n$. That means we can increase the number of quanv filter rapidly.

Quantum computers can access kernel functions in highdimensional Hilbert spaces much more efficiently than classical computers.

→ Coverage faster

Disadvatanges

- Number of measurements
- Number of quanvolutional filters

Future works

- Test on another dataset and deeper QNN.
- Make quanvolutional layer trainable or change by time by the evolutional / genetic algorithm. => Use QNG in quanvolutional layer

Properties of quantum machine learning

TABLE I. A comparison of typical properties of problems studied in quantum computing versus problems solved by machine-learning algorithms. Looking at this table, it is no surprise that quantum machine learning is a tough candidate for applications with a quantum advantage.

Property	Problems studied in quantum computing	Problems solved by machine learning
Classical performance	*Low*—problems are carefully selected to be provably difficult for classical computers	*High*—machine learning is applied on an industrial scale and many algorithms run in linear time in practice
Size of inputs	*Small*—near-term algorithms are limited by small qubit numbers, while fault-tolerant algorithms usually take short bit strings	*Very large*—may be millions of tensors with millions of entries each
Problem structure	*Very structured*—often exhibiting a periodic structure that can be exploited by interference	"*Messy*"—problems are derived from the human or "real-world" domain and are naturally complex to state and analyze
Theoretical accessibility	*High*—there is a large bias toward problems about which we can theoretically reason	*Shifting*—theory is currently being rebuilt around the empirical success of deep learning
Evaluating performance	*Computational complexity*—the dominant measure to assess the performance of an algorithm is asymptotic run-time scaling	*Practical benchmarks*—machine-learning research puts a strong emphasis on empirical comparisons between methods

Quantum advantages

(a) The asymptotic run time of a particular machine-learning algorithm; for example, an optimizer used to solve the empirical risk-minimization problem in Eq. (3) [41–43,46]
(b) Whether or not a learning problem (such as the one in Definition 1) is efficiently solvable for a particular data distribution $p(x)$ [70,71]
(c) The expressivity of a model class $\mathcal{F}$ [59,72]
(d) The number M of samples needed to learn [64,73]
(e) Average or worst-case generalization errors (which measure the difference between expected and empirical loss) [60–62]
(f) The structure of the optimization landscape, giving us an idea of how easy it is to solve Eq. (3) with gradient-based methods [56,57]
(g) The test error on some small-scale practical benchmark [7,74,75].