Quantum Autoencoder for MNIST classification

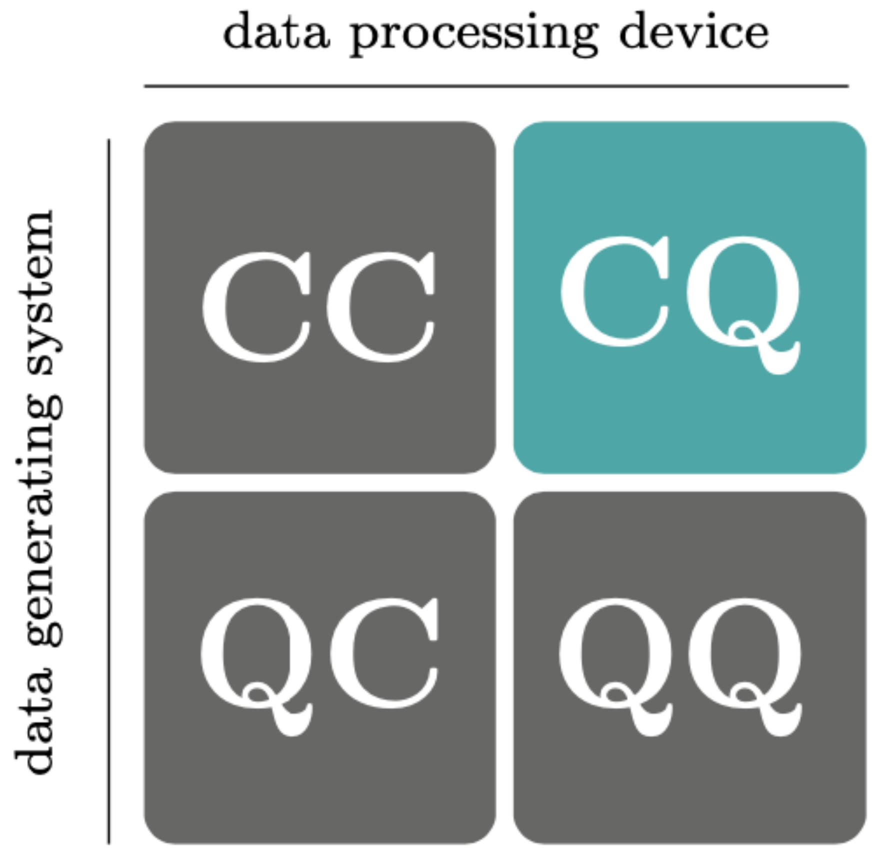
Mosè Guerini

Quantum Computing and Technologies

Academic year 2023/2024

Quantum Machine Learning

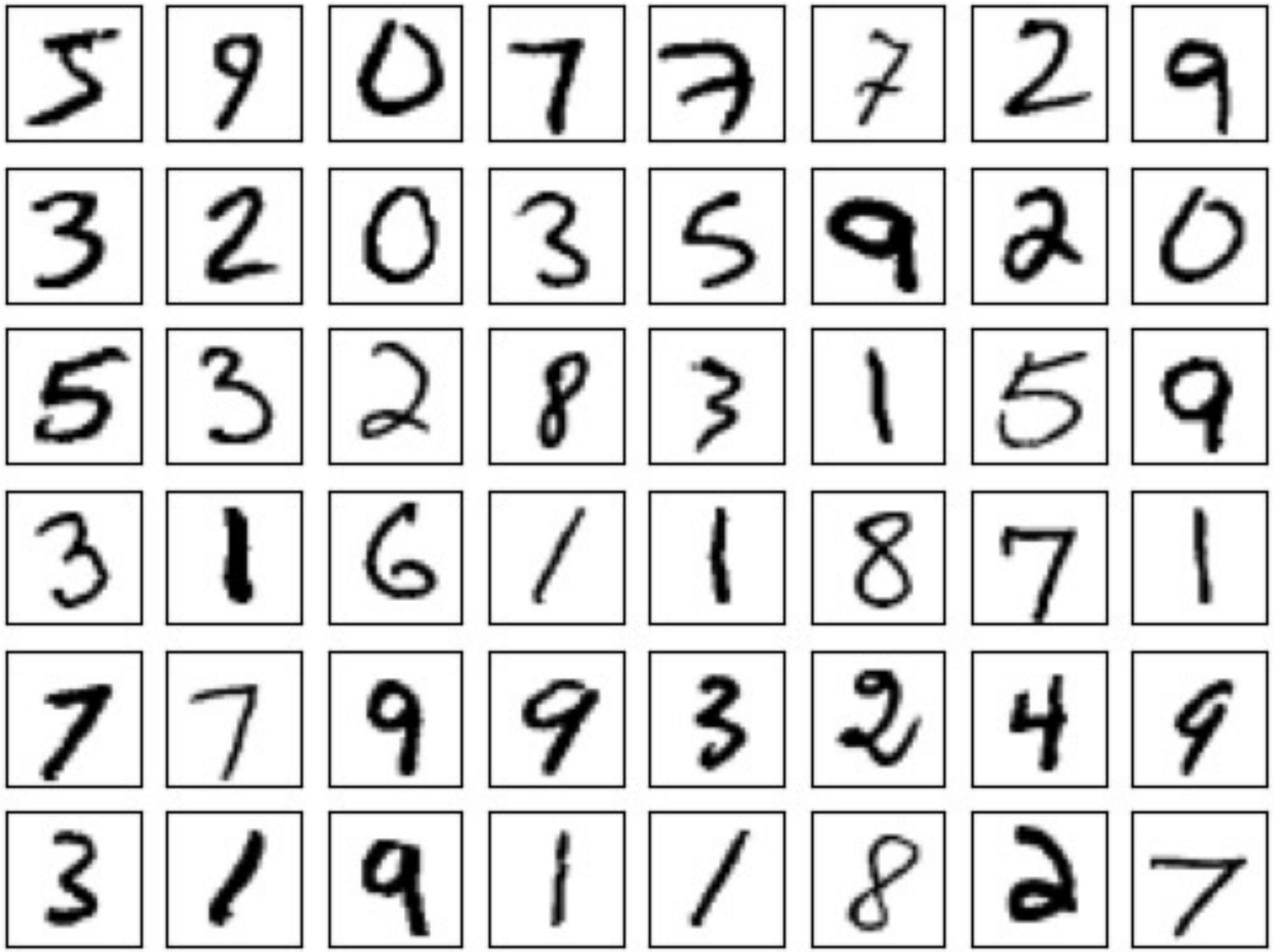
- **ML** ⟶ making computers learn from data how to solve problems
- **QC** ⟶ information processing with devices based on QM



The **Project**:

1. MNIST database
2. PCA reduction
3. Quantum Autoencoder
4. Random Forest Classifier

MNIST database

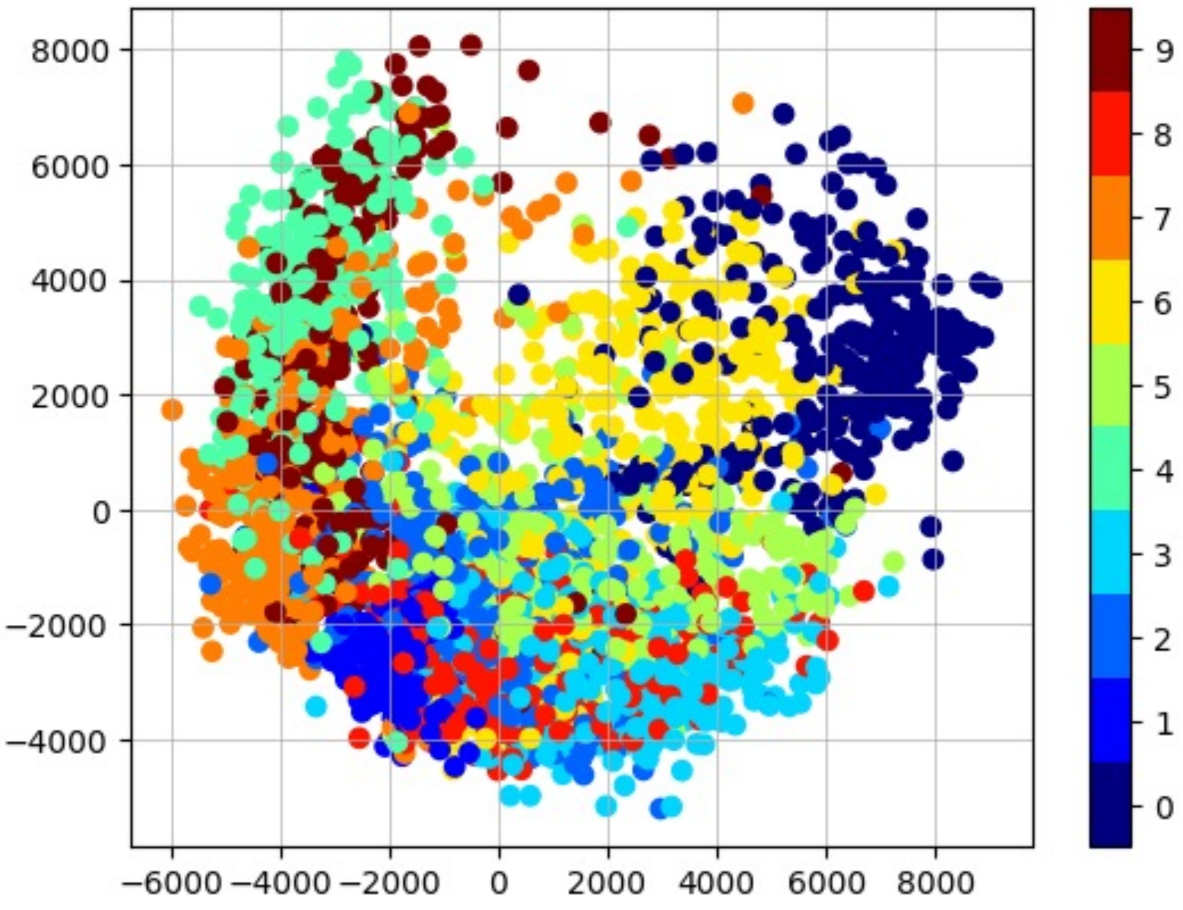


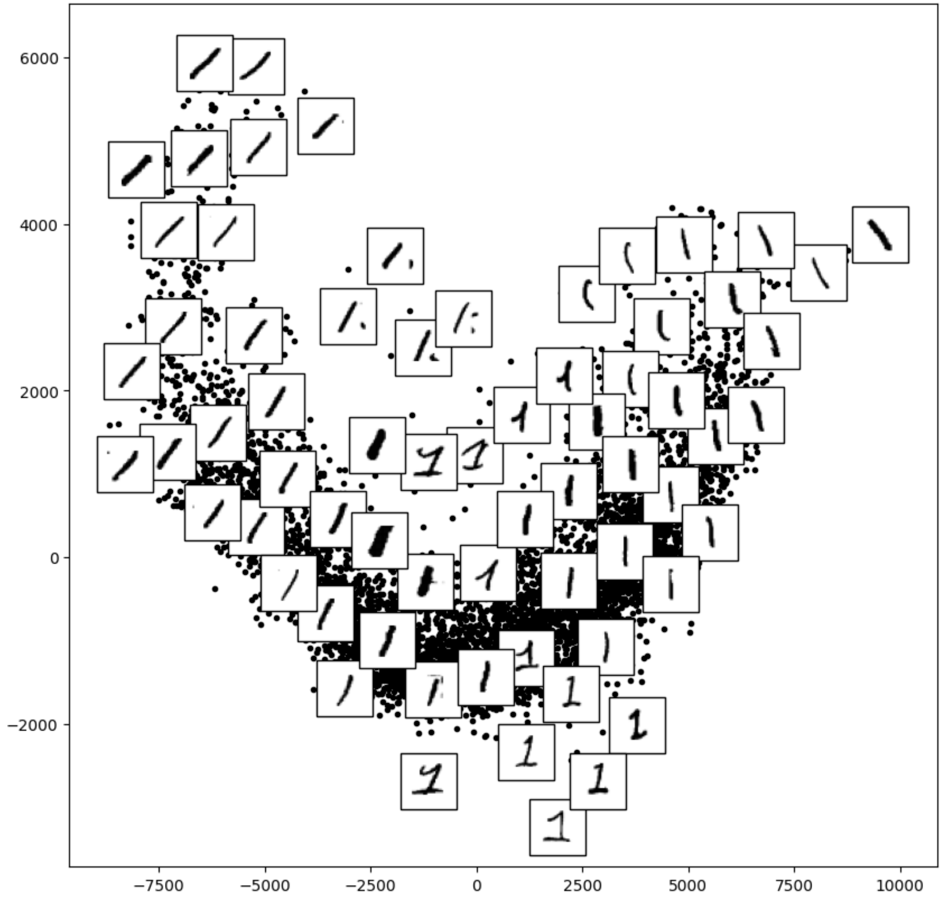
- 70.000 handwritten digits
- 10 classes
- Images 28×28 pixels

$\Longrightarrow$ **784 features**

Isomap

784-D parameter space → Projection to 2-D parameter space





Principal Component Analysis (PCA)

Pixel basis $\longrightarrow$ $\text{image}(x) = x_1 \cdot \text{pixel } 1 + x_2 \cdot \text{pixel } 2 + \cdots + x_{784} \cdot \text{pixel } 784$

PCA basis $\longrightarrow$ $\text{image}(x) = \text{mean} + x_1 \cdot \text{basis } 1 + x_2 \cdot \text{basis } 2 + \cdots + x_{784} \cdot \text{basis } 784$

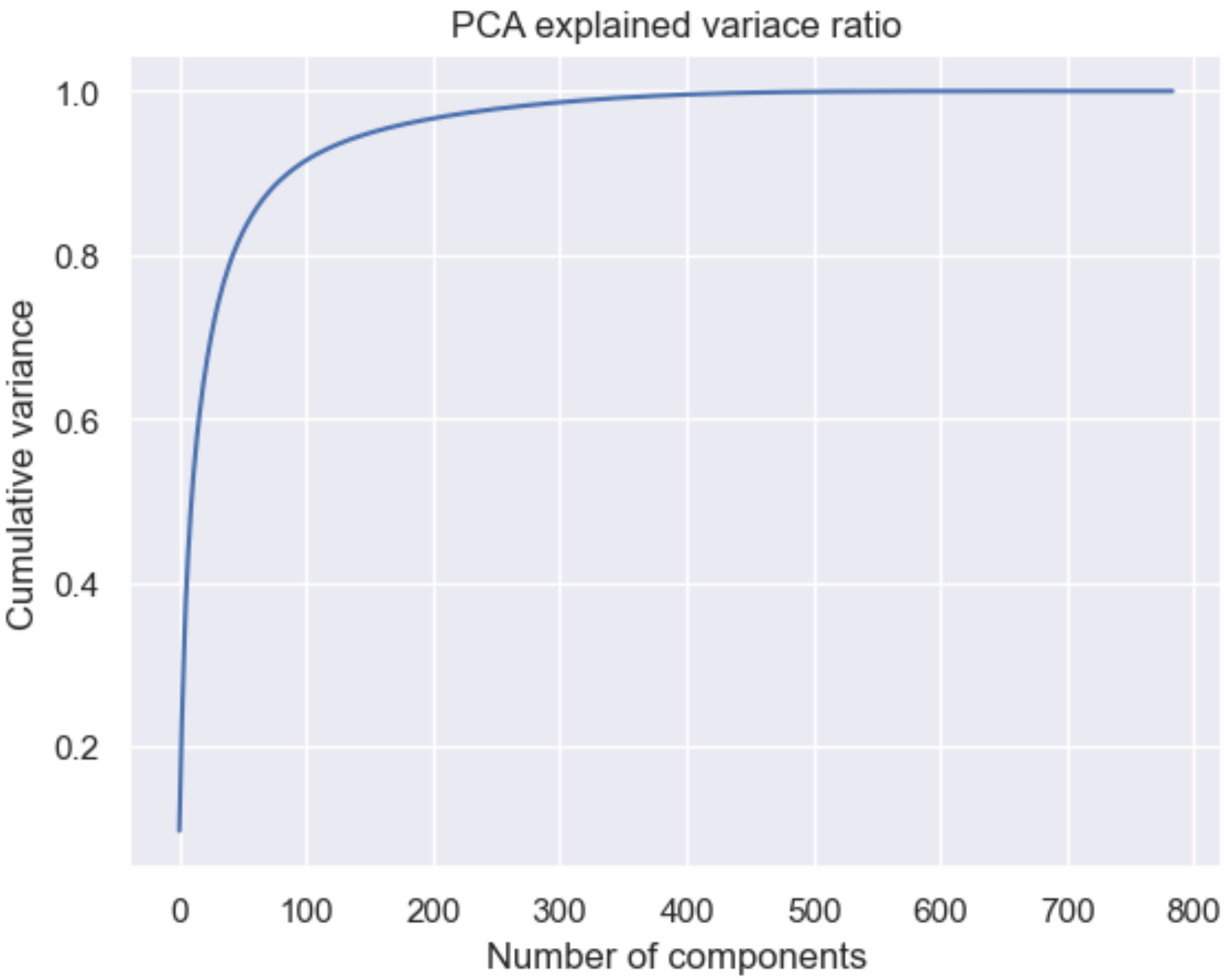
1. **Covariance matrix**: $\begin{bmatrix} Cov(x_1, x_1) & \cdots & Cov(x_1, x_{784}) \\ \vdots & \ddots & \vdots \\ Cov(x_{784}, x_1) & \cdots & Cov(x_{784}, x_{784}) \end{bmatrix}$

2. **Diagonalization:**

Eigenvectors — Direction where there is the most variance (Principal Components)

Eigenvalues λ_i — Amount of variance carried in each direction

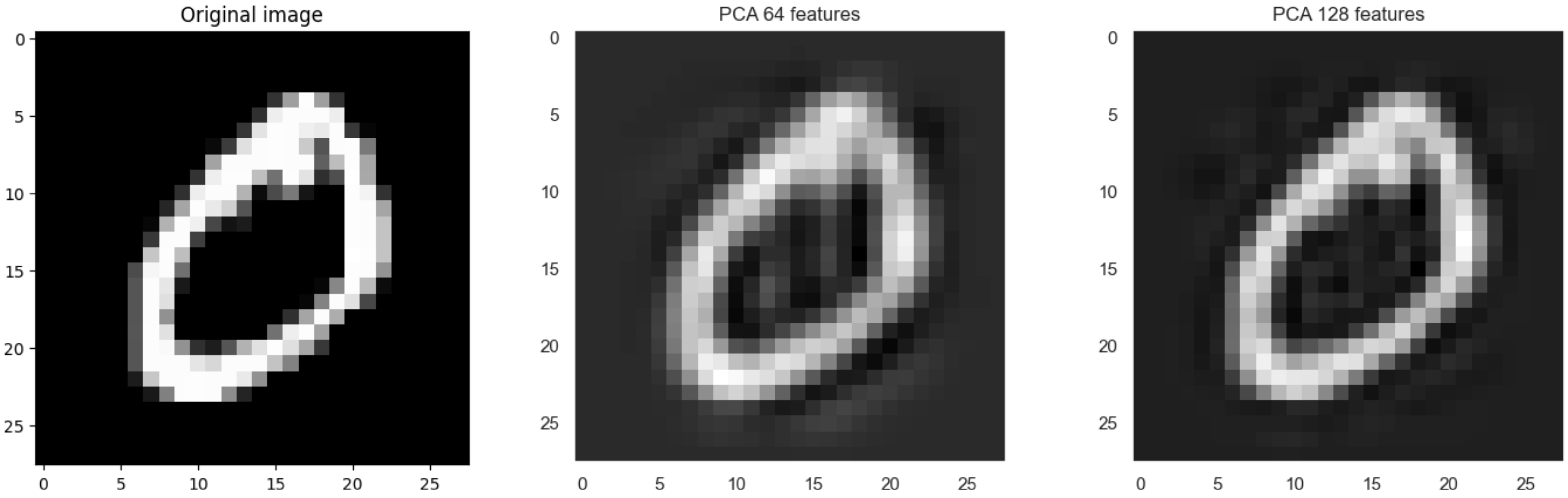
PCA explained variance ratio



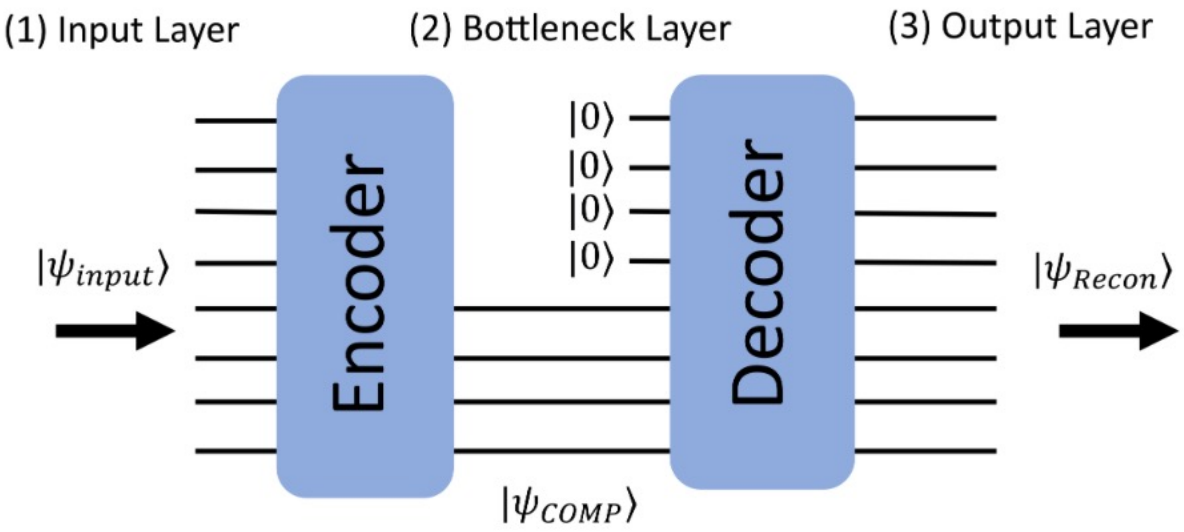
$$\text{exp. var. ratio} = \frac{\sum_1^N \lambda_i}{\sum_1^{784} \lambda_i}$$

Choose $N = 64$

PCA reconstruction



Autoencoder



$|\psi_{input}\rangle = n + k\ qubits$

$|\psi_{COMP}\rangle = n\ qubits$

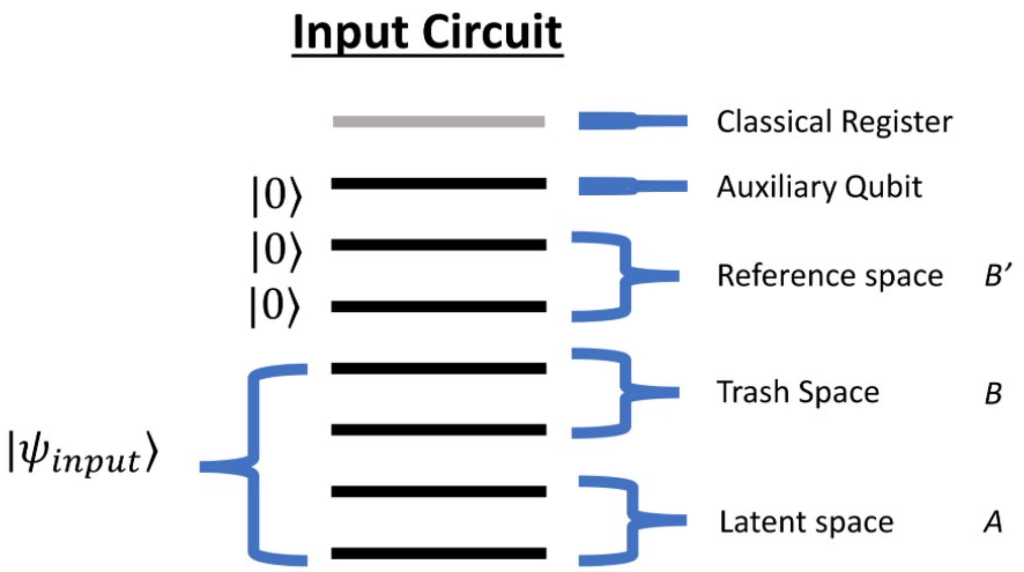
$|\psi_{Recon}\rangle = n + k\ qubits$

Encoder function $\boldsymbol{h} = f(\boldsymbol{x})$

Decoder function $\boldsymbol{r} = g(\boldsymbol{h})$

$g(f(\boldsymbol{x})) = \boldsymbol{x}$ not so useful $\Rightarrow$ learn useful properties of the dataset

Quantum Autoencoder in Qiskit



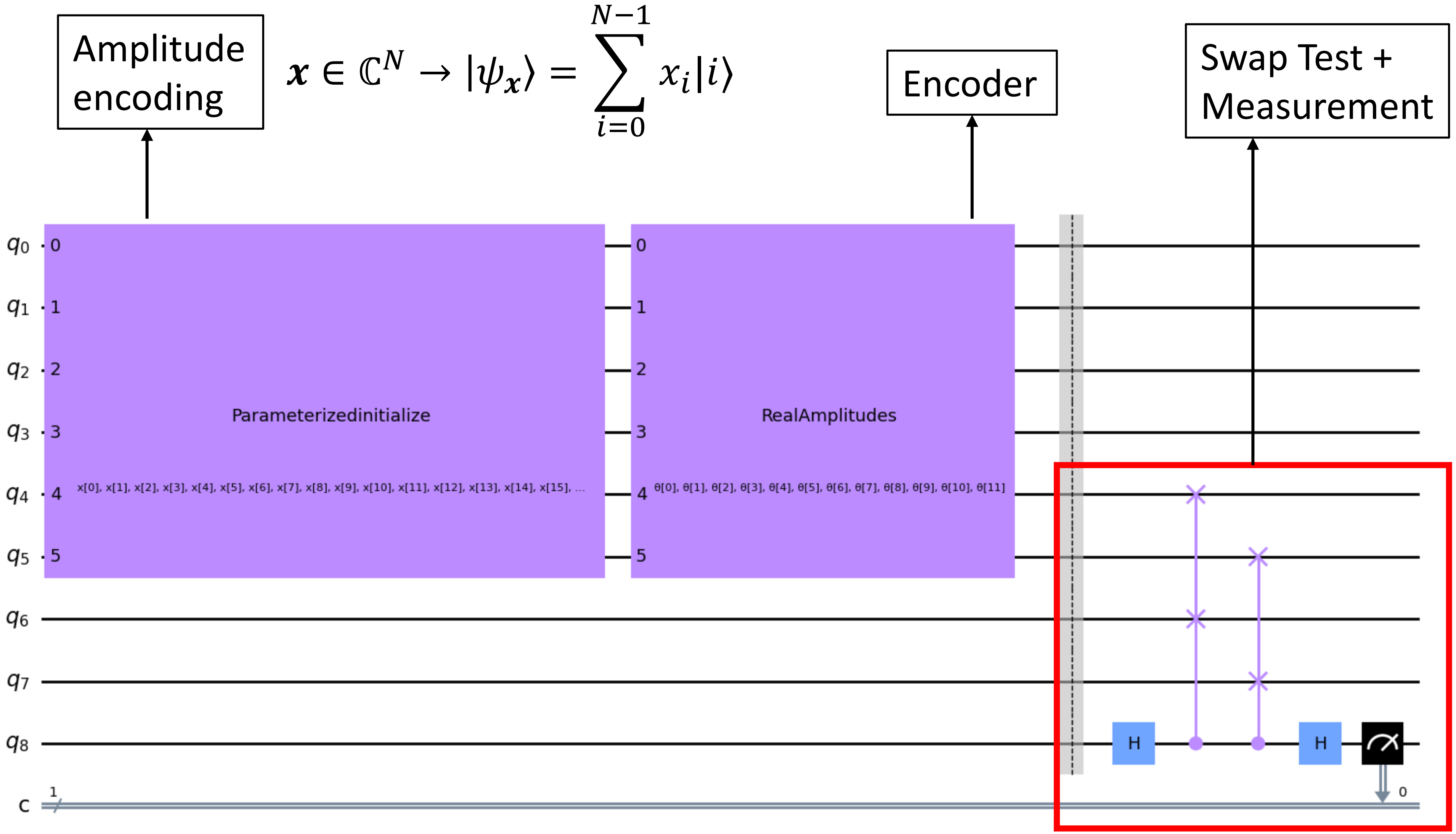
The "**algorithm**":

1. Amplitude encoding $\rightarrow |\psi\rangle_{AB} \otimes |a\rangle_{B'}$
2. Encoder $\rightarrow U(\theta)_{AB}\,(|\psi\rangle_{AB} \otimes |a\rangle_{B'})$
3. Swap test $\rightarrow V_{BB'}\, U(\theta)_{AB}\,(|\psi\rangle_{AB} \otimes |a\rangle_{B'})$
4. Cost function + optimization
5. Decoder $\rightarrow U^{\dagger}\,(\theta)_{AB'} V_{BB'} U(\theta)_{AB}\,(|\psi\rangle_{AB} \otimes |a\rangle_{B'}$

$$\rho_{out} = U^{\dagger}(\theta)_{AB'} Tr_B\big[V_{BB'} U(\theta)_{AB}[\psi_{AB} \otimes a_{B'}] U^{\dagger}(\theta)_{AB} V^{\dagger}_{BB'}\big] U(\theta)_{AB'}$$

Goal $\longrightarrow$ $\boxed{\max F(|\psi\rangle_{AB}, \rho_{out})}$ $\Longleftrightarrow$ $\boxed{\max F\big(Tr_A[U(\theta)_{AB}\psi_{AB}U^{\dagger}(\theta)_{AB}], a_{B'}\big)}$

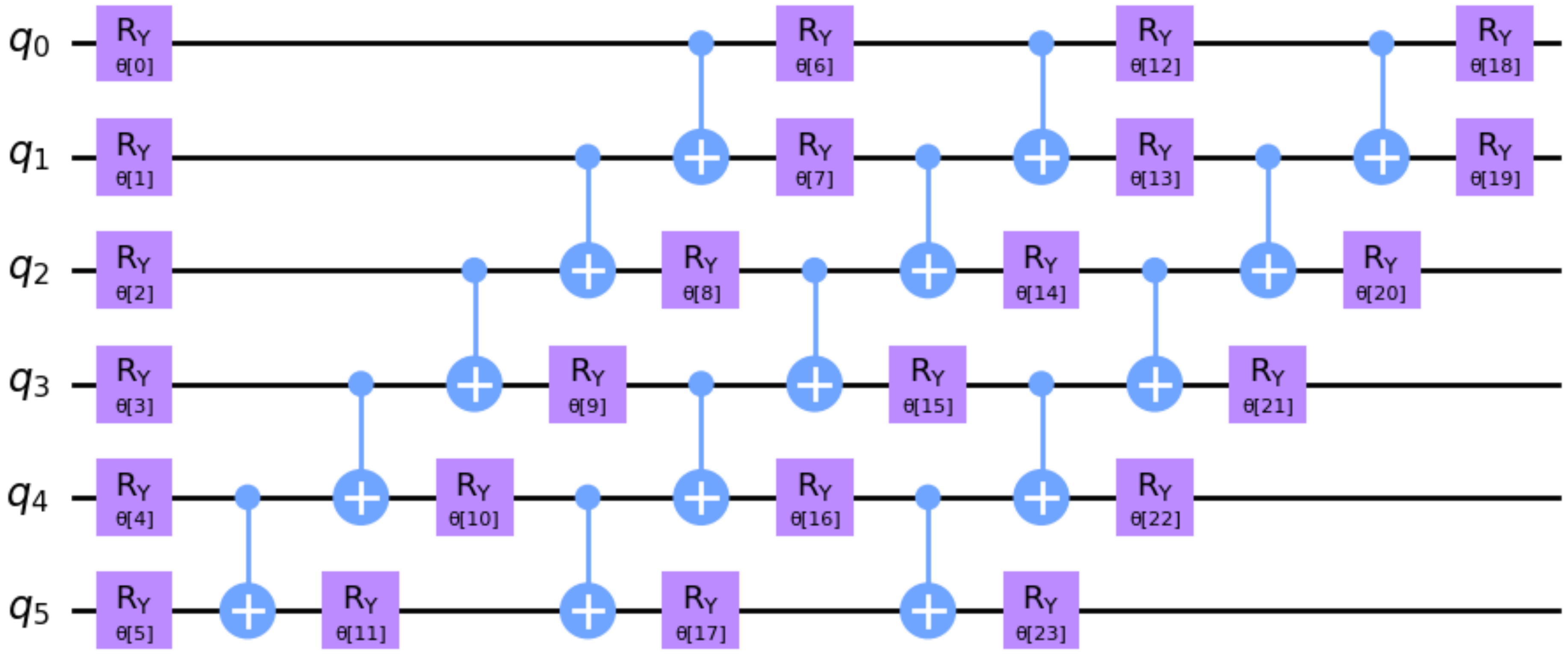
The circuit



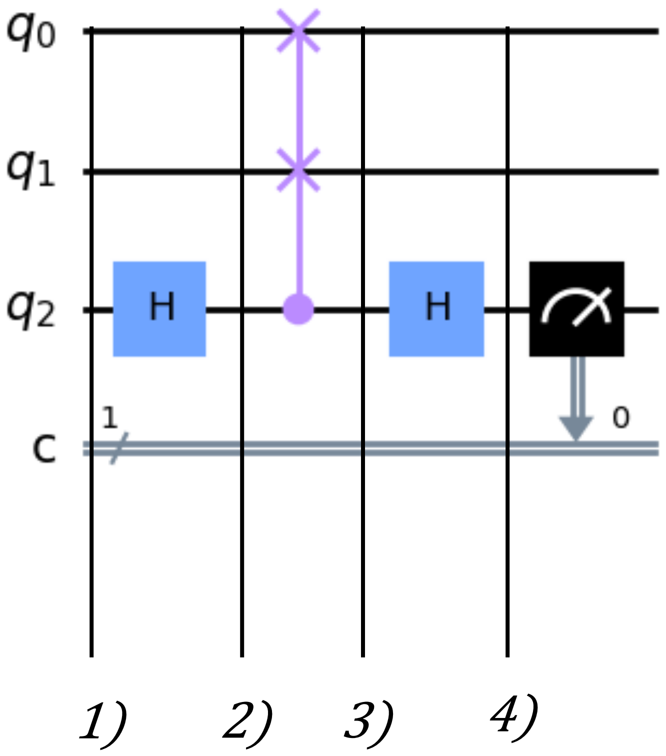
Amplitude encoding
$\boldsymbol{x} \in \mathbb{C}^N \to |\psi_{\boldsymbol{x}}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$
Encoder
Swap Test + Measurement
q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
c
Parameterizedinitialize
x[0], x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10], x[11], x[12], x[13], x[14], x[15], ...
RealAmplitudes
θ[0], θ[1], θ[2], θ[3], θ[4], θ[5], θ[6], θ[7], θ[8], θ[9], θ[10], θ[11]
H
H

Encoder

- *RealAmplitudes* circuit → Layers of **Y** rotations + **C-X** entanglement
- Parameters → Trainable weights for QNN



Swap Test



1) $|\psi\rangle_1 = |a\rangle|b\rangle \otimes |0\rangle$

2) $|\psi\rangle_2 = |a\rangle|b\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

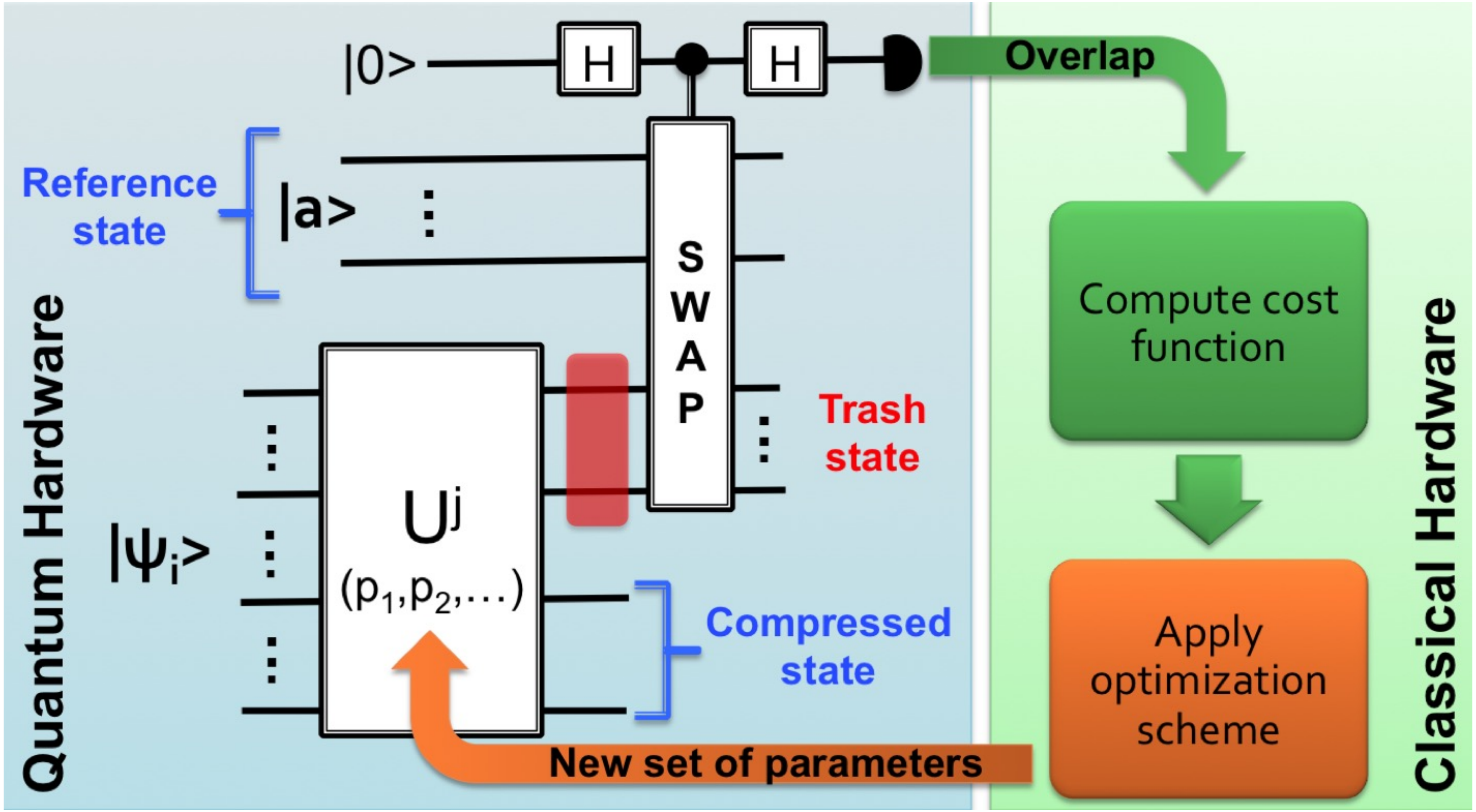
3) $|\psi\rangle_3 = \frac{1}{\sqrt{2}}(|a\rangle|b\rangle \otimes |0\rangle + |b\rangle|a\rangle \otimes |1\rangle)$

4) $|\psi\rangle_4 = \frac{1}{2}(|a\rangle|b\rangle + |b\rangle|a\rangle) \otimes |0\rangle + \frac{1}{2}(|a\rangle|b\rangle - |b\rangle|a\rangle) \otimes |1\rangle$

Measurement: $p(0) = |(\mathbb{I} \otimes \langle 0|)|\psi\rangle_4|^2 = \frac{1}{2} + \frac{1}{2}|\langle a|b\rangle|^2$

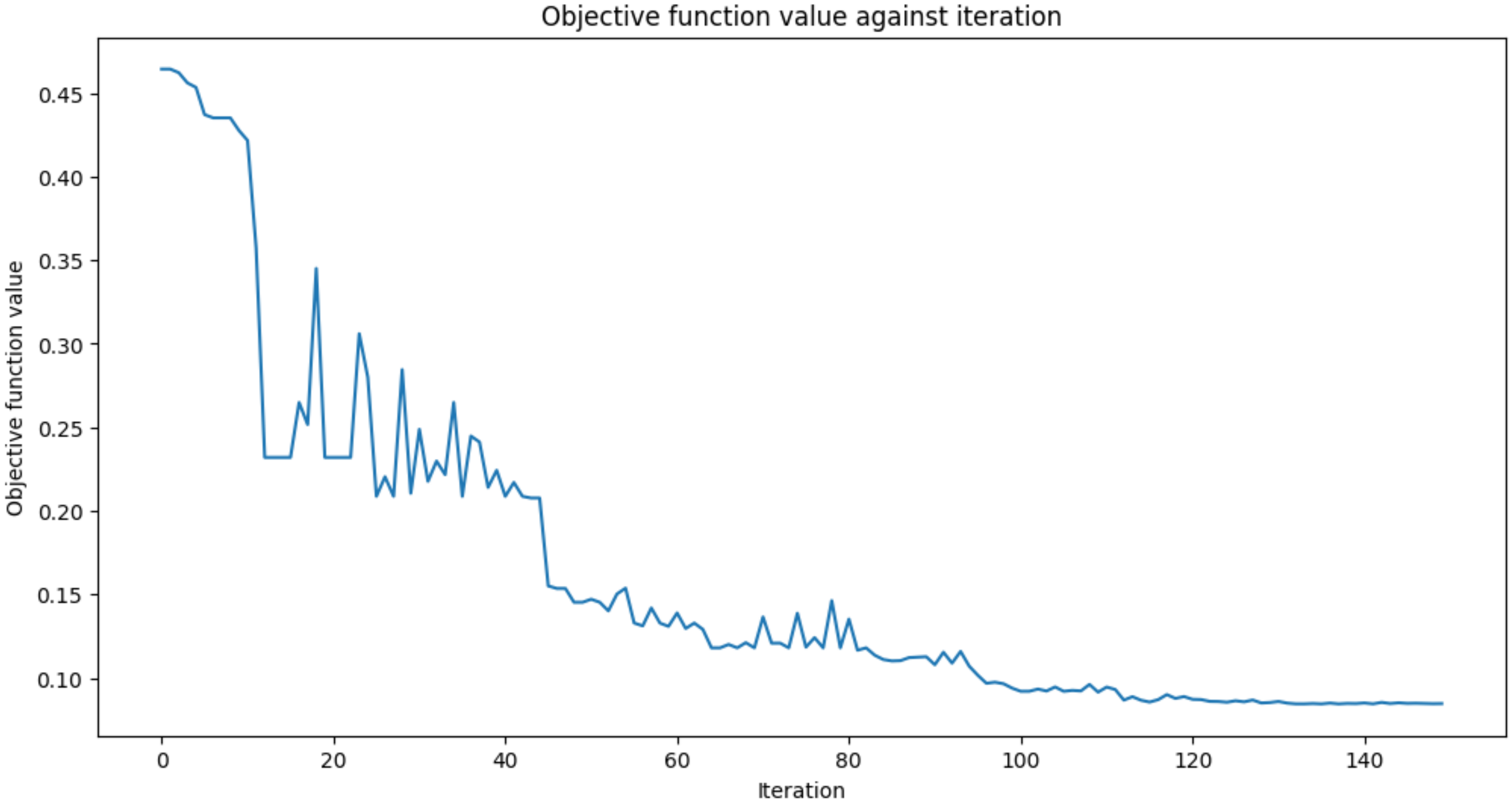
$\boxed{|\langle a|b\rangle|^2 = 2p(0) - 1 = 1 - 2p(1)}$ $\Rightarrow$ Cost function

Optimization Routine



Optimization Routine

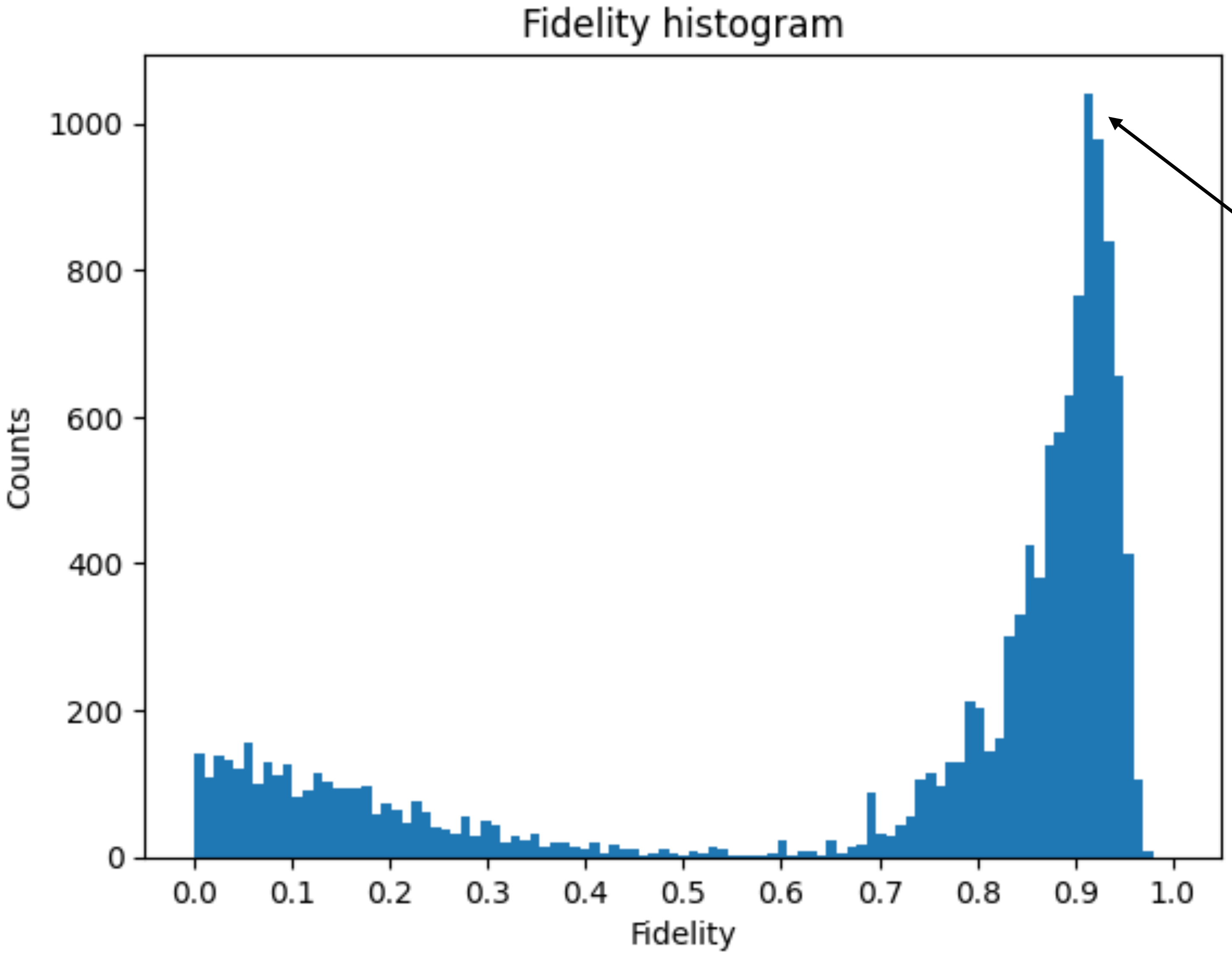
Time consuming $\Longrightarrow$ only $1/10$ of $0'$s and $1'$s $\begin{cases} \simeq 1200 \text{ train} \\ \simeq 250 \text{ test} \end{cases}$



Result:
Optimal parameters for $U(\theta)$

- RealAmplitudes, 3 repetitions
- Time $\simeq$ 40 min.

Fidelity

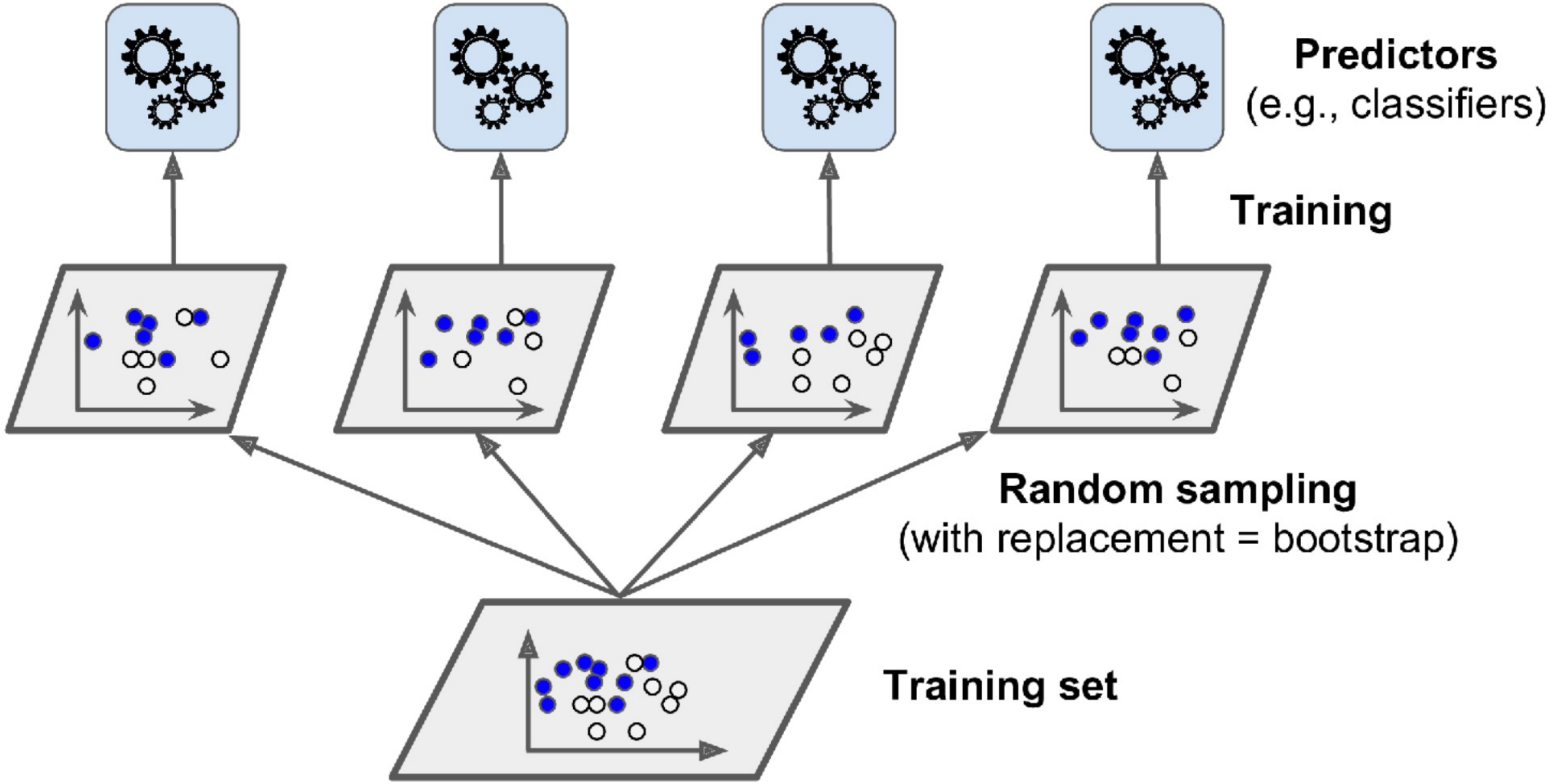


$$F = |\langle \psi_{decoder} | \psi_{initialized} \rangle|^2$$

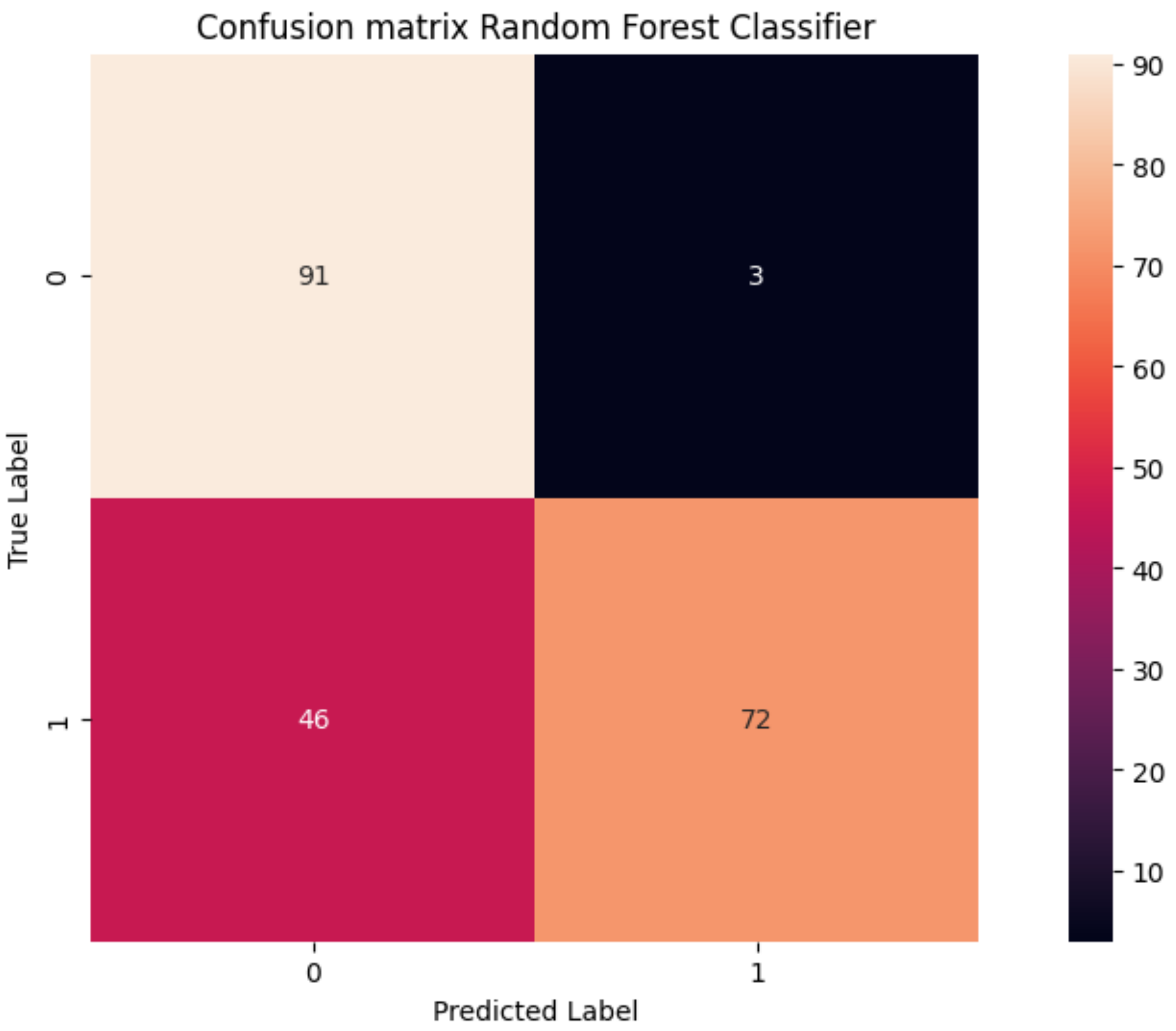
Peak at $F \simeq 0.9$

Random Forest Classifier

- *Ensemble learner* built on **decision trees**
- Decision tree ⟶ Binary splitting



Results 784 → 16 features



Class	Precision	Recall	f1-score	Support
0	0.66	0.97	0.79	94
1	0.96	0.61	0.75	118

Tp = True positive

Fp = False positive

Fn = False negative

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \qquad \text{Recall} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}}$$

Results $784 \rightarrow 16$ features

RA, 1 rep

Class	Precision	Recall	f1-score	Support
0	0.69	0.93	0.79	94
1	0.92	0.66	0.77	118

RA, 2 rep

Class	Precision	Recall	f1-score	Support
0	0.73	0.94	0.82	94
1	0.93	0.72	0.81	118

PCA

Class	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	94
1	1.00	1.00	1.00	118

References

1. Maria Schuld, Francesco Petruccione, Machine Learning with Quantum Computers, Springer Cham, 18 October 2021.
2. Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.
3. Romero, Jonathan, Jonathan P. Olson, and Alan Aspuru-Guzik. “Quantum autoencoders for efficient compression of quantum data.” Quantum Science and Technology 2.4 (2017): 045001.
4. Qiskit Textbook Machine Learning, https://qiskit-community.github.io/qiskit-machine-learning/tutorials/12_quantum_autoencoder.html