CODE RED’24

QUANTUM THREADS

Scalable Hybrid Quantum-Classical Neural Networks With Distributed Processing

TEAM A

PROBLEM STATEMENT CODE : CR11

Team lead name : Nagaraj Poojari

INSTITUTION NAME

M S Ramaiah Institute Of Technology

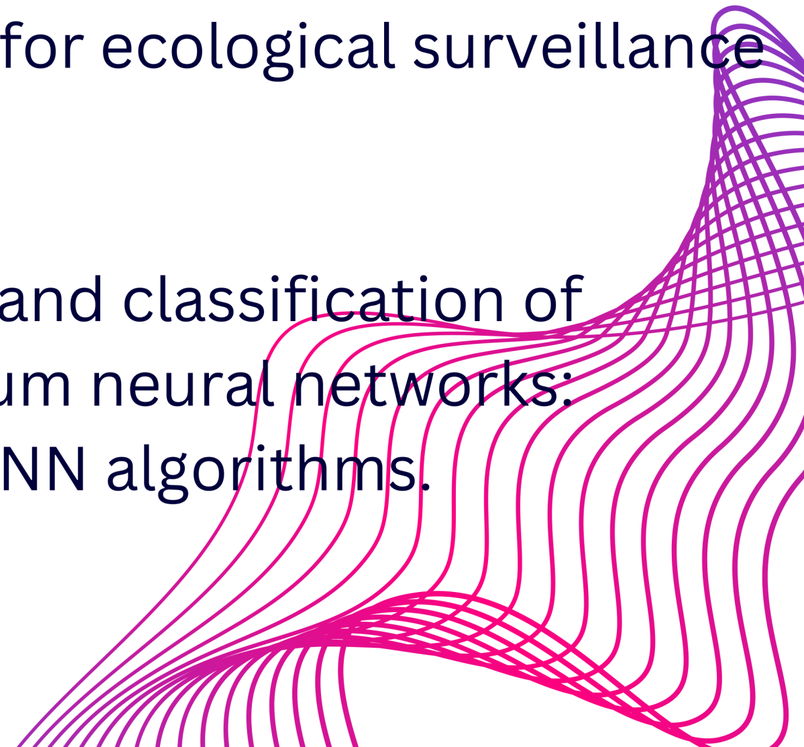
Bengaluru

Problem statement

Underwater Animal Identification and Classification Using a Hybrid Classical–Quantum Algorithm

- Underwater Animal Identification and Classification is gaining significant importance in recent times due to the growing demand for ecological surveillance and biodiversity monitoring.

- Create a web application to demonstrate identification and classification of underwater animals using three hybrid Classical-Quantum neural networks: ResNet50-QCNN, ResNet18-QCNN and InceptionV3-QCNN algorithms.



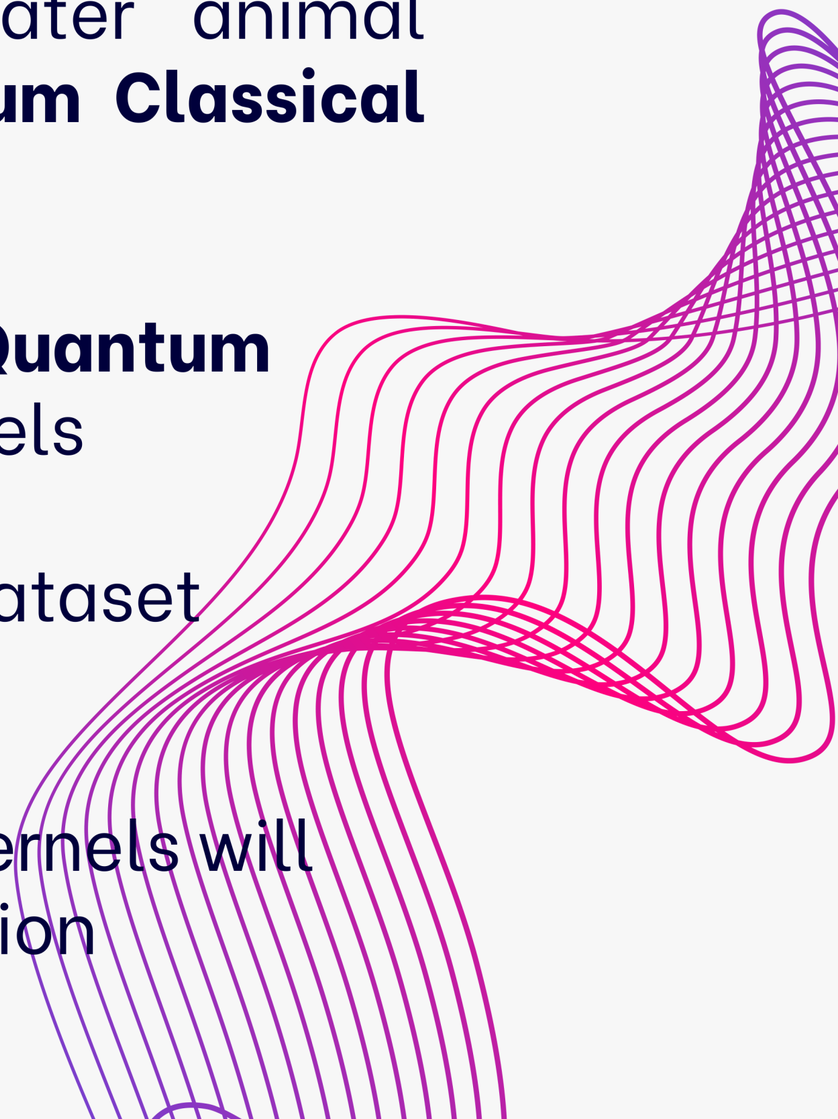
Our Solution

- An end to end web app for under water animal classification backboned by **Hybrid Quantum Classical ML model**

- Hybrid Quantum ML model will use PQC as **Quantum kernels** along with pretrained classical models

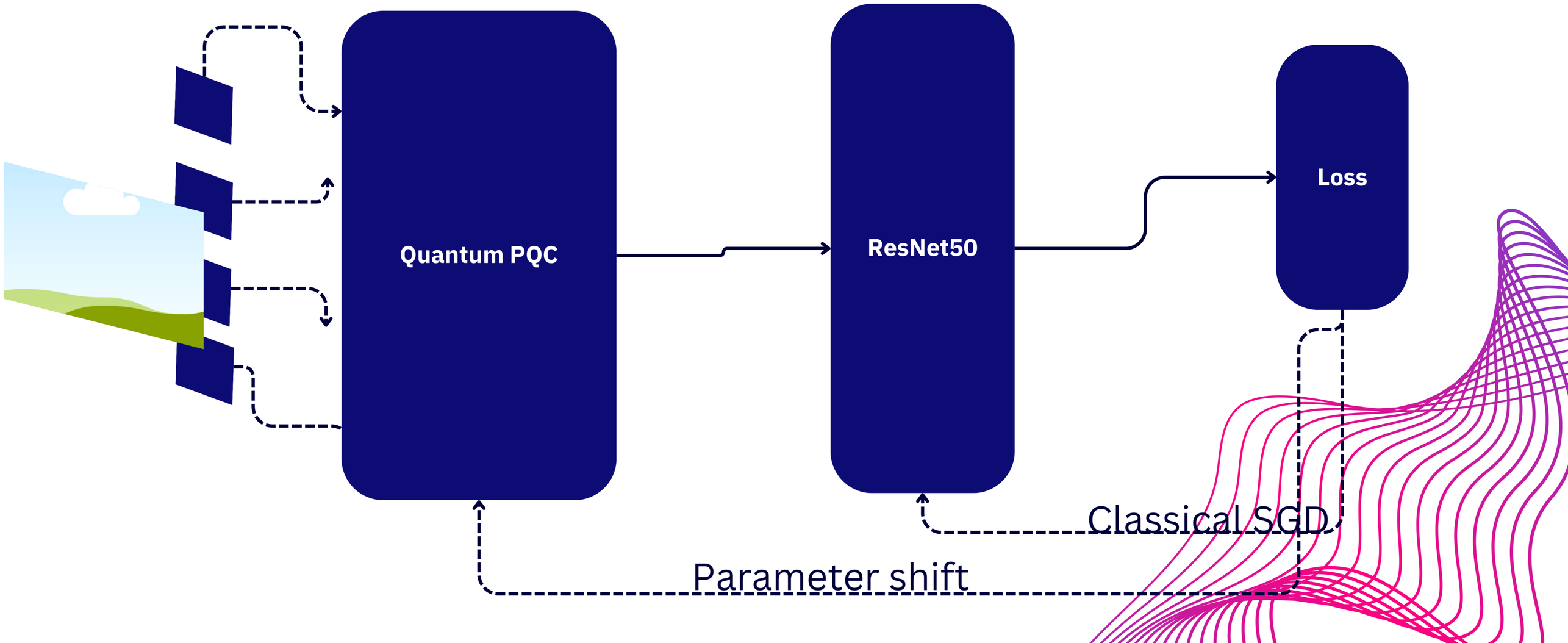
- Whole model will be trained on UAD-2023 dataset containing >13k images for 23 classes

- Entire model is made learnable , quantum kernels will use **classical gradient descent** for optimization

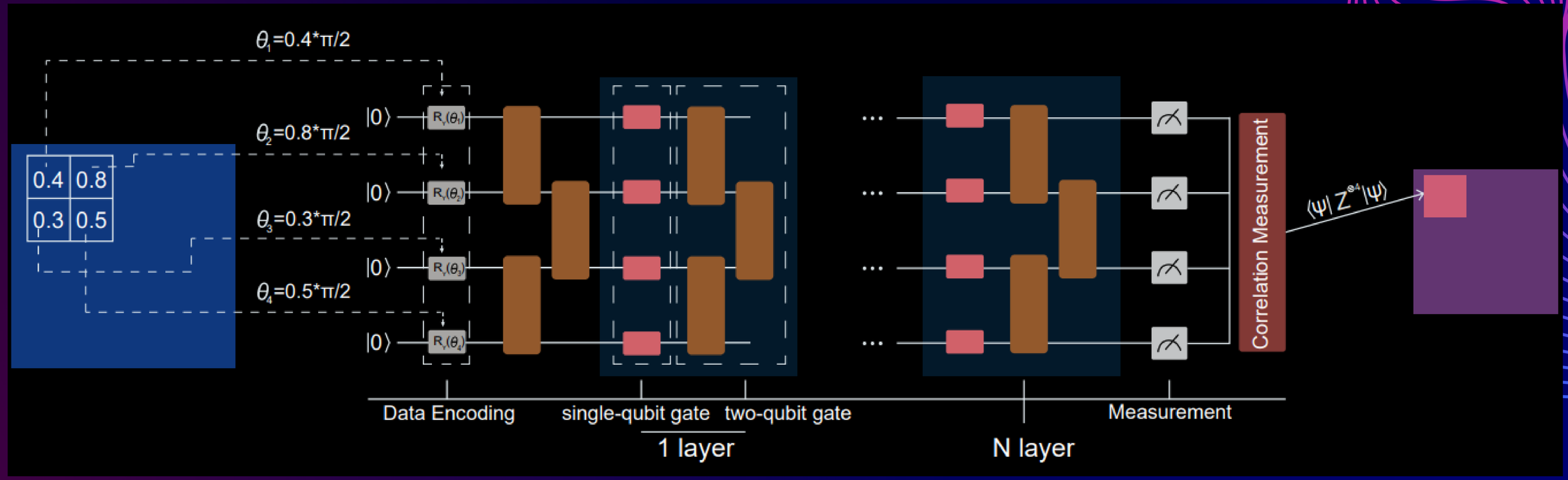


CODE RED’24

Model architecture



Quantum Kernel

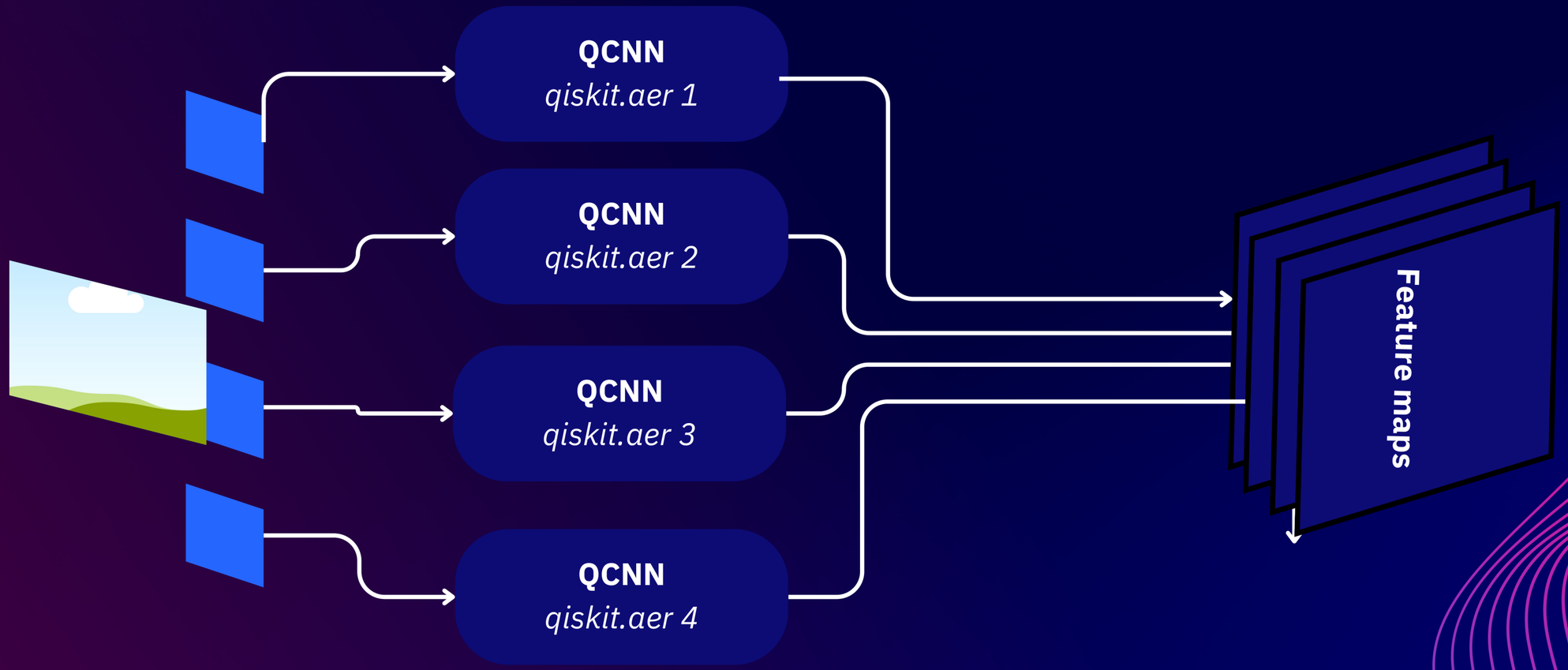


What’s wrong ?

- Practically it needs **lots of qubits** and obviously not suitable for NISQ
- Again can’t even process all kernels parallelly because of same issue
- **Quantum gradient descent is expensive** , needs exponential time complexity
- **Totally not so scalable**

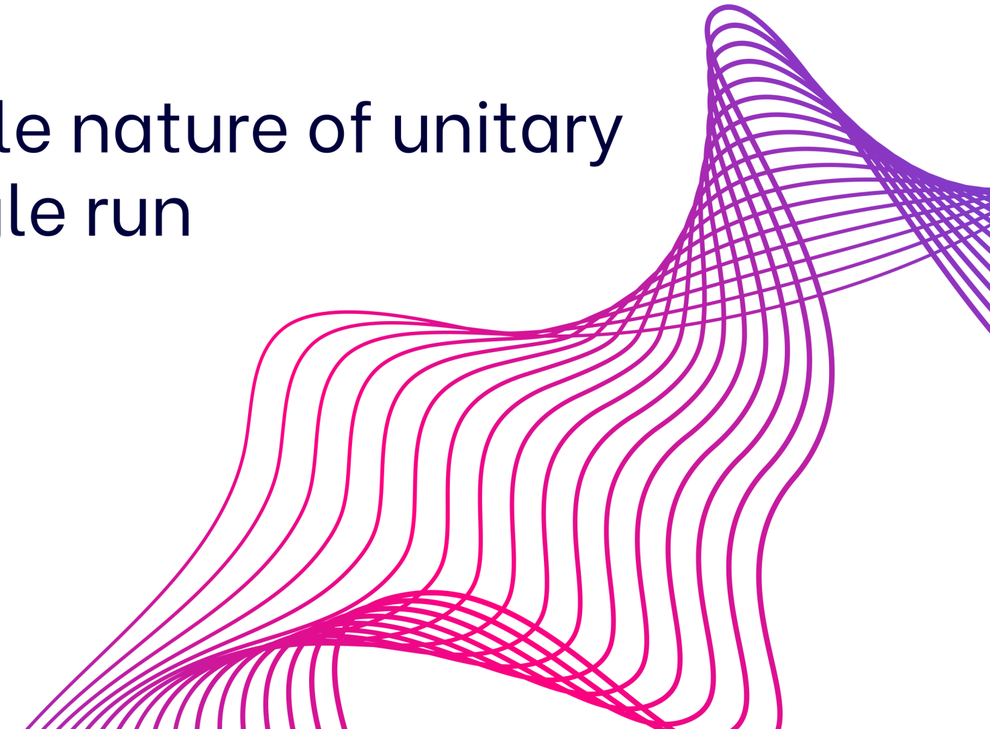
Then , How to tackle these problems ?

#1 Process Parallelly

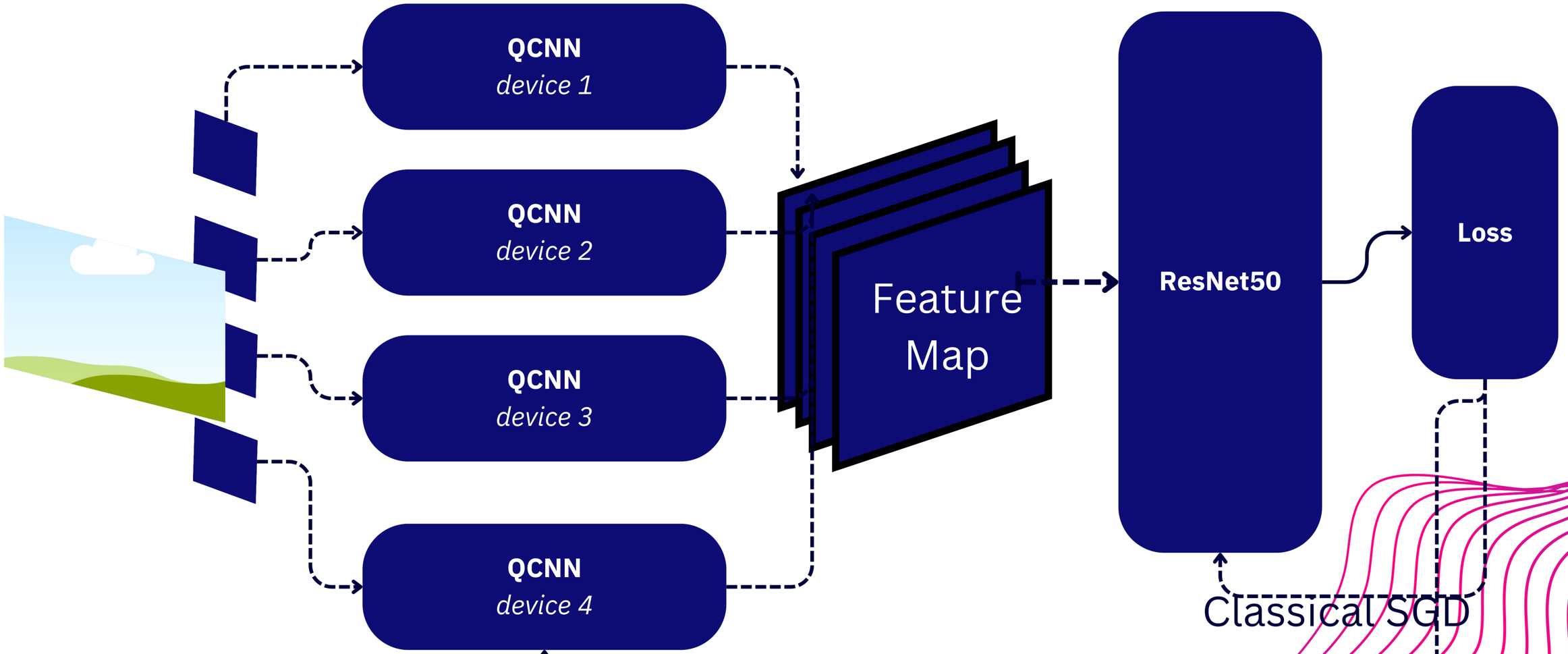


#2 Adjoint differentiation

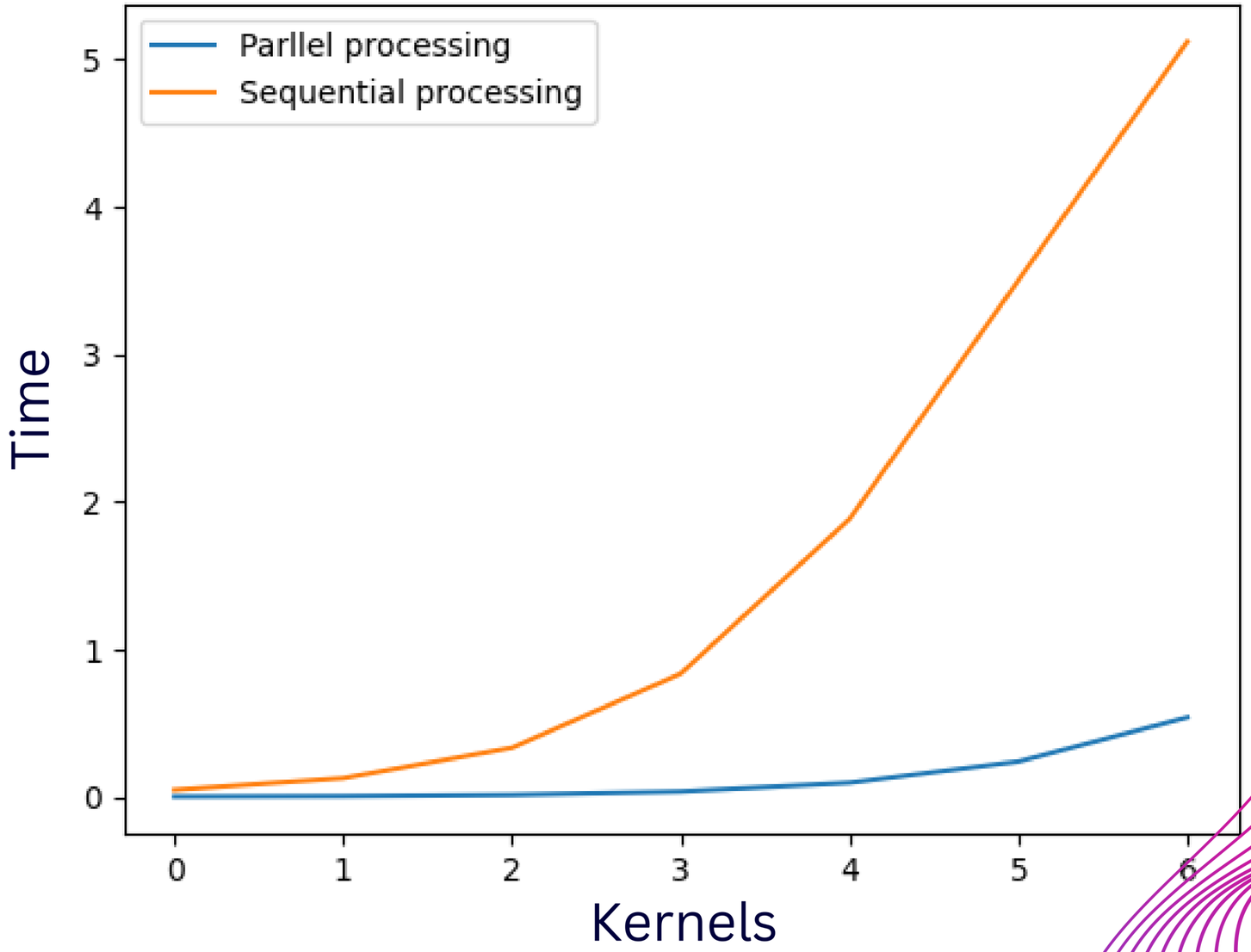
- Existing technique is to use Parameter-shift for gradient calculation which need ***2xp*** circuit runs
- **Adjoint differentiation can do that in 1 run**
- Adjoint differentiation uses reversible nature of unitary circuits to calculate gradient in single run
- And it requires **less Quantum RAM**



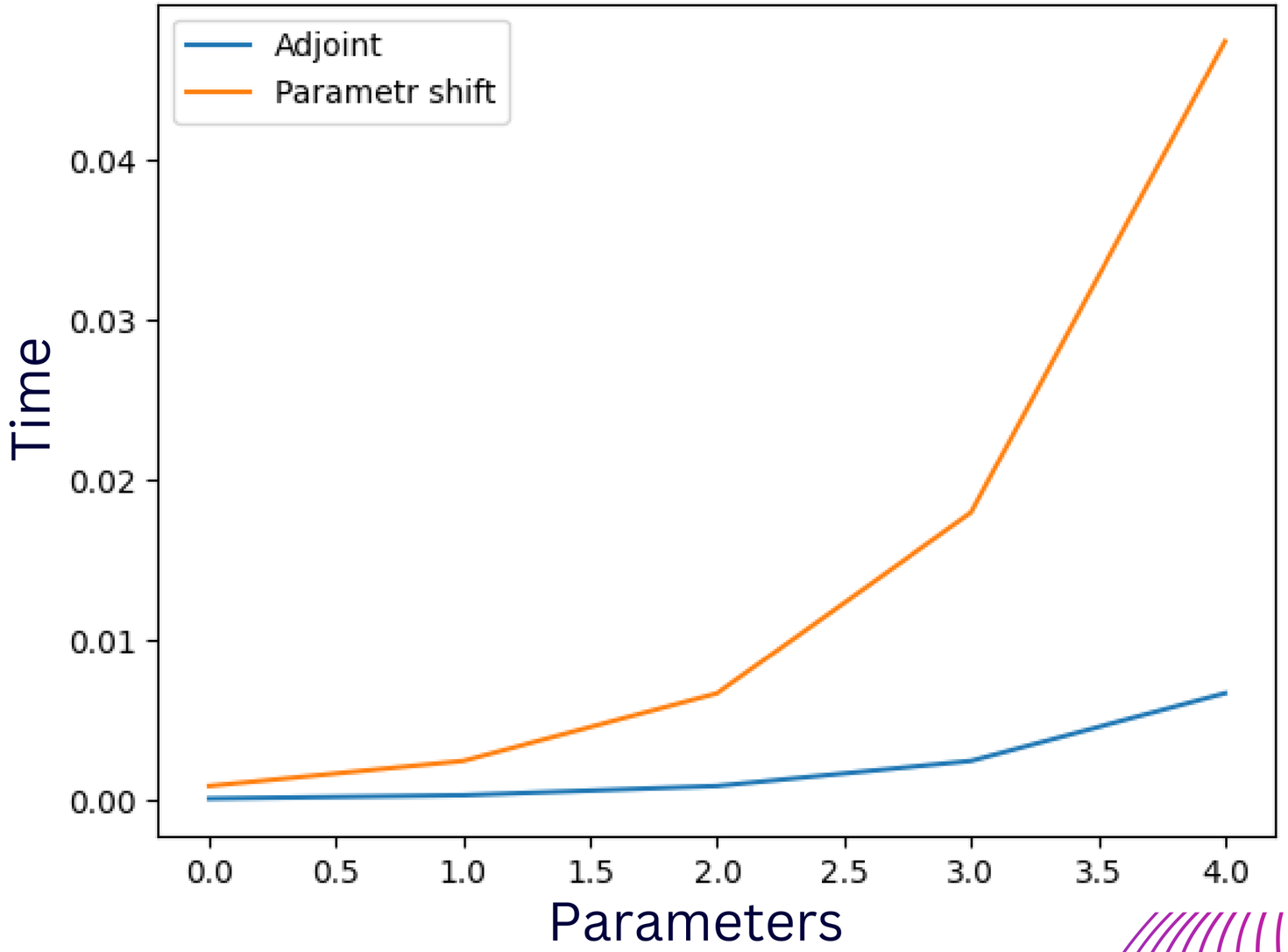
Final model


QCNN
device 1
QCNN
device 2
QCNN
device 3
QCNN
device 4
Feature
Map
ResNet50
Loss
Classical SGD
Adjoint diifferentiation

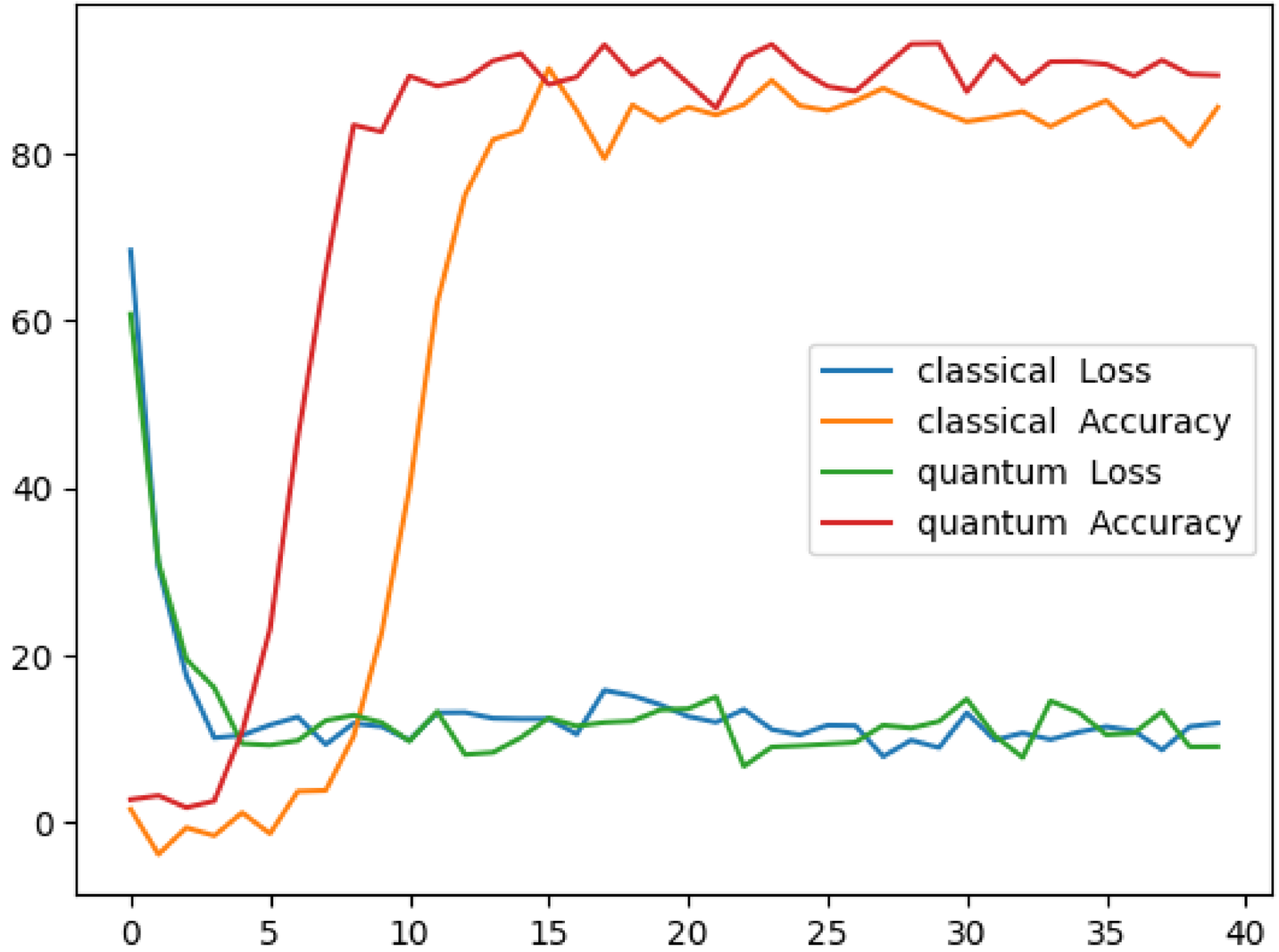
#1 Parallel vs Sequential



#2 Adjoint vs Parameter shift



#3 Classical ResNet50 vs Quantum ResNet50 - Training



#4 Classical ResNet50 vs Quantum ResNet50 - Validation