High Energy Physics and Quantum Computing

2024 IonQ 2nd week meeting

2024. 3. 5. 10:00 PM(GMT +9)

Weekly goal

[3. 5.] Review the reference [1] thoroughly and understand basic idea.

[3.12.] Implement the idea using Qiskit and/or PennyLane, and reproduce their results for two examples in the paper.

[3.19.] Then consider a more complex and more realistic example, taking electron-positron production at the Large Hadron Collider (pp → e+e–). Effectively, this problem involves four-dimensional integration.

[3.26.] Try to optimize the circuits with different choice of the cost function or variations of quantum circuits.

[4. 2.] Study Monte-Carlo sampling with above integration method, and how to implement the importance sampling into a variational quantum circuit. (Check Ref. [2])

Basic Idea of Ref. 1

- Quantum Machine Learnig(QML) : Variational Quantum Circuit(VQC)

- **Data Re-uploading<Model>**
  How to input the data to the quantum circuit

Data re-uploading for a universal quantum classifier

Adrián Pérez-Salinas[1,2], Alba Cervera-Lierta[1,2], Elies Gil-Fuster[3], and José I. Latorre[1,2,4,5]

[1]Barcelona Supercomputing Center
[2]Institut de Ciències del Cosmos, Universitat de Barcelona, Barcelona, Spain
[3]Dept. Física Quàntica i Astrofísica, Universitat de Barcelona, Barcelona, Spain.
[4]Nikhef Theory Group, Science Park 105, 1098 XG Amsterdam, The Netherlands.
[5]Center for Quantum Technologies, National University of Singapore, Singapore.

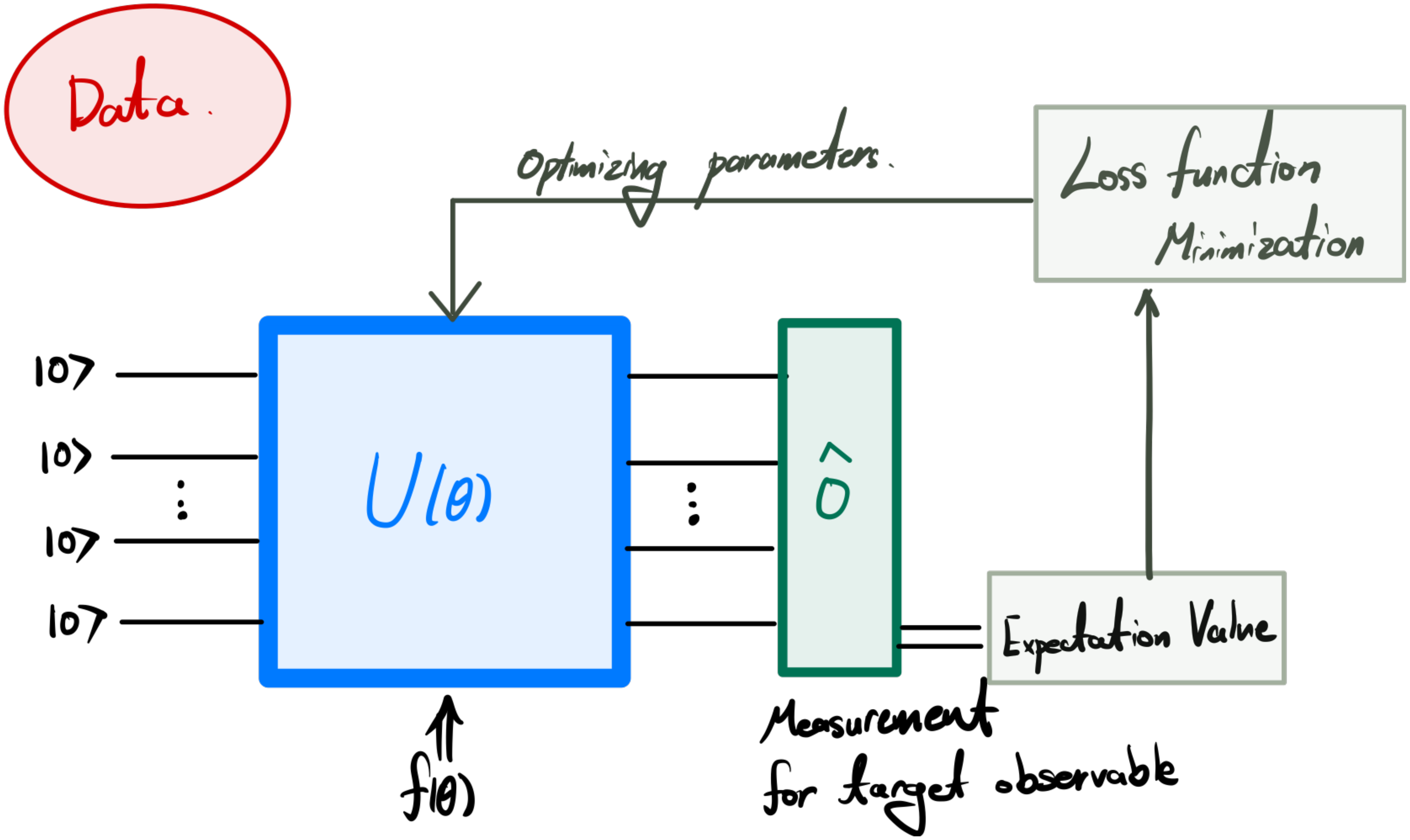
- **Parameter shift rule(PSR)<Optimizing Step>**

General parameter-shift rules for quantum gradients

David Wierichs[1,2], Josh Izaac[1], Cody Wang[3], and Cedric Yen-Yu Lin[3]

[1]Xanadu, Toronto, ON, M5G 2C8, Canada
[2]Institute for Theoretical Physics, University of Cologne, Germany
[3]AWS Quantum Technologies, Seattle, Washington 98170, USA

QML - VQC



Data.
Optimizing parameters.
Loss function
Minimization
|0⟩
|0⟩
|0⟩
|0⟩
U(θ)
Ô
Expectation Value
f(θ)
Measurement
for target observable

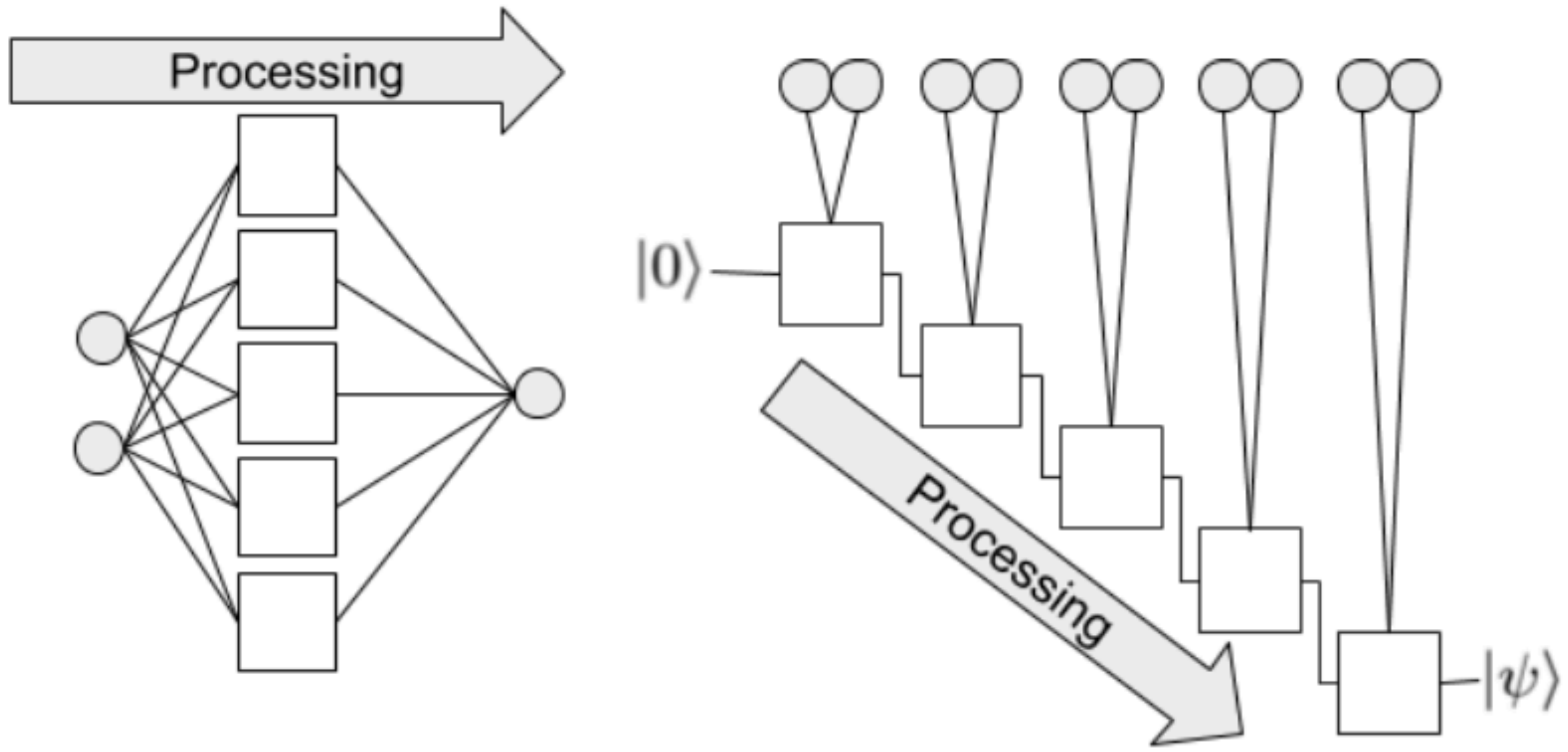
Data Re-uploading

How we input the data for the quantum circuit? ⇒ Data Re-uploading

Data re-uploading for a universal quantum classifier

Adrián Pérez-Salinas[1,2], Alba Cervera-Lierta[1,2], Elies Gil-Fuster[3], and José I. Latorre[1,2,4,5]

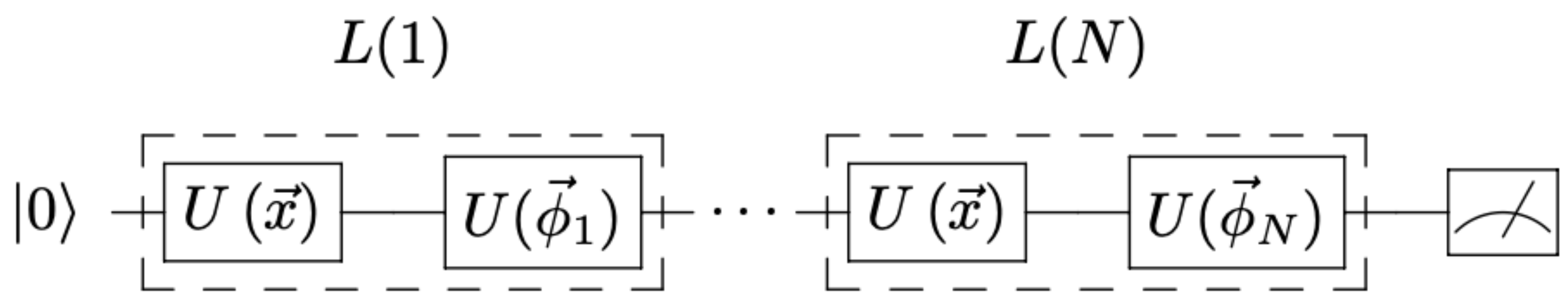
[1]Barcelona Supercomputing Center
[2]Institut de Ciències del Cosmos, Universitat de Barcelona, Barcelona, Spain
[3]Dept. Física Quàntica i Astrofísica, Universitat de Barcelona, Barcelona, Spain.
[4]Nikhef Theory Group, Science Park 105, 1098 XG Amsterdam, The Netherlands.
[5]Center for Quantum Technologies, National University of Singapore, Singapore.



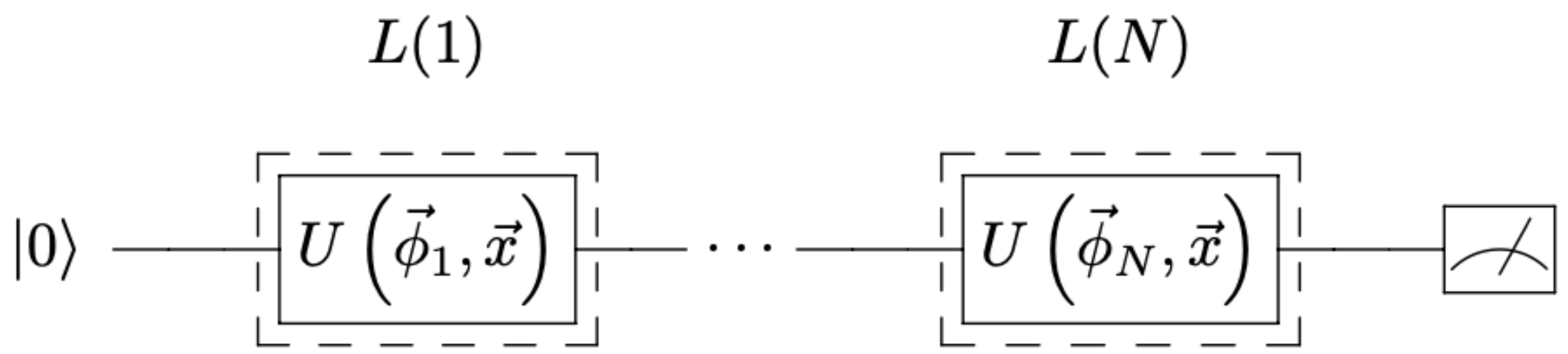
(a) Neural network (b) Quantum classifier

Data Re-uploading

2 Ways for re-uploading



(a) Original scheme

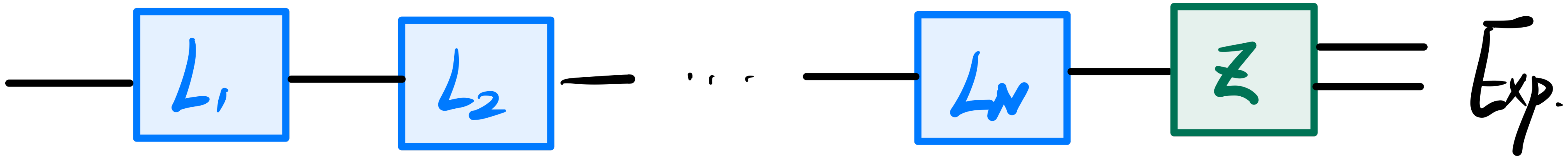


(b) Compressed scheme

$\vec{x}$: Data

$\overrightarrow{\phi}$: parameters

$\text{parameter}_i = \vec{\theta}_i + \overrightarrow{\omega}_i \circ \vec{x}$ for layer L_i



Using the fidelity cost function

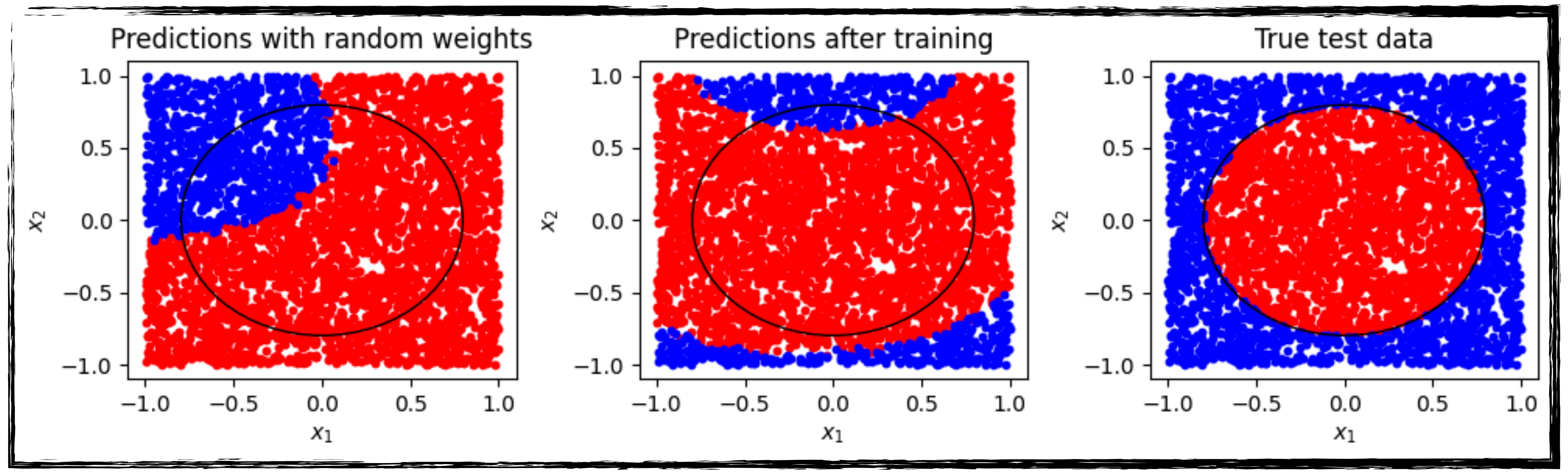
Data Re-uploading

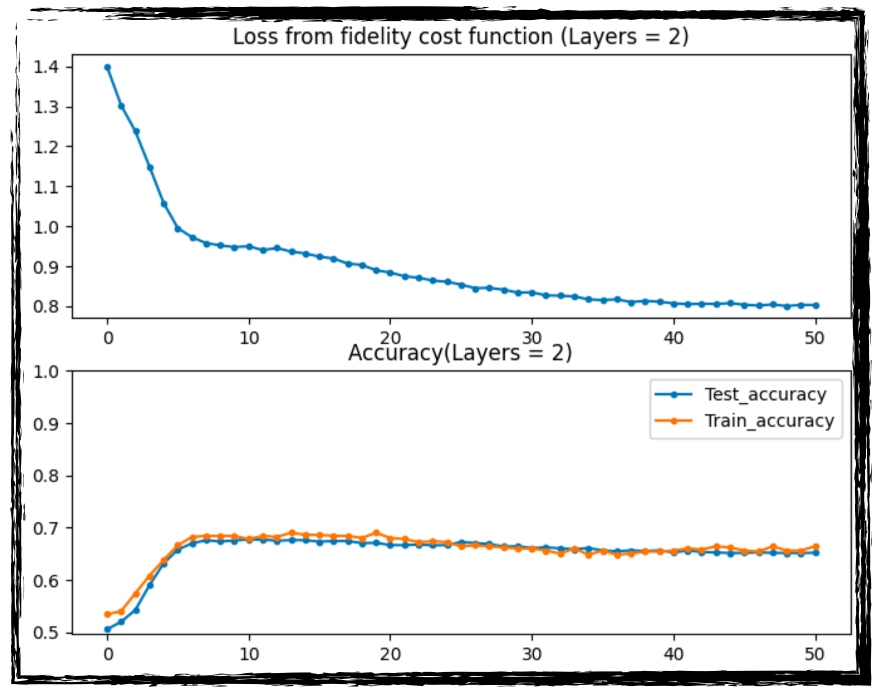
Reproducing Results with Pennylane(Single qubit Classifier)

Circle(Origin Scheme)

Layer 2

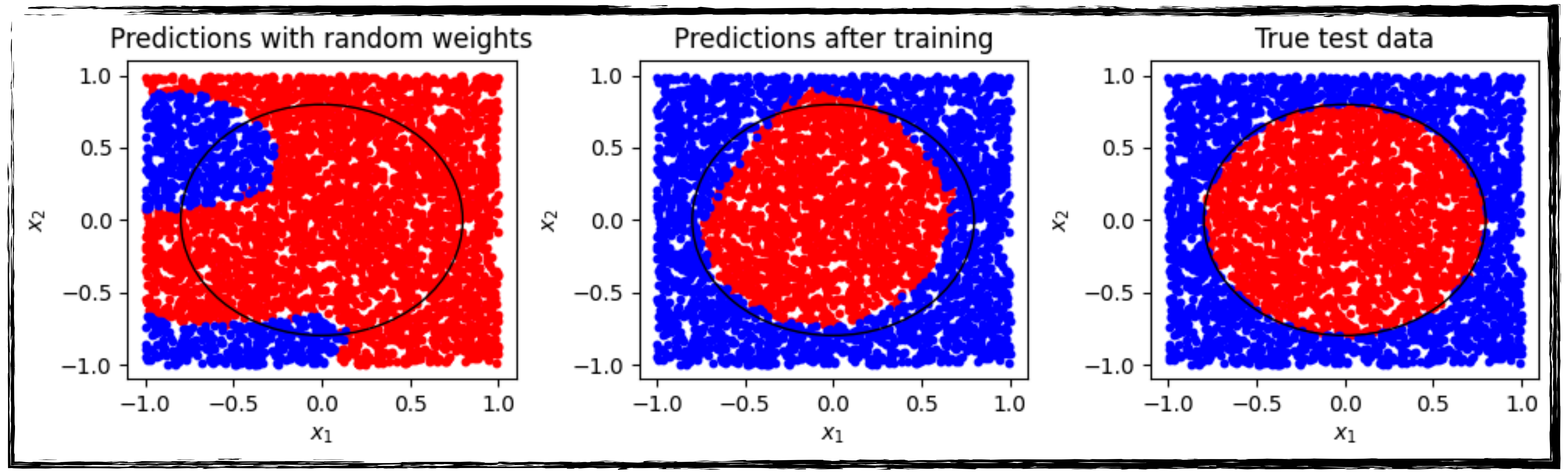
0.78

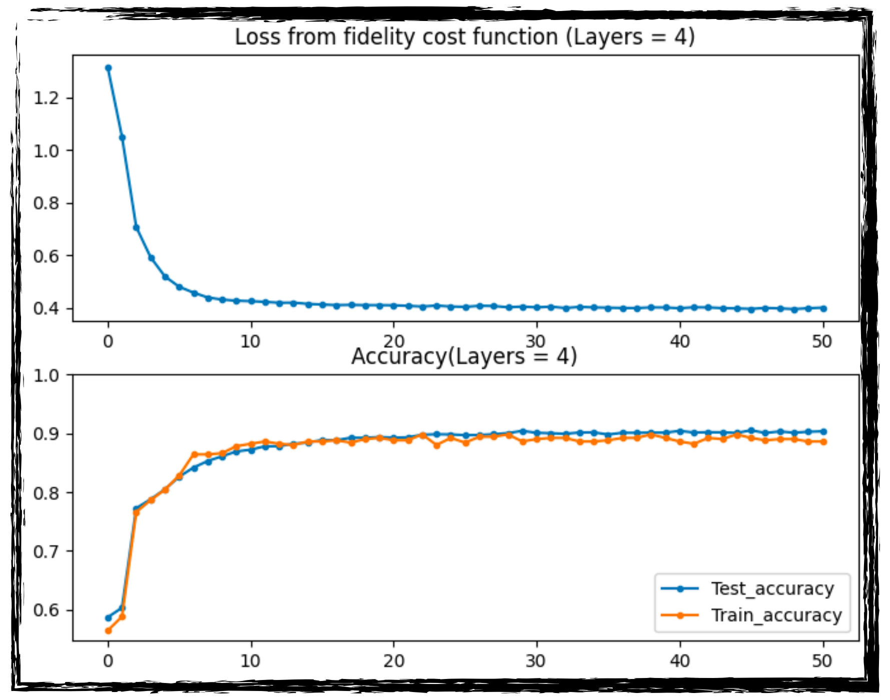




Layer 4

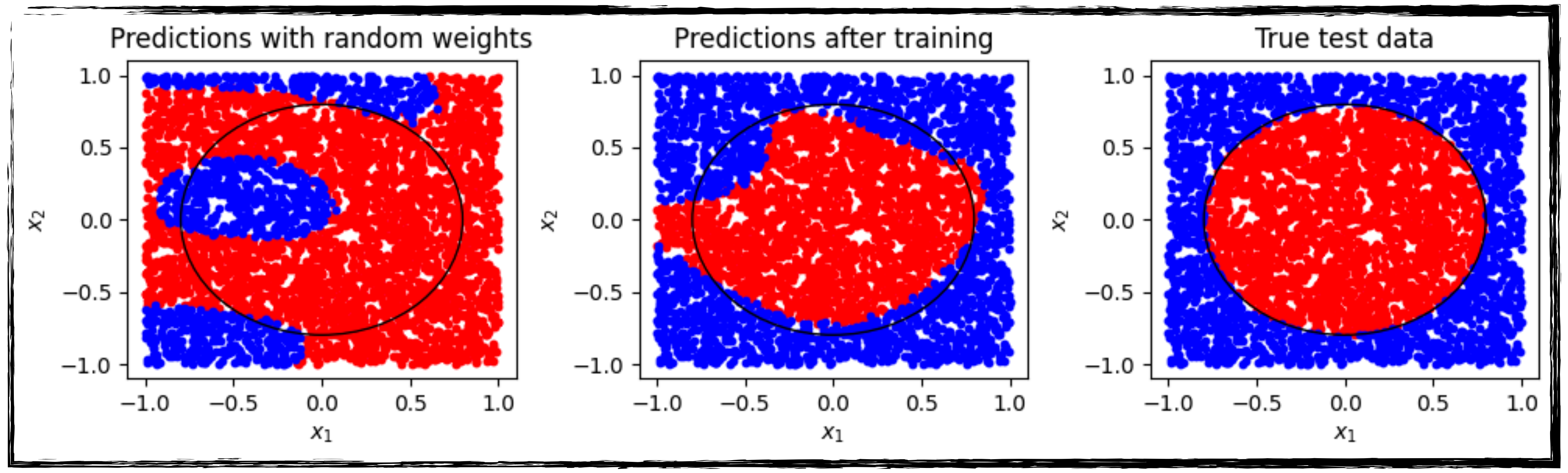
0.89

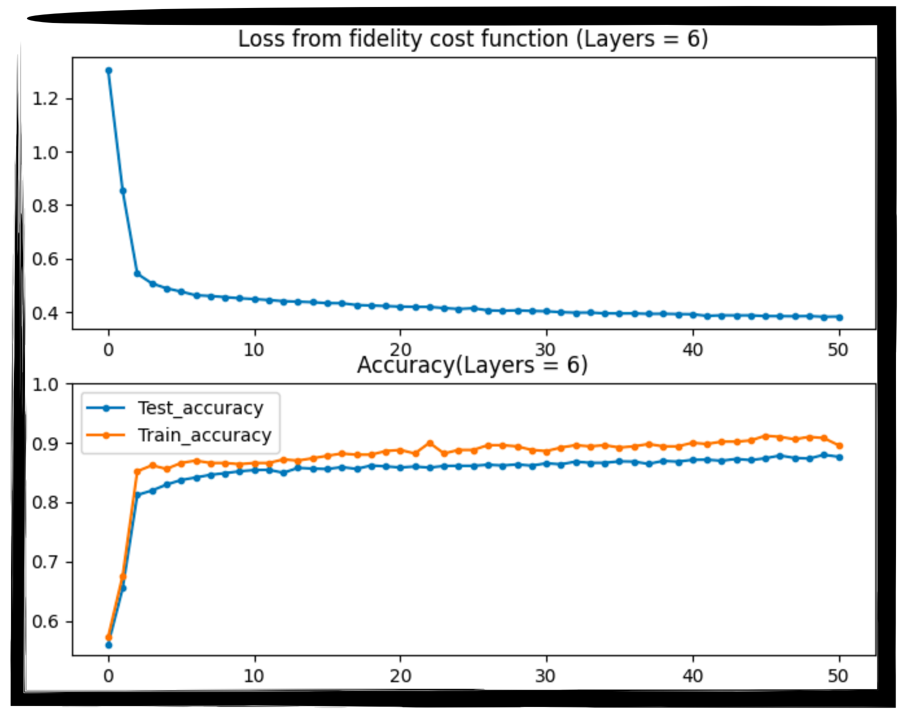




Layer 6

0.87





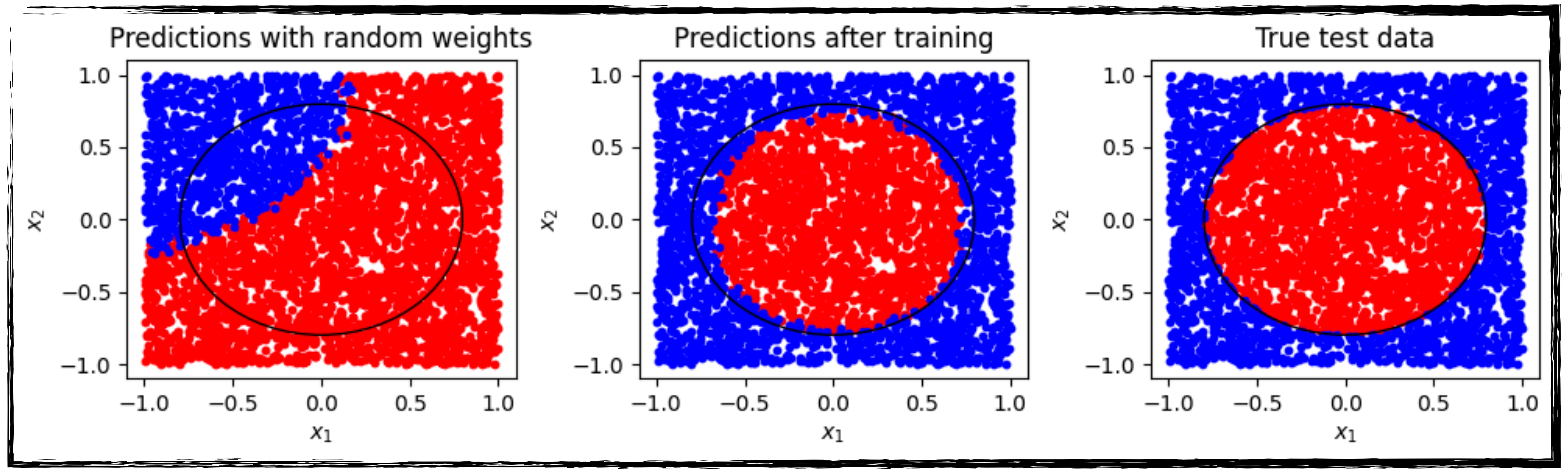
Data Re-uploading

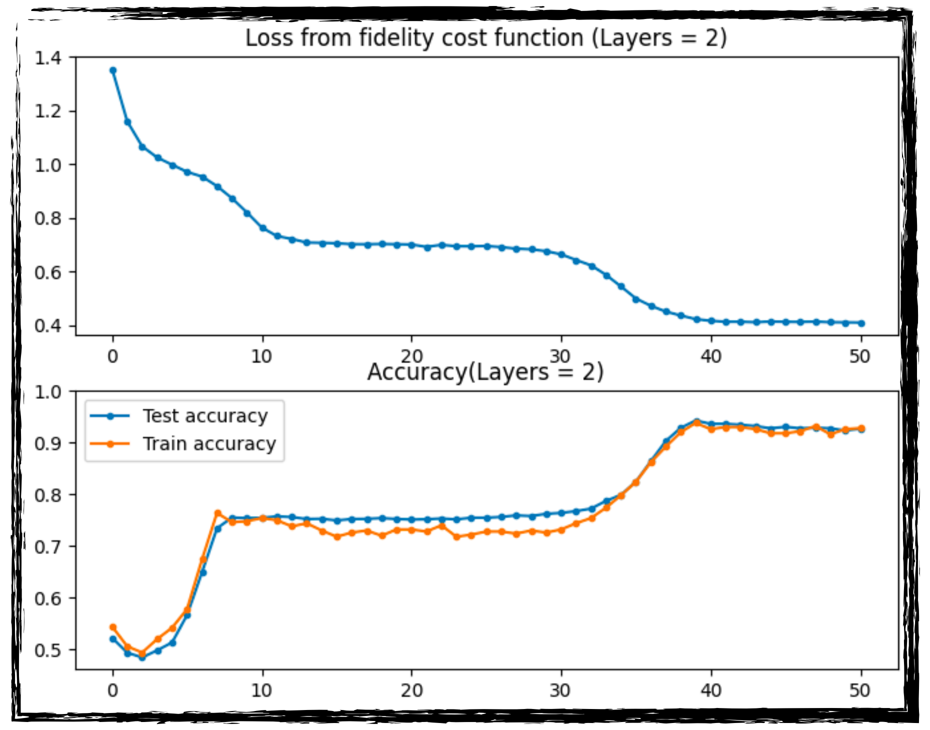
Reproducing Results with Pennylane(Single qubit Classifier)

Circle(Compressed Scheme)

Layer 2

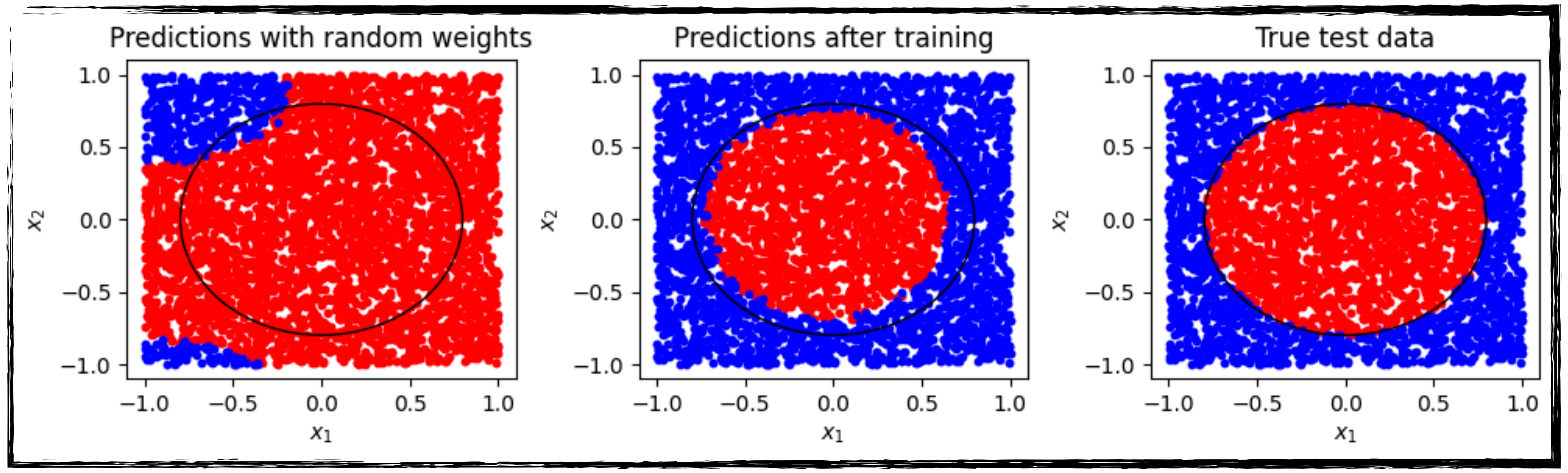
0.93

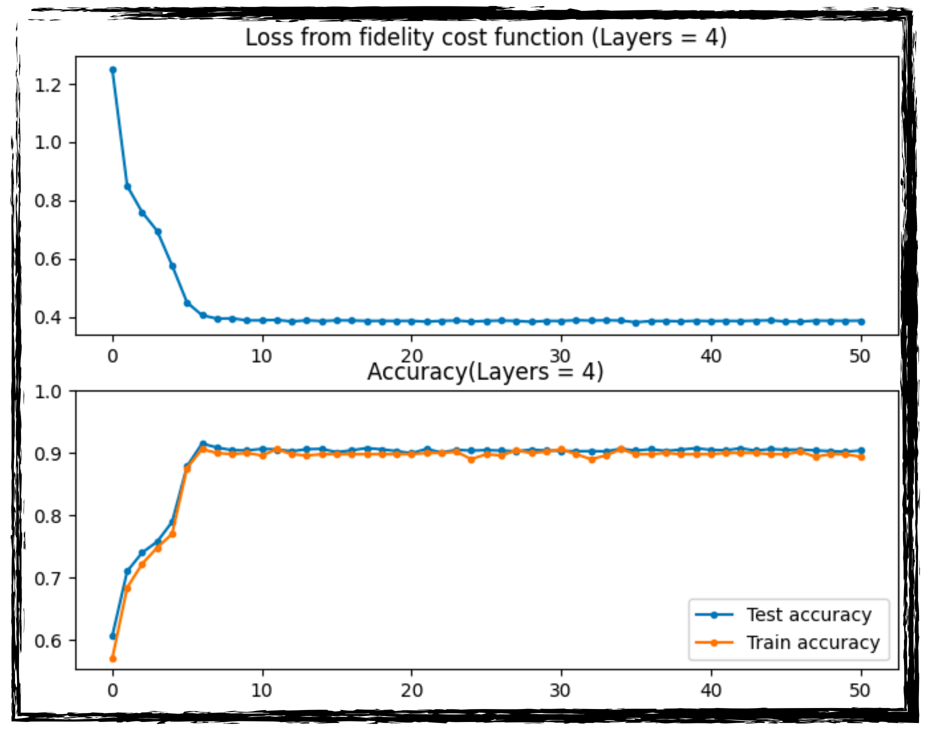




Layer 4

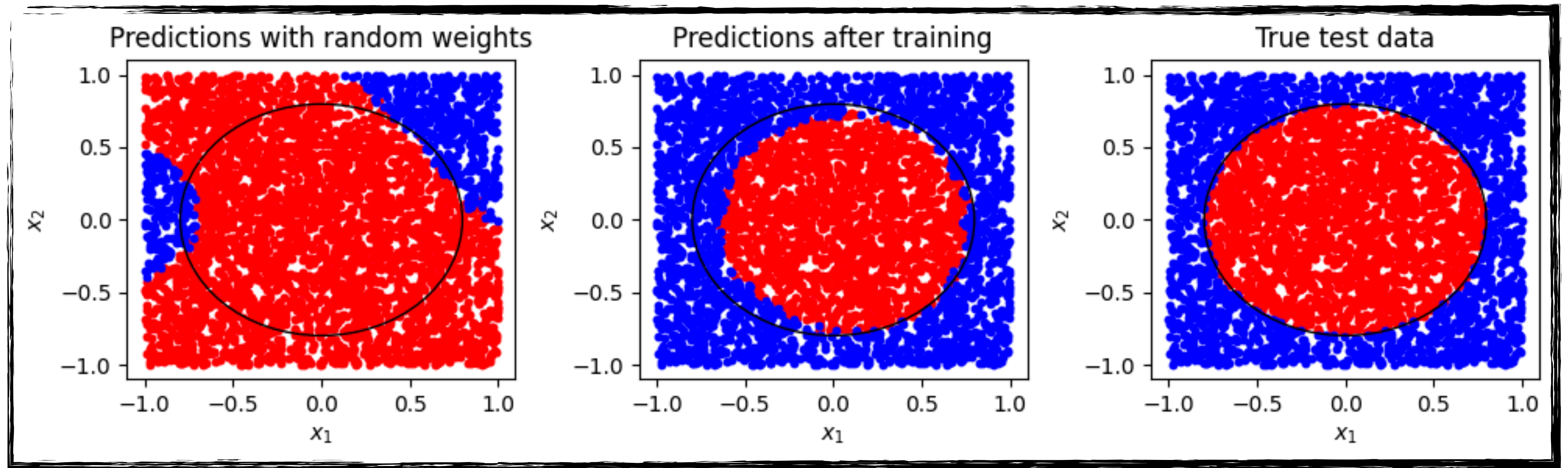
0.91

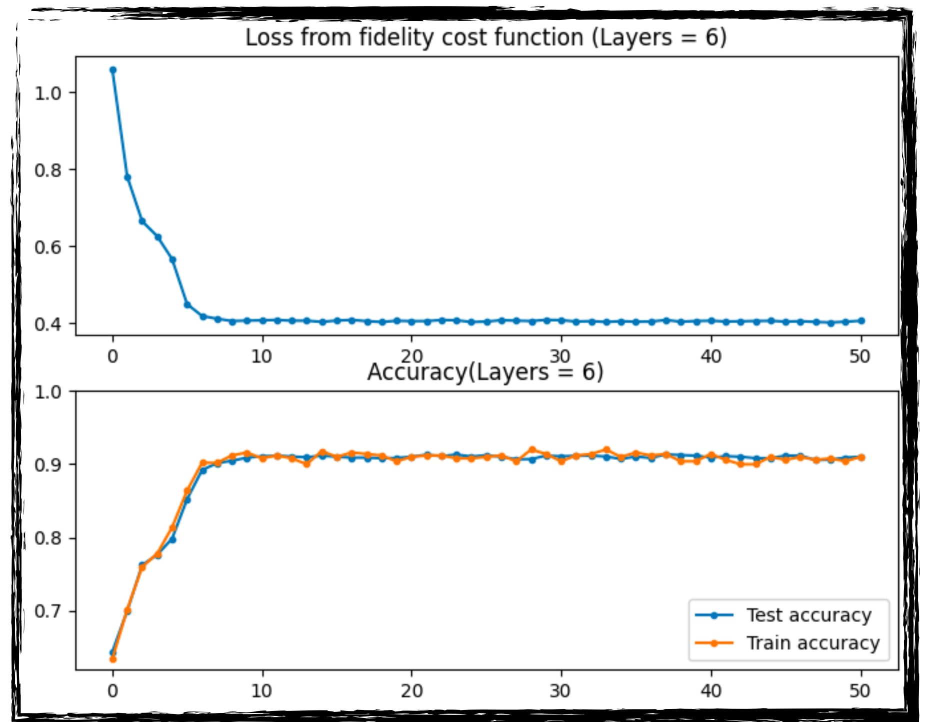




Layer 6

0.91





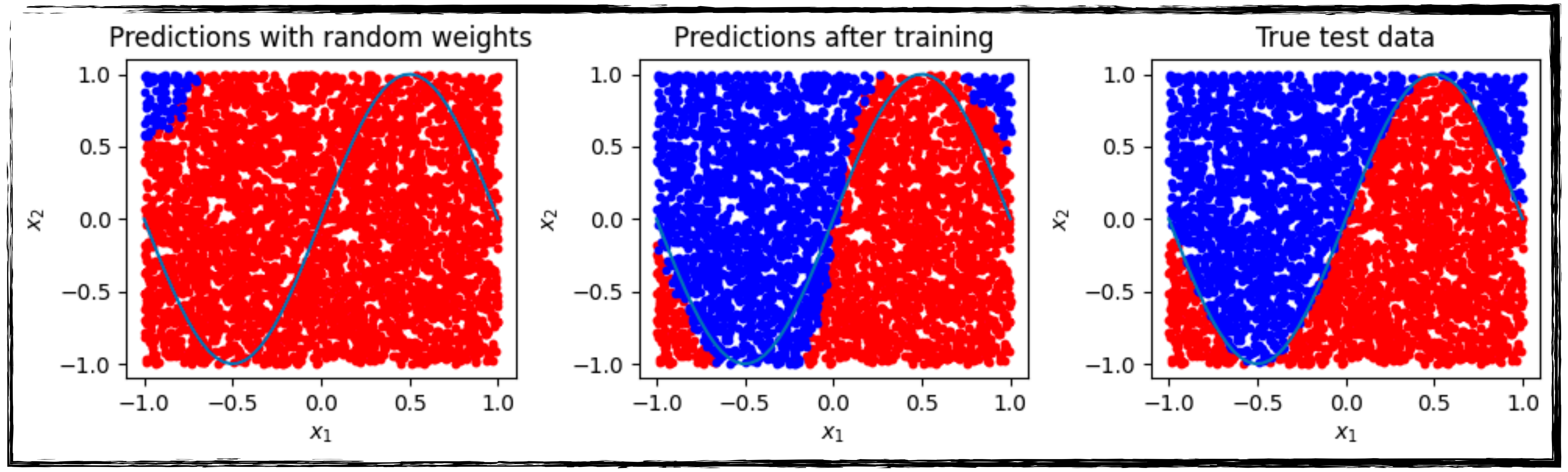
Data Re-uploading

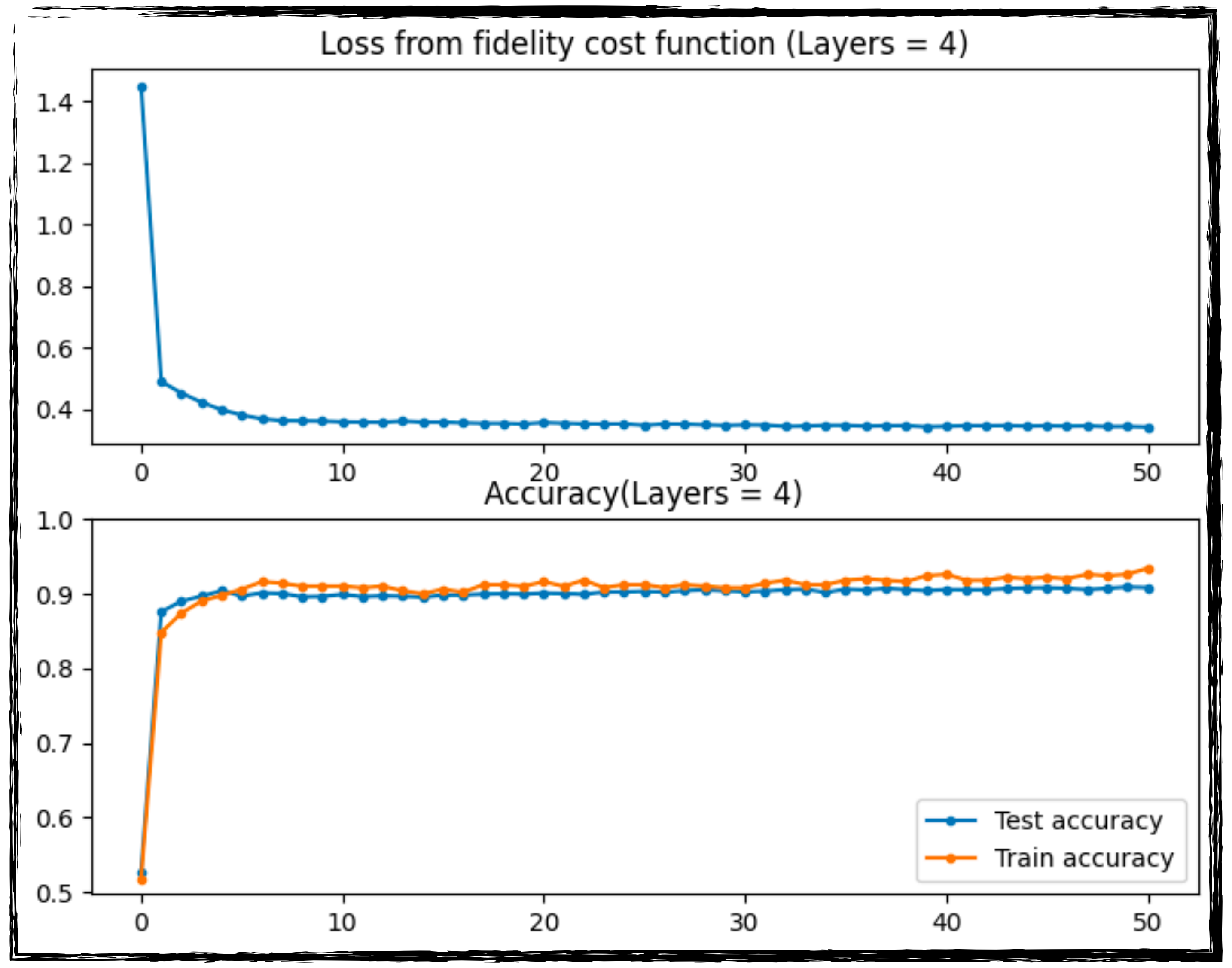
Reproducing Results with Pennylane(Single qubit Classifier)

Sin(Compressed Scheme)

Layer 4

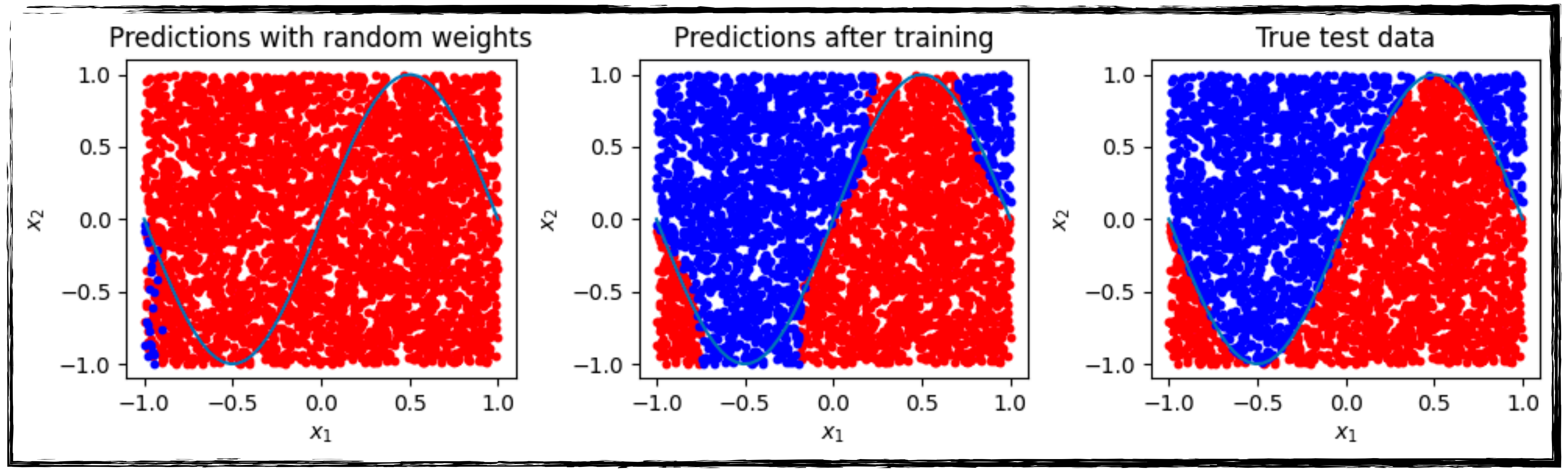
0.90

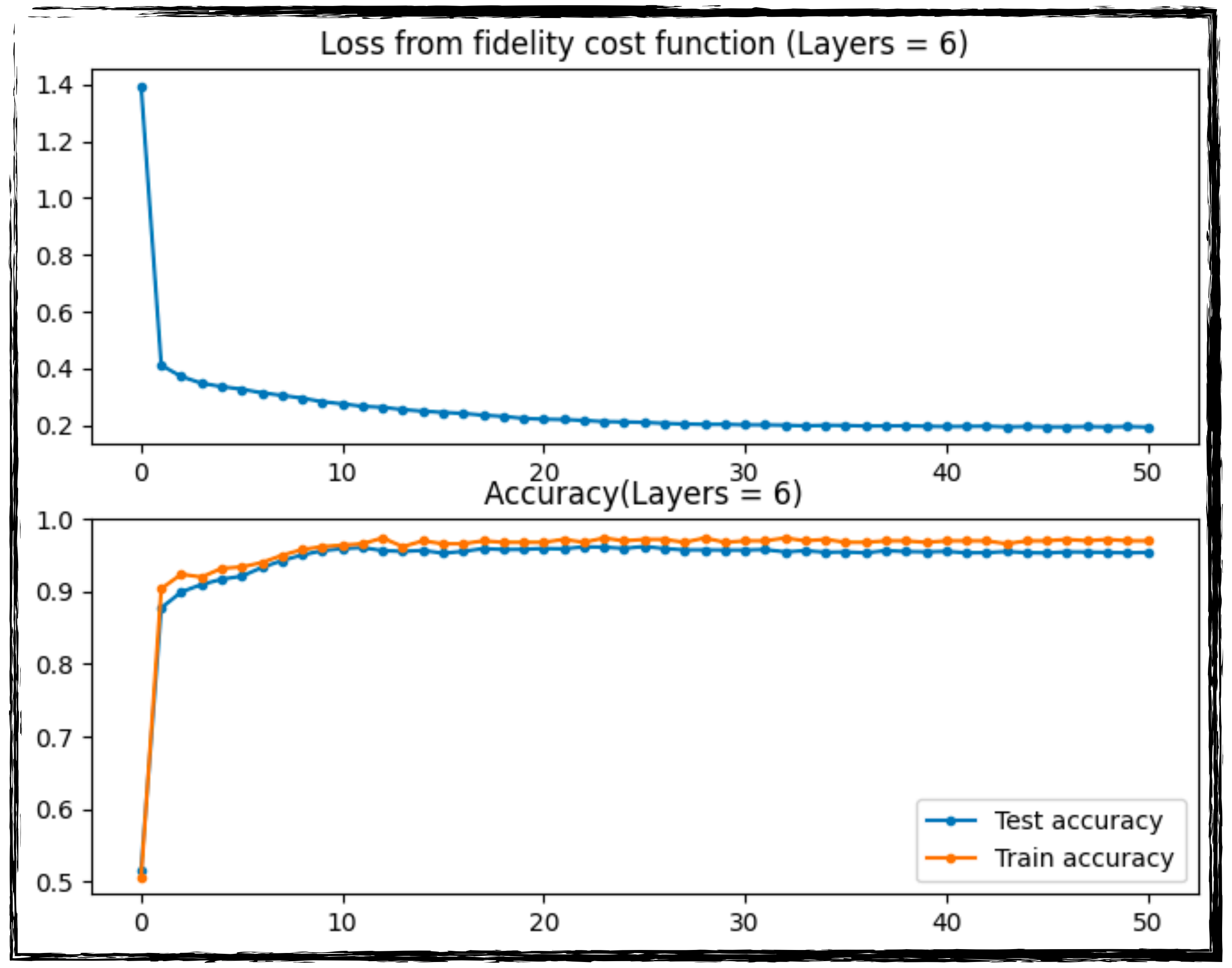




Layer 6

0.94





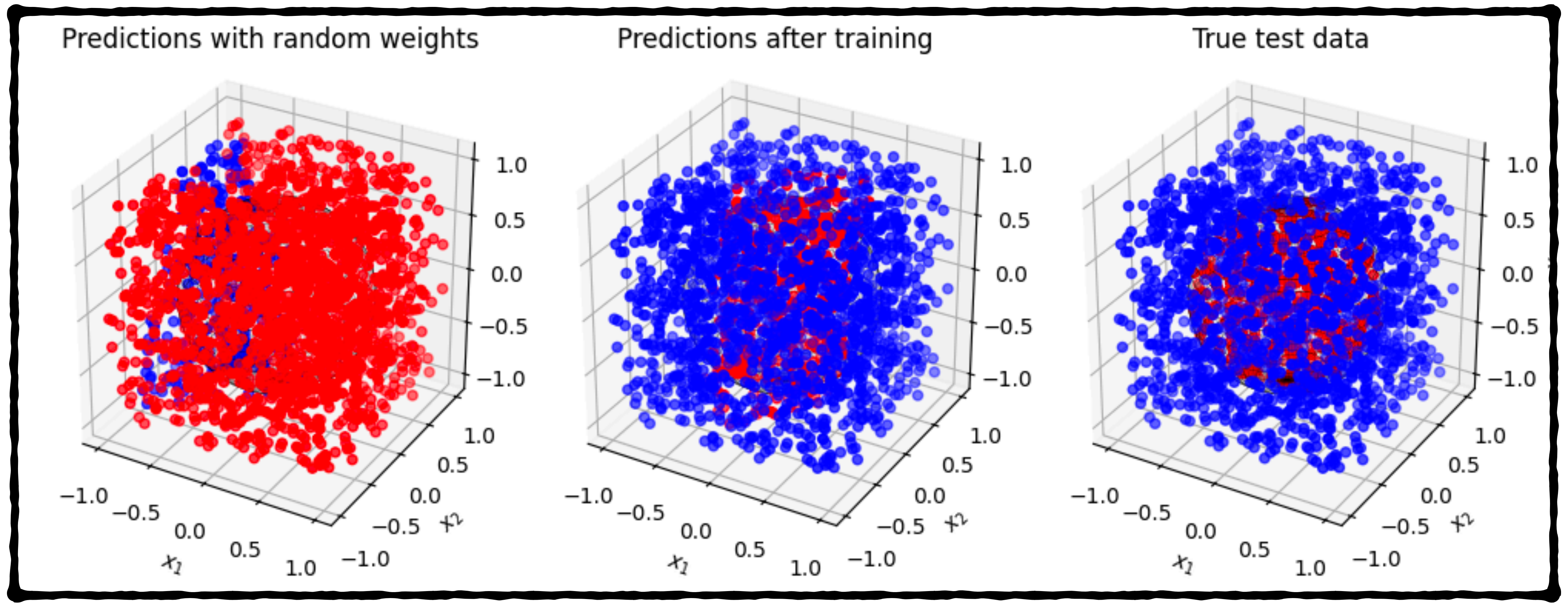
Data Re-uploading

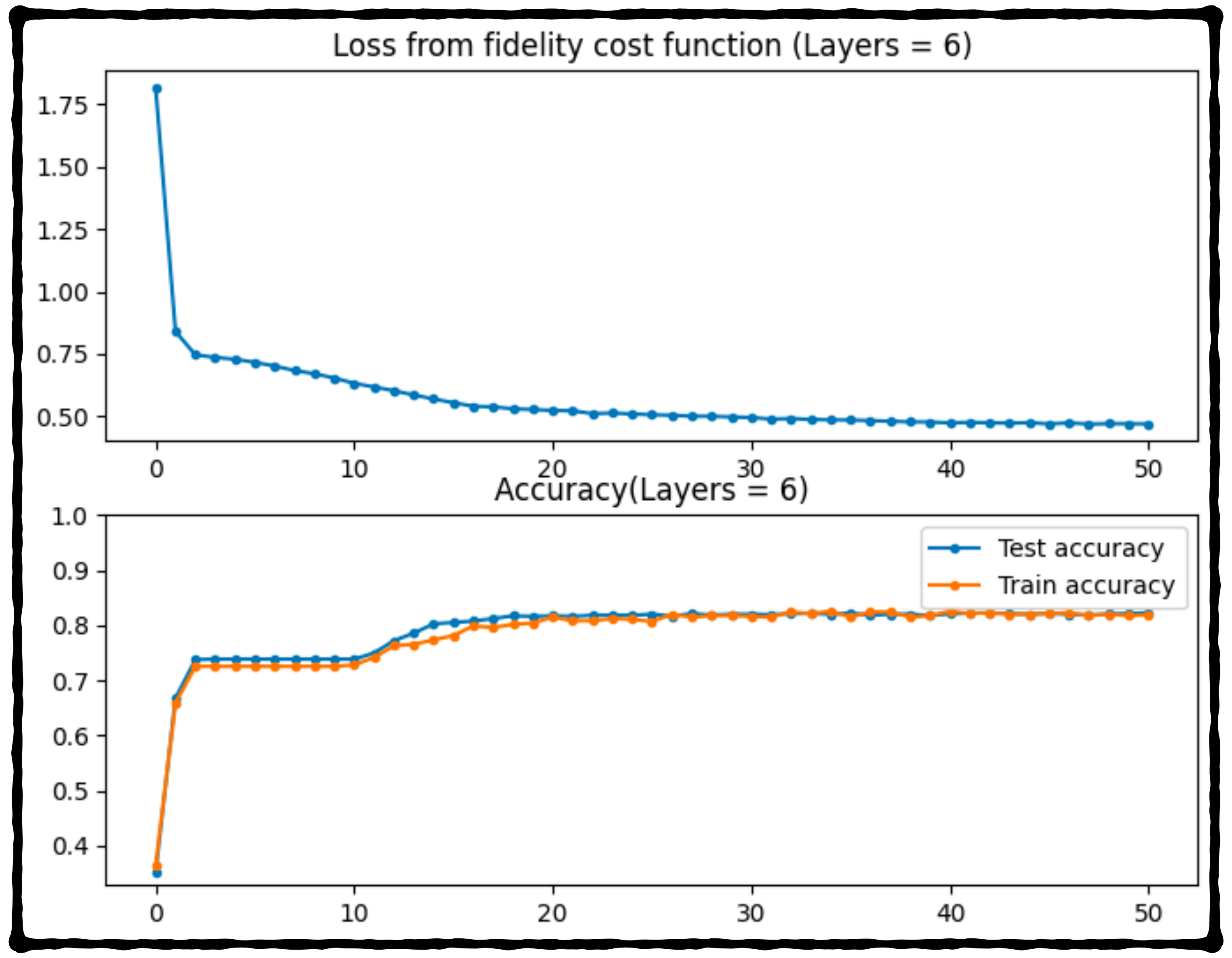
Reproducing Results with Pennylane(Single qubit Classifier)

Sphere(Compressed Scheme)

Layer 6

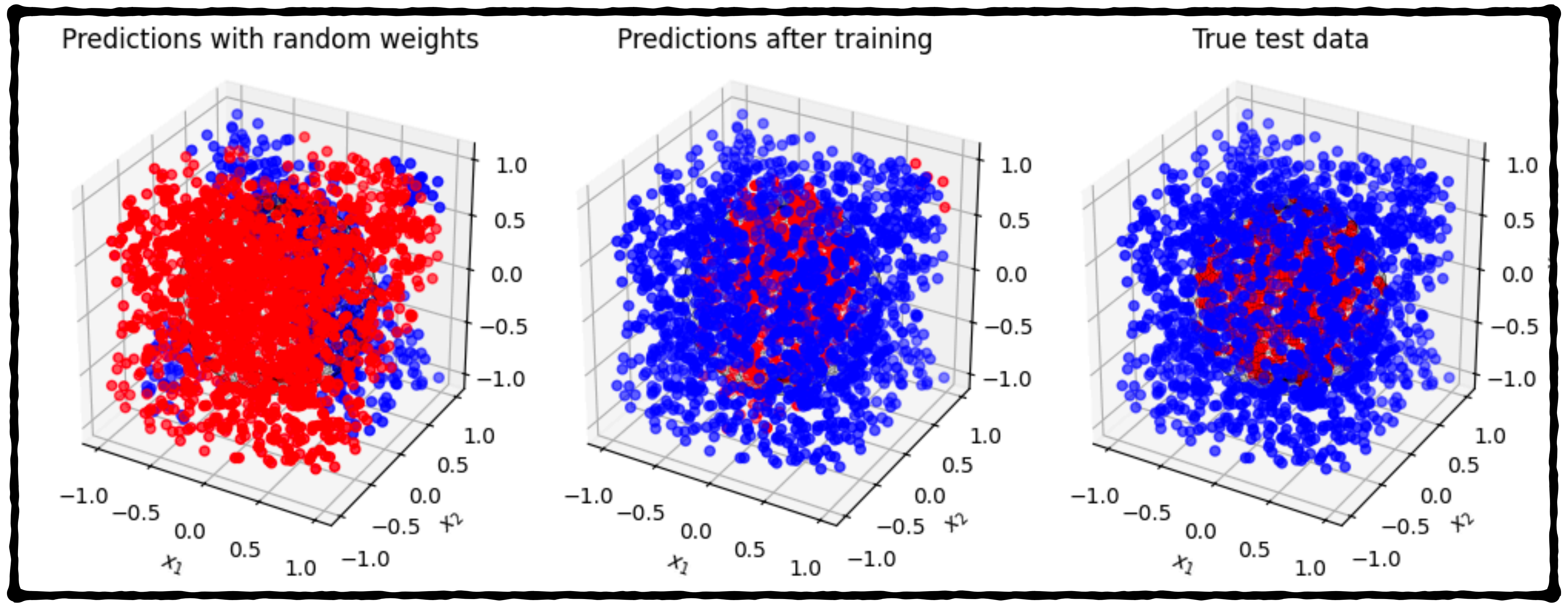
0.82

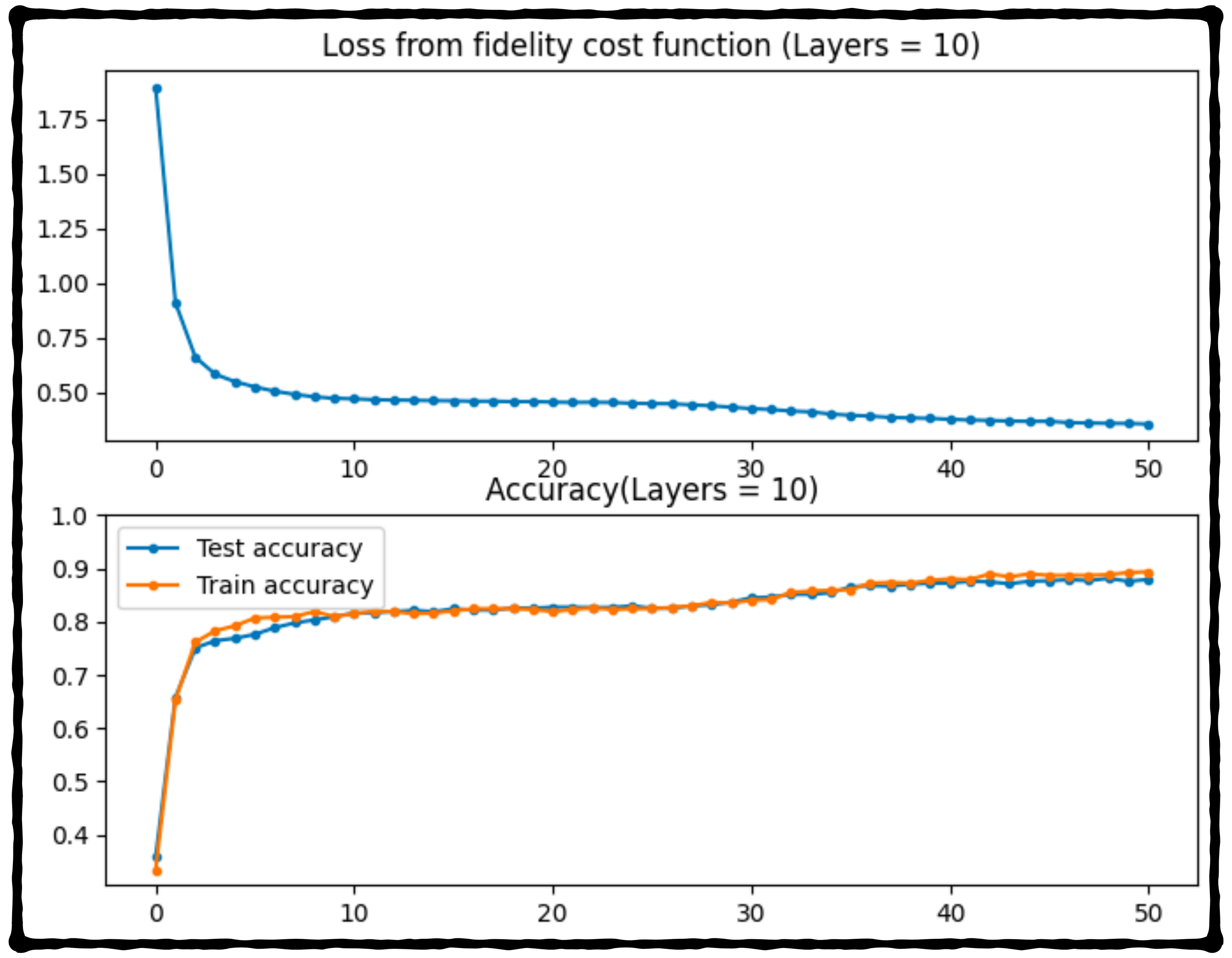




Layer 10

0.85





Data Re-uploading

Question

1) Correction in Universal Approximation theorem for qubit gate ?

$$\mathcal{U}(\vec{x}) = \exp\left[i\sum_{i=1}^{N}\vec{\omega}(\vec{\phi}_i(\vec{x}))\cdot\vec{\sigma} + \mathcal{O}_{corr}\right].$$

In the paper, they derive the universal approximation theorem for qubit gate....

$$\mathcal{U}(\vec{x}) = e^{i\vec{\xi}(\vec{x})\cdot\vec{\sigma}} = e^{i\vec{f}(\vec{x})\cdot\vec{\sigma}+i\vec{g}(\vec{x})\cdot\vec{\sigma}}.$$

Is it ignorable???

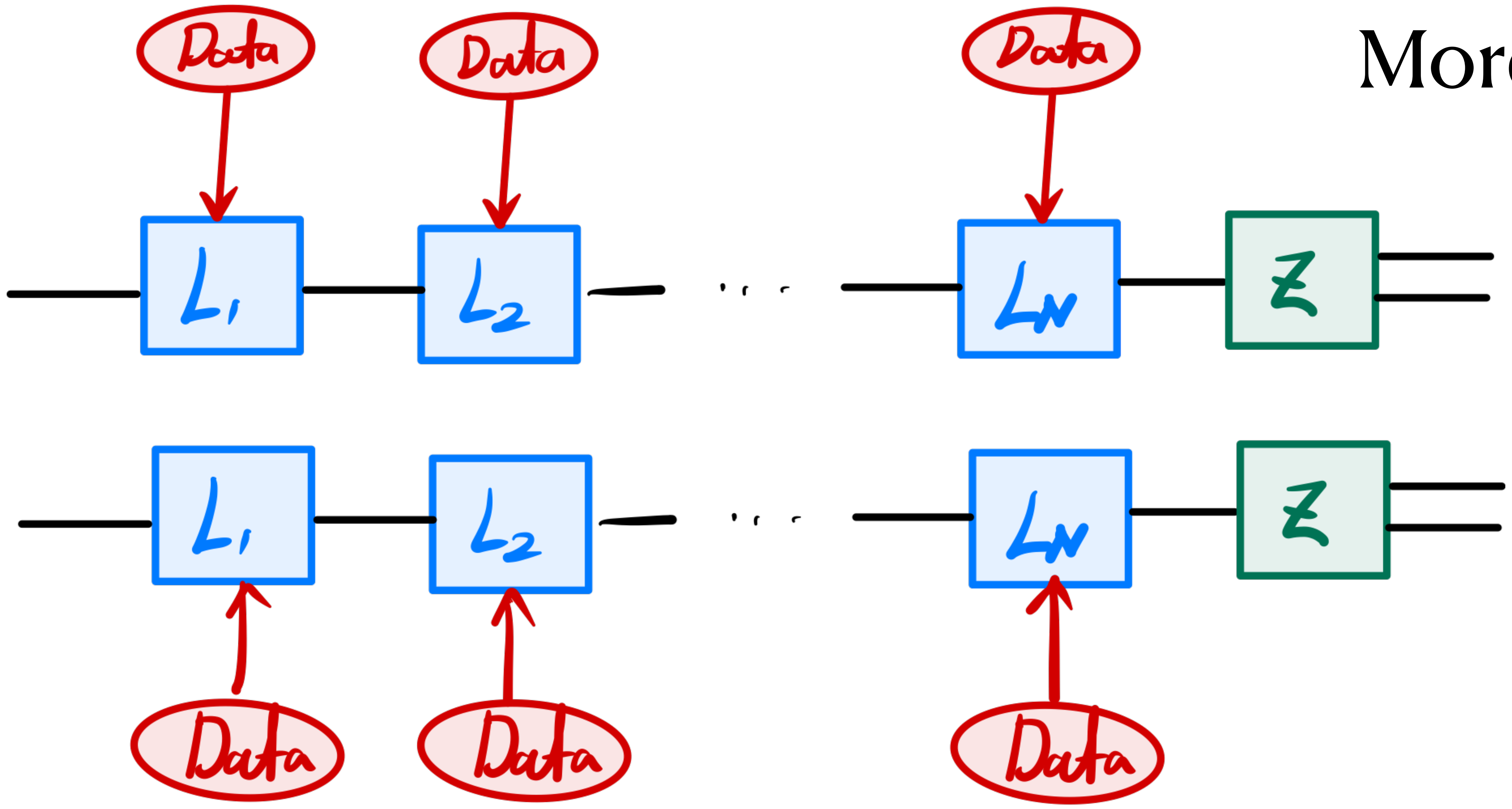
⇒ I think this lead us to using the compressed scheme

Data Re-uploading

Question

1) What is the benefit when we use more qubits?

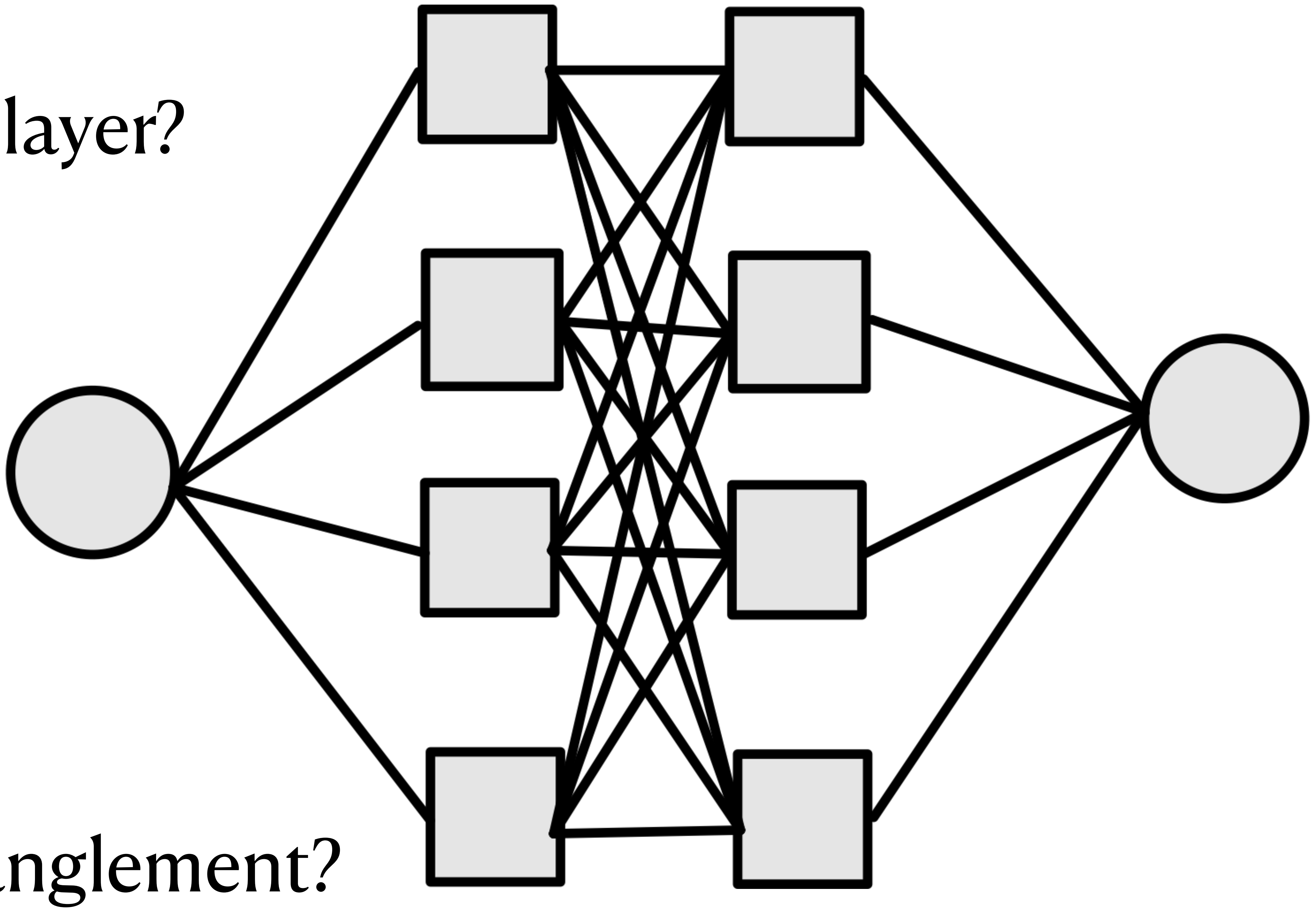
2) How can we compare or correspond with deep neural Network?



More hidden layer?

VS

Benefits of Entanglement?



Parameter Shift Rule

In Ref. 1, It is used for gaining <u>the loss function</u>.

PSR have some benefits to compute both quantum function and the gradient of the quantum function

In Ref 1. ,

<u>**Training the derivative of VQC such that it appoximates the function '*g*' at any point in the integration limits**</u>

$$I(\boldsymbol{\alpha}) = G(x_b; \boldsymbol{\alpha}) - G(x_a; \boldsymbol{\alpha}), \qquad G(x; \boldsymbol{\alpha}) = \int g(\boldsymbol{\alpha}; x)\mathrm{d}x.$$

Parameter Shift Rule

For single parameter,

$$f(x;\theta_i) = \langle 0 \,|\, U_0^\dagger(x) U_i^\dagger(\theta_i) \hat{O} U_i(\theta_i) U_0(x) \,|\, 0 \rangle$$

$$= \langle x \,|\, U_i^\dagger(\theta_i) \hat{O} U_i(\theta_i) \,|\, x \rangle$$

$$\leftarrow U_i^\dagger(\theta_i) \hat{O} U_i(\theta_i) = M_{\theta_i}(\hat{O})$$

$$\Rightarrow \nabla_{\theta_i} f(x;\theta_i) = \langle x \,|\, \nabla_{\theta_i} M_{\theta_i}(\hat{O}) \,|\, x \rangle \in \mathbb{R}$$

$$\Leftarrow \nabla_{\theta_i} M_{\theta_i}(\hat{O}) = c[M_{\theta_{i+s}}(\hat{O}) - M_{\theta_{i-s}}(\hat{O})]$$

$\sim$ Numerical finite difference method for computing derivatives

For Pauli Gate example $U_i(\theta_i) = \exp(-i\frac{\theta_i}{2}\hat{P}_i)$ where $\hat{P}_i$ is a Pauli operator

$$\Rightarrow \nabla_\theta f(x;\theta) = \frac{1}{2}[f(x;\theta + \frac{\pi}{2}) - f(x;\theta - \frac{\pi}{2})]$$

Ref: https://pennylane.ai/qml/glossary/parameter_shift/

Parameter Shift Rule

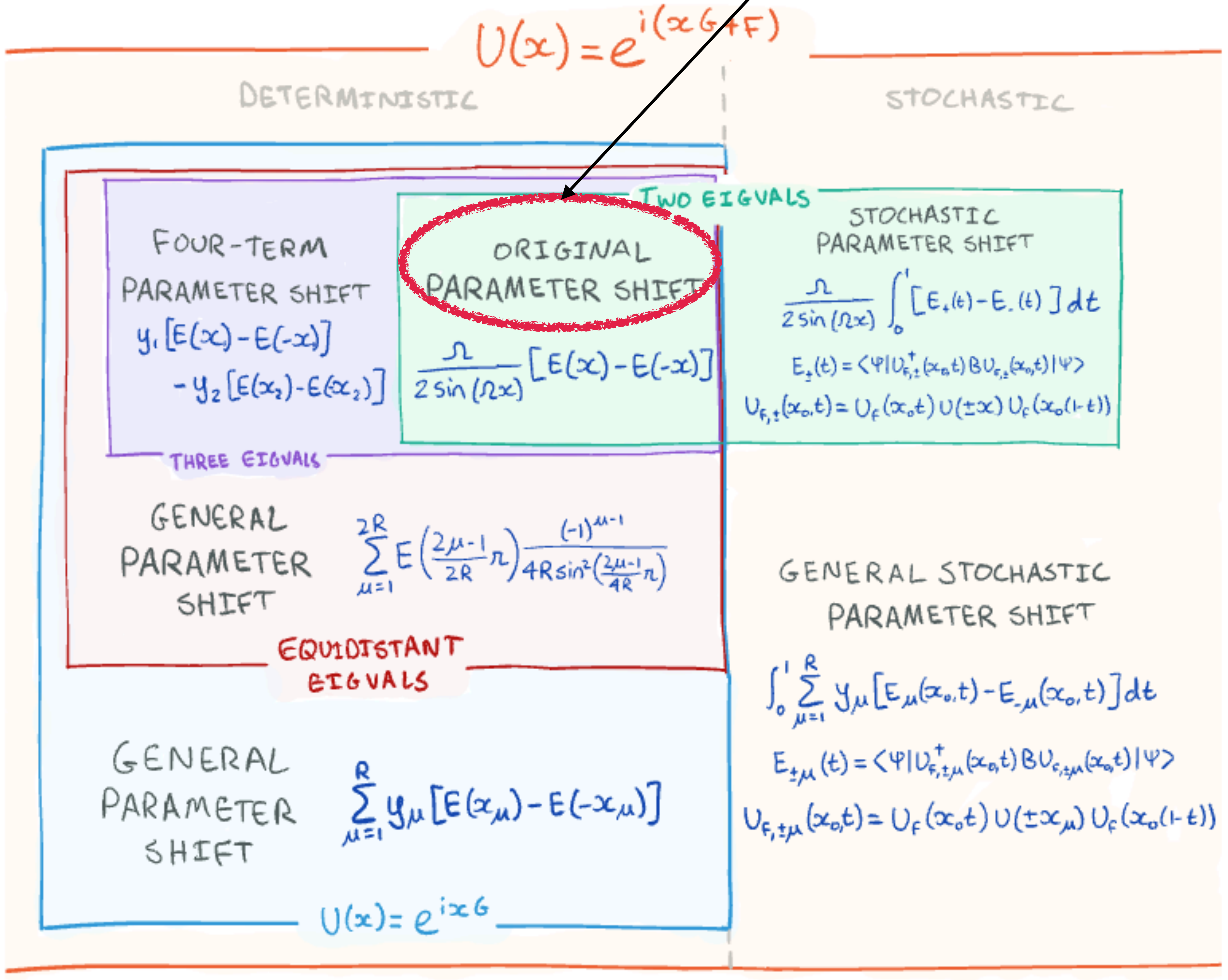
General parameter-shift rules for quantum gradients

David Wierichs[1,2], Josh Izaac[1], Cody Wang[3], and Cedric Yen-Yu Lin[3]

[1]Xanadu, Toronto, ON, M5G 2C8, Canada
[2]Institute for Theoretical Physics, University of Cologne, Germany
[3]AWS Quantum Technologies, Seattle, Washington 98170, USA

In paper...? Simplest way

$$g(\mu) \equiv \partial_\mu G(\boldsymbol{\theta}) = r(G(\mu^+) - G(\mu^-)),$$



Then...

How about using the other rules?

I will upload reviewing and reproducing about it in my GitHub...