Third (Group) Assignment

Marijana Petojevic – 12017529
Vedad Hadzic – 12042758
Sam de Vries – 12503805

191.023 Hybrid Quantum–Classical Systems
Winter Semester 2025

Date of Submission: January 23, 2026

1. Variational Quantum Algorithms: QML

Note: The complete implementation, including detailed explanations, is provided in the notebook `HQCS2025_GROUP_10_ASSIGNMENT3_EX_1.ipynb`.

In this task, we investigated variational quantum classifiers (VQCs) and the effect of data re-uploading on their performance. We evaluated a baseline classifier and a re-uploading variant on three datasets of increasing difficulty: Glühweindorf, Lebkuchenstadt, and Krampuskogel. The datasets differ in noise level and class overlap, allowing us to analyze how data complexity affects training behavior and generalization.

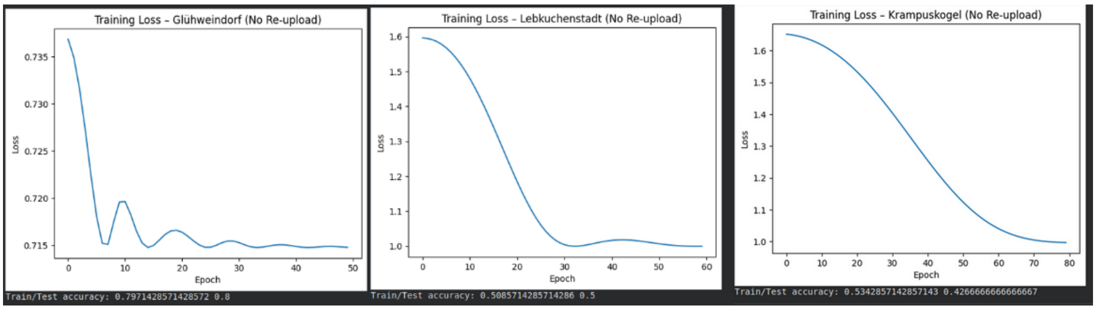

Figure 1: Training loss of the variational quantum classifier on all three datasets without data re-uploading.

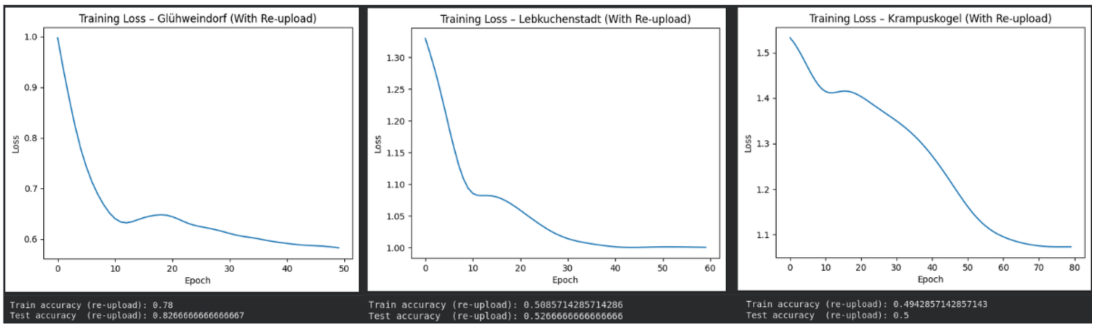

Figure 2: Training loss of the variational quantum classifier on all three datasets with data re-uploading.

For the Glühweindorf dataset, both the baseline and re-upload models achieve high training and test accuracy, indicating that the data is relatively clean and easy to classify. Data re-uploading leads to faster convergence and a small improvement in test accuracy, showing that increased circuit expressivity can slightly enhance performance when the underlying structure is simple.

The Lebkuchenstadt dataset is more challenging due to systematic noise. While data re-uploading improves training loss and optimization stability, the classification accuracy remains close to random guessing for both models. This shows that improved expressivity does not necessarily translate into better generalization when noise and class overlap dominate the data.

Krampuskogel is the most difficult dataset. Both baseline and re-upload models struggle to generalize, although data re-uploading leads to a small improvement in test accuracy compared to the baseline. The confusion matrices reveal a strong bias toward predicting (see Figure 3) a single class, highlighting the limitations of variational quantum classifiers on highly noisy data with limited model capacity.

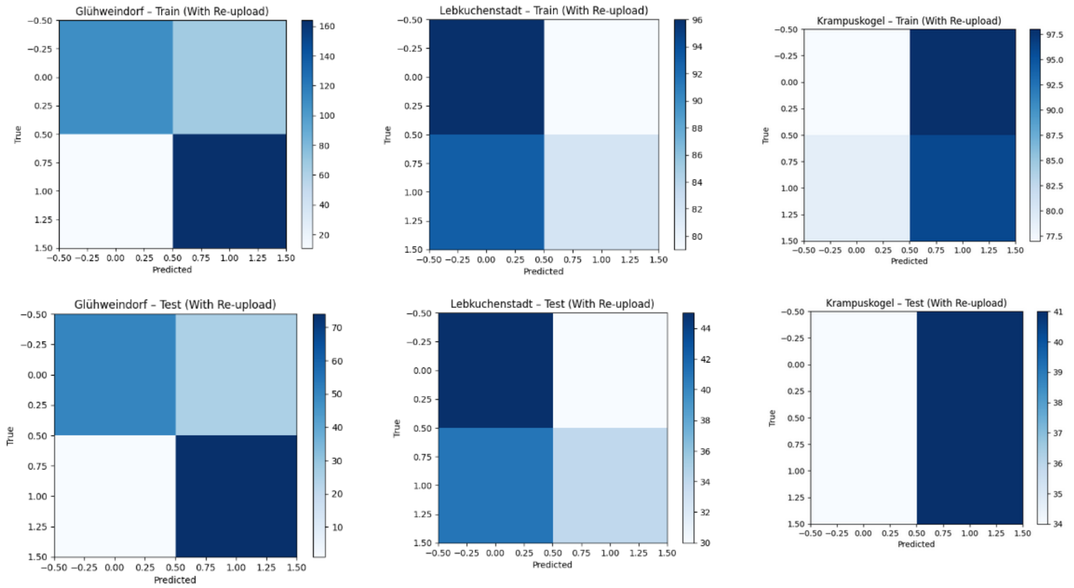

Figure 3: Confusion matrix of the re-uploading variational quantum classifier on train and test datasets.

Overall, data re-uploading consistently improves training dynamics and optimization behavior, but its impact on test performance strongly depends on dataset difficulty. These results demonstrate both the potential and the limitations of variational quantum machine learning models in realistic, noisy settings.

2. Classical Shadows

Note: The full implementation with detailed explanations can be found in the notebook `HQCS2025_GROUP_10_ASSIGNMENT3_EX_2.ipynb`.

In this exercise, we have studied the *classical shadow* framework as an efficient method for estimating expectation values of many quantum observables using randomized measurements.

We first constructed a quantum circuit acting on eight qubits. The circuit consists of Hadamard layers, randomized single-qubit rotations derived from the matriculation numbers, and entangling CZ gates. This structure creates strong entanglement and spreads information across the entire system, so that it cannot be accessed by single-qubit measurements alone.

Using a statevector simulator, we computed the exact expectation values of several Pauli observables, which serve as reference values. The results showed that observables acting on a single qubit have expectation values close to zero, meaning these gifts are neither likely nor unlikely. In contrast, the two-qubit observable X_1X_2 have a strong positive expectation value, which refers us to a dominant correlation between qubits 1 and 2. Interpreted in the context of the task, this suggests that the most likely gift is a new bike. To estimate the same observables using classical shadows, we performed random Pauli measurements on all qubits. The required number of samples depends on the *shadow norm* of the observables. For Pauli observables measured with random Pauli bases, the square of the shadow norm scales as 3^k, where k is the number of qubits the observable acts on. As a result, two-qubit observables require more samples than single-qubit observables.

Using the classical shadow sample complexity bound, we obtained the following required numbers of samples for different target accuracies ε (with 99% confidence):

- $\varepsilon = 0.1 \quad \Rightarrow \quad N \geq 12{,}763$
- $\varepsilon = 0.05 \quad \Rightarrow \quad N \geq 51{,}049$
- $\varepsilon = 0.005 \quad \Rightarrow \quad N \geq 5{,}104{,}856$
- $\varepsilon = 0.001 \quad \Rightarrow \quad N \geq 127{,}621{,}384$

This clearly illustrates the $1/\varepsilon^2$ scaling, as predicted before.

We then analyzed the convergence behavior of the classical shadow estimates by increasing the number of measurement snapshots in steps of 100 and comparing the estimates to the exact expectation values for $\varepsilon = 0.05$.

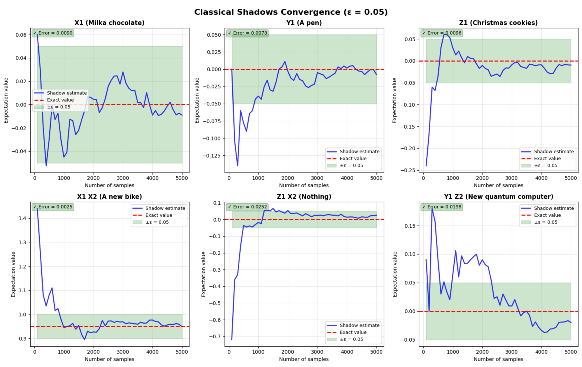

Figure 4: Convergence of classical shadow estimates compared to exact values.

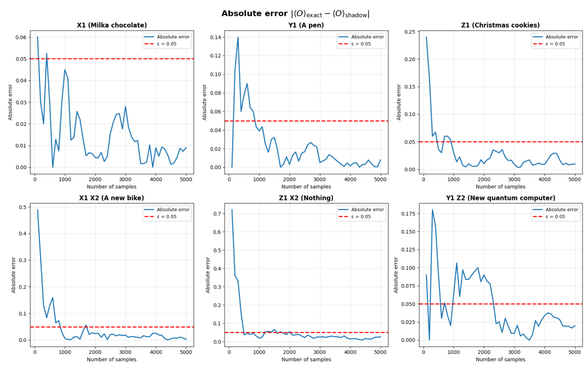

Figure 5: Absolute error $|\langle O\rangle_{\text{exact}} - \langle O\rangle_{\text{shadow}}|$ as a function of samples.

The plots show that single-qubit observables converge quickly and remain close to zero, which confirms us that individual qubits are locally balanced. The two-qubit observable X_1X_2 converges to a strong positive value, clearly revealing the dominant correlation between qubits 1 and 2. Other two-qubit observables converge more slowly, which is expected due to their higher variance and weaker signal.

Finally, we analyzed the absolute error $|\langle O\rangle_{\text{exact}} - \langle O\rangle_{\text{shadow}}|$ as a function of the number of samples. The error decreases steadily as more snapshots are used, and all observables eventually fall below the target accuracy $\varepsilon = 0.05$.

Single-qubit observables reach this threshold faster, while two-qubit observables require more samples, in agreement with the theoretical predictions based on the shadow norm.

3. BONUS: QML + Classical Shadows

Note: The complete implementation, including detailed explanations, is provided in the notebook `HQCS2025_GROUP_10_ASSIGNMENT3_EX_3.ipynb`.

In this exercise, we studied how noise affects the training trajectory of a variational quantum classifier (VQC). Additionally, we gained insight into the use of classical shadows in the presence of noise. This was realized by taking the VQC from Task 1 and adding noise to the circuit. The training process was done with noise. For the evaluation on the test data set, we used a circuit returning a classical shadow. We did this for different gradations of noise to investigate the effect of noise on:

- Training loss curves
- Gradient norms over time
- Accuracy degradation
- Convergence
- Re-upload

For Glühweindorf, the baseline classifier seems to get heavily impacted by the noise, meaning that the loss converges at a lower value. However, for a low level of noise, the model does show a higher performance on accuracy. When data re-uploading is applied, the noise level heavily impacts the value of loss it converges on and the accuracy.

The Lebkuchenstadt dataset is more challenging due to systematic noise and class overlap. Therefore, the noise seems to affect this data set for the baseline classifier less compared to Glühweindorf. When we apply data re-uploading we see that the accuracy seems to improve with noise and the loss value drops fast for higher levels of noise.

Krampuskogel is the most difficult dataset. This is because it is a messy dataset from itself. Therefore, we also observe that the noise does not really affect this dataset for the baseline classifier. For the data re-uploading classifier, we see that the noise does affect the loss curve. Although the accuracy seems best for a higher level of noise, it still only performs slightly above chance.

The overall consensus over this exercise was that the messier the dataset, the less it is affected by noise. For that reason, Glühweindorf was most affected by the noise and

Krampuskogel the least. We also noticed that small amounts of noise seems to improve the classifier on each dataset.

4. Quantum Circuit Compilation

Note: The complete implementation, including detailed explanations, is provided in the notebook `HQCS2025_GROUP_10_ASSIGNMENT3_EX_4.ipynb`.

In this exercise, we discovered circuit compilation under varying circumstances. Circuit compilation entails rewriting a quantum circuit into a form that are usable by the backend. This can be realised by, amongst other techniques, changing gates and swapping qubit positions. At first, we discovered that transpiling a circuit into suitable gates can be done using different optimization levels. For the exercise, we had to transpile a circuit using optimization levels 0 and 3 (OL0 and OL3). This quantum circuit is a Bernstein-Vazirani (BV) algorithm with 5 qubits and '10101' as hidden string implemented in Qiskit.

When comparing the circuits for OL0 and OL3, we noticed that they greatly differed in size of the circuit. We noticed that the max distance in the coupling graph is 2. This caused the circuit for OL0 to substitute a CNOT of a larger distance for many more CNOTs. However, OL3 has the ability to swap qubits. Therefore, the qubits could be swapped in a way that no distance exceeded 2. Resulting in a much smaller graph. The detailed explanation and graphs are included in the corresponding Python notebook.

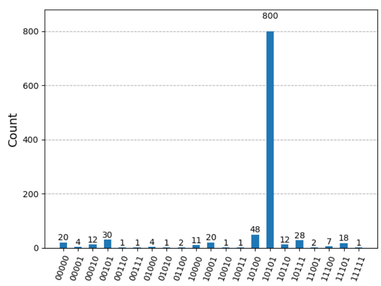

Figure 6: The results of the BV-algorithm using a transpiler with OL0

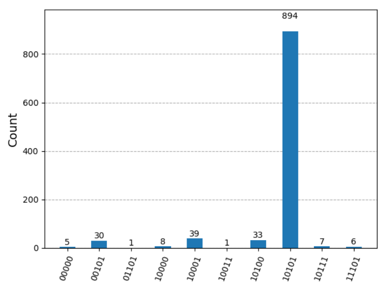

Figure 7: The results of the BV-algorithm using a transpiler with OL3

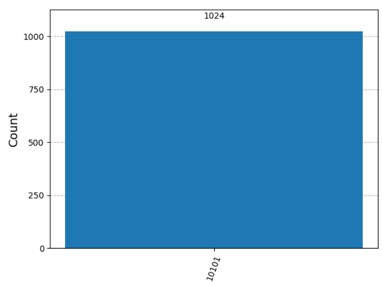

Figure 8: The results of the BV-algorithm using a transpiler of a noise-free backend

Figures 6, 7, and 8 show the results of the BV-algorithm run under different circumstances. Figure 6 shows the highest distribution of possible outcomes, which is most likely caused by the lack of optimization in the transpilation process. Figure 8 uses the AerSimulator as backend, meaning that it is noise-free, which causes the result to be deterministic, resulting in a single possible outcome. The consensus over all of these plots is that the correct outcome is significant over all other outcomes.

The reaming questions are answered detailed in the correpsonding python notebook.

First, we argue that the CNOT can be exchanged for a CZ gate in combination with two

Hadamard gates. Therefore, we can use this to flip a 'CNOT' if that is not allowed.

Finally, we explain how CNOT(q1, q3) can be implemented for a max distance in the coupling graph of one. This can be done using a sequence of CNOT gates.