

Quantum i OsloMet

Sergiy Denysov, Professor OsloMet



OSLOMET



Quantum at OsloMet

Sergiy Denysov

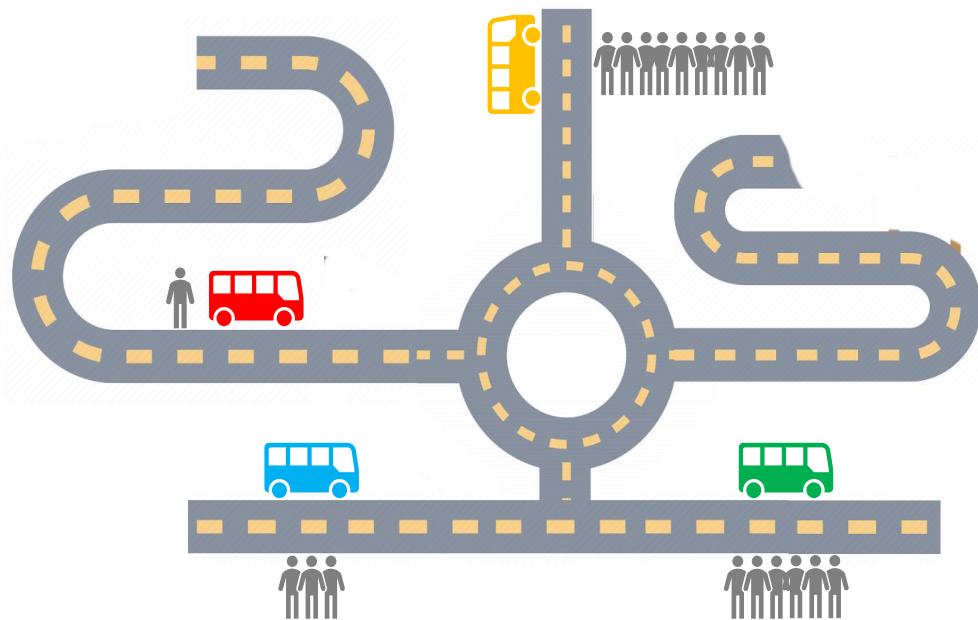
Capacity prediction with Quantum AI



Sebastian Testanière Overskott
OsloMet, MSc student




Aleksandar Davidov
OsloMet, PhD student



- A real-life use case
- Implementation of quantum AI algorithm (*Qboost*) on *D-Wave* annealer (5000+qubits)
- Training the algorithm with an actual database
- Benchmarking and comparative analysis of the performance

Problem Details ▾ Solver Details ▾

Target - QPU ▾

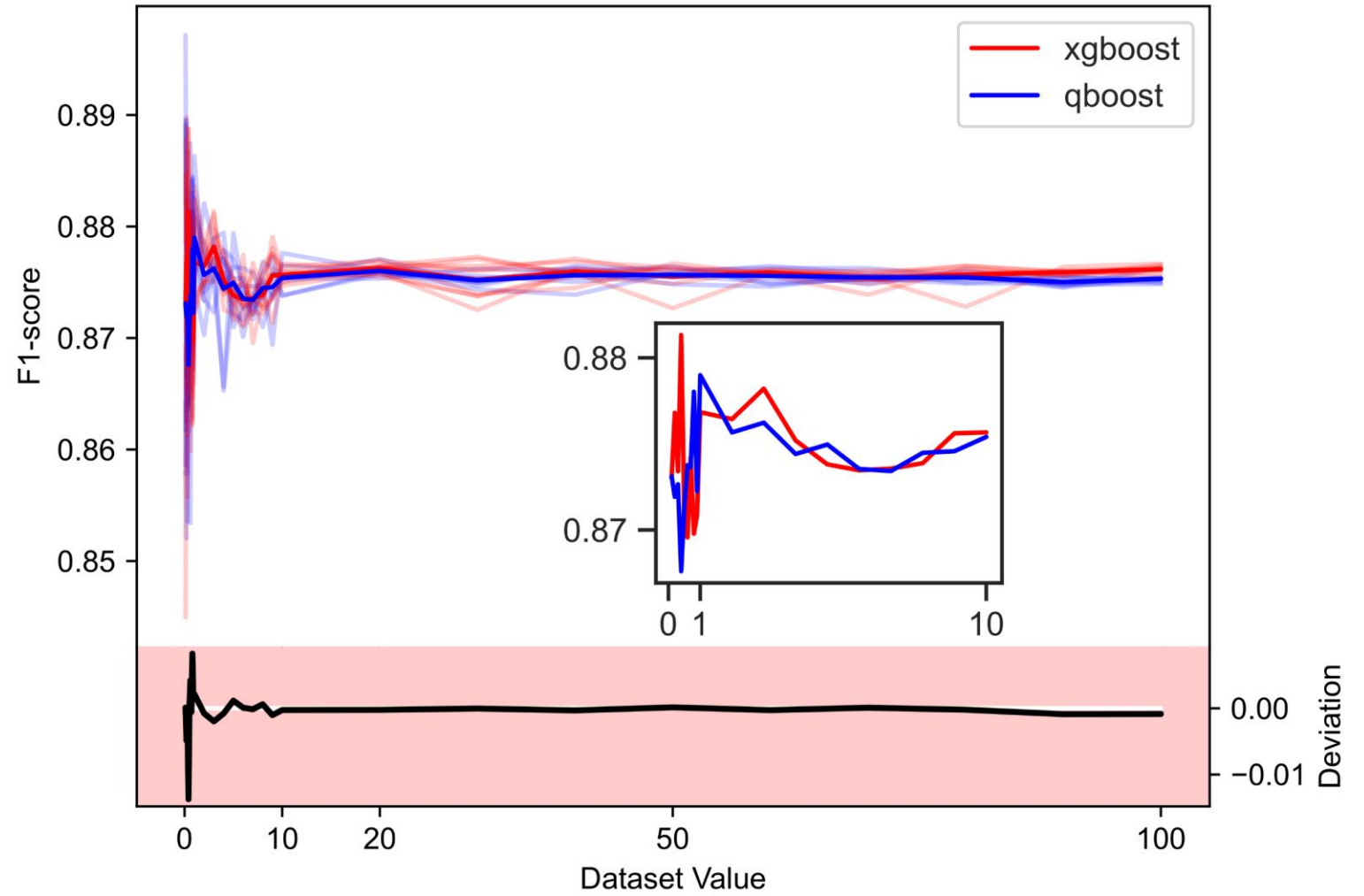


NODE 1734
 VARIABLE v7.0
 BIAS 0.12876699
 SOLUTION -1

Console 1 — 21 OF 21 < >

Type	Description
Chain Length Warning	Chain length greater than 7
Chain Length Warning	Chain length greater than 7
Chain Length Warning	Chain length greater than 7

F1-Score vs Dataset Value



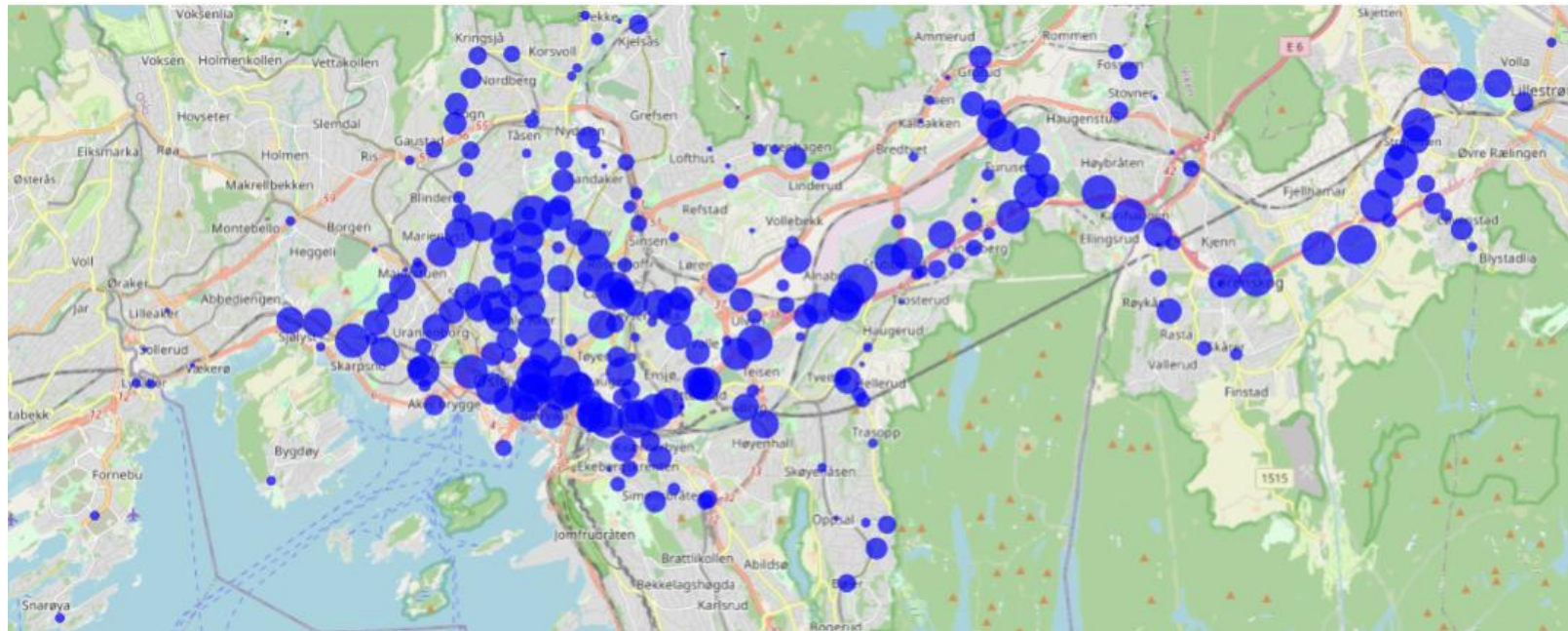
Ruter#

Optimal ticket-control team deployment



Maryam Loffigolian
OsloMet, PhD student

Aleksandar Davidov
OsloMet, PhD student



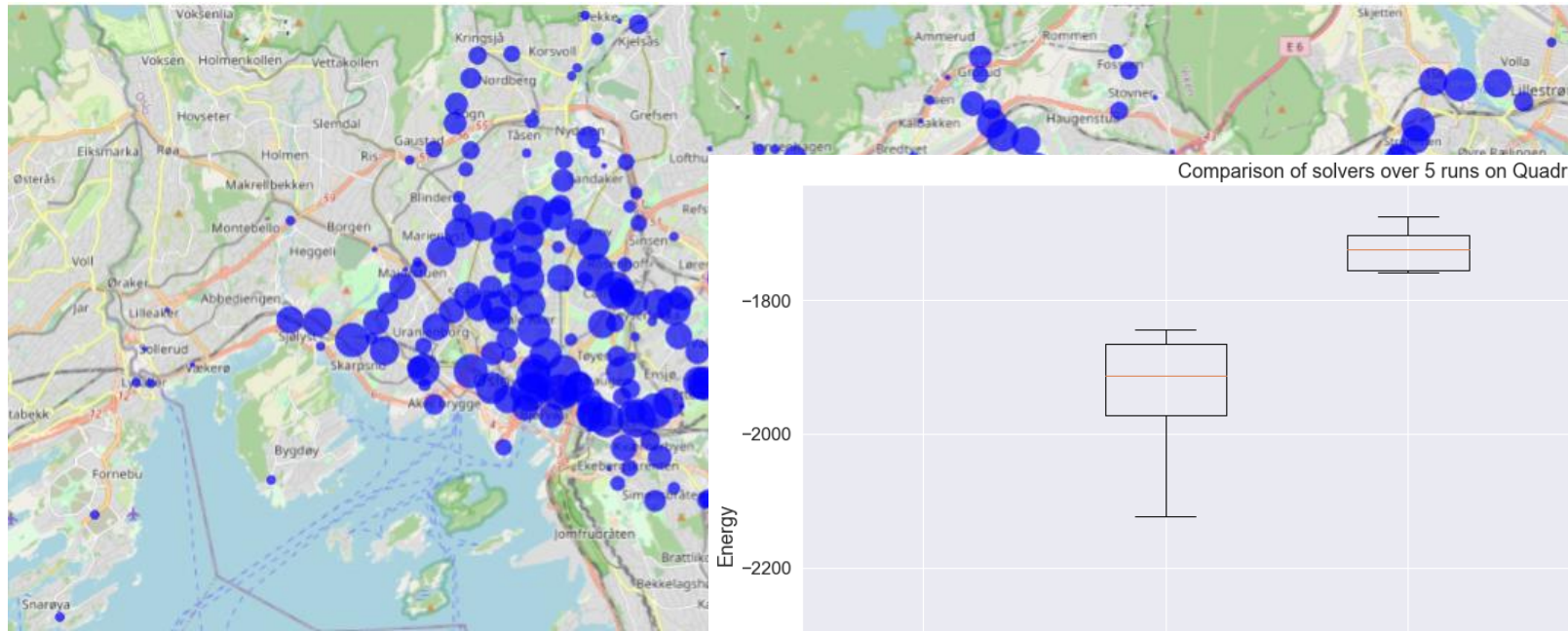
Ruter#

Optimal ticket-control team deployment

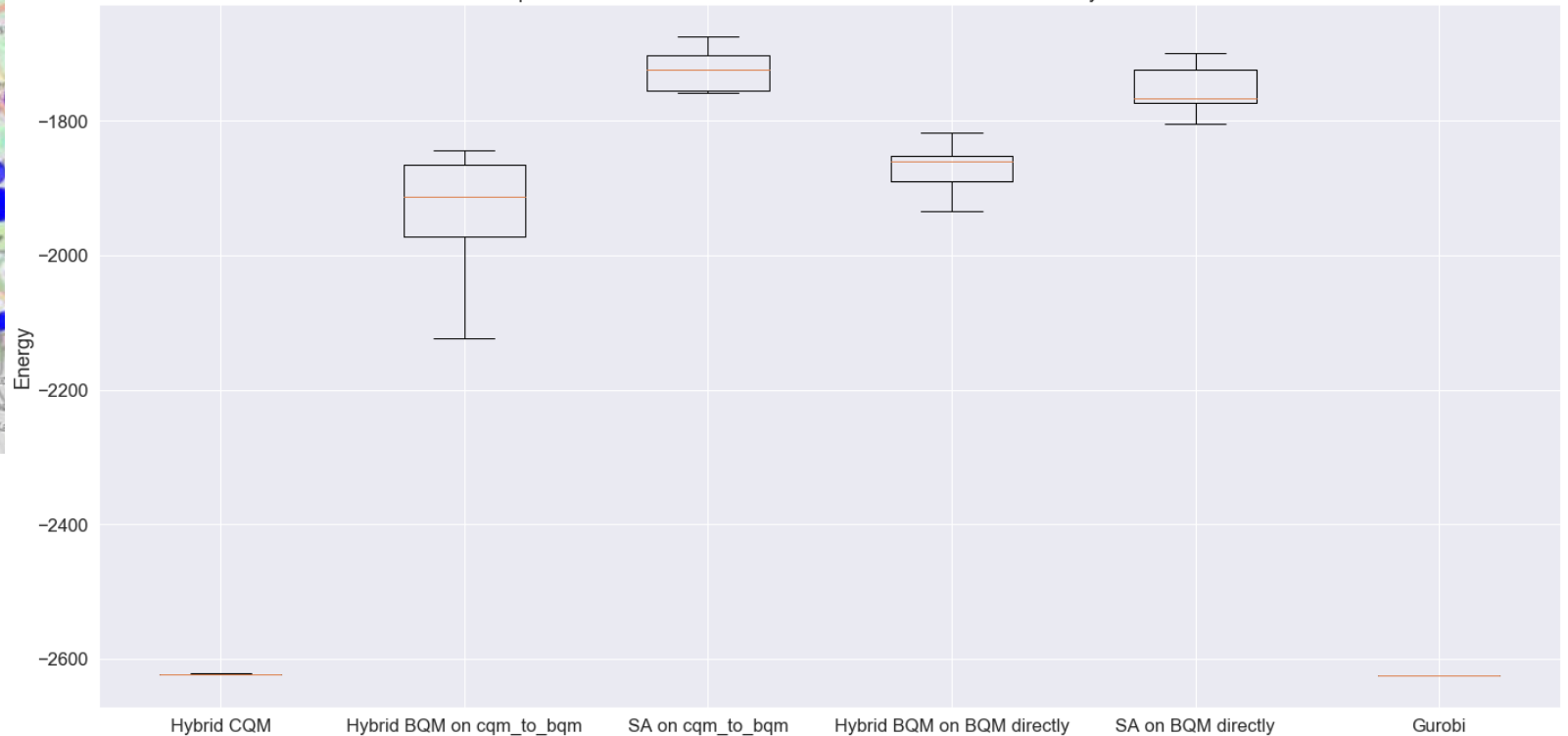


Maryam Lotfigolian
OsloMet, PhD student

Aleksandar Davidov
OsloMet, PhD student



Comparison of solvers over 5 runs on Quadratic Model with 600 binary variables

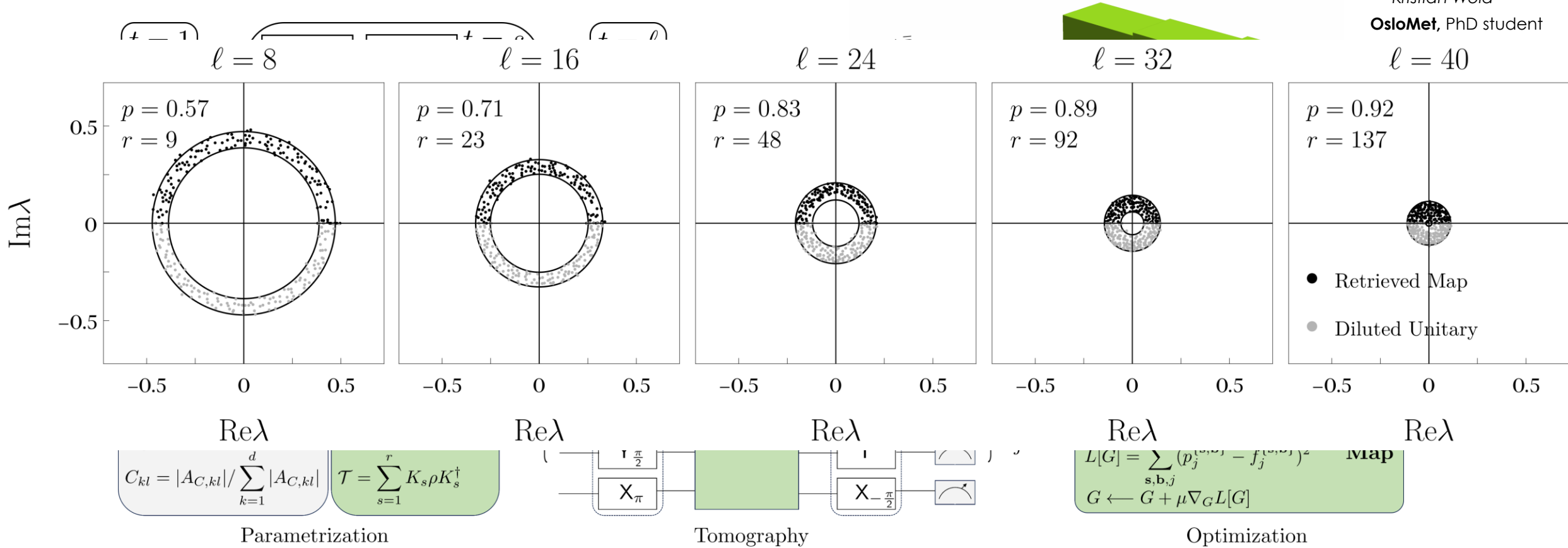


Experiments on Noisy Intermediate Scale Quantum (NISQ) computing platforms



Kristian Wold

OsloMet, PhD student



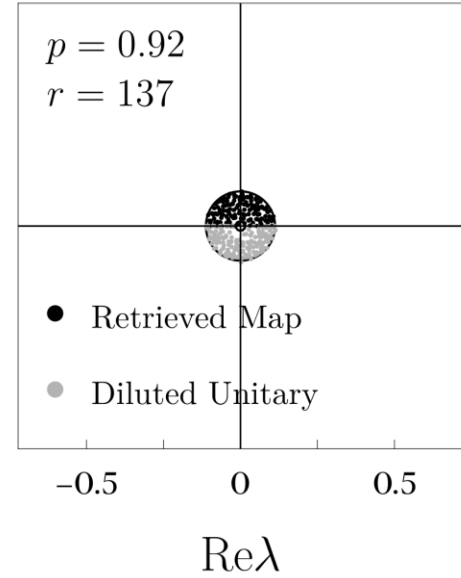
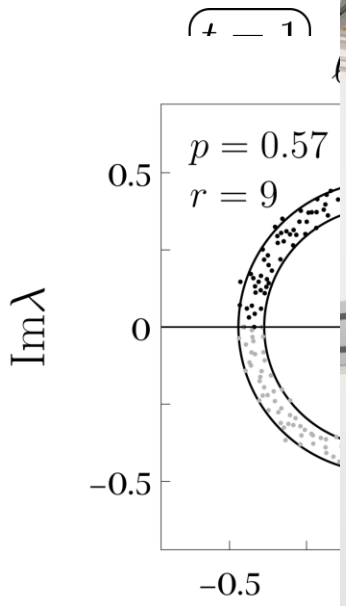
Experiments on Noisy Intermediate Scale Quantum (NISQ) computing platforms



Kristian Wold

OsloMet, PhD student

$$\ell = 40$$



$$C_{kl} = |A_{C,k}$$

$f_j^{(t)}$ map

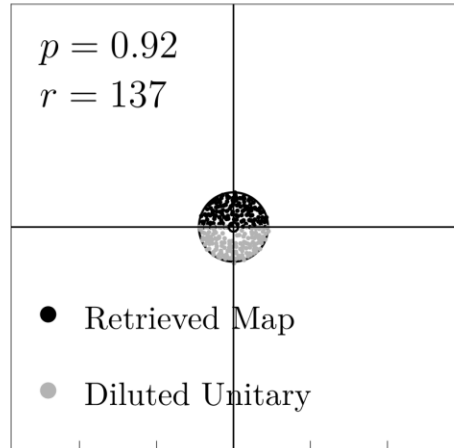
imization

Experiments on Noisy Intermediate Scale Quantum (NISQ) computing platforms



Kristian Wold
OsloMet, PhD student

$$\ell = 40$$



The screenshot shows a Jupyter Notebook with the following content:

```

def create_random_circuit(n, layers):
    """Creates a quantum circuit with n qubits and a given number of random layers."""
    qc = QuantumCircuit(n, n)
    for _ in range(layers):
        add_random_layer(qc, n)

    # Add measurements on all qubits
    qc.measure(range(n), range(n))

    return qc

# Example usage
n = 5 # Number of qubits
layers = 8 # Number of random Layers
circuit = create_random_circuit(n, layers)
print(circuit.draw())
    
```

Below the code, the output shows a quantum circuit diagram for 5 qubits (q0 to q4). The circuit consists of 8 layers of random gates, including single-qubit rotation gates (R_x, R_y, R_z) and two-qubit entangling gates (XX+YY). The diagram shows the sequence of operations and the resulting state of each qubit at each step.

$f_j^{(k)}$ map
[G]
imization

Experiments on Noisy Intermediate Scale Quantum (NISQ) computing platforms



Kristian Wold
OsloMet, PhD student

The screenshot displays a JupyterLab environment with the following components:

- File Explorer (Left):** Lists files such as 'ondemand', 'Chaos_aggre...', 'chaos2.ipynb', 'Helmi_nonint...', 'Helmi_Testi...', 'Mine_chaos...', 'Mine.ipynb', 'Untitled.ipynb', and 'XXZ.ipynb'.
- Code Editor (Top Left):** Contains Python code for creating a quantum circuit:


```
def create_random_circuit(n, layers):
    """Creates a quantum circuit with n qubits and a
    qc = QuantumCircuit(n, n)
    for _ in range(layers):
        add_random_layer(qc, n)

    # Add measurements on all qubits
    qc.measure(range(n), range(n))

    return qc

# Example usage
n = 5 # Number of qubits
layers = 8 # Number of random Layers
circuit = create_random_circuit(n, layers)
print(circuit.draw())
```
- Code Editor (Bottom Left):** Shows the execution of `circuit.draw("mpl")`, resulting in a quantum circuit diagram with 5 qubits (q0-q4) and 8 layers of operations including R_x , R_z , and $(XX+YY)$ gates.
- Code Editor (Top Right):** Shows the execution of `plot_histogram(counts1)`, displaying a histogram of measurement results for 32 different bit strings. The most frequent result is '00000' with a count of 67140.
- Code Editor (Bottom Right):** Shows another execution of `plot_histogram(counts1)`, displaying a zoomed-in histogram for a single bit string with a count of 10895.

Quantum Computing as a means to improve decision-making strategies of personalized cancer treatment

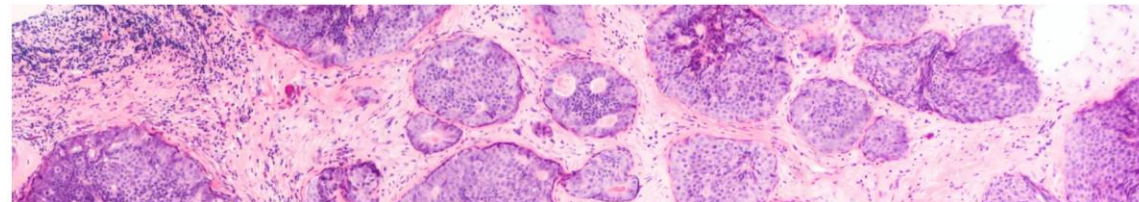


Heine Olsson Aabø
OsloMet, PhD student

Aftenposten

Viten | Forskning og vitenskap

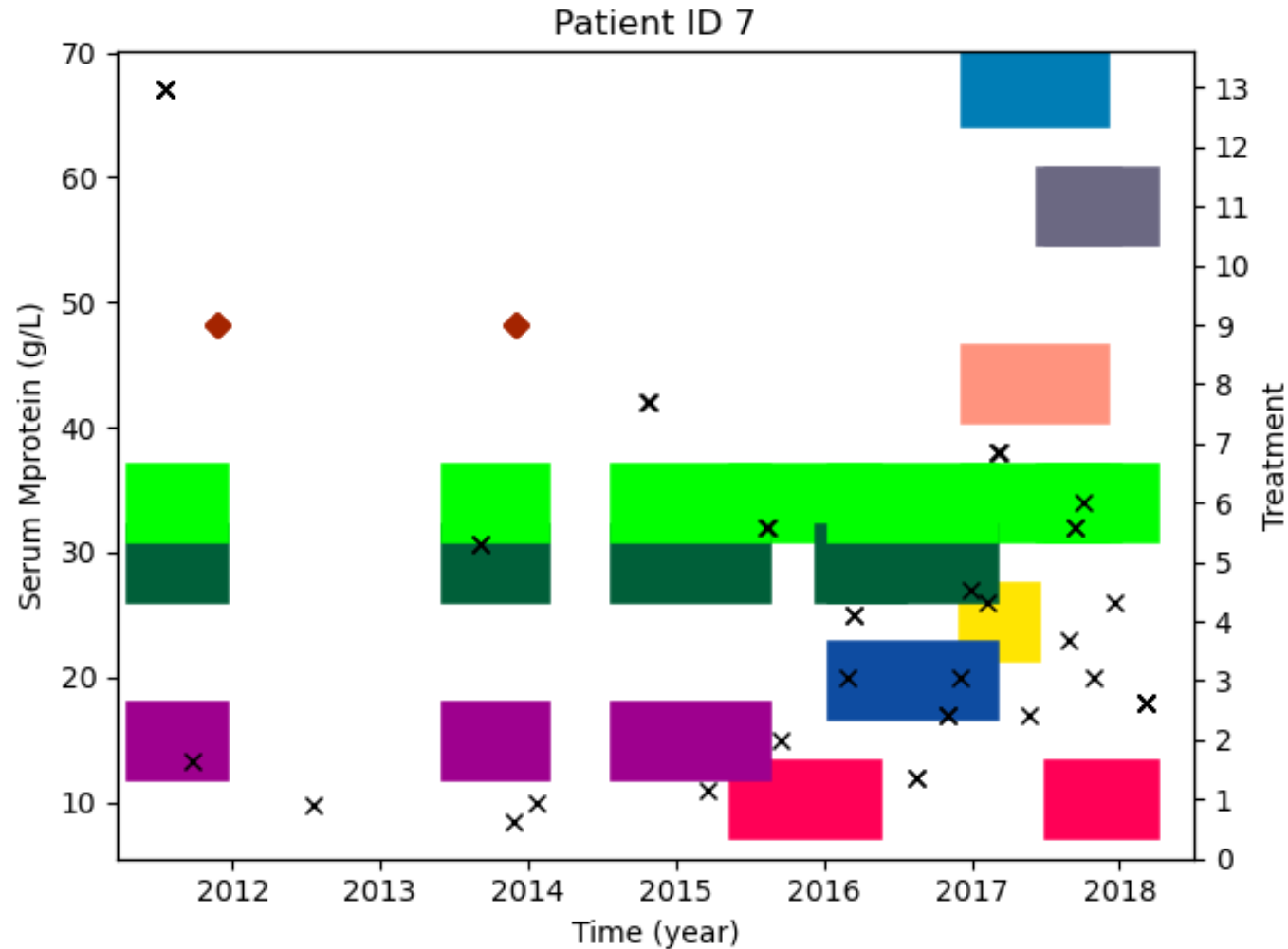
Paradokset er at vi sannsynligvis allerede har tilgang på effektive behandlinger for de fleste svulster, kanskje for alle, men å finne dem manuelt, er som å finne nålen i høystakken.



Quantum Computing as a means to improve decision-making strategies of personalized cancer treatment



Heine Olsson Aabø
OsloMet, PhD student



...e har tilgang på
...ster, kanskje for alle,
...e nålen i høystakken.

