DAQC HE Ansatz

RQC Hackathon team
presenter: Kuzmichev Artem

Main points

1. Qiskit Nature precalculation of Hamiltonian for analog block
2. Time parametrized sDAQC
3. Layerwise VQC training
4. Comparison

Qiskit Nature

We can use qiskit nature to precalculate required operator

```
+ 0.16892753870087907 * IZIZ
+ 0.045232799946057826 * YYYY
+ 0.045232799946057826 * XXYY
+ 0.045232799946057826 * YYXX
+ 0.045232799946057826 * XXXX
+ 0.1661454325638241 * ZIIZ
+ 0.1661454325638241 * IZZI
+ 0.17464343068300453 * ZIZI
+ 0.12091263261776627 * ZZII
```

```
qubit_converter = QubitConverter(mapper=ParityMapper(), two_qubit_reduction=Tr
qubit_op = qubit_converter.convert(
    second_q_op["ElectronicEnergy"], num_particles=es_problem.num_particles
)
print(qubit_op)
```

```
-1.0523732457728594 * II
+ 0.39793742484317784 * IZ
- 0.3979374248431793 * ZI
- 0.011280104256233686 * ZZ
+ 0.18093119978423117 * XX
```

[209]

```
H2_op = (-1.052373245772859 * I ^ I) + \
        (0.39793742484318045 * I ^ Z) + \
        (-0.39793742484318045 * Z ^ I) + \
        (-0.01128010425623538 * Z ^ Z) + \
        (0.18093119978423156 * X ^ X)

print(f'Number of qubits: {H2_op.num_qubits}')
```

```
Number of qubits: 2
```

Linear connectivity

Exploiting create_zz_hamiltonian for linear connectivity

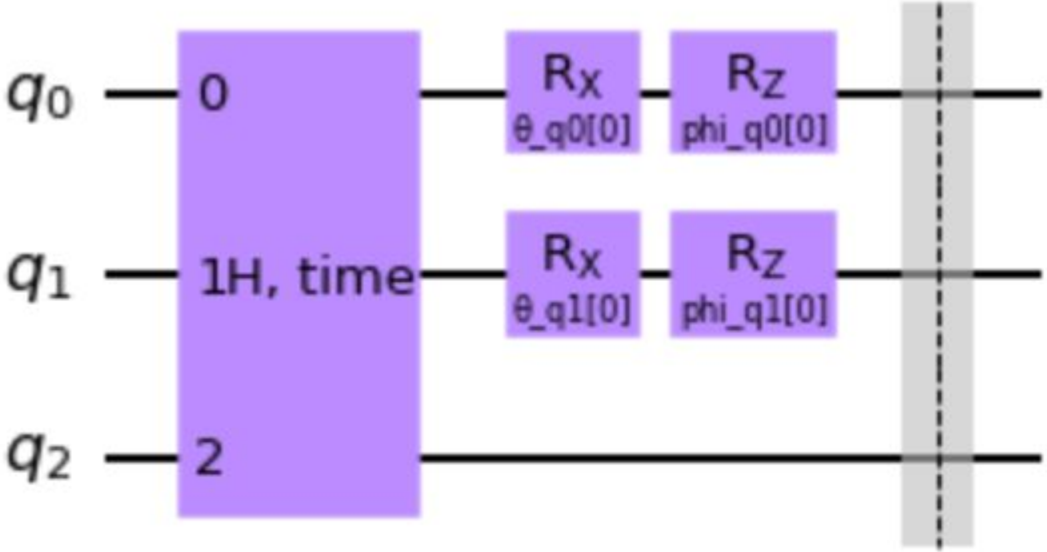
```python
num_qubits = 3
hamiltonian = create_zz_hamiltonian(num_qubits, [[0, 1], [1, 2]], [1, 2])
analog_block = HamiltonianGate(data=hamiltonian, time=3)

ham_matrix = HamiltonianGate(data=hamiltonian, time=2).params[0]
print(np.array_str(ham_matrix.real.astype('int'), precision=1, suppress_small=True))
```

```
[[ 3  0  0  0  0  0  0  0]
 [ 0 -1  0  0  0  0  0  0]
 [ 0  0 -3  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0]
 [ 0  0  0  0  1  0  0  0]
 [ 0  0  0  0  0 -3  0  0]
 [ 0  0  0  0  0  0 -1  0]
 [ 0  0  0  0  0  0  0  3]]
```

Ansatz

One layer with time parametrized for 3 qubits

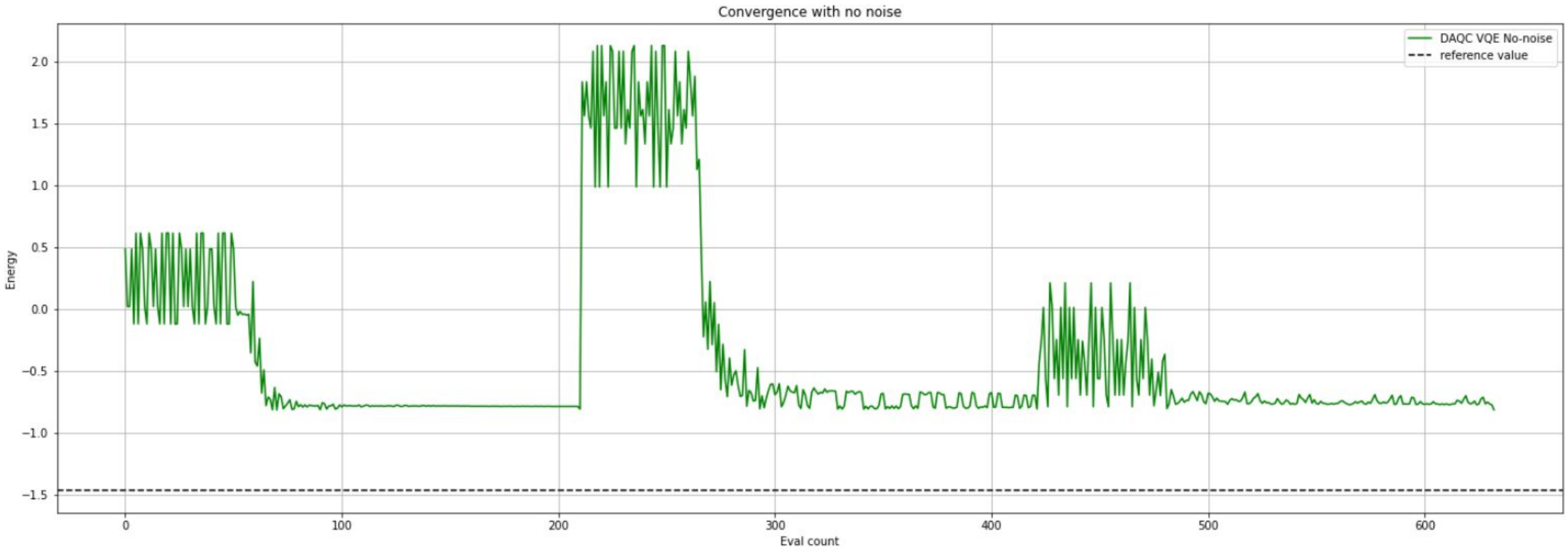


Layerwise trainining

Best parameters for previous layer

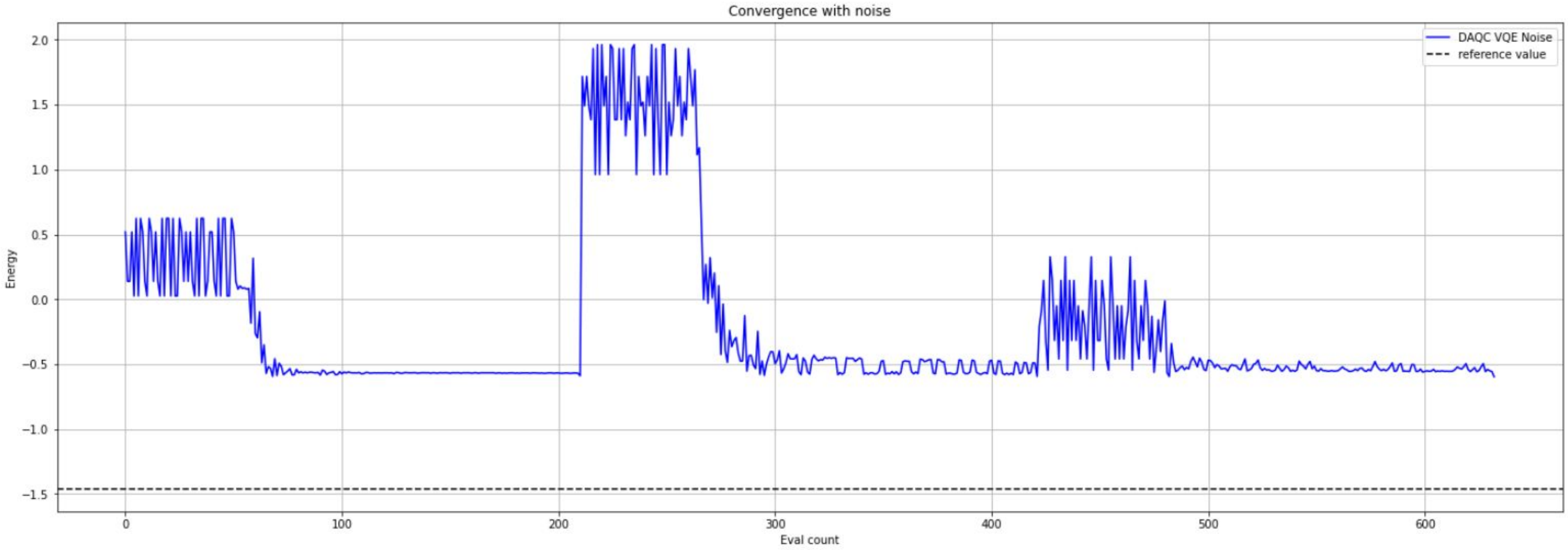
```
learning layer 0
         ┌───────────┐┌──────────────┐┌────────────────┐ ░ 
q_0: ┤0          ├┤ Rx(θ_q0[0]) ├┤ Rz(phi_q0[0]) ├─░─
         │  H, time  │└──────────────┘└────────────────┘ ░ 
q_1: ┤1          ├─────────────────────────────────░─
         └───────────┘                                   ░ 
{ParameterVectorElement(phi_q0[0]): 1.246711733858144, ParameterVectorElement
VQE result for layer 0: -1.79087
learning layer 1
         ┌───────────┐┌─────────────────────────┐┌───────────────────────┐ ░  »
q_0: ┤0          ├┤ Rx(-3.22856080539681) ├┤ Rz(1.24671173385814) ├─░──»
         │  H, time  │└─────────────────────────┘└───────────────────────┘ ░  »
q_1: ┤1          ├──────────────────────────────────────────────────░──»
         └───────────┘                                                    ░  »
«         ┌───────────┐┌──────────────┐┌────────────────┐ ░ 
«q_0: ┤0          ├┤ Rx(θ_q0[0]) ├┤ Rz(phi_q0[0]) ├─░─
«         │  H, time  │└──────────────┘└────────────────┘ ░ 
«q_1: ┤1          ├─────────────────────────────────░─
«         └───────────┘                                   ░ 
{ParameterVectorElement(phi_q0[0]): 3.1379865201803487, ParameterVectorElemer
VQE result for layer 1: -1.79158
```

Results



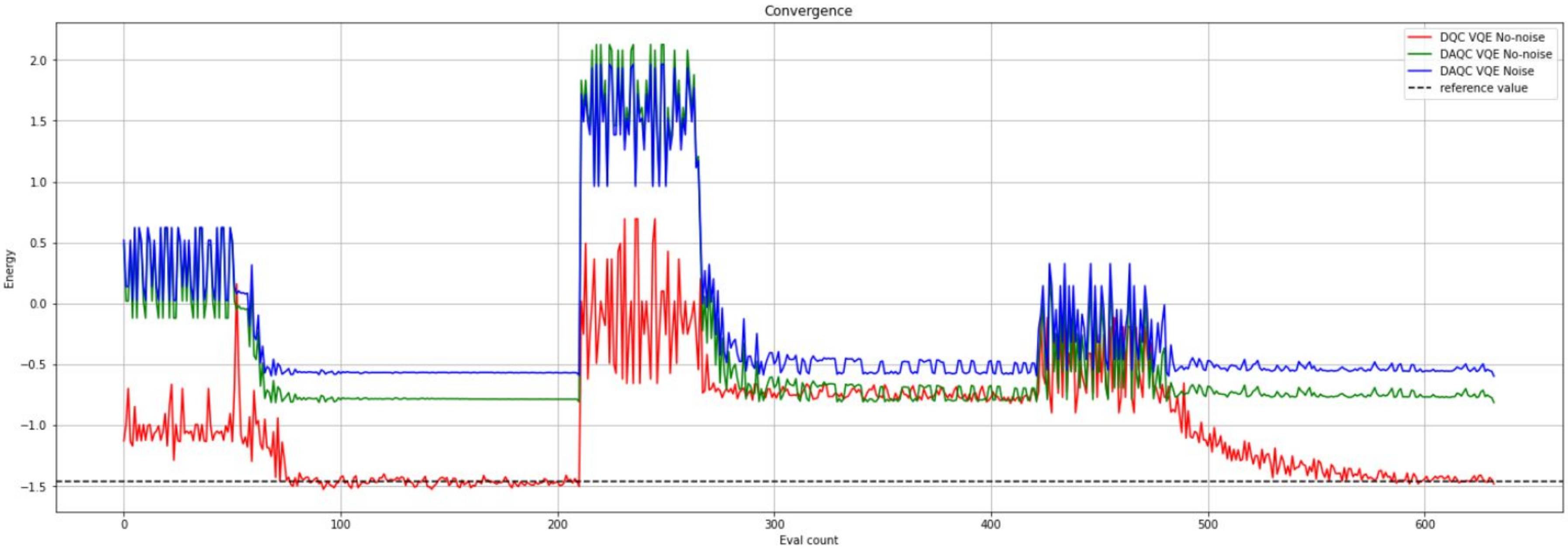
Convergence with no noise
DAQC VQE No-noise
reference value
Energy
Eval count
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
100
200
300
400
500
600

Results



Convergence with noise
DAQC VQE Noise
reference value
Energy
Eval count
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
100
200
300
400
500
600

Results



Convergence
DQC VQE No-noise
DAQC VQE No-noise
DAQC VQE Noise
reference value
Energy
Eval count
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
100
200
300
400
500
600

References

- Paper "Digital-Analog Quantum Computation" by Adrian Parra-Rodriguez, Pavel Lougovski, Lucas Lamata, Enrique Solano, and Mikel Sanz: https://arxiv.org/abs/1812.03637
- Paper "Approximating the Quantum Approximate Optimisation Algorithm" by David Headley, Thorge Müller, Ana Martin, Enrique Solano, Mikel Sanz, and Frank K. Wilhelm: https://arxiv.org/abs/2002.12215
- Overview article about DAQC: https://arxiv.org/abs/2101.0844
- Qiskit example: https://qiskit.org/textbook/ch-applications/vqe-molecules.html
- Superconducting Circuit Architecture for Digital-Analog Quantum Computing J. Yu https://arxiv.org/abs/2103.15696