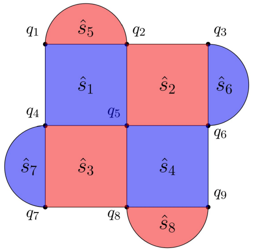
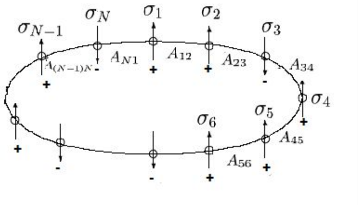
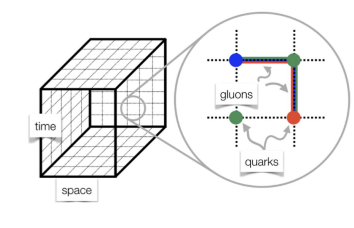
2025 YQuantum QuEra Challenge

QuBruin: John Ye, Mu Niu, Victor Yu, Harry Wang, and Hanyu Wang

Overview



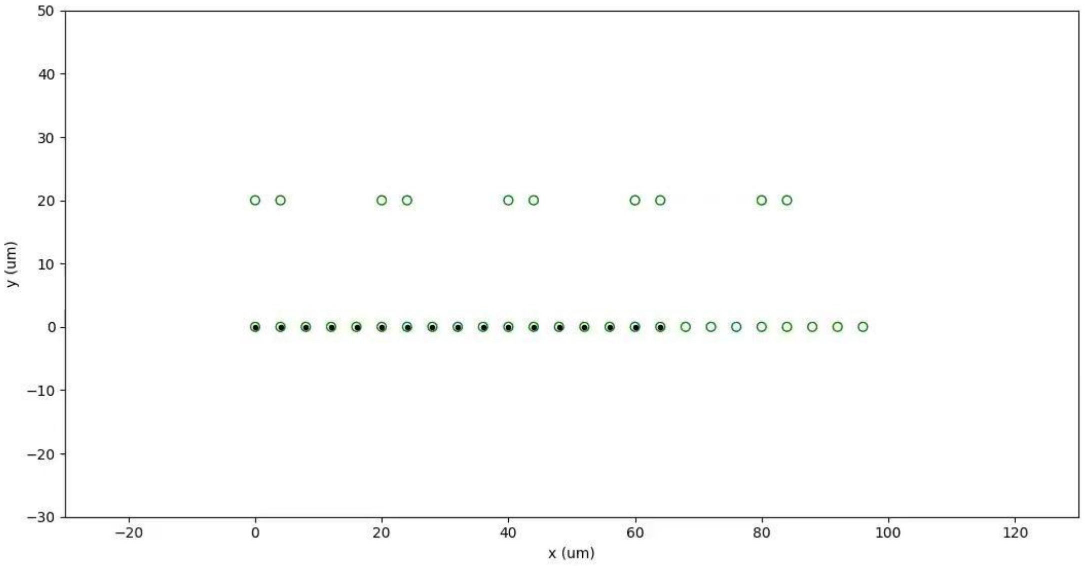




Bloqade

Kirin

Video Demo



New features

```python
@qasm2.extended
def repetition_code():

    qreg = qasm2.qreg(5)
    creg = qasm2.creg(5)
    '''
    Srror propagation and syndrome extraction
    '''
    qasm2.cx(qreg[0], qreg[3])
    qasm2.cx(qreg[1], qreg[3])
    qasm2.cx(qreg[1], qreg[4])
    qasm2.cx(qreg[2], qreg[4])

    qasm2.measure(qreg[0], creg[0])
    qasm2.measure(qreg[1], creg[1])
    qasm2.measure(qreg[2], creg[2])
    qasm2.measure(qreg[3], creg[3])
    qasm2.measure(qreg[4], creg[4])
    '''
    Decoding and error correction
    '''
    if creg[3] == 0 and creg[4] == 1:
        qasm2.x(qreg[2])
    if creg[3] == 1 and creg[4] == 1:
        qasm2.x(qreg[1])
    if creg[3] == 1 and creg[4] == 0:
        qasm2.x(qreg[0])

    return creg
```

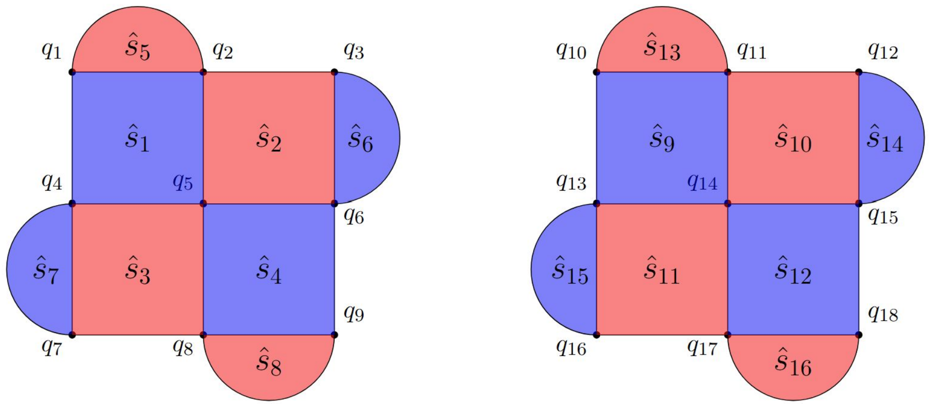
```python
@qasm2.extended
def add_X_syndrome_circuit(qreg: qasm2.QReg,creg:qasm2.CReg,ndataqubits:int,stabindex:int,index_tuple:tuple[int, ...]):
    qasm2.h(qreg[ndataqubits+stabindex])
    for i in range(len(index_tuple)):
        qasm2.cx(qreg[ndataqubits+stabindex],qreg[index_tuple[i]])
    qasm2.h(qreg[ndataqubits+stabindex])
    qasm2.measure(qreg[ndataqubits+stabindex],creg[stabindex])


@qasm2.extended
def add_Z_syndrome_circuit(qreg: qasm2.QReg,creg:qasm2.CReg,ndataqubits:int,stabindex:int,index_tuple:tuple[int, ...]):
    for i in range(len(index_tuple)):
        qasm2.cx(qreg[index_tuple[i]],qreg[ndataqubits+stabindex])
    qasm2.measure(qreg[ndataqubits+stabindex],creg[stabindex])
```

```python
suf=surfaceCode(3)
suf.draw_surface()
suf.print_stab()
```

```
Q1----Q2----Q3
|     |     |
|     |     |
Q4----Q5----Q6
|     |     |
|     |     |
Q7----Q8----Q9
X1-X2-X4-X5
X4-X5-X7-X8
X2-X3
X8-X9
Z2-Z3-Z5-Z6
Z5-Z6-Z8-Z9
Z1-Z4
Z3-Z6
```

```python
@qasm2.extended
def surface_code_d3_circuit():
    qreg = qasm2.qreg(2*3**2-1)
    creg = qasm2.creg(3**2-1)
    add_X_syndrome_circuit(qreg,creg,9,0,(0,1,3,4))
    add_X_syndrome_circuit(qreg,creg,9,1,(3,4,6,7))

    add_X_syndrome_circuit(qreg,creg,9,2,(1,2))
    add_X_syndrome_circuit(qreg,creg,9,3,(7,8))

    add_Z_syndrome_circuit(qreg,creg,9,4,(1,2,4,5))
    add_Z_syndrome_circuit(qreg,creg,9,5,(4,5,7,8))


    add_Z_syndrome_circuit(qreg,creg,9,6,(0,3))
    add_Z_syndrome_circuit(qreg,creg,9,7,(2,5))
```

Simple Logical algorithm



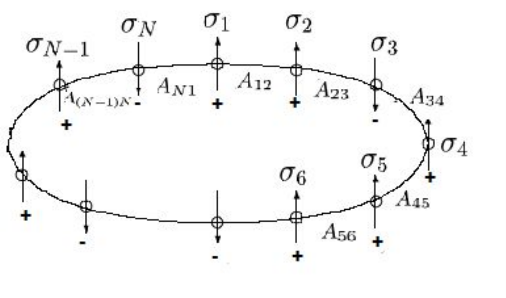
Recursive implementation of fault-tolerant Syn

```python
@qasm2.extended
def add_X_syndrome_circuit_recursive(result:int,round:int,qreg: qasm2.QReg,creg:qasm2.CReg,ndataqubits:int,stabindex:int,index_tuple:tuple[int, ...]):
    if round==3:
        return

    qasm2.h(qreg[ndataqubits+stabindex])
    for i in range(len(index_tuple)):
        qasm2.cx(qreg[ndataqubits+stabindex],qreg[index_tuple[i]])
    qasm2.h(qreg[ndataqubits+stabindex])
    qasm2.measure(qreg[ndataqubits+stabindex],creg[stabindex])


    if creg[stabindex]==result:
        return add_X_syndrome_circuit_recursive(result,round+1,qreg,creg,ndataqubits,stabindex,index_tuple)

    qasm2.reset(qreg[ndataqubits+stabindex])
    return add_X_syndrome_circuit_recursive(1-result,1,qreg,creg,ndataqubits,stabindex,index_tuple)
```

The 1D Ising Model and Trotterization

$$H = -J\sum_{i=0}^{L-1} Z_i Z_{i+1} - h\sum_{i=0}^{L-1} X_i.$$

Ising model Hamiltonian



Visualization of a spin chain with periodic boundary conditions

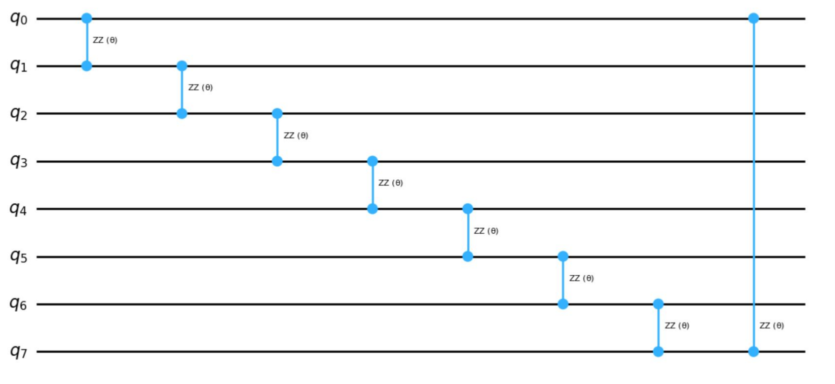
Trotterizations

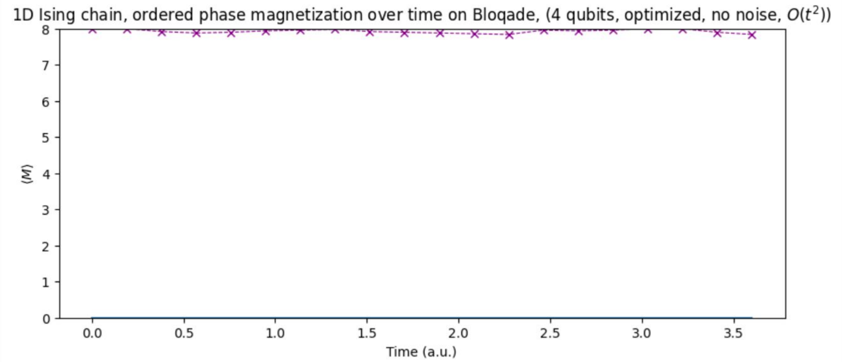
$$e^{(A+B)\Delta t} = e^{A\Delta t}e^{B\Delta t} + O(\Delta t)^2.$$

$$e^{(A+B)\Delta t} = e^{A\Delta t/2}e^{B\Delta t}e^{A\Delta t/2} + O(\Delta t)^3.$$

Image sources:
https://www.researchgate.net/profile/Petrus-Anjos/publication/353284171/figure/fig6/AS:1046021979242508@1626402482421/The-1D-Ising-model-with-periodic-boundary-condition.ppm

Time Independent Evolution



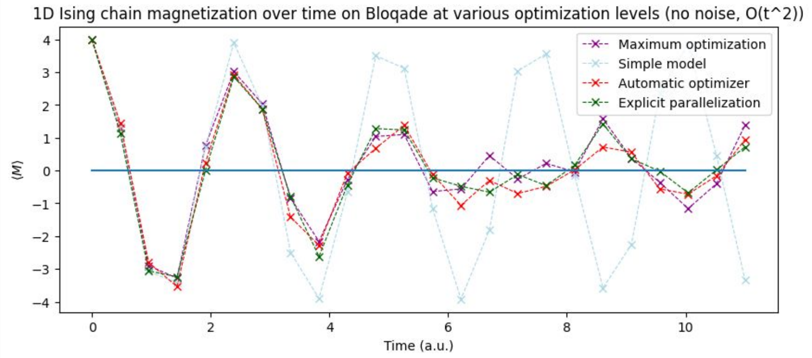


Naive implementation of Trotterization:

```
for i in range(n_qubits):
        qasm2.cx(qreg[i], qreg[(i+1)%n_qubits])
        qasm2.rz(qreg[(i+1)%n_qubits], 2*J*timestep)
        qasm2.cx(qreg[i], qreg[(i+1)%n_qubits])
for i in range(n_qubits):
        qasm2.rx(qreg[i], 2*h*timestep)
```

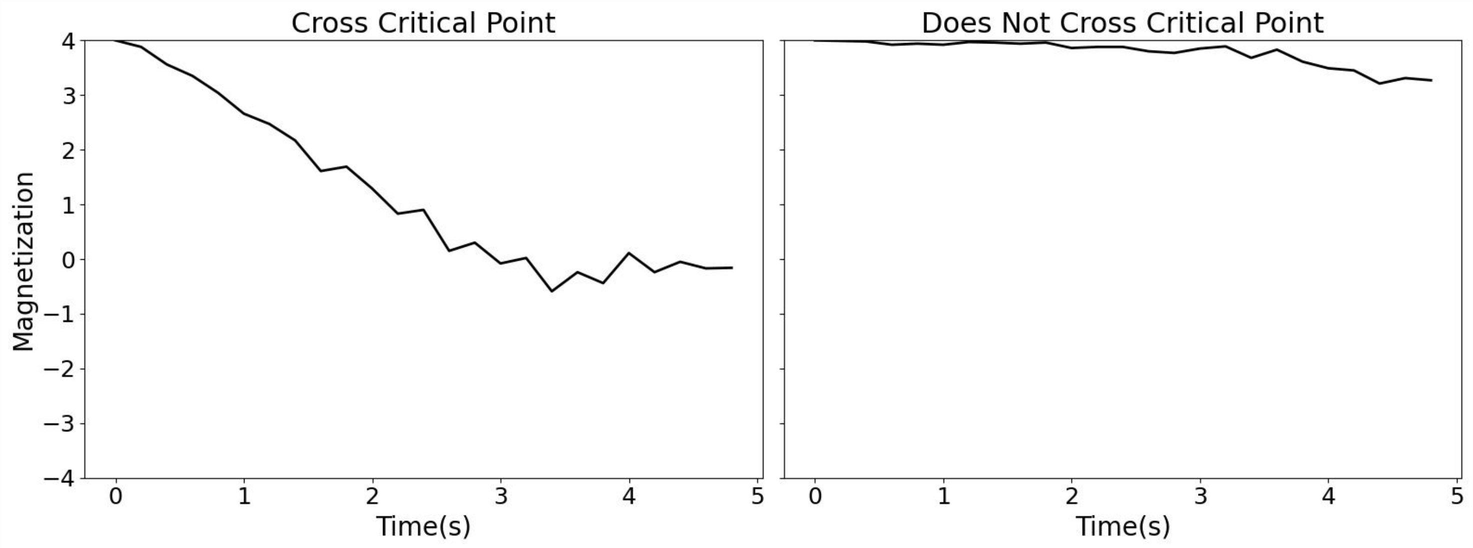
To improve performance:

- Parallelize the circuit
- Use the optimizer pass

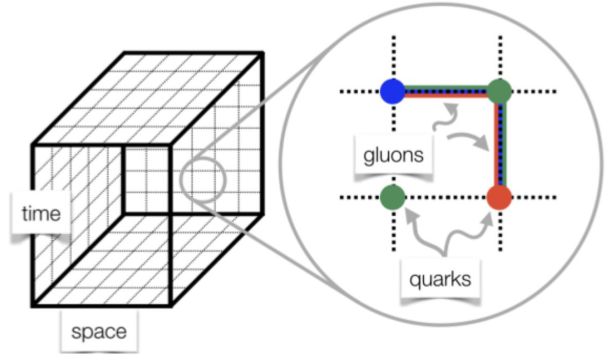


Time Dependent Hamiltonian Evolution

- We have detected two phases in time independent evolution
- Tune h and J throughout time steps show state volution when cross / not cross the critical point (h=J)



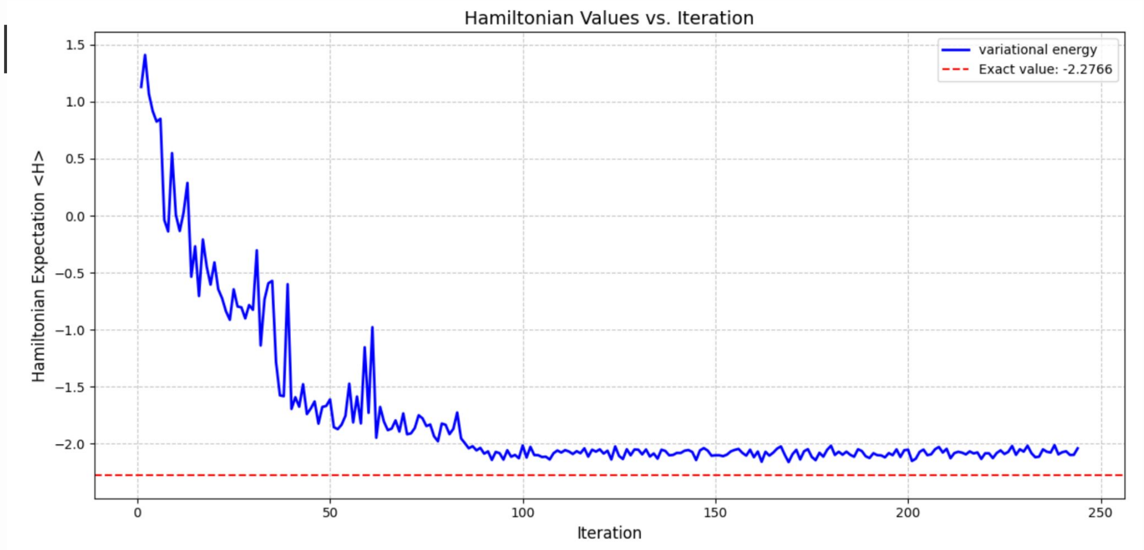
The 1+1D Lattice Schwinger Model



$$H_{\text{Schwinger}} = \frac{g^2 a}{2} \sum_{n=1}^{N-1} \left[\frac{1}{2} \sum_{k=1}^{n} (\sigma_k^z + (-1)^k) \right]^2 + \frac{m}{2} \sum_{n=1}^{N} (-1)^n \sigma_n^z + \frac{1}{2a} \sum_{n=1}^{N-1} (\sigma_n^+ \sigma_{n+1}^- + h.c.)$$

Image sources:
https://www.researchgate.net/profile/Petrus-Anjos/publication/353284171/figure/fig6/AS:1046021979242508@1626402482421/The-1D-Ising-model-with-periodic-boundary-condition.ppm

Variational Method for Ground State Pre



N=4,
a=m=g=1

Pros:

1. Connectivity for entangling layer
2. Parallelization for rotation block: Native Uz Uxy
3. Scalability for multi-qubits

Cons:

1. Measurements against Hamiltonian
2. Circuit compile complexity in multi-layers
3. Two qubit gate fidelity

Image sources:
https://www.researchgate.net/profile/Petrus-Anjos/publication/353284171/figure/fig6/AS:1046021979242508@1626402482421/The-1D-Ising-model-with-periodic-boundary-condition.ppm

Table 2. Qualitative comparison between different circuit rewriting flows

Circuit	**Cost**
Original Ising Model	268.80
Original Ising Model + Qiskit	76.80
Original Ising Model + Qiskit + UOpToParallel	72.10
Original Ising Model + Qiskit + ZX	120.00
Original Ising Model + Qiskit + ZX + UOpToParallel	65.40

Table 1. Cost metric for the qualitative comparison

Gate Type	**Cost Weight**
Local 1-qubit	0.2
Local 2-qubit	0.4
Global 1-qubit	0.1
Global 2-qubit	0.4

Thank you!

Q&A