



# Mitigating Noise in Quantum Hardware #2

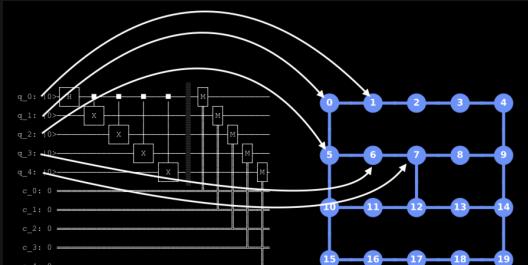
Nick Brønn, Ph.D.

Transpilation: take circuit to “smarter” circuit

### Qubit Mapping

(be smart about the qubits you use)

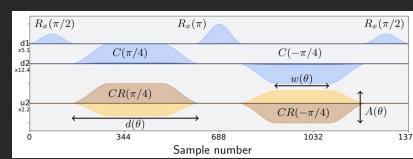
- Routing
- Layout



### Error Suppression

(use physics to reduce errors)

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measure Error Mitigation

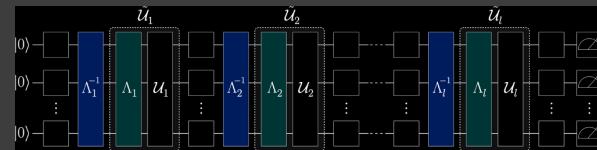


Transpile differently + Analyze

### Error Mitigation

(analyze circuit run in multiple ways)

- Zero Noise Extrapolation
- Probabilistic Error Cancellation



# Installation



## qiskit-partners

```
>>> pip install mapomatic
```

- Layout

```
>>> pip install mthree
```

- Matrix-free Measurement  
(Error) Mitigation

## qiskit-research

```
>>> git clone  
https://github.com/qiskit-  
research/qiskit-research
```

```
>>> cd qiskit-research
```

```
>>> pip install -e .
```

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling

## qiskit-ibm-runtime

```
>>> pip install qiskit-ibm-  
runtime
```

- Twirled Readout Error  
eXtinction
- Zero Noise Extrapolation
- Probabilistic Error  
Cancellation

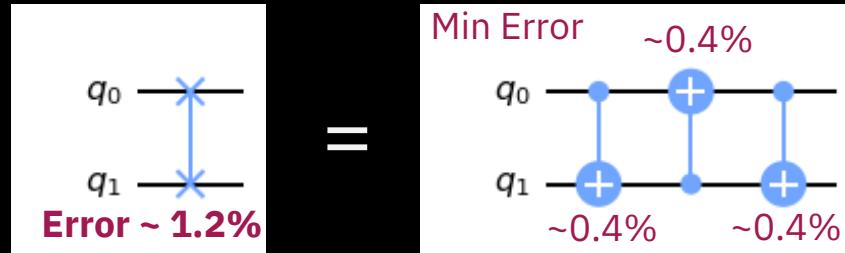
# Qubit Mapping



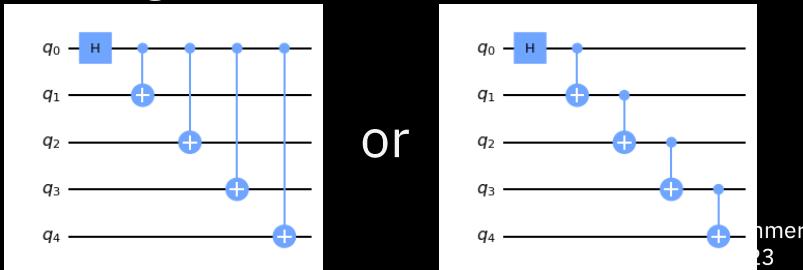
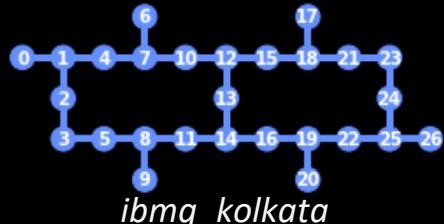
- **Routing**

Limited connectivity → add SWAP gates to **route** quantum information

- **Layout**



NP hard problem → employ stochastic algorithm (Sabre: [arXiv: 1809.02573])

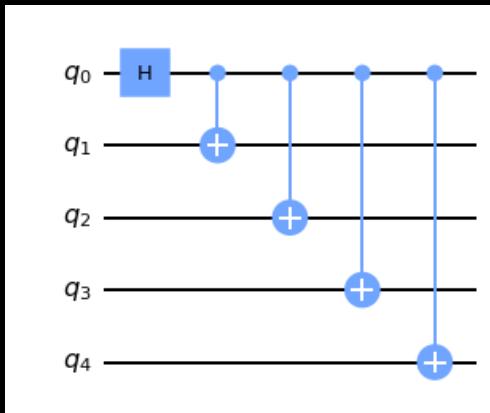


# Qubit Mapping



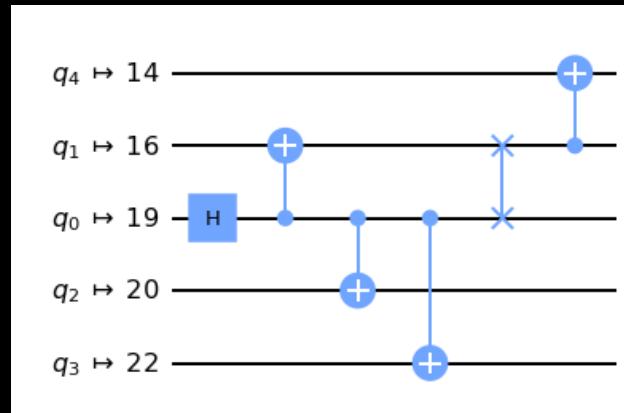
- **Routing**

```
from qiskit.circuit import QuantumCircuit  
qc = QuantumCircuit(5)  
qc.h(0)  
qc.cx(0, 1)  
qc.cx(0, 2)  
qc.cx(0, 3)  
qc.cx(0, 4)  
qc.draw('mpl')
```



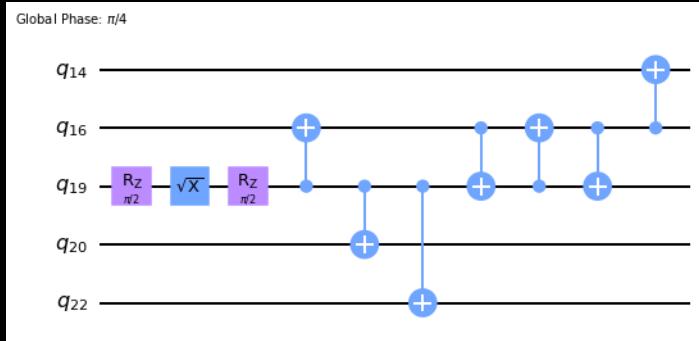
- **Layout**

```
from qiskit.transpiler.preset_passmanagers  
      import generate_preset_pass_manager  
from qiskit_ibm_provider import IBMProvider  
provider = IBMProvider()  
backend = provider.get_backend('ibmq_kolkata')  
pm = generate_preset_pass_manager(2, backend)  
qc_l = pm.layout.run(qc)  
qc_l.draw('mpl', idle_wires=False)
```



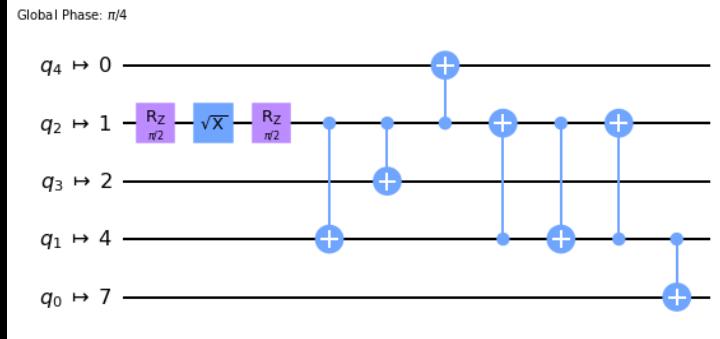
# Qubit Mapping

- Routing



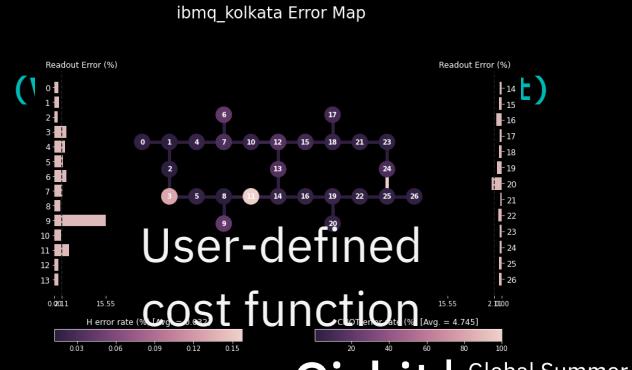
- Layout

```
from qiskit import transpile  
qc_m = transpile(qc_d, backend,  
                  initial_layout=init_layout)  
qc_m.draw('mpl', idle_wires=False)
```



```
from mapomatic import deflate_circuit,  
                    evaluate_layouts, matching_layouts  
qc_d = deflate_circuit(qc_1)  
layouts = matching_layouts(qc_d,  
                           backend.configuration().coupling_map)  
scored_layouts = evaluate_layouts(  
    qc_d, layouts, backend  
)  
init_layout = scored_layouts[0][0]  
print(f'The optimal layout on {backend.name}  
      is {init_layout}')
```

The optimal layout on ibmq\_kolkata is [7, 4, 1, 2, 0]

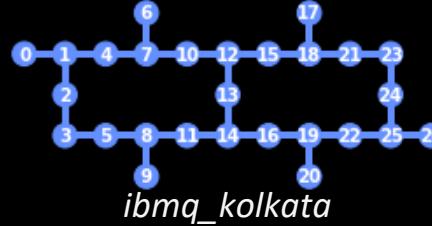
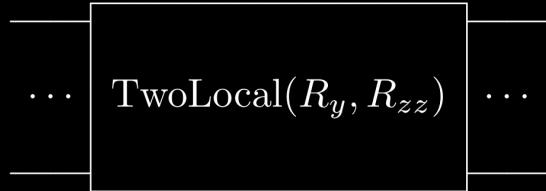


# Error Suppression

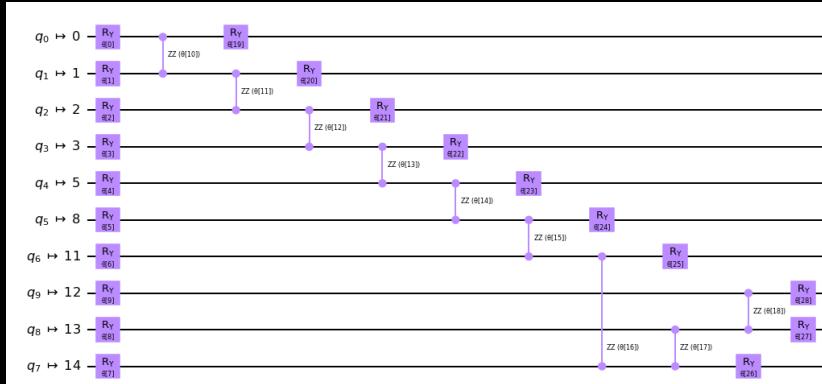
## Ansatz Design



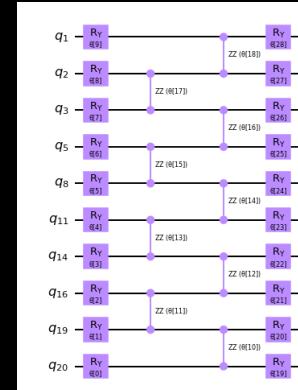
- **Parallelization**
- **Pulse Scaling**
- **Dynamical Decoupling**
- **Pauli Twirling**
- **Measurement Error Mitigation**



Linear Entanglement → No added SWAPs



Staircase Structure

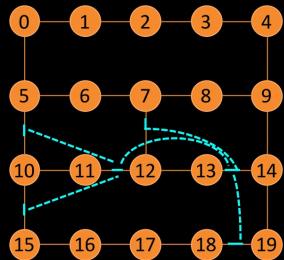
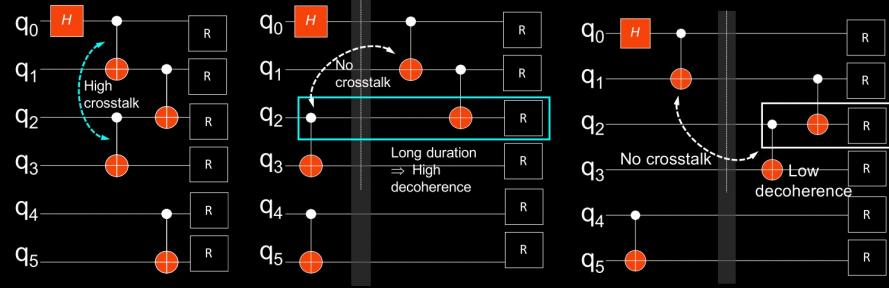


Brickwork Structure  
Qiskit Global Summer School 2023

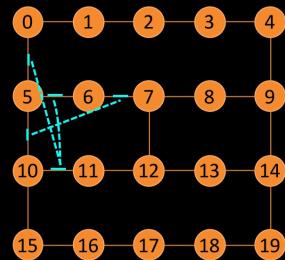
# Crosstalk Effect on Algorithms



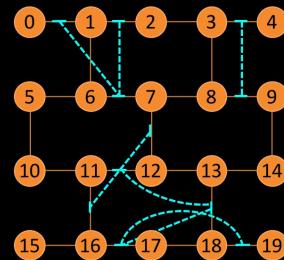
- Crosstalk characterization comparing independent and simultaneous randomized benchmarking
- Depending on algorithm, may be harmful or may have no effect



(a) IBMQ Poughkeepsie



(b) IBMQ Johannesburg

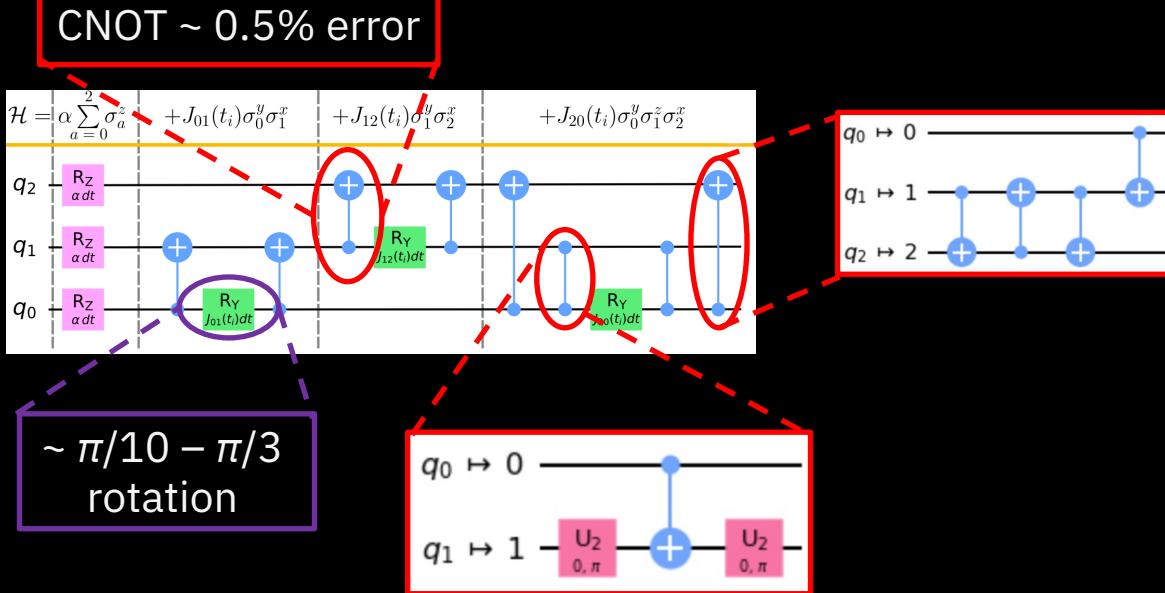


(c) IBMQ Boeblingen

# Error Suppression



- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation

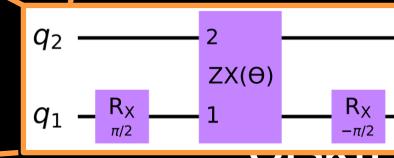
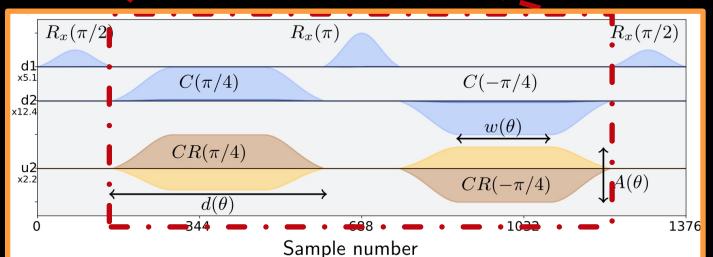
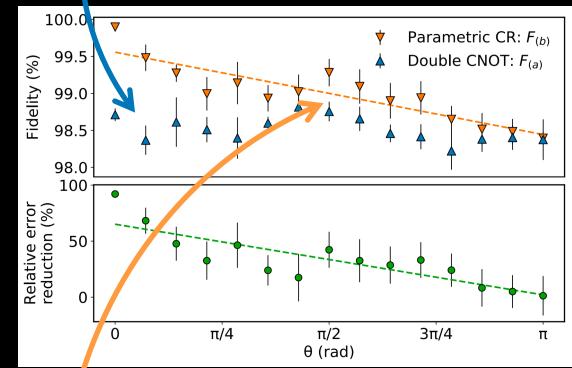
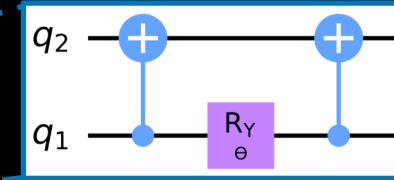
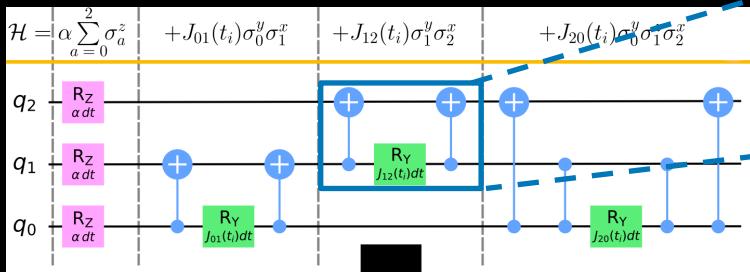


# Error Suppression

`qc_rzx = scale_cr_pulses(qc, backend, param_bind=param_bind)`



- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation

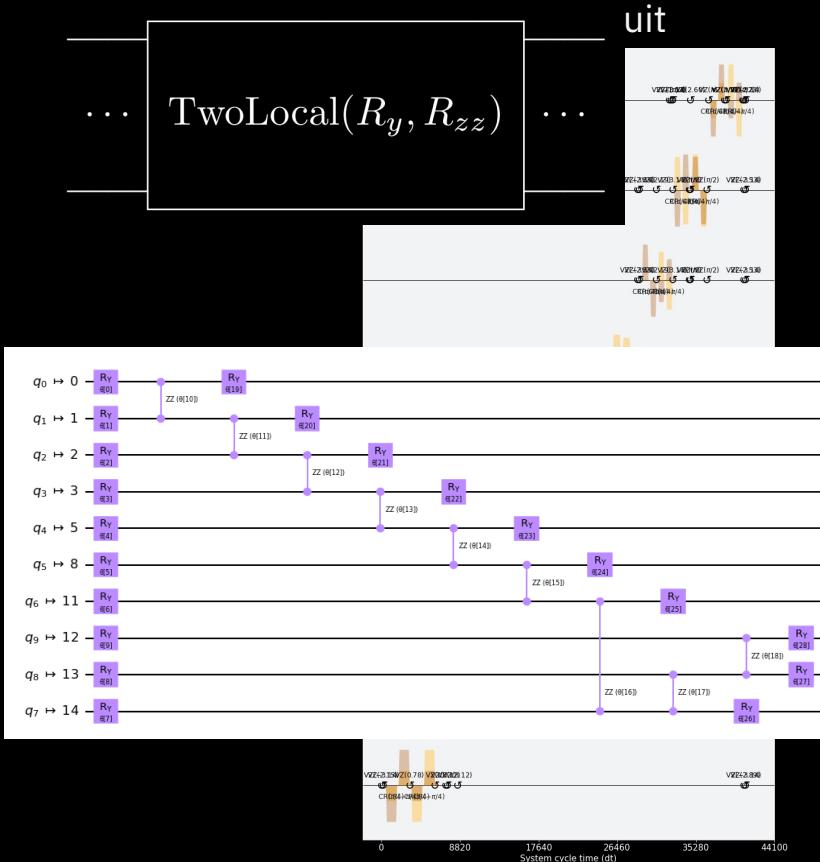


Global Summer School 2023

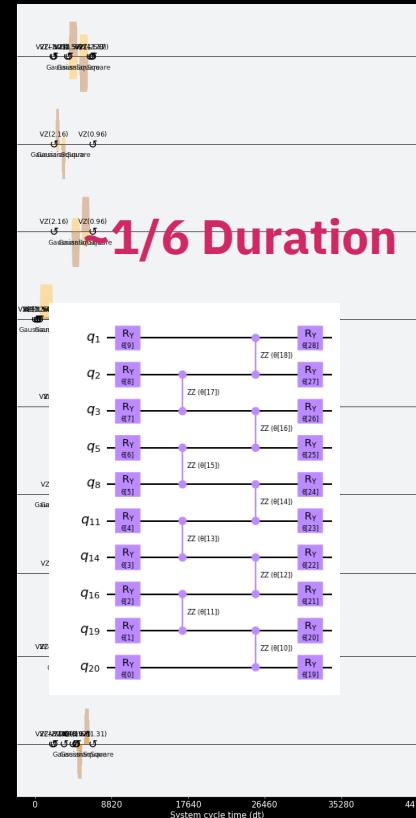


# Error Suppression: Example

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation

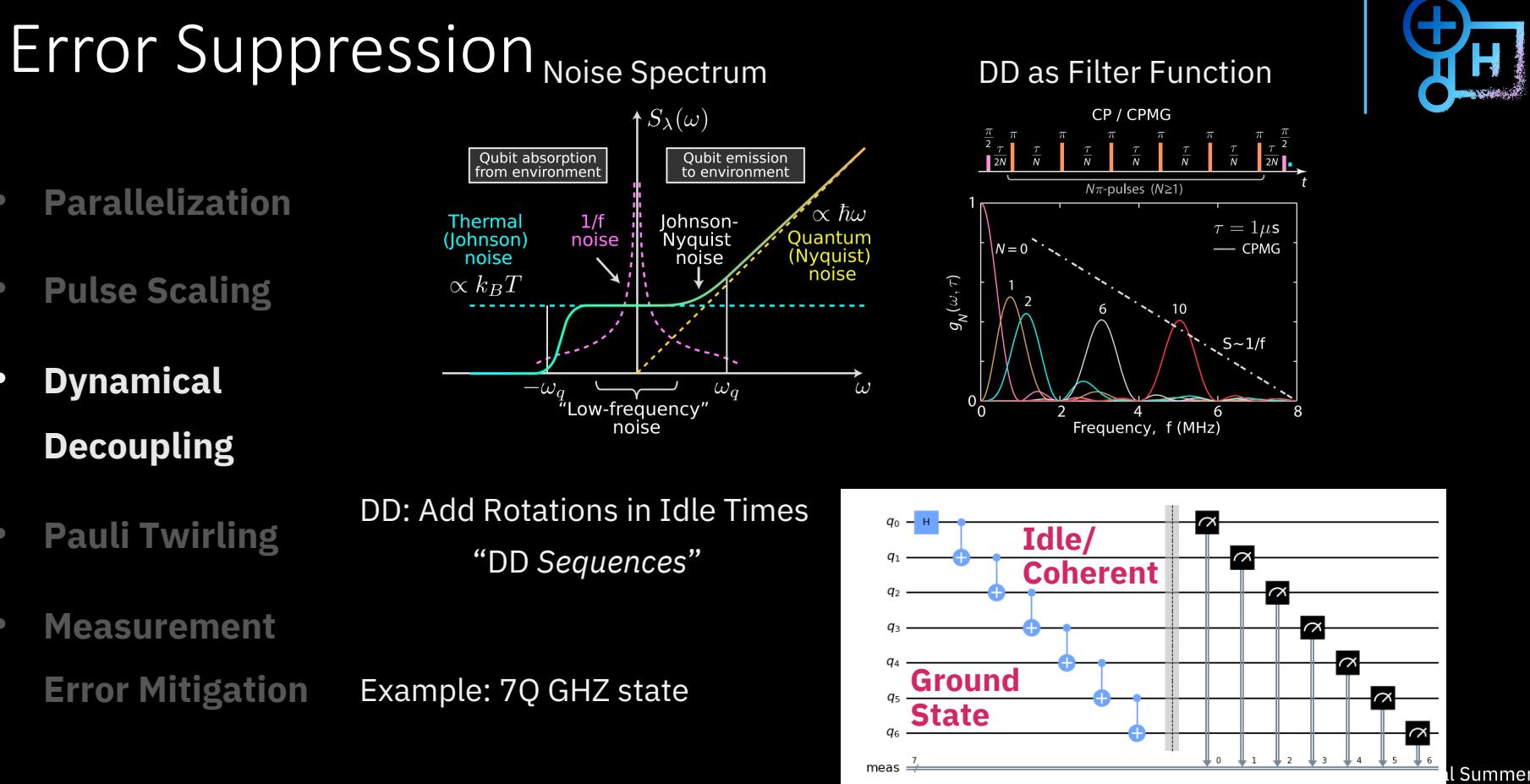


Bricked & Pulse Scaled



# Error Suppression

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement
- Error Mitigation



Example: 7Q GHZ state

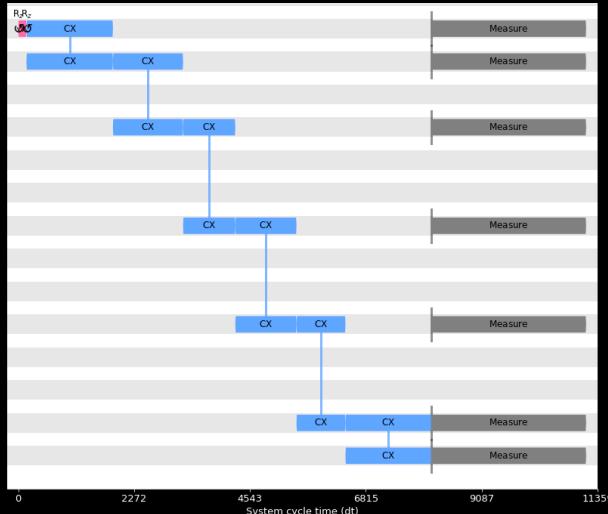




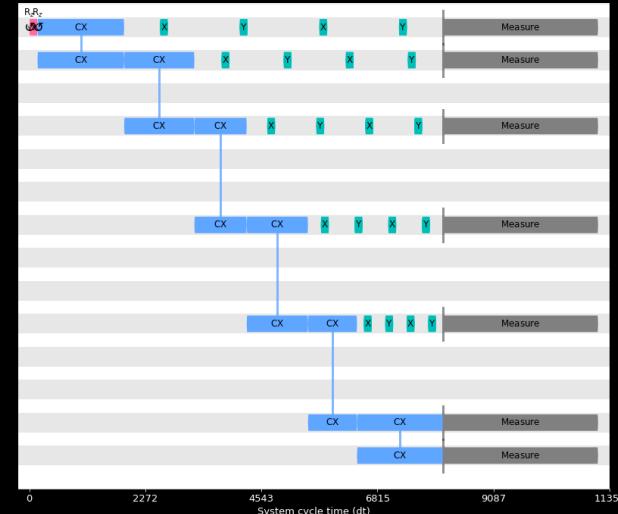
# Error Suppression: Example

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation

As-is Circuit



XY4 Dynamical Decoupling

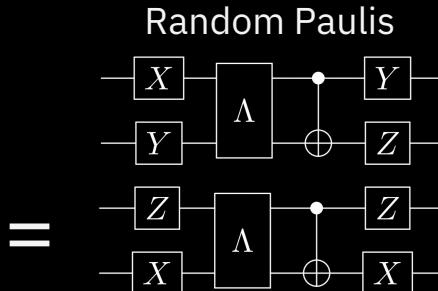
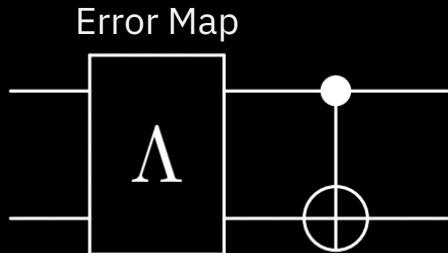


```
add_dynamical_decoupling(circs, backend, "XY4", add_pulse_cals=True)
```

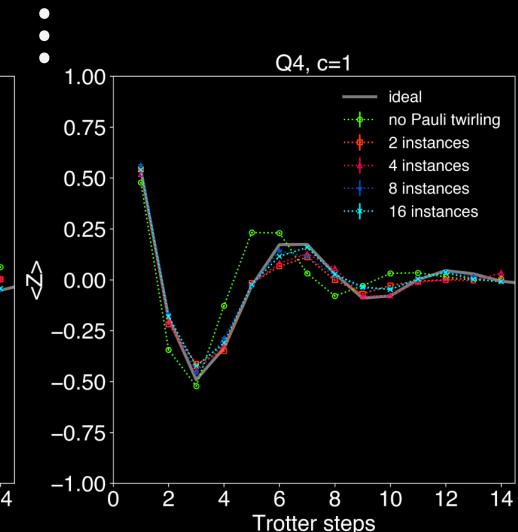
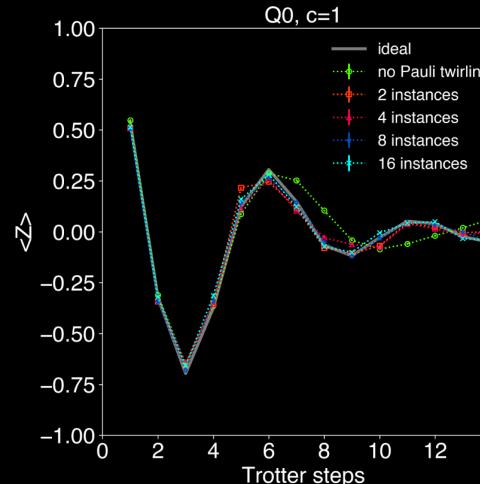
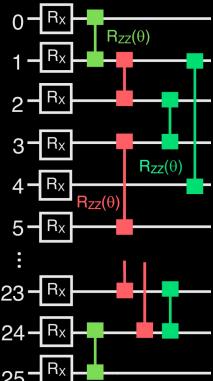
Qiskit Runtime: `options.optimization_level = 3 # also 1, 2`  
`options.resilience_level = 0`

# Error Suppression

- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation



Trotter Step

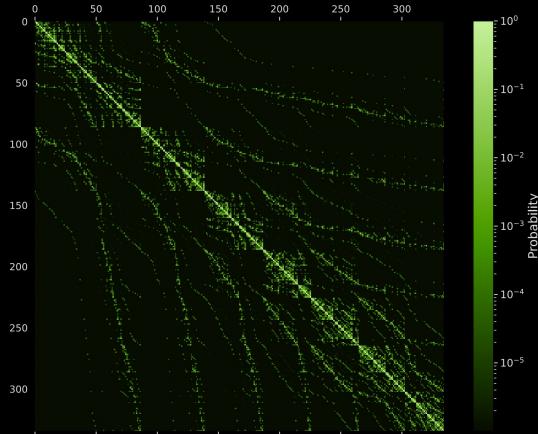
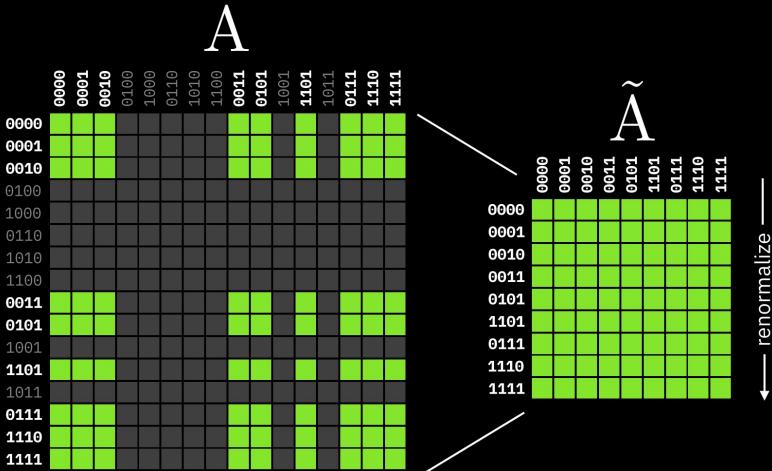


```
twirled_circs = add_pauli_twirls(circs, num_twirled_circuits=num_twirls,
                                  seed=12345, transpile_added_paulis=True)
```

# Error Suppression



- Parallelization
- Pulse Scaling
- Dynamical Decoupling
- Pauli Twirling
- Measurement Error Mitigation



```
from mthree import M3Mitigation  
  
mit = M3Mitigation(backend)  
mit.cals_from_system(init_layout)  
quasi_probs = mit.apply_correction(job.result().get_counts(), my_layout)
```

Qiskit Runtime: `options.resilience_level = 1`

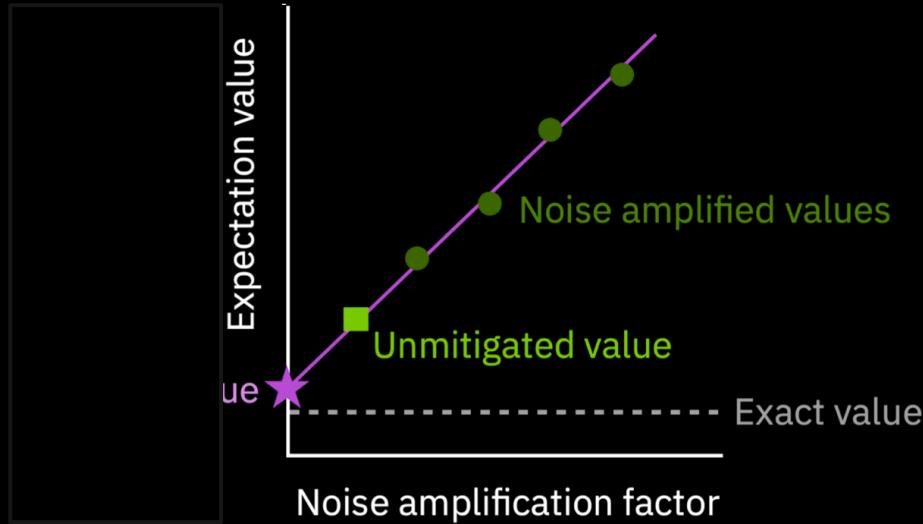
[Paul Nation *et al*, PRX (2021)]

[Ewout van den Berg *et al*, PRA (2022)]

# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**

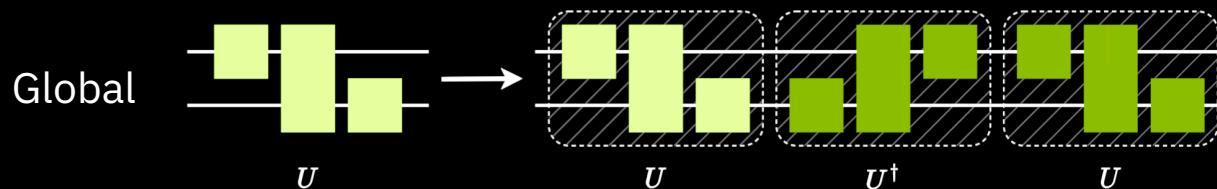


# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**

## Digital Folding





# Error Mitigation

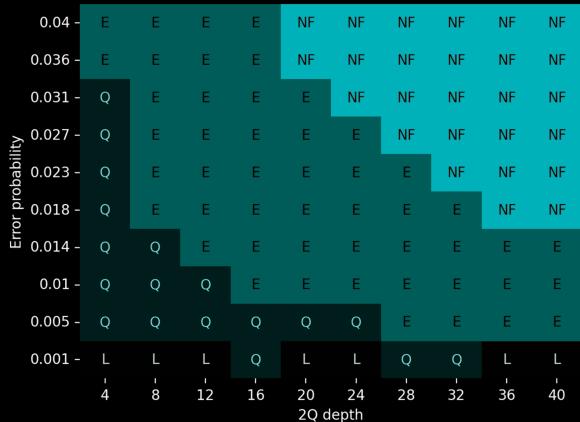
- **Zero Noise**

## Extrapolation

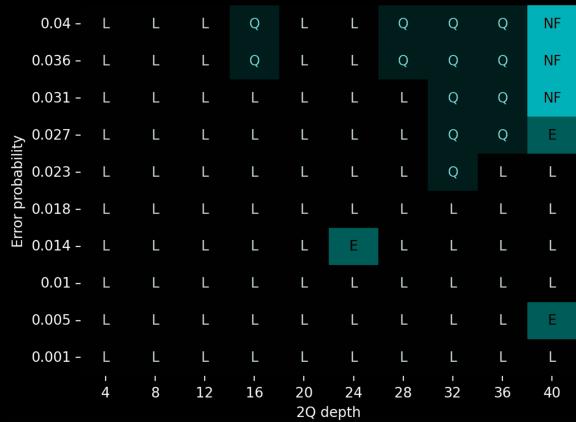
- **Probabilistic Error**

## Cancellation

Noise Factors: 1, 3, 5



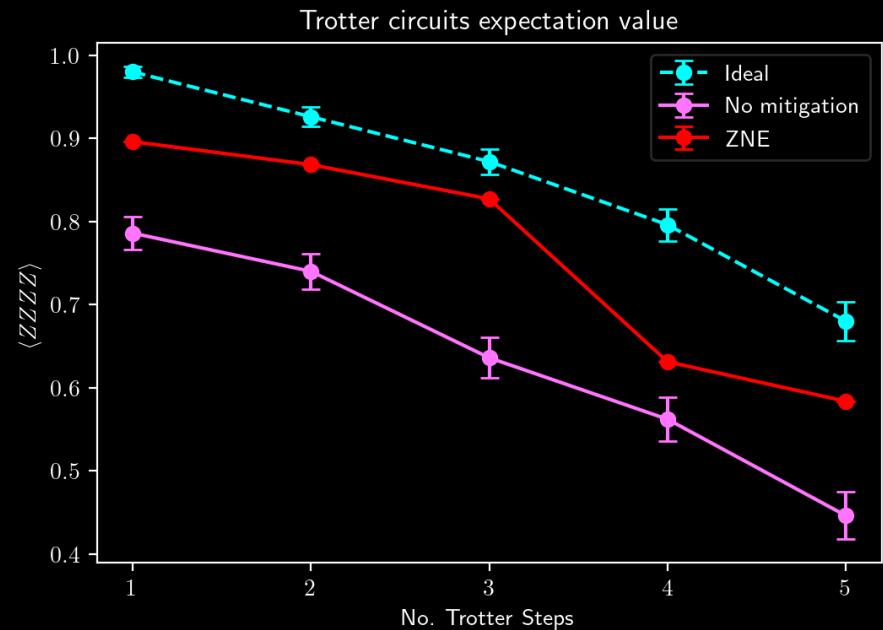
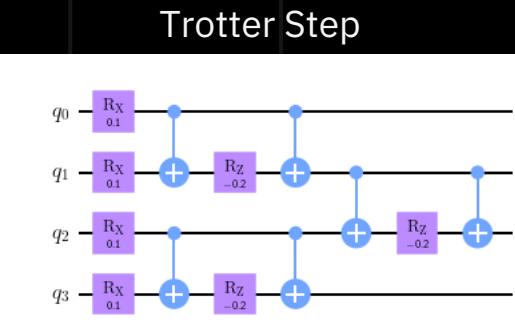
Noise Factors: 1, 1.1, 1.2



# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**

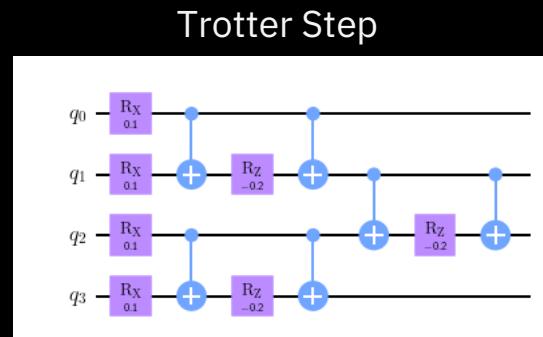
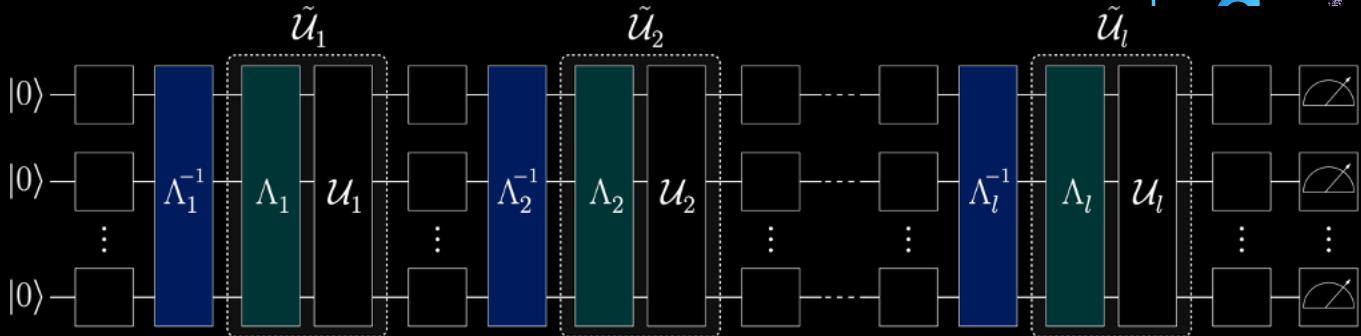


Qiskit Runtime: `options.resilience_level = 2`

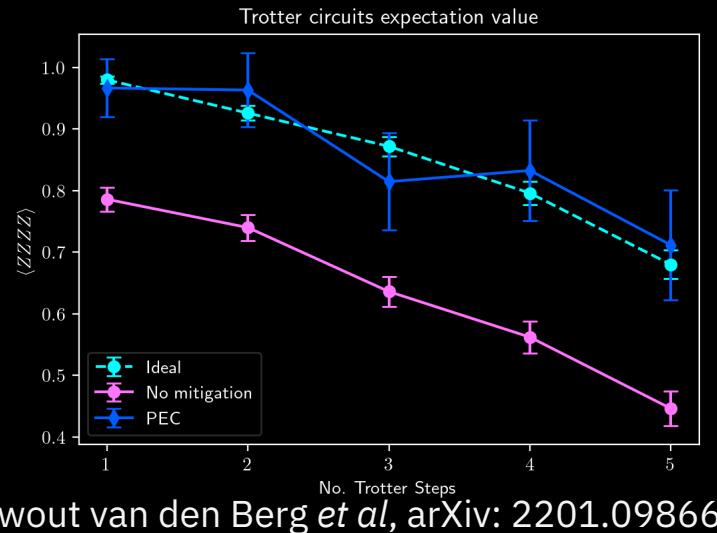
# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**



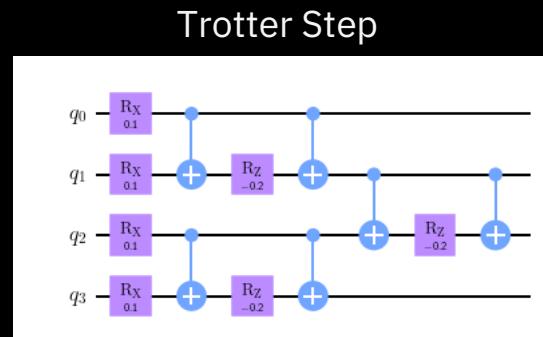
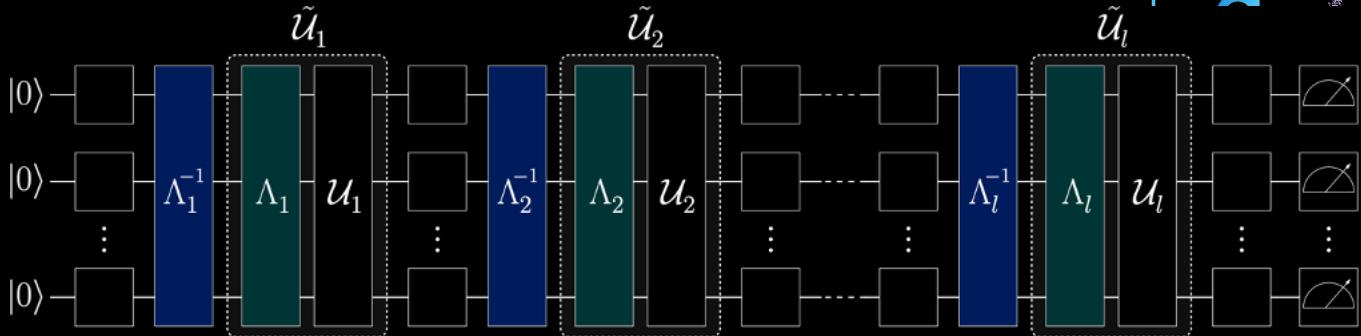
Qiskit Runtime: `options.resilience_level = 3`



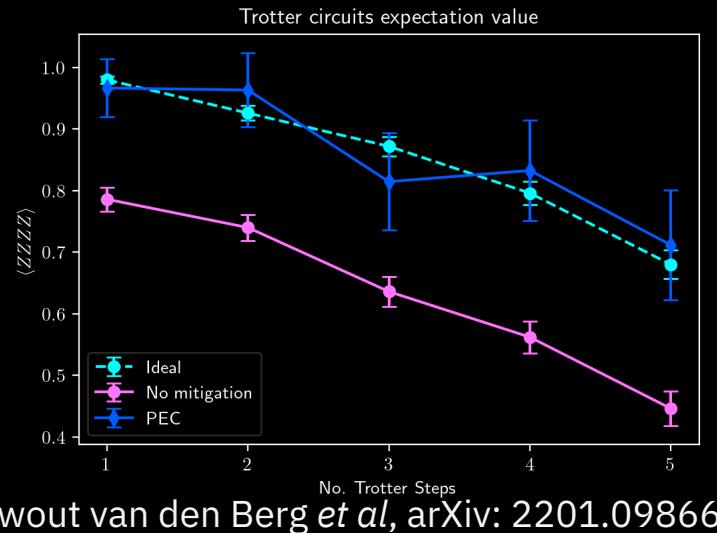
# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**



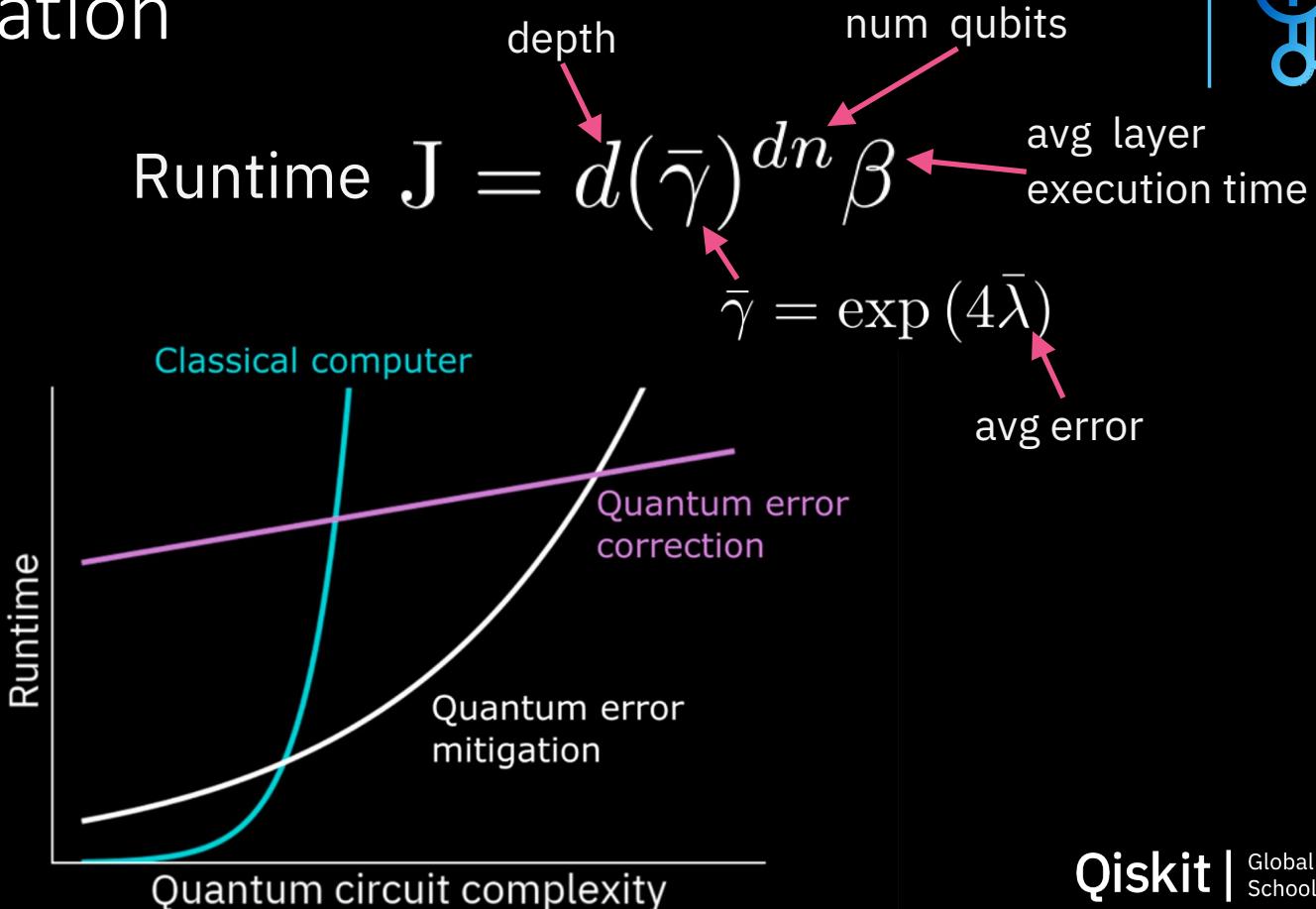
Qiskit Runtime: `options.resilience_level = 3`



# Error Mitigation



- **Zero Noise Extrapolation**
- **Probabilistic Error Cancellation**



# Thank you

