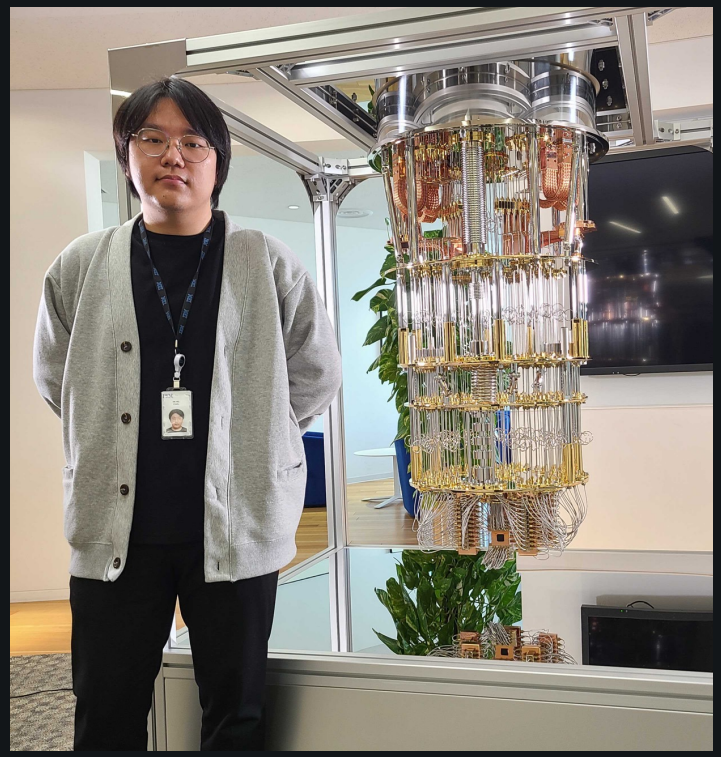
Qiskit Runtime Primitive 소개

Inho Choi
Qiskit Advocate

IBM Quantum / © 2021 IBM Corporation

Qiskit



최인호

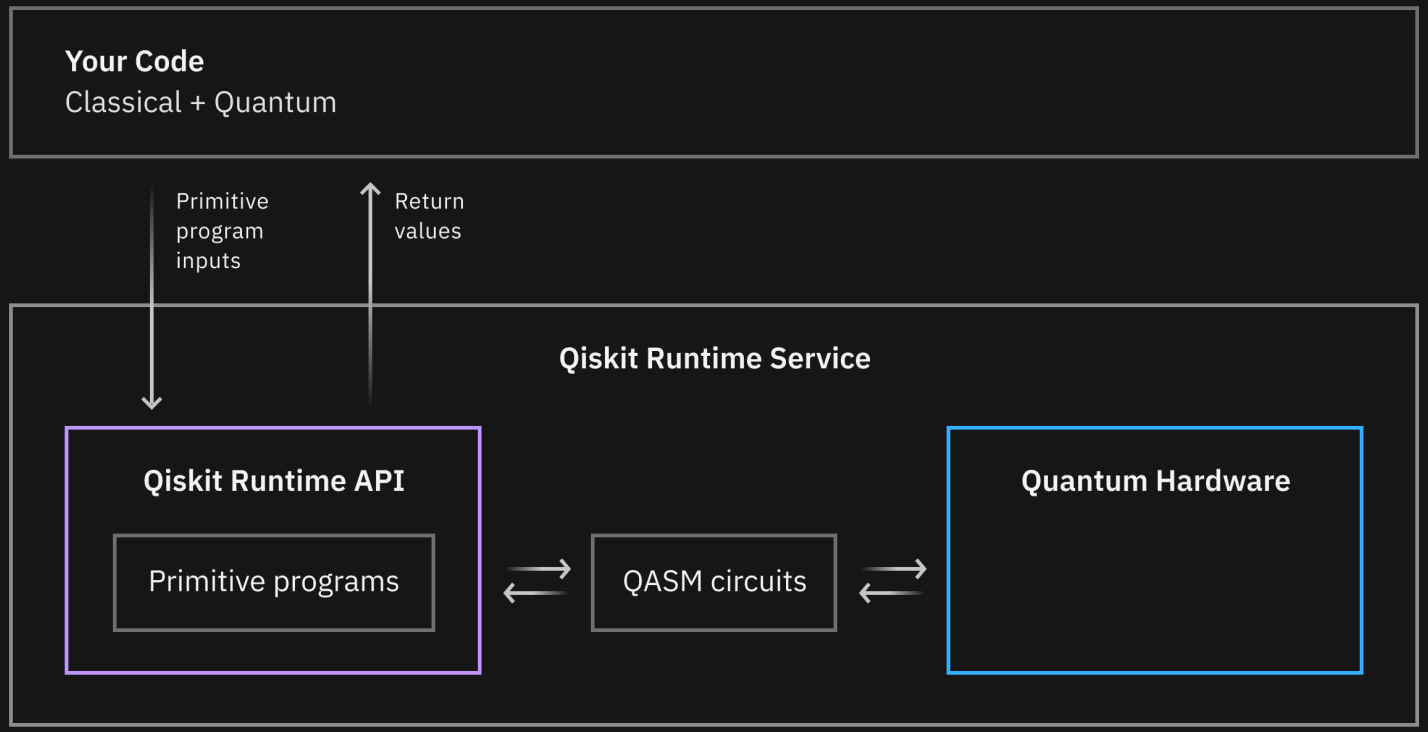
- Majoring Physics, Electronic Engineering and Computer Scinece in HKUST
- Qiskit Advocate
- Interested in Quantum Hardware

- Qiskit Slack @Inho Choi

IBM Quantum / © 2021 IBM Corporation

Qiskit Runtime

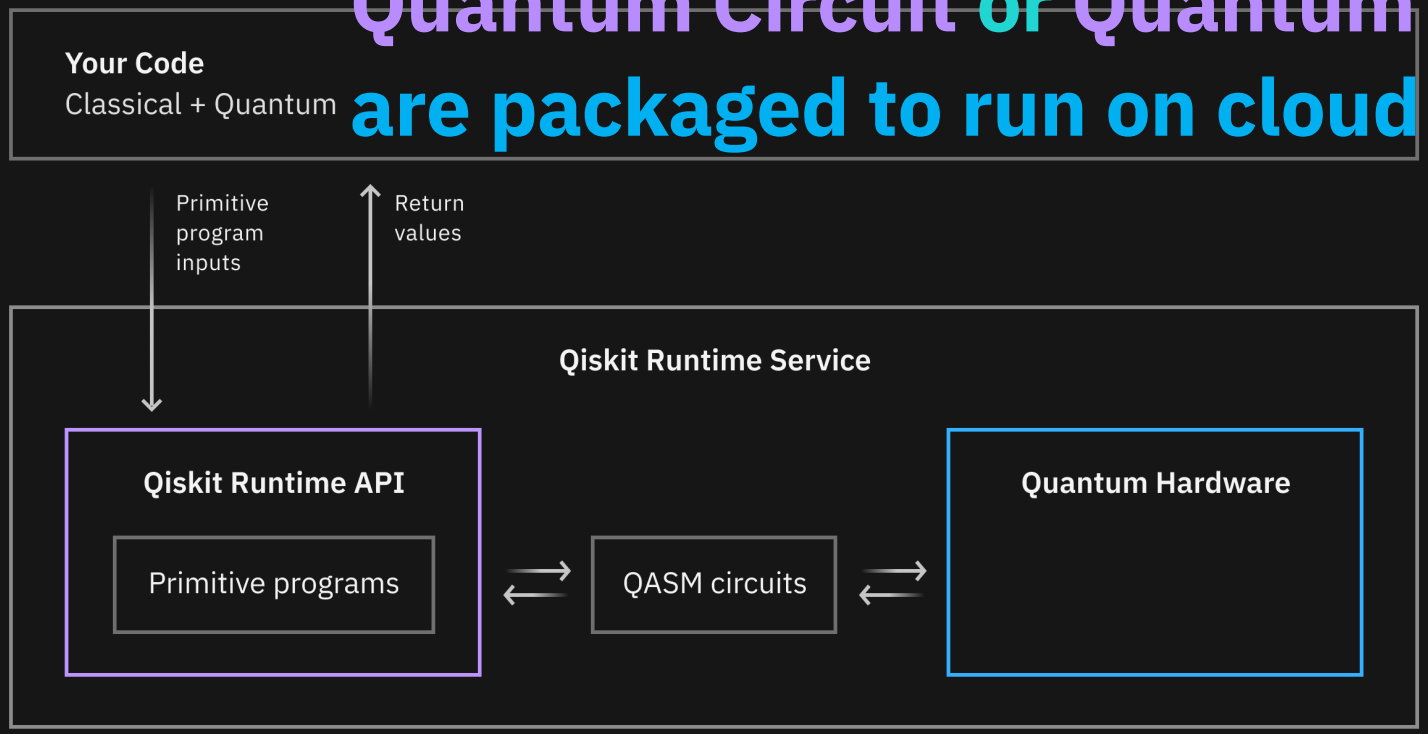
Qiskit



Qiskit Runtime

Qiskit

Quantum Circuit or Quantum Routines are packaged to run on cloud



Qiskit Runtime

Qiskit

- Remove lengthy communication trips back-and-forth to user's computers.
- Program executed in Qiskit Runtime -> More efficient
- Spend Less time waiting in queues -> Run **more** circuits in a day

Qiskit

Save Latency

Reduce overhead for iterative loops

IBM Quantum / © 2021 IBM Corporation

Qiskit

Efficiency

Save Latency

Reduce overhead for iterative loops

IBM Quantum / © 2021 IBM Corporation

Qiskit Runtime

Qiskit

Efficiency **Consistency**

Save Latency

Reduce overhead for iterative loops

Qiskit Runtime

Qiskit

Efficiency **Consistency** **Customizability**

Save Latency

Reduce overhead for iterative loops

Qiskit Runtime

Qiskit

Efficiency **Consistency** **Customizability**

Save Latency

Reduce overhead for iterative loops

Error Mitigation and Suppression

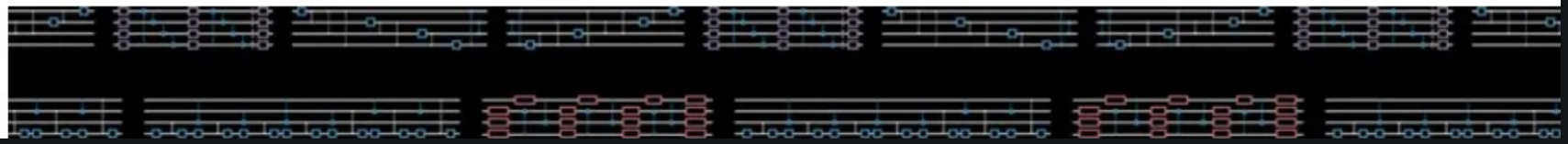
IBM Quantum / © 2021 IBM Corporation

120X Speedup

Qiskit

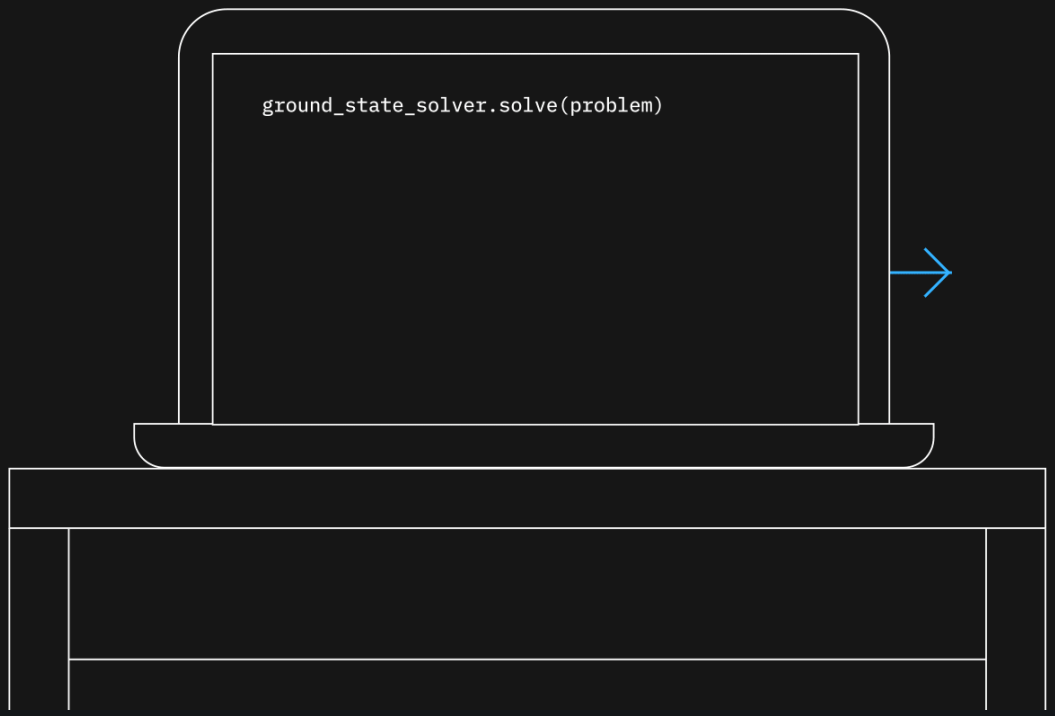
IBM Quantum delivers 120x speedup of quantum workloads with Qiskit Runtime

We're pleased to announce that the team demonstrated a 120x speedup in simulating molecules thanks to a host of improvements, including the ability to run quantum programs entirely on the cloud with Qiskit Runtime.



IBM Runtime workflow

Qiskit



Introduction to Primitives

Qiskit

Basic Element

that serve as a building block for more complex elements

Primitives with Quantum?

Qiskit

- Higher level core primitive terms with categorization
- Enable to build a program more accessibility
- Do not need low machine code level

Install Qiskit Runtime

Qiskit

Getting started ¶

Install Qiskit packages ¶

Install these packages. They let you create circuits and work with primitive programs via Qiskit Runtime.

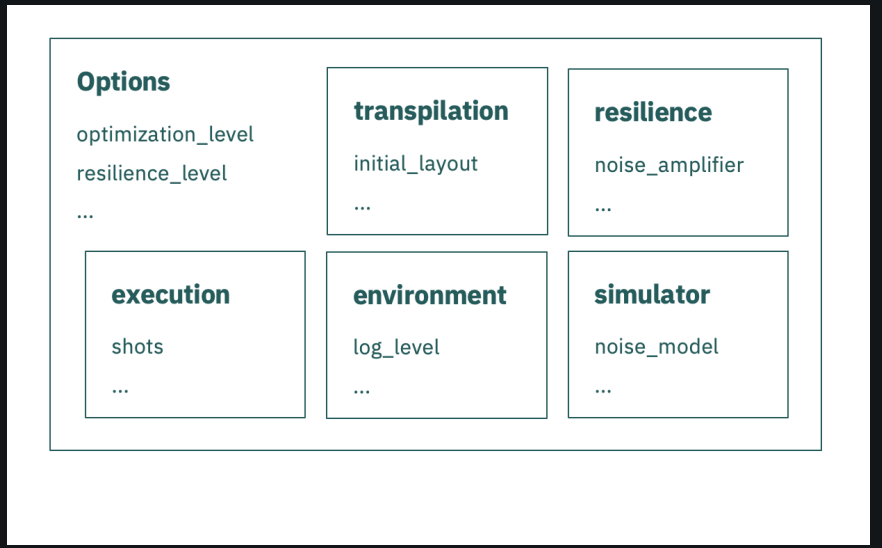
```
pip install qiskit
pip install qiskit-ibm-runtime
```

Qiskit Runtime Primitive keywords

Qiskit

- **Session**
  - Open and Close session to use the primitives on cloud
  - Define a job as collection of iterative calls to the quantum computer

- **Options**
  - Configure the current session and parameter
  - Control execution environment



IBM Quantum / © 2021 IBM Corporation

Qiskit Runtime Primitive

Qiskit

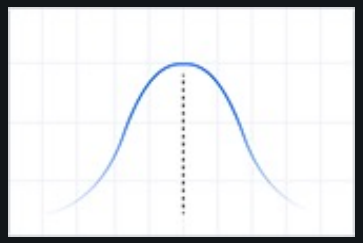
Sampler

Estimator

Qiskit Runtime Sampler

Qiskit

- Generates an error-mitigated readout of quasi-probabilities
  - **Input**: User Circuit

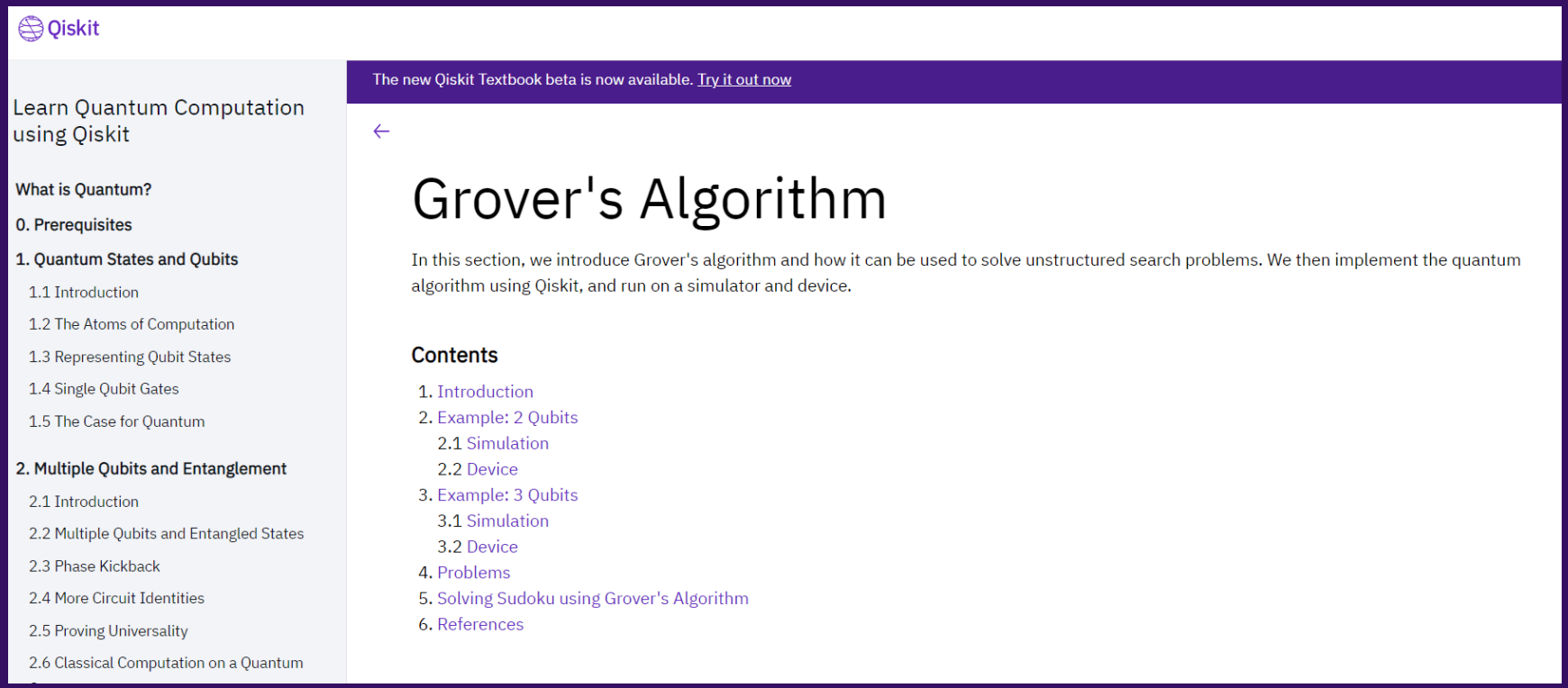


- **Better** evaluate shot result
  - Error mitigation
  - More efficient evaluation of the possibility of multiple relevant data points
    - **Where**: the context of destructive interference

https://qiskit.org/documentation/partners/qiskit_ibm_runtime/tutorials/how-to-getting-started-with-sampler.html

Grover Algorithm Demonstration

Qiskit



Qiskit Runtime Estimator

Qiskit

- Efficiently calculate and interpret expectation values of quantum operators required for many algorithms
- Allow selectively group between circuits and observables for execution
  - Efficient evaluation of expectation values and variances for given parameter input.

Qiskit Runtime Estimator

Qiskit

- Efficiently calculate and interpret expectation values of quantum operators required for many algorithms
- Allow selectively group between circuits and observables for execution
  - Efficient evaluation of expectation values and variances for given parameter input.

“No Measurements”

Parameterized circuits

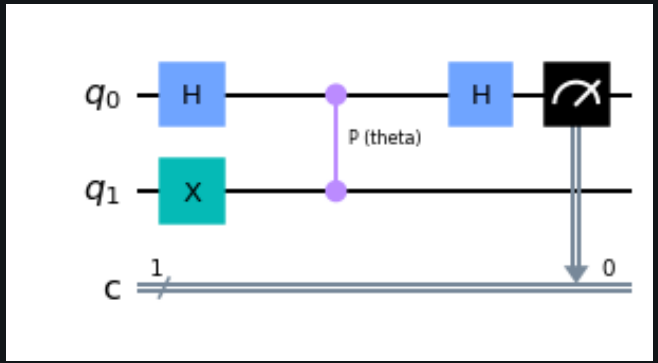
Qiskit

Primitive allows simplification of binding multiple parameters in parameterized circuits.

Parameterized circuits

Qiskit

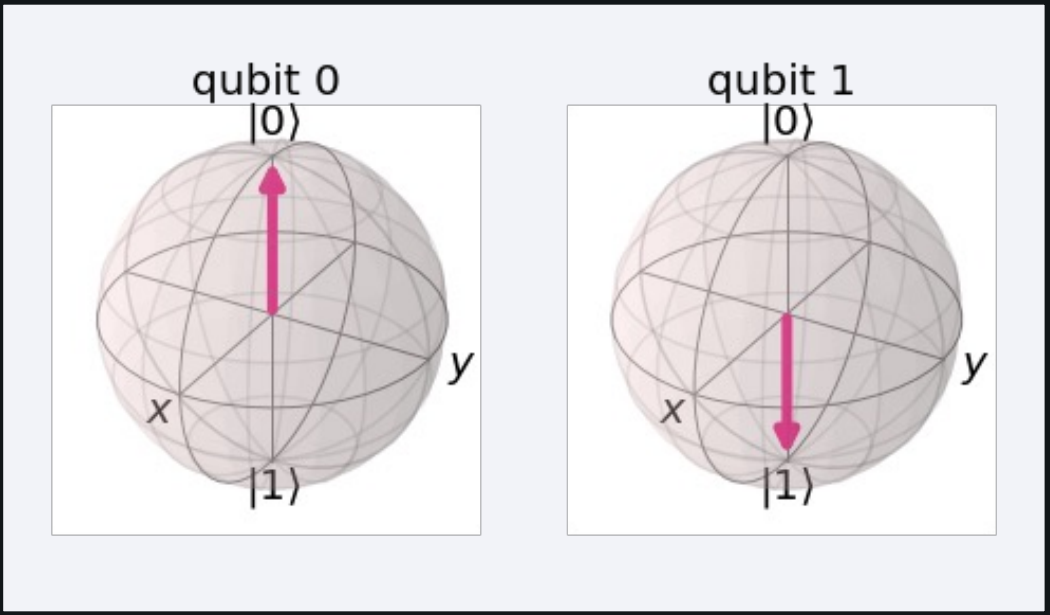
Primitive allows simplification of binding multiple parameters in parameterized circuits.



0 to 2π: divided over 50 evenly spaced points

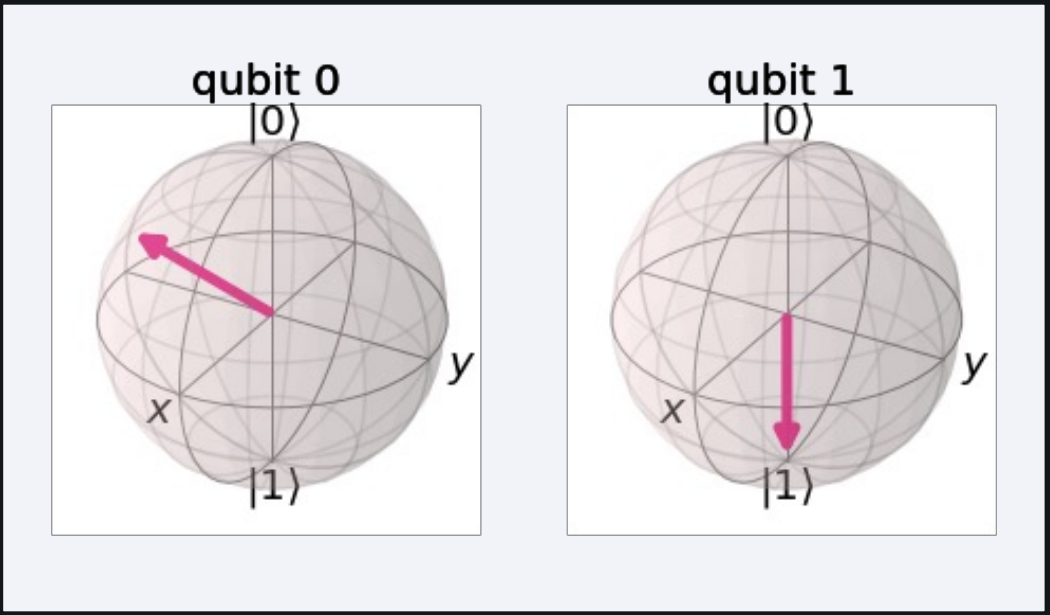
Parameterized circuits

Qiskit



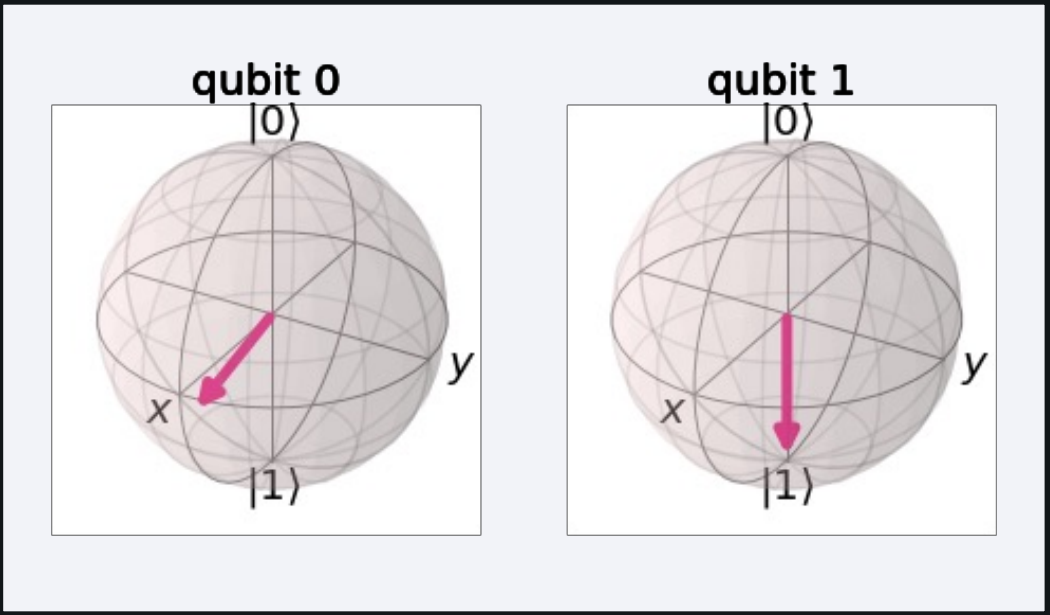
Parameterized circuits

Qiskit



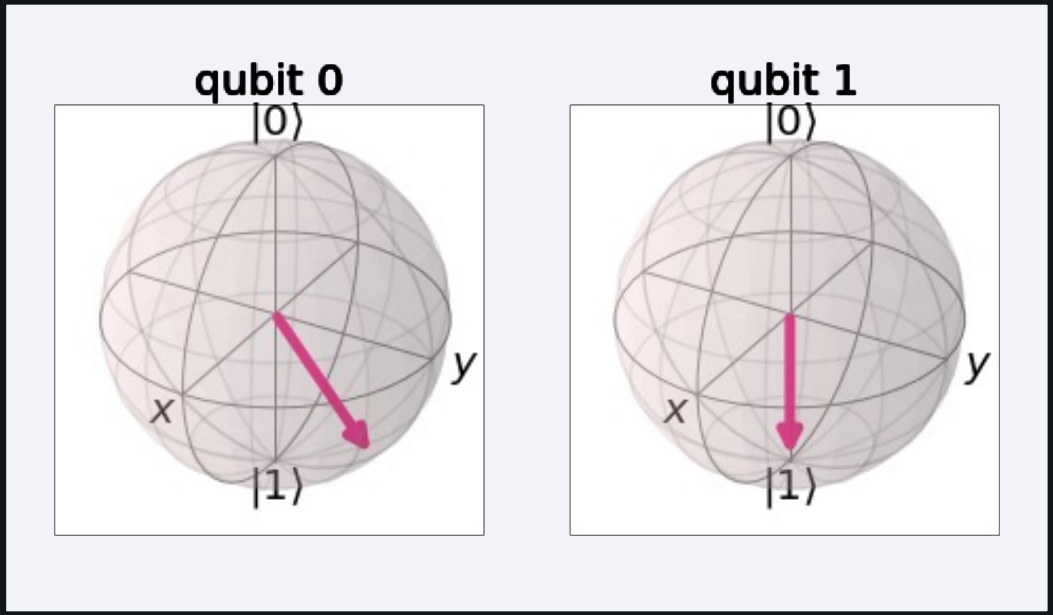
Parameterized circuits

Qiskit



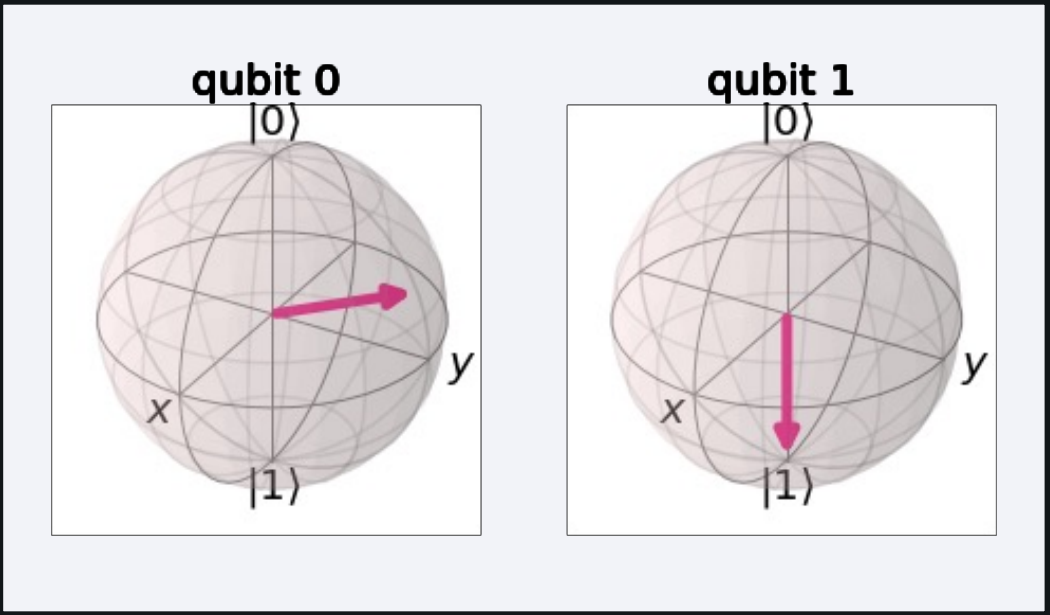
Parameterized circuits

Qiskit

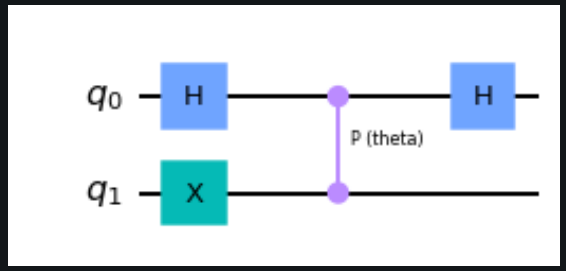


Parameterized circuits

Qiskit



Parametrized Circuit



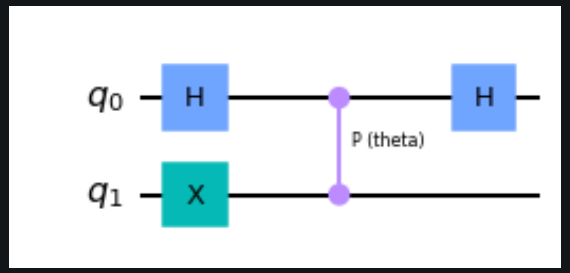
Parametrized Circuit

Qiskit

Calculate expectation value

$$\langle ZZ \rangle = \langle \psi | ZZ | \psi \rangle = \langle \psi | (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| - |1\rangle\langle 1|) | \psi \rangle = |\langle 00|\psi\rangle|^2 - |\langle 01|\psi\rangle|^2 - |\langle 10|\psi\rangle|^2 + |\langle 11|\psi\rangle|^2$$

```
ZZ = SparsePauliOp.from_list([("ZZ", 1)])
```



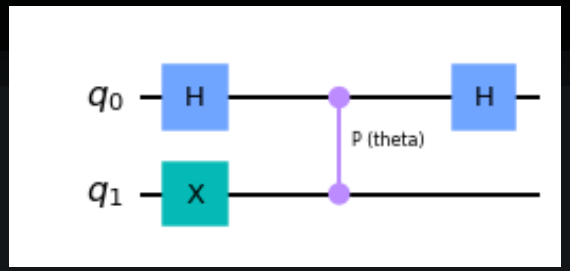
Parametrized Circuit

Qiskit

Run using Estimator

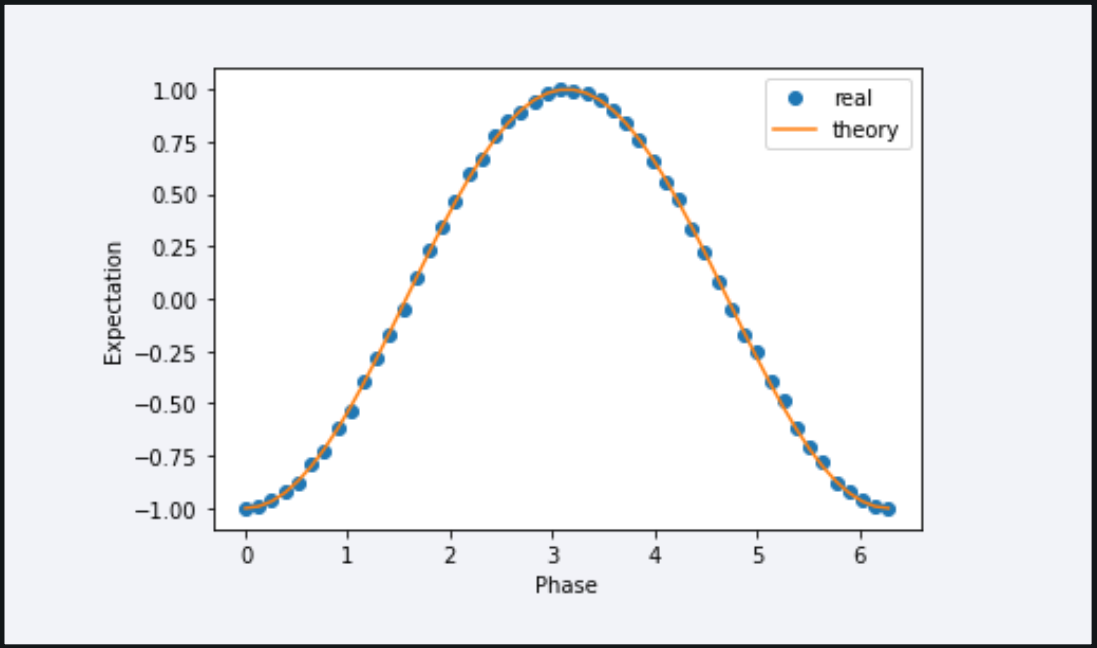
```
options = Options(simulator={"seed_simulator": 42}, resilience_level=0)

with Session(service=service, backend=backend):
    estimator = Estimator(options=options)
    job = estimator.run(circuits=[qc_no_meas]*len(phases), parameter_values=individual_phases,
observables=[ZZ]*len(phases))
```



Parametrized Circuit

Qiskit



Error Suppression

Qiskit

- The most basic level of error handling
- Use knowledge about the undesirable effects to introduce customization
  - **Anticipate** and **avoid** the potential impacts of these effects

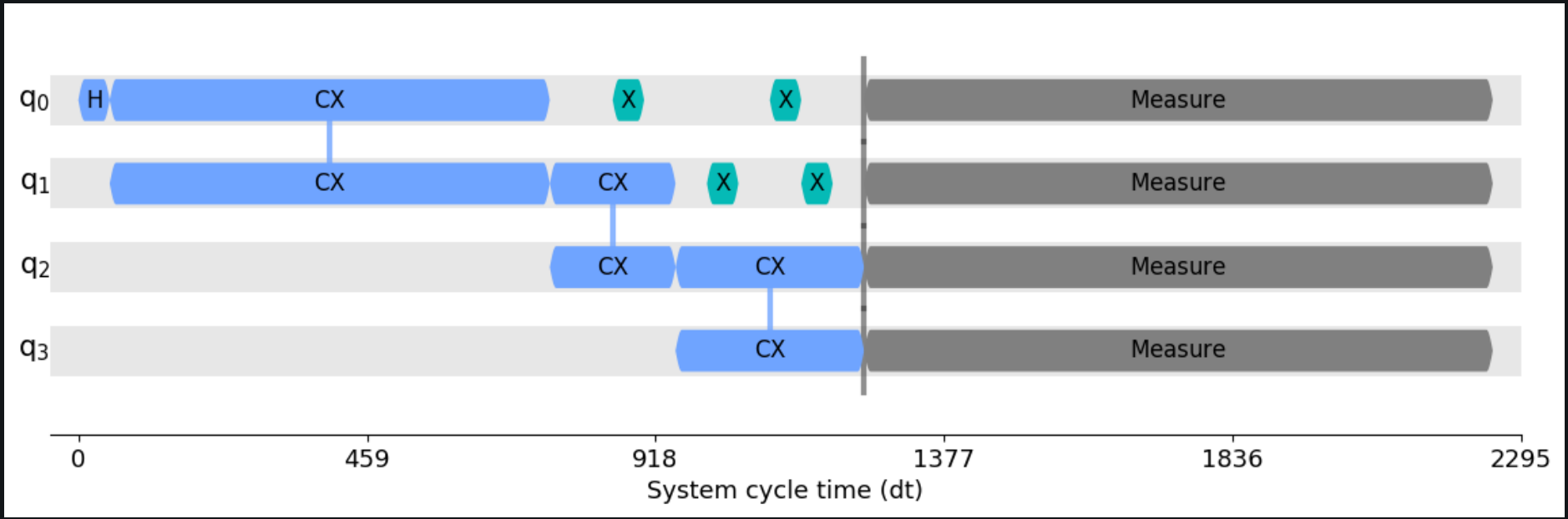
Dynamical Decoupling

- This feature is enabled by default on Qiskit Runtime
- Used to increase the lifetime of quantum information

Dynamical Decoupling

Qiskit

- Effectively disconnect the environment using decoupling methods
  - Prevent idle state of qubit – leakage of information to surrounding due to decoherence.



Error Mitigation

Qiskit

- **Reduce** the error effects with a much smaller overhead
  - Instead of completely eliminating using a huge amount of resourecs
- Error correction is not practically feasible in the current NISQ
  - Number of qubits
  - Controlled error rates
  - Extra circuit depth and measurements

Error Mitigation with Sampler

Qiskit

- Matrix-free Measurement Mitigation (M3)

https://qiskit.org/documentation/partners/mthree/

M3

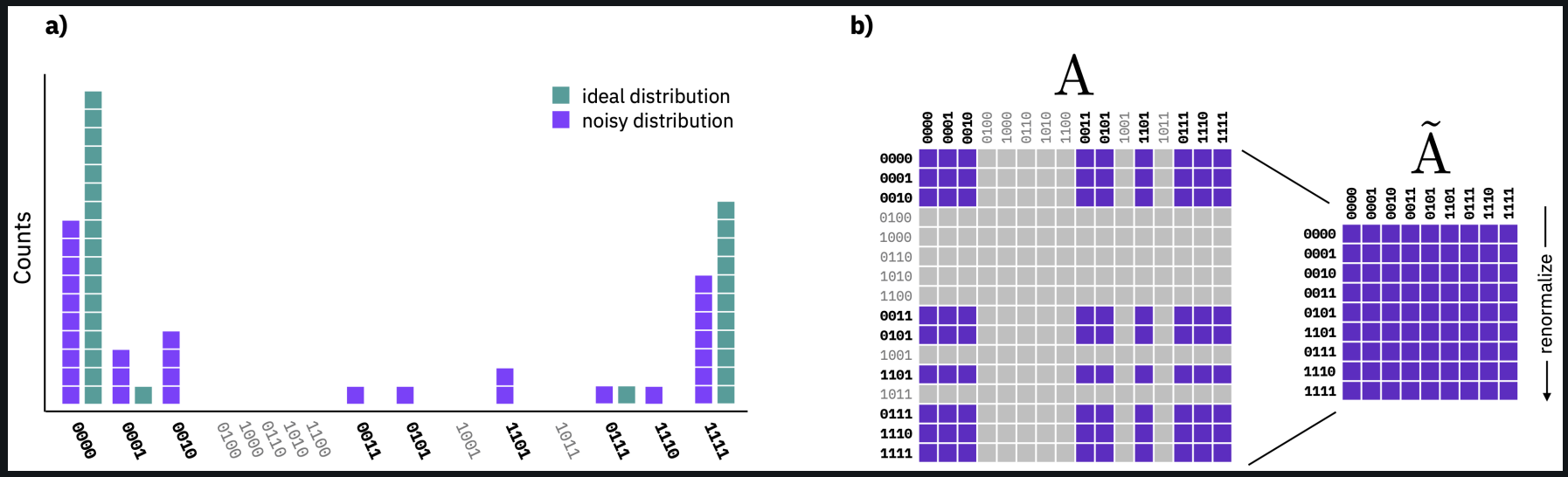
Qiskit

- This feature is enabled by setting `resilience_level=1` in `Options` with Sampler
- Scalable quantum measurement error mitigation
  - Need not explicitly form the assignment matrix or its inverse.

M3

Qiskit

- M3 works in a reduced subspace defined by the noisy input bitstrings that are to be corrected.
- Resulting linear system of equations is nominally much easier to solve.
  - Much smaller number of unique bitstrings than full multi-qubit Hilbert space



M3

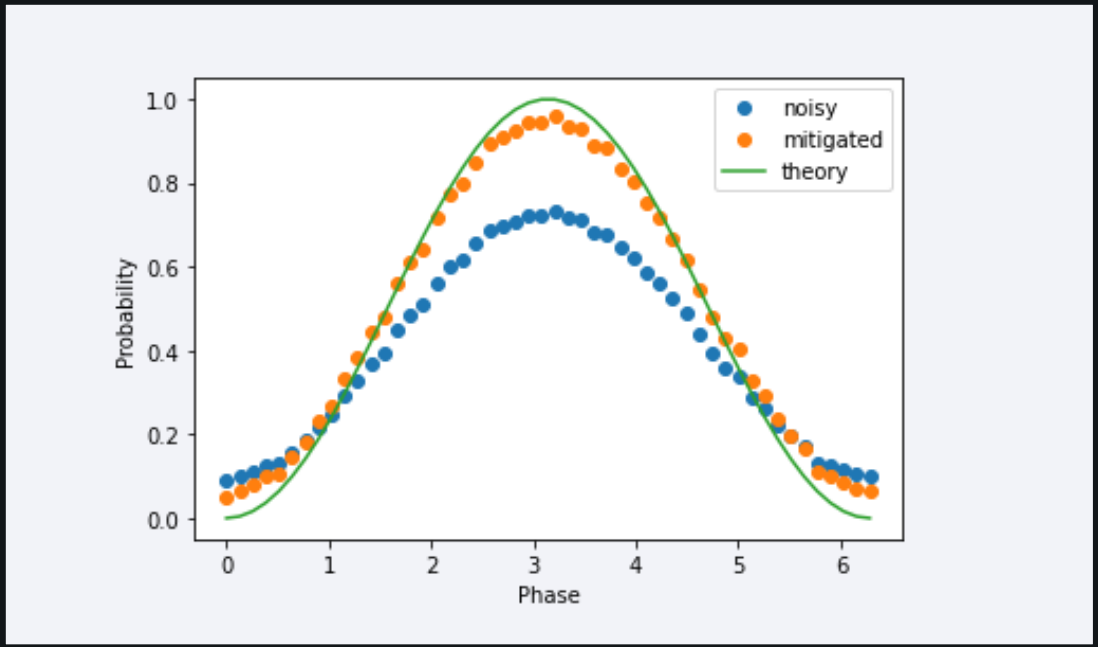
Qiskit

```python
# Import FakeBackend
fake_backend = FakeManila()
noise_model = NoiseModel.from_backend(fake_backend)

# Set options to include noise_model and resilience_level
options_with_em = Options(
    simulator={
        "noise_model": noise_model,
        "seed_simulator": 42,
    },
    resilience_level=1
)
```

M3

Qiskit


noisy
mitigated
theory
Probability
Phase
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6

Error Mitigation with Estimator

Qiskit

- Twirled Readout Error eXtinction (T-Rex)
- Zero Noise Extrapolation (Digital ZNE)
- Probabilistic error cancellation (PEC)

•T-REx: arXiv:2012.09738
•ZNE: arXiv:1612.02058
•PEC: arXiv:1612.02058

T-Rex

Qiskit

- This feature is enabled by setting `resilience_level=1` in `Options` with Estimator (Default)
- Implementation which involves "twirling" of gates
- View noise as a set of **extra probabilistic gates** on top of our perfect circuit implementation
  - Conjugate this noisy gate set with a gate randomly chosen from a set of gates
- Inserts pairs of Pauli gates (I, X, Y, Z) **before** and **after** entangling gates such that the overall unitary is the same
- Turning **coherent errors** into **stochastic errors**
  - **Stochastic errors can be eliminated by sufficient averaging**

T-Rex

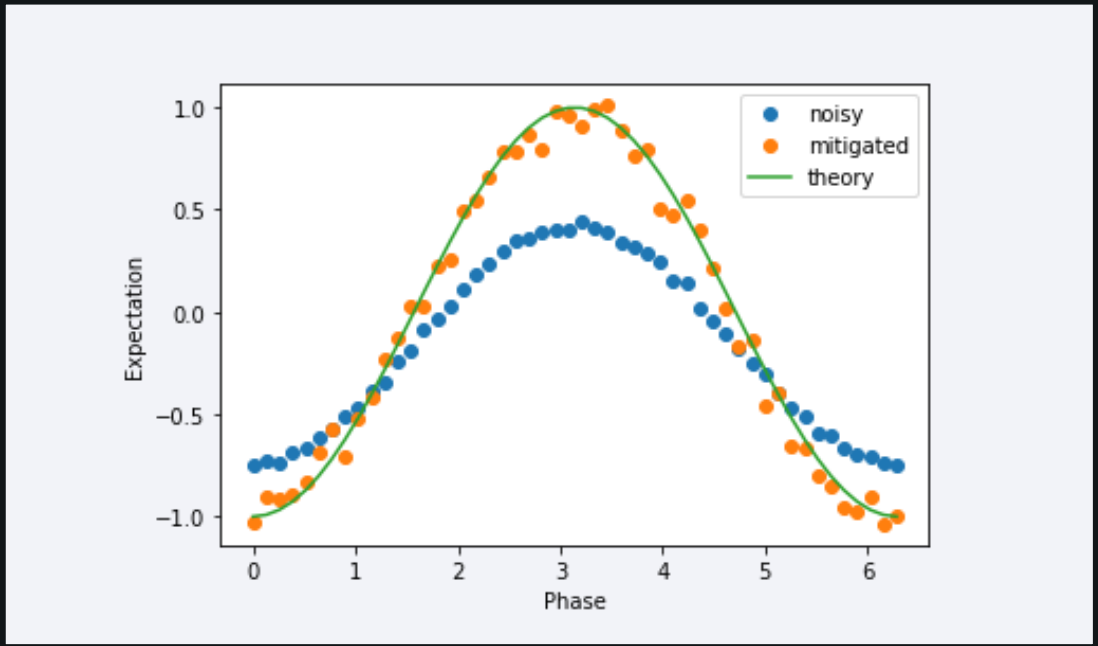
Qiskit

```
# Import FakeBackend
fake_backend = FakeManila()
noise_model = NoiseModel.from_backend(fake_backend)

# Set options to include noise_model and resilience_level
options_with_em = Options(
    simulator={
        "noise_model": noise_model,
        "seed_simulator": 42,
    },
    resilience_level=1
)
```

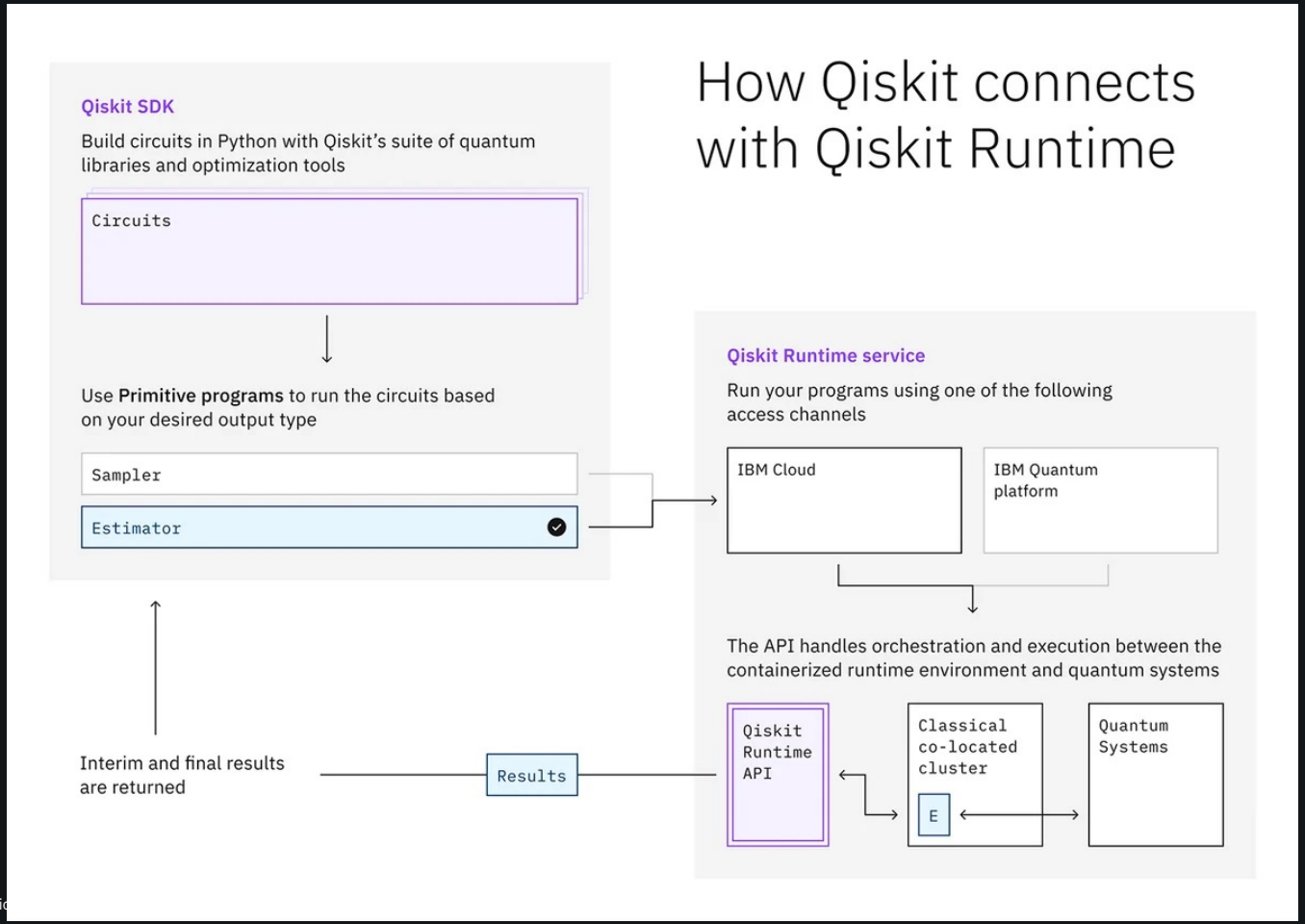
T-Rex

Qiskit


noisy
mitigated
theory
Expectation
Phase
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
6

Digital ZNE

Qiskit

- This feature is enabled by setting `resilience_level=2` in `Options` with Estimator
- Mitigating errors in noisy quantum computers without the need for additional quantum resources.
- A quantum program is altered to run at different effect levels of processor noise
- The result of the computation is extrapolated to an estimated value at a noiseless level.
- Digital way means not using physical pulse.
- Still an active research question around which method is the best to use

How Qiskit connects to Qiskit Runtime