Building multipartite entangled states (GHZ states) with minimal error

Video tutorial

Jaeun Kim, IBM Quantum

Overview

- **Theme**: Multipartite entangled state (GHZ state)
  - Many applications: ex. Quantum sensing, quantum communication
  - Latest Nobel prize in physics
- **Learning objective**: Understanding the low-level operations for superconducting qubits
  - Simulation with Qiskit Dynamics
  - Pulse calibration with Qiskit Pulse & Experiments (Prob #2)
- **Task:** error mitigation
  - Readout error mitigation (Prob #1)
  - Zero-noise extrapolation (Prob #3)

Resources

- GitHub repo: https://github.com/qiskit-community/2023-qhackathon-korea/blob/main/preparation%26installation.md
- Qiskit textbook & tutorial
- Slack channel: #2023-qhackathon-korea
- Mentor office hour

Prob #1-3: Readout Error Mitigation

- M = Noise matrix
- $C_{noisy} = MC_{ideal}$, Thus $C_{ideal} = M^{-1}C_{noisy}$
- Do NOT use Qiskit Terra's CompleteMeasFitter() as it's deprecated
- Option 1: Implementing readout error mitigation on your own
  - Step 1. Prepare 8 circuits that give $|000\rangle, |001\rangle, \ldots |111\rangle$, respectively.
  - Step 2. For each circuit, measure w.r.t. computational basis to get M
  - Step 3. Calculate M^{-1} and apply it to the results from your GHZ circuit

More points if you come up with an improved scheme!

Prob #1-3: Readout Error Mitigation

- Option 2: Using Qiskit Runtime's Sampler primitive

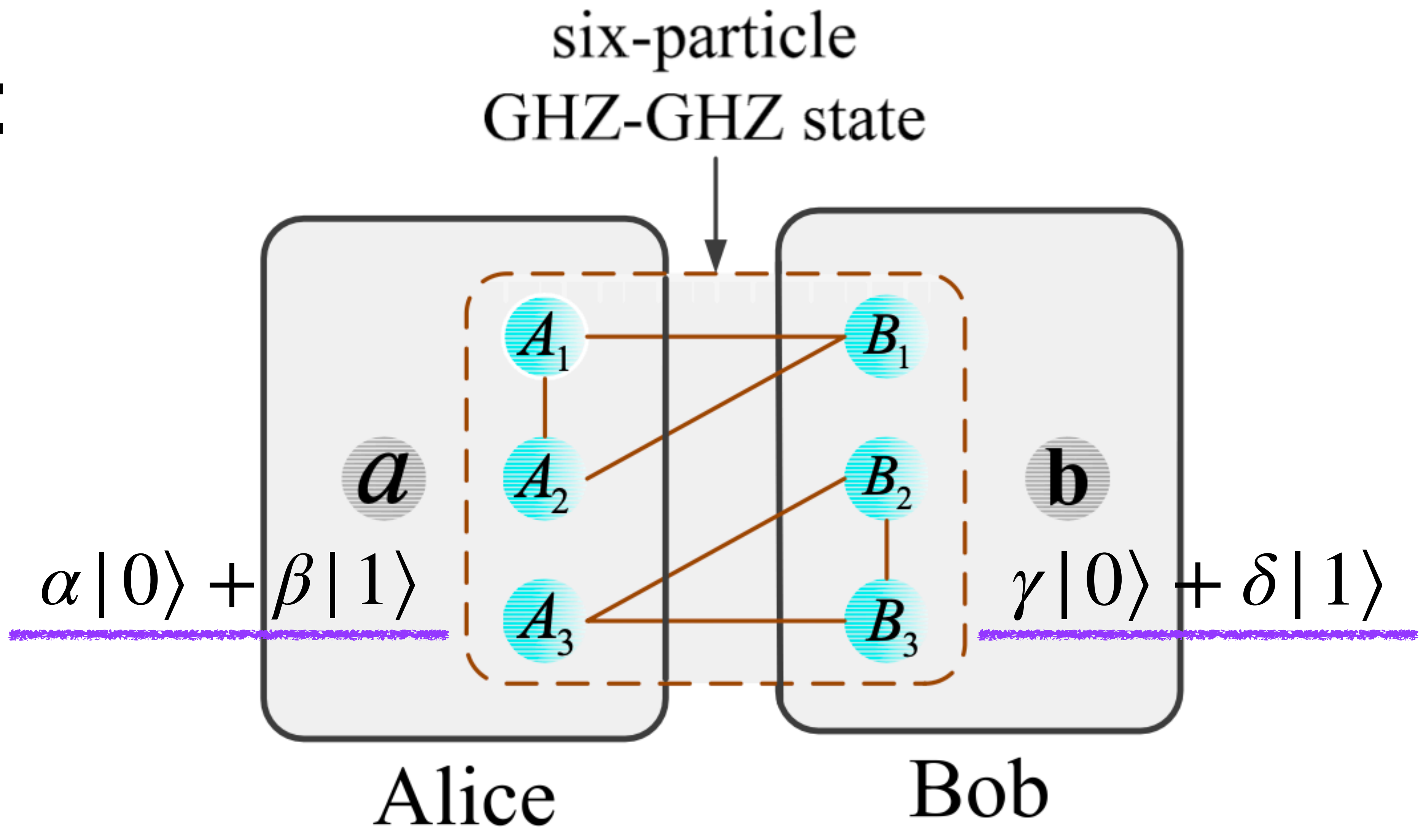
```
from qiskit_ibm_runtime import QiskitRuntimeService, Session, Sampler, Options

service = QiskitRuntimeService()
options = Options()
options.resilience_level = 1
options.optimization_level = 3

with Session(service=service, backend="ibmq_qasm_simulator") as session:
    sampler = Sampler(session=session, options=options)
```

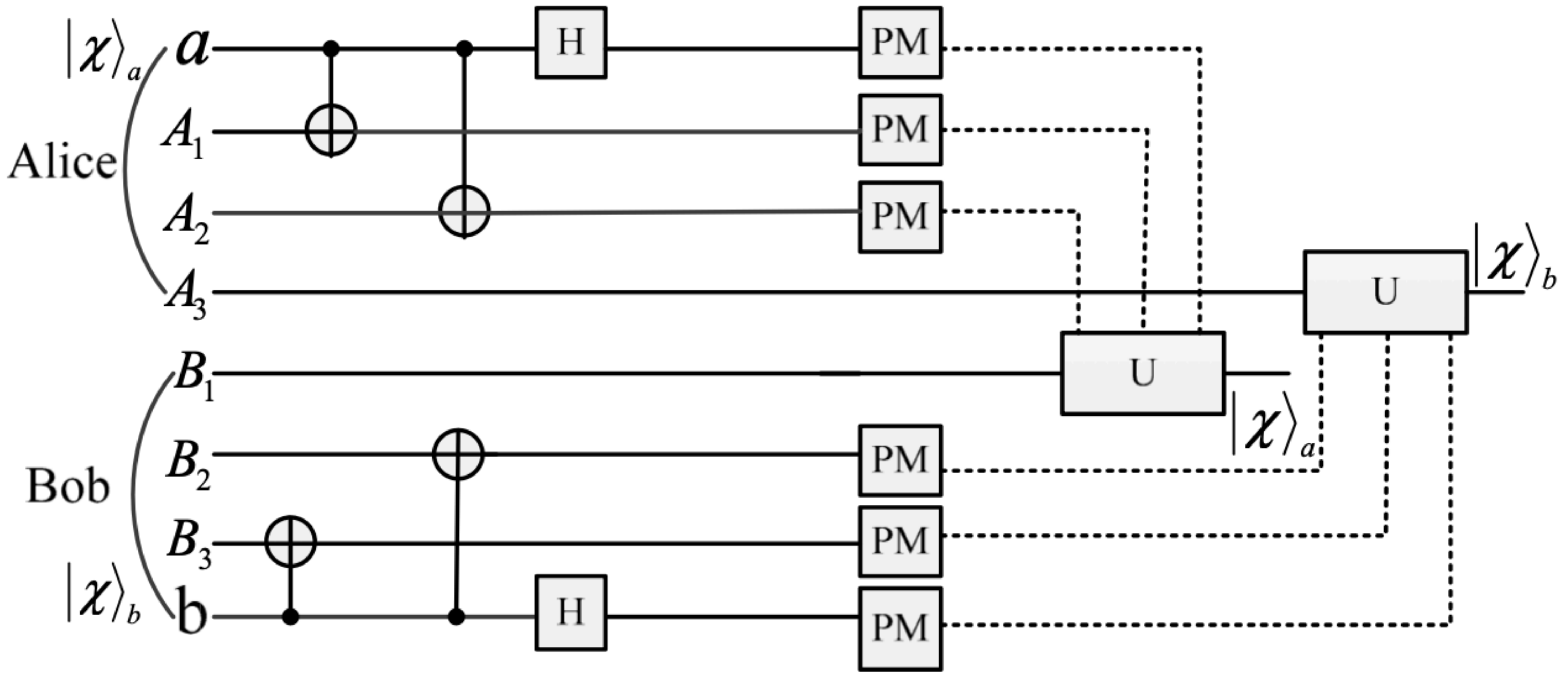
  - Estimator: expectation values
  - Sampler: probability distribution ✅
    - Set resilience_level to 1

Prob #1-4: Quantum Communication using GHZ

One-hop bidirectional quantum communication

- Entangled states: resources for quantum communication
  - No-cloning theorem (only teleportation is possible)
  - Correlation
- Task:

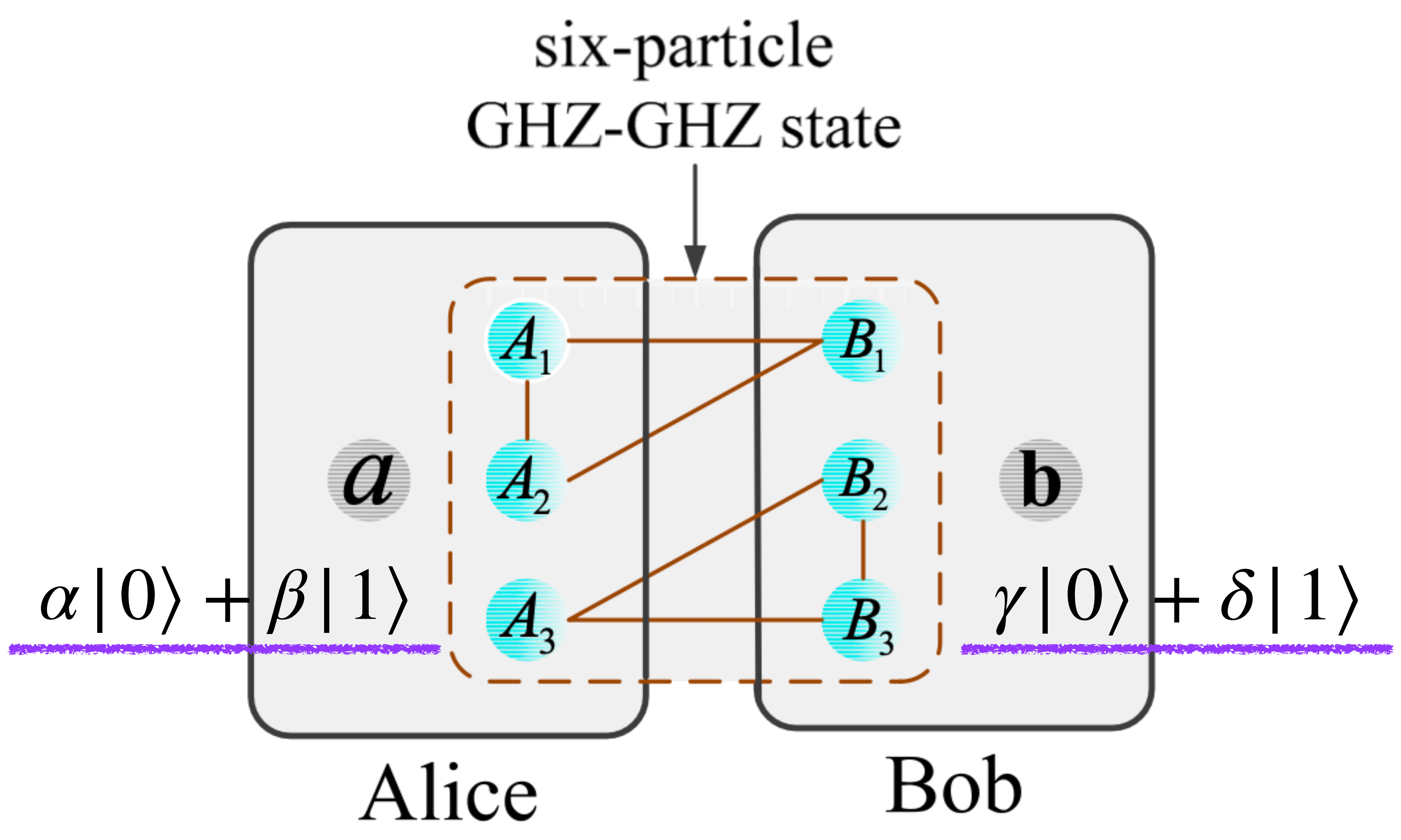




Prob #1-4: Quantum Communication using GHZ

One-hop bidirectional quantum communication

- Cf) DON'T consider other types of GHZ states ($|\psi^2\rangle, |\psi^3\rangle$, etc)



$$
\begin{aligned}
|\Upsilon'_1\rangle = \tfrac{1}{4} \Big\{ & [|000\rangle_{aA_1A_2}(\alpha|0\rangle + \beta|1\rangle)_{B_1} + |100\rangle_{aA_1A_2}(\alpha|0\rangle - \beta|1\rangle)_{B_1} \\
& + |011\rangle_{aA_1A_2}(\alpha|1\rangle + \beta|0\rangle)_{B_1} + |111\rangle_{aA_1A_2}(\alpha|1\rangle - \beta|0\rangle)_{B_1}] \\
& \otimes [|000\rangle_{bB_2B_3}(\gamma|0\rangle + \delta|1\rangle)_{A_3} + |100\rangle_{bB_2B_3}(\gamma|0\rangle - \delta|1\rangle)_{A_3} \\
& + |011\rangle_{bB_2B_3}(\gamma|1\rangle + \delta|0\rangle)_{A_3} + |111\rangle_{bB_2B_3}(\gamma|1\rangle - \delta|0\rangle)_{A_3}] \Big\}
\end{aligned}
$$

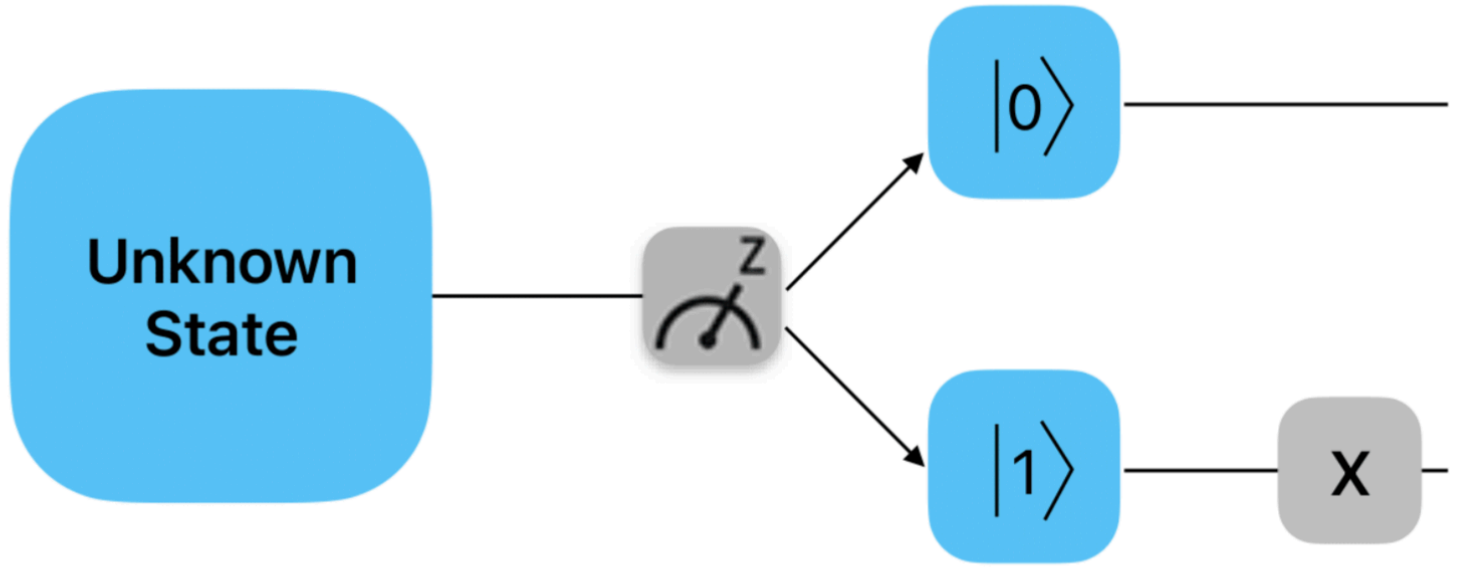
Table 1. The relations between Alice's and Bob's measurement results and unitary operations.

Received Measurement Result	Unitary Operation
$|000\rangle$	I
$|011\rangle$	X
$|100\rangle$	Z
$|111\rangle$	ZX

Prob #1-4: Quantum Communication using GHZ

One-hop bidirectional quantum communication

- Dynamic circuits: Quantum circuits with mid-circuit measurements & classical feedforward



Cond (Cbit, value)

```
with qc.if_test((b0, 0)) as else_:
    qc. Operation ( Qubit )
with else_:
    qc. Operation ( Qubit )
```

Executes if TRUE

Executes if FALSE

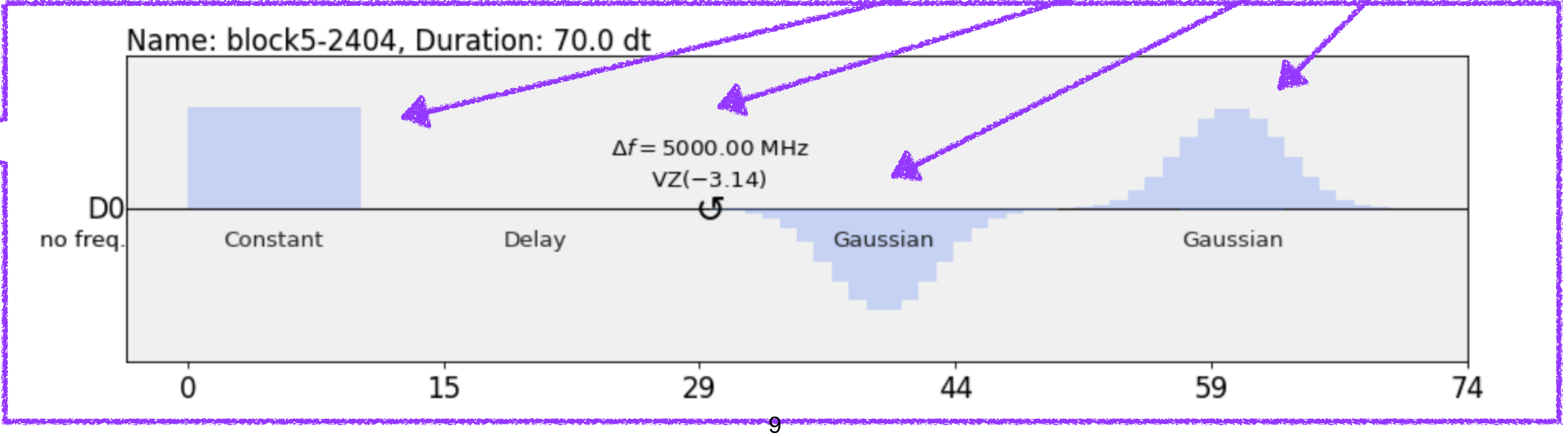
Prob #2: Pulse-level Calibration

Using Qiskit Pulse and Qiskit Experiments libraries

- Superconducting transmon qubits: gates = microwave pulses
- Pulse-level control of quantum hardware → more precise control of the quantum systems
    - ex) pulse calibration to get gates with higher fidelity
- Qiskit Pulse library

Instructions
(→ gates)

Schedule
(→ circuit)



.play(*pulse*)

Single-qubit
gaussian

Two-qubit
Echoed cross resonance

Etc

Pulse Builder Syntax

Easy API to build a circuit out of Pulses

from qiskit.circuit import Parameter
Symbolic parameter
(You'll need this for calibrations)

Schedule

```
beta = Parameter("beta")
with pulse.build() as my_pulse:
    pulse.play(drag(beta, amplitude, 0, sigma, name=None), pulse.DriveChannel(0))
```

Your quantum gate as a pulse

Required parameter for Pulse.play()
make sure to use different channels
for different qubits

Qiskit Experiments

Automates routine tasks (ex. calibration)

Schedule
(Instantiated in the previous slide)

Dynamic backend
Or
Real quantum hardware

Parameter scan range

```
cal_drag0 = RoughDrag(0, my_pulse, backend=backend, betas=np.linspace(-5, 0, 50))
cal_drag0.set_experiment_options(reps=[3, 5, 7])
drag0_data = cal_drag0.run(shots=1000).block_for_results()

df = pd.DataFrame({'rotation angle': [rotation_angle],\
            'beta.n': [drag0_data.analysis_results("beta").value.n],\
            'beta.s': [drag0_data.analysis_results("beta").value.s],\
            'data': [drag0_data]})
```

Run the experiment

Access the results

Prob #3. Zero-Noise Extrapolation

- One of error mitigation techniques to improve the <u>accuracy of the expectation value</u> obtained by noisy quantum computers
- Run circuits with varying noise levels → extrapolate to its zero-noise limit
    - Step 1. Stretch the control pulses in time using Qiskit Pulse
        - The errors will accumulate
    - Step 2. Make sure to calibrate the pulses for the new (stretched) durations!
        - Rotation angle $\propto$ area under the pulse envelope
        - If you stretch the pulse so that the area is changed, you get a nonsense circuit

Reference for ZNE

Error mitigation for short-depth quantum circuits

Kristan Temme, Sergey Bravyi and Jay M. Gambetta
IBM T.J. Watson Research Center, Yorktown Heights NY 10598
(Dated: November 7, 2017)

Two schemes are presented that mitigate the effect of errors and decoherence in short-depth quantum circuits. The size of the circuits for which these techniques can be applied is limited by the rate at which the errors in the computation are introduced. Near-term applications of early quantum devices, such as quantum simulations, rely on accurate estimates of expectation values to become relevant. Decoherence and gate errors lead to wrong estimates of the expectation values of observables used to evaluate the noisy circuit. The two schemes we discuss are deliberately simple and don't require additional qubit resources, so to be as practically relevant in current experiments as possible. The first method, extrapolation to the zero noise limit, subsequently cancels powers of the noise perturbations by an application of Richardson's deferred approach to the limit. The second method cancels errors by resampling randomized circuits according to a quasi-probability distribution.

- https://arxiv.org/pdf/1612.02058.pdf
- You will need to use the Dynamics simulator
  - Install JAX 0.4.6.!!

Good Luck!