

2023 年 10 月 4 日 Quantum Tokyo

量子テレポーテーション

Qiskit Advocate
Yusuke Mizutani

勉強内容 Qiskit Textbookより

- 量子テレポーテーションを理解する
 - 言葉で理解する
 - 量子回路と数式で理解する
- 量子テレポーテーションの動作を検証する
 - エンタングルメントしたペアの片割れを量子テレポーテーションして検証する

本勉強会のゴール：最新の研究でも量子テレポーテーション役に立っていることを知る

1 | 量子テレポーテーションを理解する

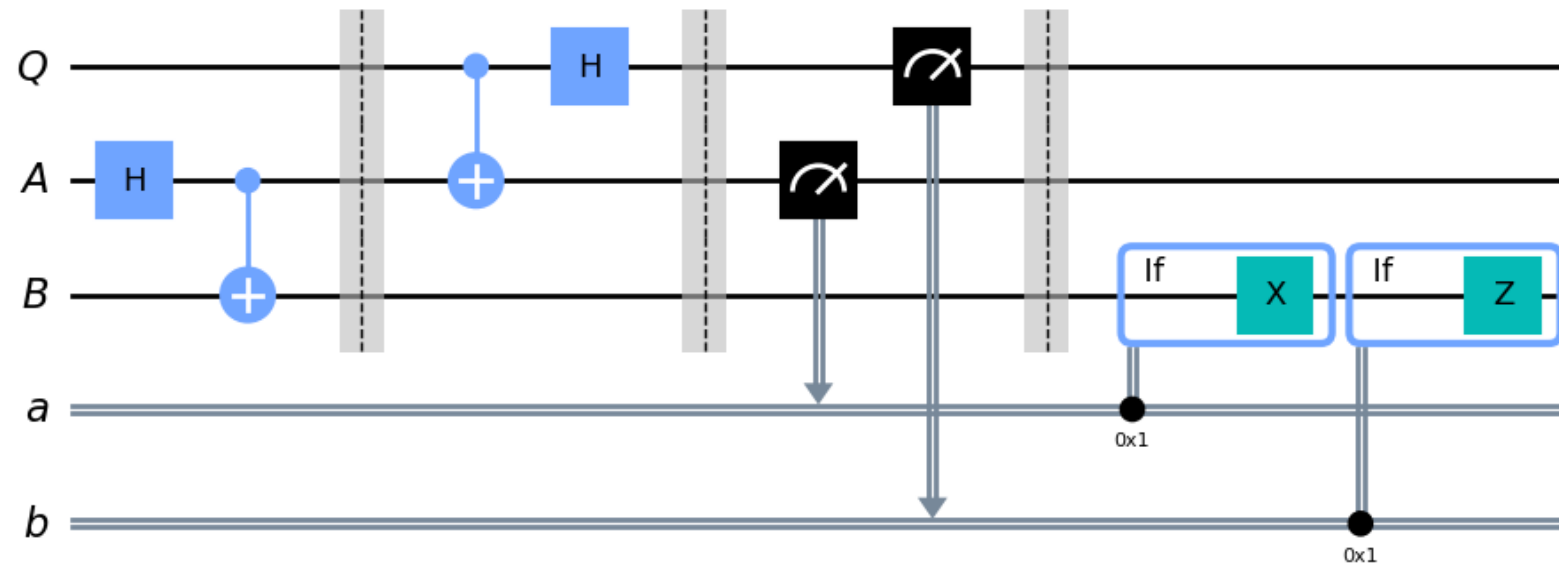
量子テレポーテーションとは

Aliceが未知の量子状態を遠く離れたBobに送信するプロトコル

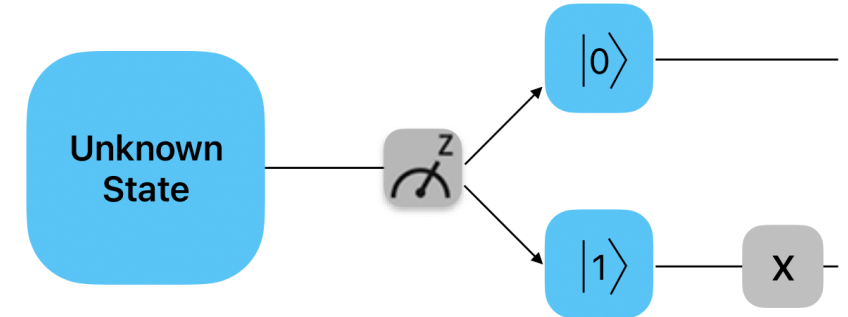
- 複製不可能定理は破ってない
- AliceとBobは**エンタングルした量子ビットのペア**を持っている
- 古典通信を使う(電話やメール)

扱う量子回路

量子テレポーテーション



Dynamic Circuit



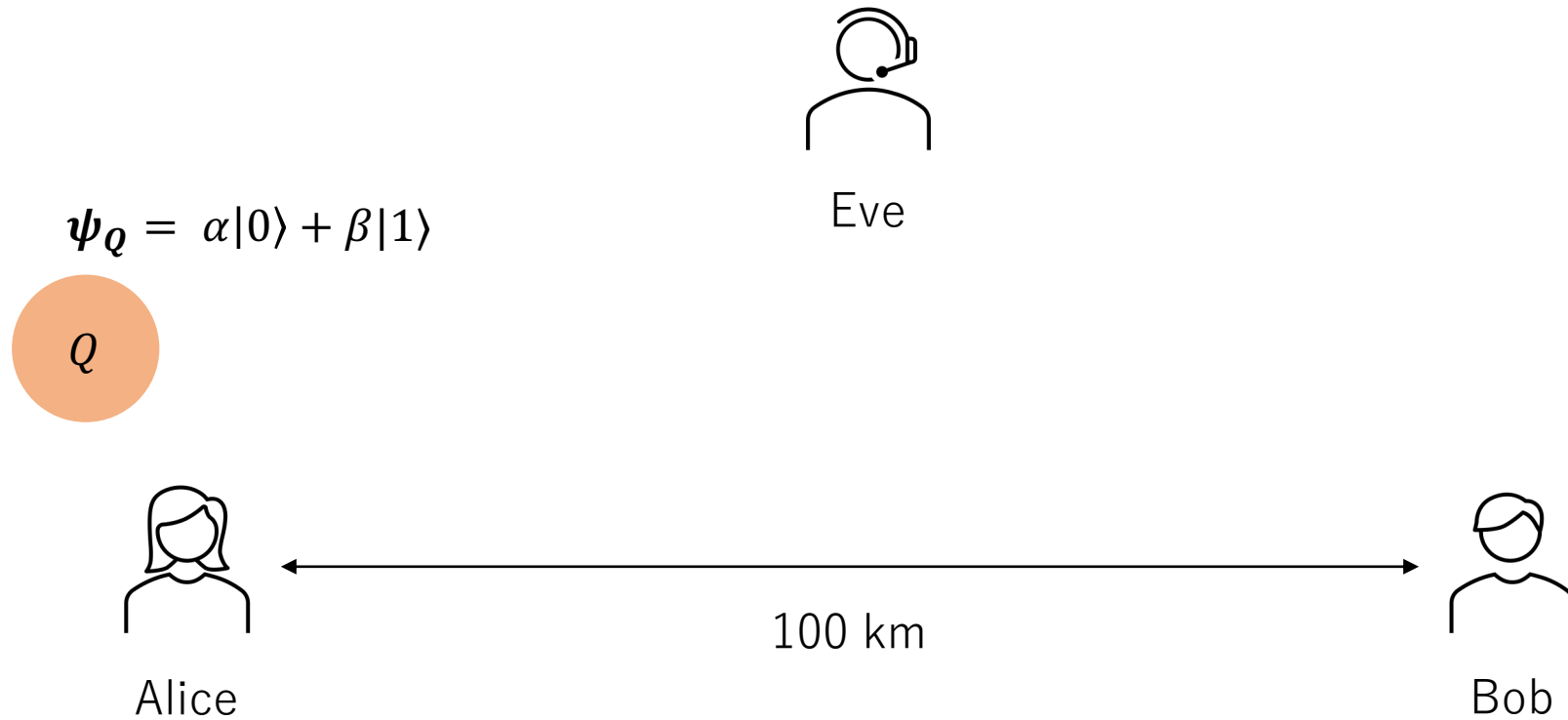
ポイント : Dynamic Circuitにより量子回路の途中で測定が行われ、その測定結果をもとに回路内の量子ゲートを条件付けることができる回路

絵で理解する



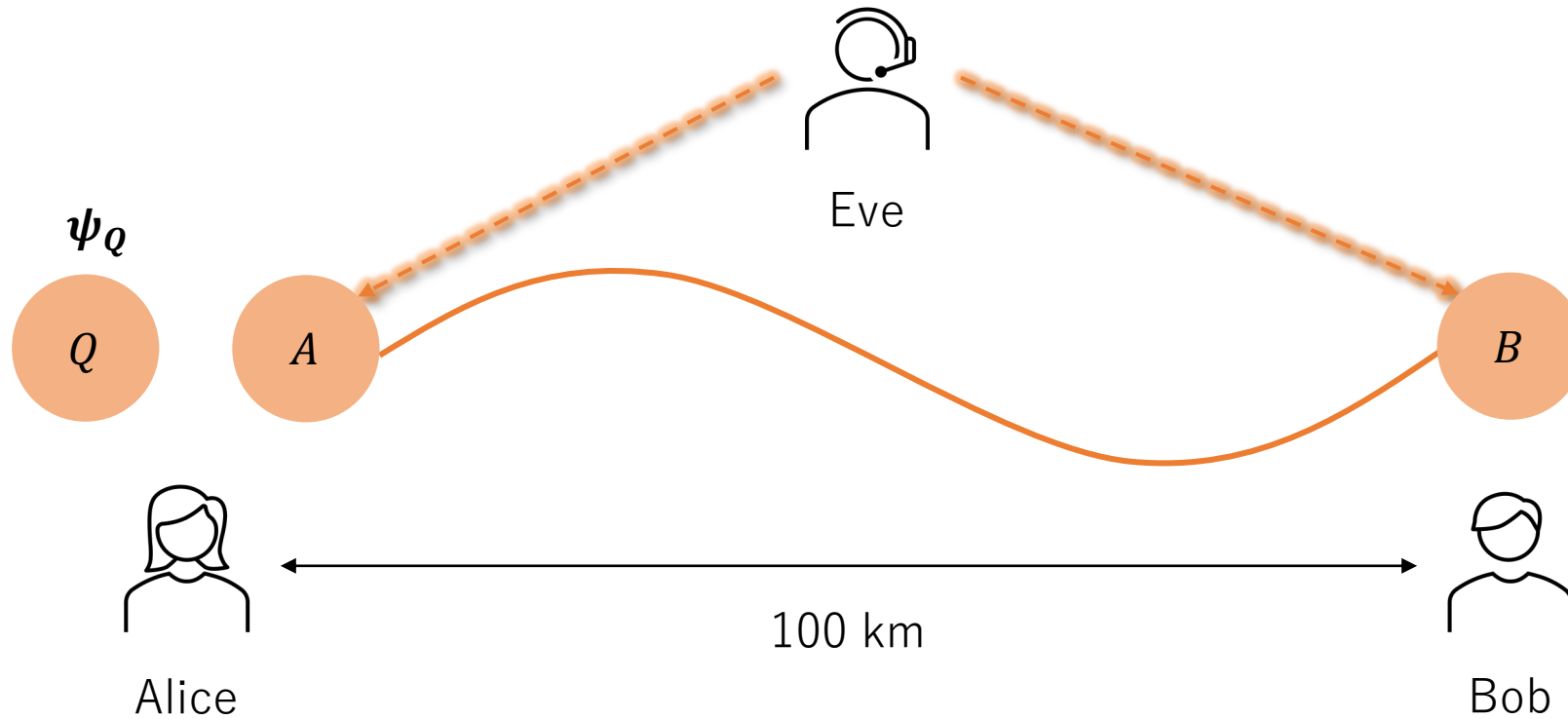
絵で理解する

- 送りたい状態を用意する



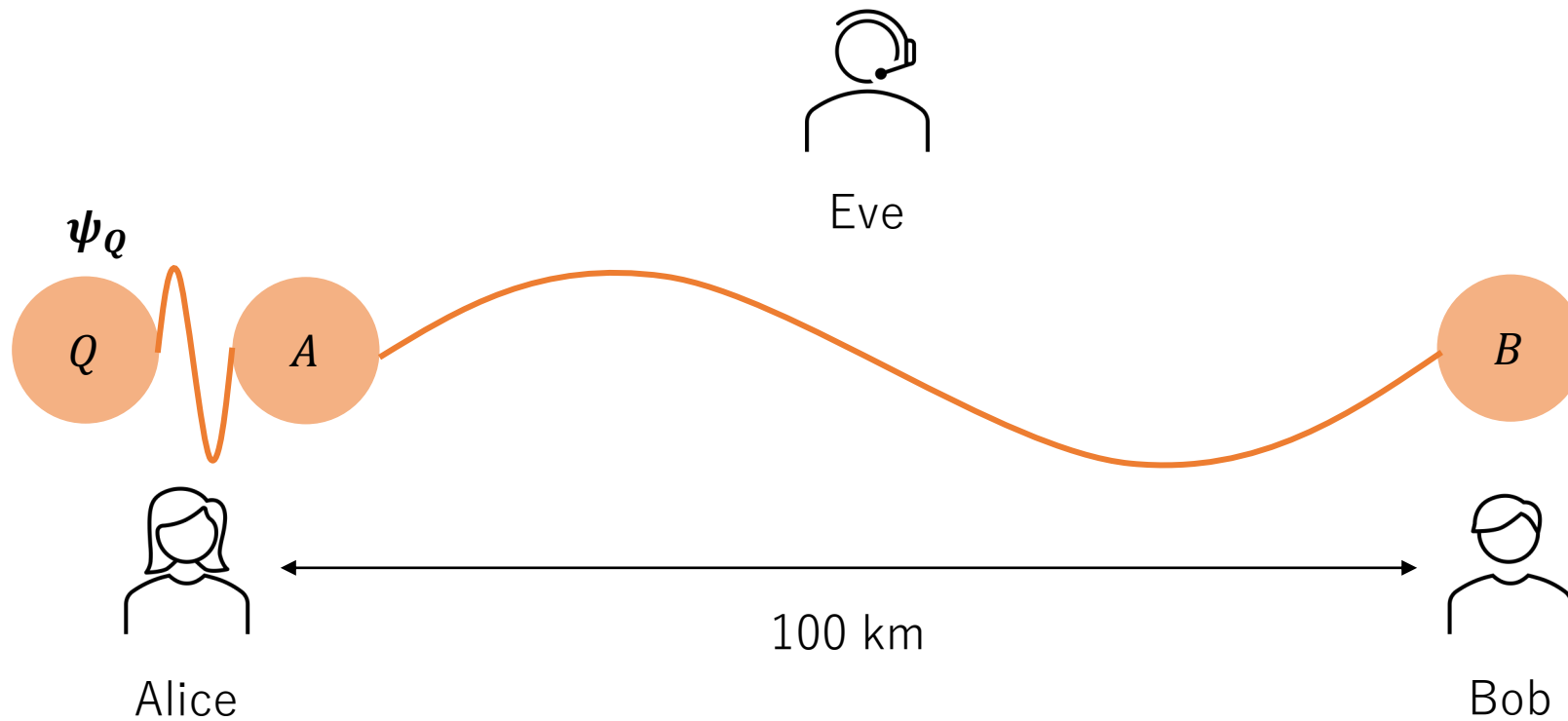
絵で理解する

- AliceとBobにエンタングルメントしたペアを配布



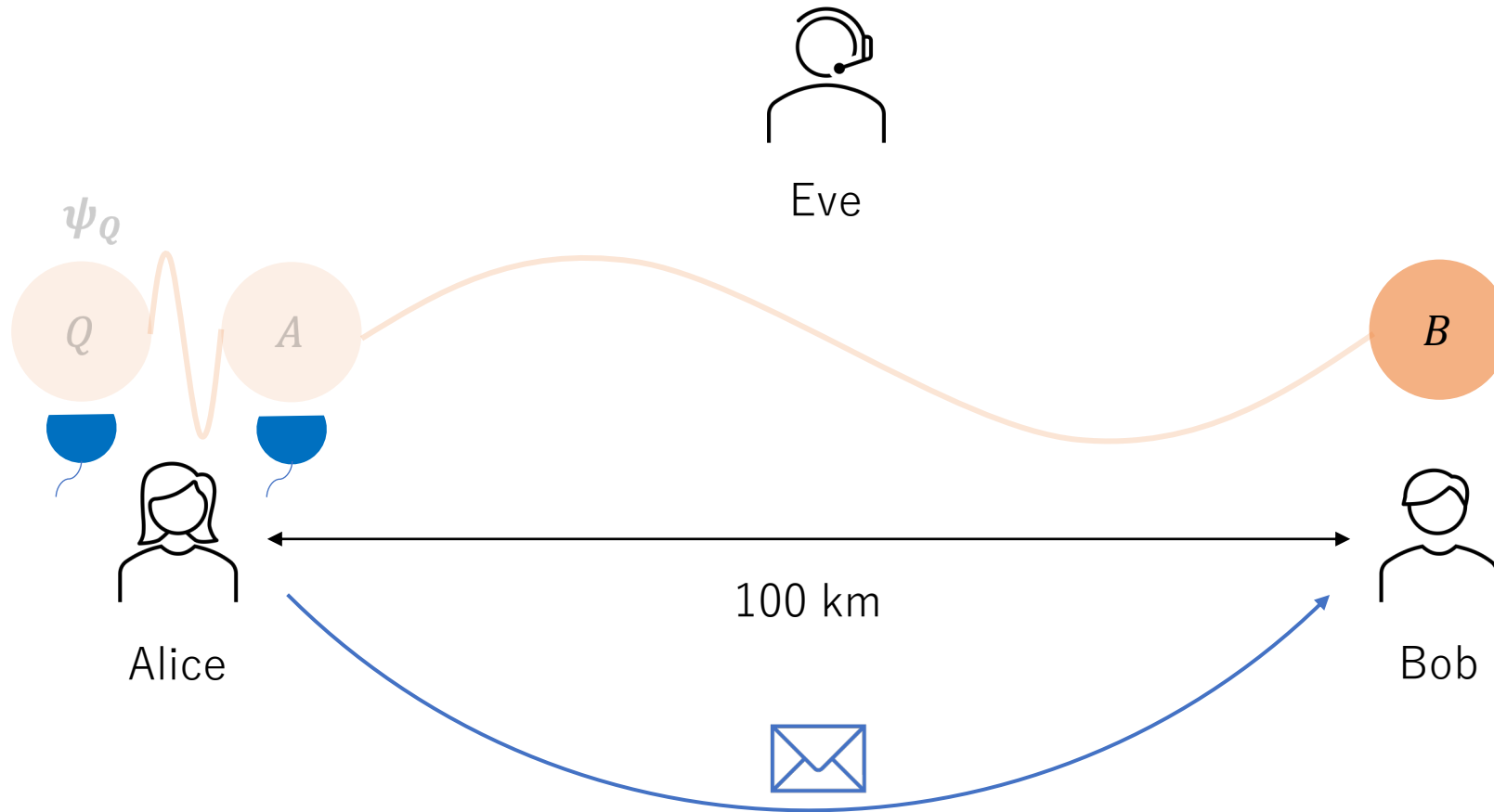
絵で理解する

- Aliceは Q と A をもつれさせる



絵で理解する

- Aliceは Q と A を測定し、結果を古典通信でBobに伝える



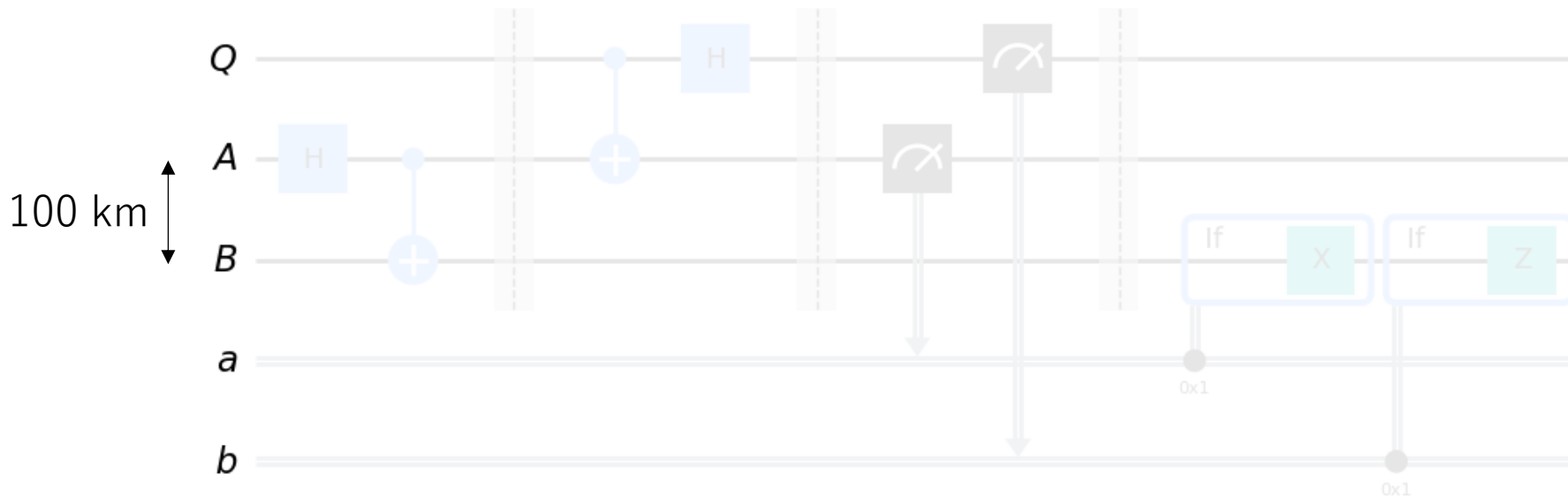
絵で理解する

- Bobは受け取った測定結果をもとに量子操作を行い、 ψ_Q を得る



量子回路と数式で理解する

- 送りたい状態を用意する



```
from qiskit import QuantumCircuit, QuantumRegister
qubit = QuantumRegister(1, "Q")
ebit0 = QuantumRegister(1, "A")
ebit1 = QuantumRegister(1, "B")
a = ClassicalRegister(1, "a")
b = ClassicalRegister(1, "b")

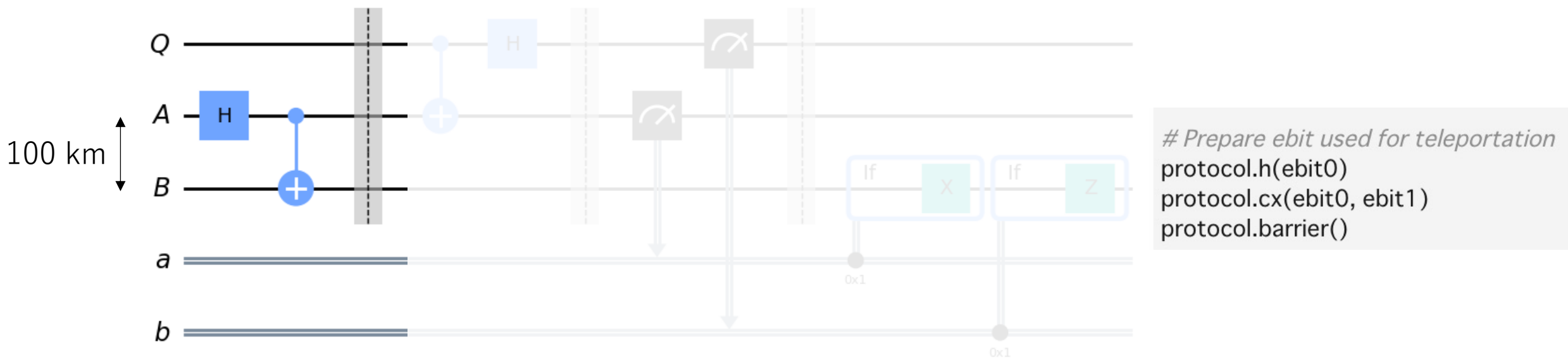
protocol = QuantumCircuit(qubit, ebit0, ebit1, a, b)
```

$$\psi_Q = \alpha|0\rangle + \beta|1\rangle$$

$$\psi_{sys} = |0\rangle_B \otimes |0\rangle_A \otimes (\alpha|0\rangle + \beta|1\rangle)$$

量子回路と数式で理解する

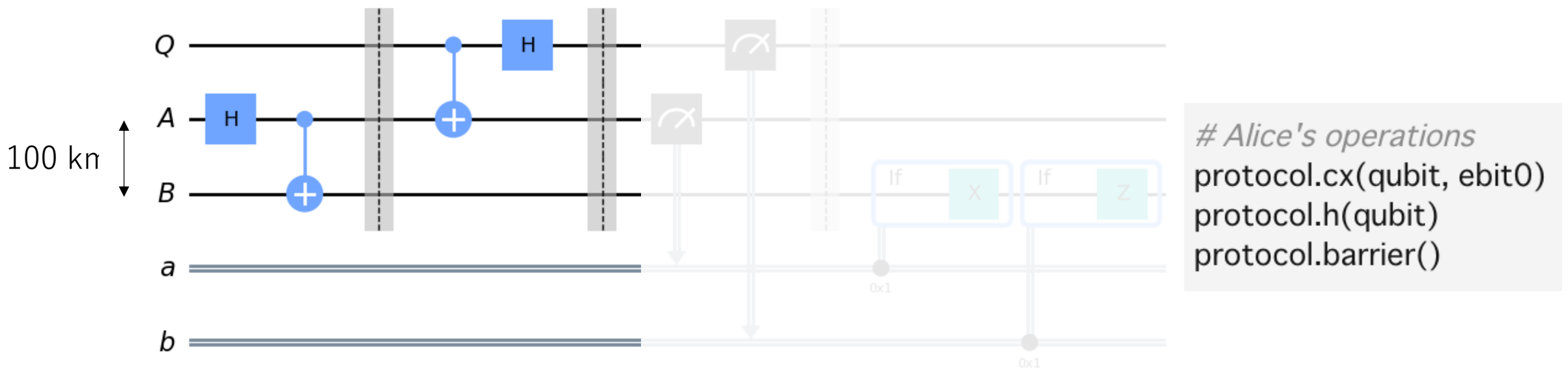
- AliceとBobにエンタングルメントしたペアを配布



$$\begin{aligned}\boldsymbol{\psi}_{sys} &= \frac{1}{\sqrt{2}}(|00\rangle_{BA} + |11\rangle_{BA}) \otimes (\alpha|0\rangle + \beta|1\rangle) \\ &= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle)\end{aligned}$$

量子回路と数式で理解する

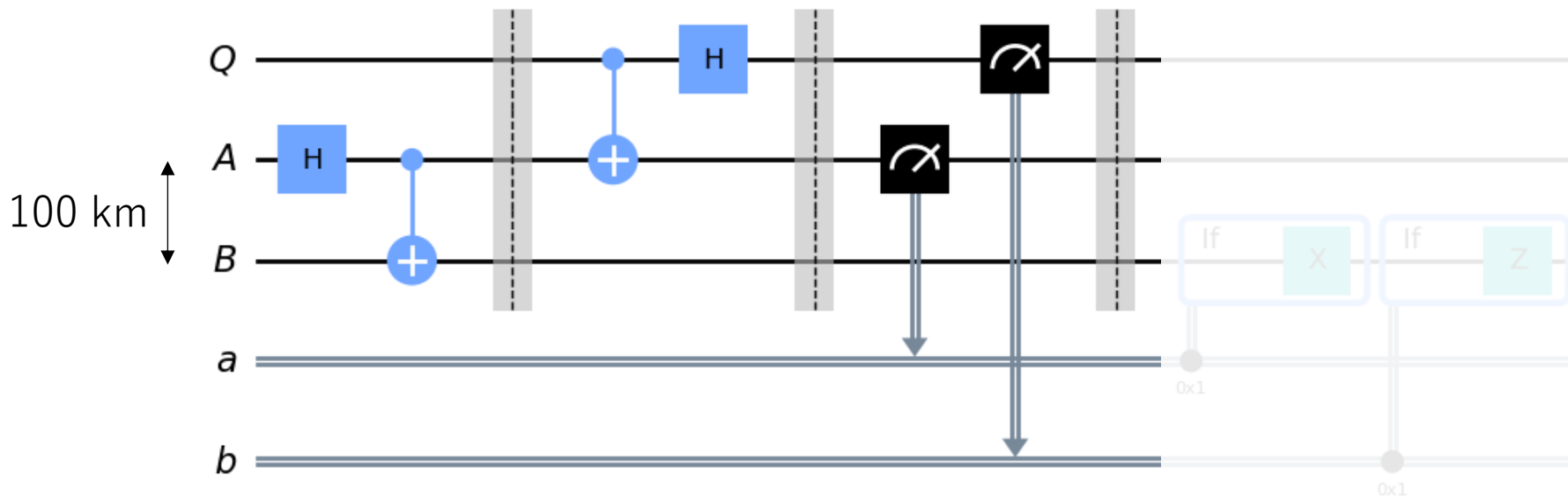
- AliceはQとAをもつれさせる



$$\begin{aligned}
 \psi_{sys} &= (I \otimes I \otimes H)(I \otimes CNOT) \frac{1}{\sqrt{2}} (\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle) \\
 &= (I \otimes I \otimes H) \frac{1}{\sqrt{2}} (\alpha|000\rangle + \alpha|110\rangle + \beta|011\rangle + \beta|101\rangle) \\
 &= \frac{1}{2} (\alpha(|000\rangle + |110\rangle + |001\rangle + |111\rangle) + \beta(|010\rangle + |100\rangle - |011\rangle - |101\rangle)) \\
 &= \frac{1}{2} ((\alpha|0\rangle + \beta|1\rangle)|00\rangle + (\alpha|0\rangle - \beta|1\rangle)|01\rangle + (\alpha|1\rangle + \beta|0\rangle)|10\rangle + (\alpha|1\rangle - \beta|0\rangle)|11\rangle)
 \end{aligned}$$

量子回路と数式で理解する

- AliceはQとAを測定し、結果を古典通信でBobに伝える



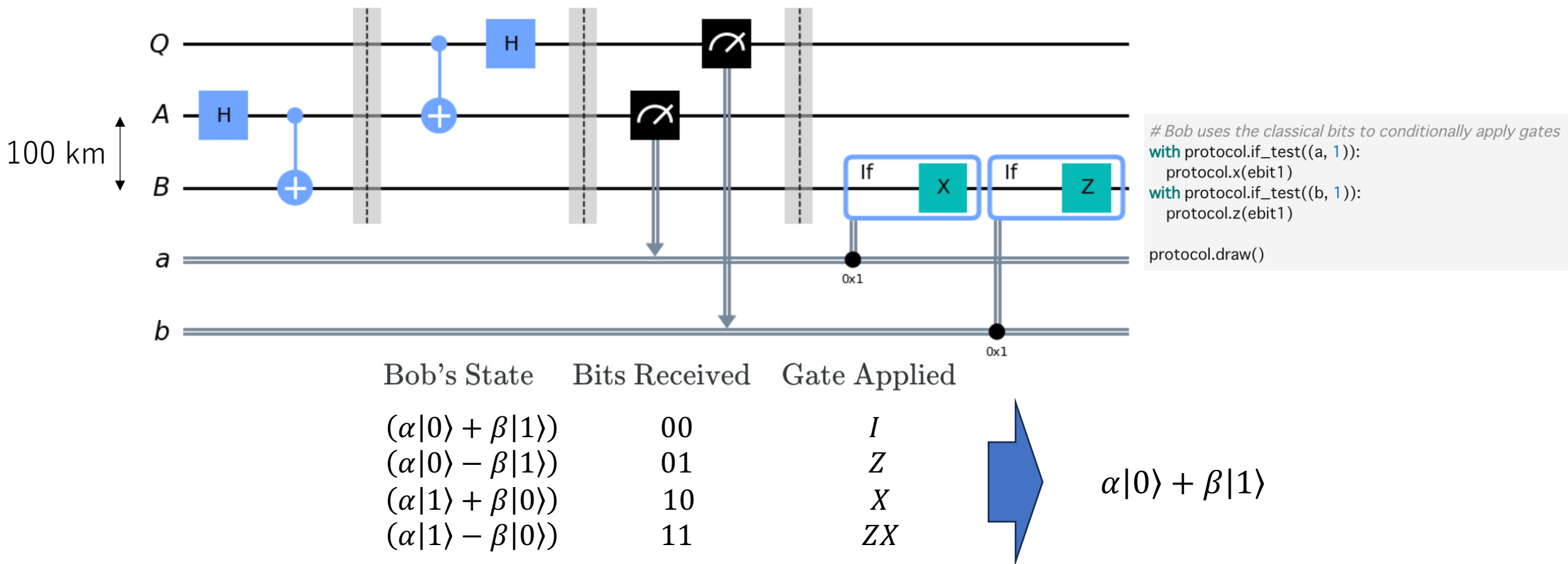
```
# Alice measures and sends classical bits to Bob
protocol.measure(ebit0, a)
protocol.measure(qubit, b)
protocol.barrier()
```

$$\psi_{sys} = \frac{1}{2} ((\alpha|0\rangle + \beta|1\rangle)|00\rangle + (\alpha|0\rangle - \beta|1\rangle)|01\rangle + (\alpha|1\rangle + \beta|0\rangle)|10\rangle + (\alpha|1\rangle - \beta|0\rangle)|11\rangle)$$

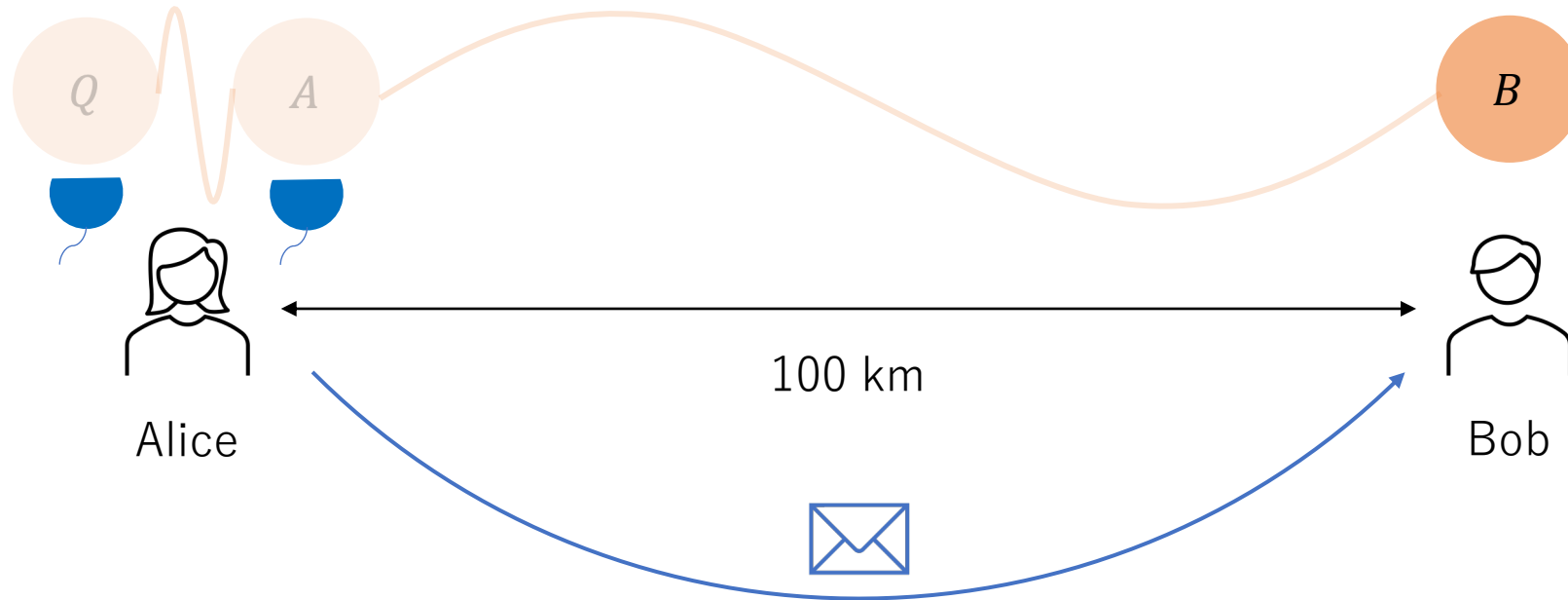
$$\begin{aligned} |00\rangle &\rightarrow (\alpha|0\rangle + \beta|1\rangle) \\ |01\rangle &\rightarrow (\alpha|0\rangle - \beta|1\rangle) \\ |10\rangle &\rightarrow (\alpha|1\rangle + \beta|0\rangle) \\ |11\rangle &\rightarrow (\alpha|1\rangle - \beta|0\rangle) \end{aligned}$$

量子回路と数式で理解する

- Bobは受け取った測定結果をもとに量子操作を行い、 ψ_Q を得る



量子テレポーテーションに対するダメ出し



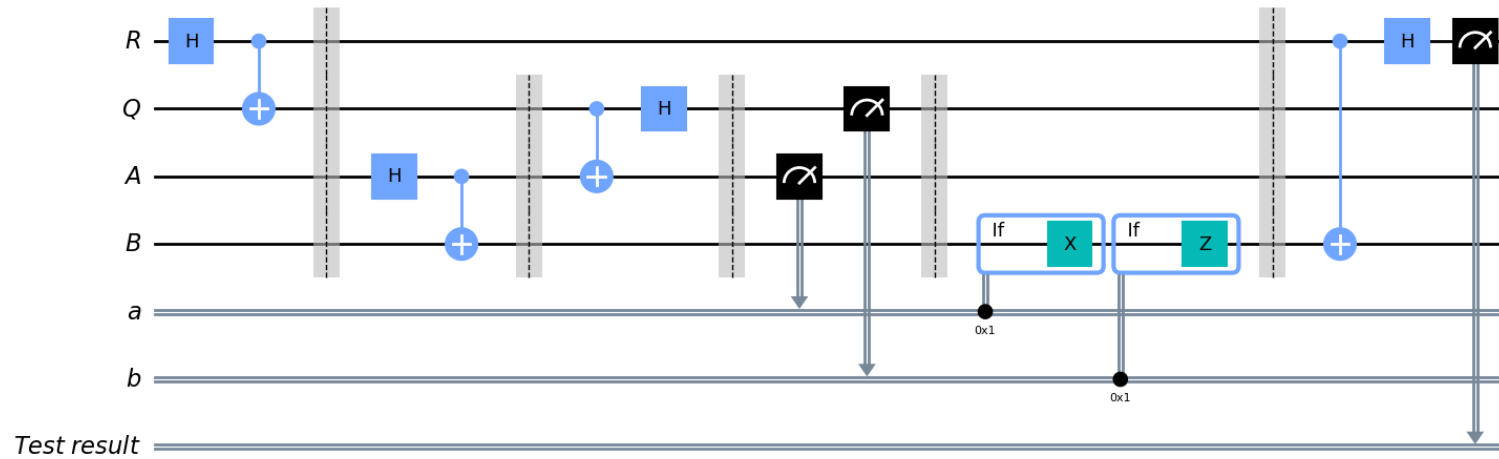
- 古典通信を用いており、光速を上回らないので遅い
- 1量子ビットの情報を送るために2量子ビットのエンタングルメントを使わなければならない

2 | 量子テレポーテーションの動作を検証する

扱う検証用の量子回路

方針：エンタングルメントしたペアの片割れを量子テレポーテーションして検証する

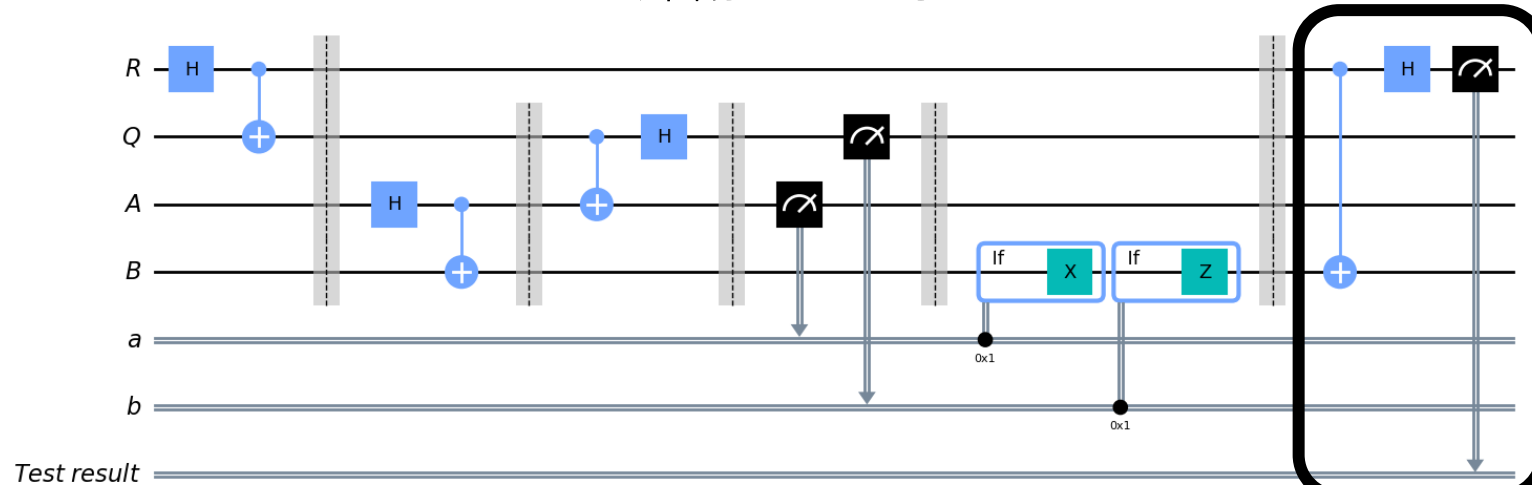
Textbook



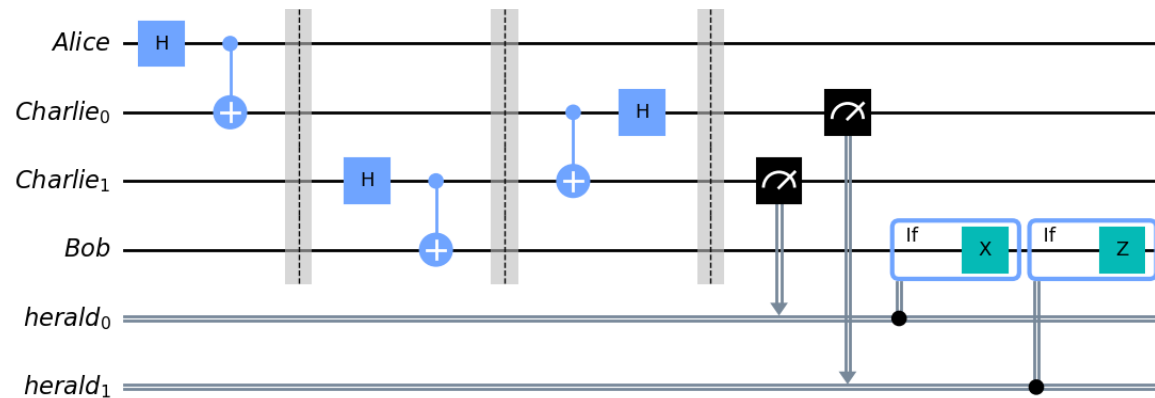
扱う検証用の量子回路

方針：エンタングルメントしたペアの片割れを量子テレポーテーションして検証する

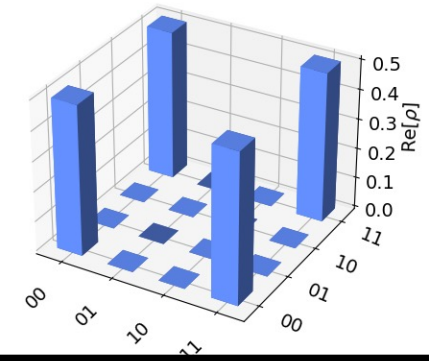
Textbook



本勉強会の回路



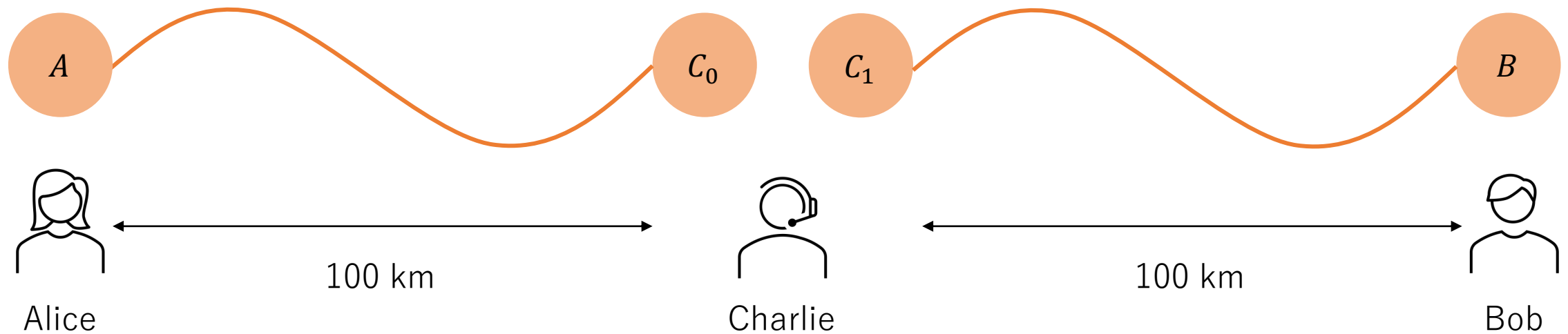
Quantum State Tomography



ポイント：量子中継を行う回路としてみることもできる！

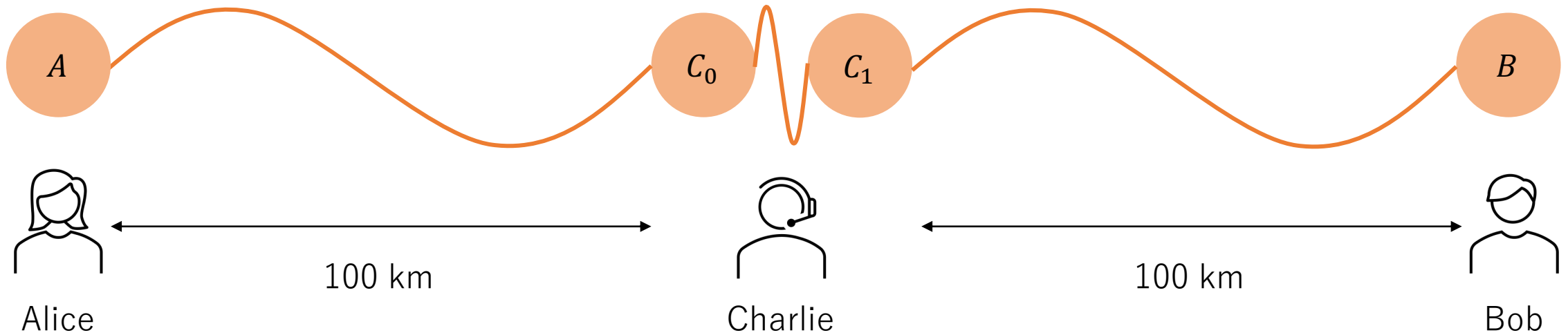
絵で理解する

- AliceとCharlie間、BobとCharlie間でエンタングルしたペアを配布



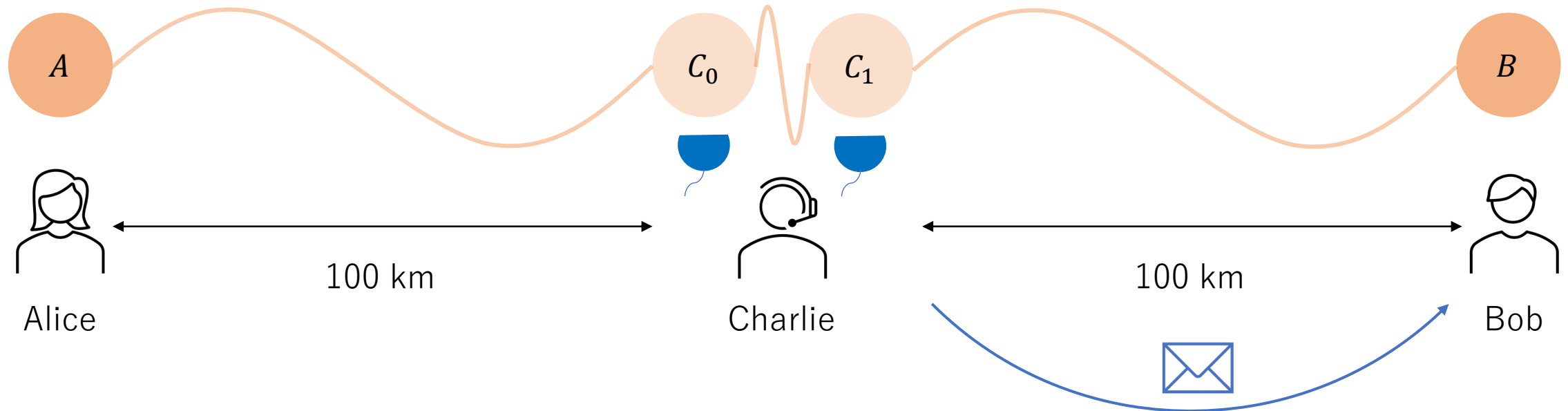
絵で理解する

- C_0 と C_1 をもつれさせる



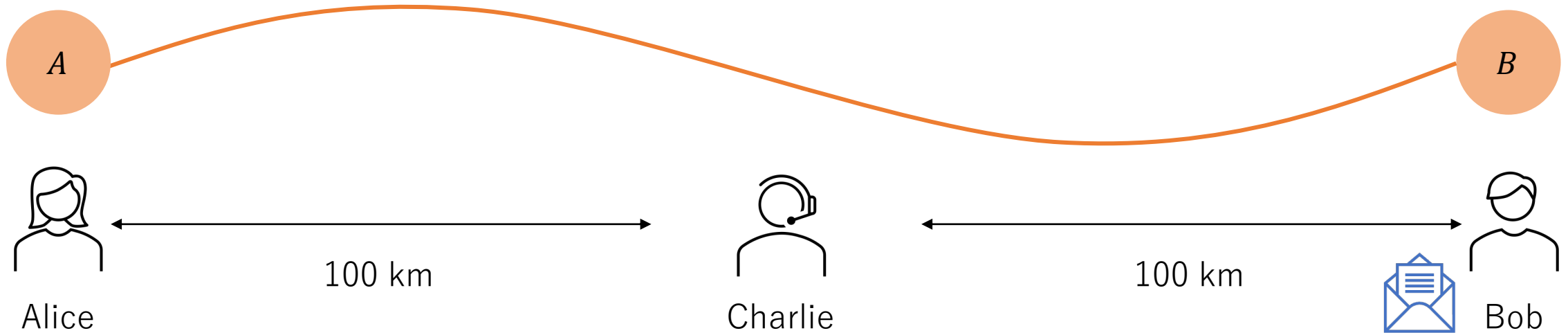
絵で理解する

- C_0 と C_1 を測定し結果を伝える



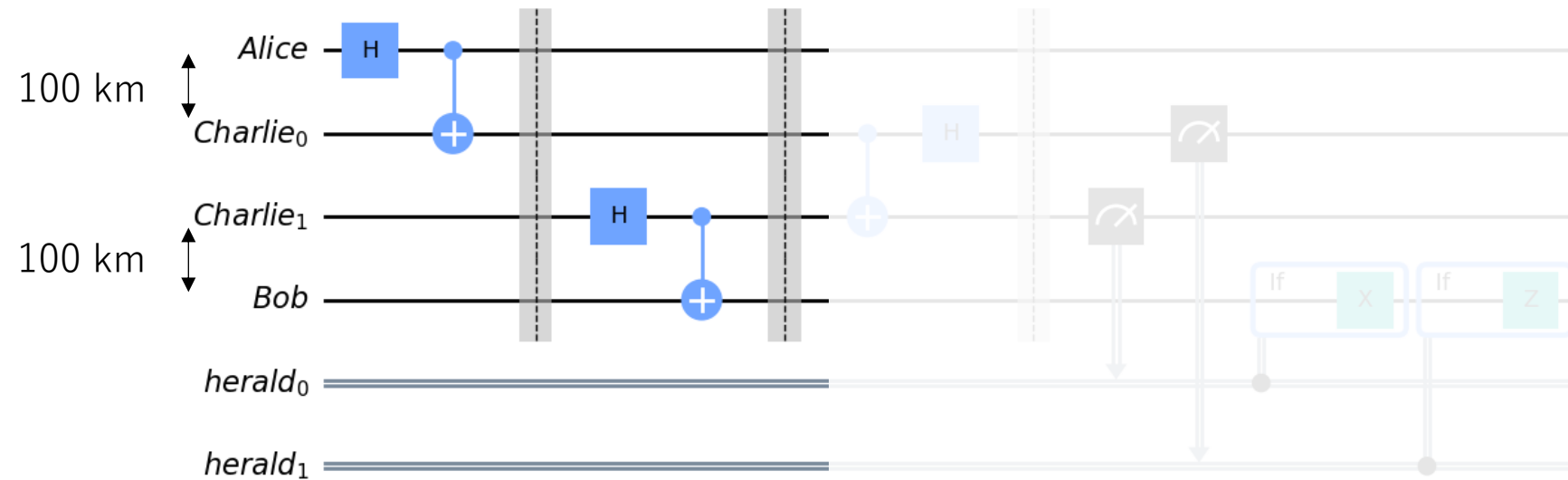
絵で理解する

- AliceとBobでエンタングルしたペアを共有



量子回路と数式で理解する

AliceとCharlie間、BobとCharlie間でエンタングルしたペアを配布



```
A = QuantumRegister(1, "Alice")
C = QuantumRegister(2, "Charlie")
B = QuantumRegister(1, "Bob")
herald = ClassicalRegister(2, 'herald')
qr = QuantumCircuit(A, C, B, herald)
```

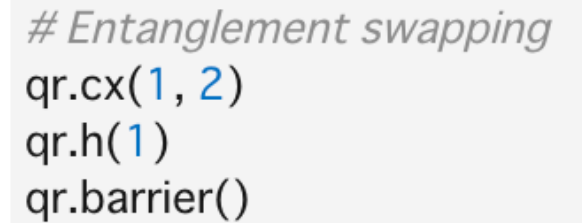
Creating bell-pairs

```
qr.h(0)
qr.cx(0, 1)
qr.barrier()
```

```
qr.h(2)
qr.cx(2, 3)
qr.barrier()
```

$$\begin{aligned}\psi_{sys} &= \frac{1}{2}(|00\rangle_{BC_1} + |11\rangle_{BC_1}) \otimes (|00\rangle_{C_0A} + |11\rangle_{C_0A}) \\ &= \frac{1}{2}(|0000\rangle + |1100\rangle + |0011\rangle + |1111\rangle)\end{aligned}$$

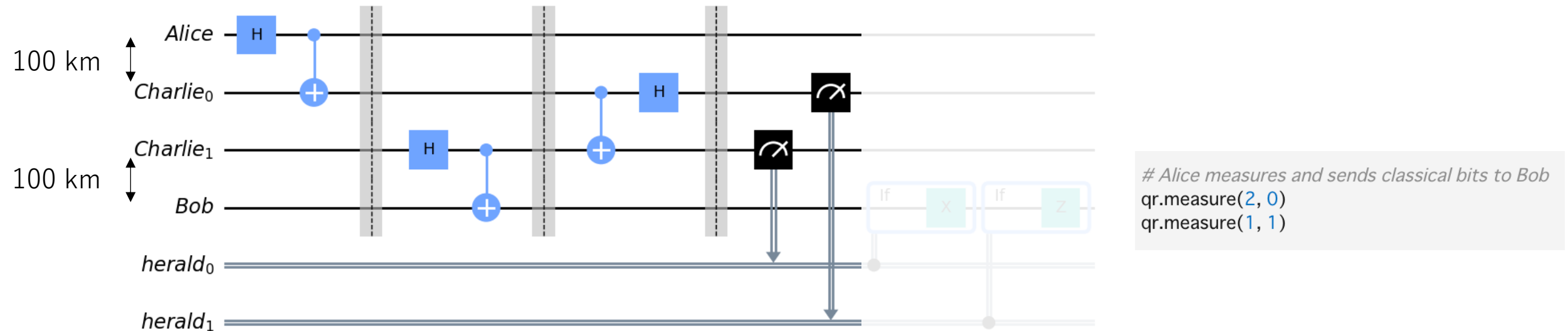
C_0 と C_1 をもつれさせる



$$\begin{aligned}
\psi_{sys} &= (I \otimes I \otimes H \otimes I)(I \otimes CNOT \otimes I) \frac{1}{2} (|0000\rangle + |1100\rangle + |0011\rangle + |1111\rangle) \\
&= (I \otimes I \otimes H \otimes I) (\frac{1}{2} (|0000\rangle + |1100\rangle + |0111\rangle + |1011\rangle)) \\
&= \frac{1}{2\sqrt{2}} ((|0000\rangle + |0010\rangle + |1100\rangle + |1110\rangle) + (|0101\rangle - |0111\rangle + |1001\rangle - |1011\rangle)) \\
&= \frac{1}{2\sqrt{2}} ((|00\rangle_{BA} + |11\rangle_{BA})|00\rangle_{C_1C_0} + (|00\rangle_{BA} - |11\rangle_{BA})|01\rangle_{C_1C_0} + (|01\rangle_{BA} + |10\rangle_{BA})|10\rangle_{C_1C_0} + (|10\rangle_{BA} - |01\rangle_{BA})|11\rangle_{C_1C_0})
\end{aligned}$$

量子回路と数式で理解する

C_0 と C_1 を測定し結果を伝える



$$\psi_{\text{sys}} = \frac{1}{2\sqrt{2}} (|00\rangle_{BA} + |11\rangle_{BA})|00\rangle_{C_1C_0} + (|00\rangle_{BA} - |11\rangle_{BA})|01\rangle_{C_1C_0} + (|01\rangle_{BA} + |10\rangle_{BA})|10\rangle_{C_1C_0} + (|10\rangle_{BA} - |01\rangle_{BA})|11\rangle_{C_1C_0})$$

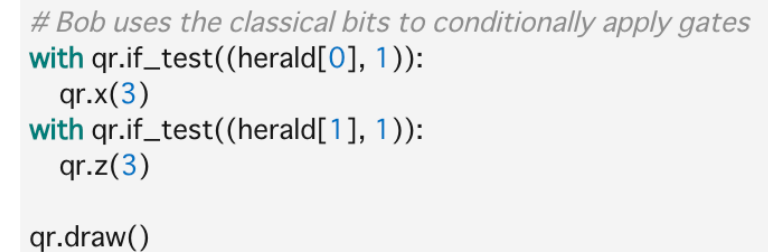
$$|00\rangle \rightarrow \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

$$|01\rangle \rightarrow \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle)$$

$$|10\rangle \rightarrow \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)$$

$$|11\rangle \rightarrow \frac{1}{\sqrt{2}} (|10\rangle - |01\rangle)$$

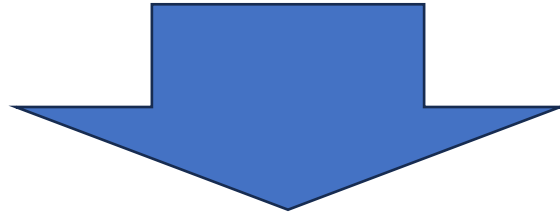
AliceとBobでエンタングルしたペアを共有



$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

プロトコルの動作検証

AliceとBobが $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ を正しく共有できているかを確認したい



実際の実験で行う手法、**量子状態トモグラフィー(QST)**を実行し、**フィデリティ**を見してみる

量子状態トモグラフィー(QST): 量子状態を推定するために、密度行列を構築する手法

フィデリティ: 量子状態が理想的なものとどれだけ近いか測る指標

準備

理想的な状態(Bell state ϕ^+)を用意する

```
pip install qiskit_experiments
```

```
from qiskit.visualization import plot_histogram, plot_state_city
from qiskit_experiments.library.tomography import TomographyExperiment
from qiskit_experiments.library import StateTomography
from qiskit.quantum_info import state_fidelity, DensityMatrix
import numpy as np
```

```
circ = QuantumCircuit(2)
```

```
circ.h(0)
circ.cx(0, 1)
```

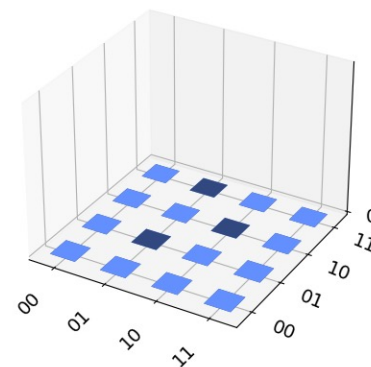
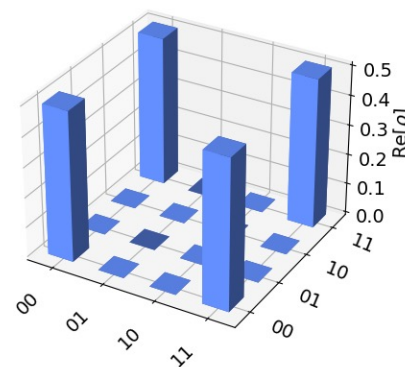
```
phi_plus = DensityMatrix.from_instruction(circ)
```

```
plot_state_city(phi_plus, title='Density Matrix')
```

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \frac{1}{\sqrt{2}}(\langle 00| + \langle 11|) = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

```
DensityMatrix([[0.5+0.j, 0. +0.j, 0. +0.j, 0.5+0.j],
               [0. +0.j, 0. +0.j, 0. +0.j, 0. +0.j],
               [0. +0.j, 0. +0.j, 0. +0.j, 0. +0.j],
               [0.5+0.j, 0. +0.j, 0. +0.j, 0.5+0.j]],
               dims=(2, 2))
```

Density Matrix



実機でQSTを実行

```
from qiskit.providers.ibmq import least_busy
from qiskit.tools.monitor import job_monitor
from qiskit_ibm_provider import IBMProvider

provider = IBMProvider()
backend = least_busy(provider.backends(filters=lambda b: b.configuration().n_qubits >= 4 and
                                             not b.configuration().simulator and b.status().operational==True))

print(backend)
```

```
<IBMBBackend('ibm_lagos')>
```

```
qstexp1 = StateTomography(qr, measurement_indices=[0, 3], target=phi_plus)
qstdata1 = qstexp1.run(backend).block_for_results()
```

```
# Print results
```

```
for result in qstdata1.analysis_results():
    print(result)
```

```
plot_state_city(qstdata1.analysis_results("state").value, title='Density Matrix')
```

実機でQSTを実行

```
from qiskit.providers.ibmq import least_busy
from qiskit.tools.monitor import job_monitor
from qiskit_ibm_provider import IBMProvider
```

```
provider = IBMProvider()
backend = least_busy(provider.backends(filters=lambda b: b.configuration().n_qubits >= 4 and
                                         not b.configuration().simulator and b.status().operational==True))
```

```
print(backend)
```

```
<IBMQBackend('ibmq_lagos')>
```

```
qstexp1 = StateTomography(qr, measurement_indices=[0, 3], target=phi_plus)
qstdata1 = qstexp1.run(backend).block_for_results()
```

```
# Print results
```

```
for result in qstdata1.analysis_results():
    print(result)
```

```
plot_state_city(qstdata1.analysis_results("state").value, title='Density Matrix')
```

AnalysisResult

- name: state

- value: DensityMatrix([[0.446375 +0.j , 0.01354167-0.01195833j,
-0.00554167-0.03229167j, 0.17275 -0.059375j],
[0.01354167+0.01195833j, 0.04829167+0.j ,
0.00325 +0.046625j , 0.00070833+0.03545833j],
[-0.00554167+0.03229167j, 0.00325 -0.046625j ,
0.06920833+0.j , -0.01370833+0.01054167j],
[0.17275 +0.059375j , 0.00070833-0.03545833j,
-0.01370833-0.01054167j, 0.436125 +0.j]],
dims=(2, 2))

- extra: <7 items>

- device_components: ['Q0', 'Q1', 'Q2', 'Q3']

- verified: False

AnalysisResult

- name: state_fidelity

- value: 0.6140000143467915

- device_components: ['Q0', 'Q1', 'Q2', 'Q3']

- verified: False

AnalysisResult

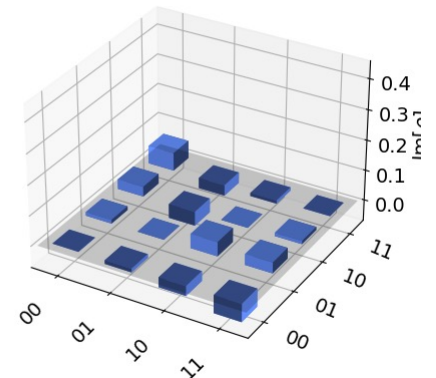
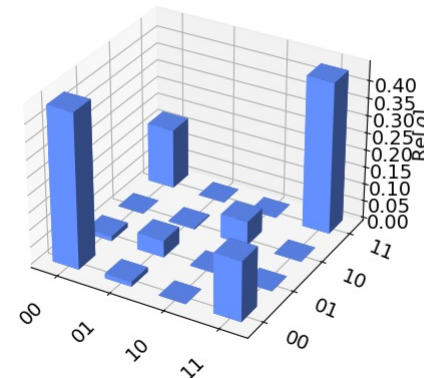
- name: positive

- value: True

- device_components: ['Q0', 'Q1', 'Q2', 'Q3']

- verified: False

Density Matrix



シミュレータでQSTを実行する

```
# QST Experiment
from qiskit_aer import AerSimulator
from qiskit.providers.fake_provider import FakeAuckland

backend = AerSimulator.from_backend(FakeAuckland())

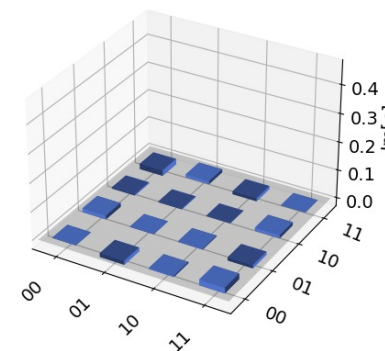
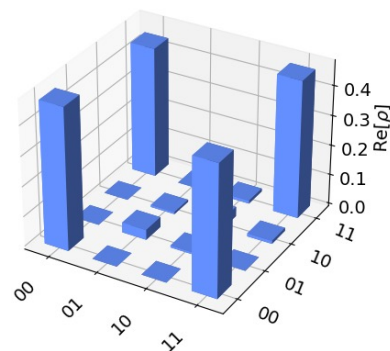
qstexp1 = StateTomography(qr, measurement_indices=[0, 3], target=phi_plus)
# qstdata1 = AerSimulator().run(qstexp1).block_for_results()
qstdata1 = qstexp1.run(backend, seed_simulation=100).block_for_results()

# Print results
for result in qstdata1.analysis_results():
    print(result)
```

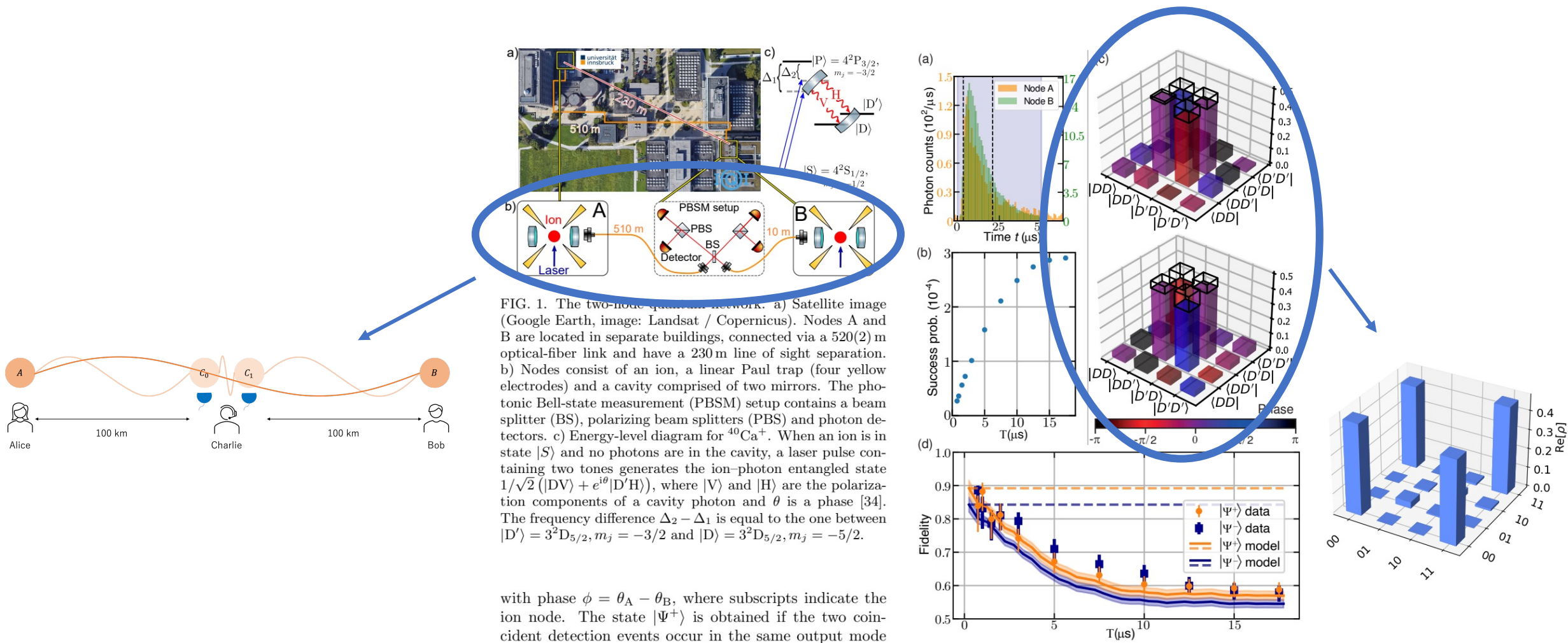
```
AnalysisResult
- name: state
- value: DensityMatrix([[ 0.47403216+0.00000000e+00j, -0.01664845-1.34269239e-02j,
-0.01338765+4.08639308e-03j, 0.43794811+1.98741245e-02j],
[-0.01664845+1.34269239e-02j, 0.03342388+4.33680869e-19j,
0.00689508+2.53669687e-03j, 0.00054993-1.10163642e-02j],
[-0.01338765-4.08639308e-03j, 0.00689508-2.53669687e-03j,
0.03211852+0.00000000e+00j, 0.01021257+1.12824832e-02j],
[ 0.43794811-1.98741245e-02j, 0.00054993+1.10163642e-02j,
0.01021257-1.12824832e-02j, 0.46042544+0.00000000e+00j]],
dims=(2, 2))
- extra: <7 items>
- device_components: ['Q0', 'Q1', 'Q2', 'Q3']
- verified: False
AnalysisResult
- name: state_fidelity
- value: 0.9051769338847774
- device_components: ['Q0', 'Q1', 'Q2', 'Q3']
- verified: False
AnalysisResult
- name: positive
- value: True
- device_components: ['Q0', 'Q1', 'Q2', 'Q3']
- verified: False
```

```
plot_state_city(qstdata1.analysis_results("state").value, title='Density Matrix')
```

AliceとBobが $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ を正しく共有できている！



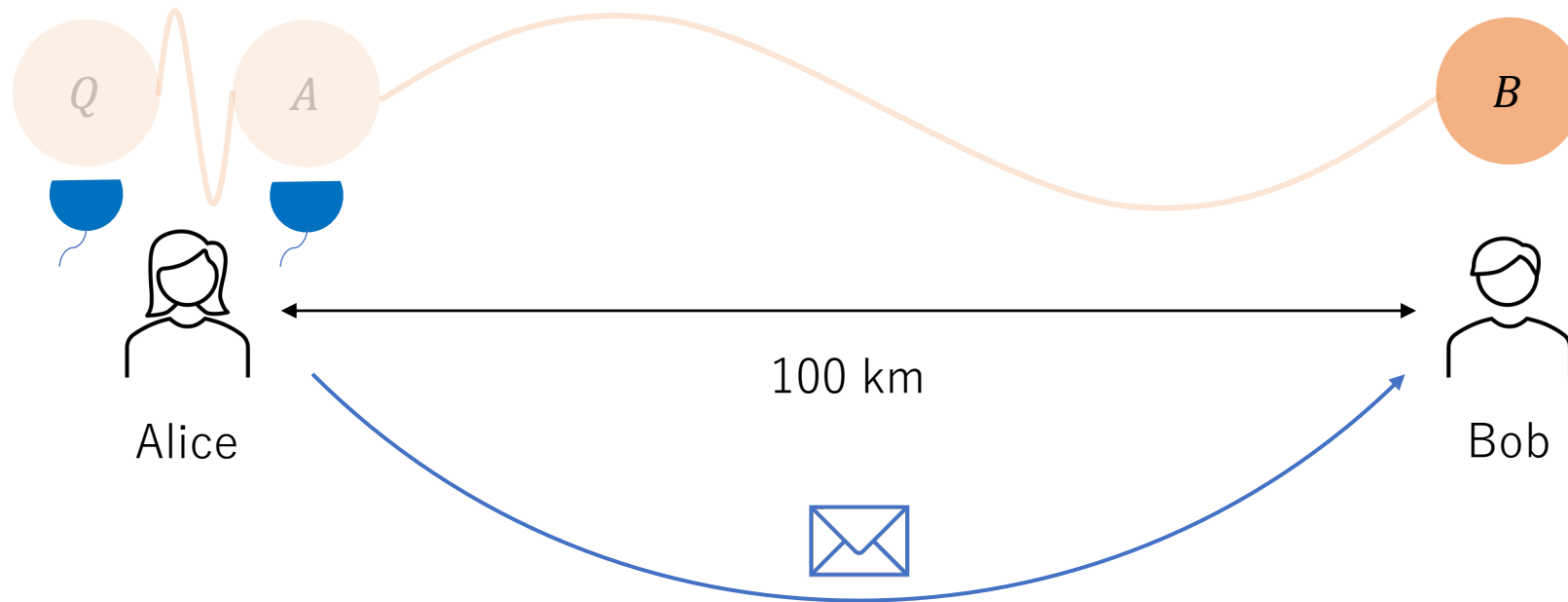
最先端の量子中継の研究も同じ流れ



Entanglement of Trapped-Ion Qubits Separated by 230 Meters

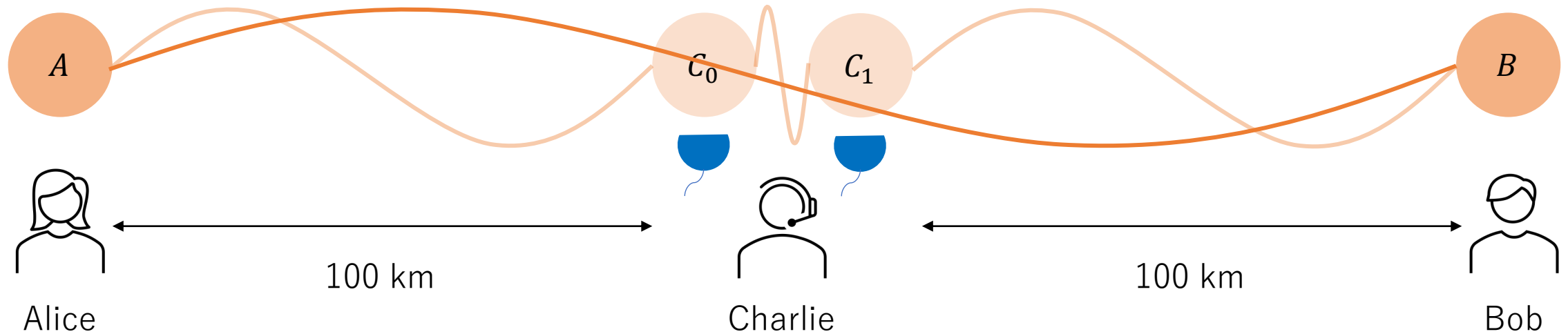
V. Krutyanskiy, M. Galli, V. Krcmarsky, S. Baier, D. A. Fioretto, Y. Pu, A. Mazloom, P. Sekatski, M. Canteri, M. Teller, J. Schupp, J. Bate, M. Meraner, N. Sangouard, B. P. Lanyon, and T. E. Northup, Phys. Rev. Lett. **130**, 050803 – Published 2 February 2023

量子テレポーテーションに対するダメ出し



- 古典通信を用いており、光速を上回らないので遅い
- 1量子ビットの情報を送るために2量子ビットのエンタングルメントを使わなければならない

量子テレポーテーションの仕組みを利用して嬉しいこと。



- セキュリティの観点で見ると複製不可能定理で盗聴が難しい
- エンタングルメントを利用した長距離量子通信によって分散量子計算や量子インターネットが可能になる

本日勉強したこと

- 量子テレポーテーションを理解する
 - 言葉で理解する
 - 量子回路と数式で理解する
 - Dynamic Circuitも活用した
- 量子テレポーテーションの動作を検証する
 - エンタングルメントしたペアの片割れを量子テレポーテーションして検証する
 - 最新の量子通信手法「量子中継」に入門する
 - 量子状態トモグラフィを実行し、フィデリティを見てみた

参考文献

- 量子テレポーテーション
 - <https://ja.learn.qiskit.org/course/basics/entanglement-in-action#entanglement-2-0>
- Quantum State Tomography
 - https://qiskit.org/ecosystem/experiments/manuals/verification/state_tomography.html
- Fake Provider
 - https://qiskit.org/documentation/apidoc/providers_fake_provider.html
- Entanglement of Trapped-Ion Qubits Separated by 230 Meters
 - <https://arxiv.org/abs/2208.14907>