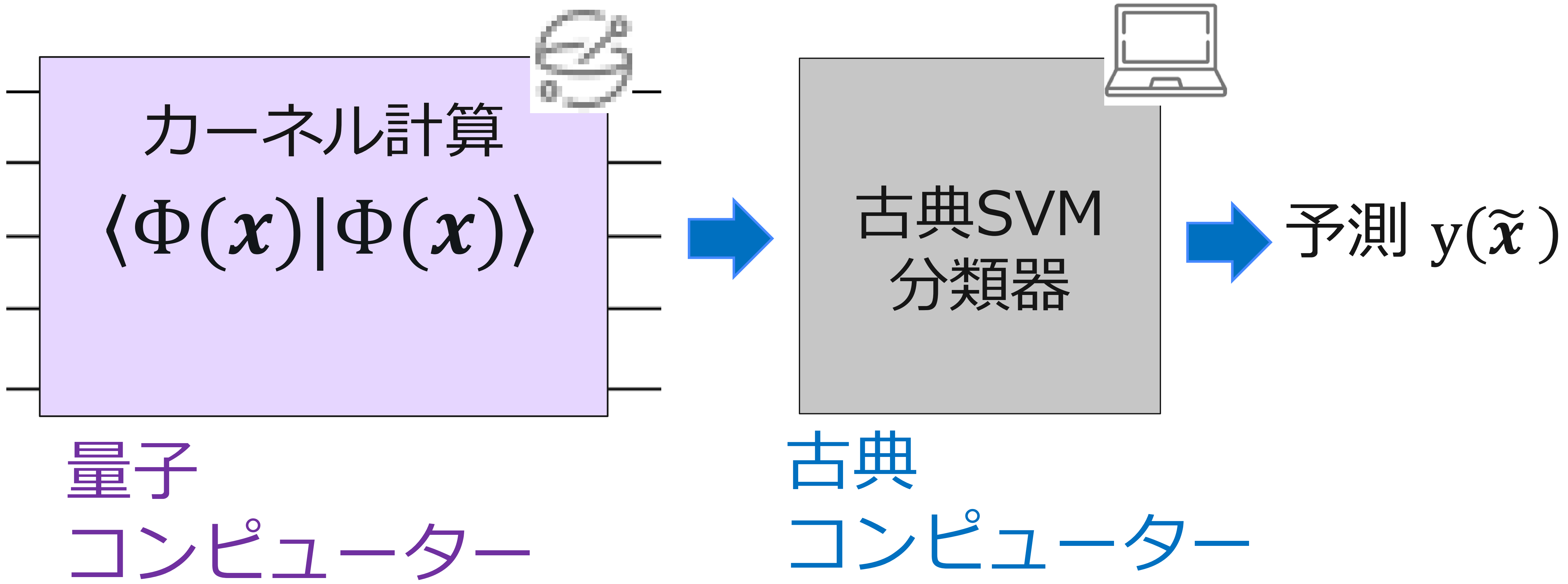
量子機械学習

Jan 20, 2025

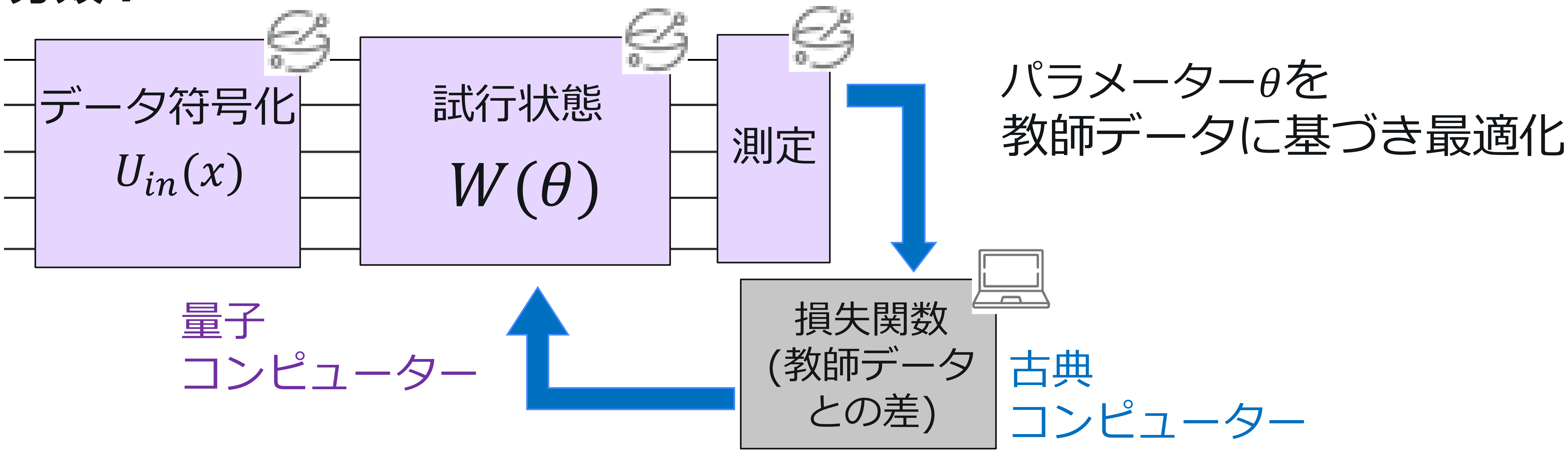
沼田祈史
Kifumi Numata
IBM Quantum

量子機械学習による分類で有名な手法は主に２種

量子カーネル分類：

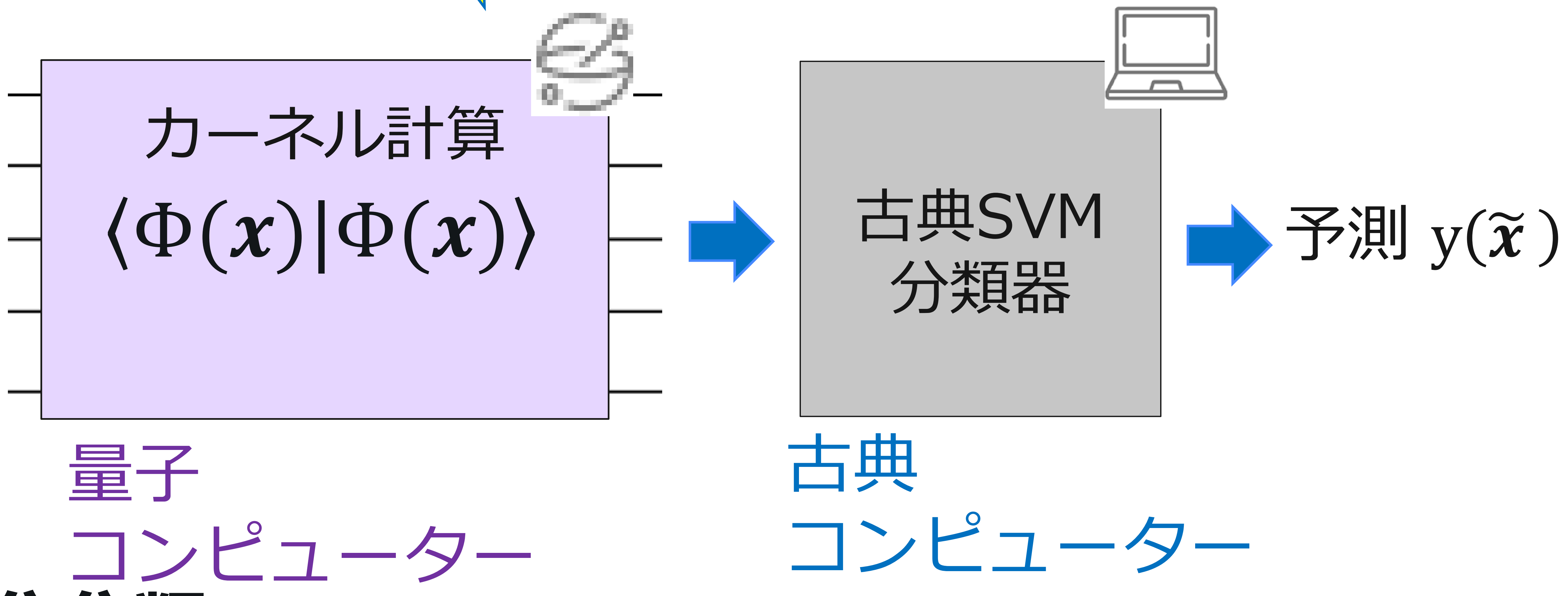


量子変分分類：

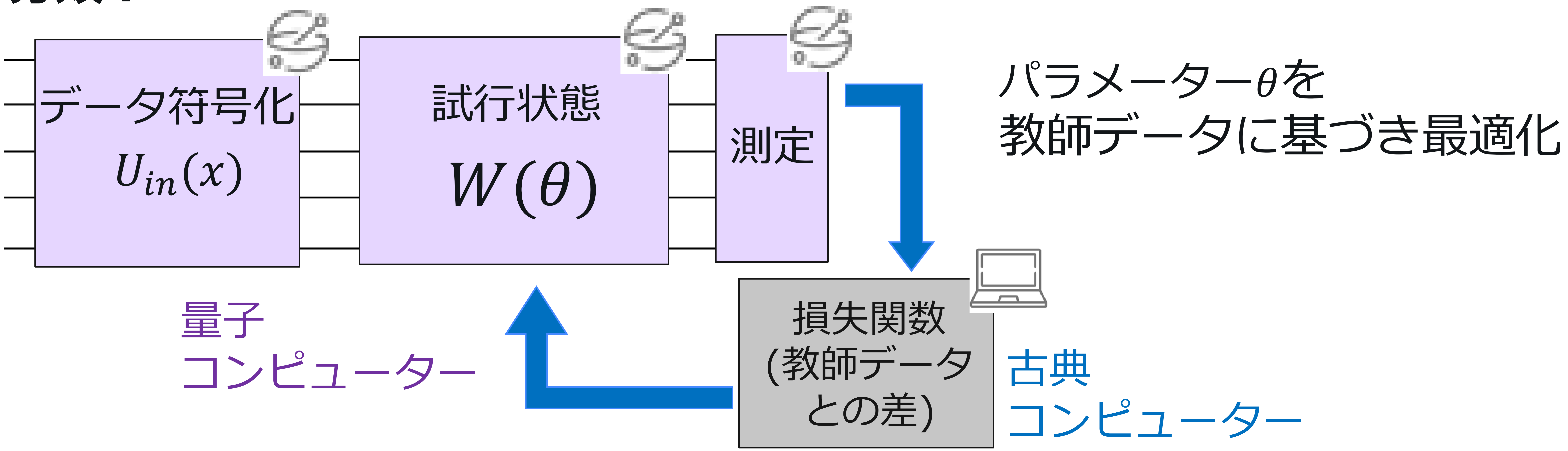


量子機械学習による分類で有名な手法は主に２種

量子カーネル分類：



量子変分分類：



最初に、皆さんの期待値を調整させてください！

量子機械学習アルゴリズムが、従来の機械学習アルゴリズムよりも効率的かつスケーラブルにすべての分類タスクを解決できるというのは
真実ではありません。

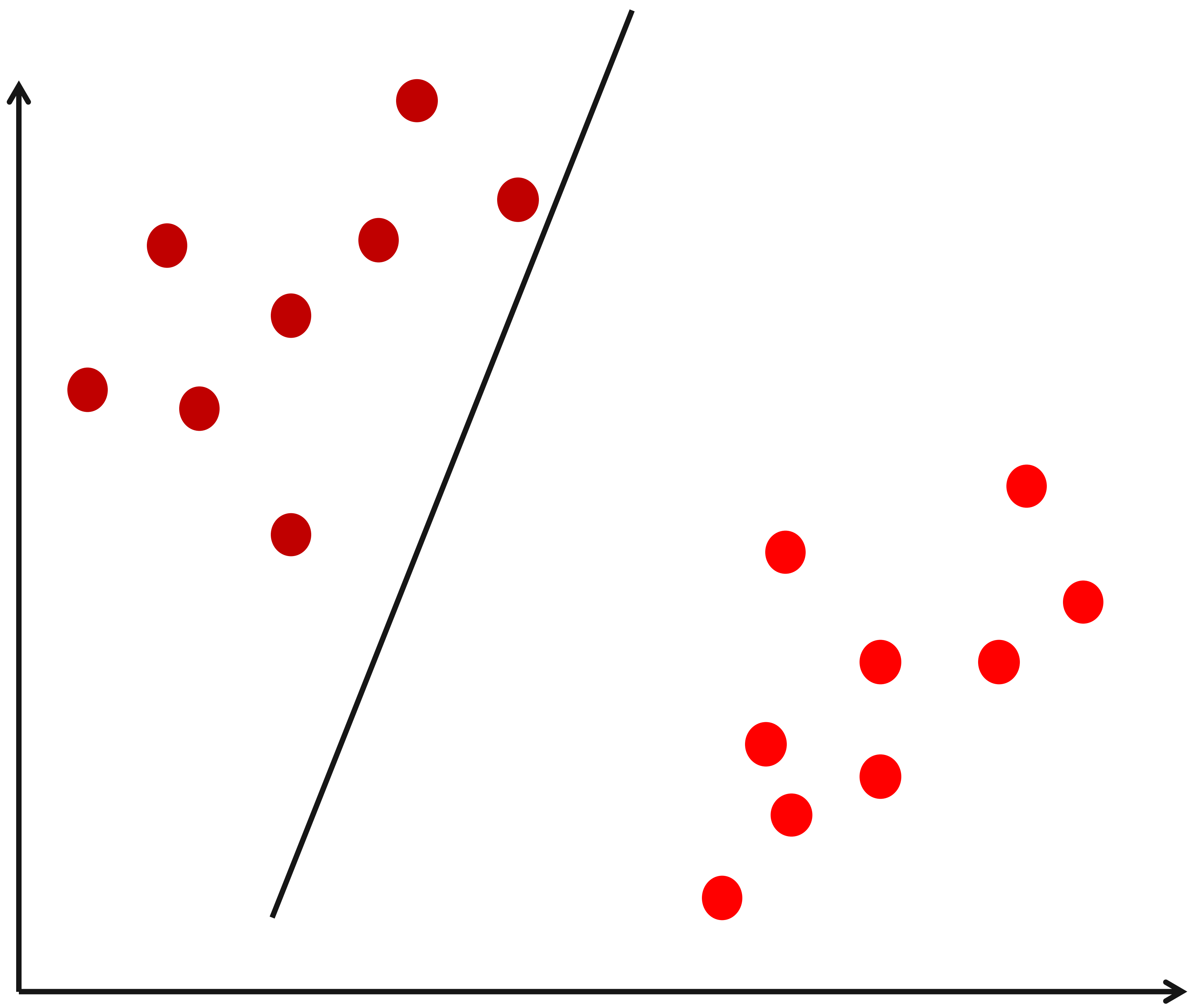
よって、ここでは、量子機械学習について以下の内容を
考えることを目的とします。

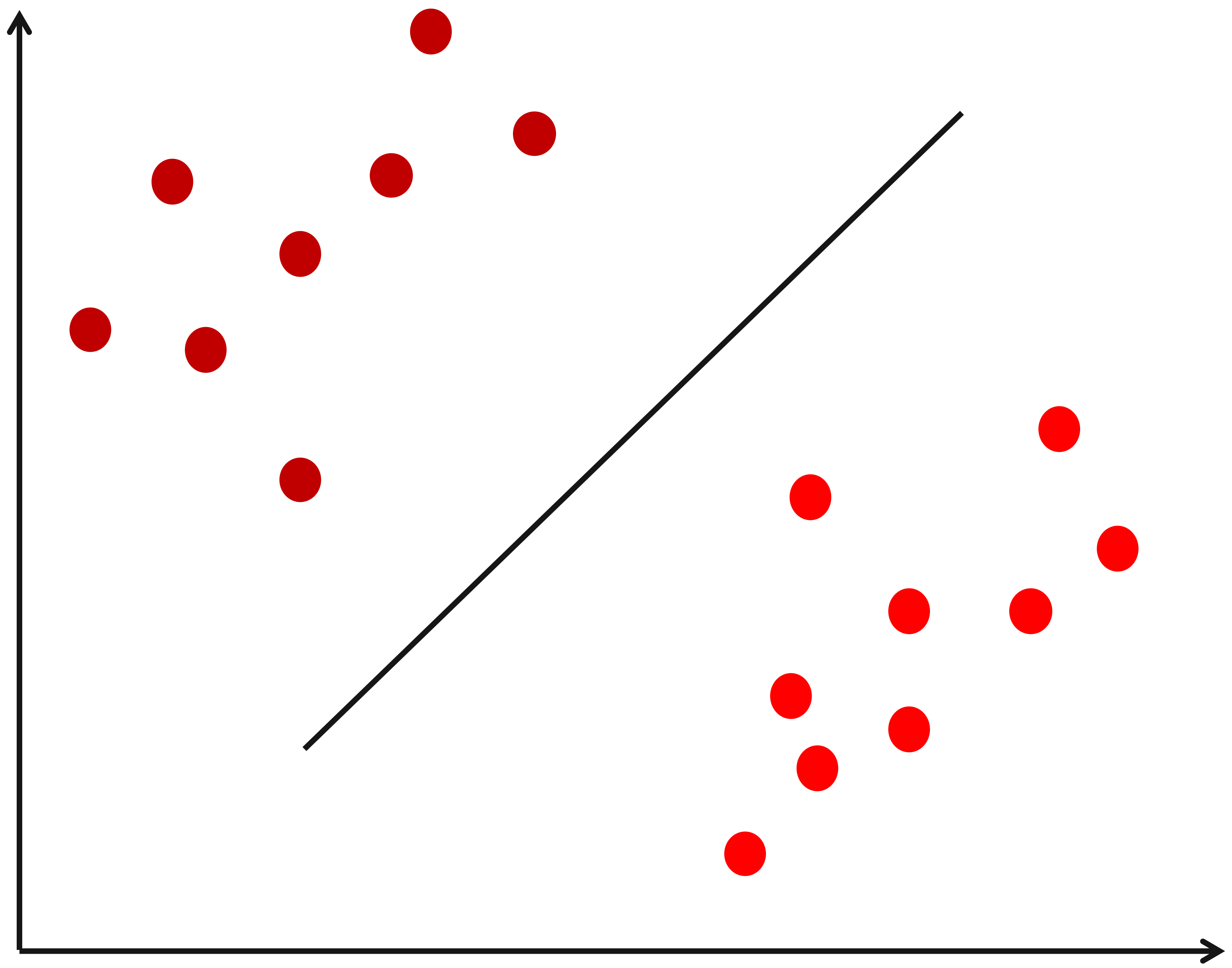
1. 古典と比較して、どのような量子回路が斬新な振る舞いをする可能性が高いのか?

2. この斬新な量子回路を使うのに最も適した性質を持つ実世界のデータは存在するか?

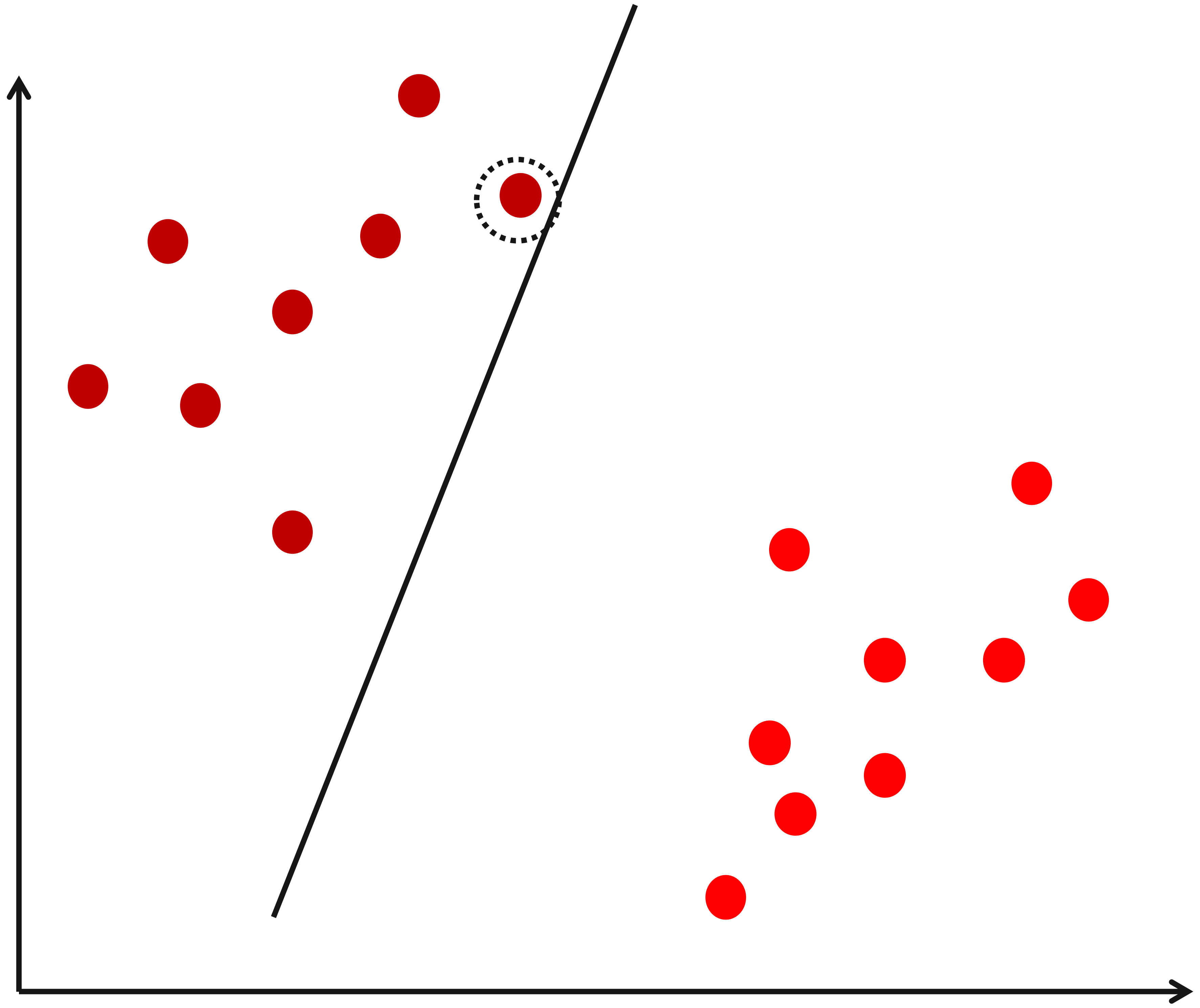
3. このような量子回路は、近い将来の量子コンピューターで実現できるか?

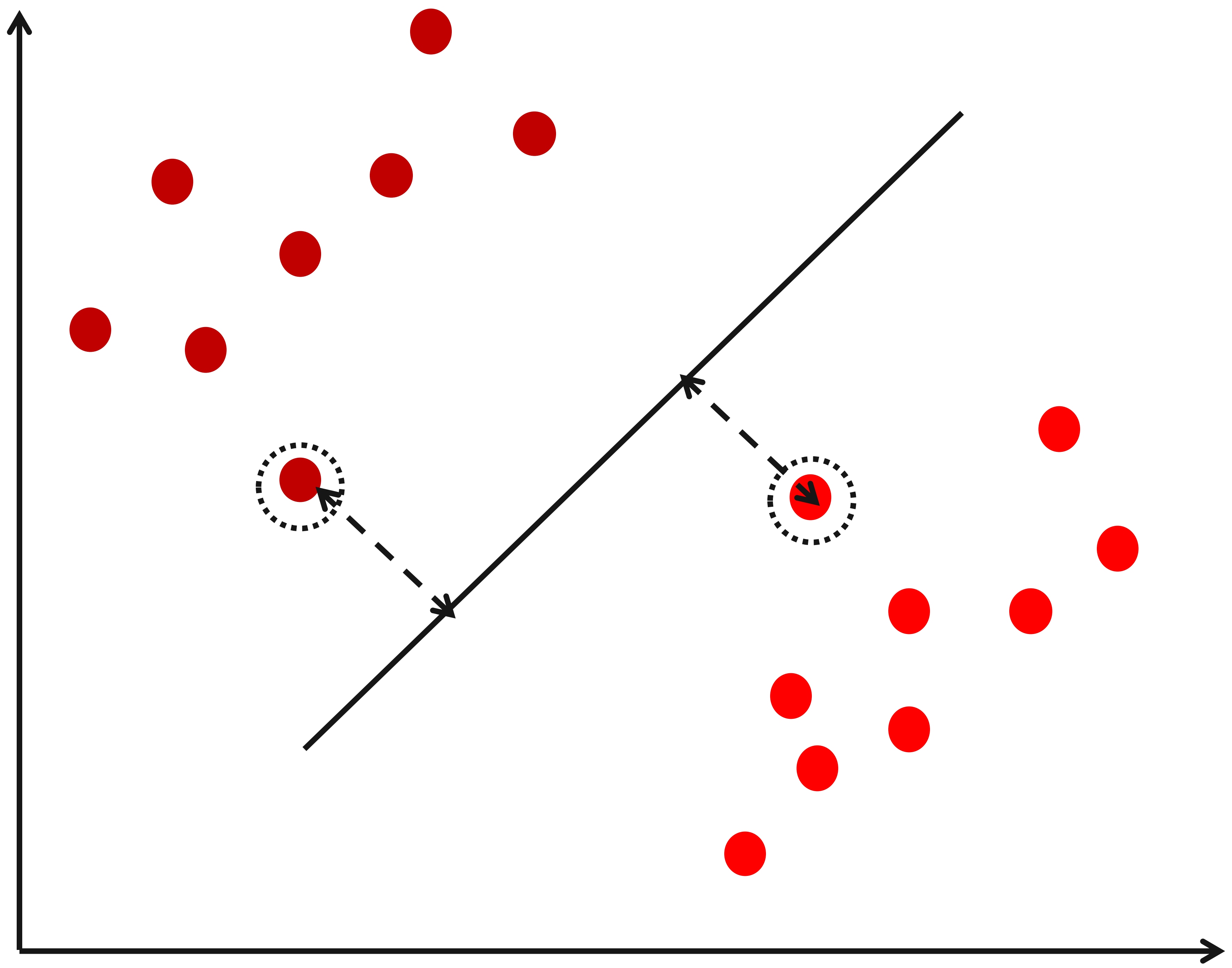
どちらの方がよく分類できているでしょうか？





右図の方が境界線と最も近いデータ点との距離が長い



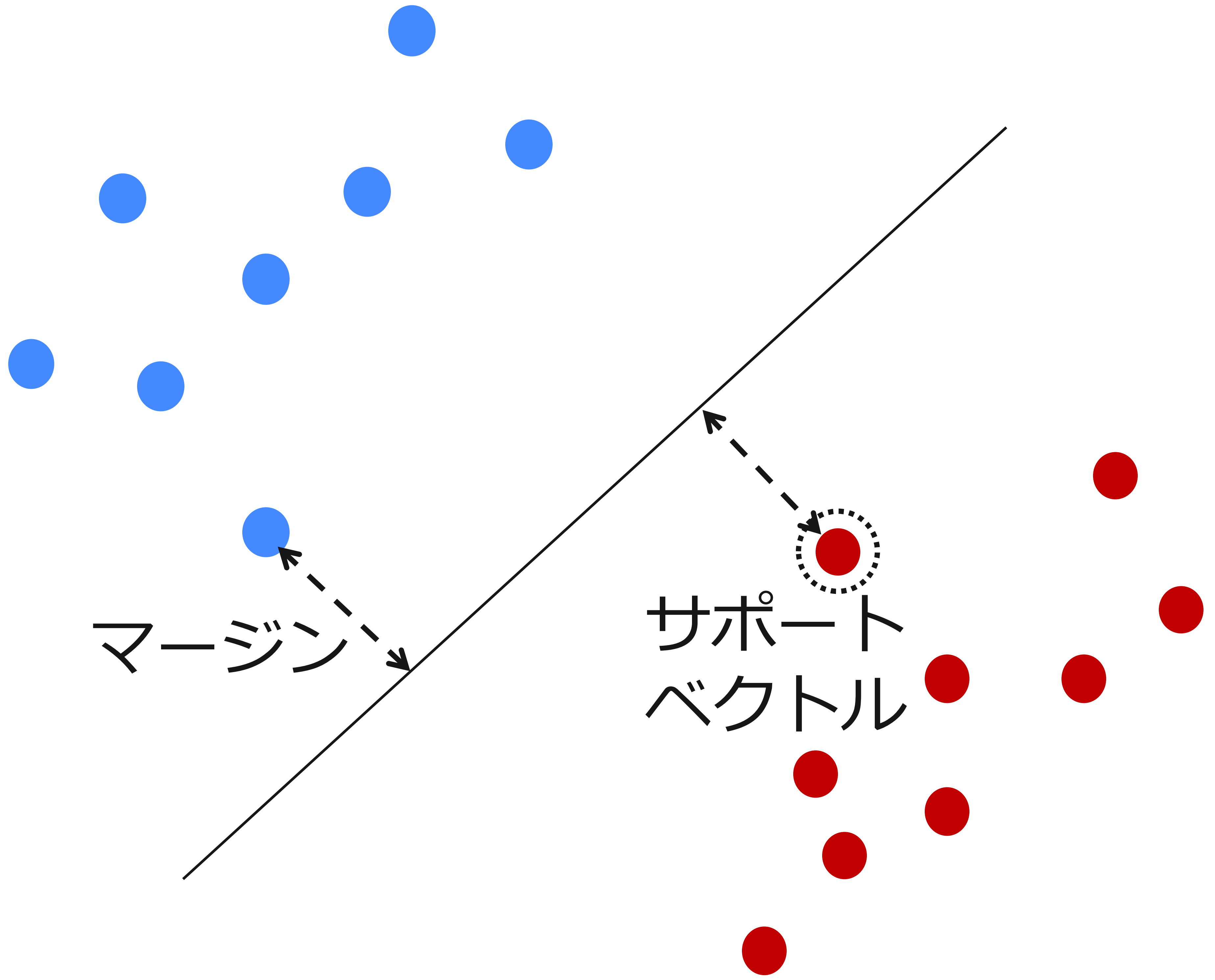


より安定した分け方

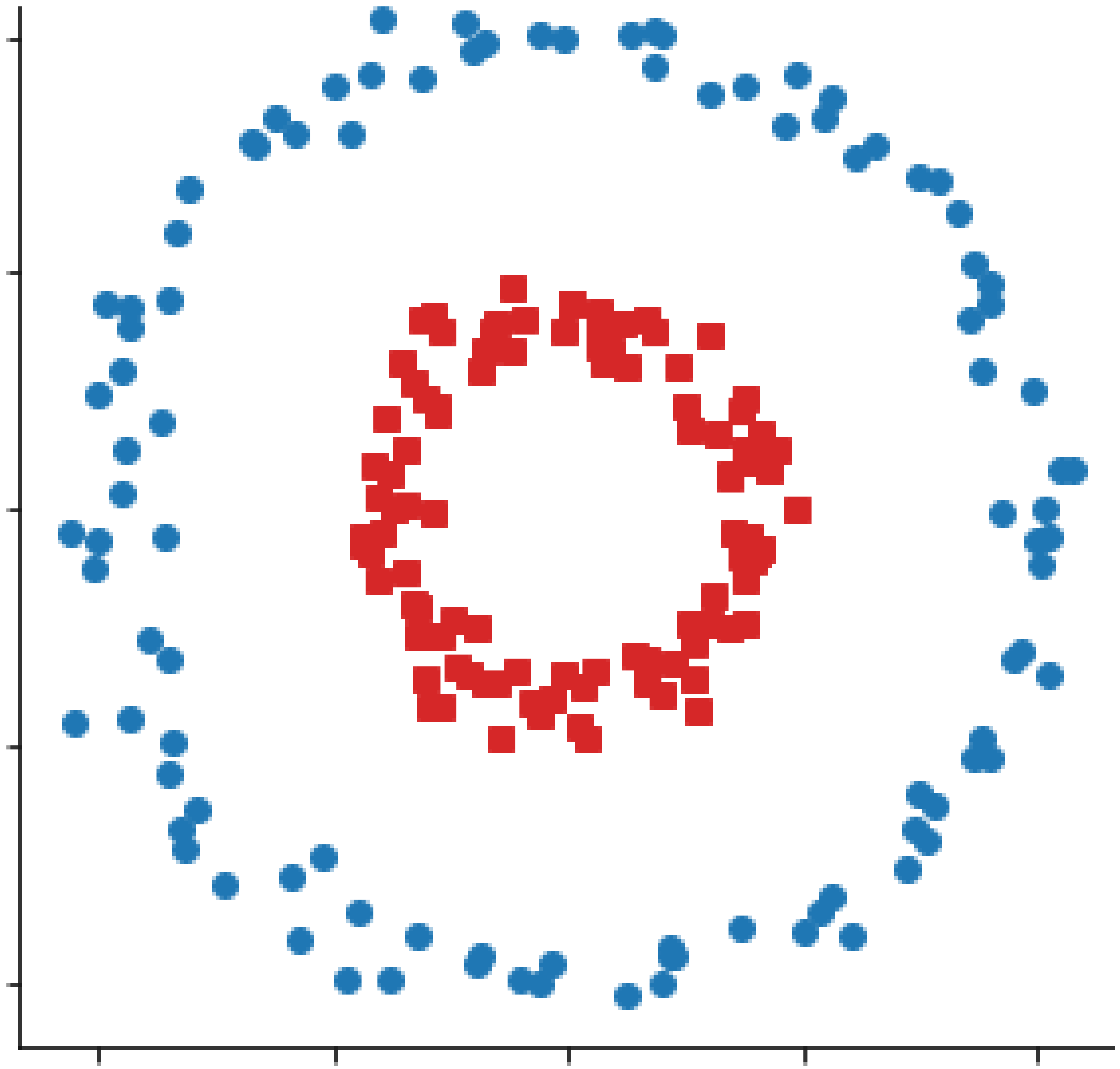
SVM(サポートベクターマシン) とは

データを2つのグループに分ける手法(2値分類)

- グループ間の境界面を定める分析手法
- マージン（境界線と最近接データ点との距離）をできるだけ大きく取るように最適化

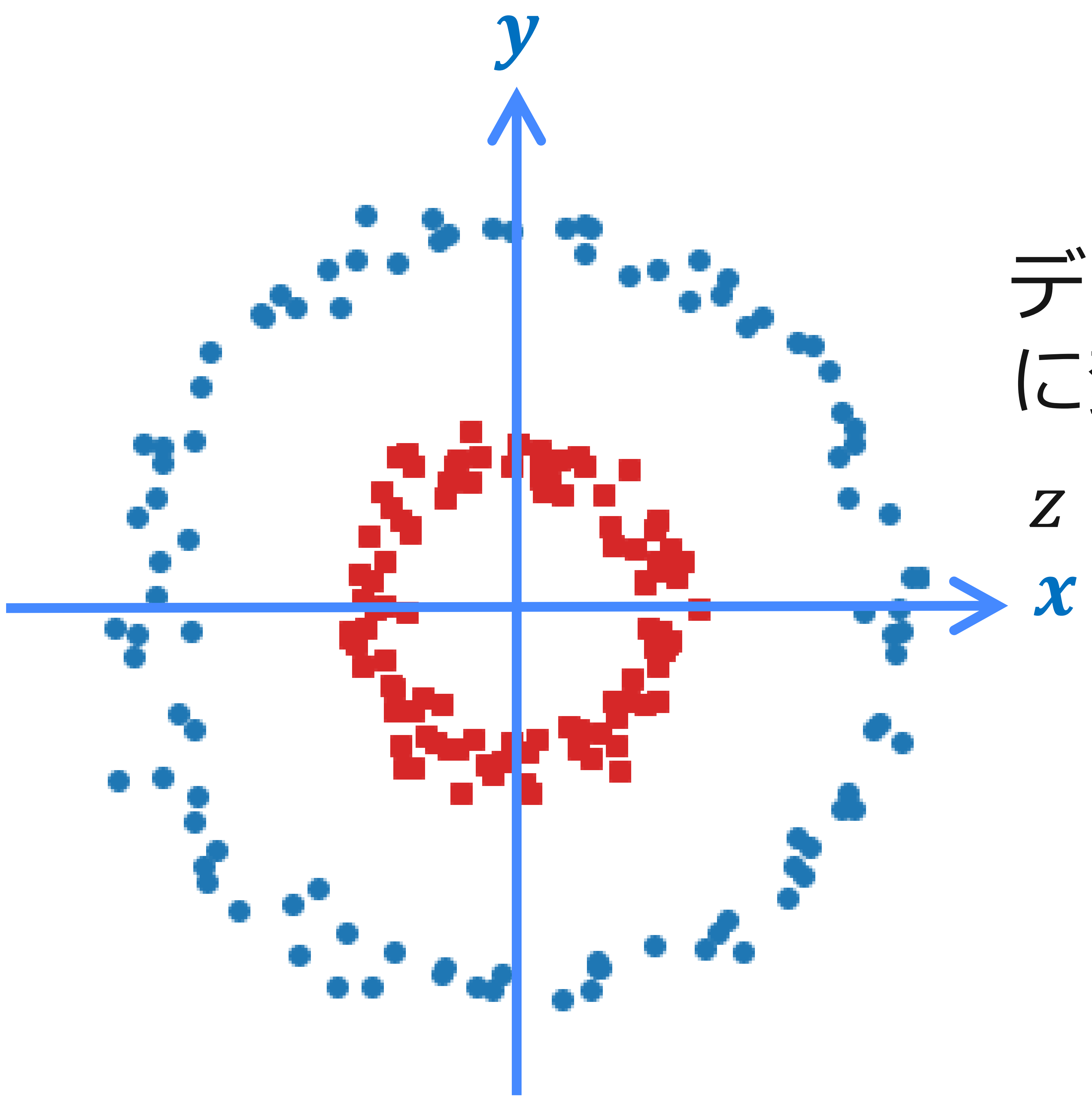


直線で分けられないデータの場合



このようなデータセットは、
2グループに分けられることは
明らかですが、
境界線が直線にはなりません。
（線形に分離できないといいます）

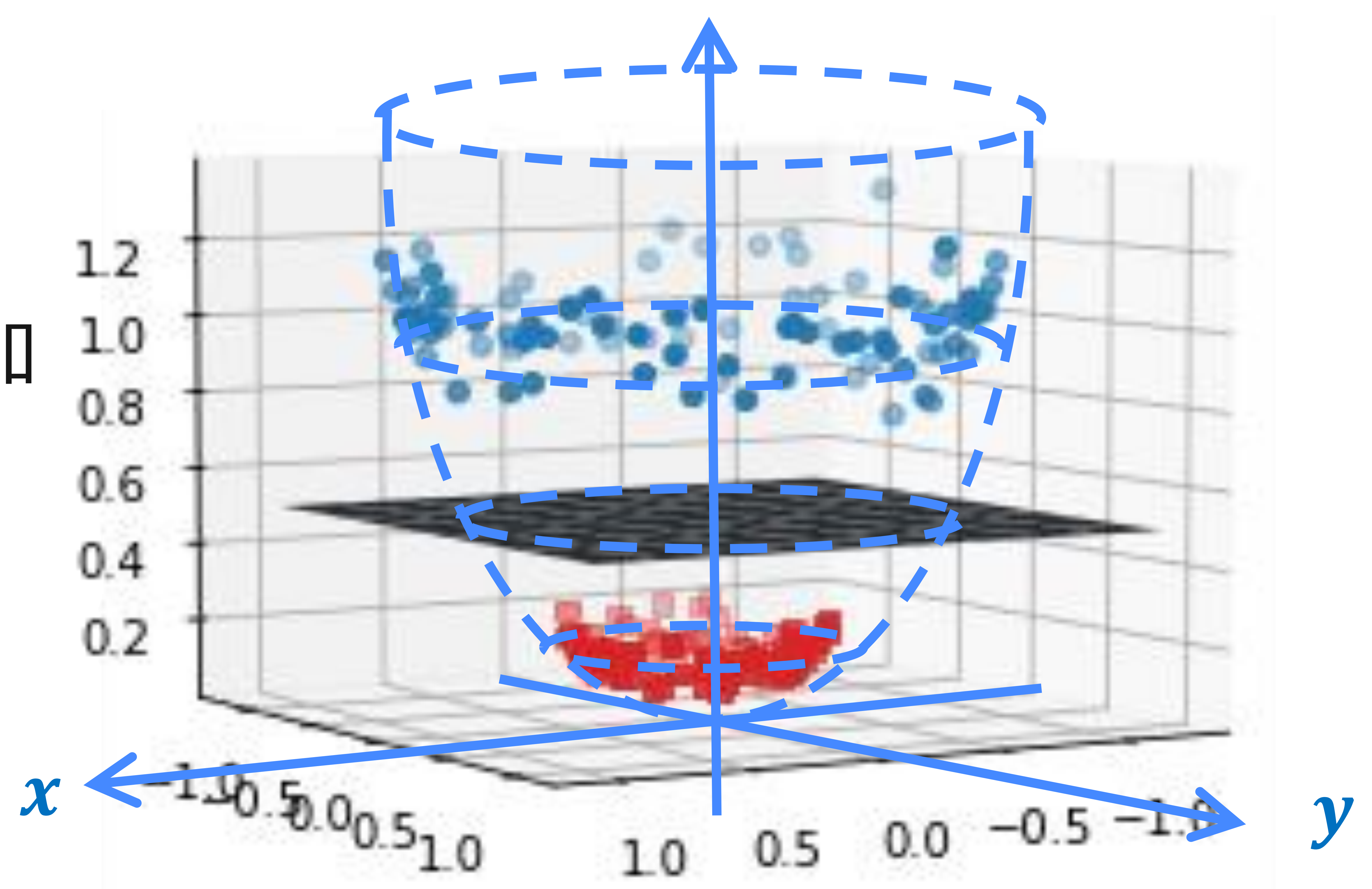
データマッピングで分類



データを3次元
に変換

$z = x^2 + y^2$を追加

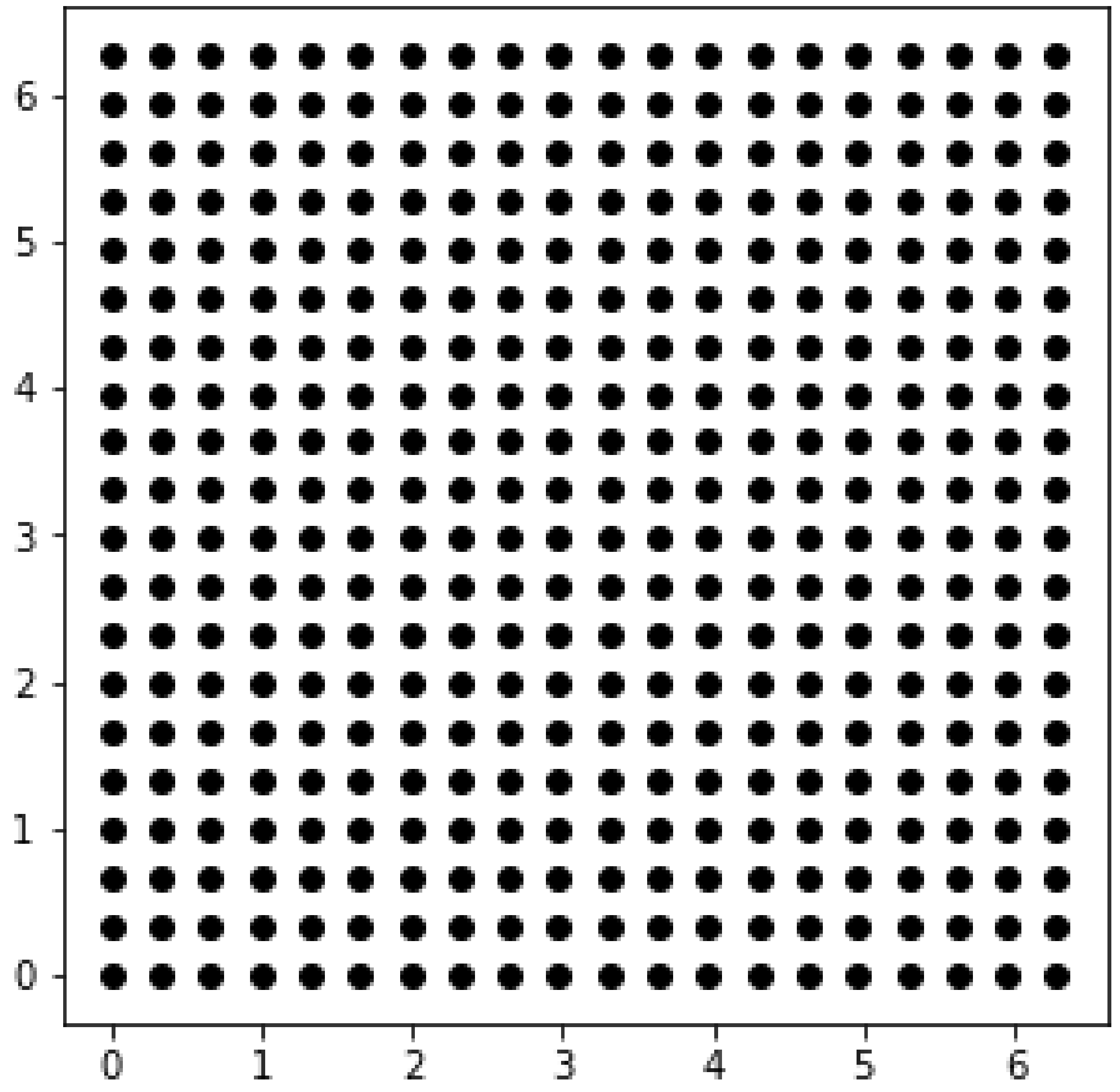
3次元では、データは平面で
分離できる



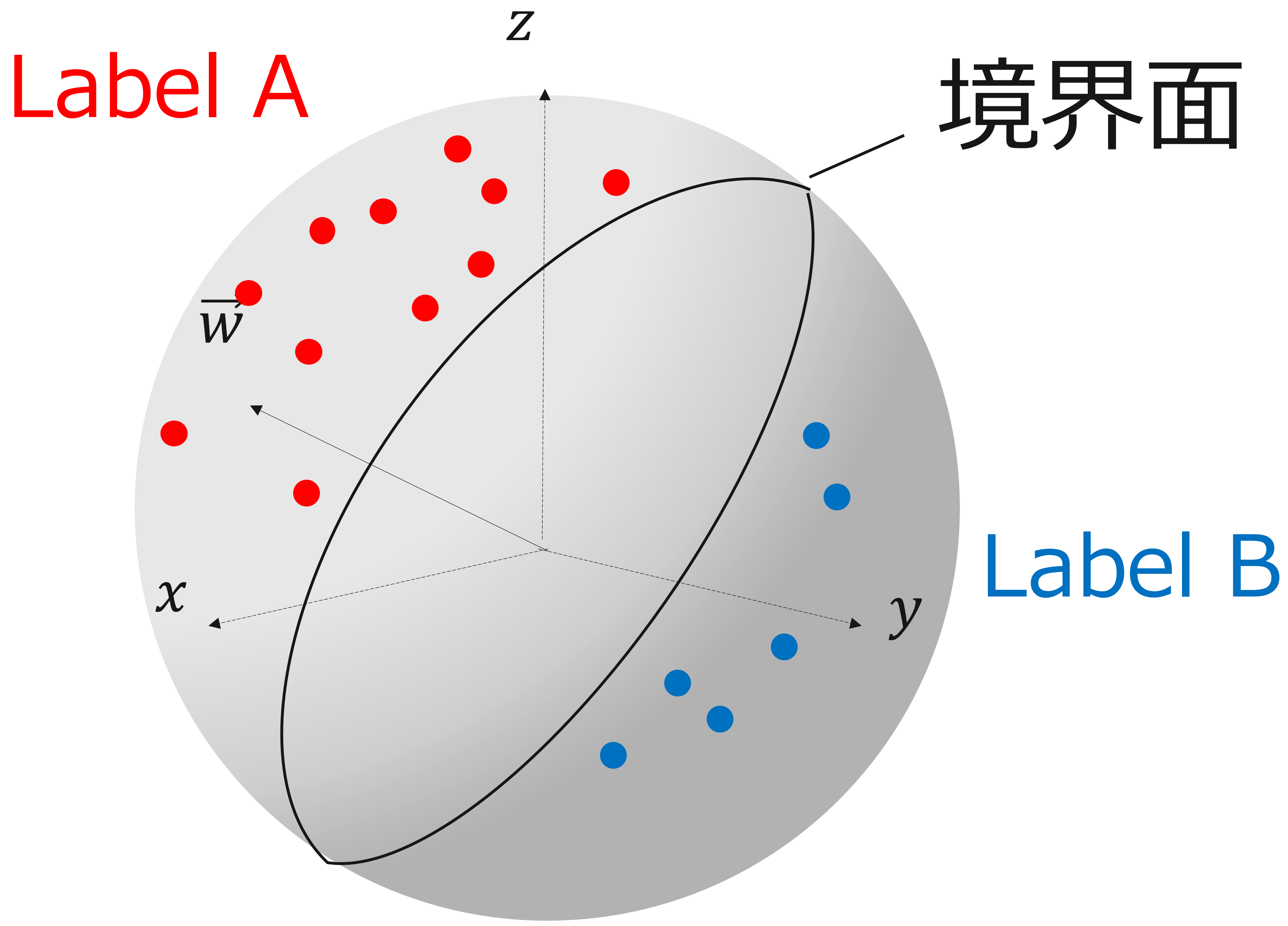
特徴量を高次元化(特徴量マッピング)することで、
平な(線形な)境界面で切り分けることができます。

量子SVM(サポートベクターマシン)

特徴量を量子空間に特徴量マッピングすることで、線形な境界面で切り分けます。

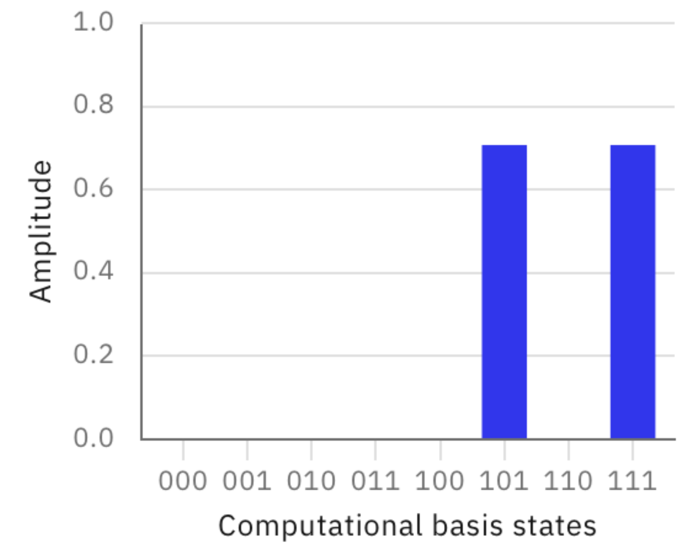


量子状態の球にマッピング



SVMを使って、データを境界平面で2つのグループに分ける。

データを量子機械学習のために符号化する手法（代表的なもの）



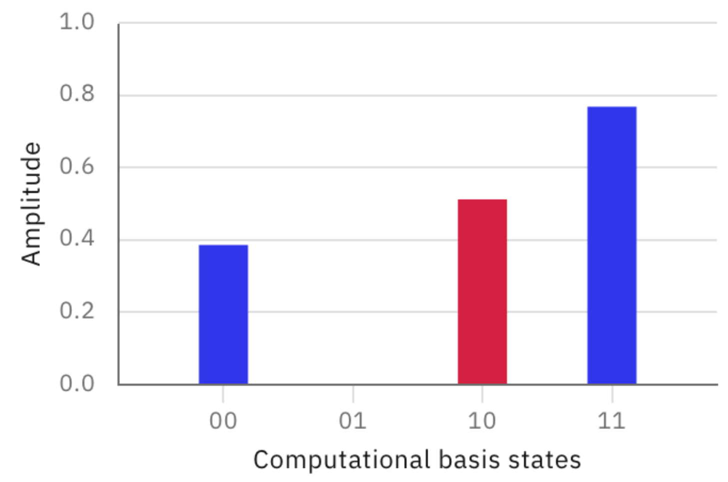
1. 計算基底符号化

量子状態

例）$X = \{x_1 = 101,\ x_2 = 111\}$ ➡ $|x\rangle = \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle)$

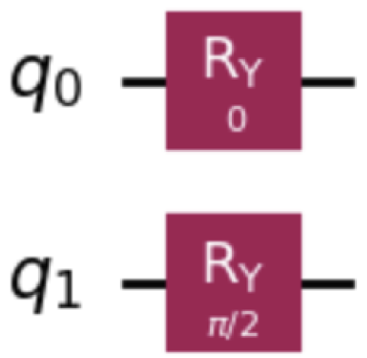
2. 振幅符号化

例）$X = \left\{x_1 = (1.5, 0),\quad x_2 = (-2, 3)\right\}$ ➡ $|x\rangle = \frac{1}{\sqrt{15.25}}(1.5|00\rangle - 2|10\rangle + 3|11\rangle)$



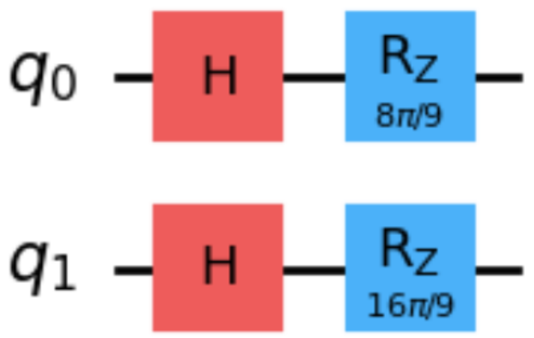
3. 角度符号化

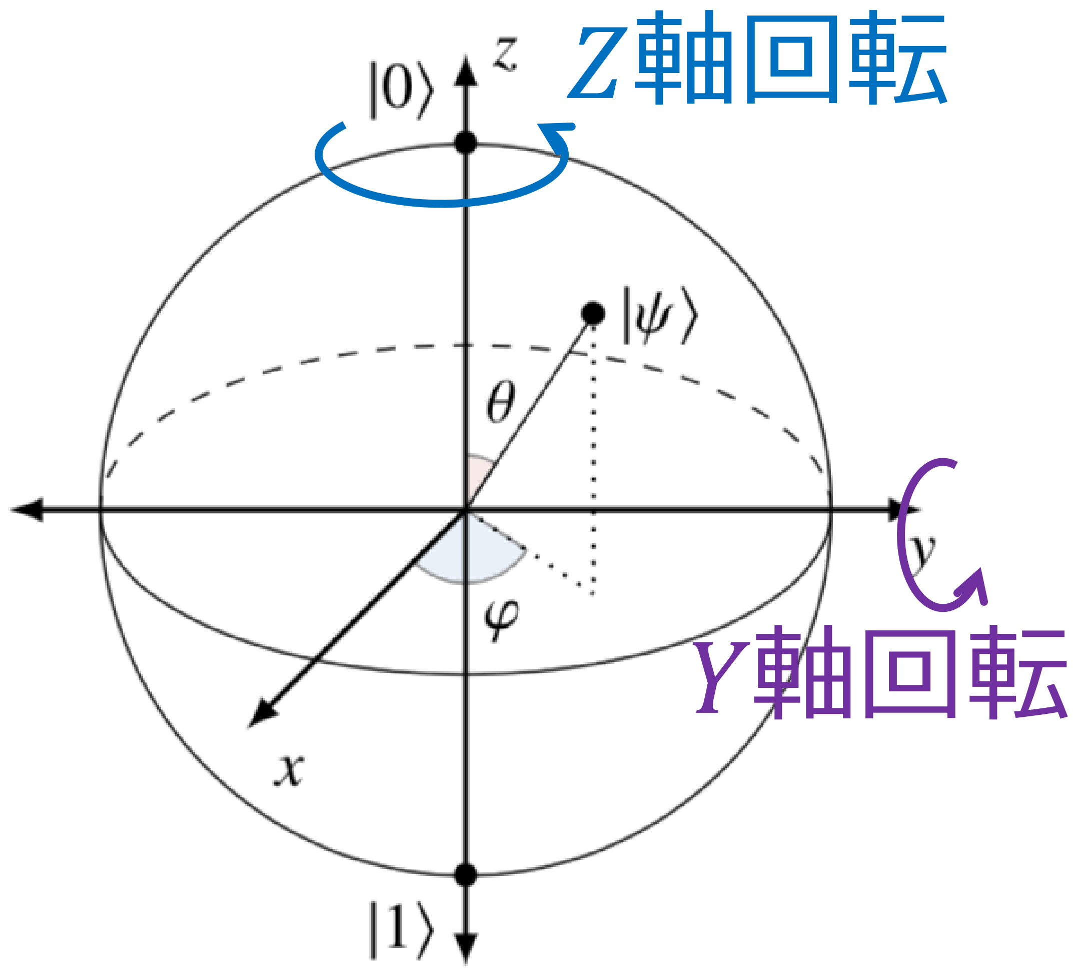
例）$x = (x_1,\ x_2)$ ➡ $S_x = RY(x_1) \otimes RY(x_2)$



4. 位相符号化

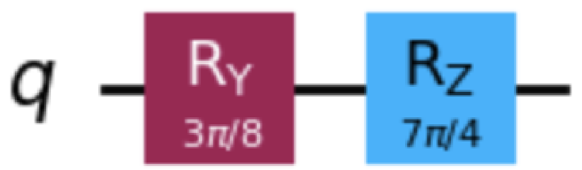
例）$x = (x_1,\ x_2)$ ➡ $S_x = P(x_1)H \otimes P(x_2)H$





5. 密な度符号化

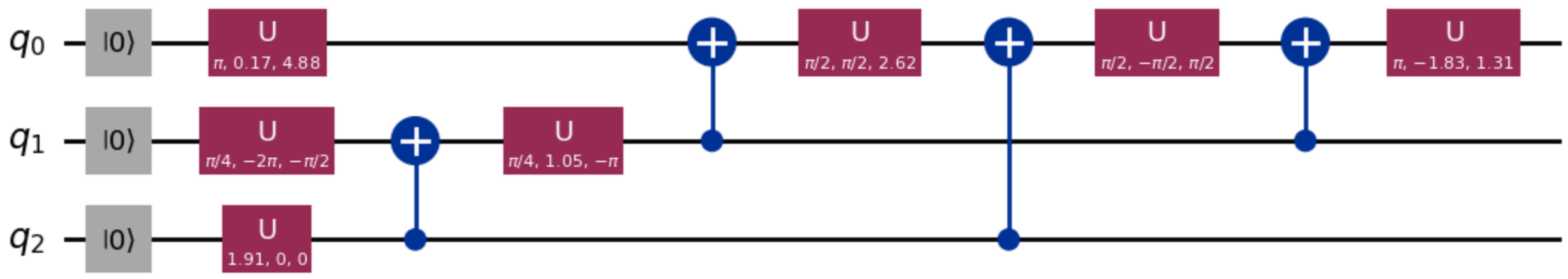
例）$x = (x_1,\ x_2)$ ➡ $S_x = P(x_2)RY(x_1)$



データを量子機械学習のために符号化する手 回路の深さに注意！

1. 計算基底符号化

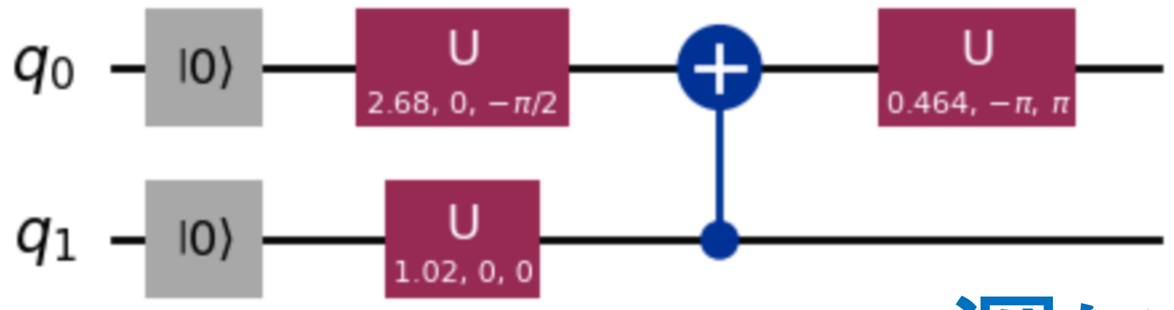
例） $X = \{x_1 = 101,\ x_2 = 111\}$ ➡ 量子状態 $|x\rangle = \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle)$



深い

2. 振幅符号化

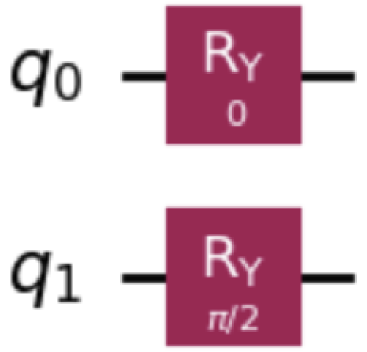
例） $X = \left\{x_1 = (1.5, 0),\quad x_2 = (-2, 3)\right\}$ ➡ $|x\rangle = \frac{1}{\sqrt{15.25}}(1.5|00\rangle - 2|10\rangle + 3|11\rangle)$



深い

3. 角度符号化

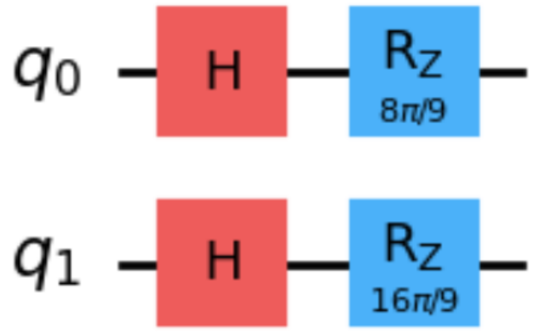
例） $x = (x_1,\ x_2)$ ➡ $S_x = RY(x_1) \otimes RY(x_2)$



Depth = 1

4. 位相符号化

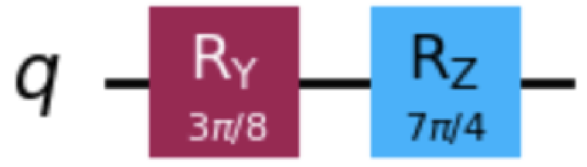
例） $x = (x_1,\ x_2)$ ➡ $S_x = P(x_1)H \otimes P(x_2)H$



Depth = 2

5. 密な度符号化

例） $x = (x_1,\ x_2)$ ➡ $S_x = P(x_2)RY(x_1)$

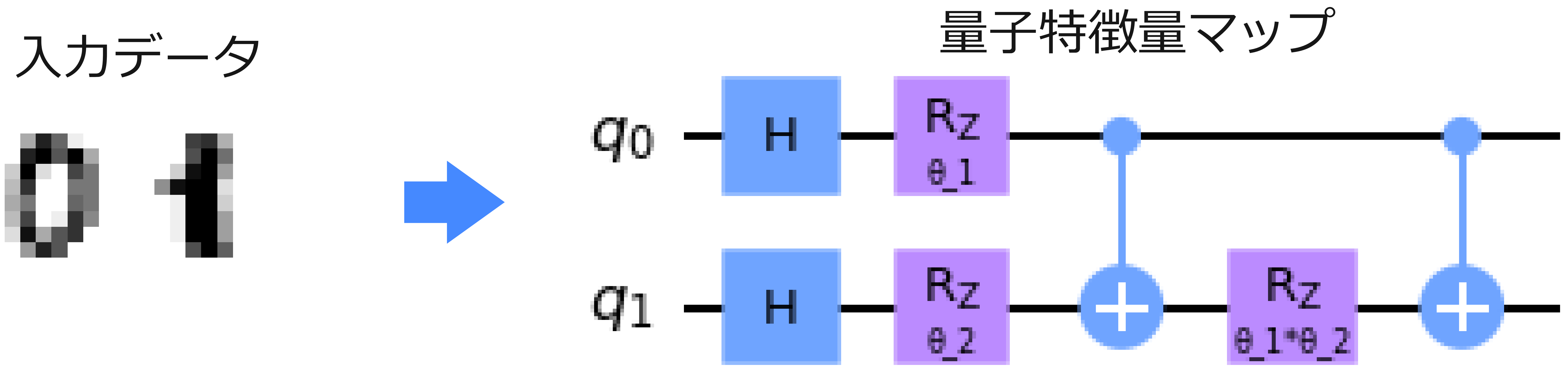


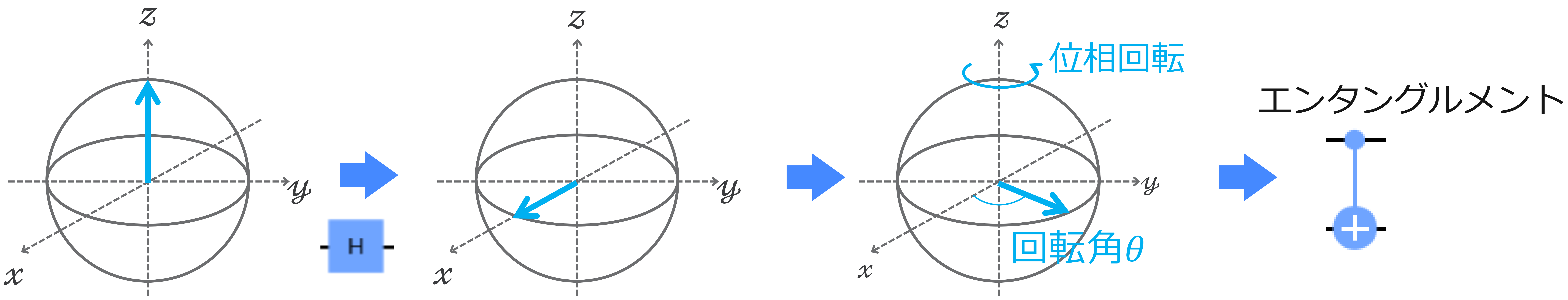
この場合は、
Depth=2、
量子ビット数/データ数 = 1/2

6. 密な角度符号化の拡張版

Qiskitの組み込み符号化スキーム

量子特徴量マップ(Feature Map) が
密な角度符号化の拡張版として組み込まれています。

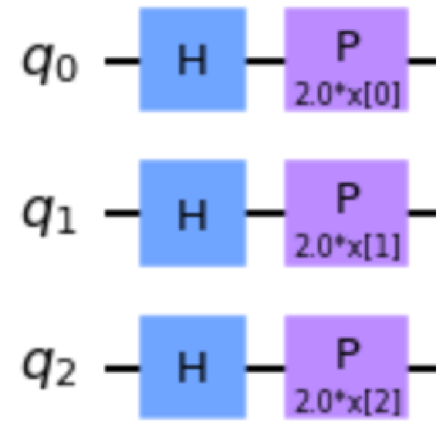


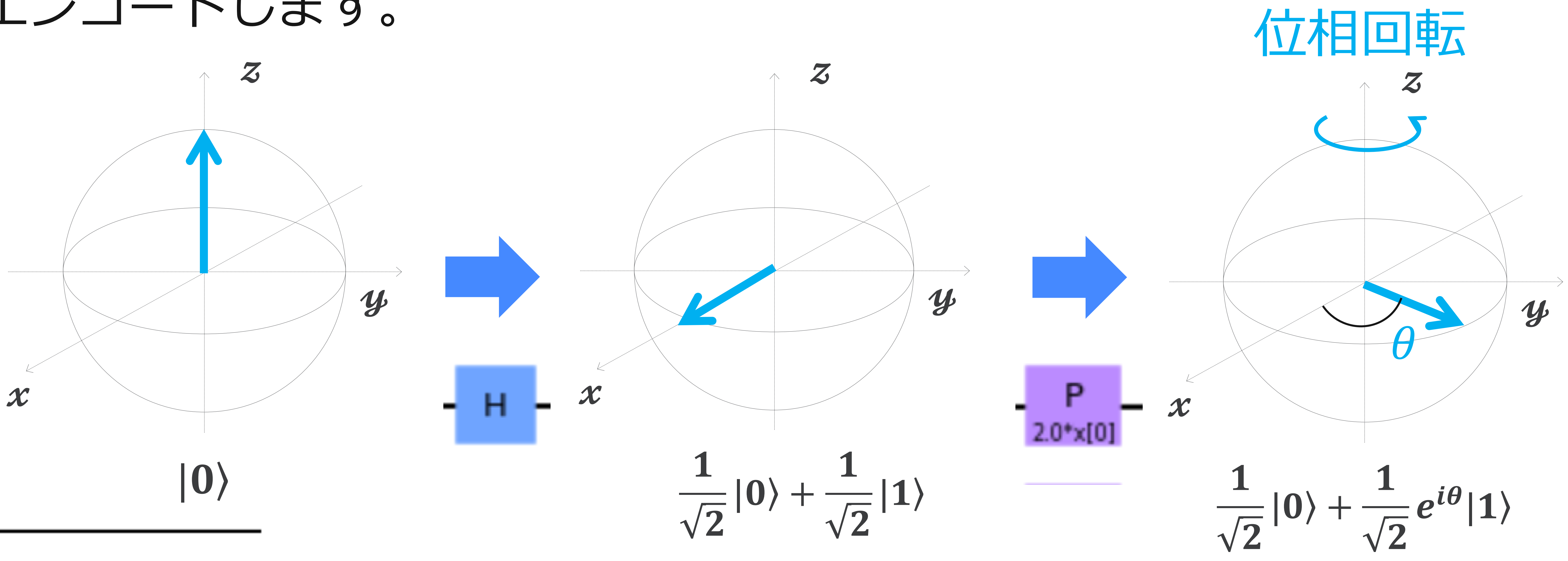


Quantum feature map

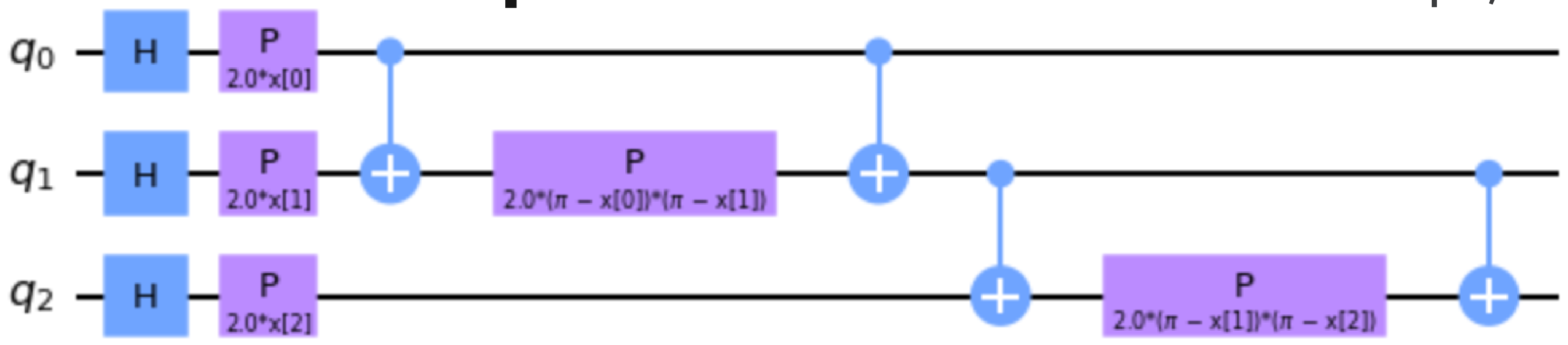
量子ゲートの回転角にデータをエンコードします。

Z feature map



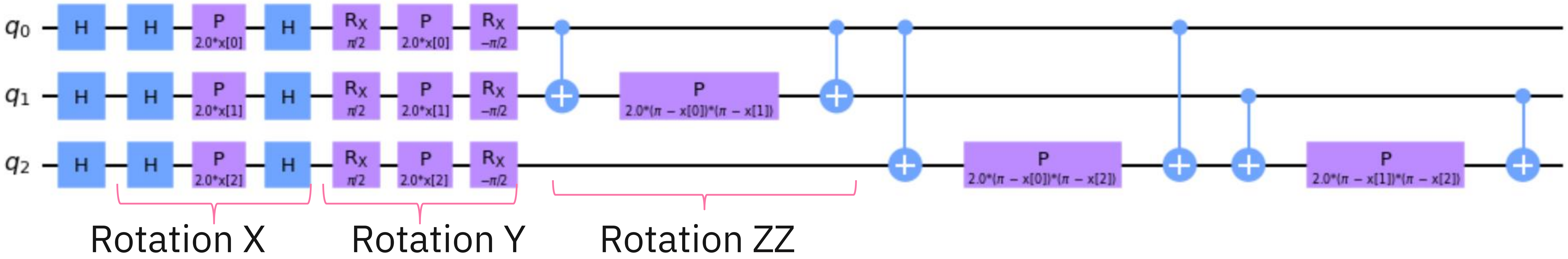


ZZ feature map



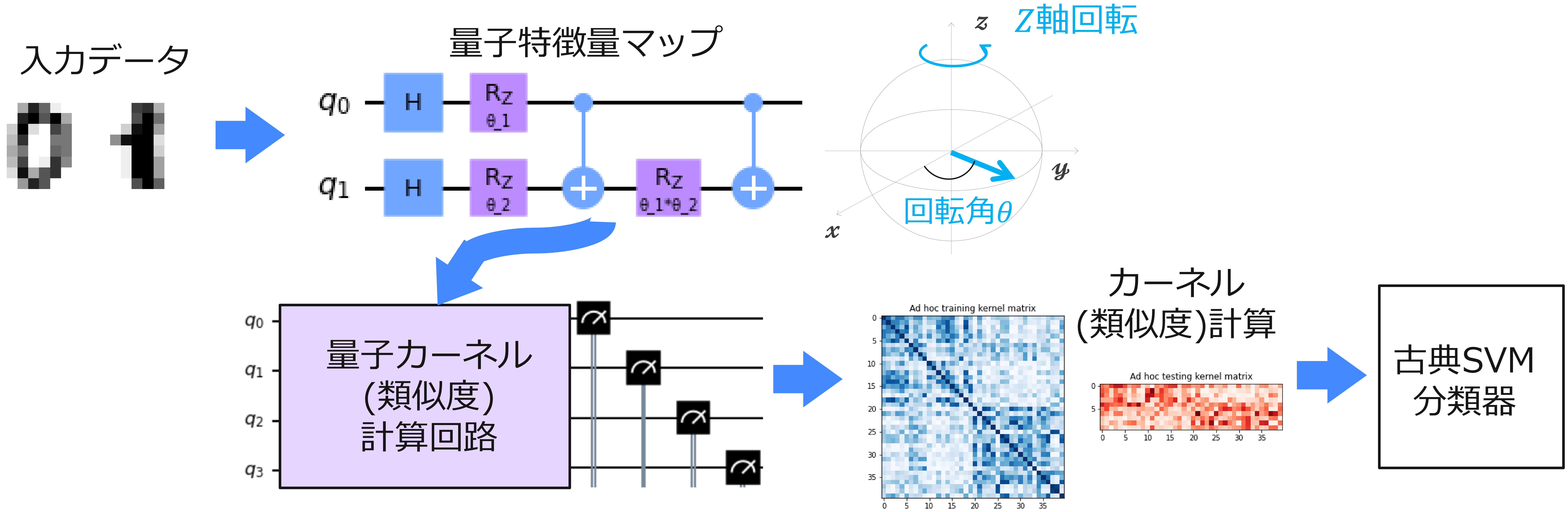
Qiskit組み込みのFeature mapを使わずに、より良いFeature map回路を自作してもよいです。

Pauli feature map



量子カーネルSVM

1. 量子特徴量マップを使ってデータを量子状態にエンコードする。
2. 量子コンピューターを用いて量子カーネルを計算する。
3. 古典コンピューターの古典SVMで学習・分類する。



量子カーネルは内積計算で求める

カーネル関数は、特徴量空間のベクトルを引数として受け取り、
その内積 $\langle\Phi(x)|\Phi(y)\rangle$を返します。

- データベクトル： $\overrightarrow{x_i}$
- $\overrightarrow{x_i}$のエンコードとマッピングを行う回路： $\Phi(\overrightarrow{x_i})$

マップされた状態は：

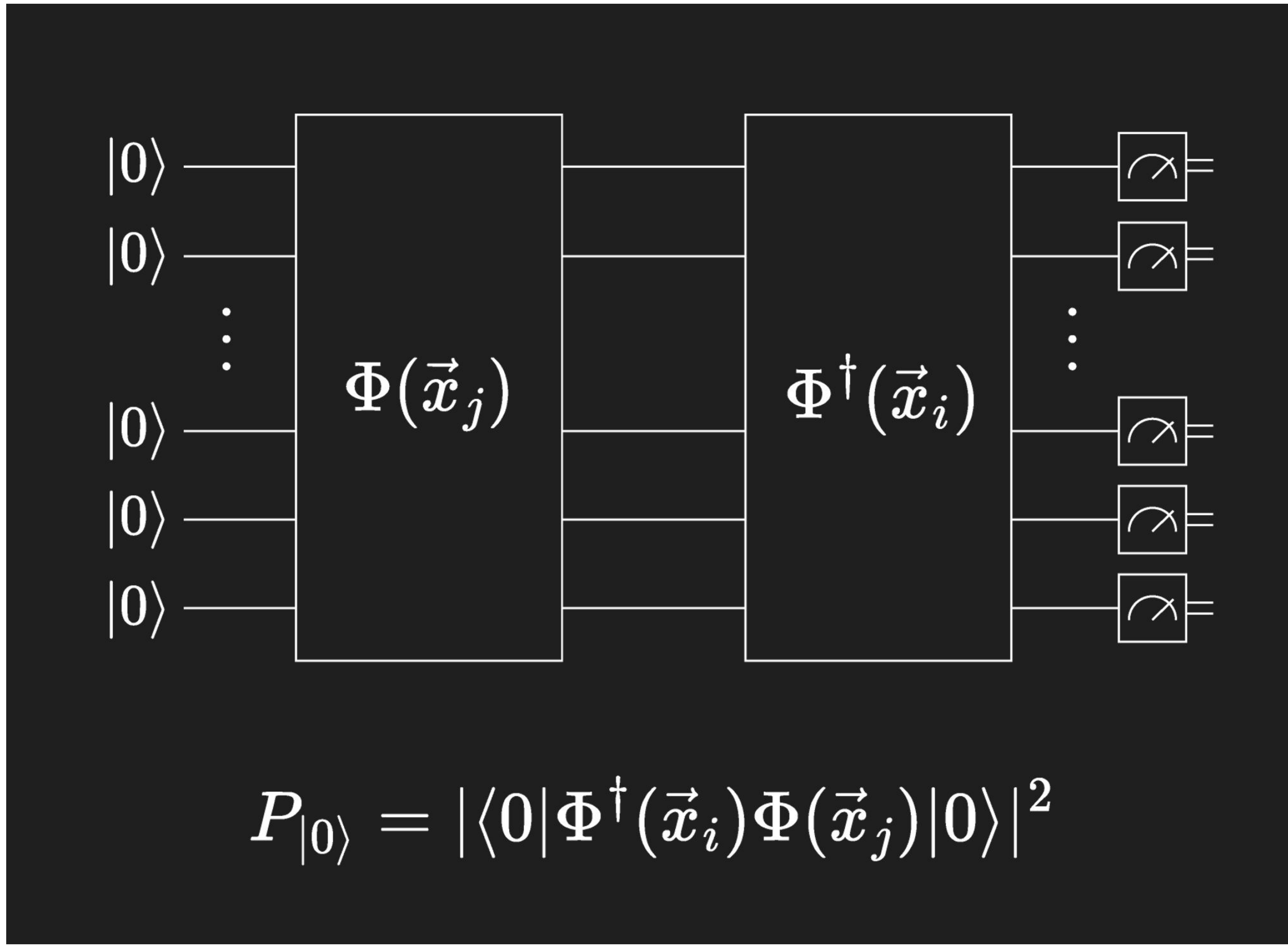
$$|\psi(\overrightarrow{x_i})\rangle = \Phi(\overrightarrow{x_i})|0\rangle^{\otimes N}$$
$$|\psi(\overrightarrow{x_j})\rangle = \Phi(\overrightarrow{x_j})|0\rangle^{\otimes N}$$

その内積は：

$$\langle\psi(\overrightarrow{x_j})|\psi(\overrightarrow{x_i})\rangle = \langle 0|^{\otimes N}\Phi^\dagger(\overrightarrow{x_j})\Phi(\overrightarrow{x_i})|0\rangle^{\otimes N}$$

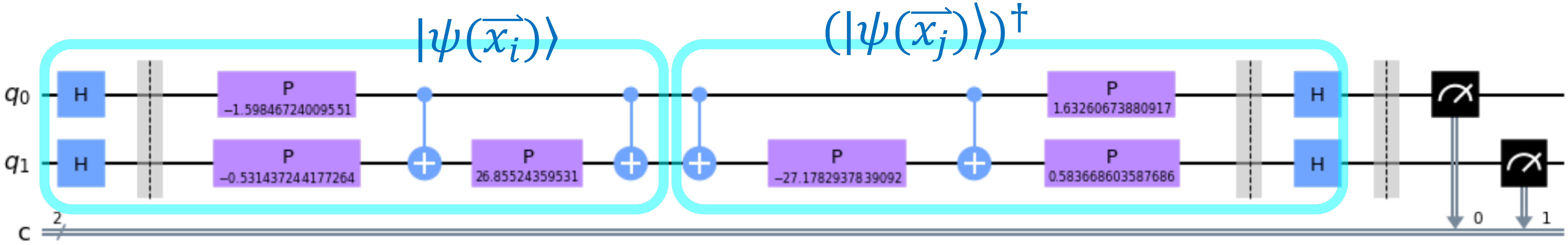
カーネル行列の要素は、状態$|0\rangle^{\otimes N}$ を観測する確率：

$$P_0 = \left|\langle 0|^{\otimes N}\Phi^\dagger(\overrightarrow{x_j})\Phi(\overrightarrow{x_i})|0\rangle^{\otimes N}\right|^2$$



量子カーネル回路

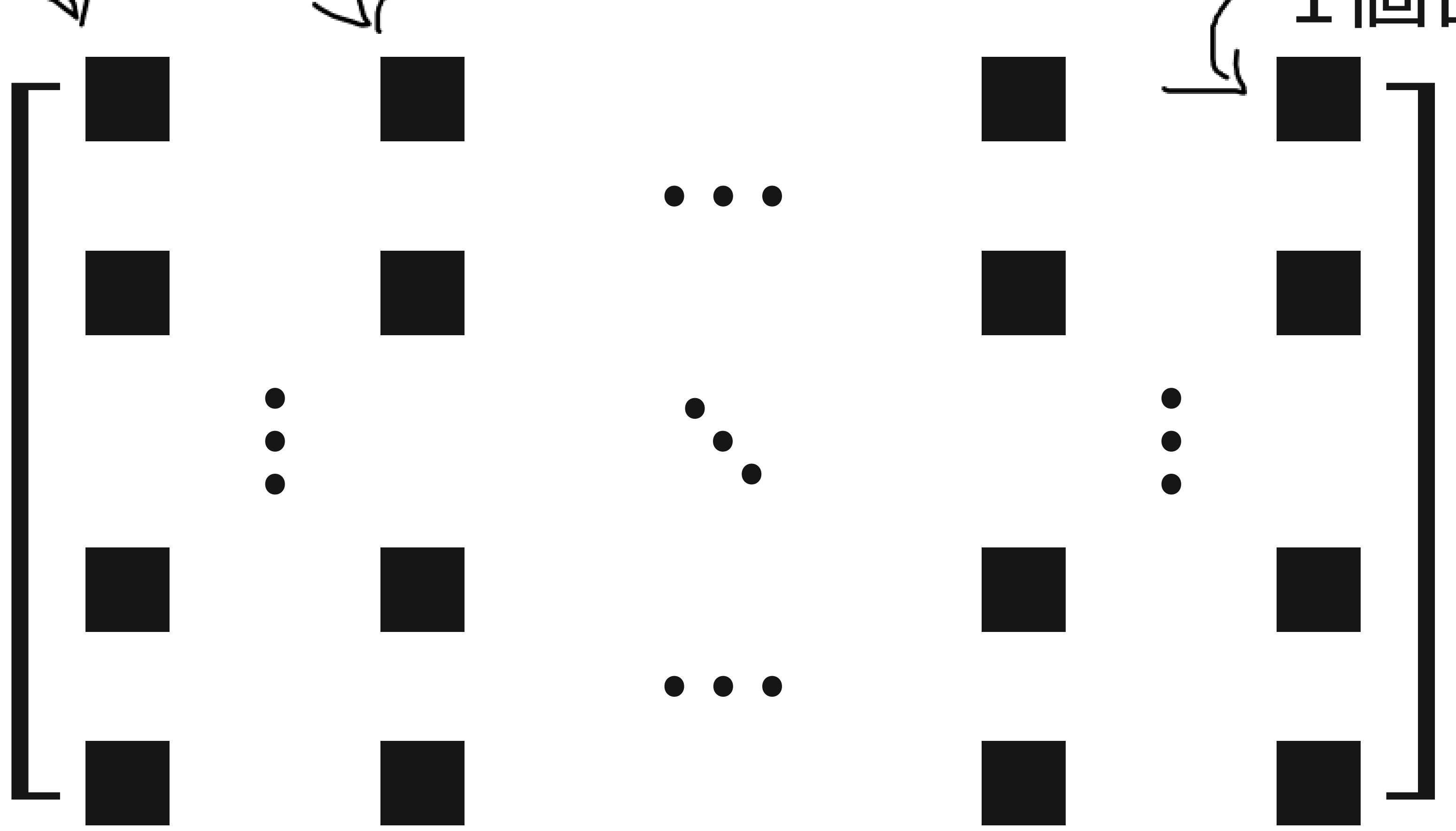
各データ対に対して内積（量子カーネル）$\langle\psi(\vec{x_j})|\psi(\vec{x_i})\rangle$ を計算、測定して、カーネル行列 $K(x_i, x_k) = \left|\langle\psi(\vec{x_j})|\psi(\vec{x_i})\rangle\right|^2$ を作っていきます。



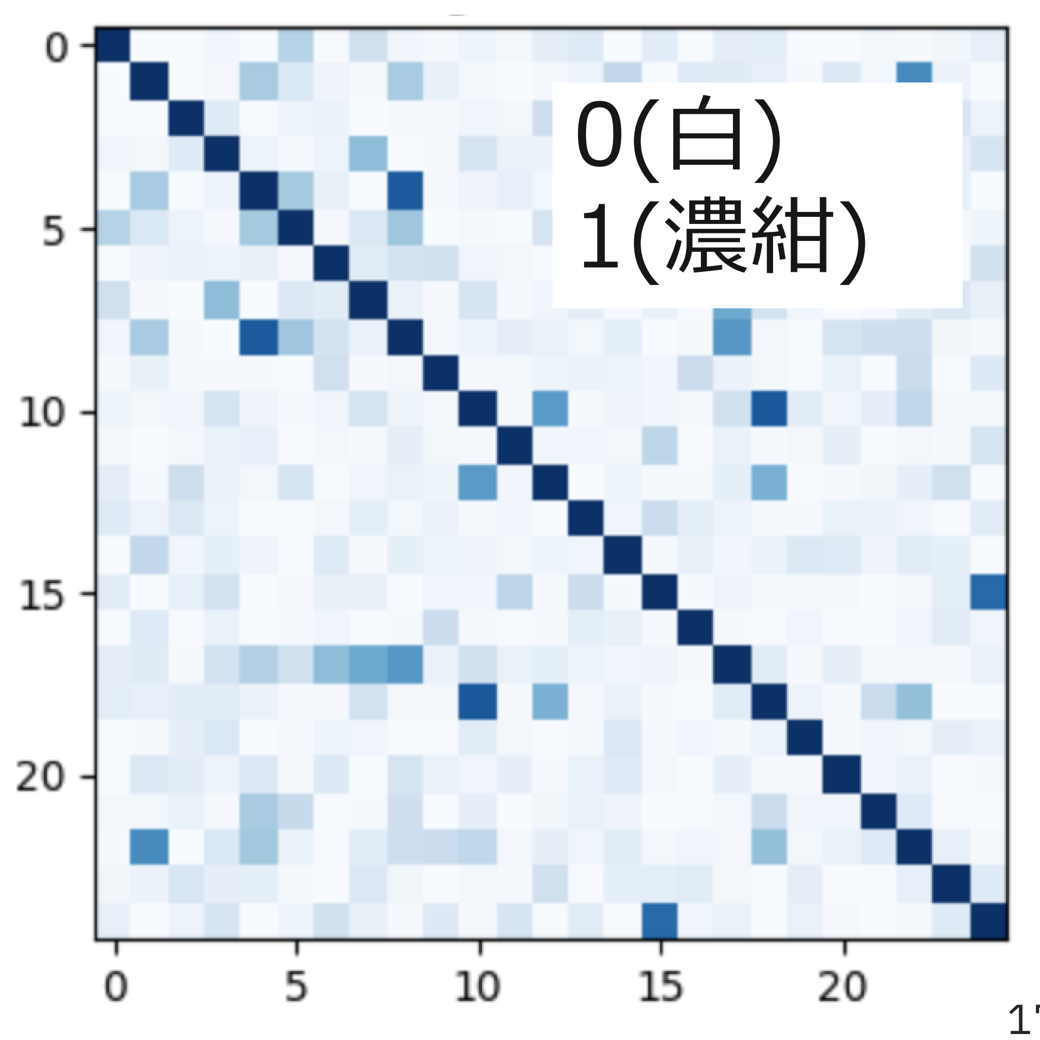
1個目のデータと1個目のデータの内積

1個目のデータと2個目のデータの内積

1個目のデータとn個目のデータの内積



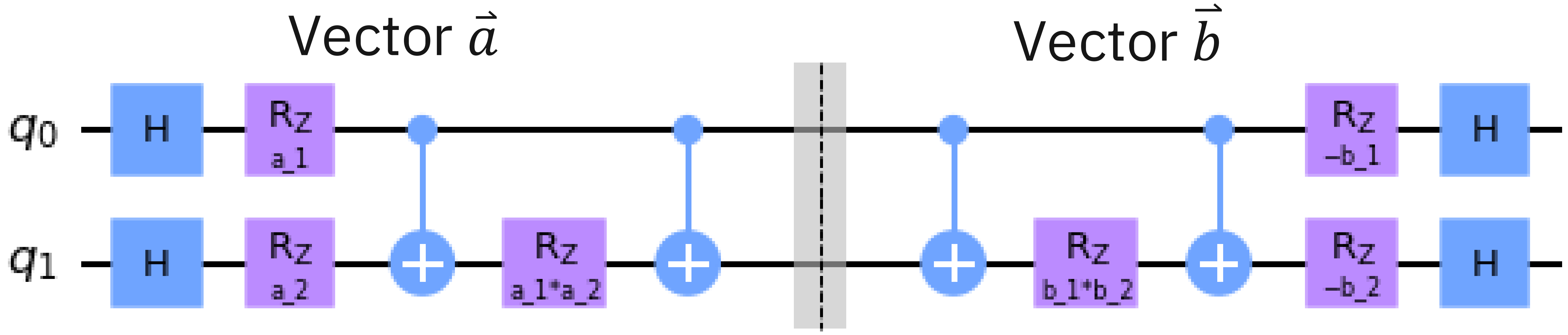
学習データが 25 個ある場合、
25 x 25 の量子カーネル行列
を作成します。

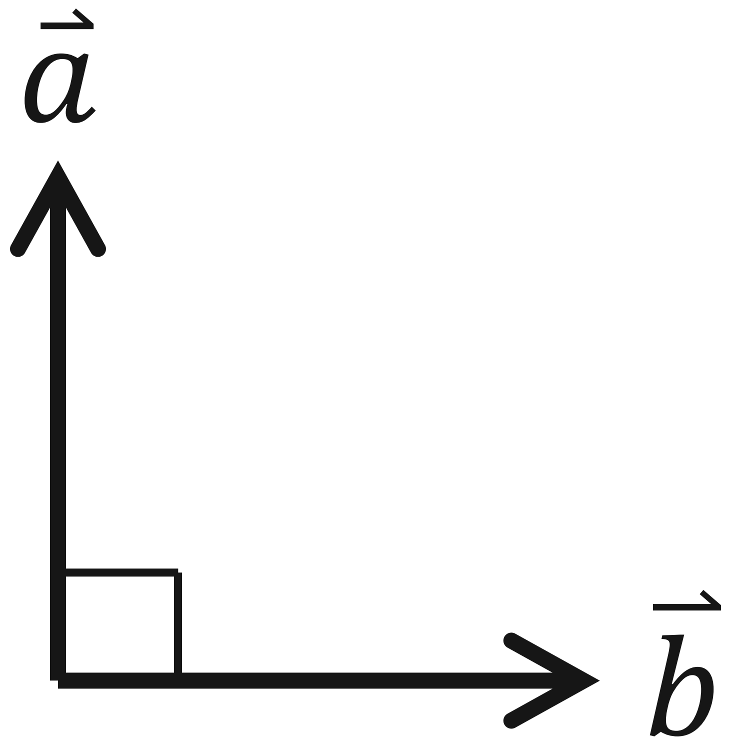


n個目のデータとn個目のデータの内積

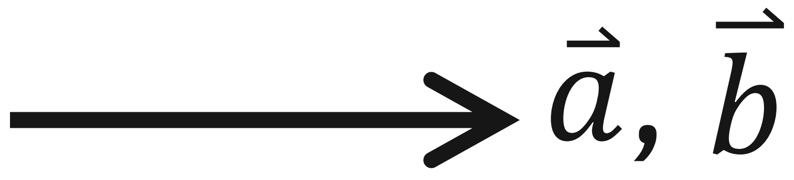
カーネル計算は類似度の計算

カーネル要素 $\langle b|a\rangle$ は、$|a\rangle$ と $|b\rangle$のフィデリティー計算と同じ。

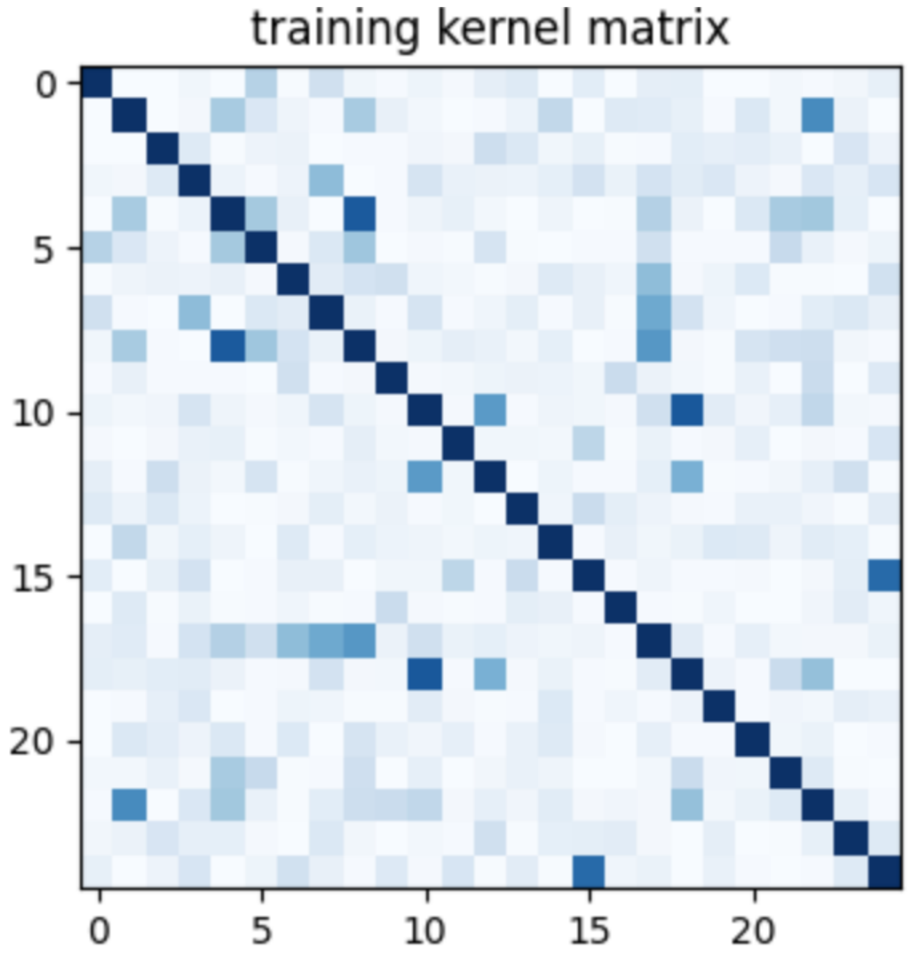




直行した
ベクトルの
内積は0



自分自身との
内積は1



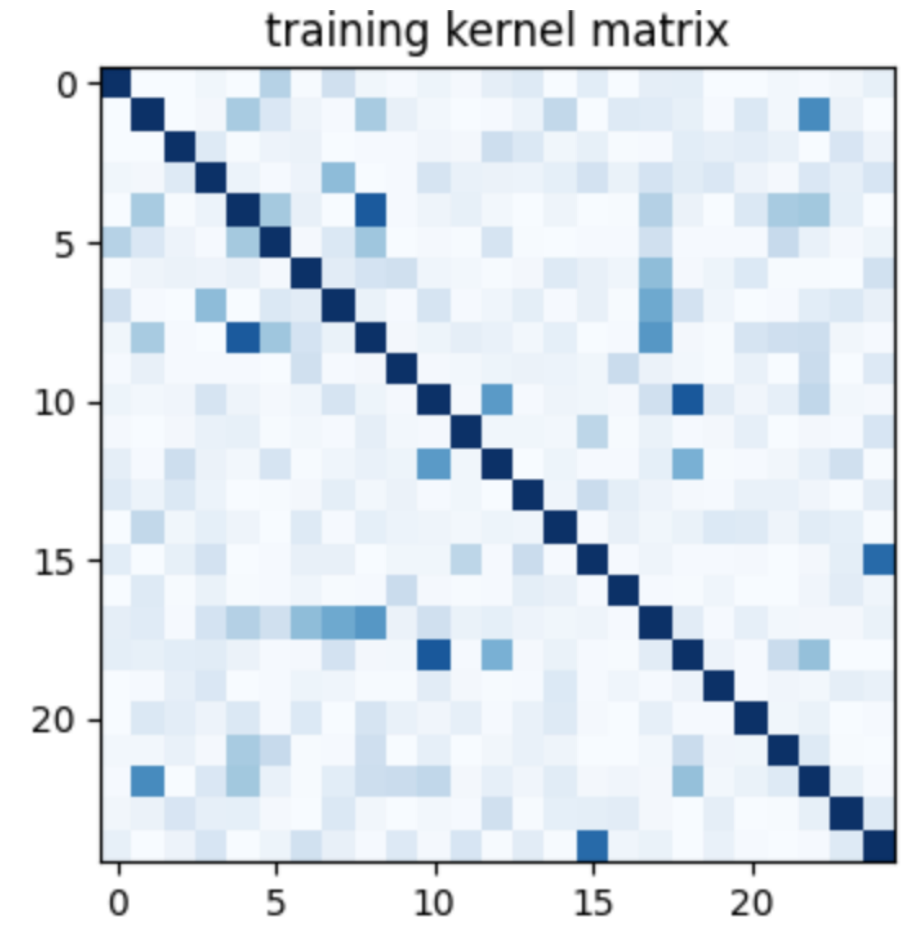
全く違う特徴：0(白)
同じ特徴：1(濃紺)

量子カーネルに必要な計算回数

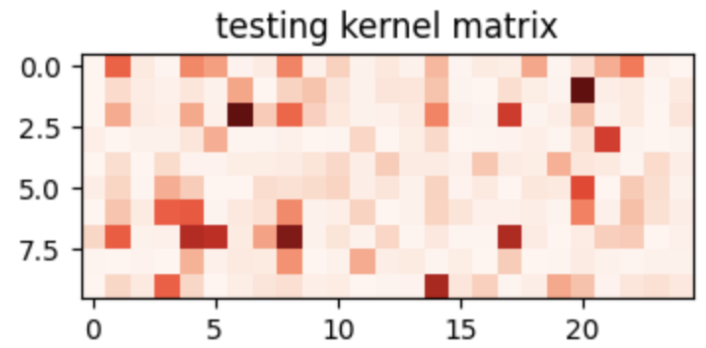
60個の学習データと20個のテストデータを使う場合、
必要な計算は
- カーネル行列: 60 x 60
- テスト行列: 20 x 60

ハンズオンでは、カーネル行列に対して簡略化を行い 、
以下の計算をします：
- 対称行列のため右上だけ計算（ 60 x 60 x ½ ）
- 対角要素は1とする

60 x 60 matrix

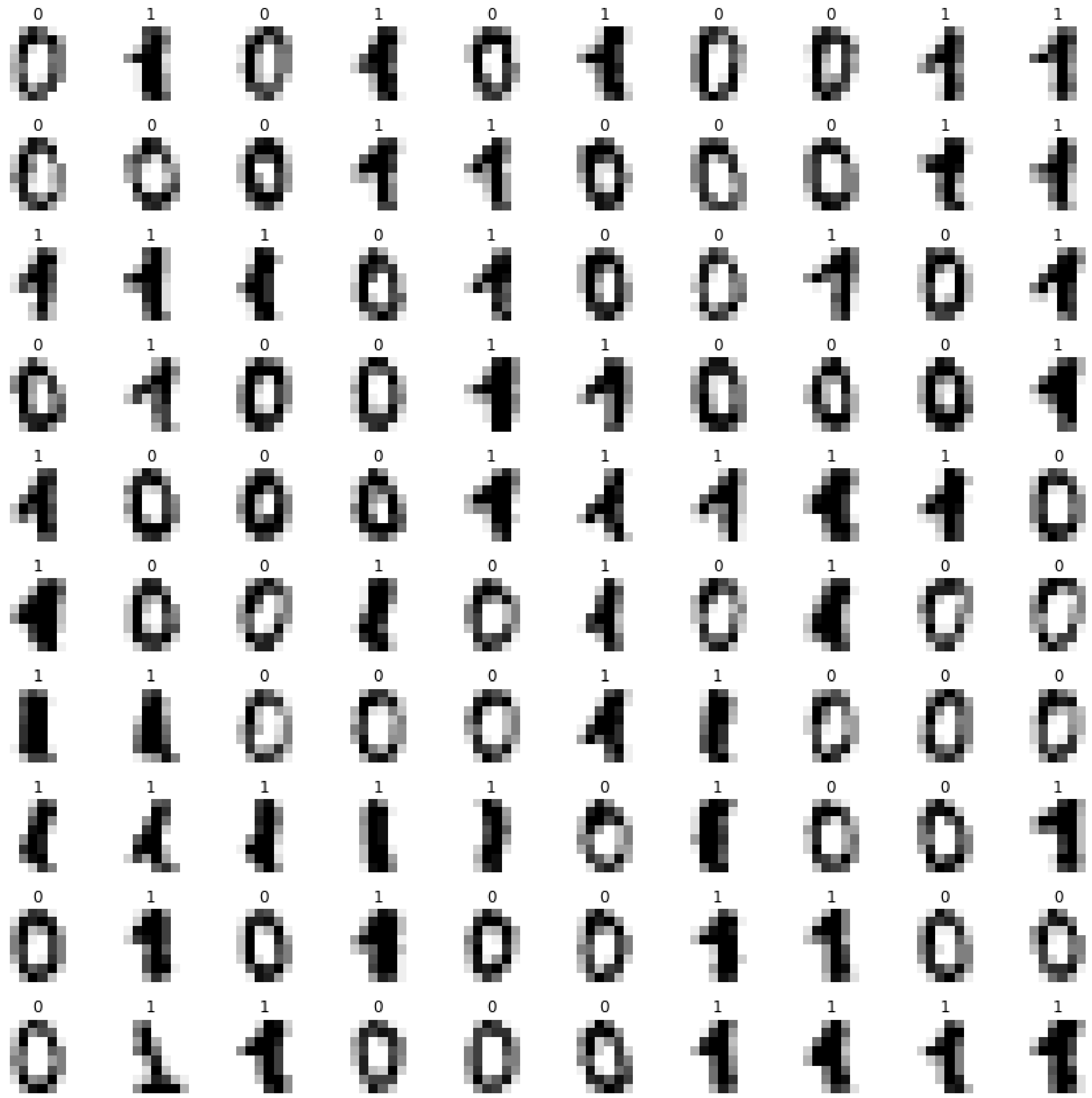


20 x 60 matrix



演習：

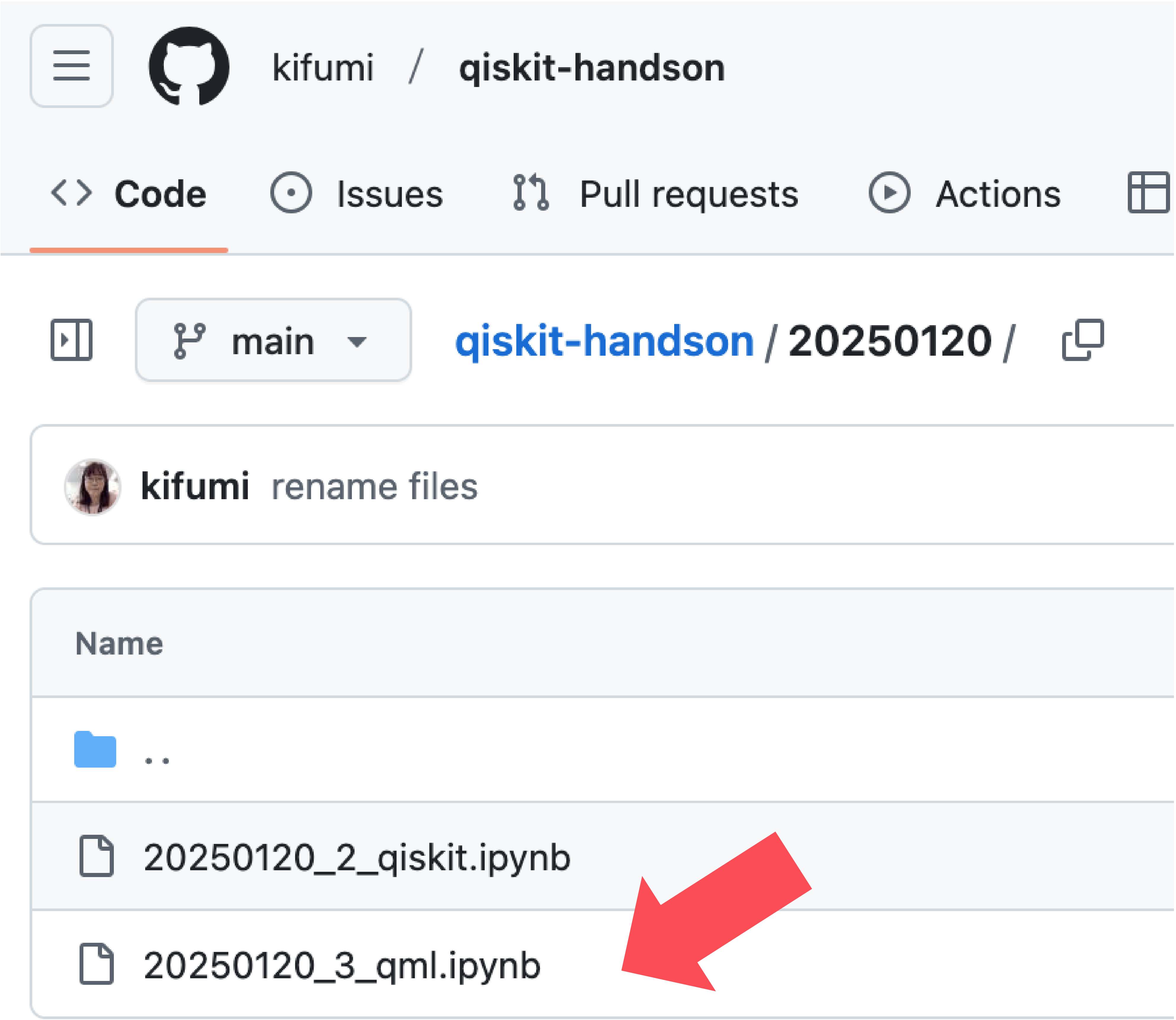
量子カーネル機械学習の実装を学んだ後、手書き数字（MNIST）の分類を行います。

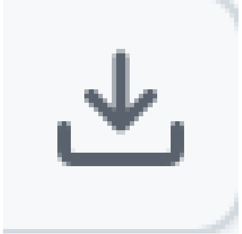


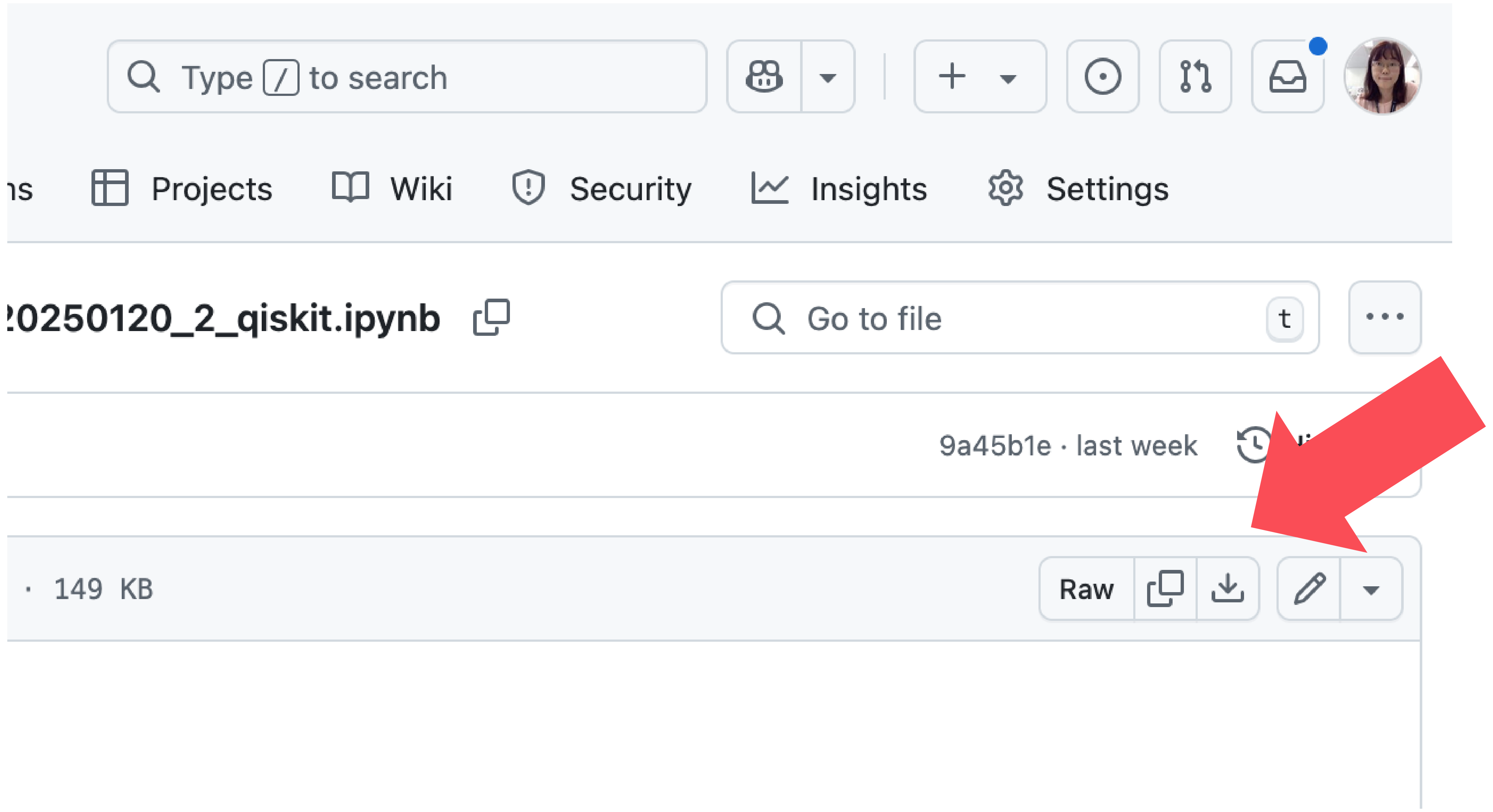
ファイルをダウンロード

URL: [ibm.biz/20240120](ibm.biz/20240120)

(1)「20250120_3_qml/ipynb」を選択



(2)右上の  アイコンからファイルをローカルにダウンロード

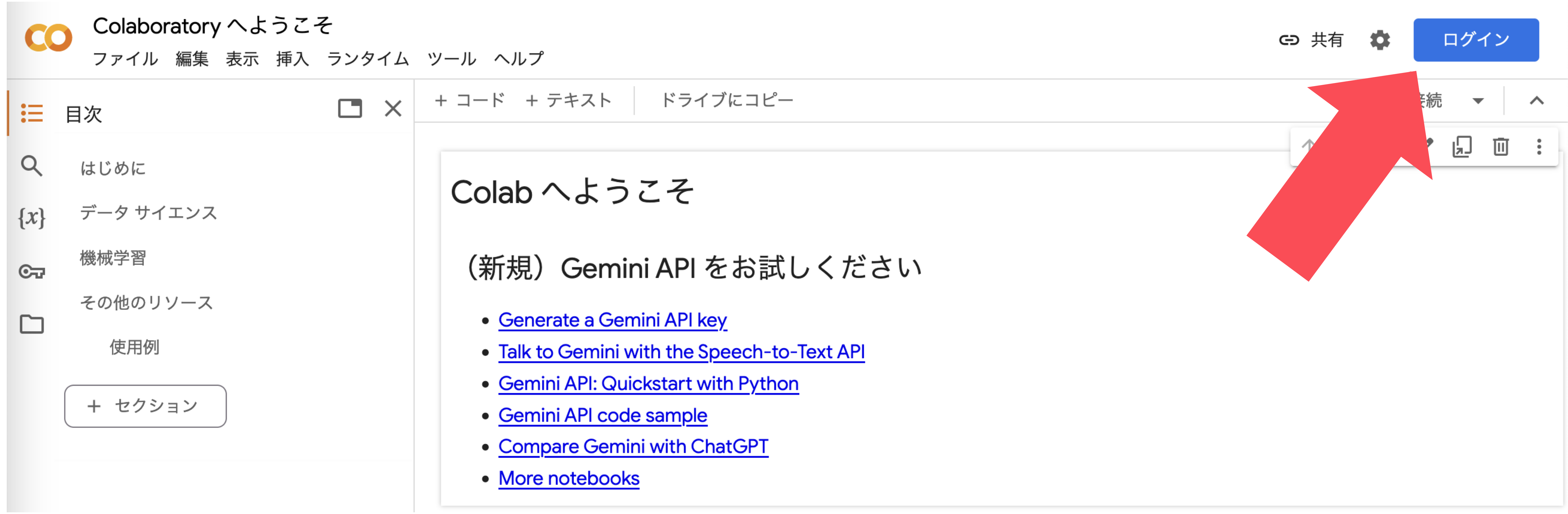


(3)ブラウザーで1つ前のページに戻ってGoogle ColabまたはqBraidの準備をする。

Google Colab または qBraid Lab を使ってQiskitを実行

(1) Google コラボにログイン

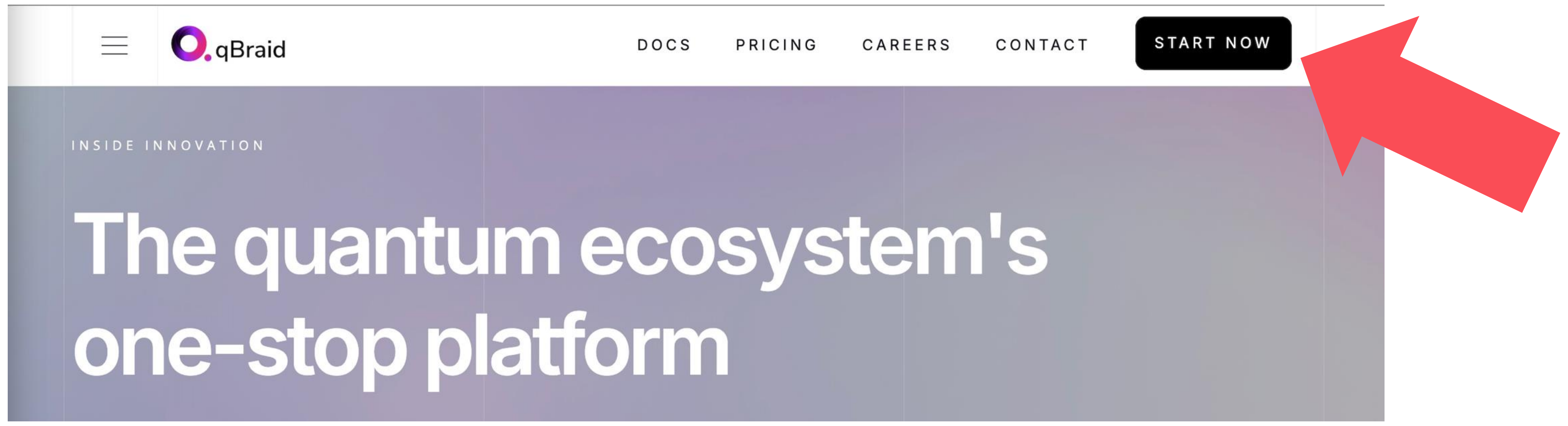
https://colab.research.google.com/



(2) 「ファイル」→「ノートブックをアップロード」

(1) qBraidにログイン

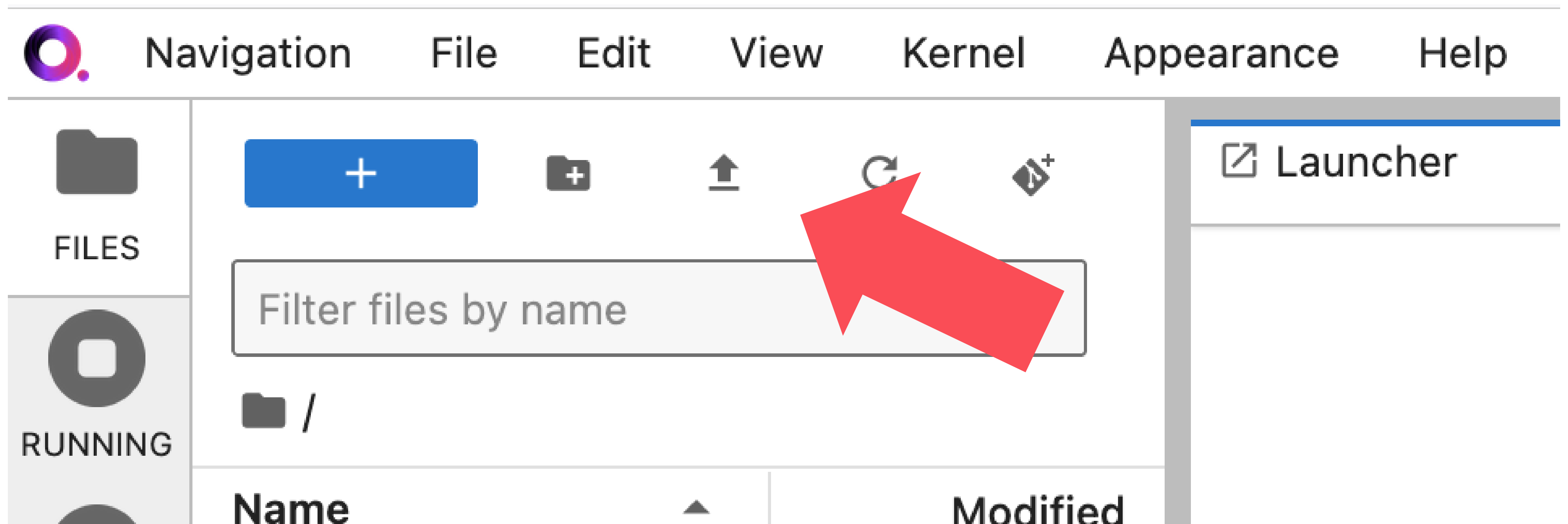
https://www.qbraid.com/



(2) 手順に従って環境作成

URL: ibm.biz/qbraidja

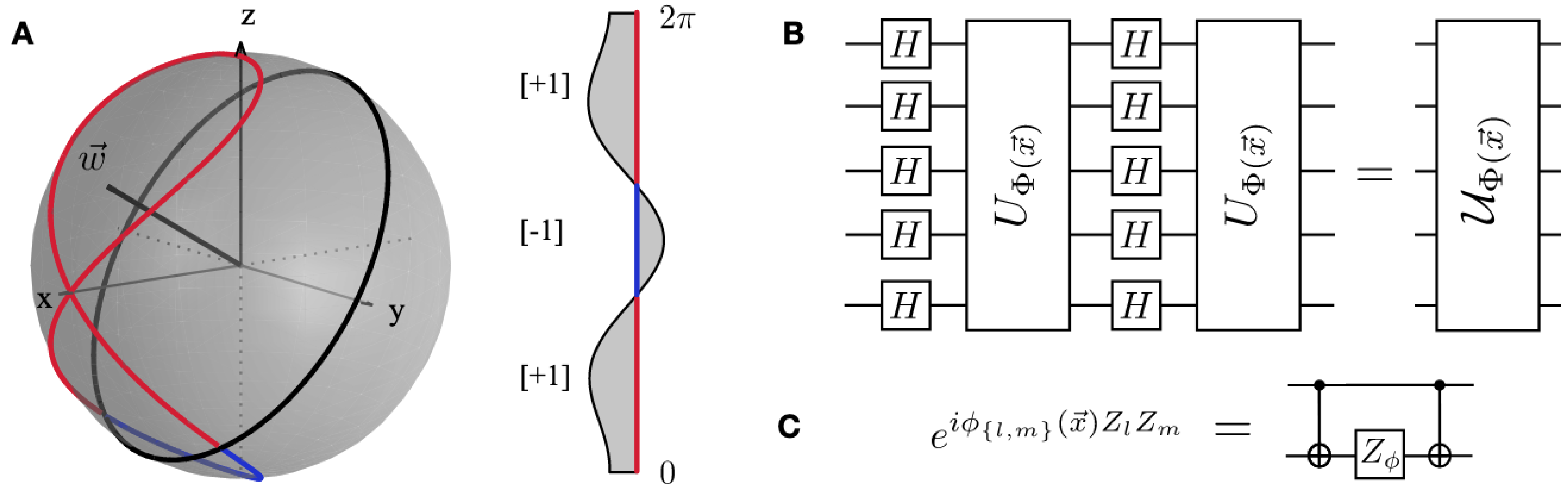
(3) 左上の マークからファイルをアップロード

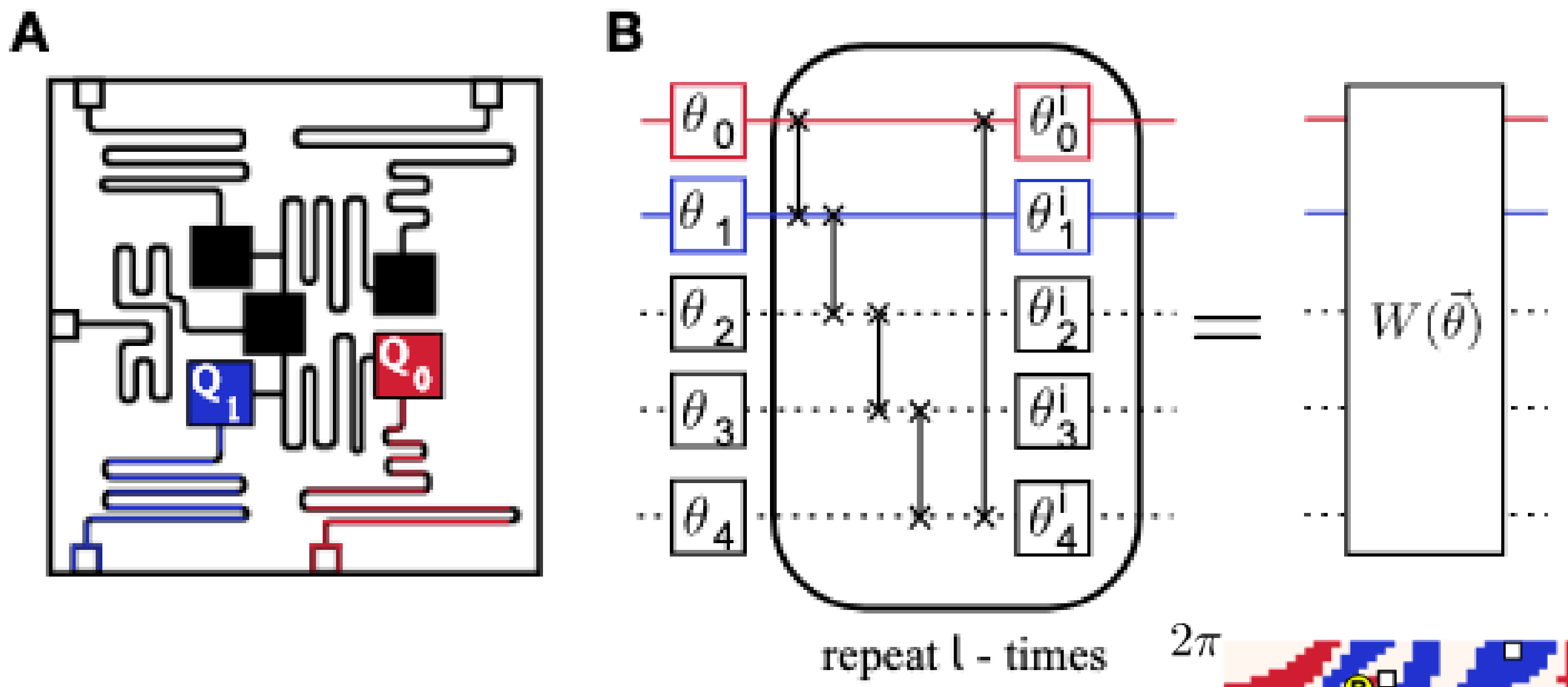


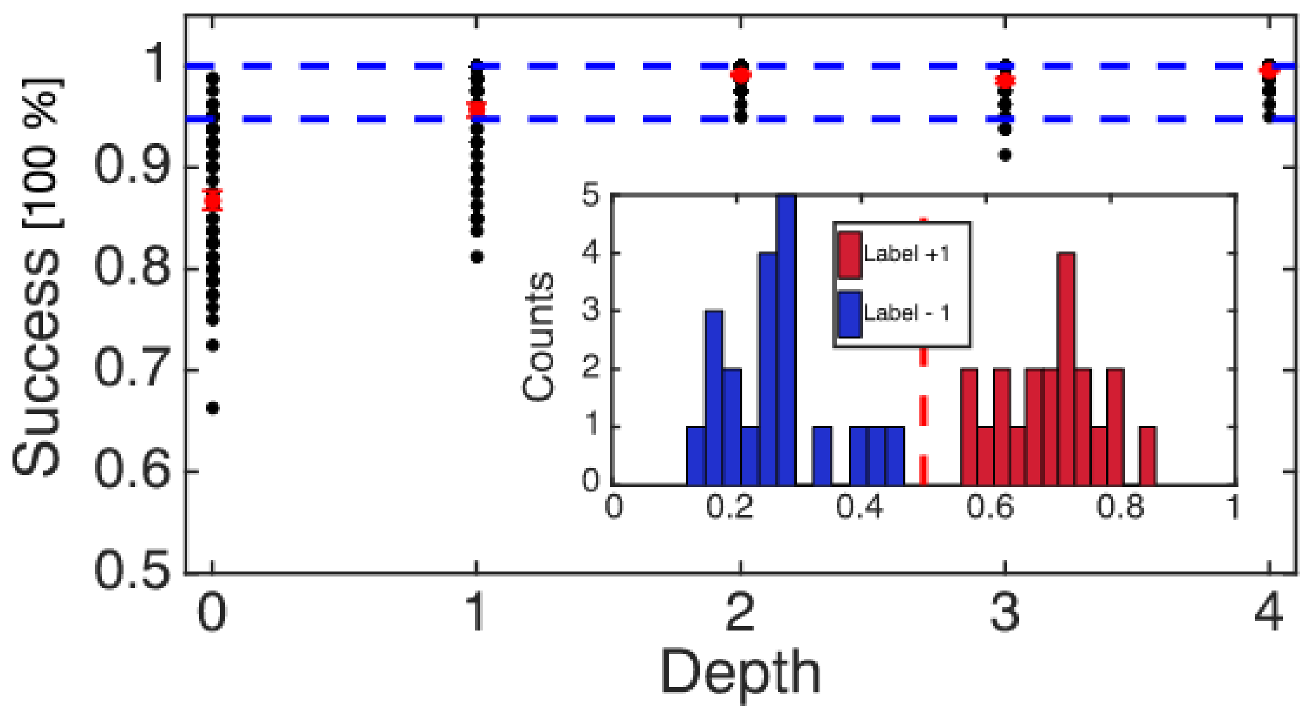
Letter | Published: 13 March 2019

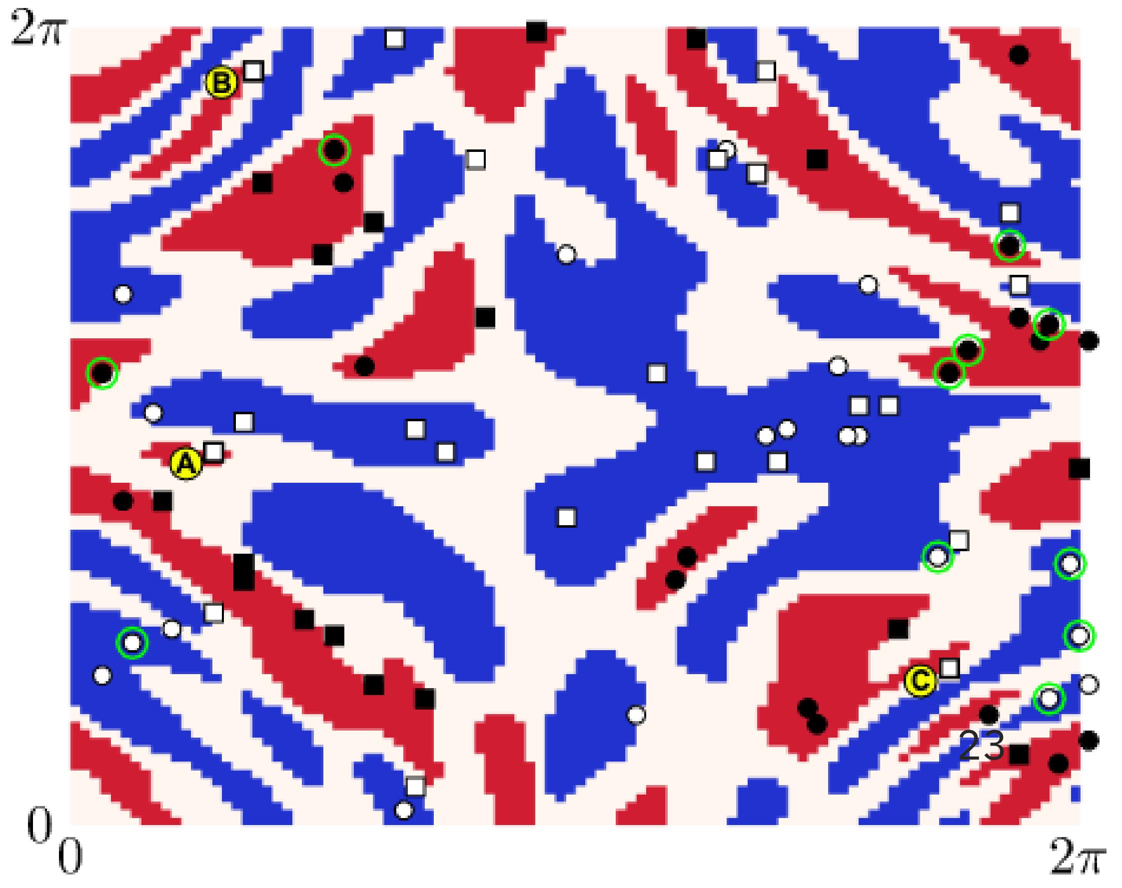
Supervised learning with quantum-enhanced feature spaces

Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow & Jay M. Gambetta









本日のアジェンダ

13:00-13:05	ご挨拶
13:05-14:00	**量子コンピューター入門**
14:00-14:45	**Qiskit入門**
14:45-15:00	休憩、および、シャンデリア見学
15:00-16:00	**量子機械学習入門**
16:00-16:30	ThinkLab 半導体/AI ツアー