# CPEN 400Q Lecture 13
# Variational algorithms: quantum gradients and the variational quantum eigensolver

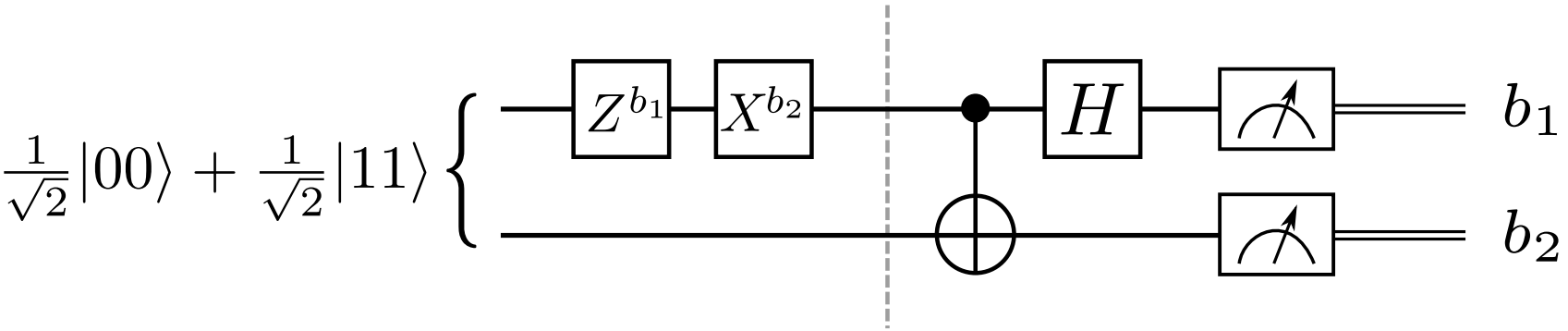
Monday 27 February 2023

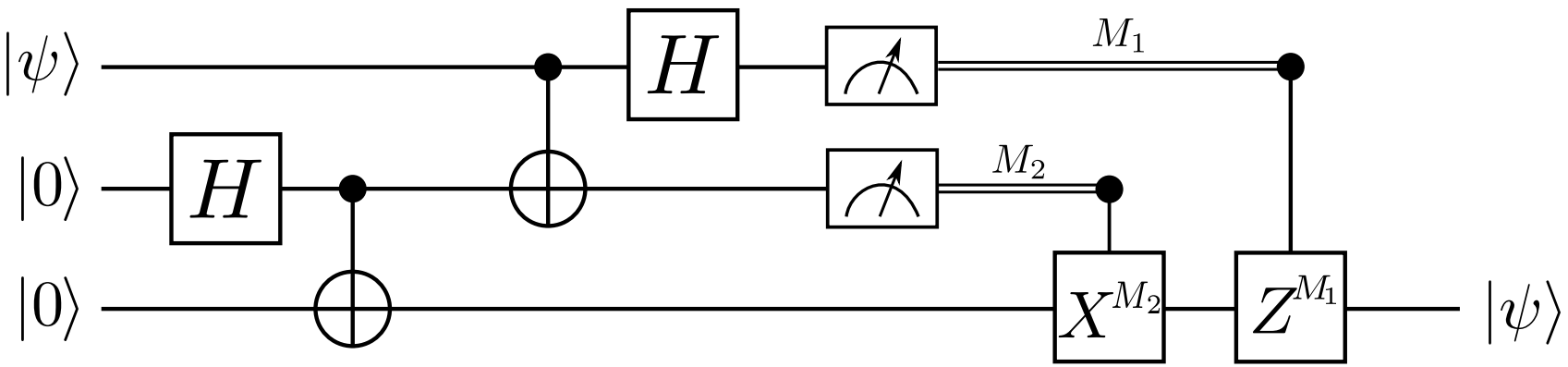
# Announcements

- Literacy assignment 2 due tonight at 23:59
- Updated class schedule:
  - Monday March 6: Zoom
  - Friday March 10: Zoom
  - Monday March 13: in person
  - Friday March 17: pre-recorded "infotainment" lecture about compilation
- Meetings *next week* for project prototypes, on Zoom (schedule selection starting tomorrow 11am on Piazza)

# Where are we going?

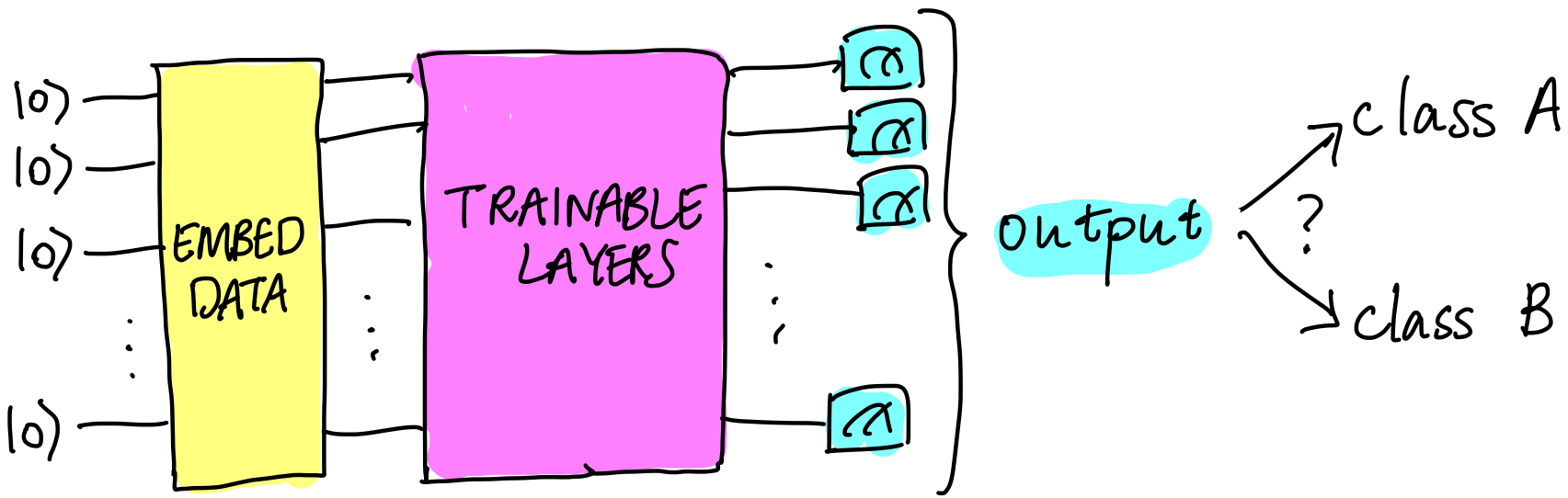
We have seen two seminal, but basic quantum algorithms: superdense coding, and teleportation.



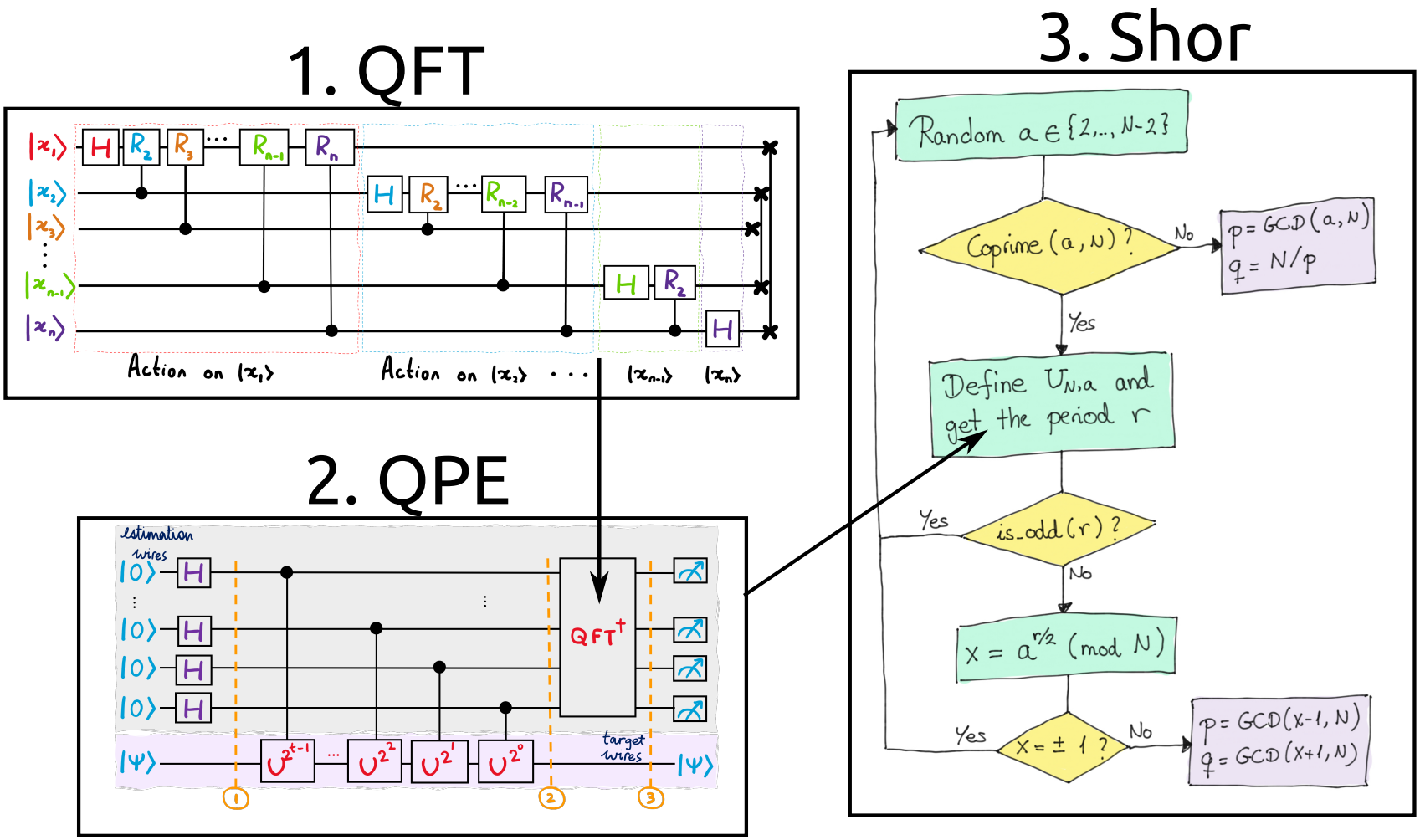
# Where are we going?

We saw a near-term-friendly algorithm, the variational classifier.



# Where are we going?

We implemented a large-scale quantum algorithm that we will be able to run on hardware in the future: Shor's algorithm.



# Where are we going?

Next 2-3 classes: back to variational algorithms

- Formalism of quantum gradients
- Hamiltonians and cost functions
- Variational quantum eigensolver
- Quantum approximate optimization algorithm (QAOA)

Subsequent 2-3 classes: the pitfalls of near-term hardware

- Mixed states, quantum channels, and noise
- Basics of quantum error mitigation

Last couple classes:

- Oracles and complexity
- Oracle-based algorithms: Deutsch, Deutsch-Josza, amplitude amplification, Grover search

# Learning outcomes

- State the parameter-shift rule and use it to compute the gradient of a quantum circuit
- Define a Hamiltonian
- Describe the variational quantum eigensolver algorithm and apply it to compute the ground state energy of a simple physical system

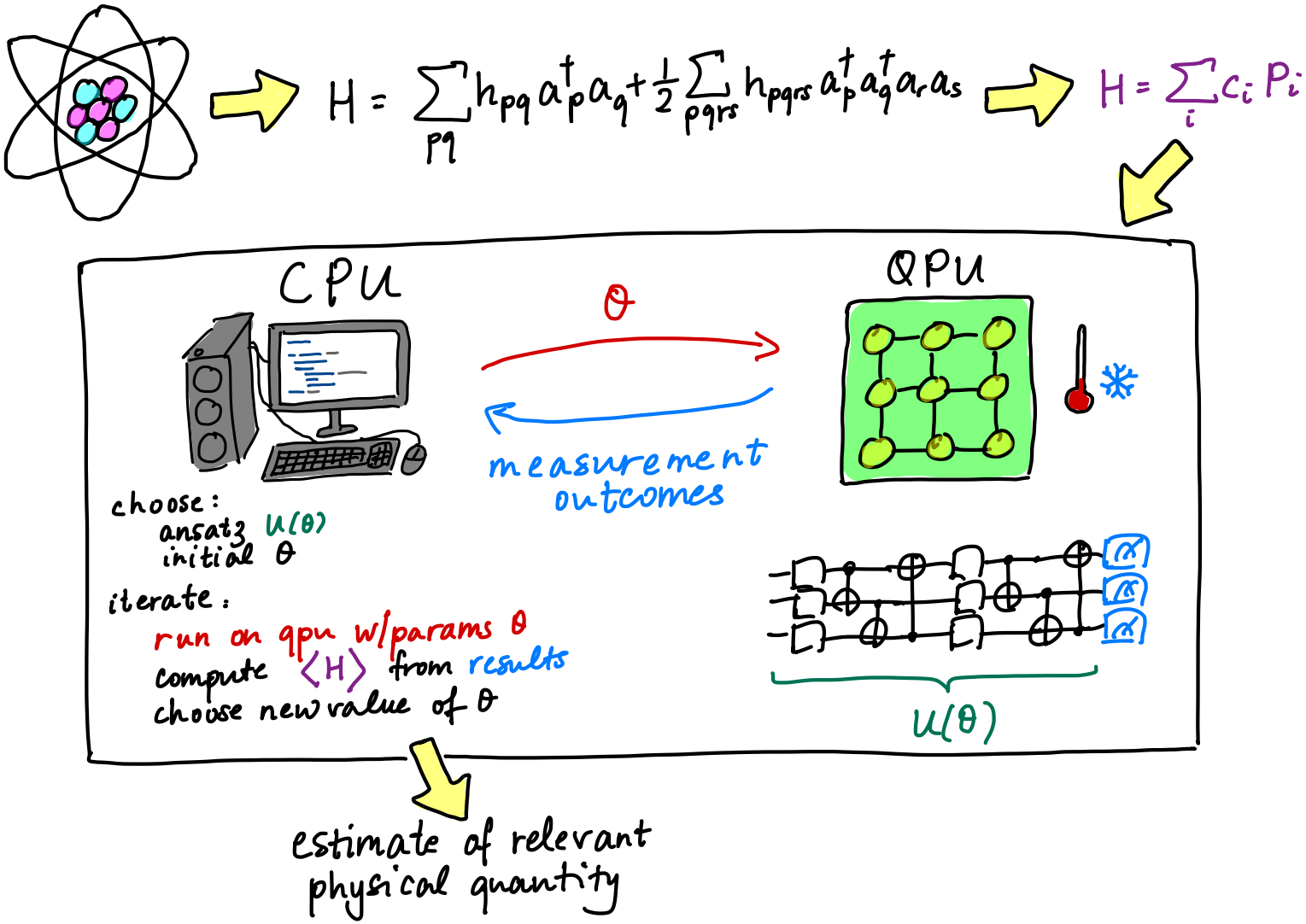
# Why variational algorithms?

What *can* we do with a NISQ device?

Suitable algorithms should:

- Not be too long
- Fit the processor architecture well
- Use a quantum computer to do something non-trivial
- Still solve an interesting problem

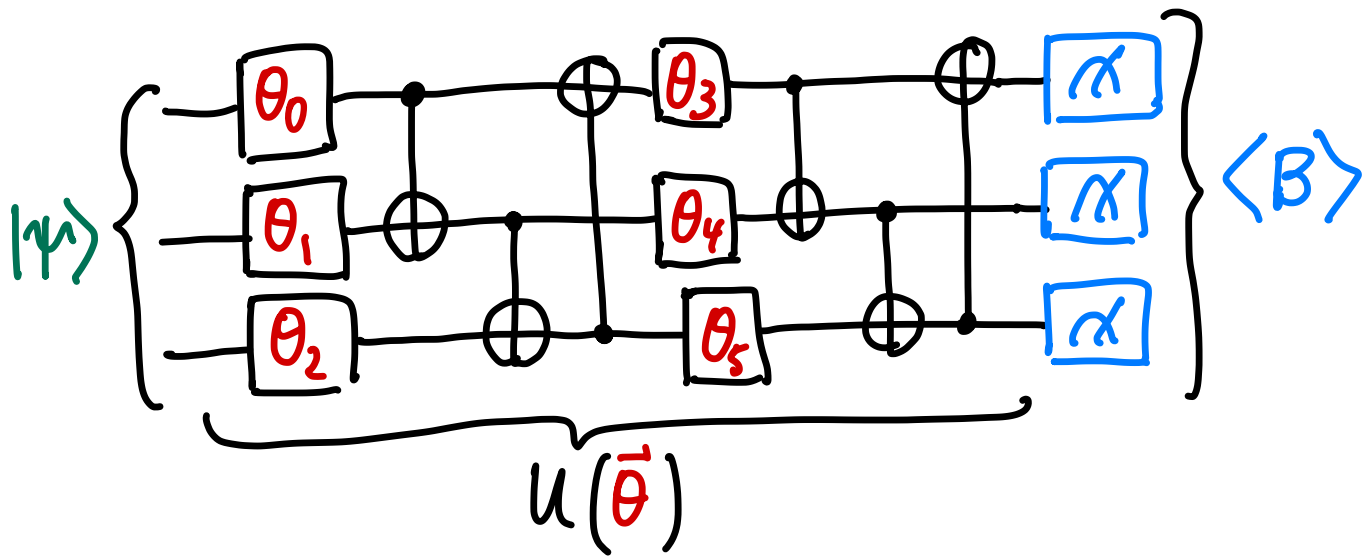
# Variational algorithms



# Parametrized quantum circuits

Find *optimal values* for circuit parameters in order to minimize a cost function that represents the solution to a problem.

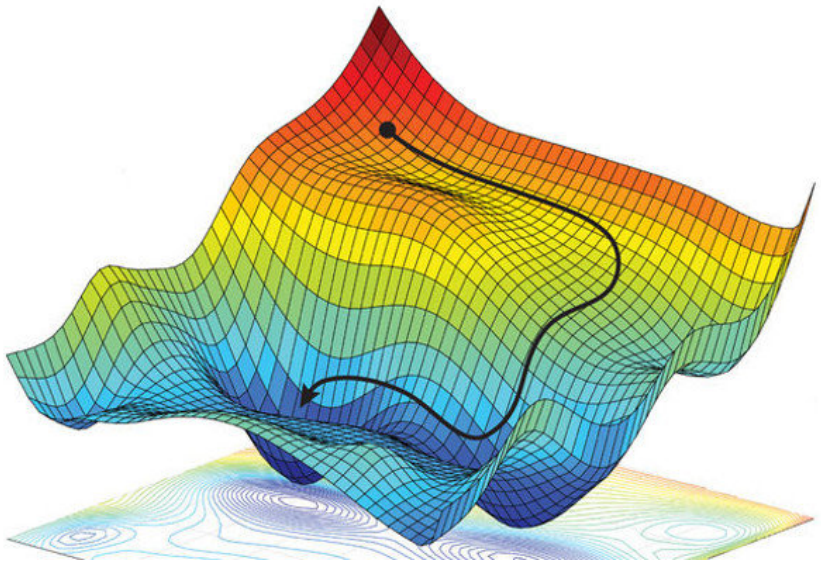
In VQC, we used expectation values to construct a cost function.




$$\min_{\vec{\theta}} \langle B \rangle = \min_{\vec{\theta}} \langle \psi | U^{\dagger}(\vec{\theta}) B U(\vec{\theta}) | \psi \rangle$$

# Training variational quantum circuits

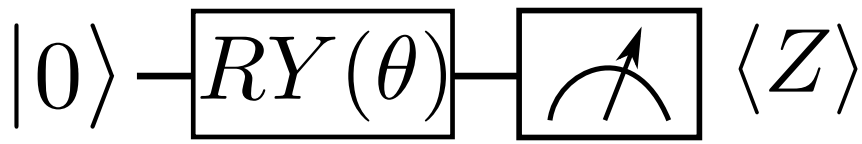
Circuits can be trained using standard optimization techniques *on a classical computer* such as gradient descent.



... but how do we compute the gradient of a quantum circuit?

Image credit: A. Amini, A. P. Soleimany, S. Karaman, D. Rus. *Spatial Uncertainty Sampling for End-to-End Control*. NIPS 2017.

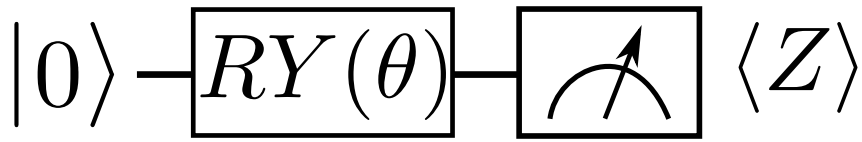
# Gradients of quantum circuits




Key point: the expectation values measured at the end are *functions* of the variational parameters, i.e.,

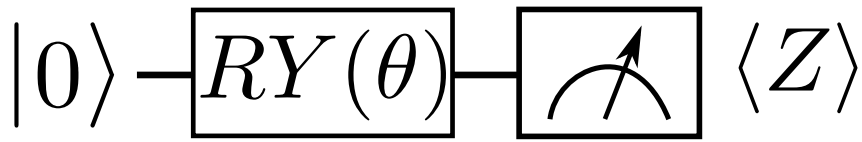
We can compute such functions, then differentiate them.

# Gradients of quantum circuits




You can compute the analytical expression for $\langle Z \rangle$:

# Gradients of quantum circuits




We can compute the derivative of the function!

But obviously, we don't want to do this by hand... use automatic differentiation instead! PennyLane will do this for us.

# qml.grad

qml.grad is a *transform*: apply to a QNode to obtain a function that computes the *gradient* of that QNode.

```
@qml.qnode(dev)
def pqc(theta):
    qml.RY(theta, wires=0)
    return qml.expval(qml.PauliZ(0))

grad_fn = qml.grad(pqc)
grad_fn(theta)
```

# Gradients of quantum circuits

Easy to do in software, but what about on hardware?

To train using gradient descent, we'd have to:

1. guess an initial value for $\theta$
2. run a circuit that computes the gradient at $\theta$
3. use those results to produce an updated value for $\theta$
4. repeat 2-3 until converged

What circuit should we run in step 2?

# The parameter-shift rule

Our circuit implements the function

The gradient of this function is

Consider the following:

# The parameter-shift rule

Simplify by noting that

Then:

# The parameter-shift rule

This is an example of a parameter-shift rule:

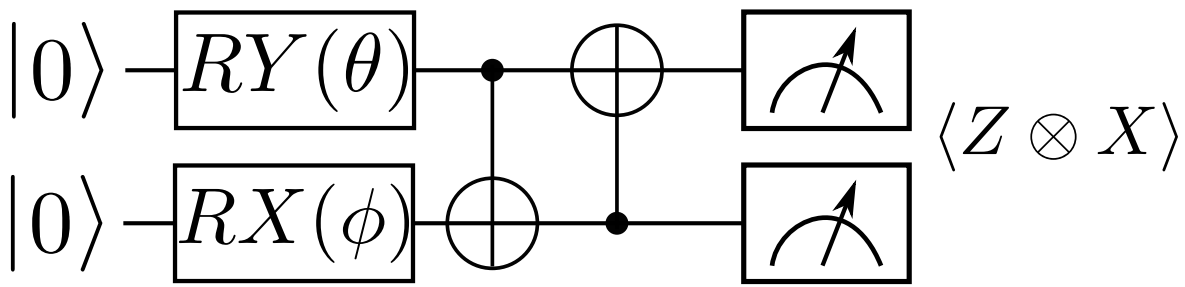
We can compute gradients with respect to parameters of a circuit by *evaluating them at shifted versions of those parameters*!

# The parameter-shift rule

More generally, for all single-qubit rotation gates $U(\theta)$[1],

where $f(\theta)$ is the function implemented by the *whole circuit*, with every other parameter held constant.

Let's try an example with more than one parameter.

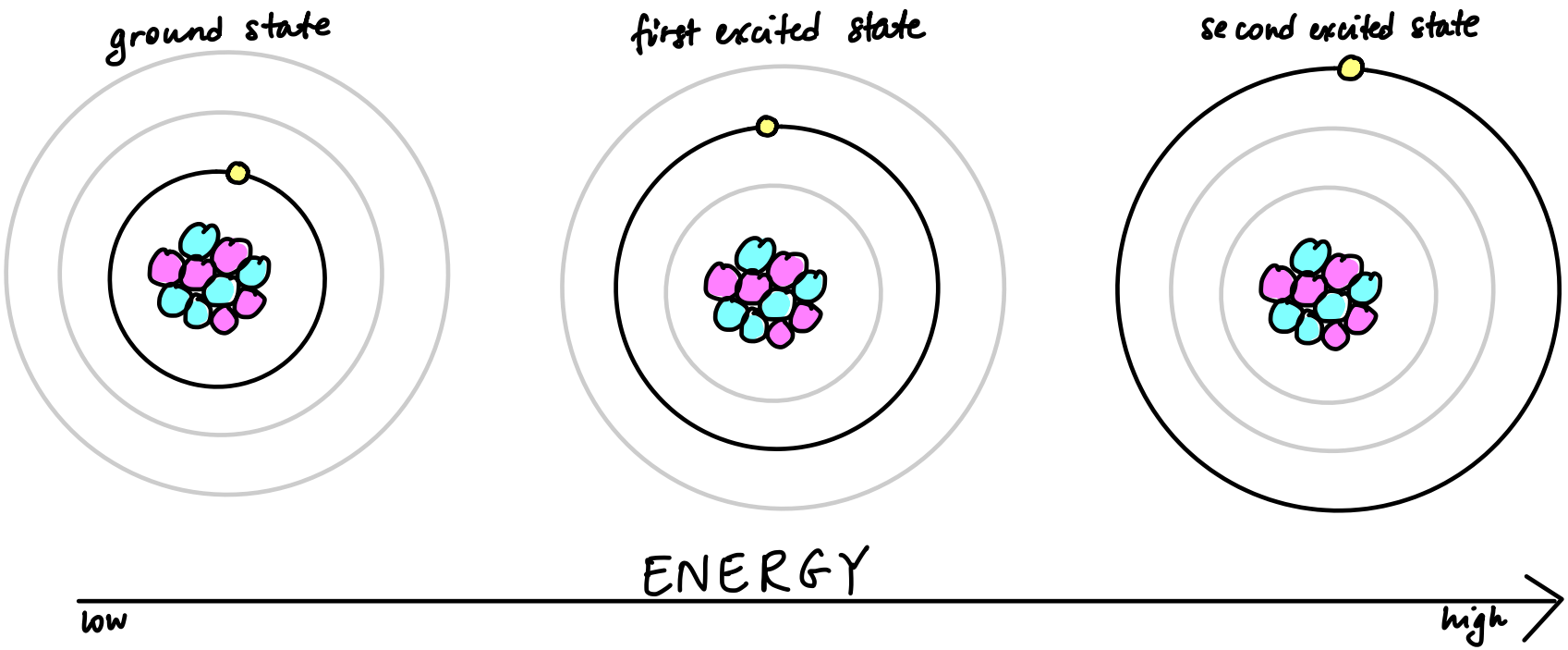



[1] If you want to see a full derivation, I've included one at the end of the lecture slides.

# Variational quantum eigensolver

# Energy of a physical system

Determining the energy levels of physical systems is relevant for applications of quantum computing like quantum chemistry.




Energy levels allow us to compute useful properties of molecules (e.g., activation energies and reaction rates).

# Hamiltonians

In physics, a special operator called a Hamiltonian, $\hat{H}$, describes the energy of a system. Hamiltonians are *Hermitian* operators.

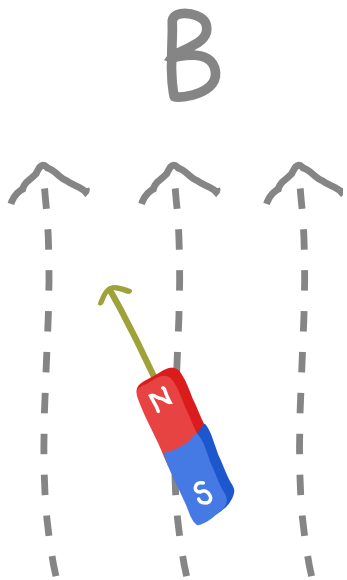
The Hamiltonian can be used to determine how a system evolves over time. If $|\psi(0)\rangle$ is the state of the system at time $t = 0$,

This comes from the Schrödinger equation in physics.

(If $\hat{H}$ is Hermitian, then $e^{-i\hat{H}t}$ is unitary, so we can run this on a quantum computer! This is called *Hamiltonian simulation*; we will cover this in the course if time permits)

# Hamiltonians

Example: a bar magnet in a magnetic field along the $Z$ direction.

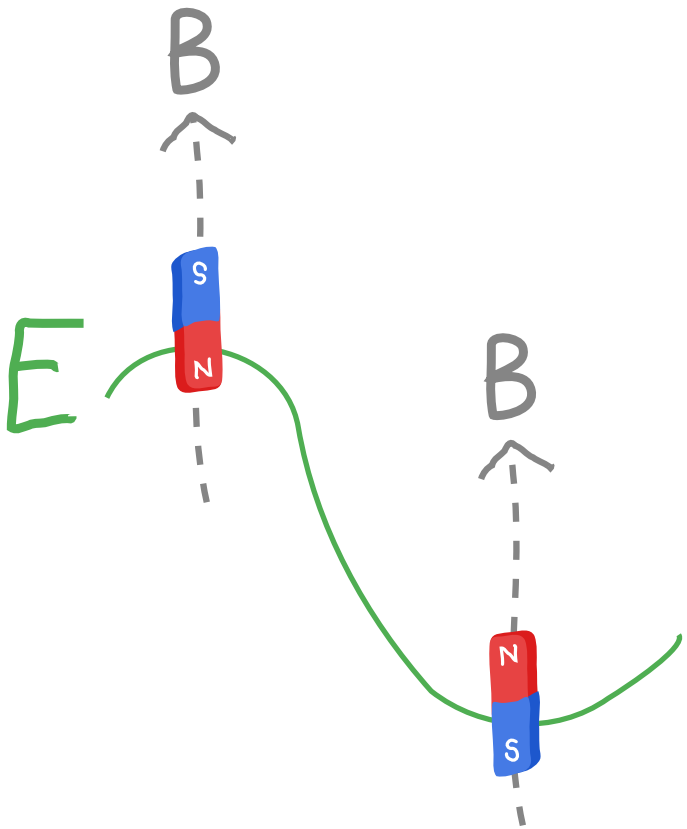



Every orientation of the magnet has an energy associated with it.

Image credit: Xanadu Quantum Codebook node H.3

# Mathematical description of energy

Bar magnets like to *align* with the external field.



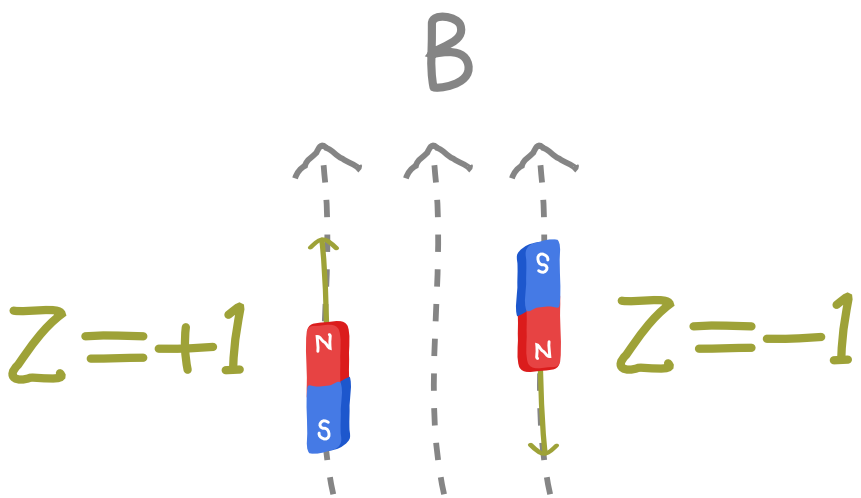

Aligned configuration is the *ground state* (lowest energy state).

Image credit: Xanadu Quantum Codebook node H.3

# Hamiltonians

There is a *Hamiltonian* that describes the energy of the system:

where $\alpha$ is a coefficient that depends on some physical values.

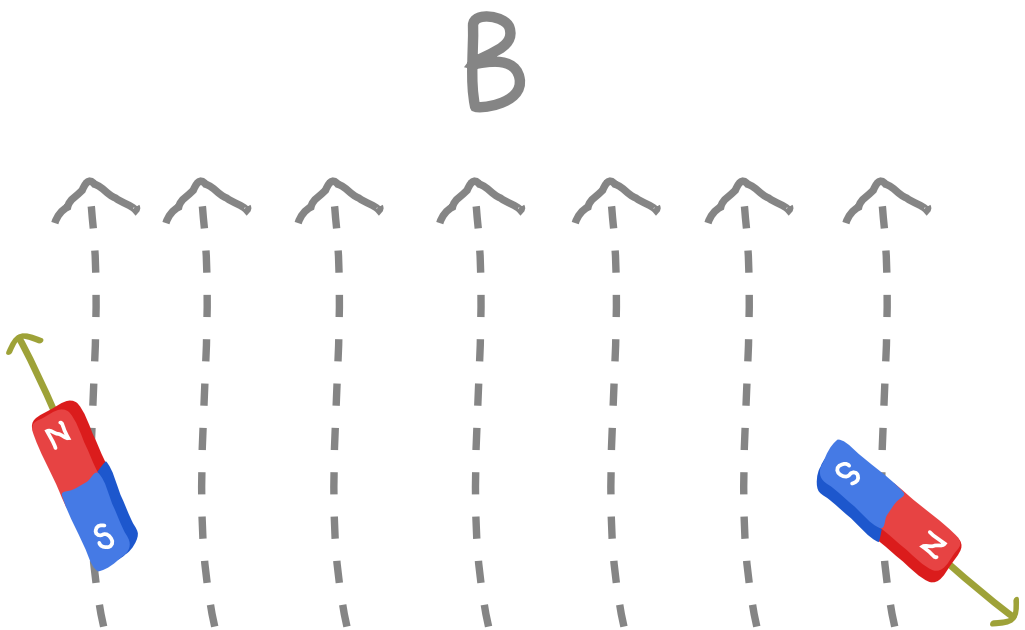



Computing the energy of the system corresponds to *measuring the expectation value of* $\hat{H}$ (which in this case, is just $-\alpha\langle Z\rangle$).

Image credit: Xanadu Quantum Codebook node H.3

# Hamiltonians

Hamiltonians can be used to describe systems with multiple parts. For example, two magnets that are far away have the Hamiltonian




The energy of this system is .

Image credit: Xanadu Quantum Codebook node H.5

# Hamiltonians

Two magnets that are close together may have different terms in the Hamiltonian that describe *how they interact*:

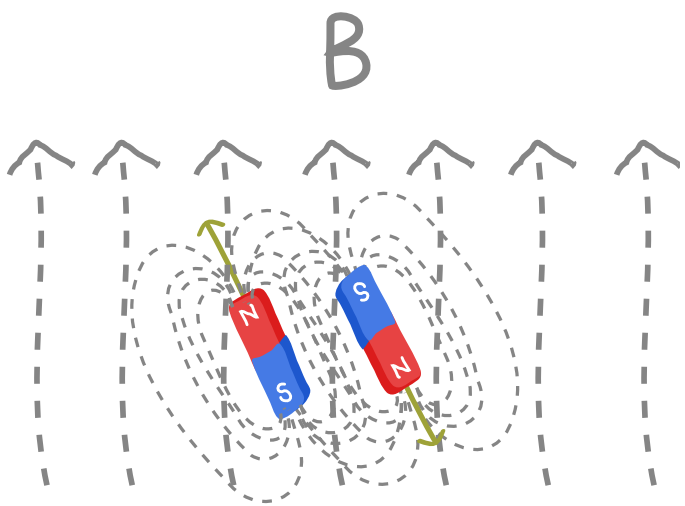



Image credit: Xanadu Quantum Codebook node H.5

# Hamiltonians

More generally, any Hermitian Hamiltonian can be expressed as

Then, computation of expectation values is linear:

# Computing the energy of a Hamiltonian

Given an arbitrary $\hat{H}$, how can we compute its energy?

Option: *exact diagonalization* to find its eigenvalues.


Many other purely classical methods too. But we're more interested in the quantum ones.

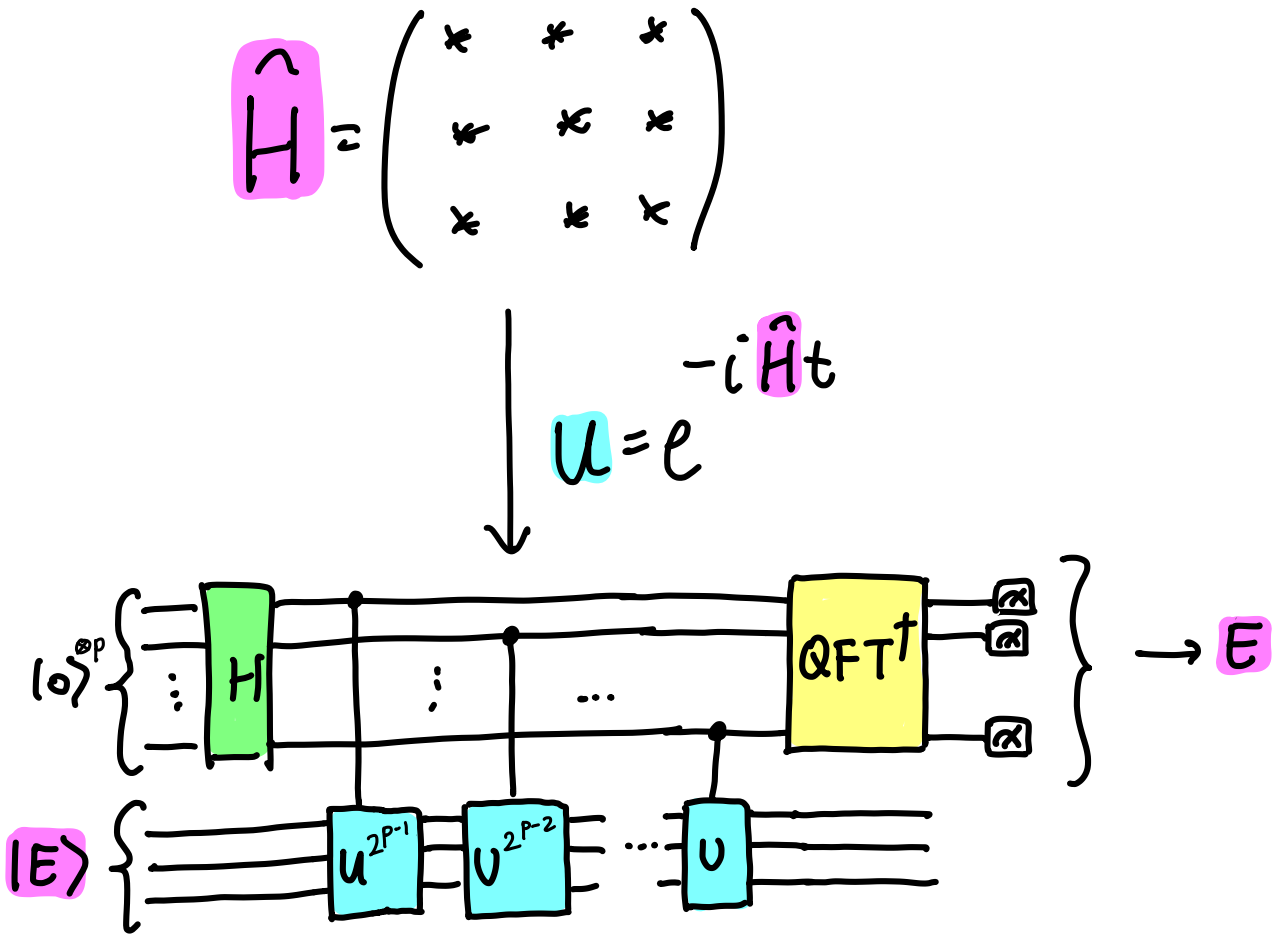
# Computing the energy of a Hamiltonian

Let $\hat{H}$ be a Hamiltonian and $|E\rangle$ an eigenstate with energy $E$. If $U = e^{-i\hat{H}t}$,

So is an eigenvalue of $U$... look familiar?

# Computing the energy of a Hamiltonian

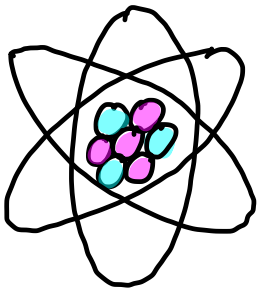
Option: *quantum phase estimation*.




But this is not practical on near-term quantum devices.

# Computing the energy of a Hamiltonian

Option: *variational quantum eigensolver* (VQE).



$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \quad \Rightarrow \quad H = \sum_i c_i P_i$$

CPU

QPU

$\theta$

measurement outcomes

choose:
ansatz $U(\theta)$
initial $\theta$

iterate:
run on qpu w/params $\theta$
compute $\langle H \rangle$ from results
choose new value of $\theta$

$U(\theta)$

estimate of relevant physical quantity

# Variational quantum eigensolver (VQE)

The VQE is a variational algorithm that works by:

- training a parametrized circuit to prepare the ground state of a system
- measuring the expectation value of the problem Hamiltonian to compute the energy

The optimization works because of the *variational principle*. If $E_g$ is the ground state energy, then

for *any other state* $|\psi\rangle$.

# VQE example

Suppose we want to *learn* the ground state energy of

$$\hat{H} = 0.5Z \otimes Z + 1.5X \otimes X$$

Since this is a small system, we can work it out directly.

$$E_0 = -2, \quad E_1 = -1, \quad E_2 = 1, \quad E_3 = 2$$

$$|E_0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \quad |E_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad |E_2\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ -1 \\ 0 \end{pmatrix}, \quad |E_3\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

# VQE example

We are lucky here: we know that the ground state is

So we need a parametrized circuit $U(\theta)$ that can produce arbitrary combinations of $|01\rangle$ and $|10\rangle$:

# VQE example

We can make a very tailored ansatz with just a single parameter here. Let

$$U(\theta)|00\rangle = \alpha|01\rangle + \beta|10\rangle$$

Since we know our ground state has real-valued amplitudes, write

What circuit makes states like this?

# Next time

Content:

- Solving combinatorial optimization problems with QAOA

Action items:

1. Finish literacy assignment 2
2. Work on prototype implementation for project

Recommended reading:

- QML glossary entries (`https://pennylane.ai/qml/glossary.html`):
  - Quantum differentiable programming
  - Parameter-shift rules
  - Quantum gradients
  - Variational circuit
- `https://arxiv.org/abs/2012.09265v2` (review paper)

# The two-term parameter-shift rule

This kind of seems like magic... where does it come from?

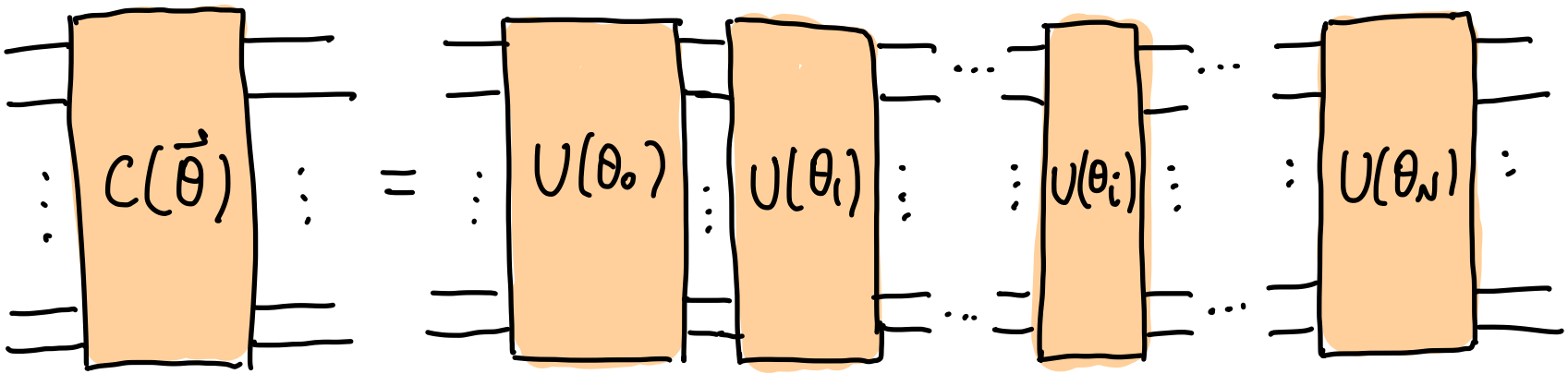
$$\frac{\partial f(\theta)}{\partial \theta} = \frac{1}{2}\left(f(\theta + \pi/2) - f(\theta - \pi/2)\right)$$

We will derive this for the case of *single-qubit Pauli rotations* (which, if you recall, are universal for single-qubit operations).

The derivation can also be done in a more general way (`https://arxiv.org/abs/1811.11184`).

# The two-term parameter-shift rule

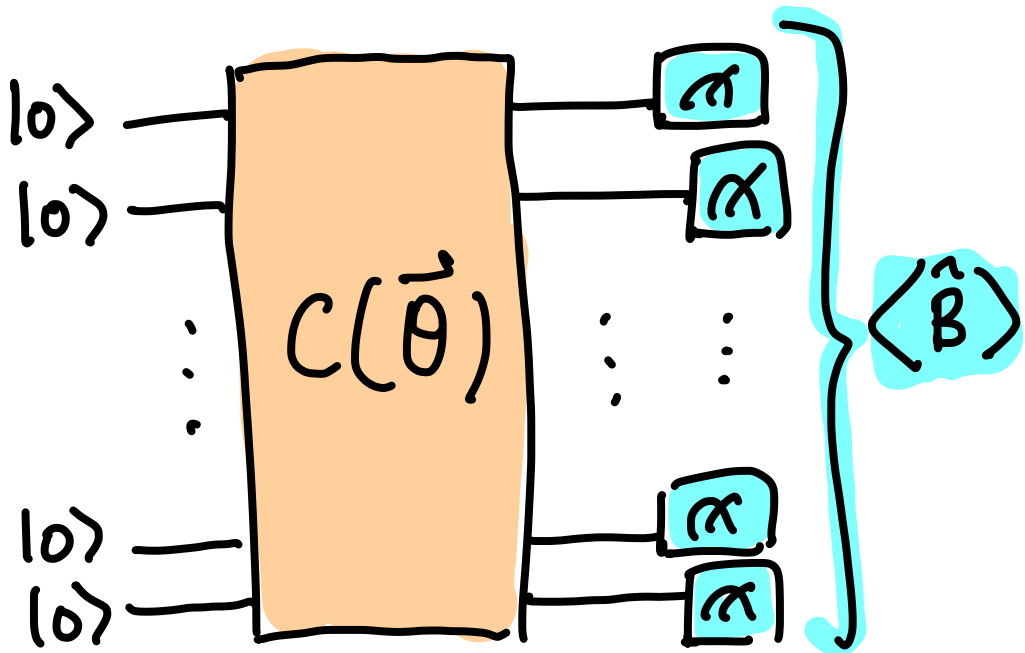
Let $C(\theta) = C(\theta_0, \theta_1, \ldots, \theta_N)$ be a quantum circuit.




Suppose we want to compute the gradient with respect to $\theta_i$.

# The two-term parameter-shift rule

We differentiate the *expectation value* of some observable, $\hat{B}$, as a function of the circuit parameters.

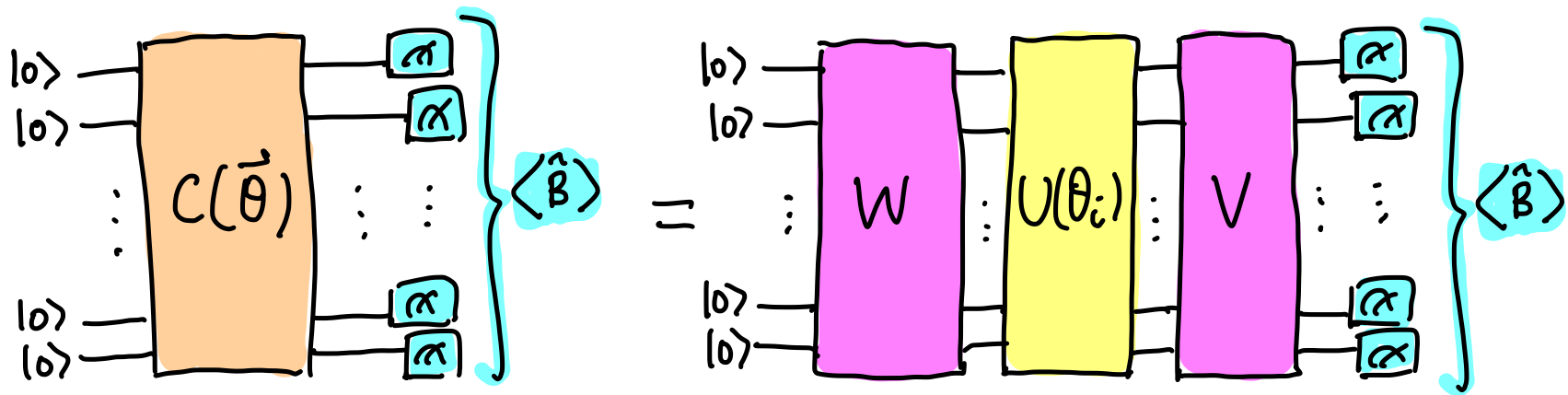



$$f(\theta) = \langle 0|\, C^\dagger(\theta)\ \ \hat{B}\ \ C(\theta)|0\rangle$$

We want to compute $\frac{\partial f(\theta)}{\partial \theta_i}$.

# The two-term parameter-shift rule

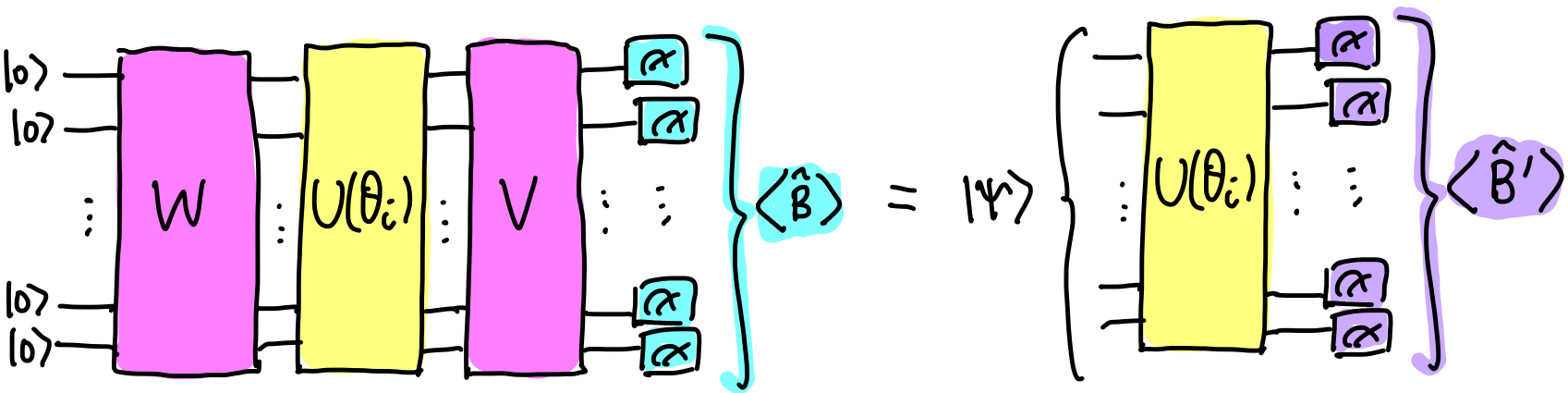
Let's isolate the parameter of interest:




$$f(\theta) = \langle 0| \, W^\dagger U^\dagger(\theta_i) V^\dagger \;\; \hat{B} \;\; V U(\theta_i) W |0\rangle$$

# The two-term parameter-shift rule

Let's also tidy things up and group things that don't depend on $\theta_i$:




$$f(\theta) = \langle\psi|\, U^\dagger(\theta_i)\;\; \hat{B}'\;\; U(\theta_i)|\psi\rangle$$

where $|\psi\rangle = W|0\rangle$, and $\hat{B}' = V^\dagger \hat{B} V$ is still a Hermitian observable.

# The two-term parameter-shift rule

Take the partial derivative of this function using the *chain rule*:

$$\frac{\partial f(\theta)}{\partial \theta_i} = \langle\psi| \frac{\partial U^\dagger(\theta_i)}{\partial \theta_i} \hat{B}' U(\theta_i)|\psi\rangle + \langle\psi| U^\dagger(\theta_i)\hat{B}' \frac{\partial U(\theta_i)}{\partial \theta_i}|\psi\rangle$$

Note that the two terms are Hermitian conjugates, i.e.,

$$\left(\langle\psi| \frac{\partial U^\dagger(\theta_i)}{\partial \theta_i} \hat{B}' U(\theta_i)|\psi\rangle\right)^\dagger = \langle\psi| U^\dagger(\theta_i)\hat{B}' \frac{\partial U(\theta_i)}{\partial \theta_i}|\psi\rangle$$

# The two-term parameter-shift rule

A general property of unitary operations is that they can be expressed in terms of a Hermitian generator, i.e.,

$$U = e^{-iGt}$$

for $G$ Hermitian, $t$ some real-valued coefficient.

We know this is true for the Pauli rotations:

$$RX(\theta) = e^{-i\frac{\theta}{2}X}, \quad RY(\theta) = e^{-i\frac{\theta}{2}Y}, \quad RZ(\theta) = e^{-i\frac{\theta}{2}Z}.$$

# The two-term parameter-shift rule

Consider that our $U(\theta_i)$ is a Pauli rotation:

$$U(\theta_i) = e^{-i\frac{\theta_i}{2}G} = e^{-i\theta_i\frac{G}{2}}, \quad G \in \{X, Y, Z\}.$$

We can compute the derivative of this operation w.r.t. $\theta_i$:

$$\frac{\partial U(\theta_i)}{\partial \theta_i} = -i\frac{G}{2}e^{-i\theta_i\frac{G}{2}} = -i\frac{G}{2}U(\theta_i).$$

# The two-term parameter-shift rule

Let's put this back in our earlier equation:

$$\begin{aligned}\frac{\partial f(\theta)}{\partial \theta_i} &= \langle\psi| \frac{\partial U^\dagger(\theta_i)}{\partial \theta_i} \hat{B}' U(\theta_i)|\psi\rangle + \langle\psi| U^\dagger(\theta_i)\hat{B}' \frac{\partial U(\theta_i)}{\partial \theta_i}|\psi\rangle \\ &= \langle\psi| U^\dagger(\theta_i) \left(i\frac{G}{2}\right) \hat{B}' U(\theta_i)|\psi\rangle \\ &\quad + \langle\psi| U^\dagger(\theta_i)\hat{B}' \left(-i\frac{G}{2}\right) U(\theta_i)|\psi\rangle\end{aligned}$$

Let's make one more substitution for now, $|\psi'\rangle = U(\theta_i)|\psi\rangle$:

$$\frac{\partial f(\theta)}{\partial \theta_i} = \langle\psi'| \left(i\frac{G}{2}\right) \hat{B}'|\psi'\rangle + \langle\psi'| \hat{B}' \left(-i\frac{G}{2}\right)|\psi'\rangle$$

# The two-term parameter-shift rule

Now we make use of the following identity: for any two operators $P$, $Q$,

$$\langle\psi|\, P^\dagger \hat{M} Q|\psi\rangle + \langle\psi|\, Q^\dagger \hat{M} P|\psi\rangle = \frac{1}{2}[\langle\psi|\,(P+Q)^\dagger \hat{M}(P+Q)|\psi\rangle - \langle\psi|\,(P-Q)^\dagger \hat{M}(P-Q)|\psi\rangle]$$

(try proving it yourself!).

We have the expression

$$\frac{\partial f(\theta)}{\partial \theta_i} = \langle\psi'|\left(i\frac{G}{2}\right)\ \hat{B}'|\psi'\rangle + \langle\psi'|\,\hat{B}'\ \left(-i\frac{G}{2}\right)|\psi'\rangle$$

Set $P = \frac{1}{\sqrt{2}}$, $Q = -i\frac{G}{\sqrt{2}}$

# The two-term parameter-shift rule

$$\langle\psi|\, P^\dagger \hat{M} Q|\psi\rangle + \langle\psi|\, Q^\dagger \hat{M} P|\psi\rangle = \frac{1}{2}[\langle\psi|\,(P+Q)^\dagger \hat{M}(P+Q)|\psi\rangle - \langle\psi|\,(P-Q)^\dagger \hat{M}(P-Q)|\psi\rangle]$$

Setting $P = \frac{1}{\sqrt{2}}$, $Q = -i\frac{G}{\sqrt{2}}$ we get

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle\psi'|\left(\frac{I}{\sqrt{2}} - i\frac{G}{\sqrt{2}}\right)^\dagger \hat{B}' \left(\frac{I}{\sqrt{2}} - i\frac{G}{\sqrt{2}}\right)|\psi'\rangle - \langle\psi'|\left(\frac{I}{\sqrt{2}} + i\frac{G}{\sqrt{2}}\right)^\dagger \hat{B}' \left(\frac{I}{\sqrt{2}} + i\frac{G}{\sqrt{2}}\right)|\psi'\rangle]$$

# The two-term parameter-shift rule

Recall that $U(\theta) = e^{-i\theta\frac{G}{2}}$ for $G$ a Pauli. Evaluate this at $\theta = \frac{\pi}{2}$:

$$
\begin{aligned}
U\left(\frac{\pi}{2}\right) &= e^{-i\frac{\pi}{2}\frac{G}{2}} \\
&= e^{-i\frac{\pi}{4}G} \\
&= I + \left(-i\frac{\pi}{4}G\right) + \frac{1}{2!}\left(-i\frac{\pi}{4}G\right)^2 + \frac{1}{3!}\left(-i\frac{\pi}{4}G\right)^3 + \cdots \\
&= \left(1 - \frac{1}{2!}\left(\frac{\pi}{4}\right)^2 + \frac{1}{4!}\left(\frac{\pi}{4}\right)^4 + \cdots\right) I \\
&\quad - i\left(\frac{\pi}{4} - \frac{1}{3!}\left(\frac{\pi}{4}\right)^3 + \cdots\right) G \\
&= \cos(\pi/4) I - i\sin(\pi/4) G \\
&= \frac{1}{\sqrt{2}}(I - iG)
\end{aligned}
$$

Can similarly show that $U(-\pi/2) = (I + iG)/\sqrt{2}$.

# The two-term parameter-shift rule

So from

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle \psi' | \left(\frac{I}{\sqrt{2}} - i\frac{G}{\sqrt{2}}\right)^\dagger \hat{B}' \left(\frac{I}{\sqrt{2}} - i\frac{G}{\sqrt{2}}\right) |\psi'\rangle - \langle \psi' | \left(\frac{I}{\sqrt{2}} + i\frac{G}{\sqrt{2}}\right)^\dagger \hat{B}' \left(\frac{I}{\sqrt{2}} + i\frac{G}{\sqrt{2}}\right) |\psi'\rangle],$$

we obtain

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle \psi' | U \left(\frac{\pi}{2}\right)^\dagger \hat{B}' U \left(\frac{\pi}{2}\right) |\psi'\rangle - \langle \psi' | U \left(-\frac{\pi}{2}\right)^\dagger \hat{B}' U \left(-\frac{\pi}{2}\right) |\psi'\rangle],$$

# The two-term parameter-shift rule

Earlier, we defined $|\psi'\rangle = U(\theta_i)|\psi\rangle$. So, we can rewrite

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle\psi'| U\left(\frac{\pi}{2}\right)^\dagger \hat{B}' U\left(\frac{\pi}{2}\right)|\psi'\rangle - \langle\psi'| U\left(-\frac{\pi}{2}\right)^\dagger \hat{B}' U\left(-\frac{\pi}{2}\right)|\psi'\rangle],$$

as

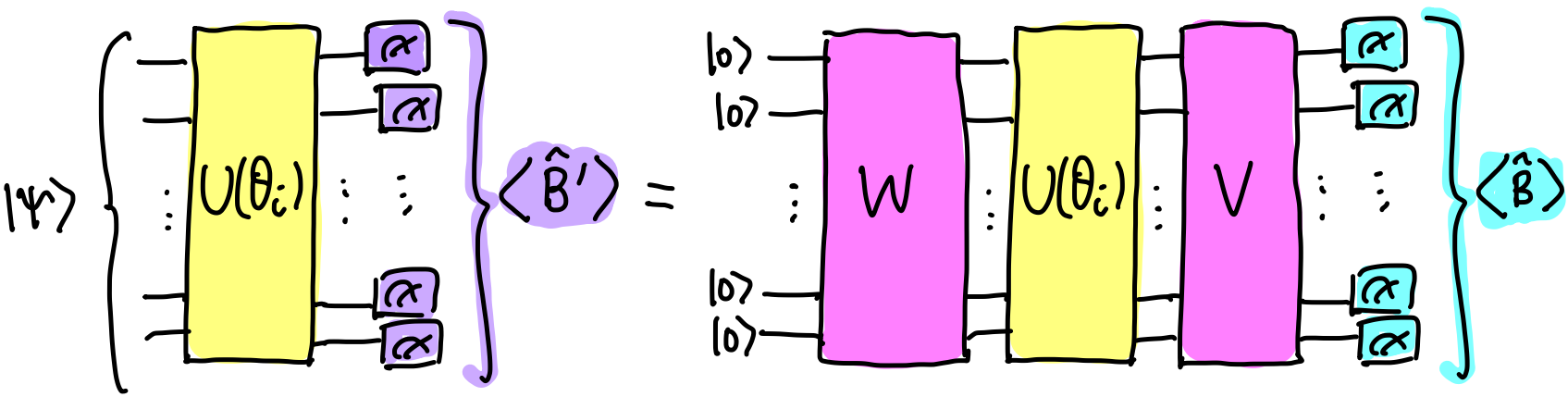
$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle\psi| U(\theta_i)^\dagger U\left(\frac{\pi}{2}\right)^\dagger \hat{B}' U\left(\frac{\pi}{2}\right) U(\theta_i)|\psi\rangle - \langle\psi| U(\theta_i)^\dagger U\left(-\frac{\pi}{2}\right)^\dagger \hat{B}' U\left(-\frac{\pi}{2}\right) U(\theta_i)|\psi\rangle]$$

Now we can merge these as they are Pauli rotations...

# The two-term parameter-shift rule

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle\psi| U\left(\theta_i + \frac{\pi}{2}\right)^\dagger \hat{B}' U\left(\theta_i + \frac{\pi}{2}\right)|\psi\rangle$$
$$- \langle\psi| U\left(\theta_i - \frac{\pi}{2}\right)^\dagger \hat{B}' U\left(\theta_i - \frac{\pi}{2}\right)|\psi\rangle]$$

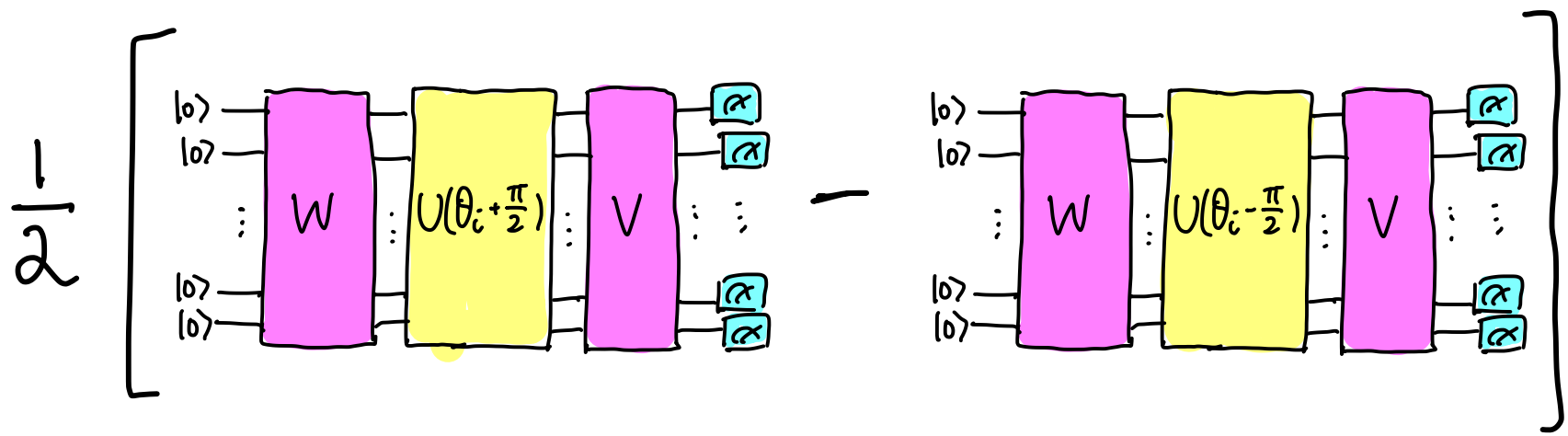
But recall we had made some other substitutions...



# The two-term parameter-shift rule

$$\frac{\partial f(\theta)}{\partial \theta_i} = \frac{1}{2}[\langle 0| W^\dagger U\left(\theta_i + \frac{\pi}{2}\right)^\dagger V^\dagger \hat{B} V U\left(\theta_i + \frac{\pi}{2}\right) W|0\rangle$$
$$- \langle 0| W^\dagger U\left(\theta_i - \frac{\pi}{2}\right)^\dagger V^\dagger \hat{B} V U\left(\theta_i - \frac{\pi}{2}\right) W|0\rangle]$$

So we've recovered the parameter-shift rule!



# Generalized parameter-shift rules

Not all operations admit the simple two-term shift rule. For example, $CRX(\theta), CRY(\theta), CRZ(\theta)$ have a four-term rule.

If $CRX(\theta)$ is in a circuit with output function $f(\theta)$,

$$\frac{\partial f(\theta)}{\partial \theta} = c_+[f(\theta+\pi/2)-f(\theta-\pi/2)]-c_-[f(\theta+3\pi/2)-f(\theta-3\pi/2)]$$

where

$$c_\pm = \frac{\sqrt{2}\pm 1}{4\sqrt{2}}$$

For more info: `https://arxiv.org/abs/2104.05695`, `https://arxiv.org/abs/2107.12390`