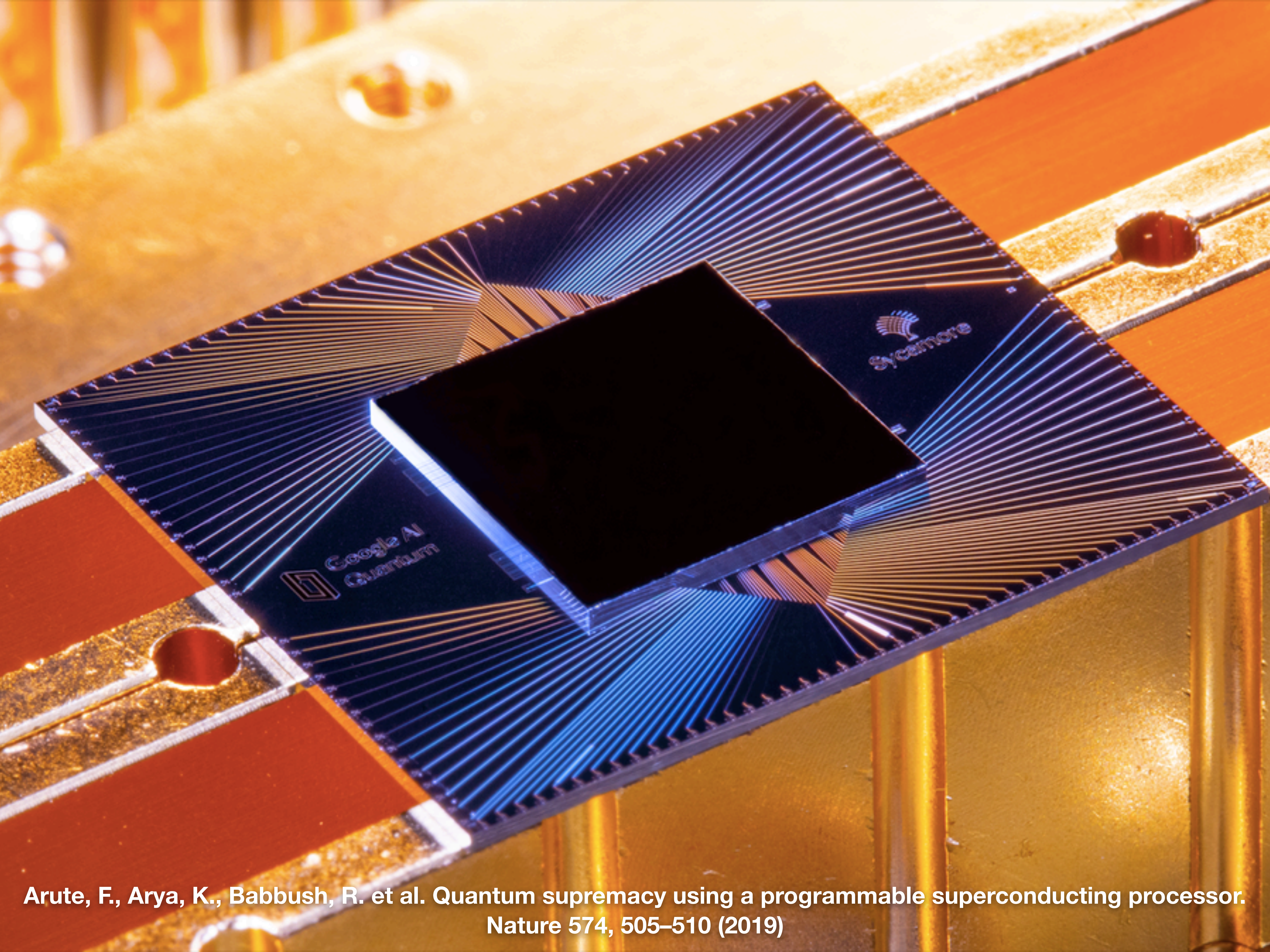
# Practical Quantum Numerical Integration (a step toward quantum rendering)

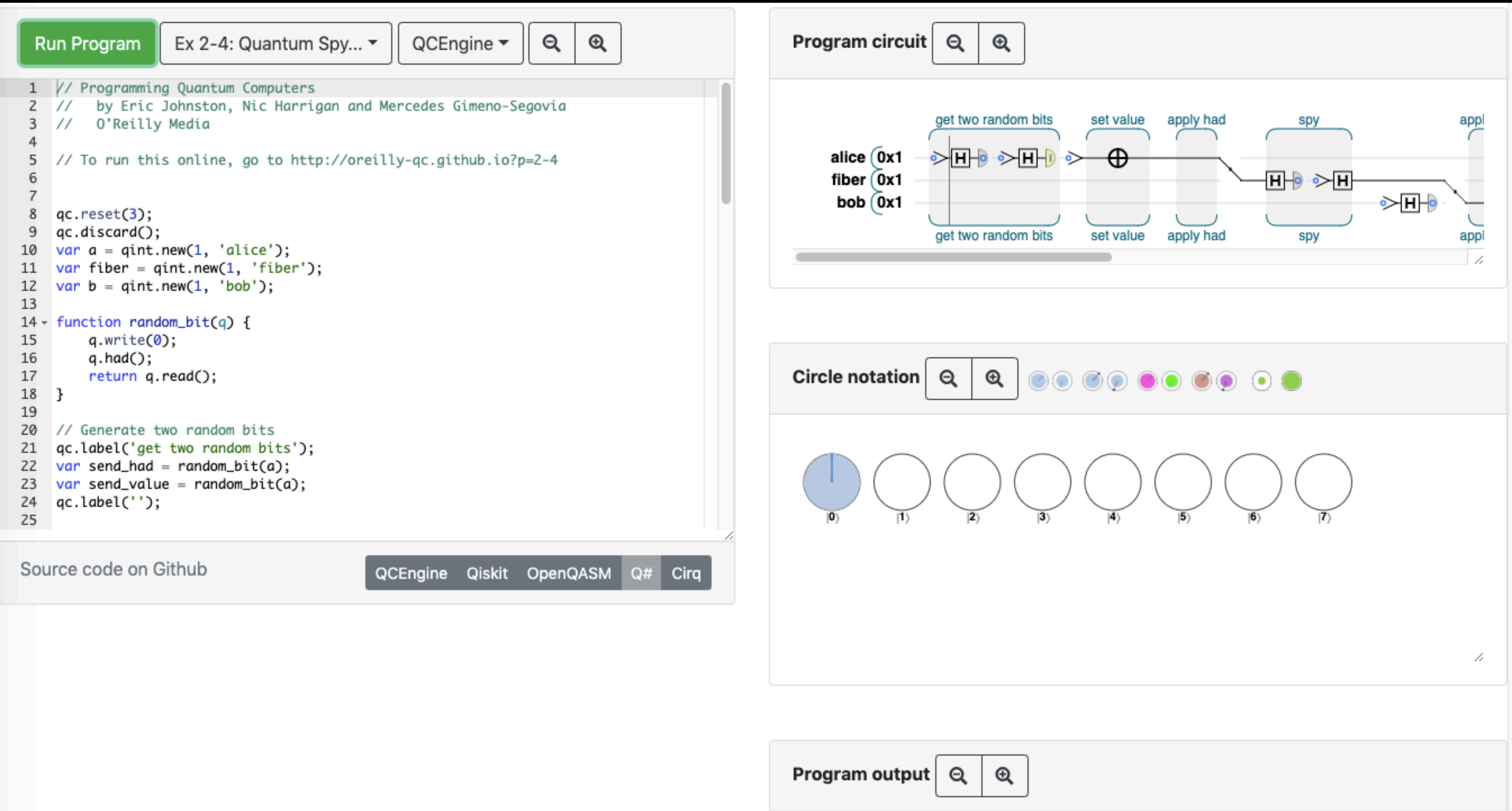
Naoharu Shimada    Toshiya Hachisuka

The University of Tokyo

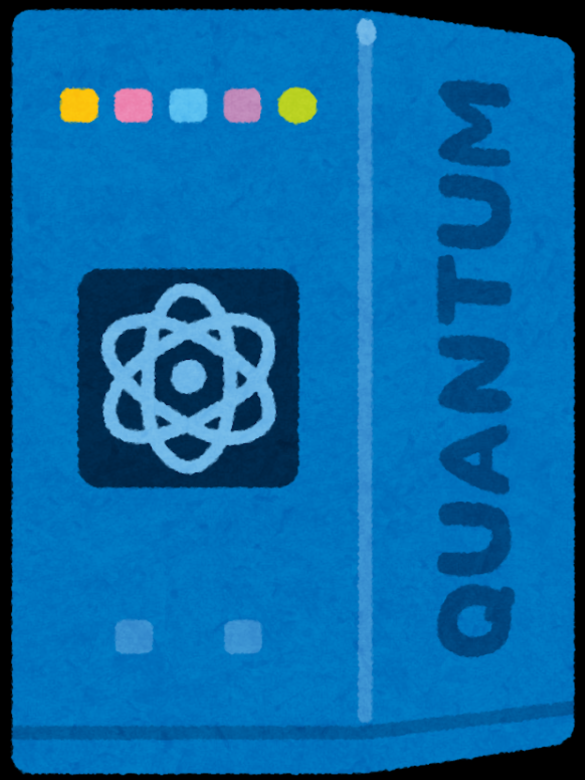
IBM “53-qubit quantum computer” (2019)




Arute, F., Arya, K., Babbush, R. et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019)



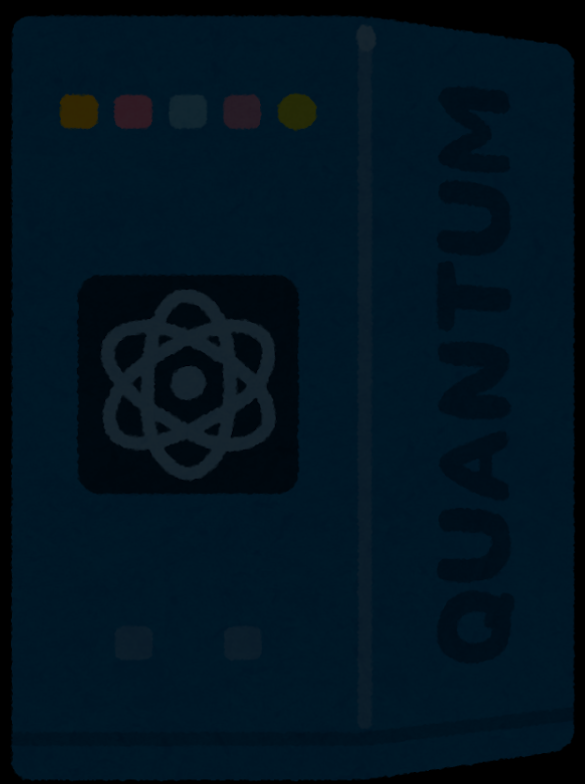

https://oreilly-qc.github.io/




**Classical computer** **Quantum computer**

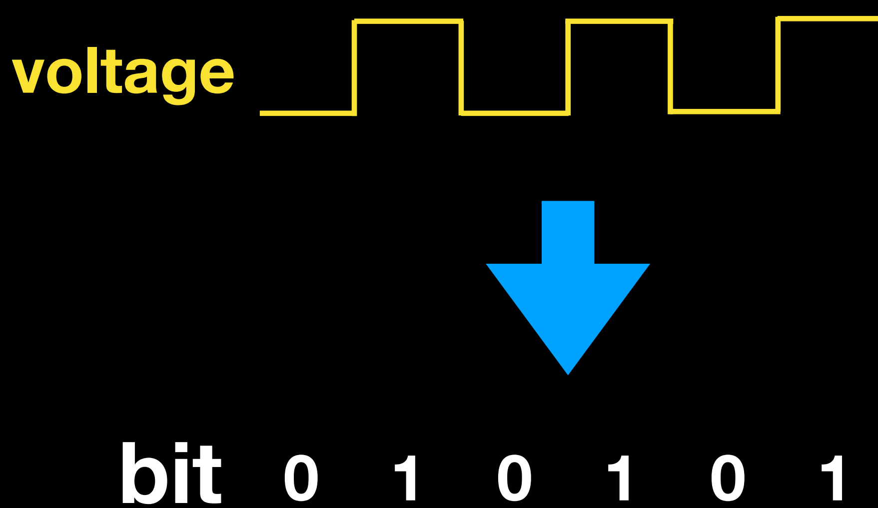

**Classical computer**

**Quantum computer**

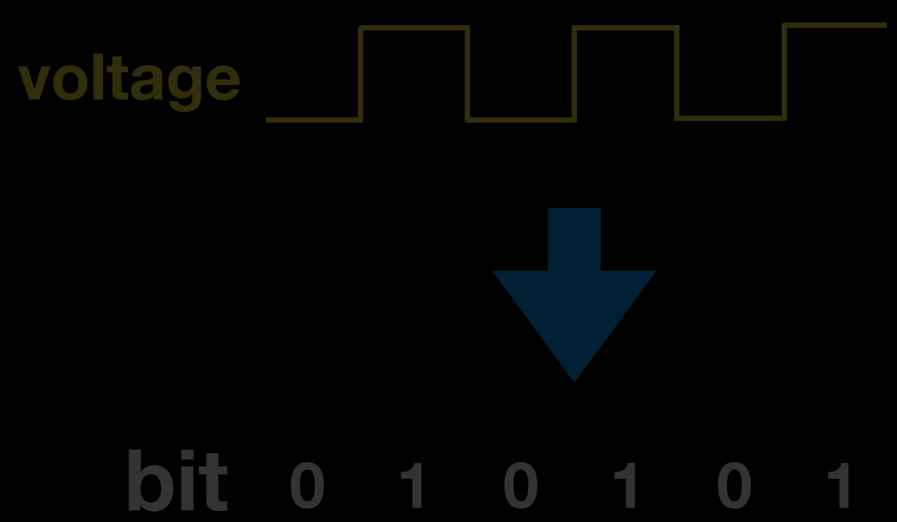

Classical computer

Quantum computer

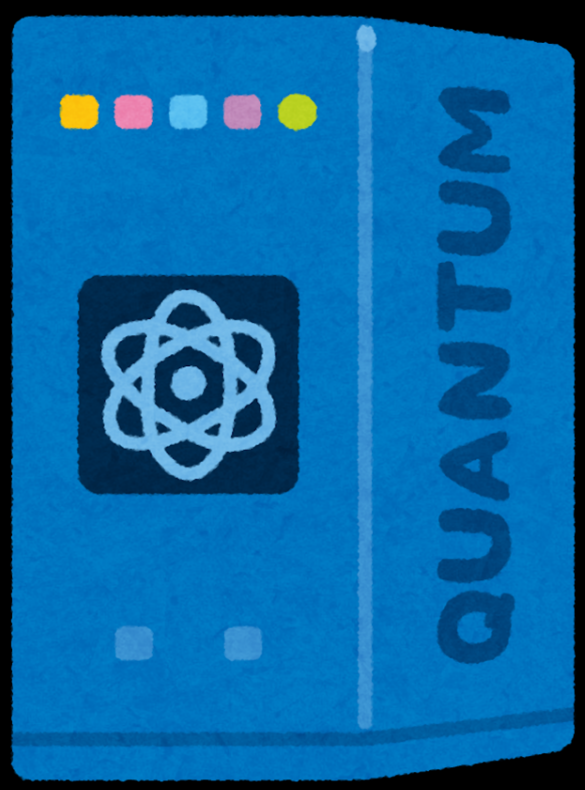



Classical computer

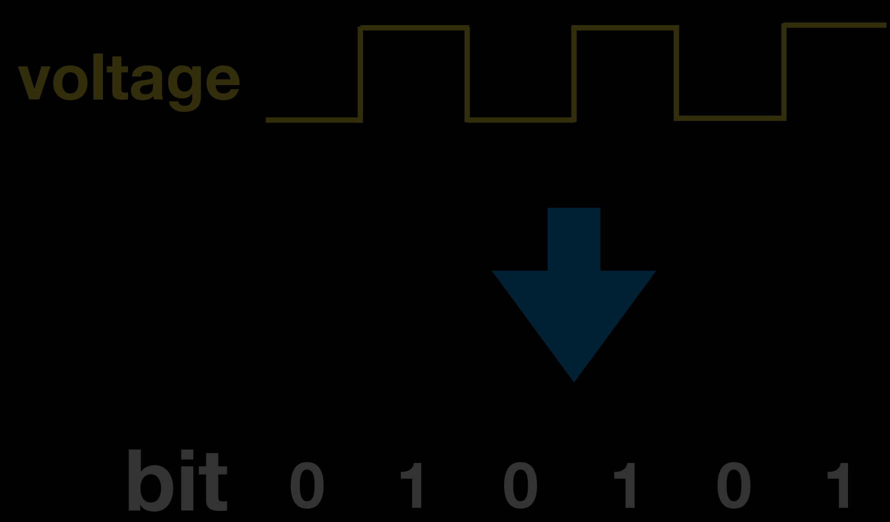

Quantum computer




Classical computer

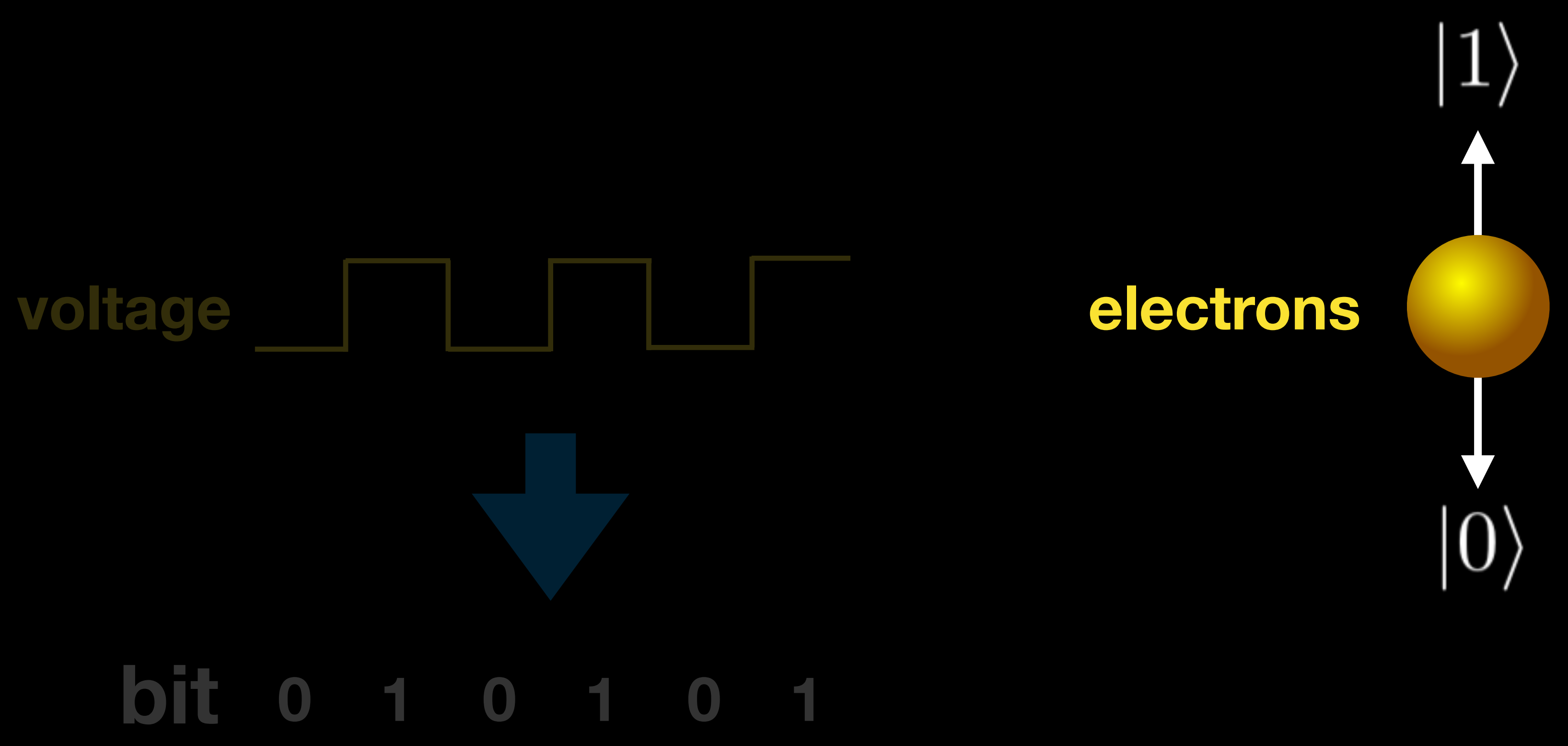



Quantum computer

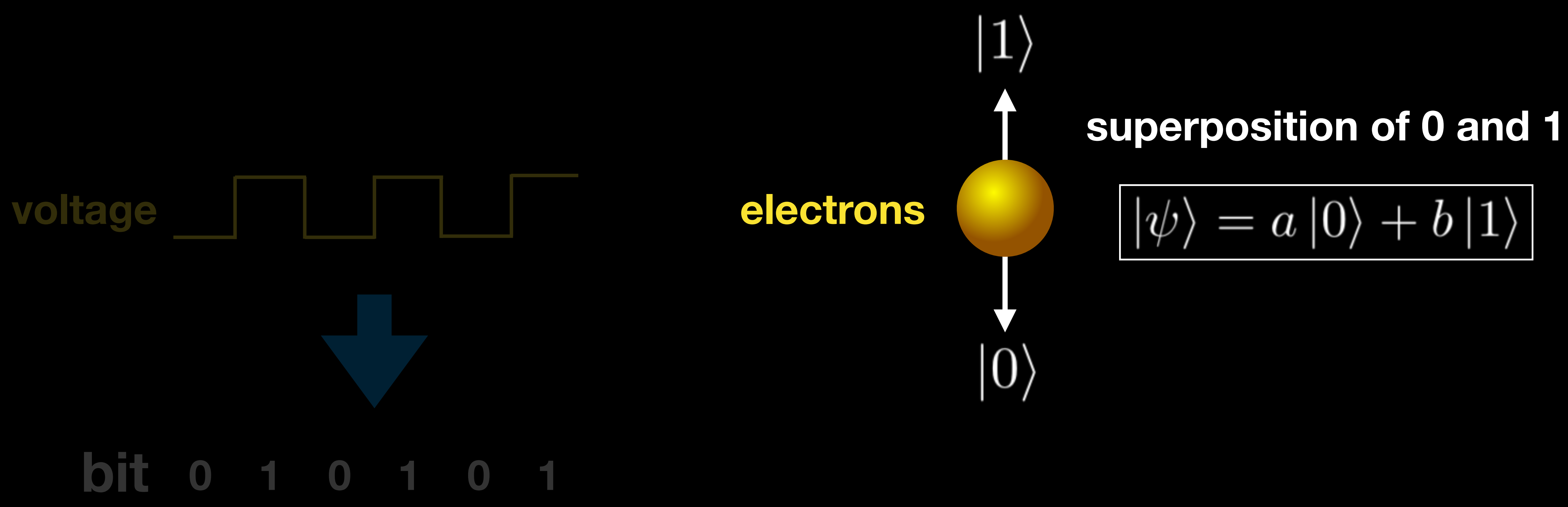



Classical computer


Quantum computer

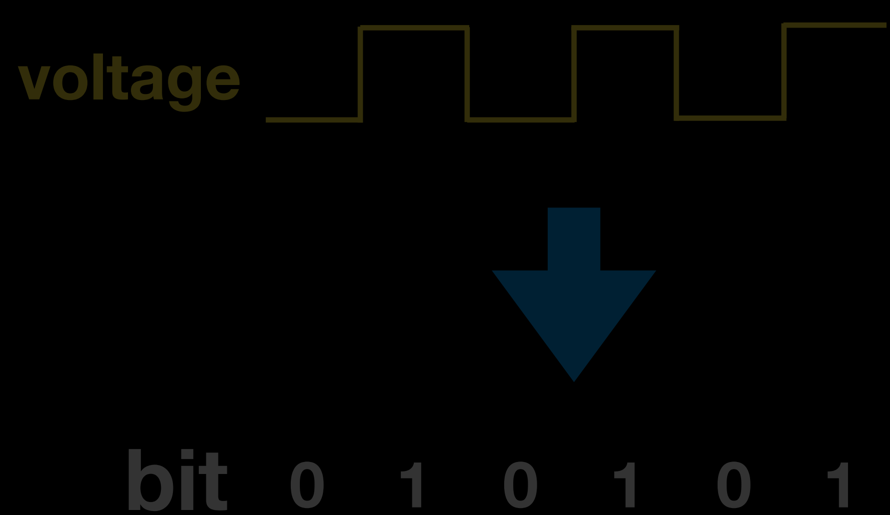



**Classical computer** **Quantum computer**

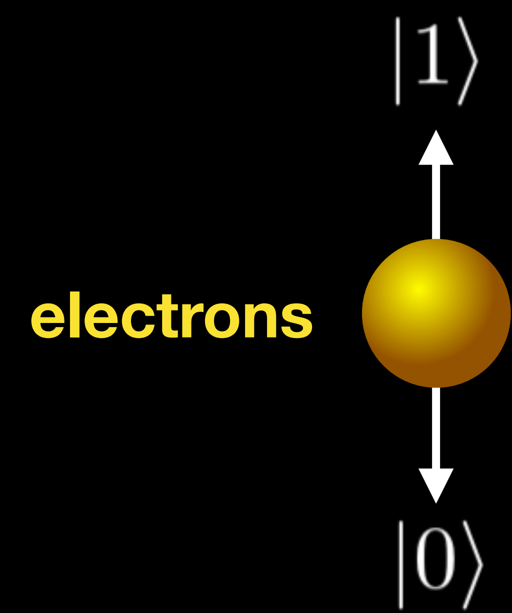



Classical computer

Quantum computer




**Classical computer**

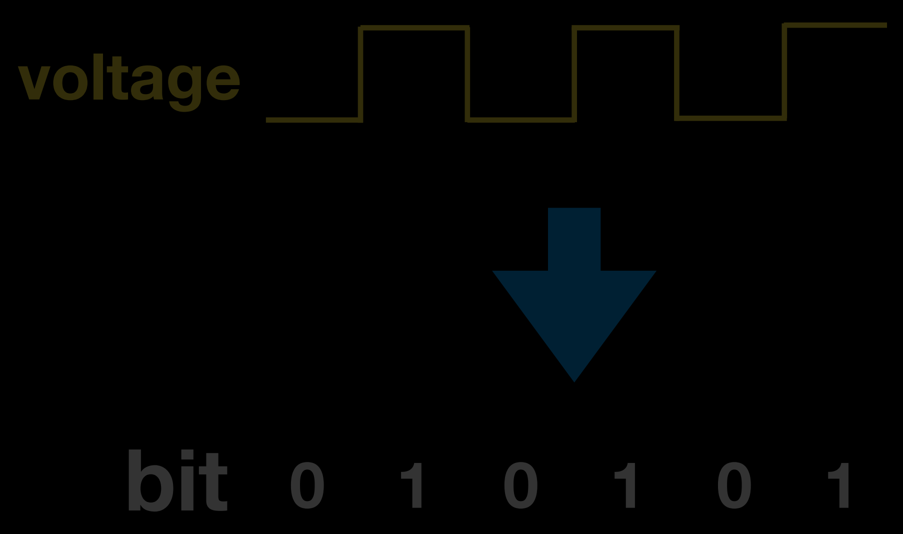


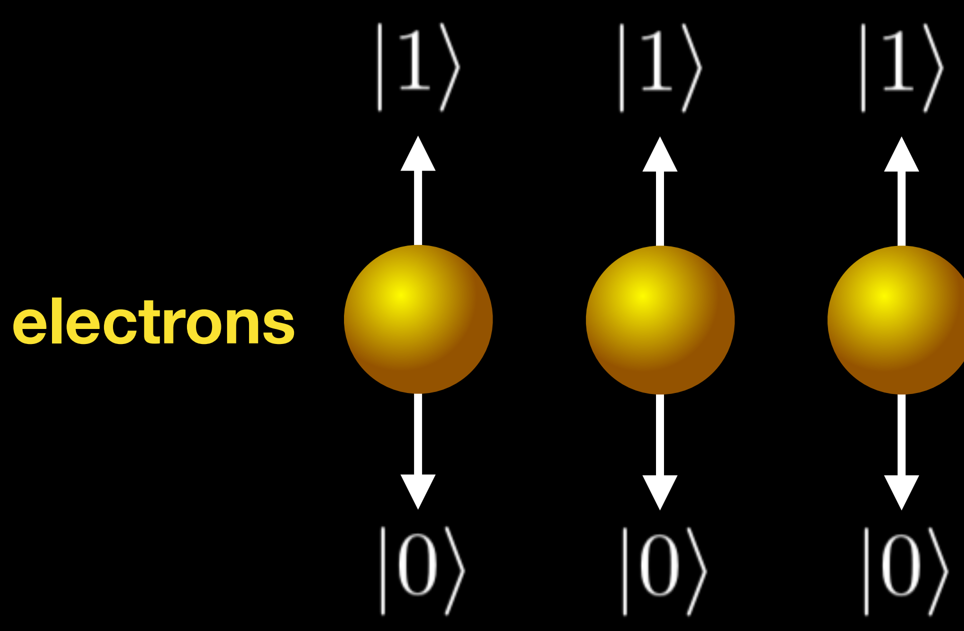

**superposition of 0 and 1**

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

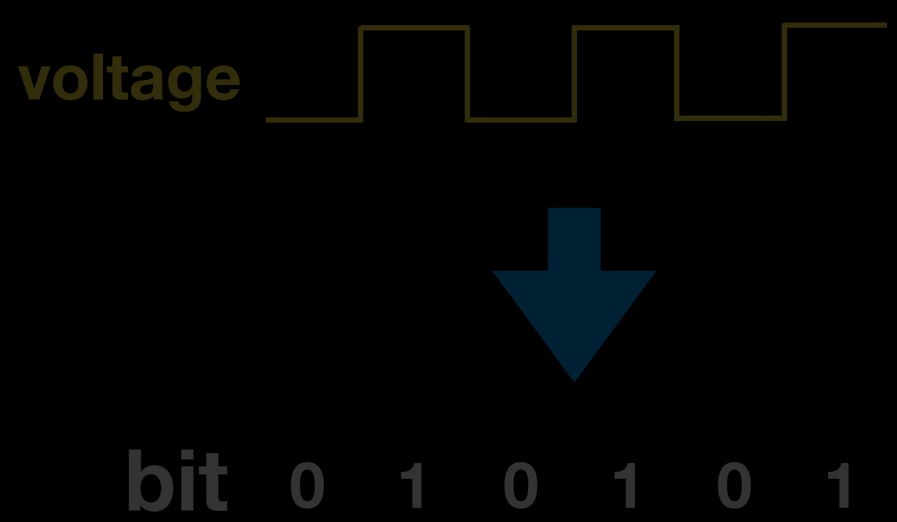
$$\begin{cases} 0 \text{ with prob. } (a^2) \\ 1 \text{ with prob. } (b^2) \end{cases}$$

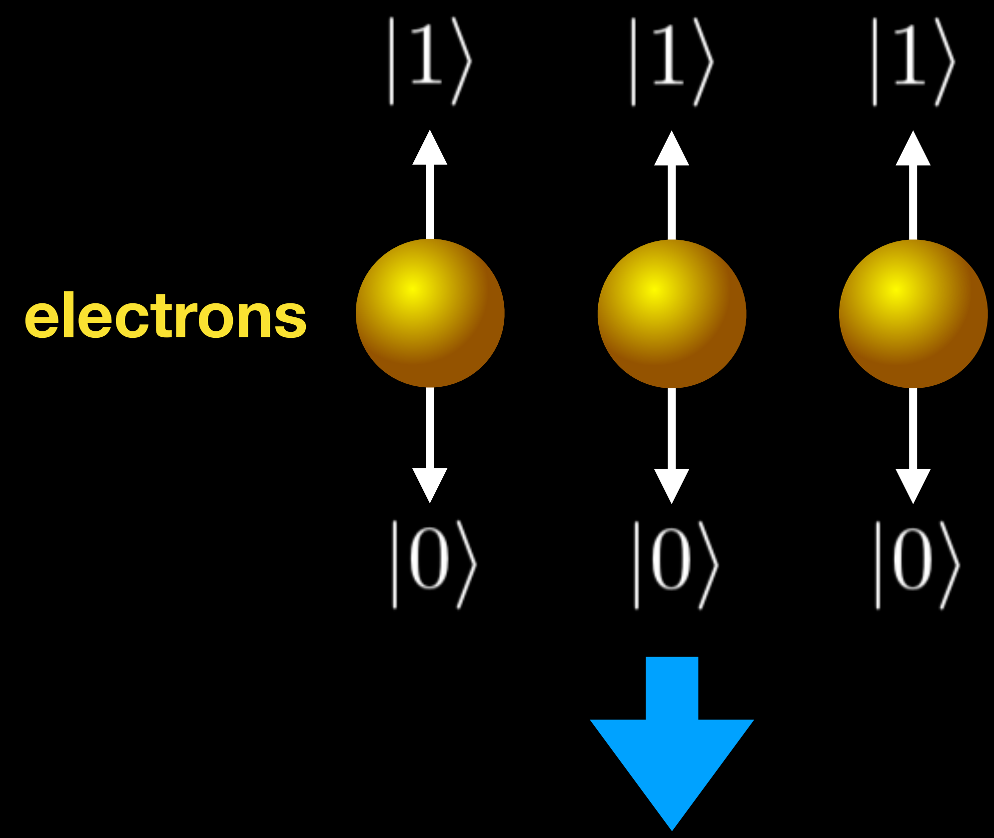
**Quantum computer**




**Classical computer** **Quantum computer**

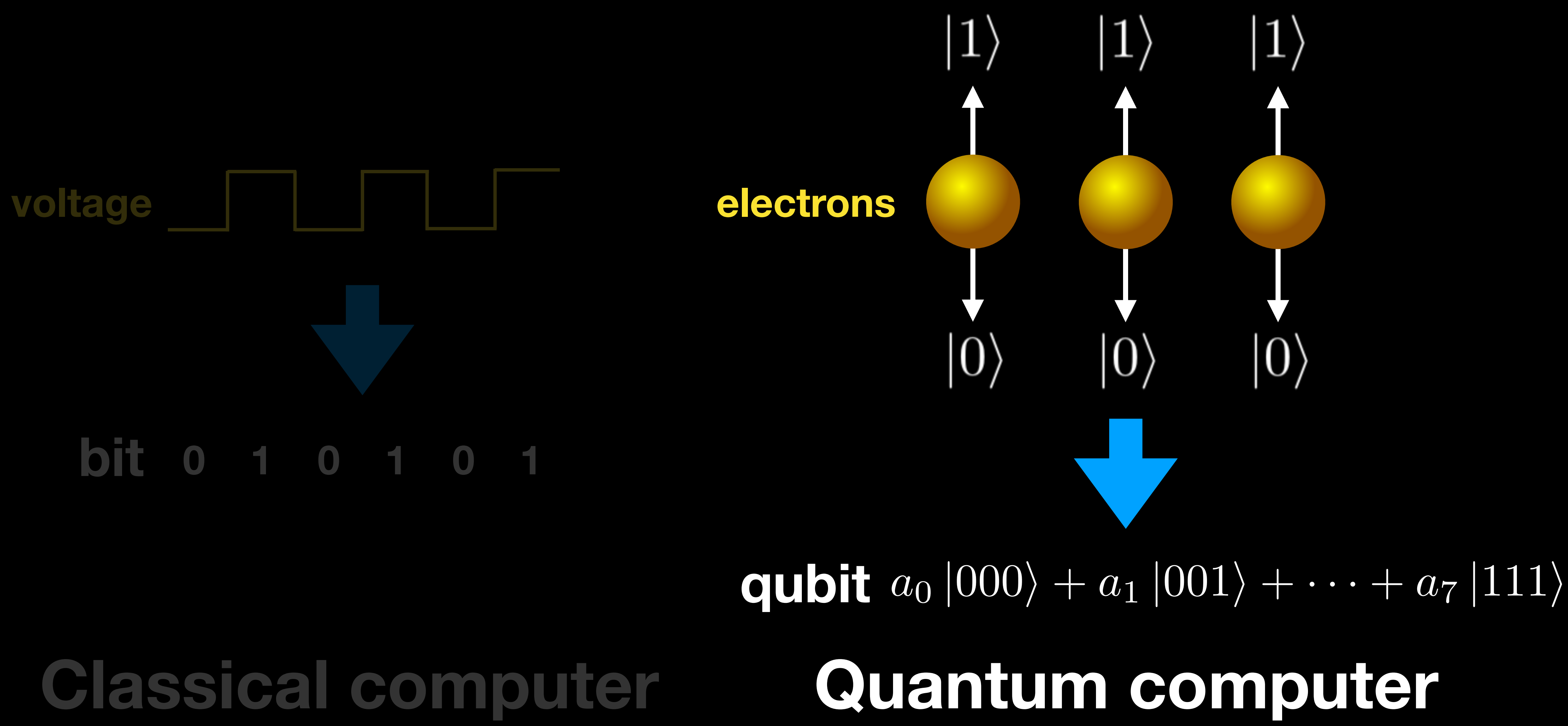



$$a_0\,|000\rangle + a_1\,|001\rangle + \cdots + a_7\,|111\rangle$$

## Classical computer

## Quantum computer


voltage
bit 0 1 0 1 0 1
Classical computer
electrons
$|1\rangle$ $|1\rangle$ $|1\rangle$
$|0\rangle$ $|0\rangle$ $|0\rangle$
qubit $a_0 |000\rangle + a_1 |001\rangle + \cdots + a_7 |111\rangle$
Quantum computer

# Bit and Qubit

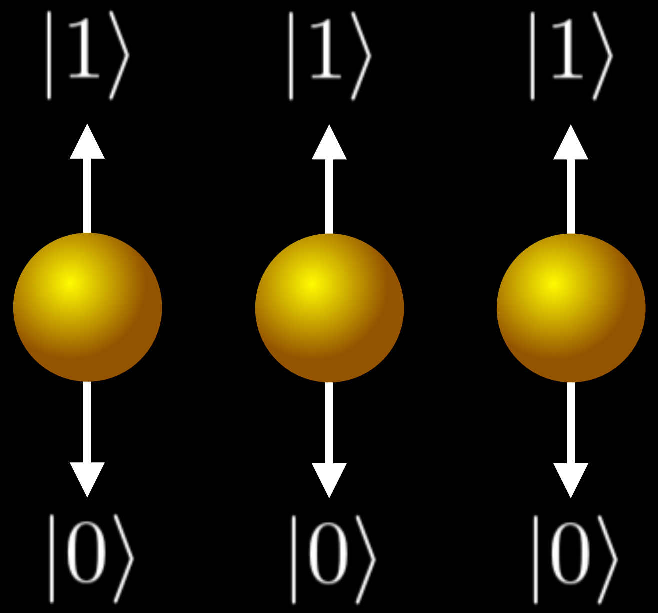
- Bit
  - Deterministically represents **either** 0 or 1
  - N bits corresponds to **1** N-bit value

# Bit and Qubit

- Bit
  - Deterministically represents **either** 0 or 1
  - N bits corresponds to **1** N-bit value
- Qubit
  - Stochastically represents **both** 0 and 1
  - N qubits correspond to $\mathbf{2^N}$ values

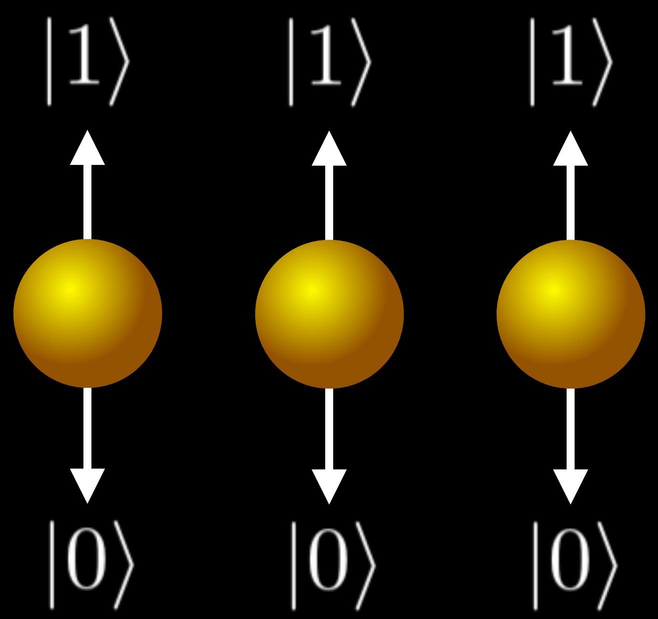
# Quantum Computing

- Utilize the **exponential** and **parallel** nature of qubits



# Quantum Computing

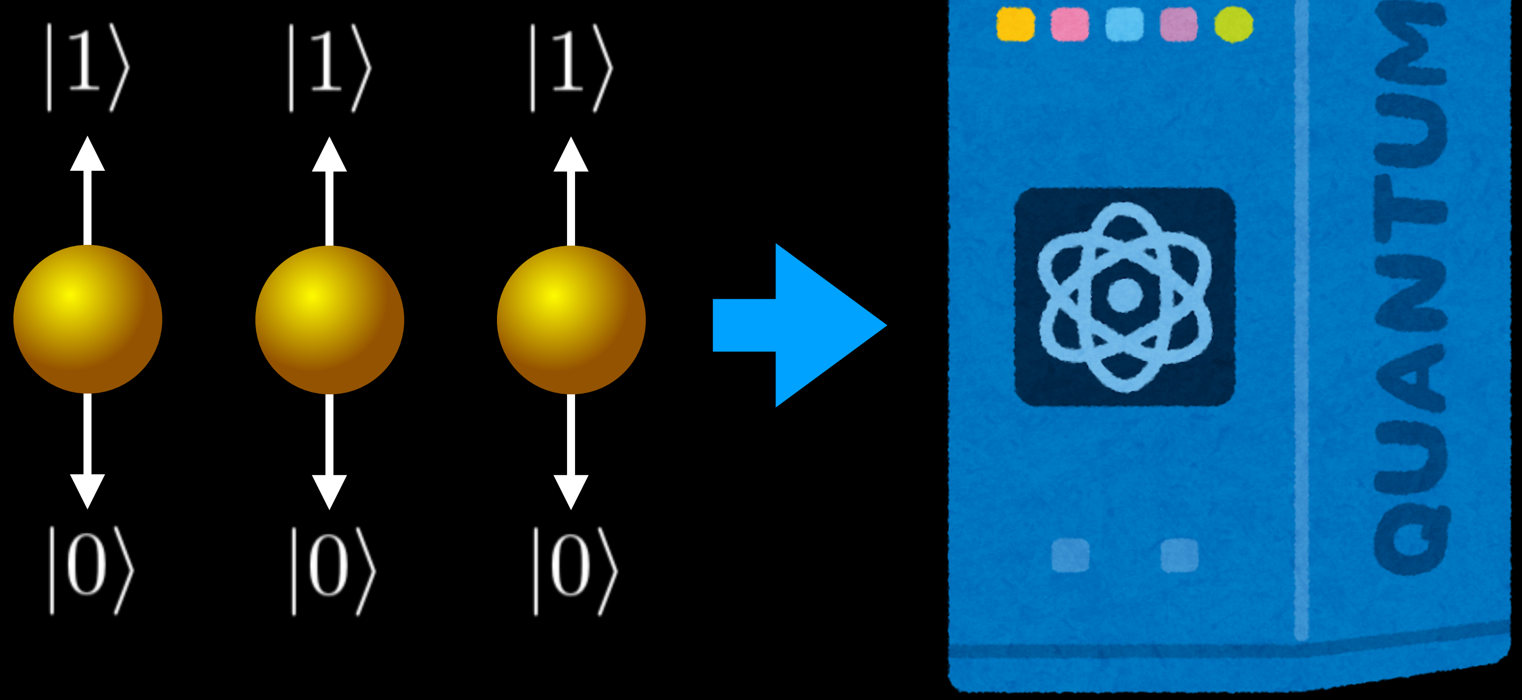
- Utilize the **exponential** and **parallel** nature of qubits




**represent**
**all the possible inputs**

# Quantum Computing

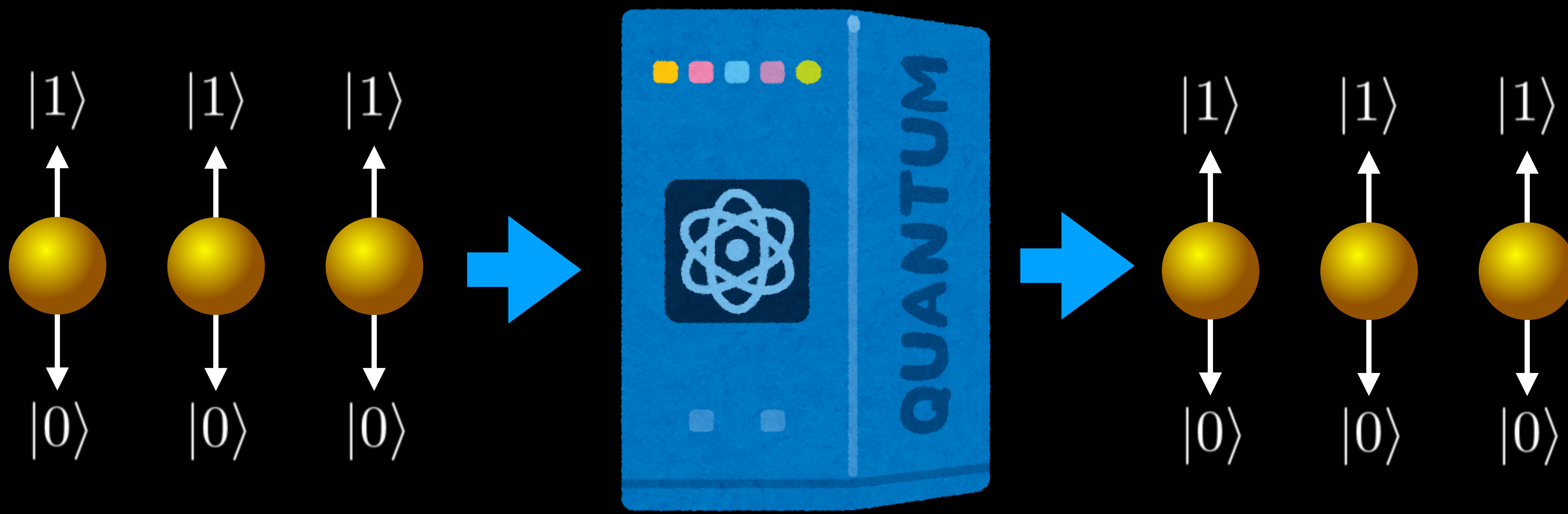
- Utilize the **exponential** and **parallel** nature of qubits




**represent**
**all the possible inputs**

# Quantum Computing

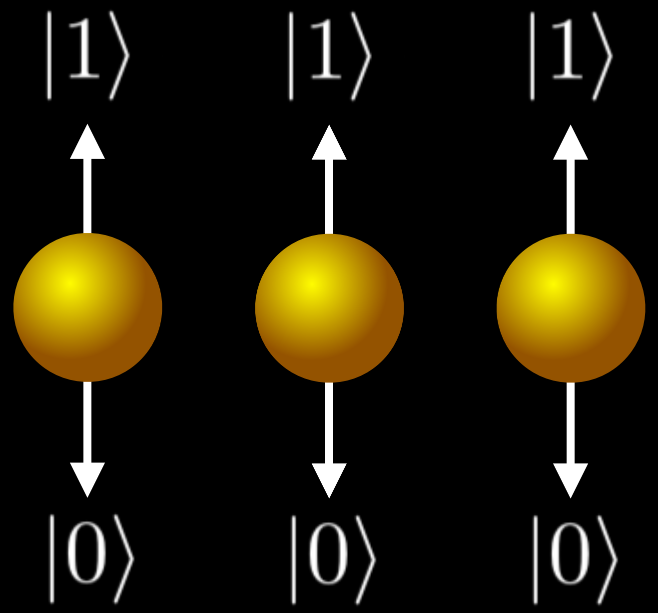
- Utilize the **exponential** and **parallel** nature of qubits



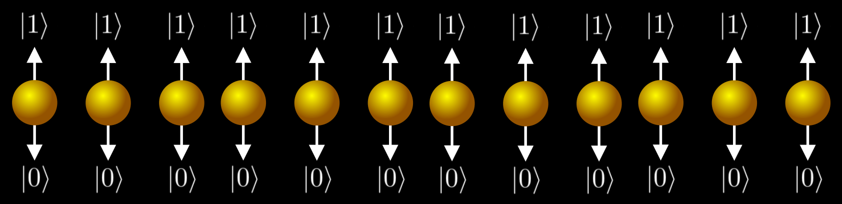

**represent**
**all the possible inputs**

**represent**
**all the possible outputs**

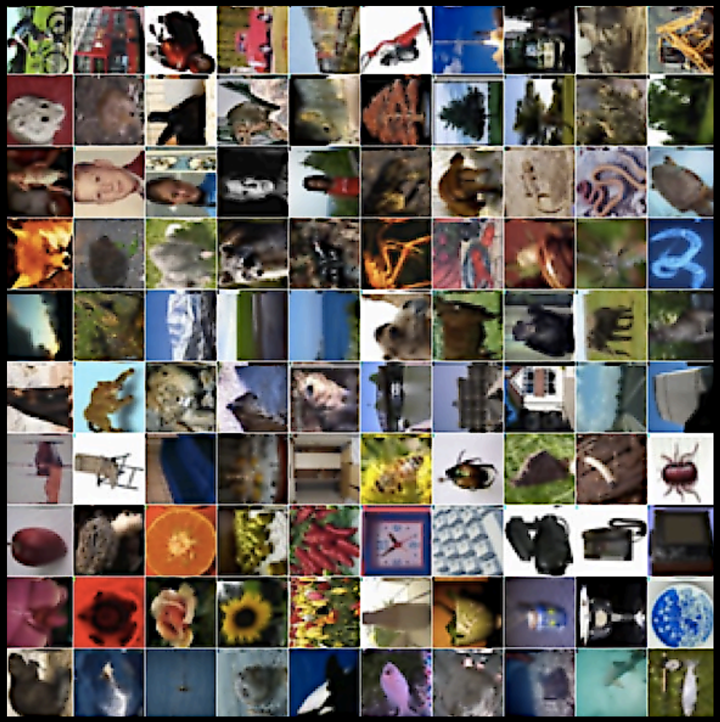
# Example: Quantum Image Blur



# Example: Quantum Image Blur

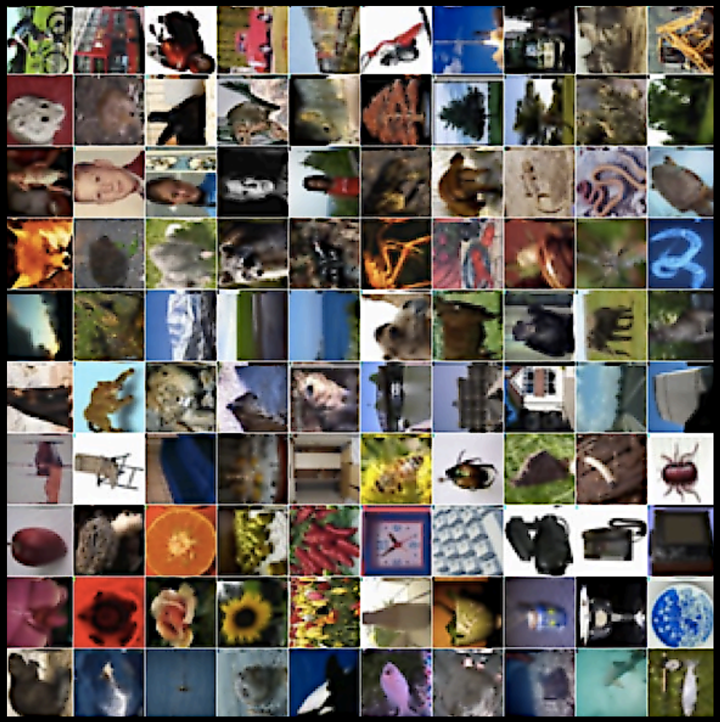


# Example: Quantum Image Blur

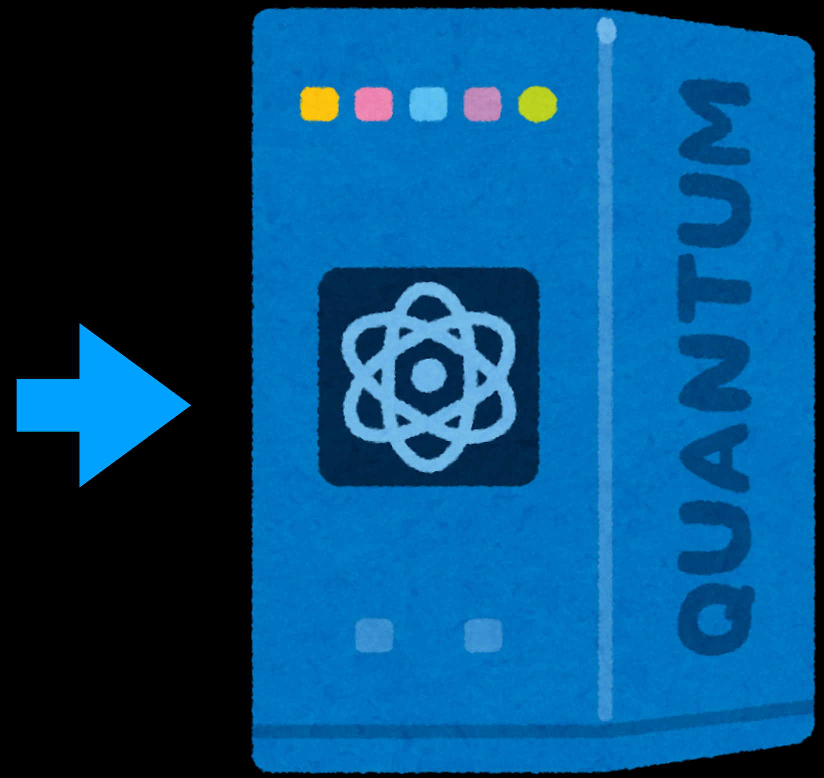


all the possible images

# Example: Quantum Image Blur

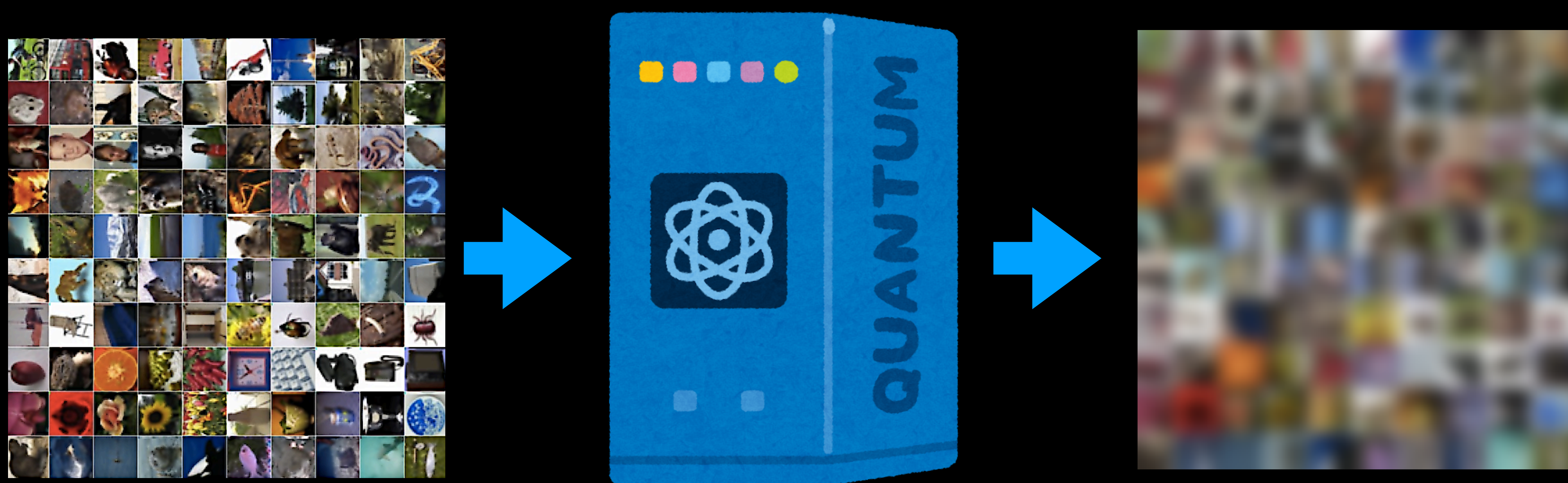


all the possible images




Blur process

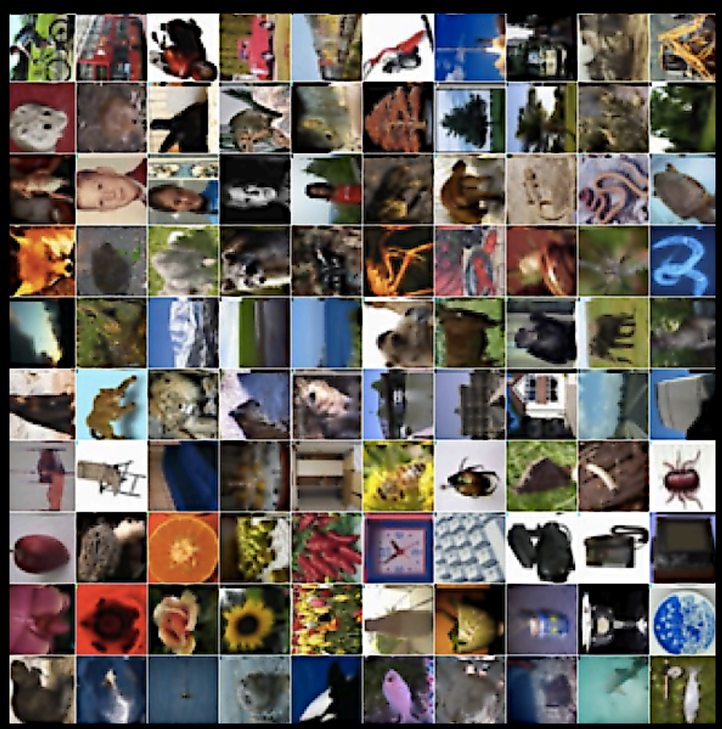
# Example: Quantum Image Blur



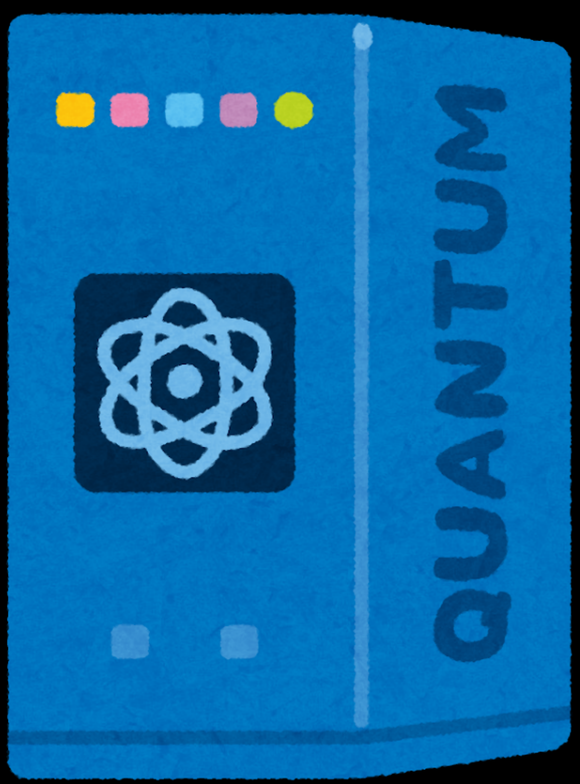

**all the possible images** | **Blur process** | **all the possible blurred images**

# Example: Quantum Image Blur

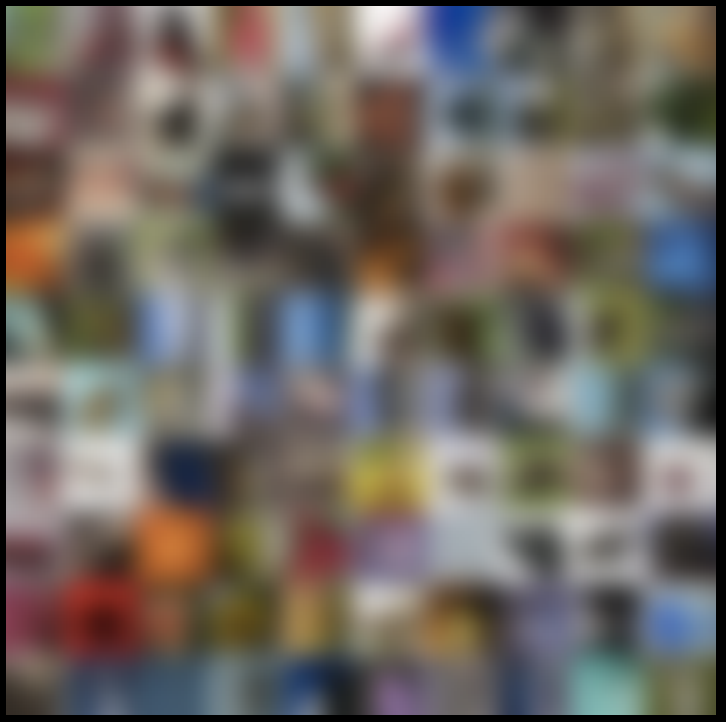
**You never need to blur an image again**



all the possible images



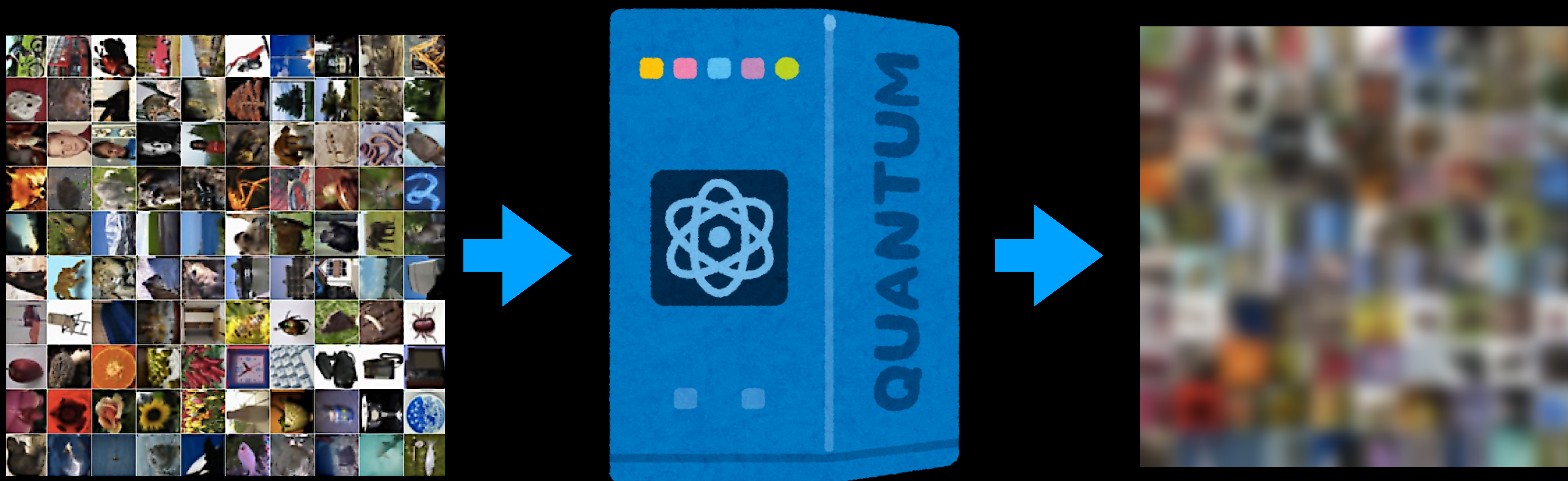

Blur process



all the possible blurred images

# Example: Quantum Image Blur

**You never need to blur an image again (or not…)**



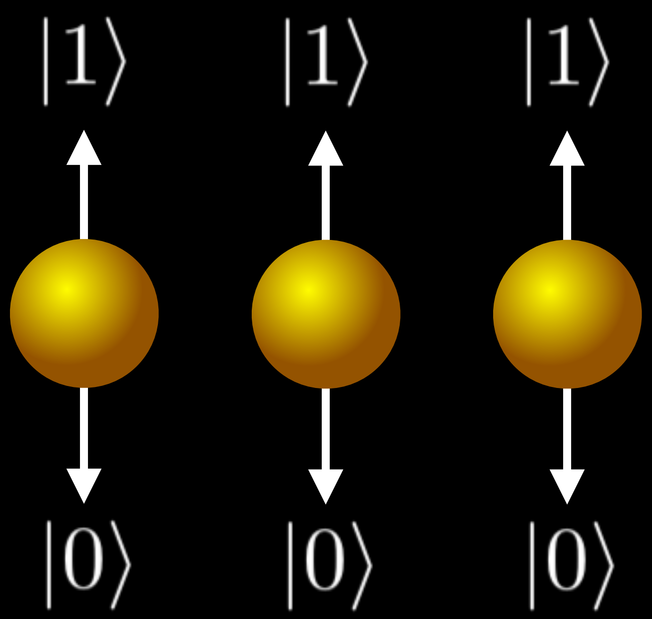

**all the possible images** | **Blur process** | **all the possible blurred images**

# Caveat #1

- Reading out qubits returns only one value *stochastically*

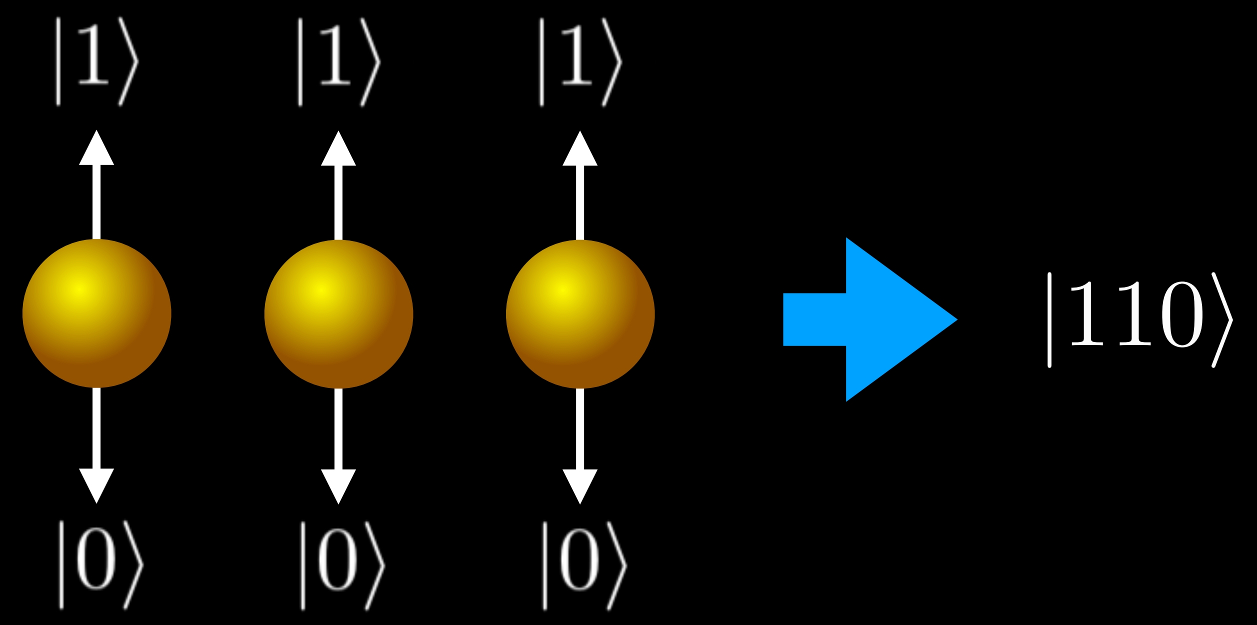
# Caveat #1

- Reading out qubits returns only one value *stochastically*



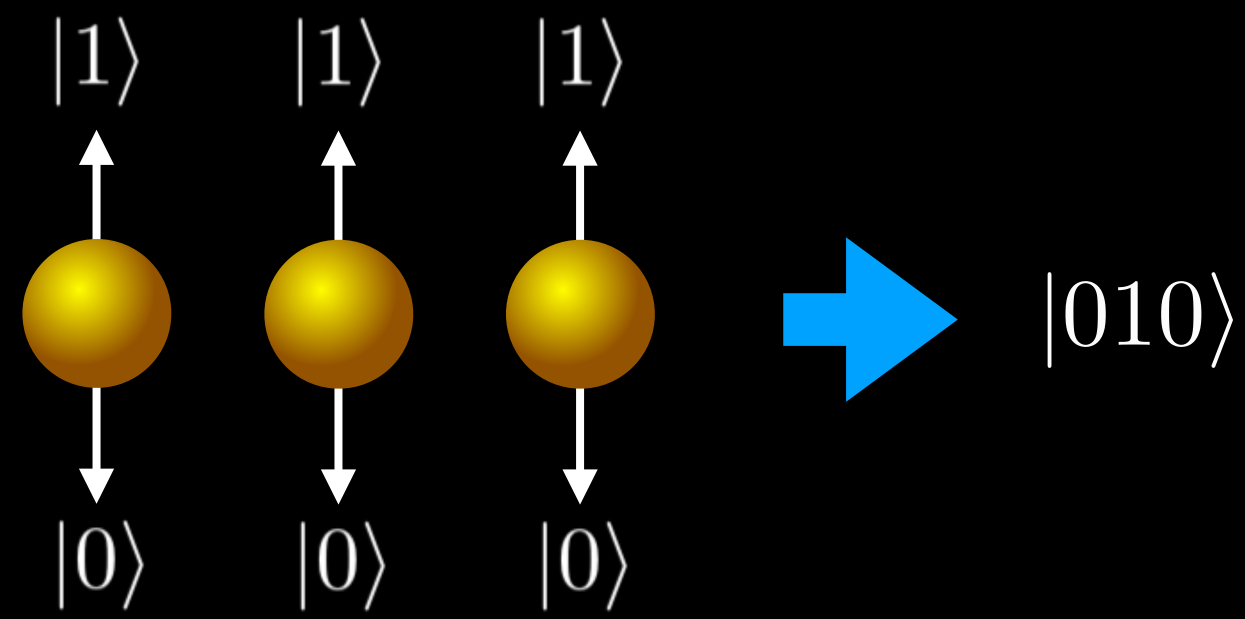
# Caveat #1

- Reading out qubits returns only one value *stochastically*



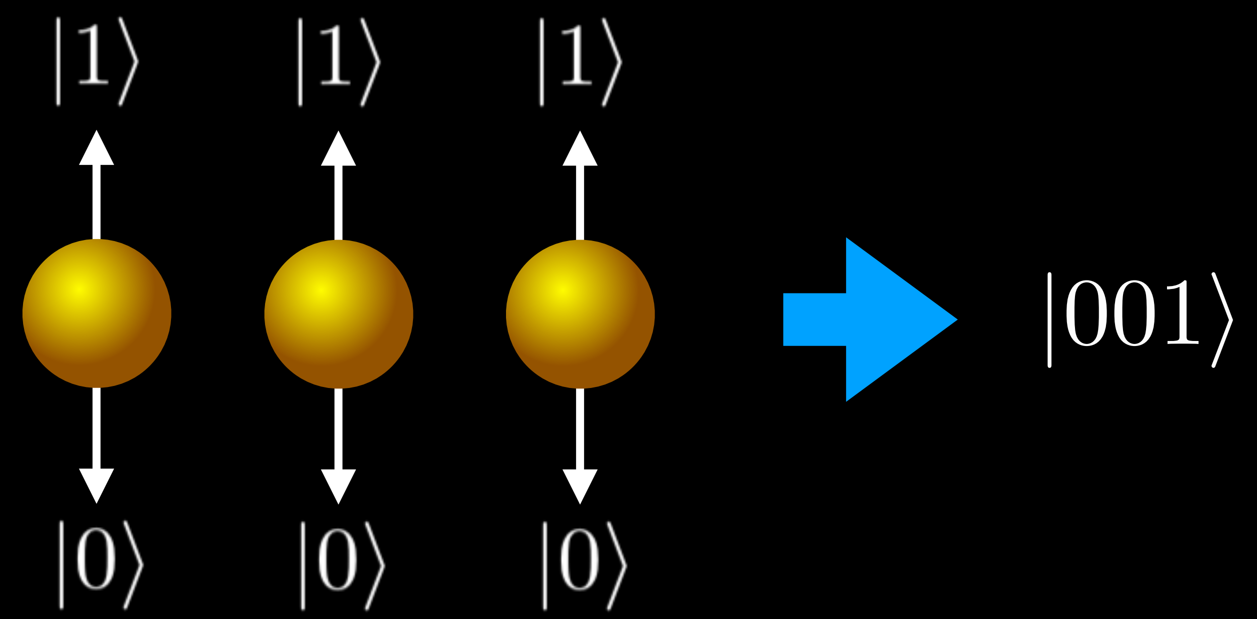
# Caveat #1

- Reading out qubits returns only one value *stochastically*



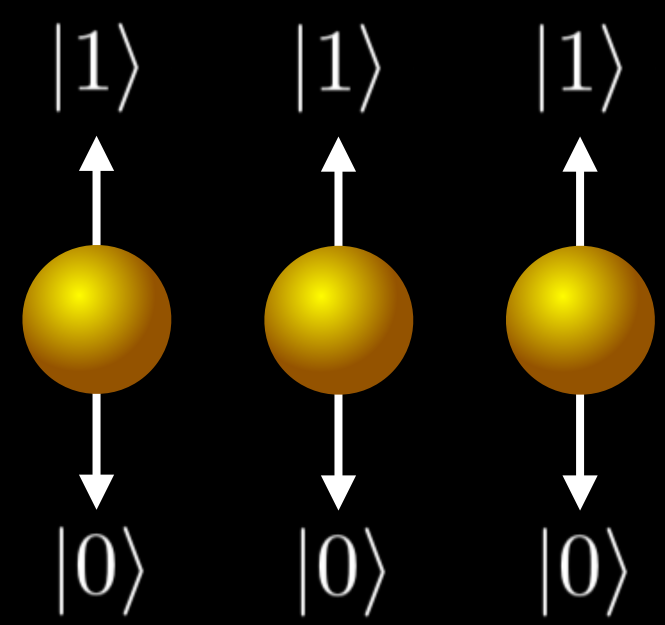
# Caveat #1

- Reading out qubits returns only one value *stochastically*



# Caveat #1

- Reading out qubits returns only one value *stochastically*




$|001\rangle$ **with the probability of** $(a_1)^2$

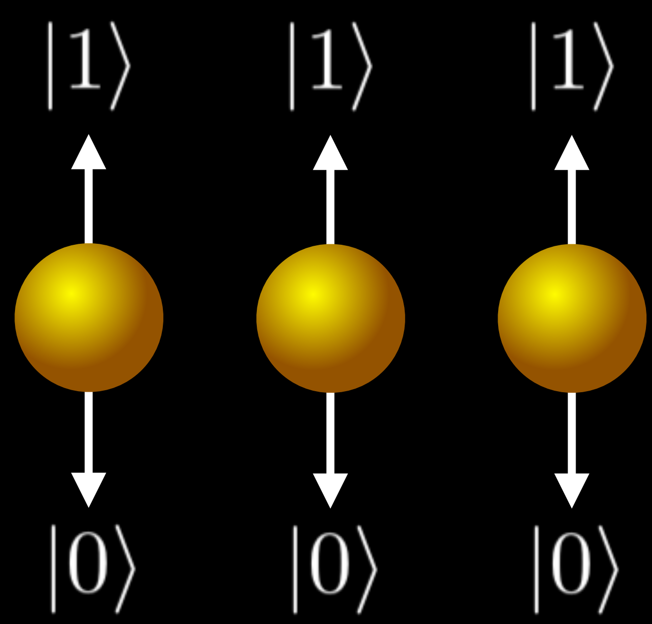
**A qubit becomes a bit after read-out**

# Caveat #2

- You cannot duplicate qubits (*no-cloning theorem*)

# Caveat #2

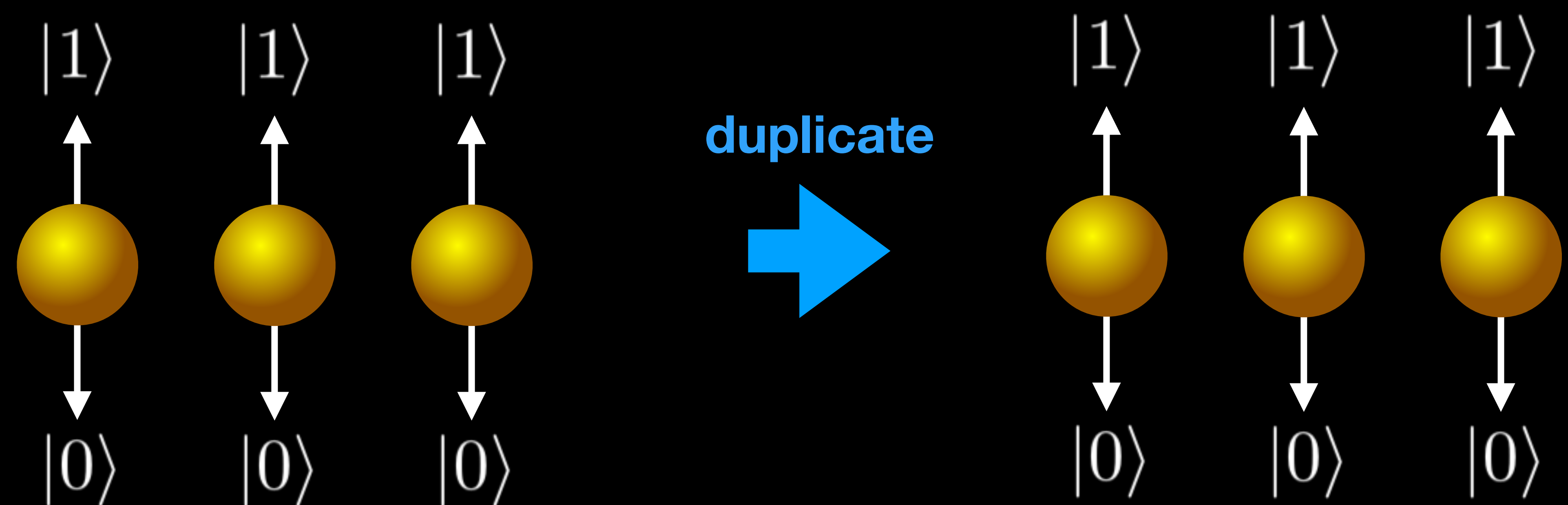
- You cannot duplicate qubits (*no-cloning theorem*)




$$a_0 |000\rangle + a_1 |001\rangle + \cdots + a_7 |111\rangle$$

# Caveat #2

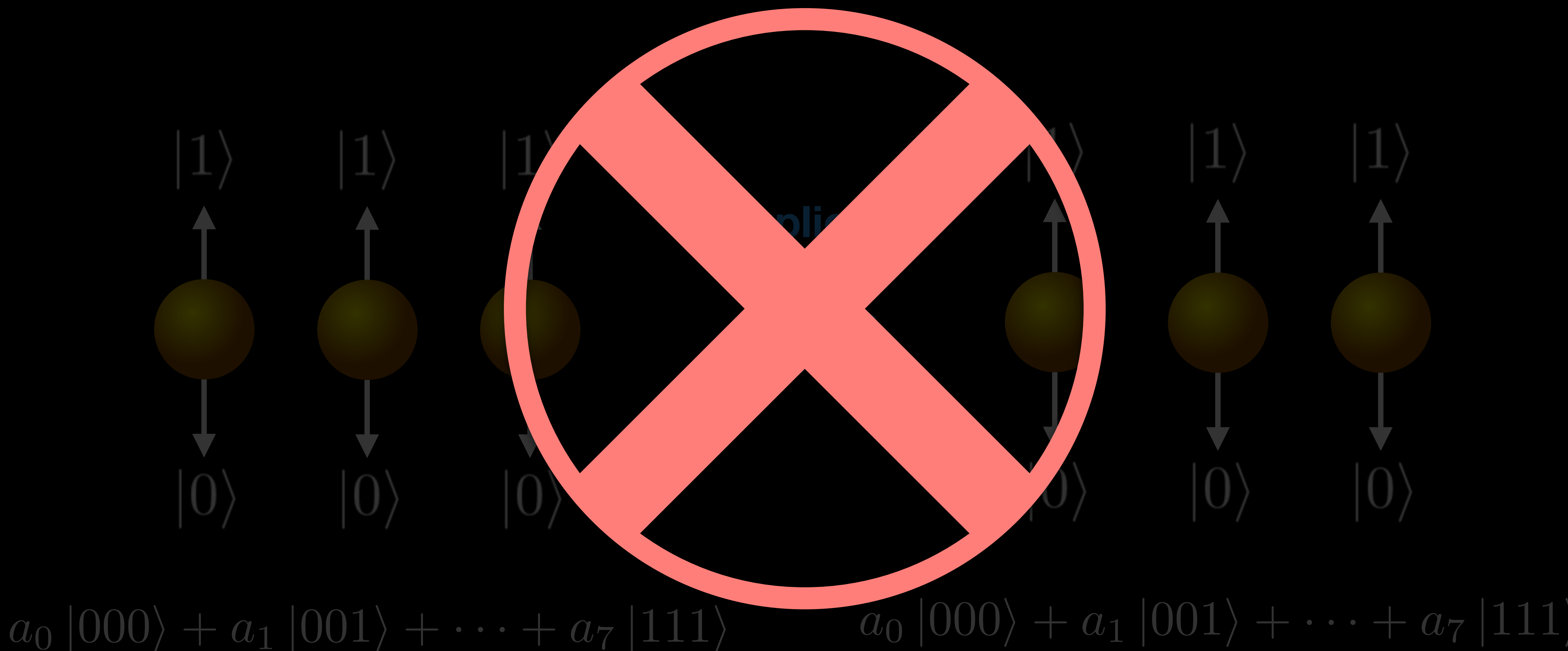
- You cannot duplicate qubits (*no-cloning theorem*)




$a_0 \left|000\right\rangle + a_1 \left|001\right\rangle + \cdots + a_7 \left|111\right\rangle$ $\quad$ $a_0 \left|000\right\rangle + a_1 \left|001\right\rangle + \cdots + a_7 \left|111\right\rangle$

# Caveat #2

- You cannot duplicate qubits (*no-cloning theorem*)




**No copying of variables**

# Caveat #3

- Current quantum computers are *noisy*
  - Results can be affected by external noise
  - Quantum operations might produce wrong results
  - "*Noisy Intermediate-Scale Quantum*" (NISQ) [Preskill18]

# Caveat #3

- Current quantum computers are *noisy*
  - Results can be affected by external noise
  - Quantum operations might produce wrong results
  - "*Noisy Intermediate-Scale Quantum*" (NISQ) [Preskill18]

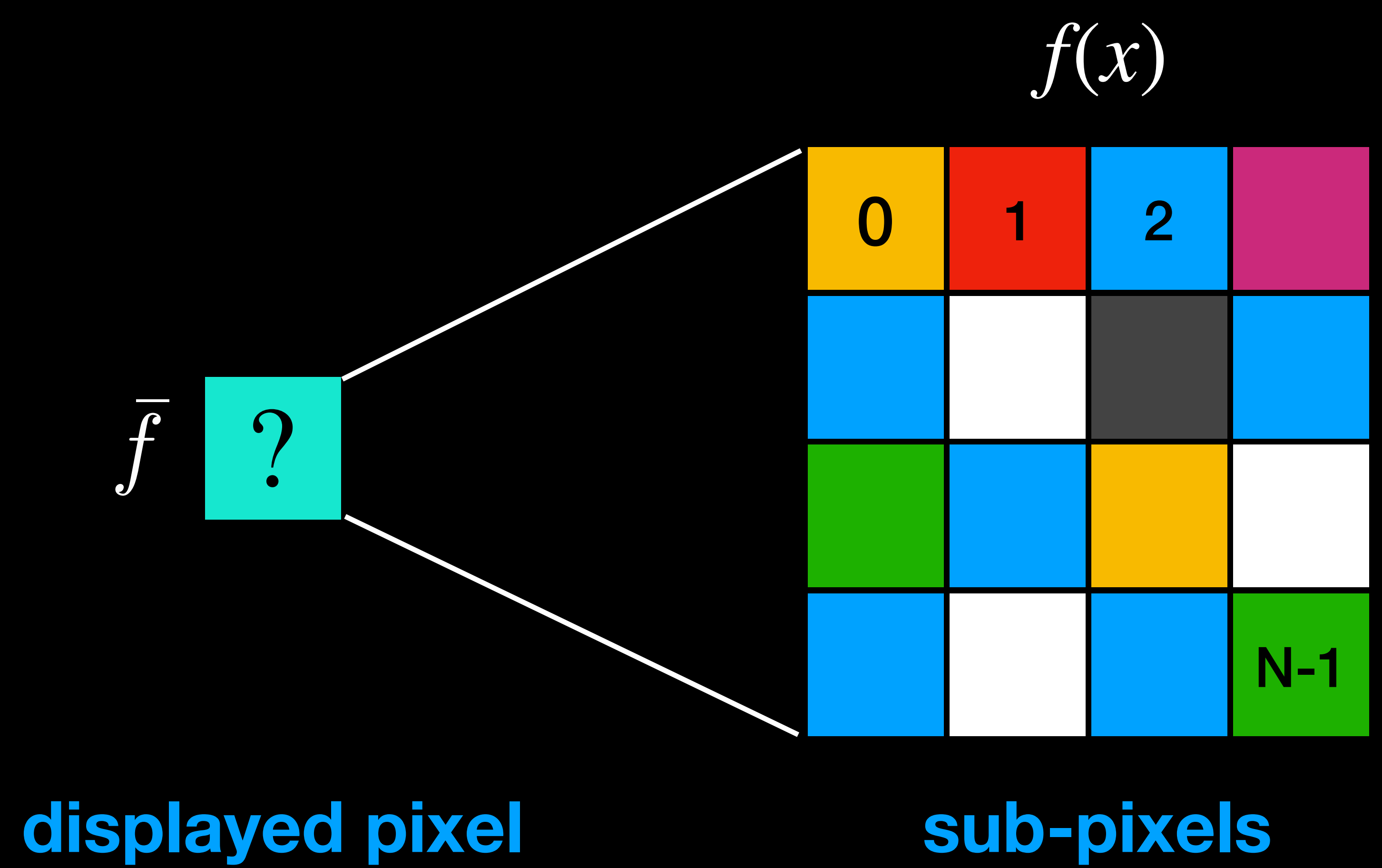
**Algorithms should be robust to noise in quantum computers**

# Quantum Numerical Integration

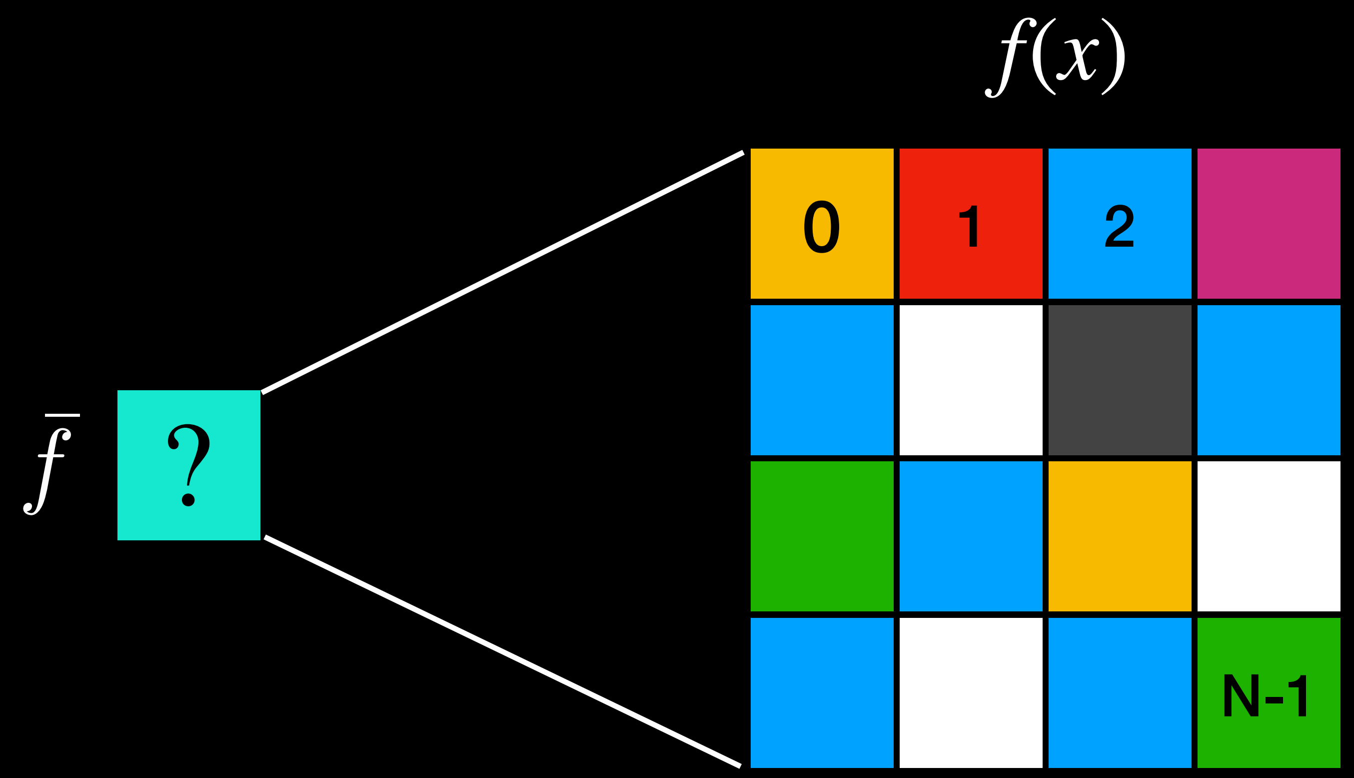
# Goals

- Utilize the **exponential** and **parallel** nature of qubits
  - Fundamentally faster than classical algorithms
  - Avoids the caveats of quantum computing
  - Works well on ***real*** quantum computers

# Pixel Supersampling



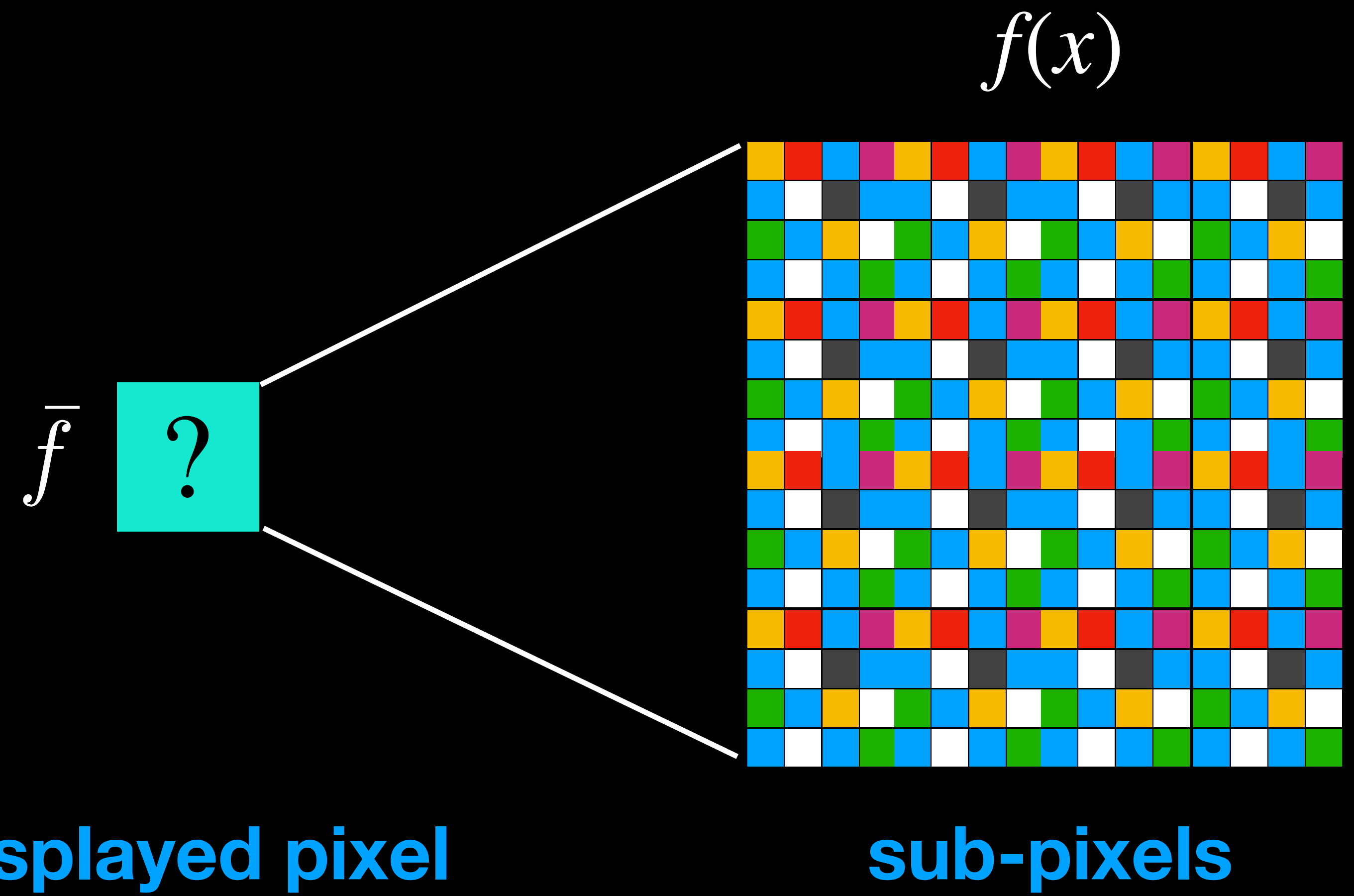
# Pixel Supersampling




**displayed pixel** **sub-pixels**

$$\bar{f} = \frac{1}{M} \sum_{i=1}^{M} f(x_i)$$

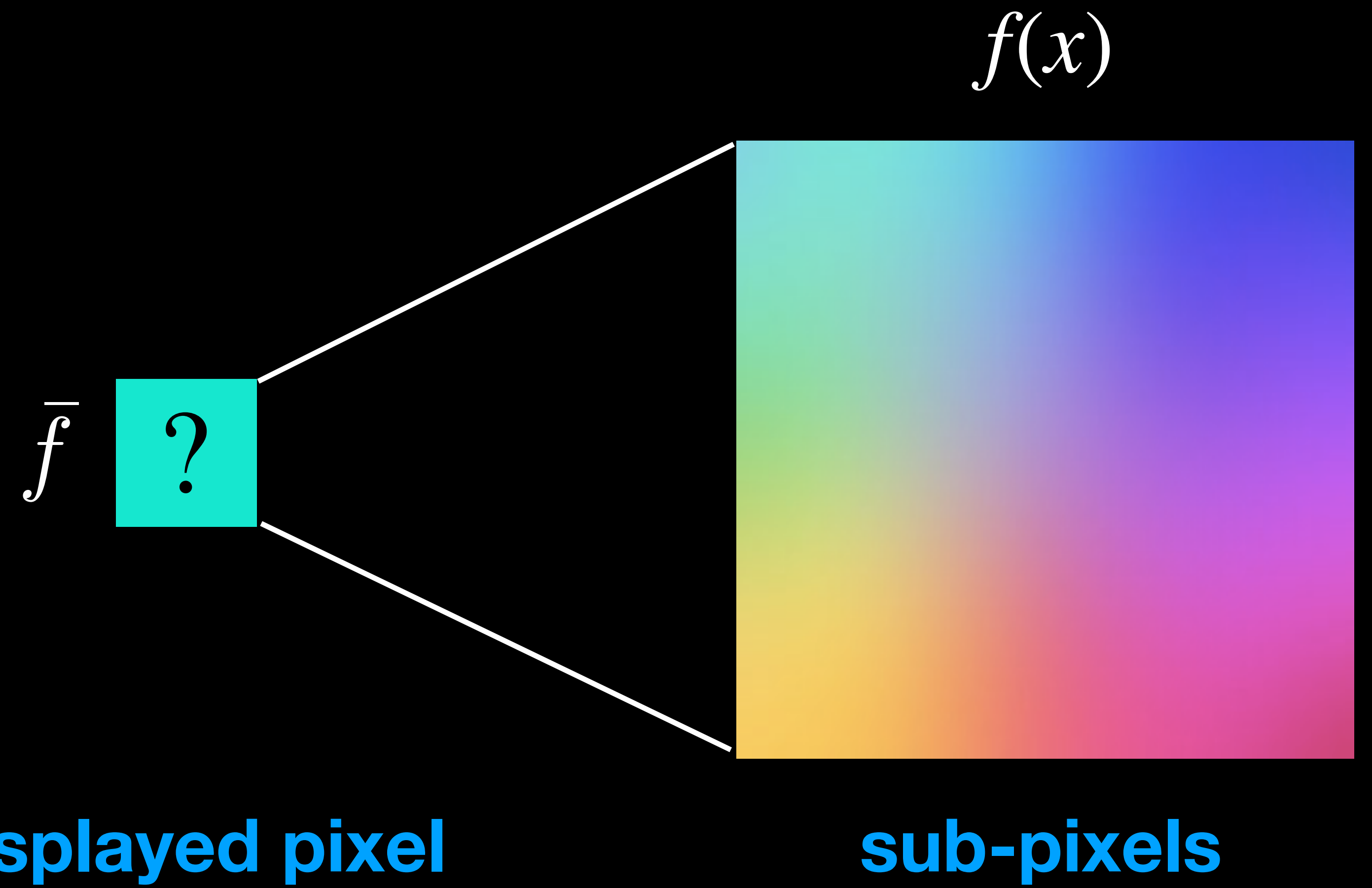
# Pixel Supersampling




$$\bar{f} = \frac{1}{M} \sum_{i=1}^{M} f(x_i)$$

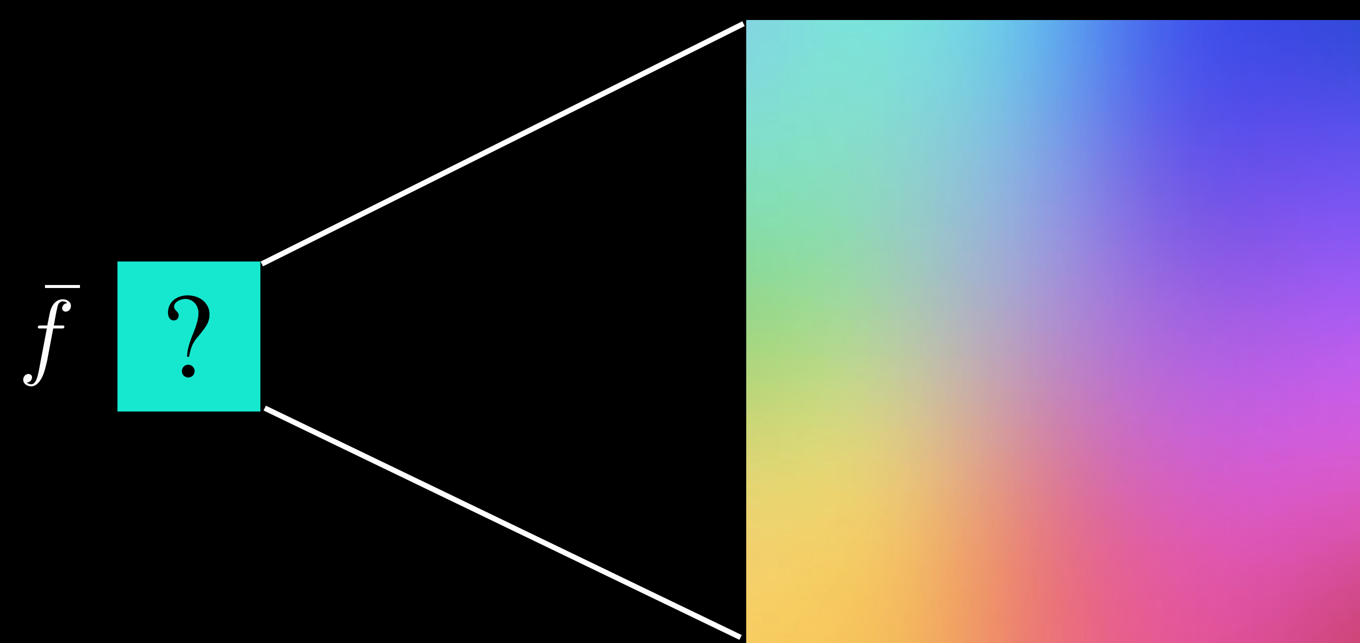
# Pixel Supersampling

Equivalent to integrating f(x) when $M = O(2^{32})$




$$\bar{f} = \int_P f(x)dx$$

# Monte Carlo Integration

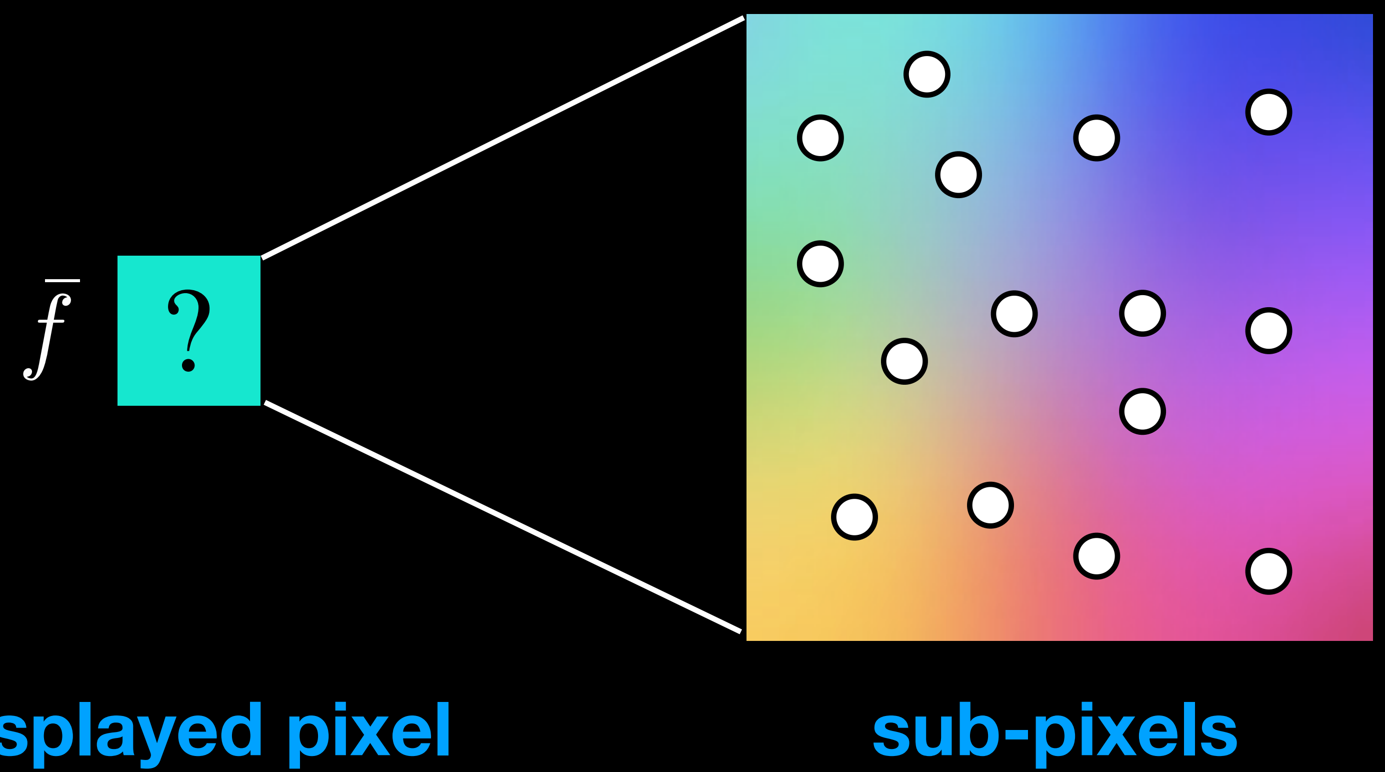



**displayed pixel** **sub-pixels**

$$\bar{f} = \int_P f(x)dx$$

# Monte Carlo Integration

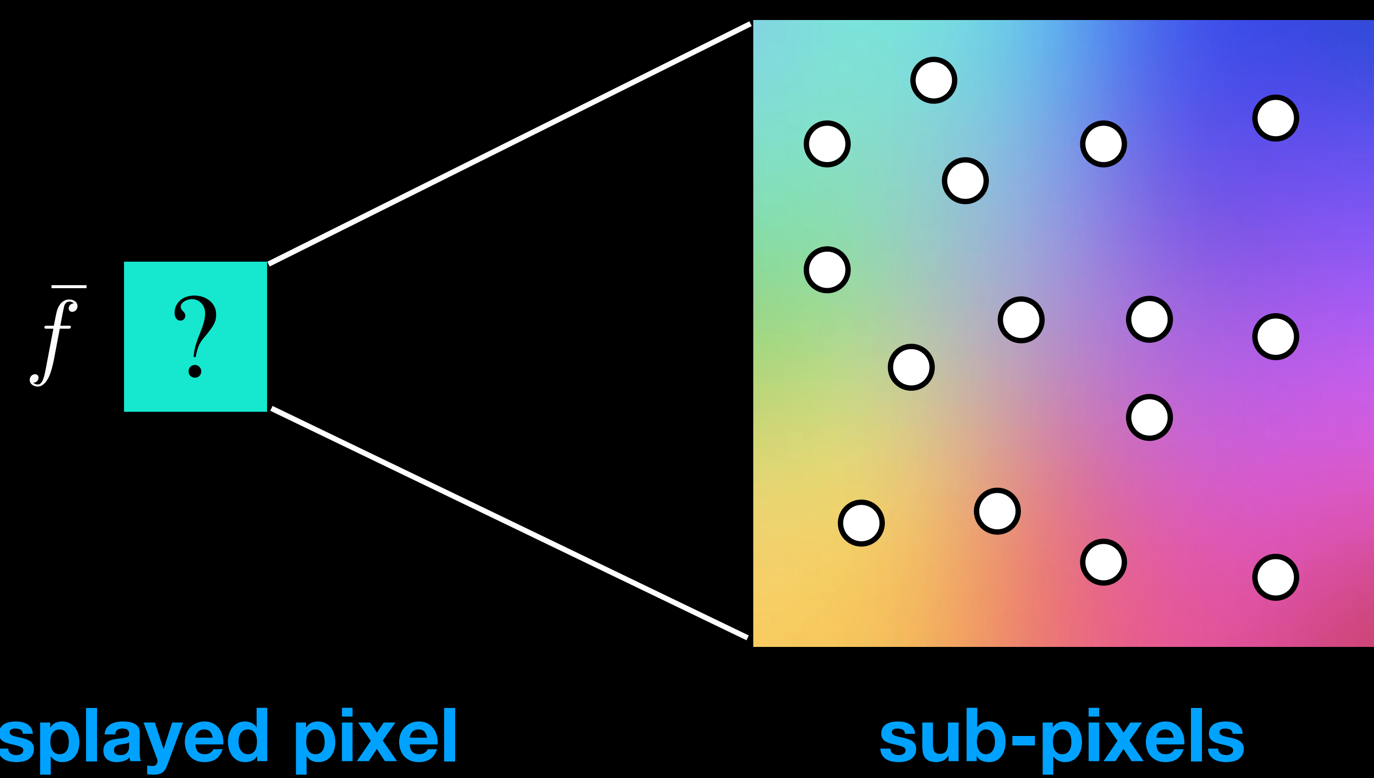
Take N samples $f(x_1), f(x_2), \cdots, f(x_N)$ $(N \ll M)$




$$\bar{f} = \int_P f(x)dx$$

# Monte Carlo Integration

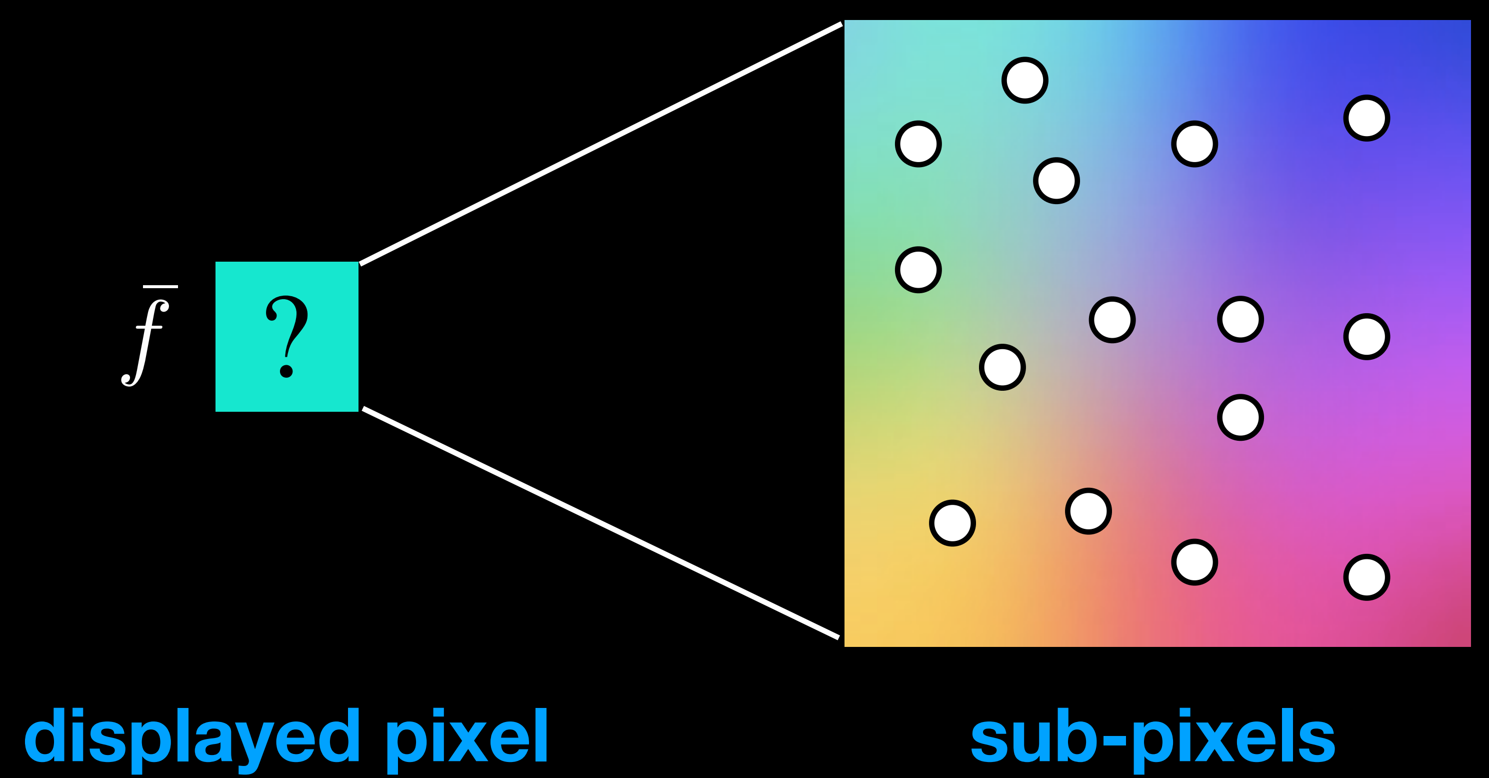
Take N samples $f(x_1), f(x_2), \cdots, f(x_N) \qquad (N \ll M)$




$$\bar{f} \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

# Monte Carlo Integration

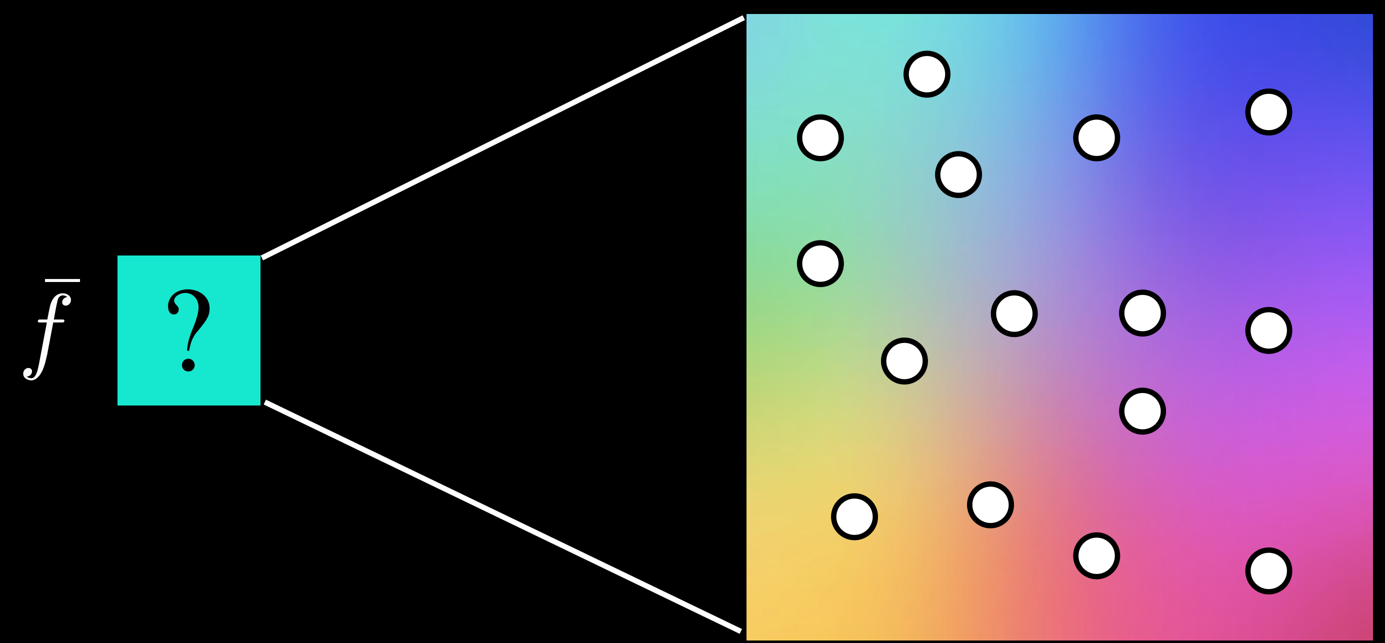
Take N samples $f(x_1), f(x_2), \cdots, f(x_N) \qquad (N \ll M)$




$$\left| \bar{f} - \frac{1}{N} \sum_{i=1}^{N} f(x_i) \right|$$

# Monte Carlo Integration

Take N samples $f(x_1), f(x_2), \cdots, f(x_N) \quad (N \ll M)$



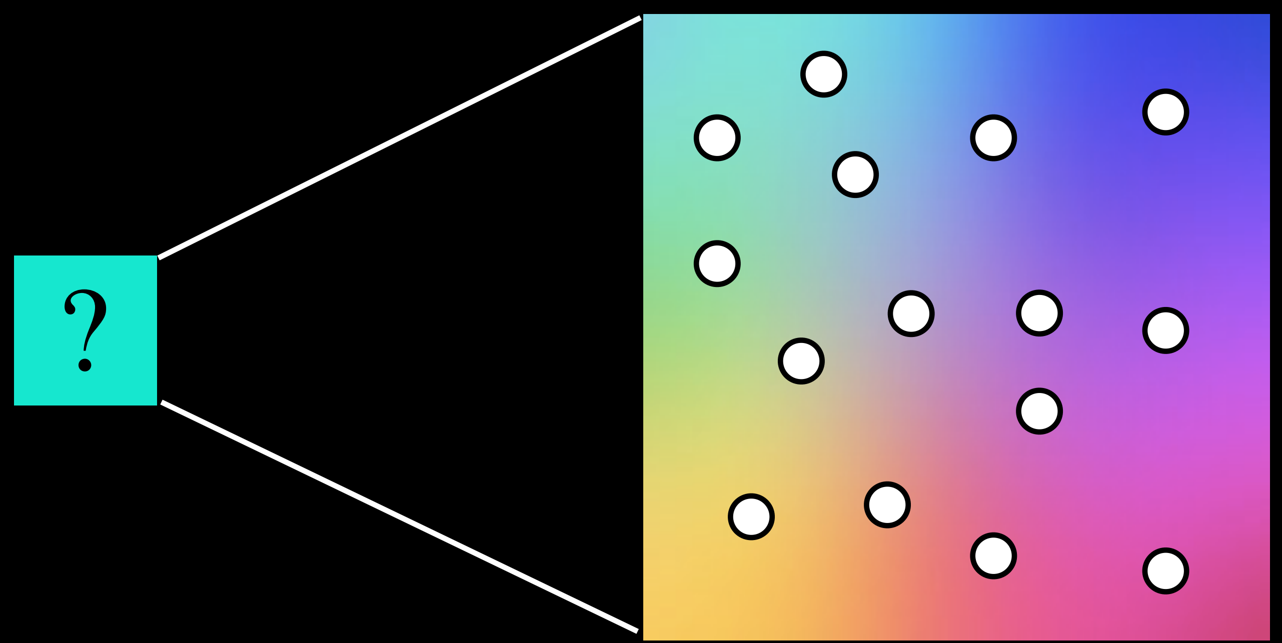

**displayed pixel** **sub-pixels**

$$\mathrm{E}\left|\bar{f} - \frac{1}{N}\sum_{i=1}^{N} f(x_i)\right| = \frac{\sigma}{\sqrt{N}}$$

# Monte Carlo Integration

Take N samples $f(x_1), f(x_2), \cdots, f(x_N) \qquad (N \ll M)$

$\bar{f}$




**displayed pixel** **sub-pixels**

$$\mathrm{E}\left|\bar{f} - \frac{1}{N}\sum_{i=1}^{N} f(x_i)\right| = \boxed{O\left(\frac{1}{\sqrt{N}}\right)}$$

# Classical Algorithm

- Monte Carlo integration
  - Evaluate the integrand N times $(N \ll M)$
  - Error converges at $O\left(\frac{1}{\sqrt{N}}\right)$
    - Independent of the number of dimensions
    - Proportional to the std. deviation of the integrand

# Classical Algorithm

- Monte Carlo integration
  - Evaluate the integrand N times
  - Error converges at $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$
    - Independent of the number of dimensions
    - Proportional to the std. deviation of the integrand

**Can we do better than this using quantum computing?**

$$\bar{f} = \int_P f(x)dx$$

$$\bar{f} = \frac{1}{M} \sum_{i=1}^{M} f(x_i)$$

$M = O(2^{32})$ **(precision of 32-bit numbers)**

$$\bar{f} = \frac{1}{M} \boxed{\sum_{i=1}^{M} f(x_i)}$$ **needs $2^{32}$ evaluations**

$$M = O(2^{32})$$ **(precision of 32-bit numbers)**

$$\bar{f} = \frac{1}{M} \boxed{\sum_{i=1}^{M} f(x_i)}$$ **needs $2^{32}$ evaluations**

$$M = O(2^{32})$$ **(precision of 32-bit numbers)**

**Use 32 qubits to perform all the $2^{32}$ evaluations in parallel!**

# Basic Idea

**Superposition of 32 qubits** $|0\rangle + |1\rangle + \cdots + |2^{32} - 1\rangle$

# Basic Idea

**Superposition of 32 qubits** $|0\rangle + |1\rangle + \cdots + |2^{32} - 1\rangle$

**one shot!**

**$2^{32}$ evaluations of** $f(0)|0\rangle + f(1)|1\rangle + \cdots + f(2^{32} - 1)|2^{32} - 1\rangle$

# Basic Idea

**Superposition of 32 qubits** $|0\rangle + |1\rangle + \cdots + \left|2^{32} - 1\right\rangle$

**one shot!**

**$2^{32}$ evaluations of** $f(0)\,|0\rangle + f(1)\,|1\rangle + \cdots + f(2^{32} - 1)\,\left|2^{32} - 1\right\rangle$

**quantum operations**

**1 qubit with the exact answer** $\sqrt{1 - \bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$

# Basic Idea

Superposition of 32 qubits $|0\rangle + |1\rangle + \cdots + \left|2^{32} - 1\right\rangle$

⬇ one shot!

$2^{32}$ evaluations of $f(0)\,|0\rangle + f(1)\,|1\rangle + \cdots + f(2^{32} - 1)\left|2^{32} - 1\right\rangle$

⬇ quantum operations

1 qubit with the exact answer $\sqrt{1 - \bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$

$$\bar{f} = \int_P f(x)dx = \frac{1}{M}\sum_{i=1}^{M} f(x_i)$$

# Basic Idea

**Superposition of 32 qubits** $|0\rangle + |1\rangle + \cdots + \left|2^{32} - 1\right\rangle$

**one shot!**

**$2^{32}$ evaluations of** $f(0)\,|0\rangle + f(1)\,|1\rangle + \cdots + f(2^{32} - 1)\left|2^{32} - 1\right\rangle$

**quantum operations**

**1 qubit with the exact answer** $\sqrt{1 - \bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$

$$\bar{f} = \int_P f(x)dx = \frac{1}{M}\sum_{i=1}^{M} f(x_i)$$

**std. deviation is irrelevant!**

# Basic Idea

Superposition of 32 qubits $|0\rangle + |1\rangle + \cdots + |2^{32} - 1\rangle$

one shot!

$2^{32}$ evaluations of $f(0)|0\rangle + f(1)|1\rangle + \cdots + f(2^{32} - 1)|2^{32} - 1\rangle$

quantum operations

**How do we estimate $\bar{f}$ ?**

$$\sqrt{1 - \bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

1 qubit with the exact answer

$$\bar{f} = \int_P f(x)dx = \frac{1}{M}\sum_{i=1}^{M} f(x_i)$$

std. deviation is irrelevant!

# Amplitude Estimation

- Cannot directly measure the probability of a qubit state

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

# Amplitude Estimation

- Cannot directly measure the probability of a qubit state

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

read-out (measurement)

$$\begin{cases} |0\rangle & (\text{probability of } 1-\bar{f}^2) \\ |1\rangle & (\text{probability of } \bar{f}^2) \end{cases}$$

# Amplitude Estimation

- Cannot directly measure the probability of a qubit state

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

**read-out (measurement)**

$$\begin{cases} |0\rangle & (\text{probability of } 1-\bar{f}^2) \\ |1\rangle & (\text{probability of } \bar{f}^2) \end{cases}$$

**Naive idea: Count the number of $|1\rangle$s you get over N readouts**

# Amplitude Estimation

- Cannot directly measure the probability of a qubit state

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

read-out (measurement)

$$\begin{cases} |0\rangle & \text{(probability of } 1-\bar{f}^2\text{)} \\ |1\rangle & \text{(probability of } \bar{f}^2\text{)} \end{cases}$$

**Naive idea: Count the number of $|1\rangle$s you get over N readouts**

$O\left(\frac{1}{\sqrt{N}}\right)$ **convergence rate again!**

# Quantum Amplitude Estimation

- Estimate the amplitude using quantum computing

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

# Quantum Amplitude Estimation

- Estimate the amplitude using quantum computing

$$\sqrt{1 - \bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

- Two algorithms with faster convergence rates:
    - Quantum super sampling [Johnston 2016]
    - Quantum coin (ours)

# Quantum Supersampling

**[Johnston 2016]**

# Quantum Supersampling

**[Johnston 2016]**

- Original idea is by Grover in 1998
  - Quantum state can be “rotated” by some operations
  - The frequency of this “rotation” depends on $\bar{f}$
  - Evaluate the frequency by quantum Fourier transform

# Quantum Supersampling

**[Johnston 2016]**

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

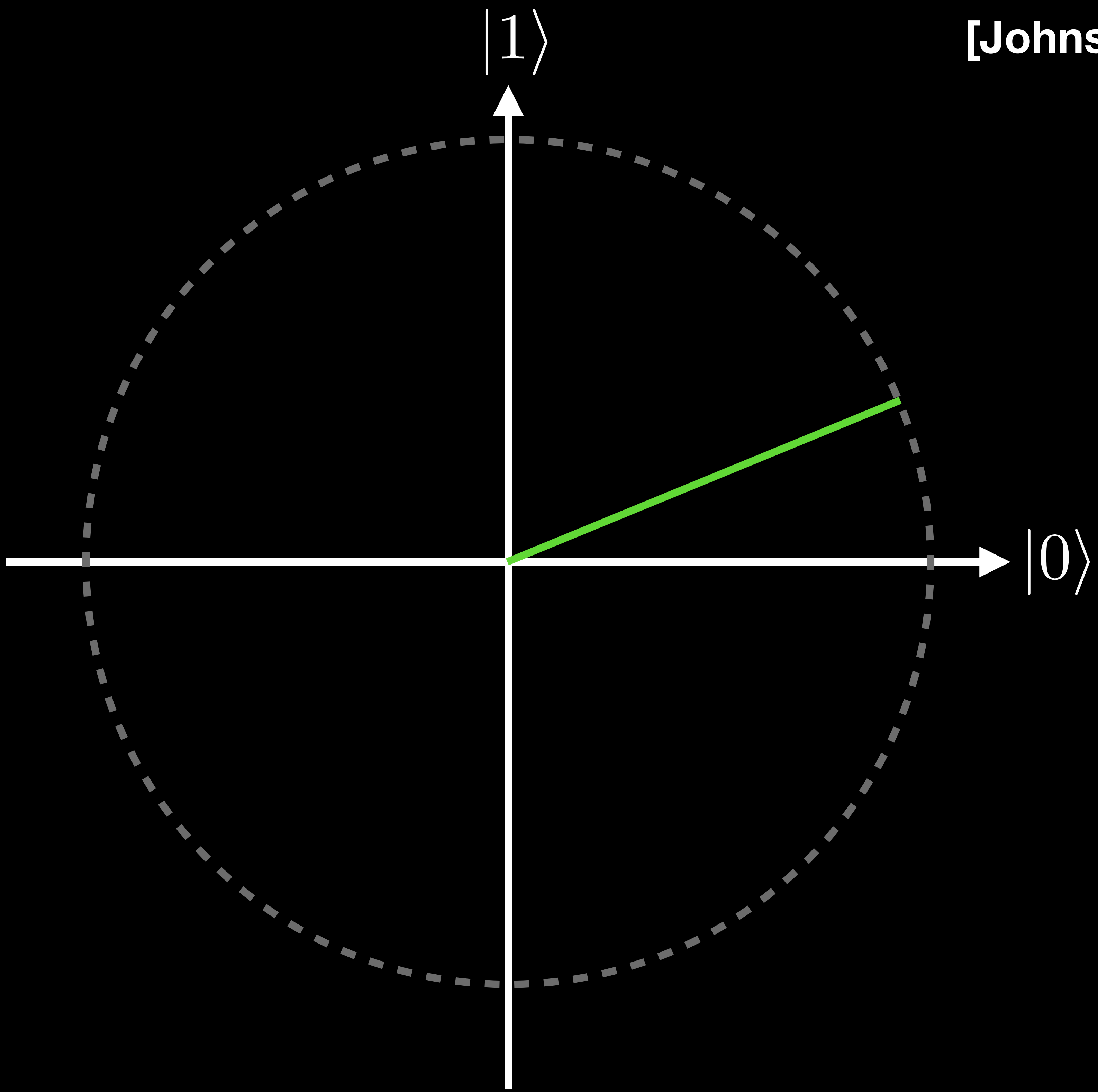
# Quantum Supersampling

**[Johnston 2016]**

$$\cos\theta\,|0\rangle + \sin\theta\,|1\rangle$$

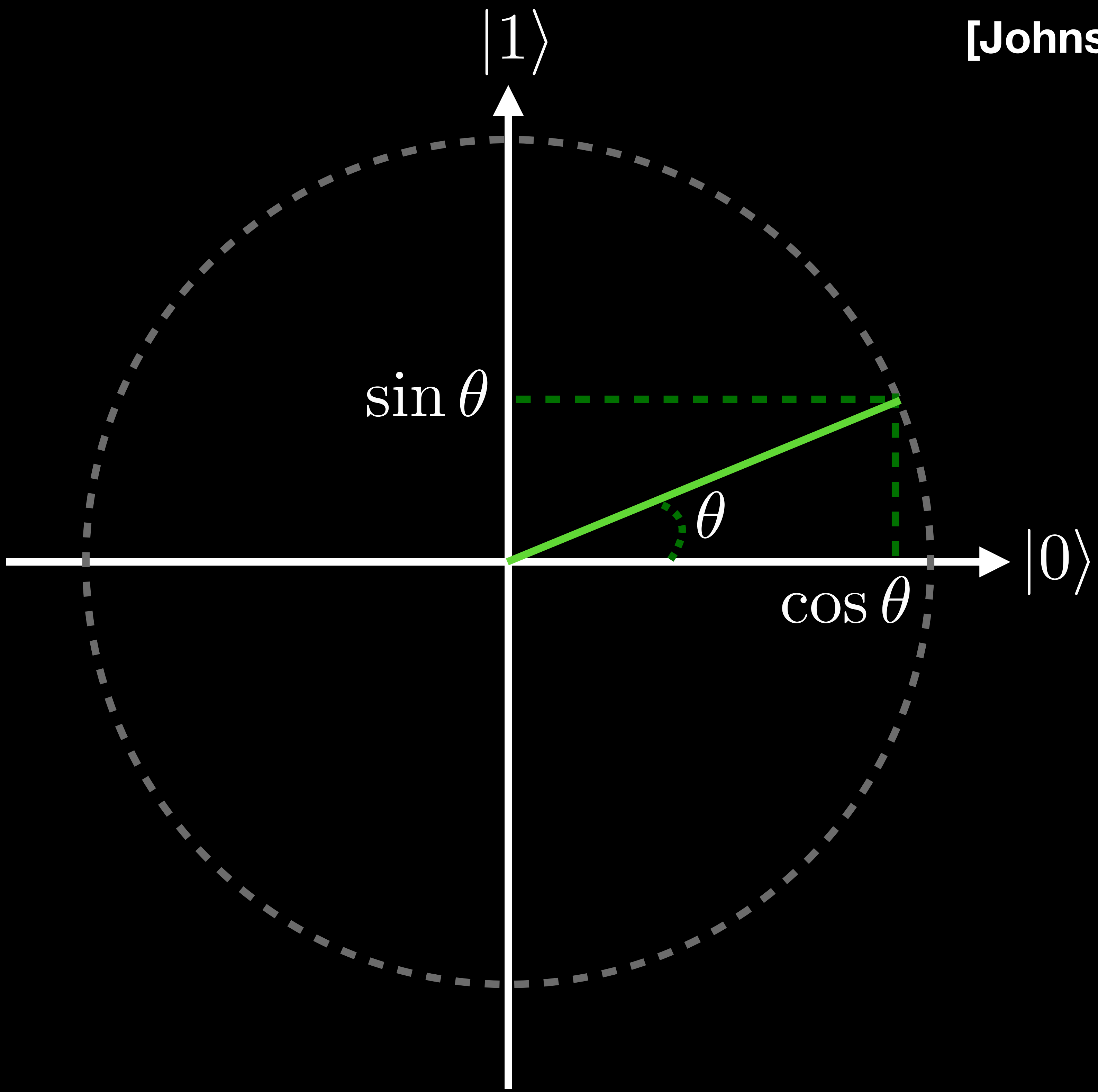
# Quantum Supersampling

[Johnston 2016]



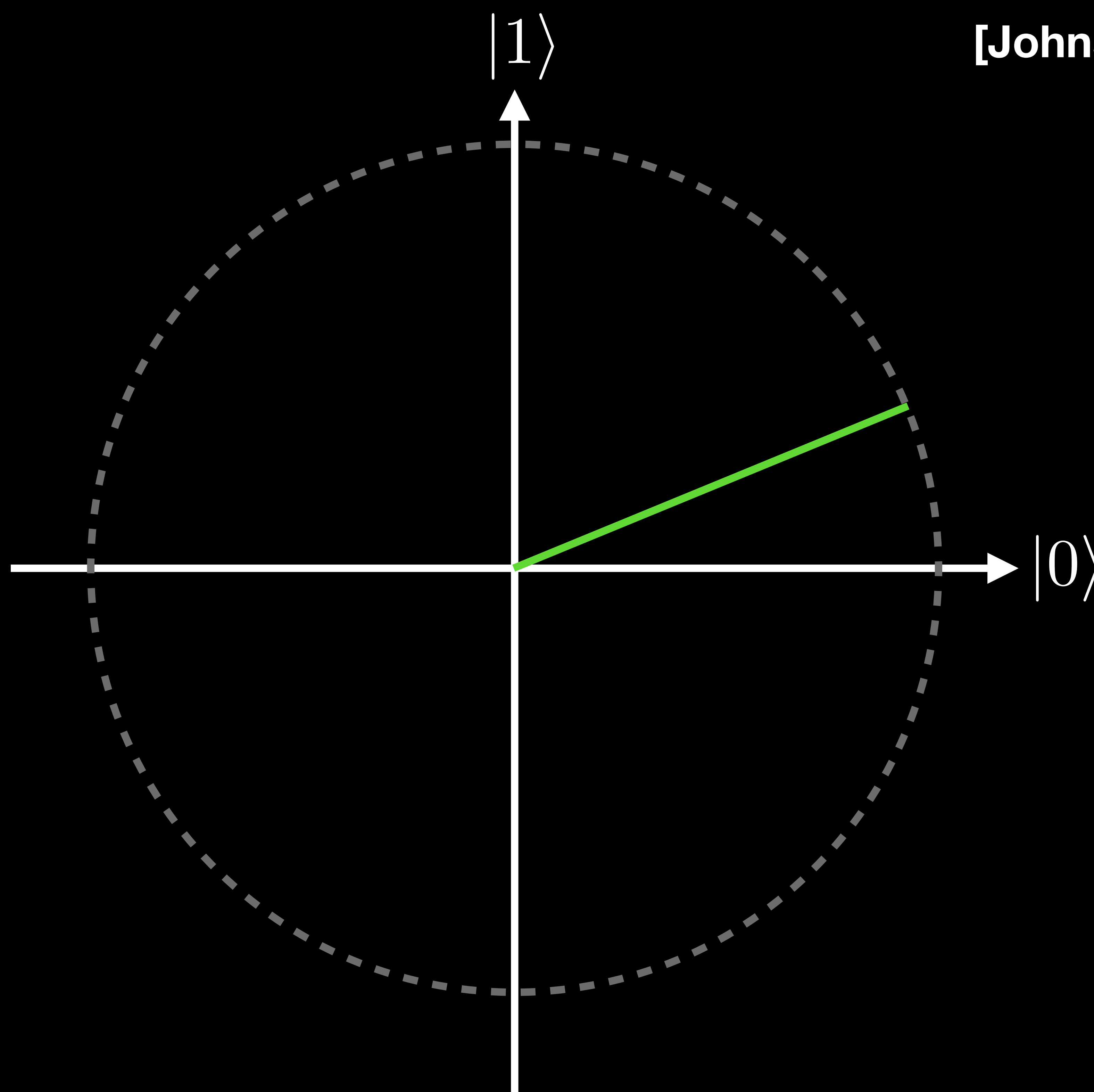
# Quantum Supersampling

**[Johnston 2016]**



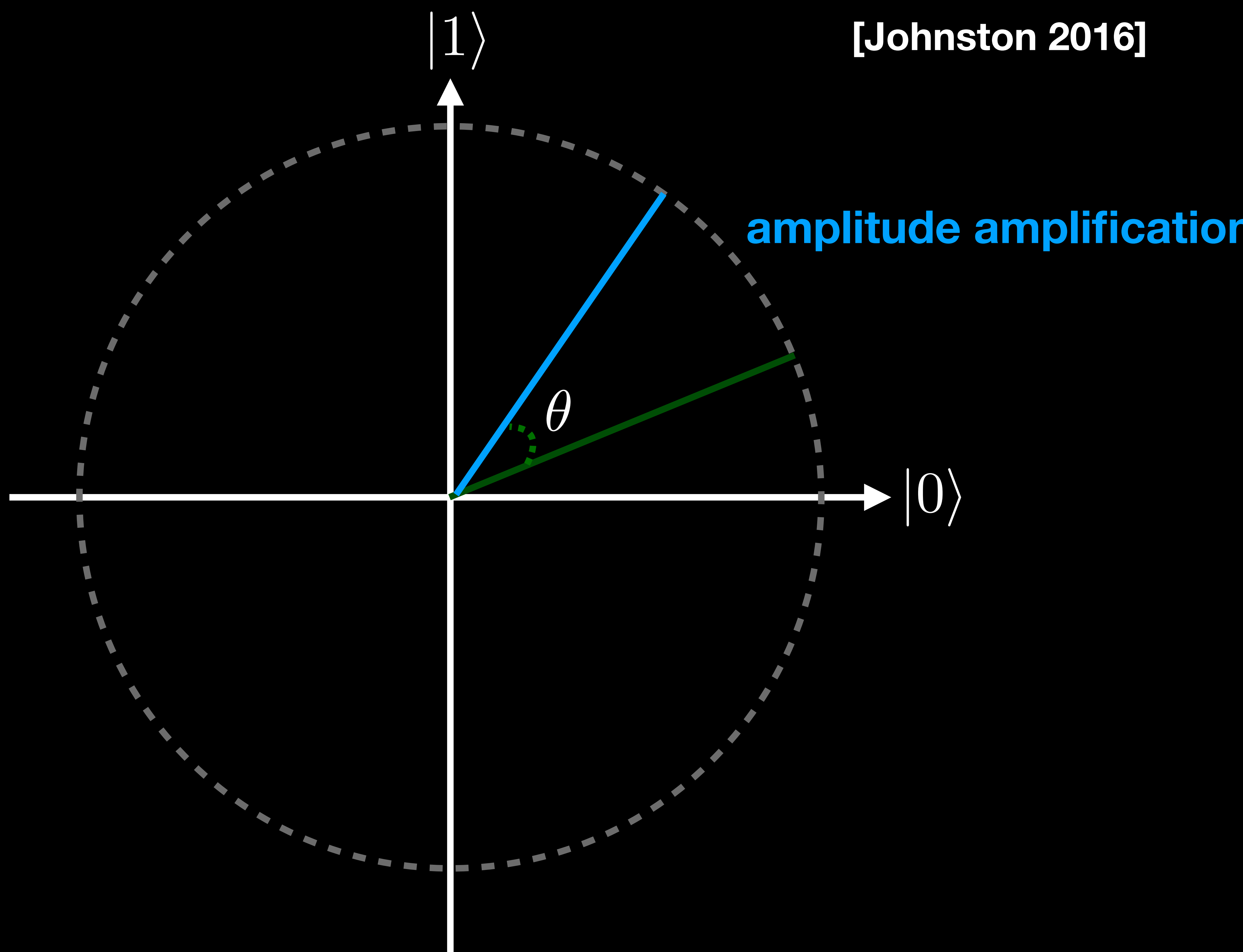
# Quantum Supersampling

**[Johnston 2016]**



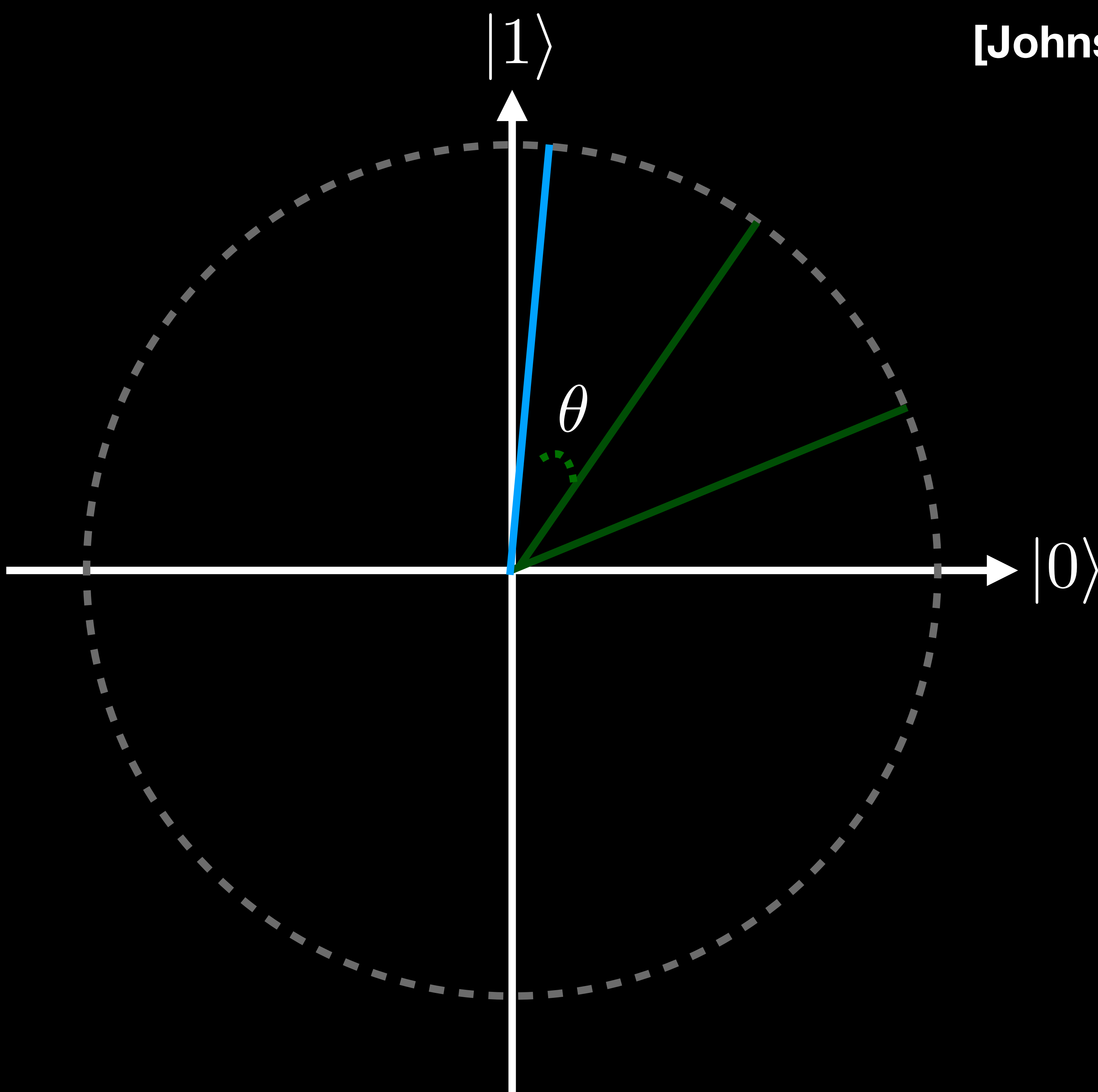
# Quantum Supersampling

[Johnston 2016]



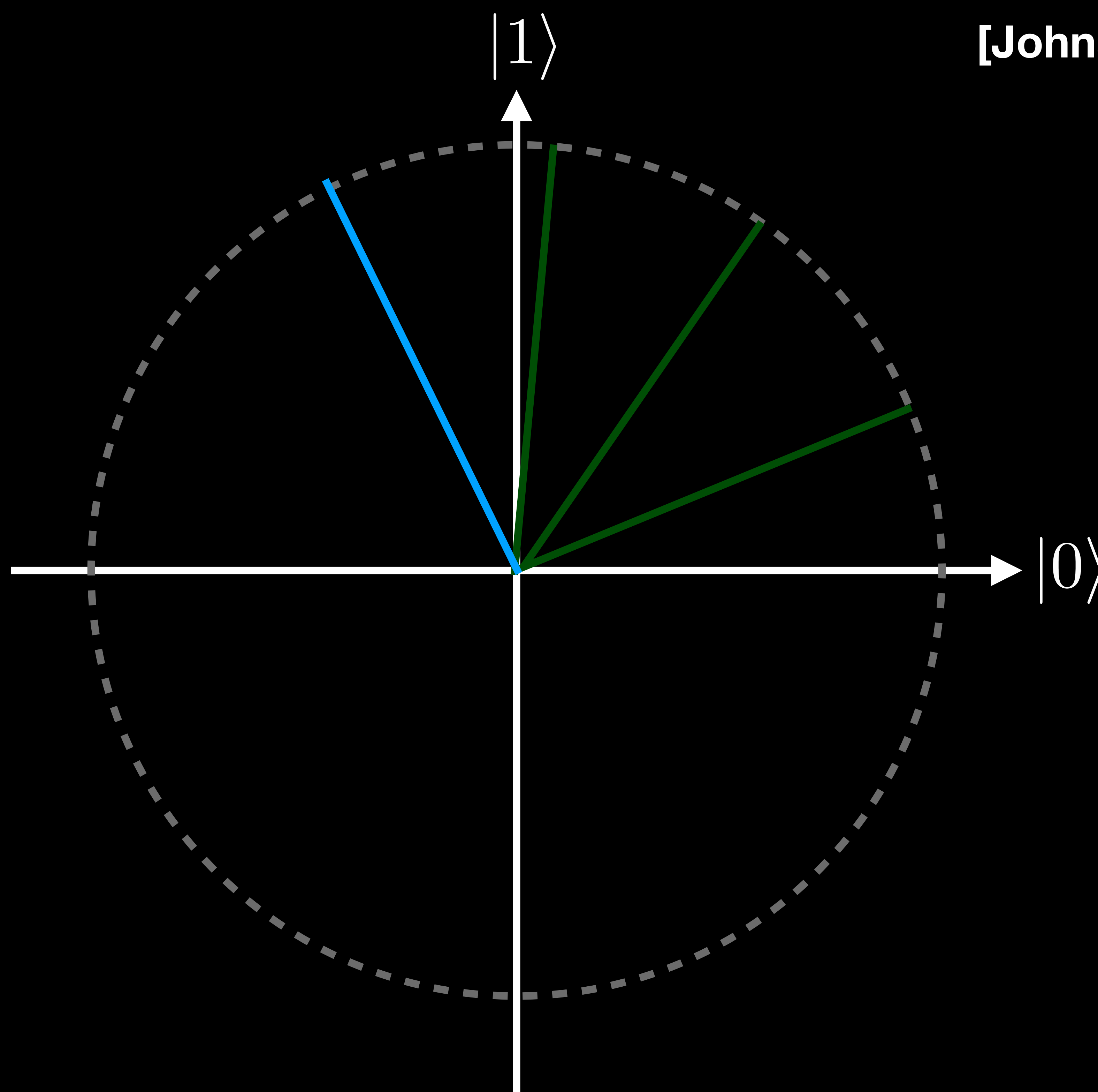
# Quantum Supersampling

[Johnston 2016]



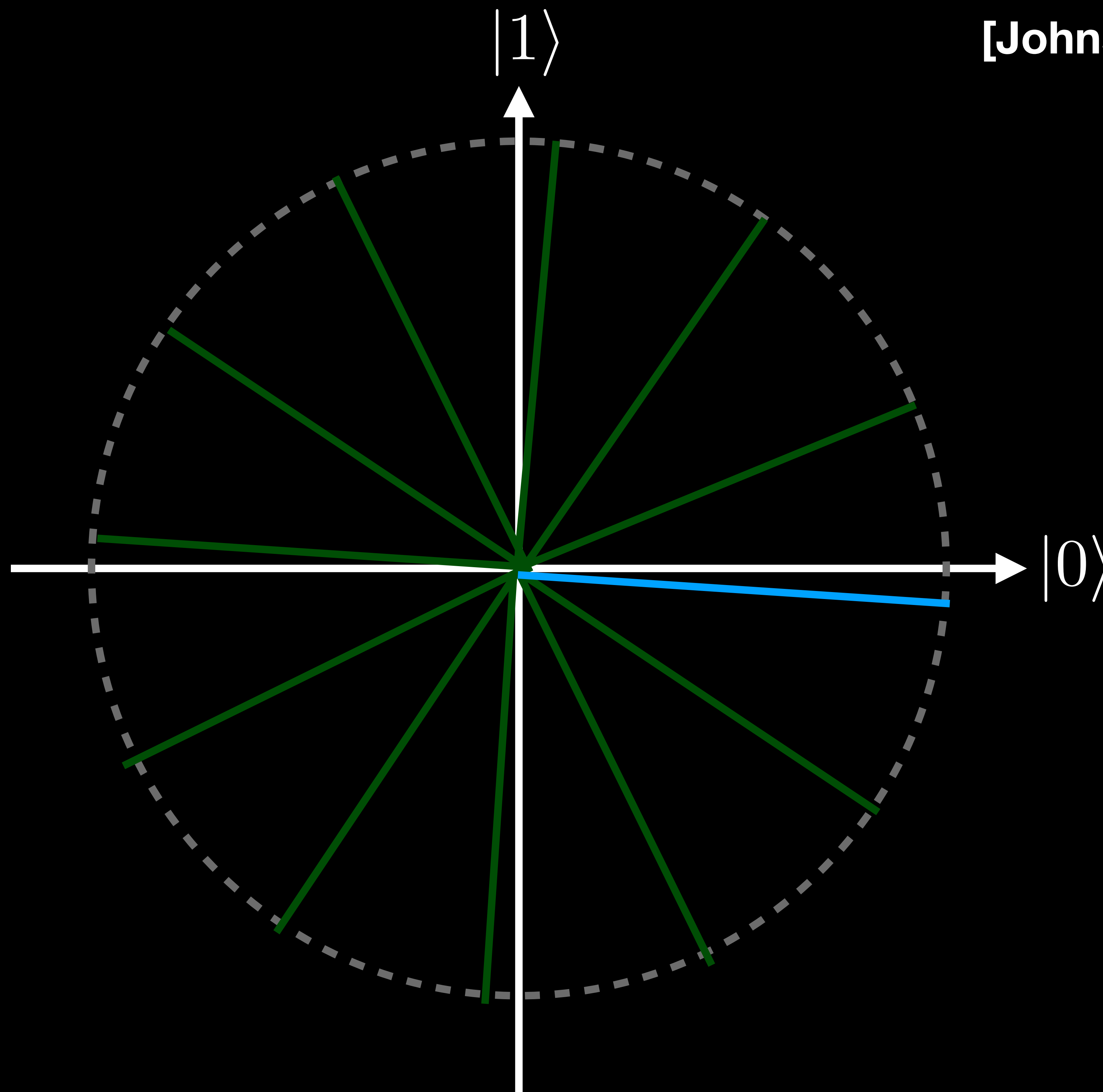
# Quantum Supersampling

[Johnston 2016]



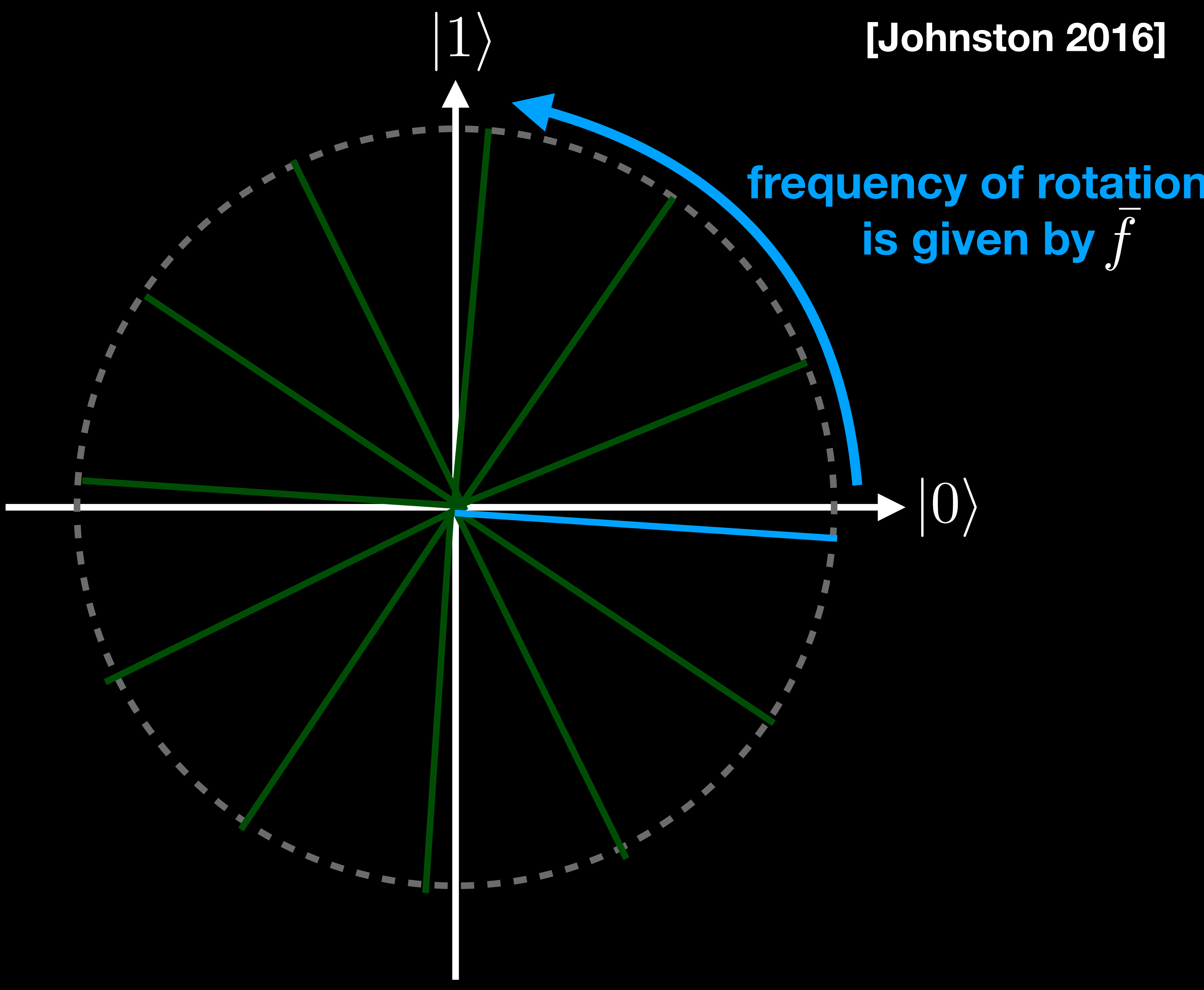
# Quantum Supersampling

[Johnston 2016]



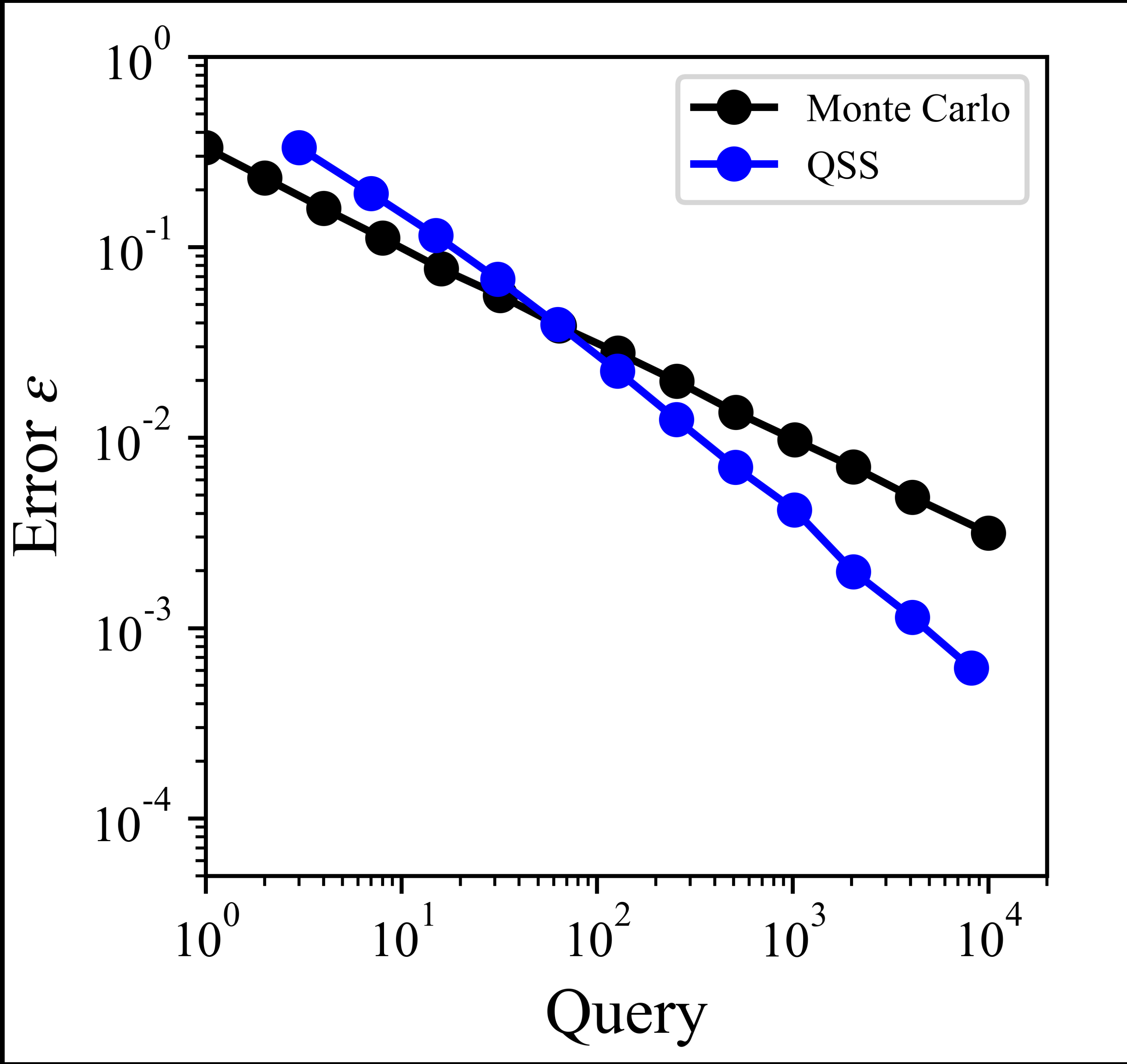
# Quantum Supersampling

[Johnston 2016]



# Quantum Supersampling

[Johnston 2016]




**Monte Carlo :**

$$O\left(\frac{1}{N^{0.5}}\right)$$ **error**

**QSS :**

$$O\left(\frac{1}{N^{0.85}}\right)$$ **error**

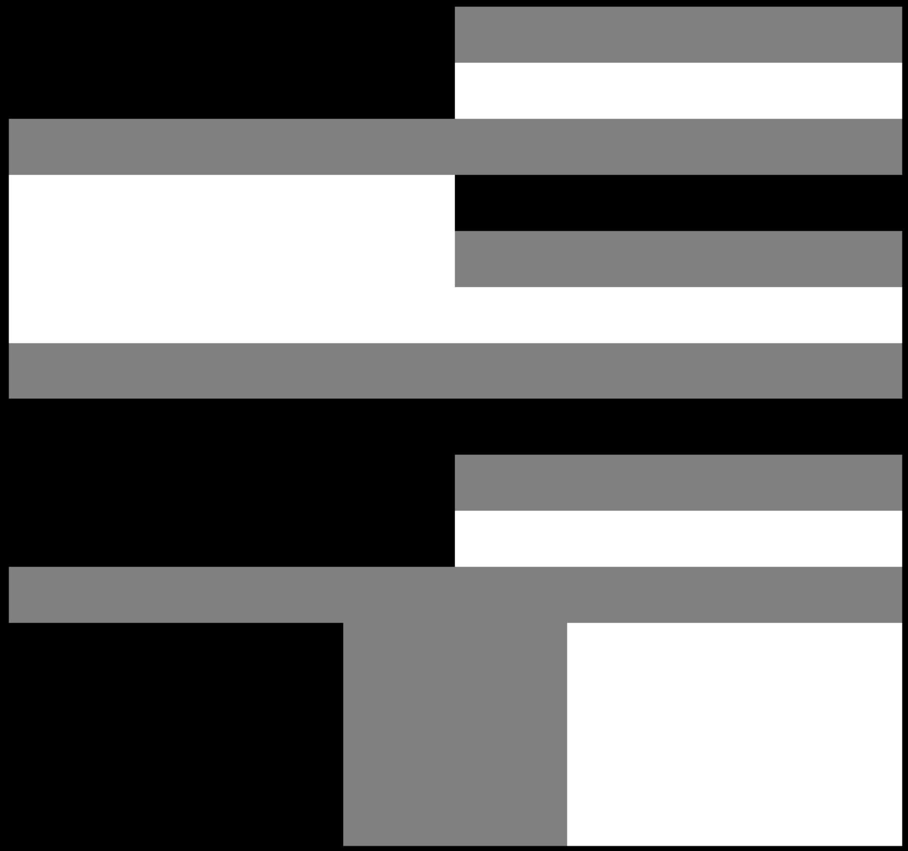
# Quantum Supersampling

[Johnston 2016]

On a simulator

# Quantum Supersampling

[Johnston 2016]



On a simulator

On a real quantum computer

# Quantum Supersampling

[Johnston 2016]

On a simulator

On a real quantum computer

Results are corrupted by noise in quantum computers!

# Quantum Supersampling

**[Johnston 2016]**

- Based on amplitude amplification + QFT by Grover 1998
  - Rotate the state and estimate its frequency
  - Each rotation takes $O(1)$
  - Asymptotically faster than Monte Carlo integration
  - Variance is no longer relevant to the error

# Quantum Supersampling

**[Johnston 2016]**

- Based on amplitude amplification + QFT by Grover 1998
  - Rotate the state and estimate its frequency
  - Each rotation takes $O(1)$
  - Asymptotically faster than Monte Carlo integration
  - Variance is no longer relevant to the error
- Does not work well on real quantum computers

# Quantum Coin

# Quantum Coin

- Original theory is by Abrams and Williams in 1999
  - Hybrid of classical/quantum algorithms
  - Uses estimated error intervals by Monte Carlo
  - Adaptively shrinks error intervals by quantum algorithms

# Quantum Coin

- Original theory is by Abrams and Williams in 1999
  - Hybrid of classical/quantum algorithms
  - Uses estimated error intervals by Monte Carlo
  - Adaptively shrinks error intervals by quantum algorithms
- We are the first to actually implement this algorithm

# Initialization

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle$$

# Initialization

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle \quad \begin{cases} |0\rangle & \text{(probability of } 1-\bar{f}^2\text{)} \\ |1\rangle & \text{(probability of } \bar{f}^2\text{)} \end{cases}$$

# Initialization

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle \quad \begin{cases} |0\rangle & (\text{probability of } 1-\bar{f}^2) \\ |1\rangle & (\text{probability of } \bar{f}^2) \end{cases}$$

↓ **MC integration with $N_0$ samples**

# Initialization

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle \quad \begin{cases} |0\rangle & (\text{probability of } 1-\bar{f}^2) \\ |1\rangle & (\text{probability of } \bar{f}^2) \end{cases}$$

**MC integration with $N_0$ samples**

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0}\sum_{i=1}^{N_0} 1_{|1\rangle}$$

# Initialization

$$\sqrt{1-\bar{f}^2}\,|0\rangle + \bar{f}\,|1\rangle \quad \begin{cases} |0\rangle & (\text{probability of } 1-\bar{f}^2) \\ |1\rangle & (\text{probability of } \bar{f}^2) \end{cases}$$

**MC integration with $N_0$ samples**

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0} \boxed{\sum_{i=1}^{N_0} 1_{|1\rangle}}$$

**The number of times you read $|1\rangle$**

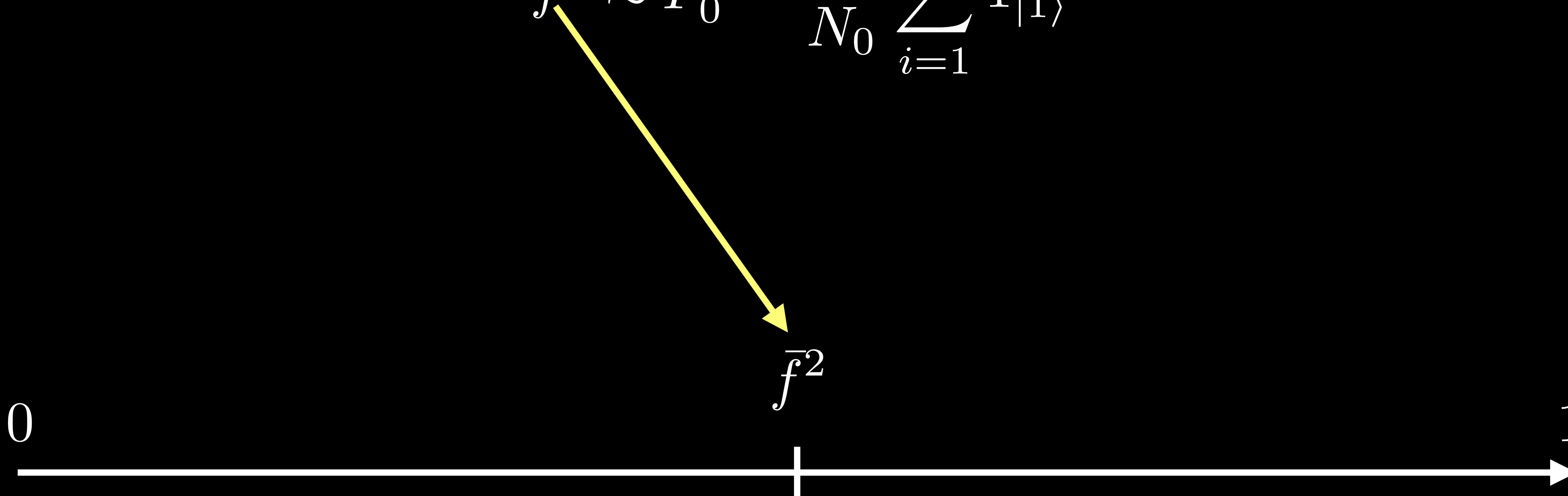
**(like flipping a coin and counting the number of heads)**

# Initialization

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0} \sum_{i=1}^{N_0} 1_{|1\rangle}$$

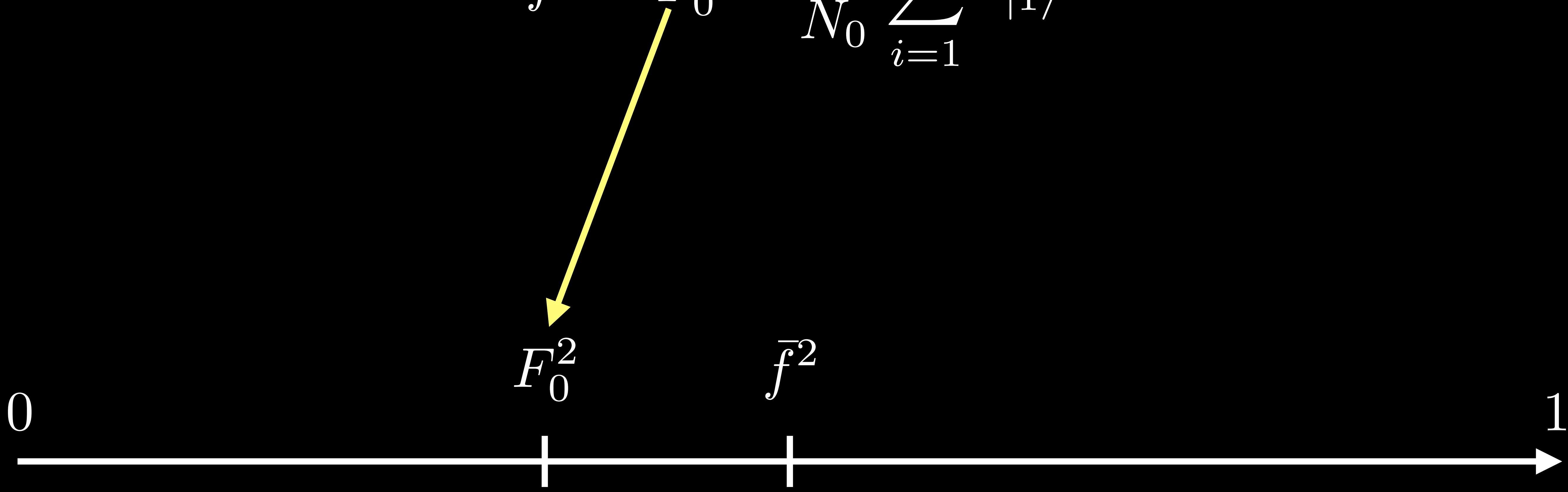
# 1st Iteration

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0} \sum_{i=1}^{N_0} 1_{|1\rangle}$$



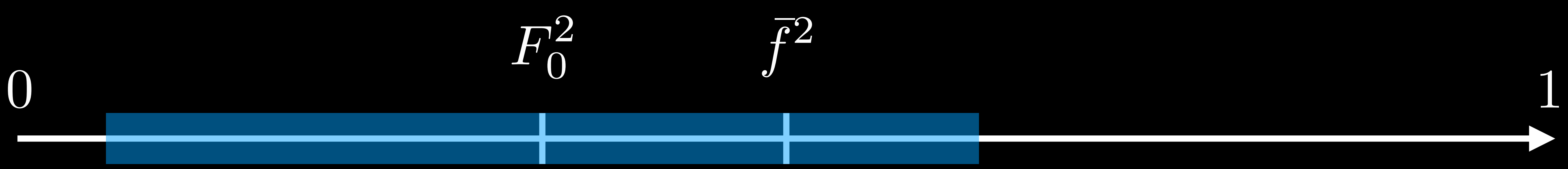
# 1st Iteration

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0}\sum_{i=1}^{N_0} 1_{|1\rangle}$$



# 1st Iteration

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0} \sum_{i=1}^{N_0} 1_{|1\rangle}$$

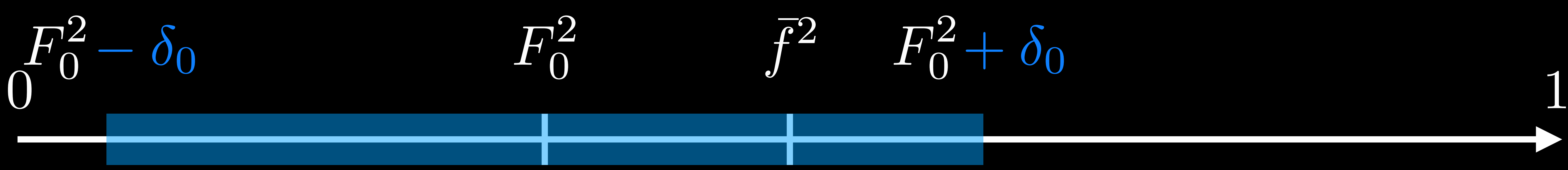



**confidence internal**
**(error interval)**

# 1st Iteration

$$\bar{f}^2 \approx F_0^2 = \frac{1}{N_0} \sum_{i=1}^{N_0} 1_{|1\rangle}$$

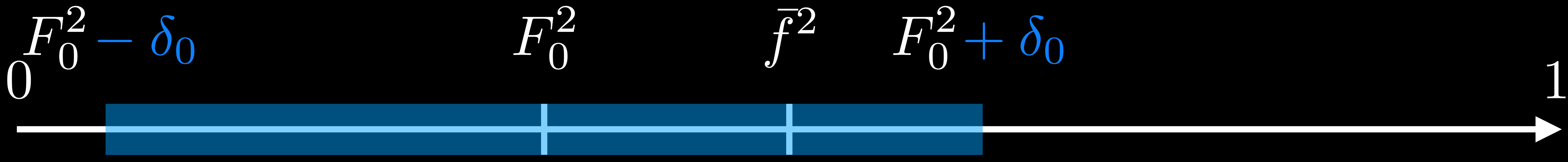
$$\bar{f}^2 \in \left[F_0^2 - \delta_0, F_0^2 + \delta_0\right]$$




**confidence internal**
**(error interval)**

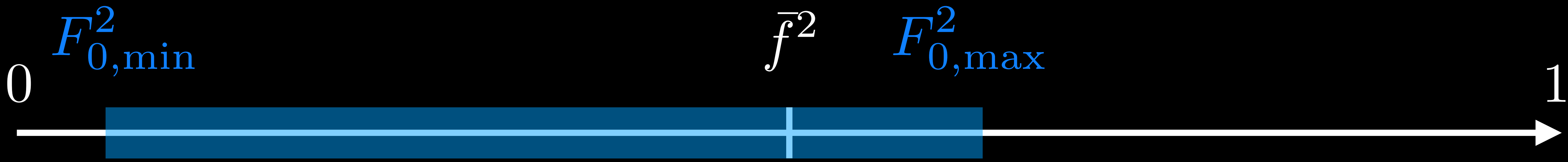
# 1st Iteration

$$\bar{f}^2 \in [F_0^2 - \delta_0, F_0^2 + \delta_0]$$



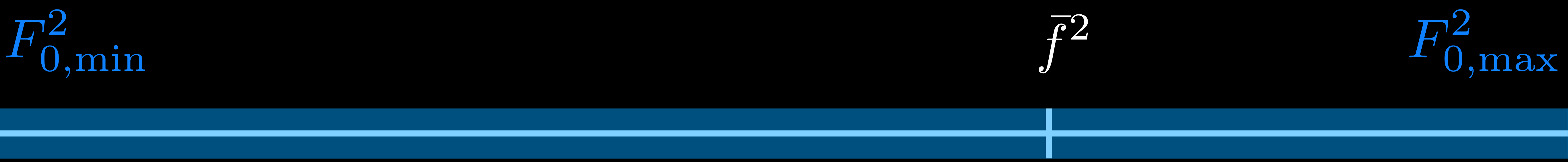
# 1st Iteration

$$\bar{f}^2 \in [\ F^2_{0,\mathrm{min}}\ ,\ F^2_{0,\mathrm{max}}\ ]$$



# 1st Iteration

$$\bar{f}^2 \in [\, F^2_{0,\min} \,,\, F^2_{0,\max} \,]$$



# 1st Iteration

$\bar{f}^2 \in [\, F_{0,\min}^2 \,,\, F_{0,\max}^2 \,]$ ➡ $\bar{f}_0^2 \in [0, 1]$

**shifting & scaling**


$$\bar{f}_0^2 = \frac{\bar{f}^2 - F_{0,\min}^2}{F_{0,\max}^2 - F_{0,\min}^2}$$

# 1st Iteration

$\bar{f}^2 \in [\,F^2_{0,\min}\ ,\ F^2_{0,\max}\,]$ $\Rightarrow$ $\bar{f}^2_0 \in [0, 1]$

**shifting & scaling**

0 $\bar{f}^2_0$ 1

$$\bar{f}^2_0 = \frac{\bar{f}^2 - F^2_{0,\min}}{F^2_{0,\max} - F^2_{0,\min}}$$

$\Downarrow$ **construct a shifted & scaled coin**

$$\sqrt{1-\bar{f}^2_0}\,|0\rangle + \bar{f}_0\,|1\rangle$$

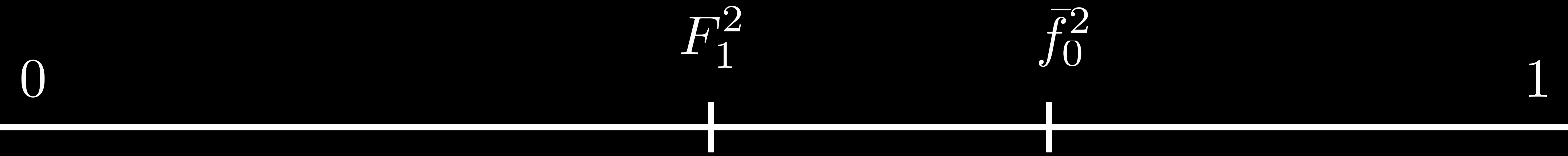
# 2nd Iteration

$$\sqrt{1-\bar{f}_0^2}\,|0\rangle + \bar{f}_0\,|1\rangle$$

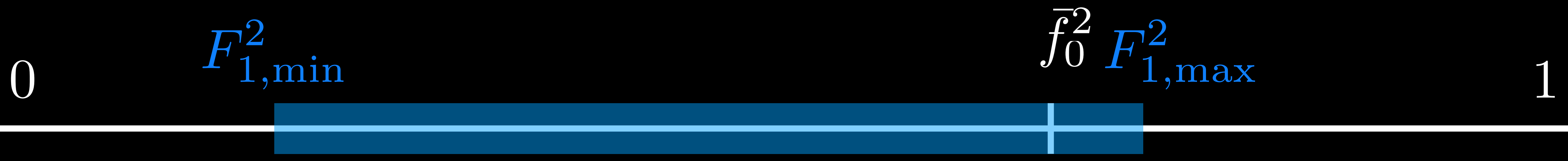
# 2nd Iteration

**MC integration with $N_1$ samples**

$$\sqrt{1-\bar{f}_0^2}\,|0\rangle + \bar{f}_0\,|1\rangle \quad \Rightarrow \quad \bar{f}_0^2 \approx F_1^2 = \frac{1}{N_1}\sum_{i=1}^{N_1} 1_{|1\rangle}$$

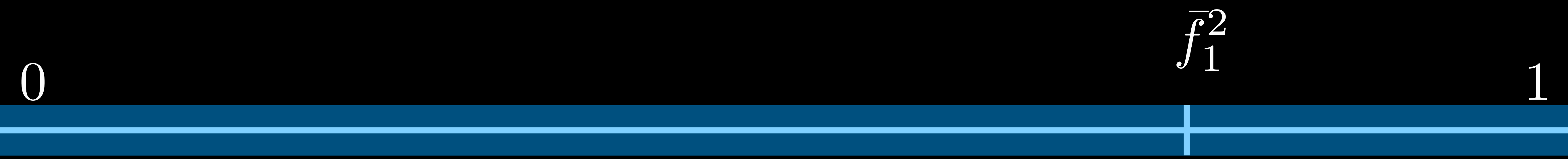


# 2nd Iteration




$$\bar{f}^2_0 \in [F^2_{1,\min}, F^2_{1,\max}]$$

# 2nd Iteration



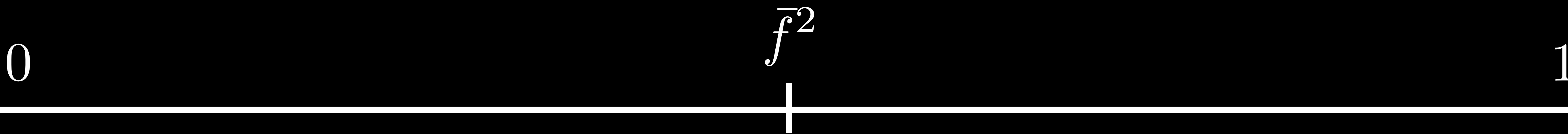

$$\bar{f}_1^2 = \frac{\bar{f}_0^2 - F_{1,\min}^2}{F_{1,\max}^2 - F_{1,\min}^2}$$

# 2nd Iteration

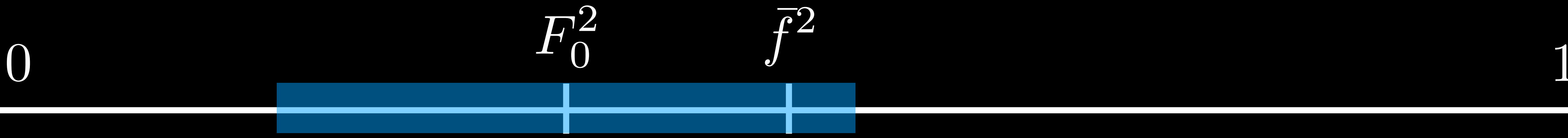

$$\sqrt{1-\bar{f}_1^2}\,|0\rangle + \bar{f}_1\,|1\rangle$$ **another shifted & scaled coin**

# Overview

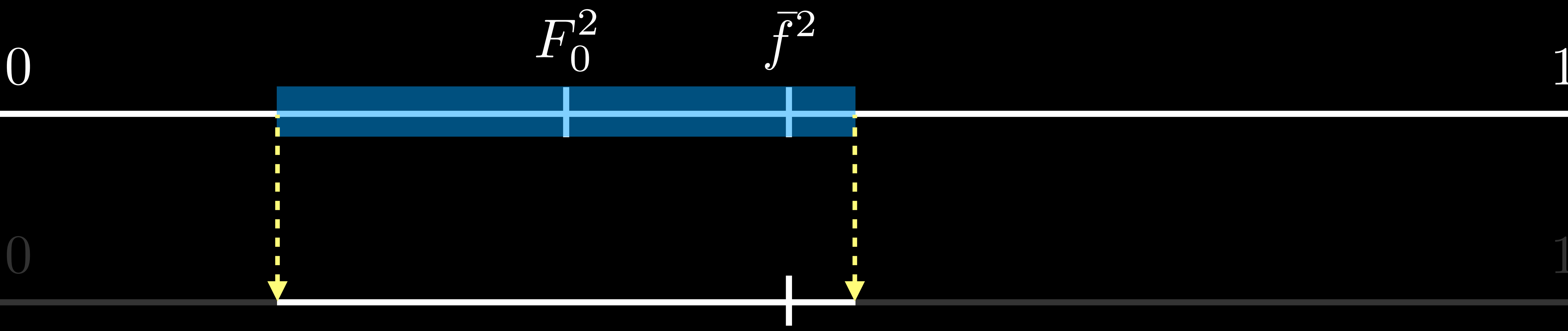
# Overview



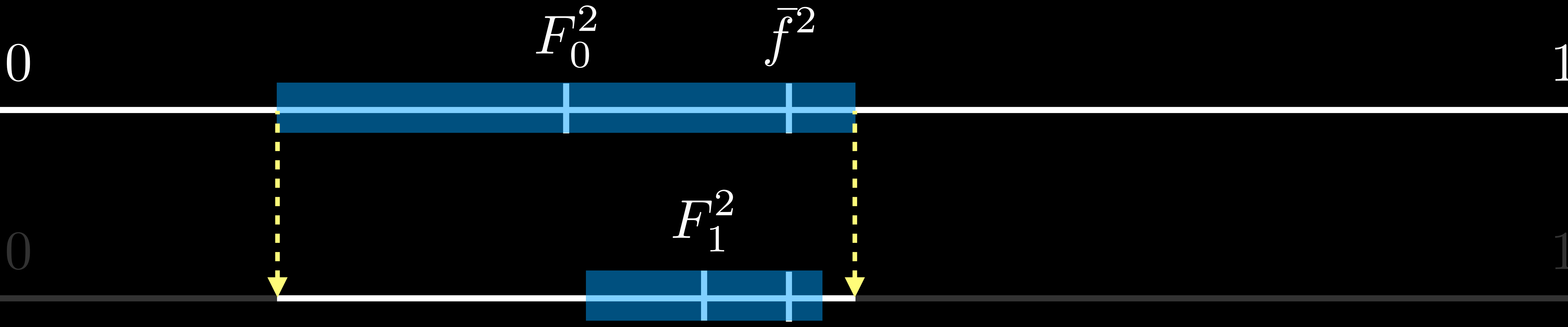
# Overview



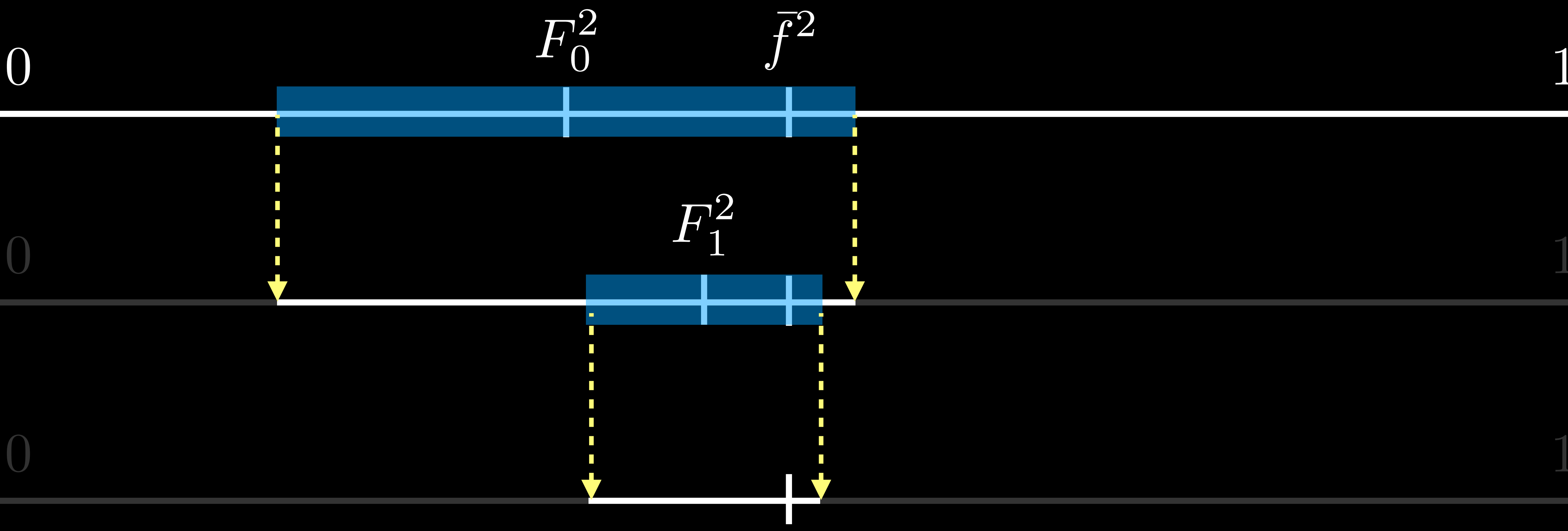
# Overview



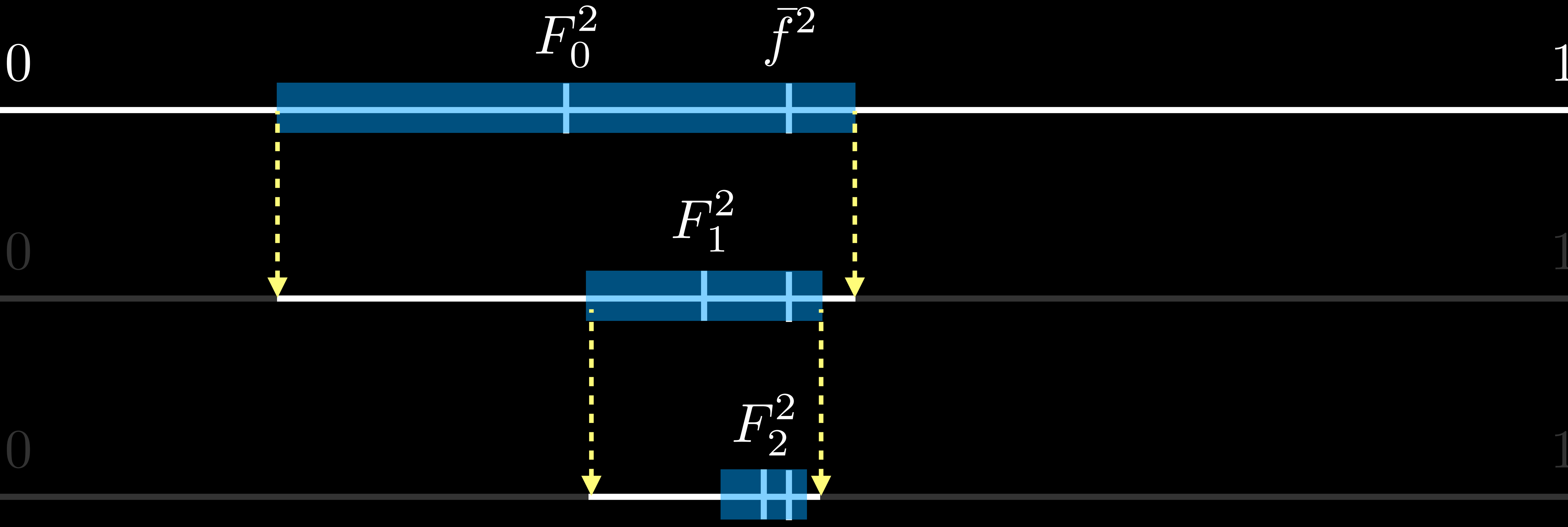
# Overview



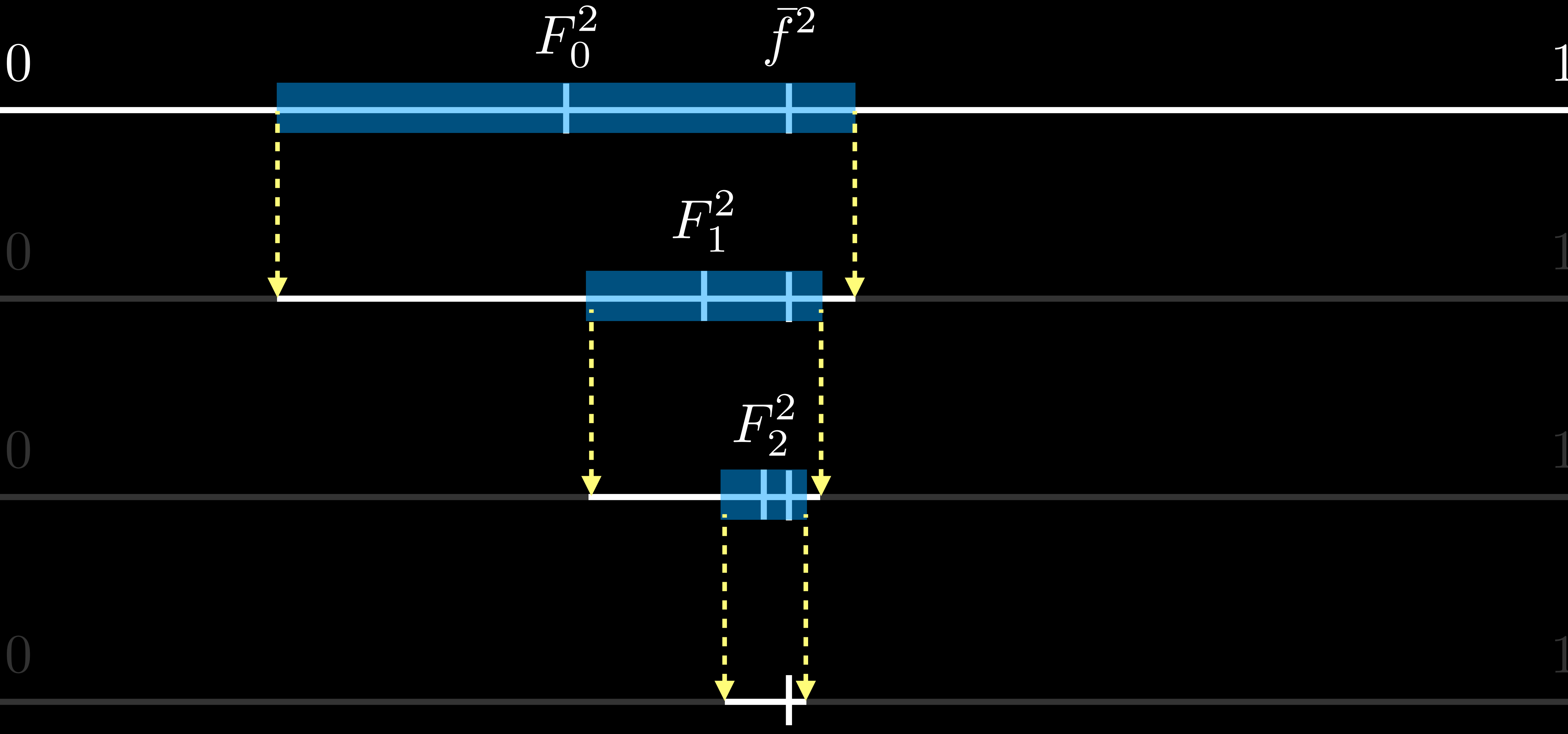
# Overview



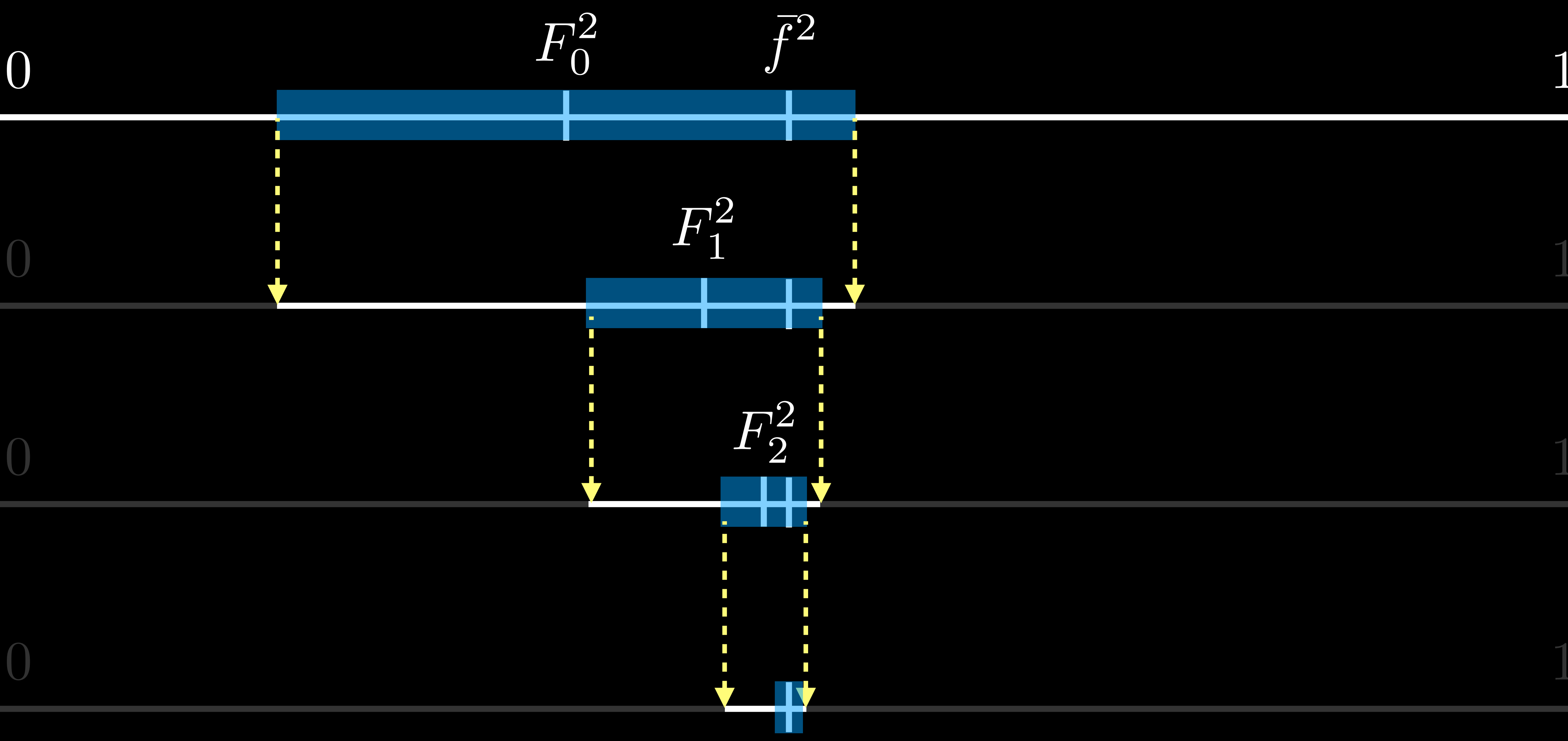
# Overview



# Overview



# Overview



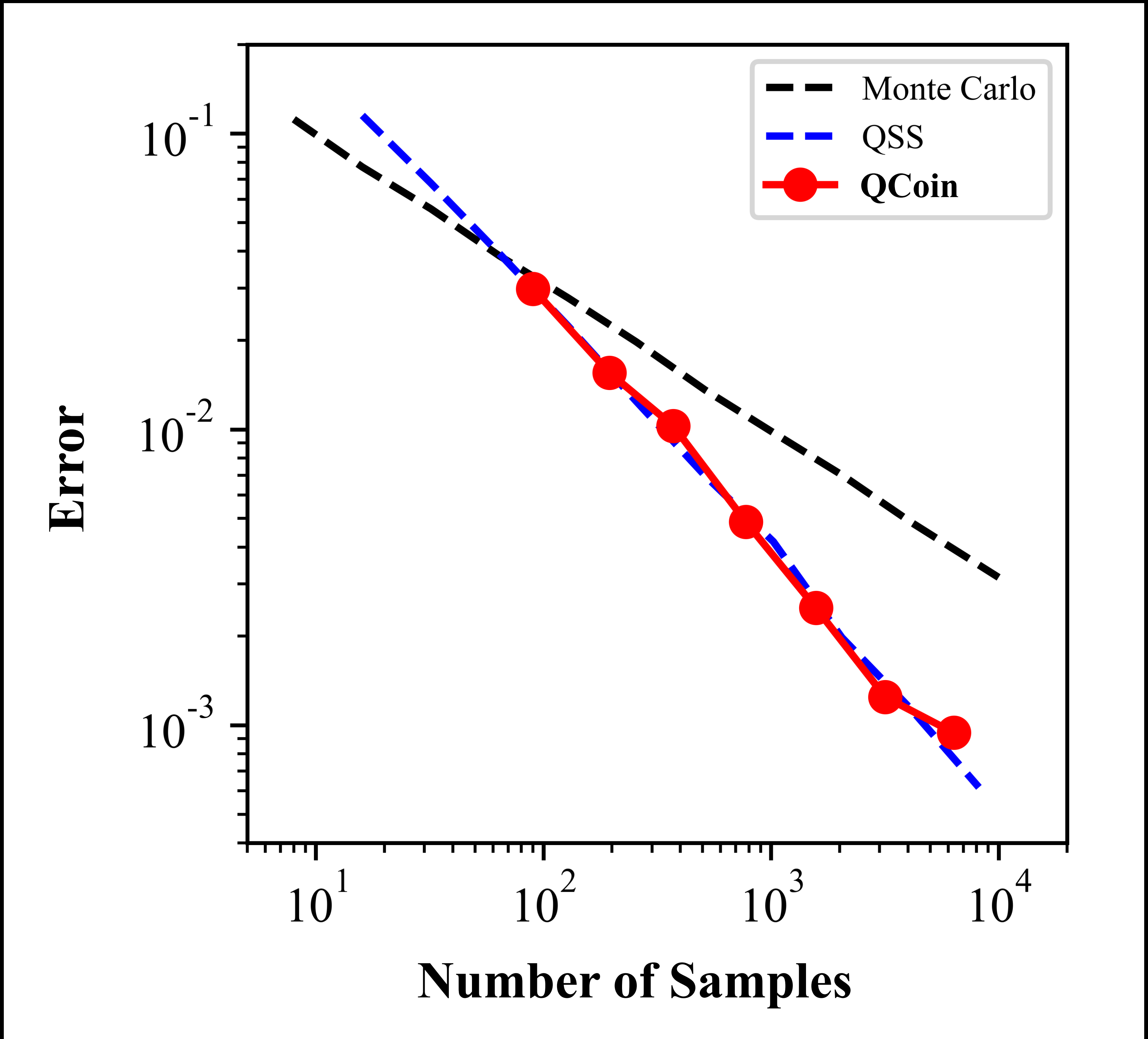
# Features

- Based on the theory by Abrams and Williams in 1999
  - Estimates error intervals by Monte Carlo integration
  - Interval adaptively shrinks toward $\bar{f}^2$
  - Quantum algorithm is used to construct a shifted and scaled quantum coin per iteration
  - Convergence rate is $O(1/N)$ and irrelevant to variance

# Results

# Comparisons

**(simulation)**



# Monte Carlo vs QCoin

**(simulation)**

**Reference**

# Monte Carlo vs QCoin

(simulation)

**Reference**

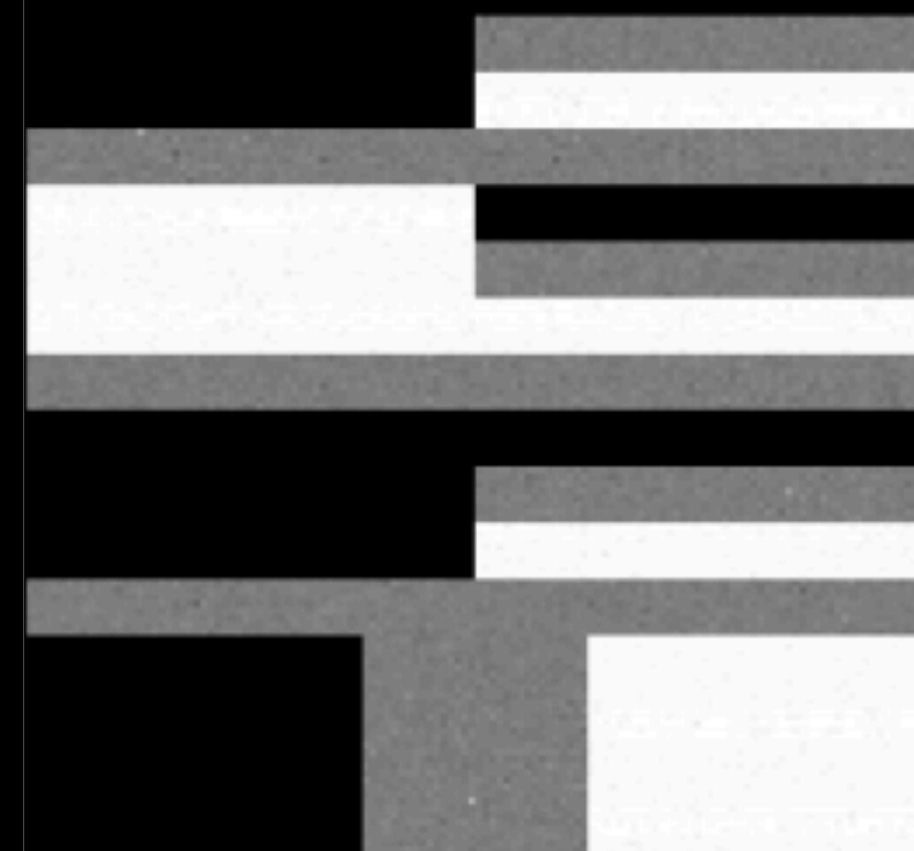
**Monte Carlo**

# Monte Carlo vs QCoin

**(simulation)**

**Reference**

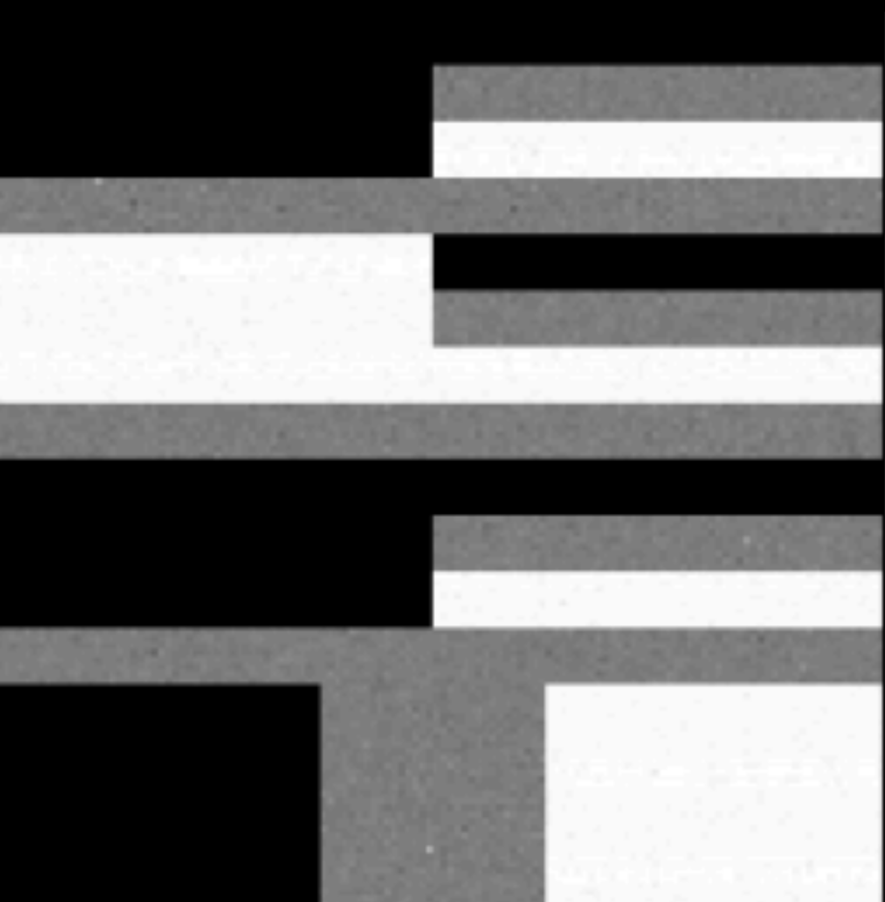
**Monte Carlo**



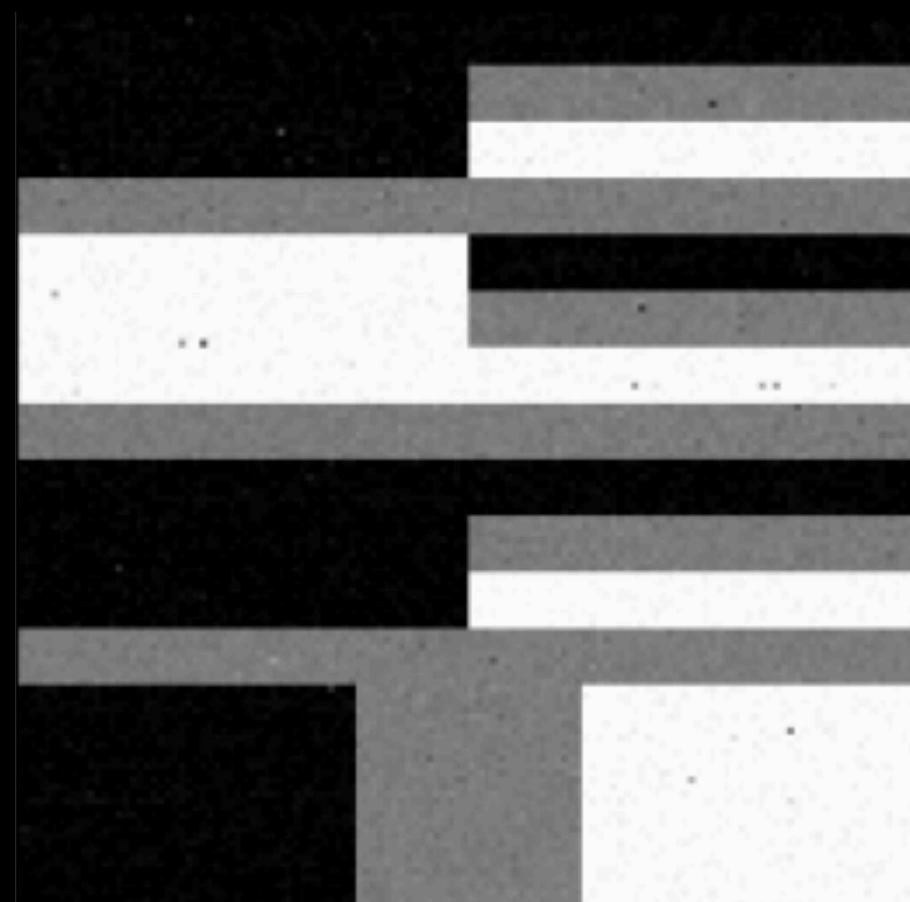
**QCoin (simulation)**

# Real Quantum Computing

(real)



QCoin (simulation)

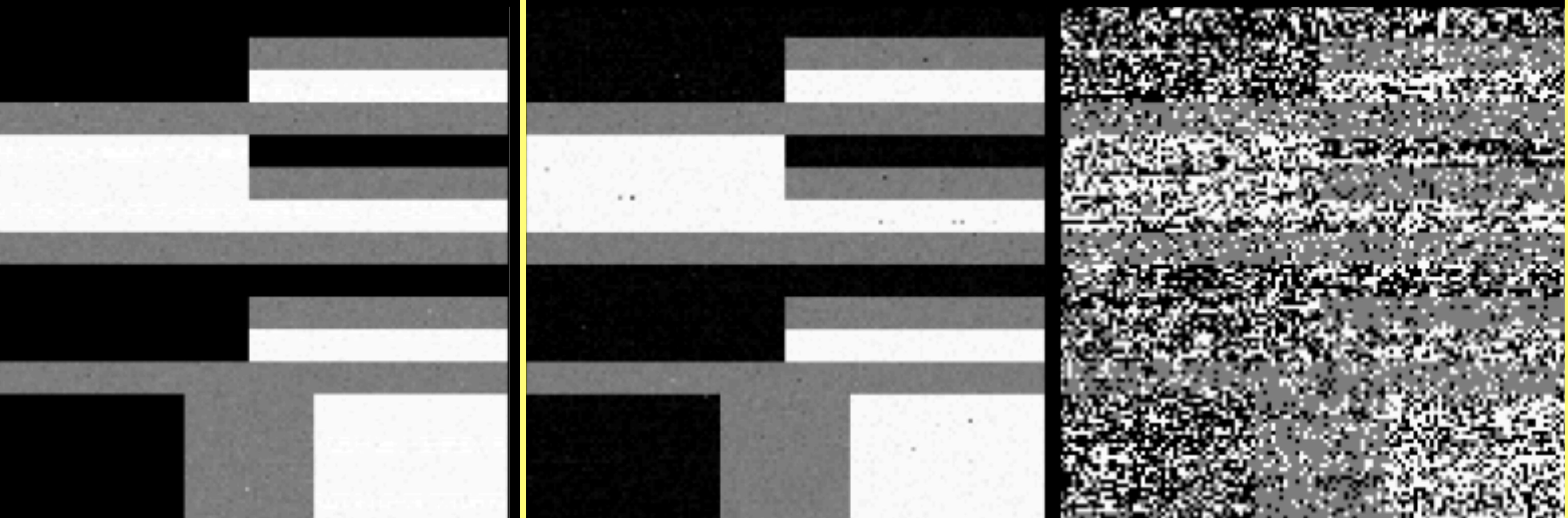


QCoin (real)

QSS (real)

# Real Quantum Computing

(real)



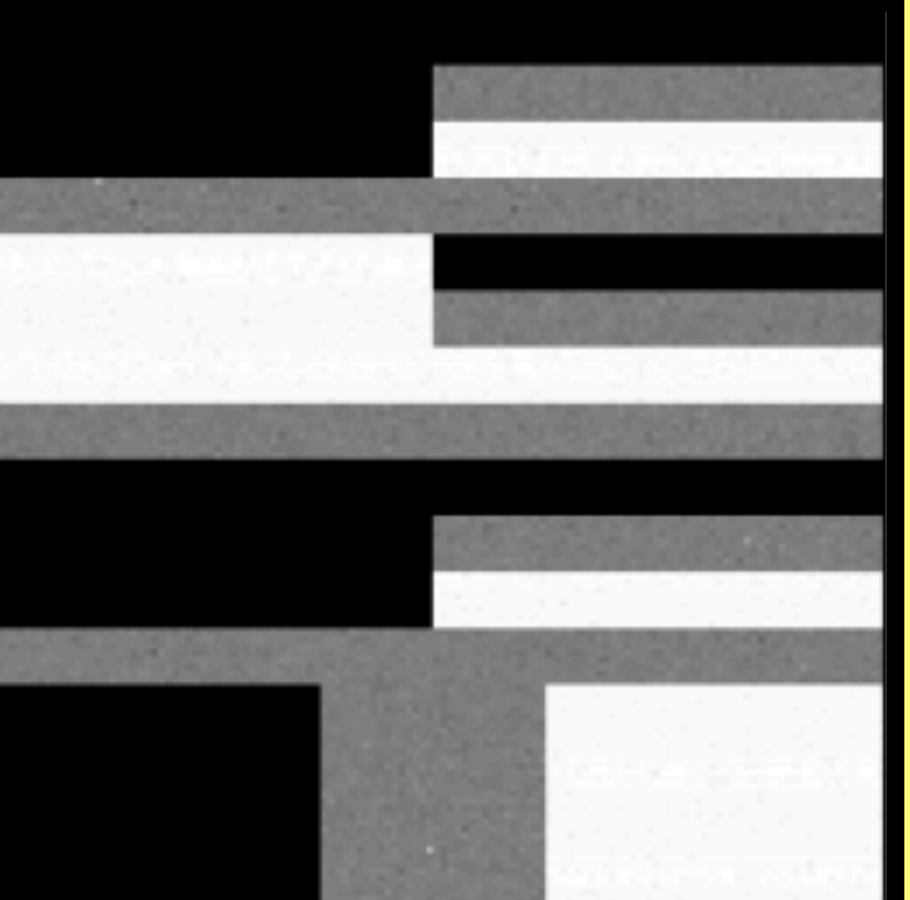
QCoin (simulation)

QCoin (real)

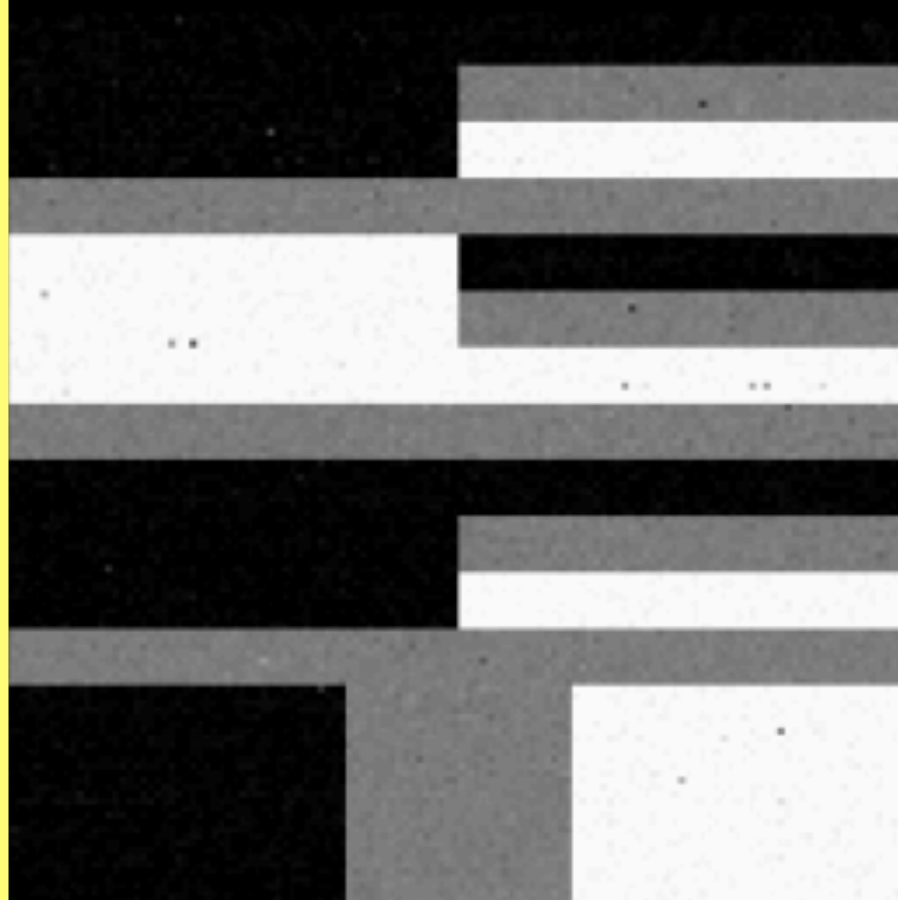
QSS (real)

# Real Quantum Computing

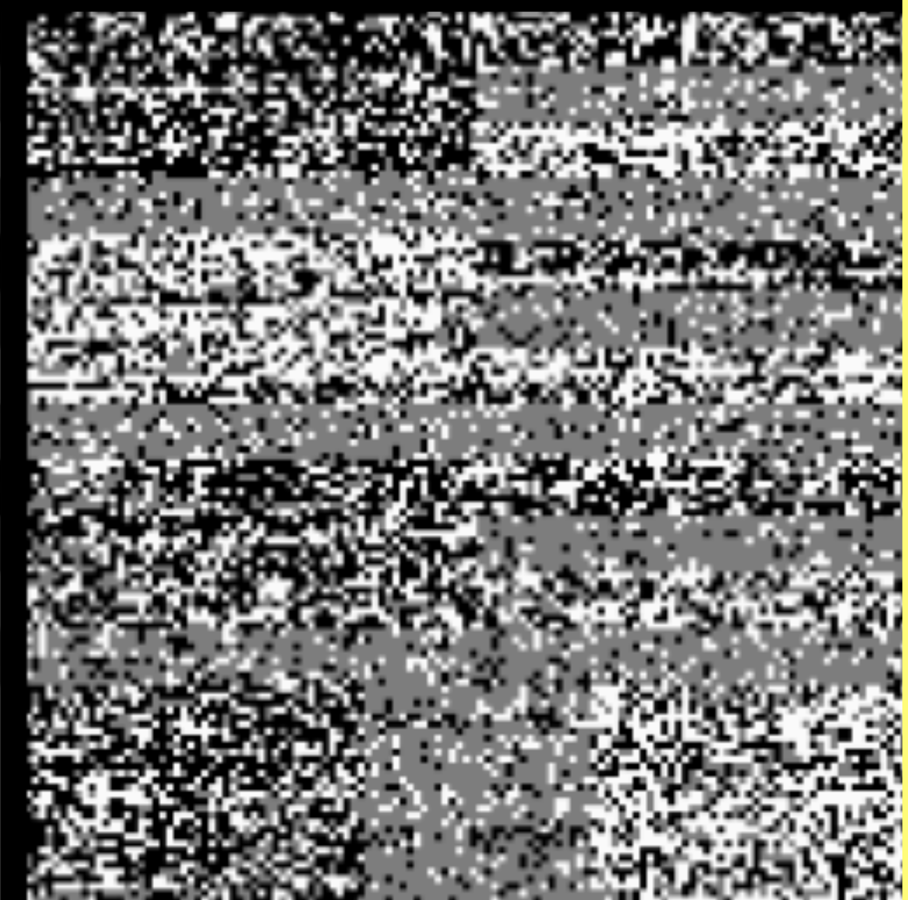
(real)



QCoin (simulation)



QCoin (real)

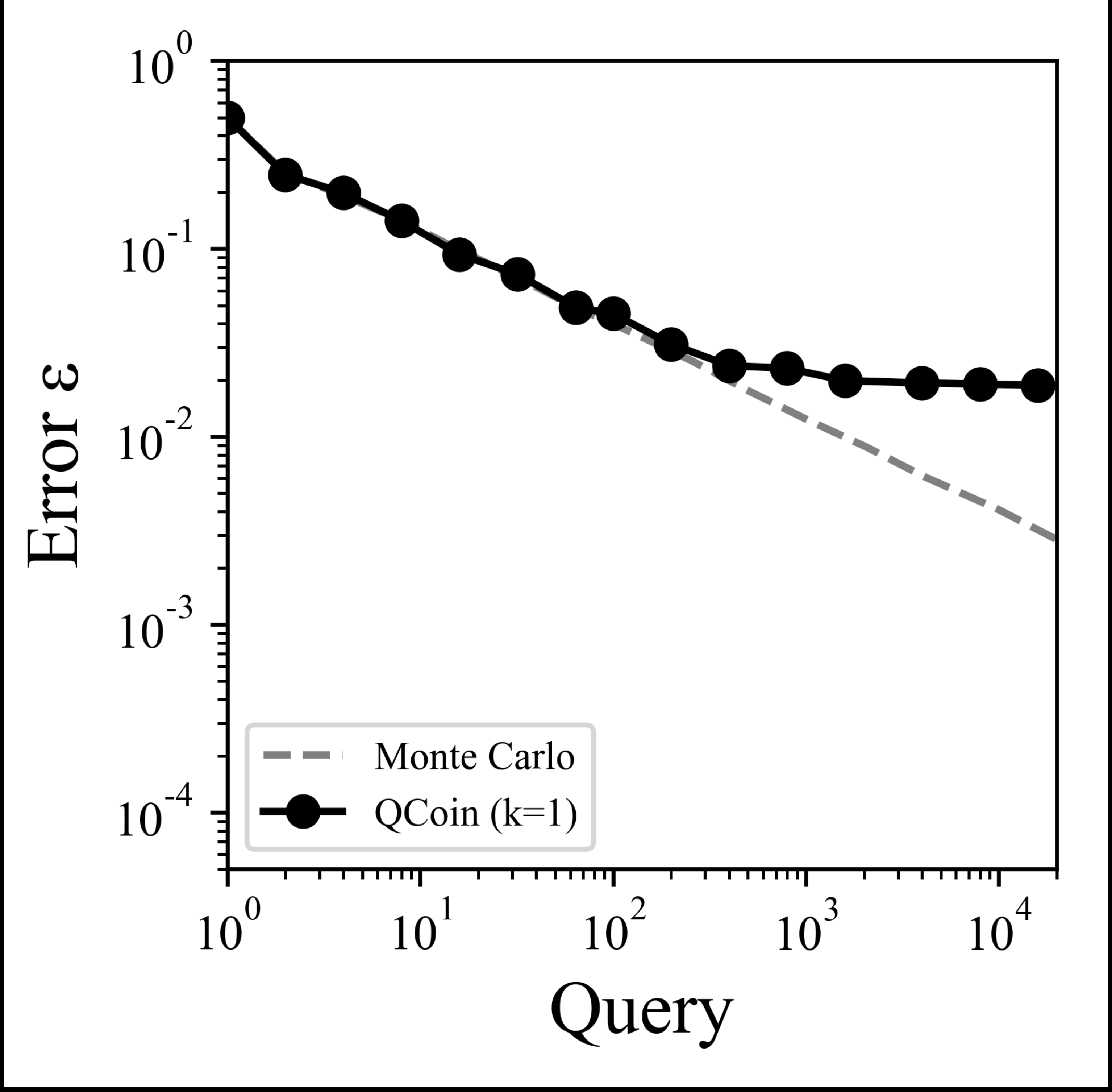


QSS (real)

**QCoin works well both on the simulator and the real computer!**

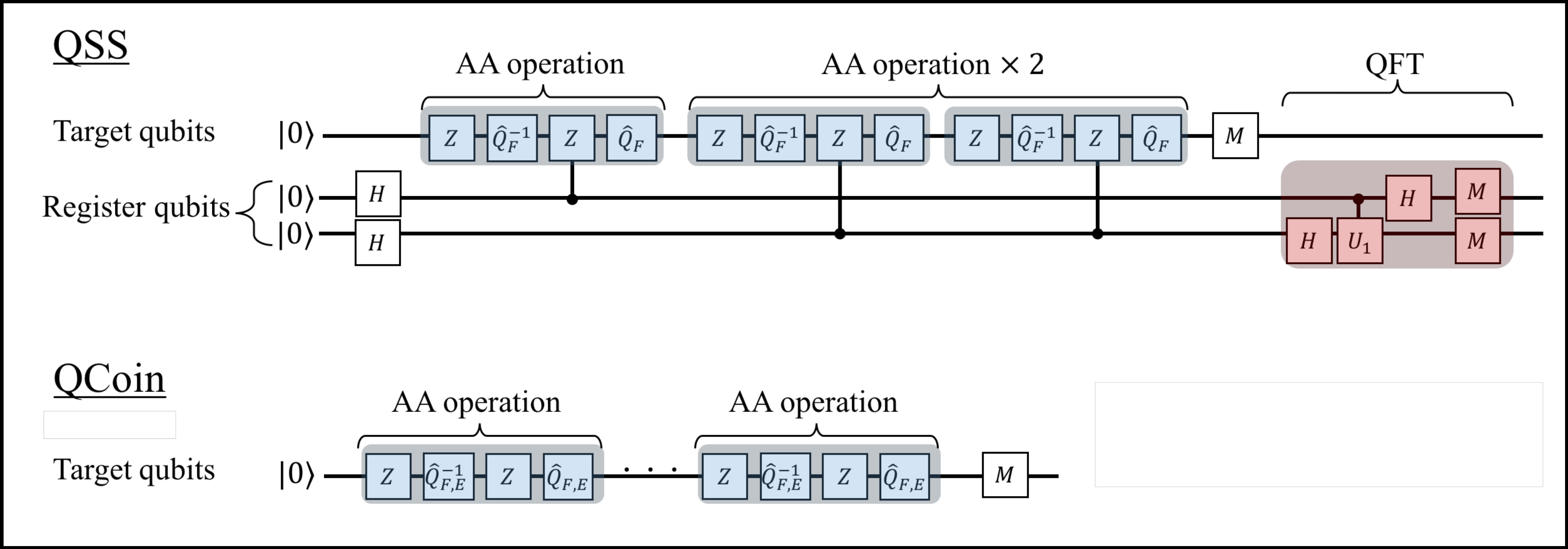
# Degenerated QCoin = MC

**(real)**



# Circuit Complexity

- Hybrid algorithms tend to have simpler quantum circuits
  - Robust against “noisy” computation

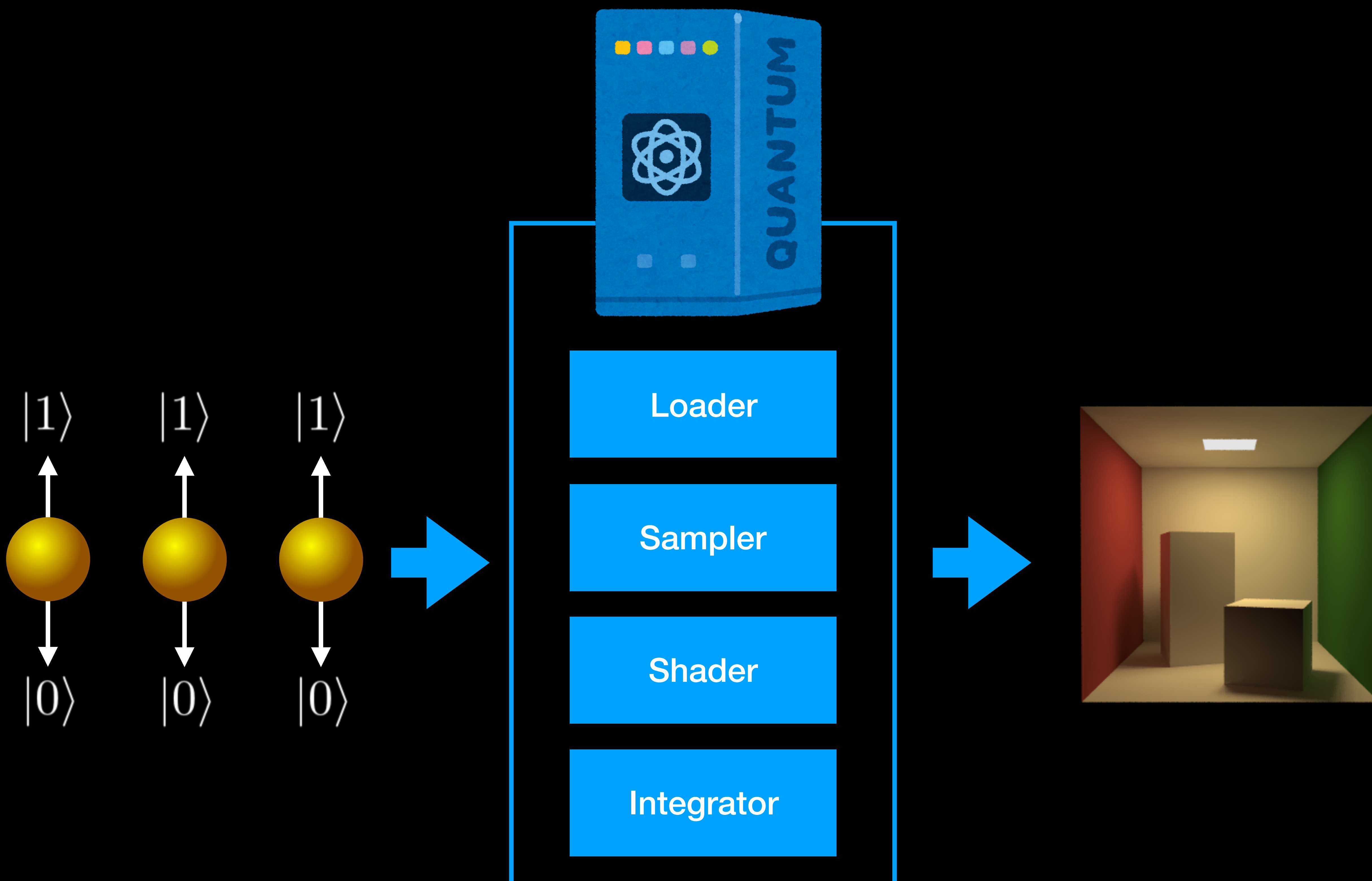


# QCoin = Quantum MC integration

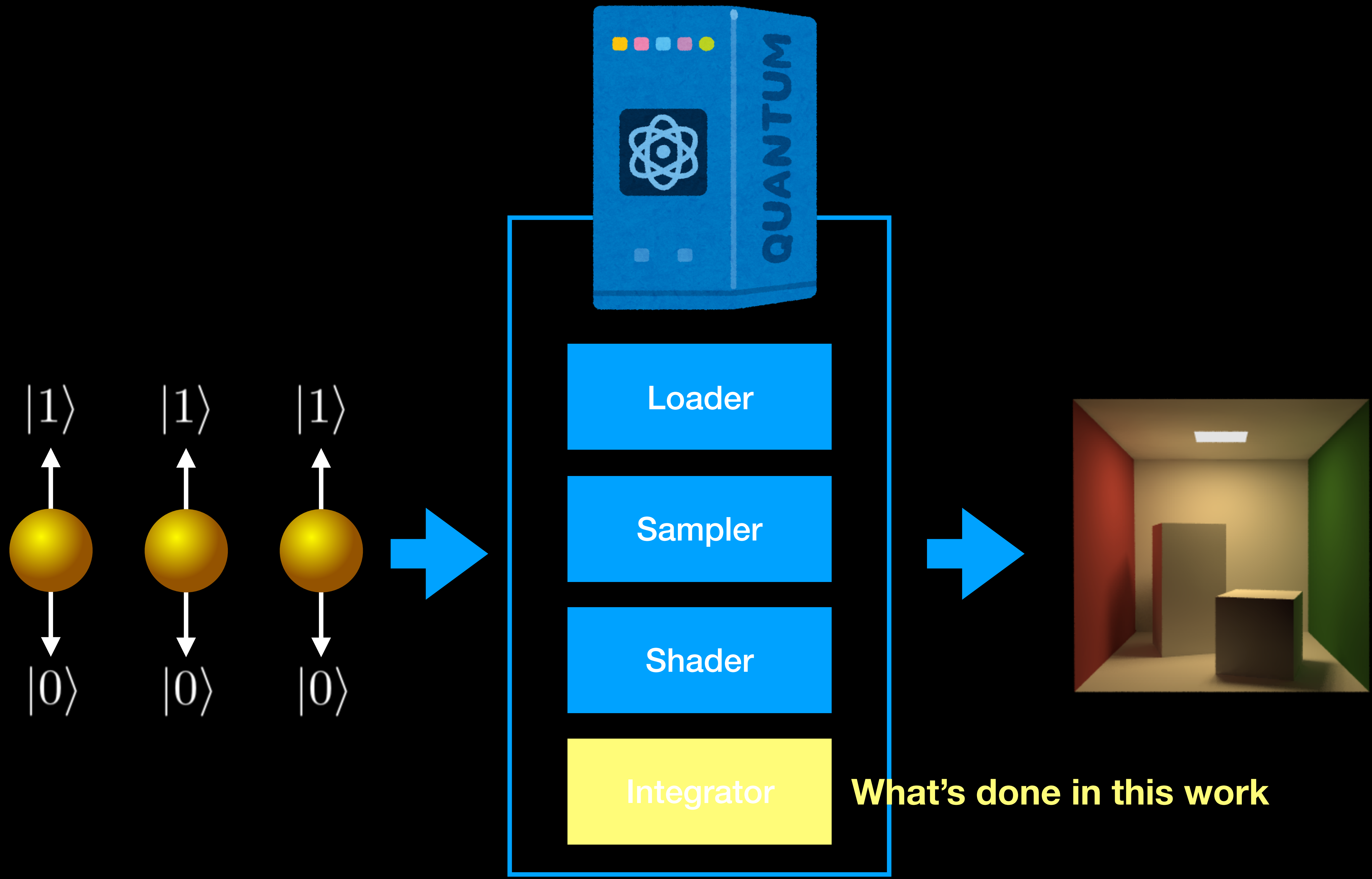
- Asymptotically faster than MC (and as fast as QSS)
- Actually works well on real quantum computers
- Easier to run than QSS
- Errors are similar to MC (denoising is possible)

# Rendering?

# Future Work



# Future Work



|1⟩
|1⟩
|1⟩
|0⟩
|0⟩
|0⟩
QUANTUM
Loader
Sampler
Shader
Integrator
What’s done in this work

# Challenge #1

- The number of qubits is limited!
  - IBMQ has 53 bits
  - Simulation of 50 qubits on a classical computer is already very difficult (consumes 16 peta bytes)

# Challenge #2

- Using amplitudes for computation is not trivial
  - Adding amplitudes is already difficult
  - Using qubits as bits can work, but doesn’t scale
  - How do we perform ray tracing?

# Challenge #3

- No-cloning theorem
  - No variable can be copied
  - Approximate cloning is possible with some cost

# Challenge #4

- Noise in quantum computers
  - In the next 10 years, only noisy quantum computers will be widely available
  - Error correction is being explored, yet not perfect

# Future Work

- We did not specify how to implement $f(x)$
  - In rendering, it is a “ray tracing function”
  - Ray tracing on quantum computers is, in theory, possible [Lanzagorta and Uhlmann 2005], but *actually running* it *efficiently* is a real challenge
- How do we represent data for ray tracing?
  - No-cloning theorem is tricky to address
  - Number of qubits is limited

# Summary

- Quantum computers can already do numerical integration
- Dusted off the theoretical result from 1999 as “QCoin“
  - Hybrid of classical/quantum algorithm
  - Works well on real computers for the first time
  - Asymptotically faster than Monte Carlo integration

**https://arxiv.org/abs/1910.00263**

**or**

**Search “quantum coin integration”**