Quantum Time-Space Tradeoff for Finding Multiple Collision Pairs

Yassine Hamoudi, Frédéric Magniez

IRIF, Université de Paris

TQC 2021

arXiv: 2002.08944

Topic: time-space tradeoff **lower bounds** in the **quantum query model.**

Topic: time-space tradeoff **lower bounds** in the **quantum query model.**

Any algorithm with ≤ S qubits of memory
must use a number T of queries such that…

Very few existing results:

[Klauck et al.'07] Sorting N numbers requires $\mathbf{T^2S \geq \Omega(N^3)}$.

Quantum Time-Space Tradeoffs

Topic: time-space tradeoff **lower bounds** in the **quantum query model.**

Any algorithm with ≤ S qubits of memory must use a number T of queries such that…

Very few existing results:

[Klauck et al.'07] Sorting N numbers requires $\mathbf{T^2S \geq \Omega(N^3)}$.

[Klauck et al.'07] Boolean Matrix-Matrix Multiplication requires $\mathbf{T^2S \geq \Omega(N^5)}$.

[Klauck et al.'07] Boolean Matrix-Vector Multiplication requires $\mathbf{T^2S \geq \Omega(N^3)}$.

[Ambainis et al.'09] Evaluating $Ax \geq (t,\ldots,t)$ requires $\mathbf{T^2S \geq \Omega(tN^3)}$ when $S < N/t$.

$\mathbf{TS \geq \Omega(N^2)}$ when $S > N/t$.

Topic: time-space tradeoff **lower bounds** in the **quantum query model.**

Any algorithm with $\leq S$ qubits of memory must use a number T of queries such that…

Very few existing results:

[Klauck et al.'07] Sorting N numbers requires $\mathbf{T^2S \geq \Omega(N^3)}$.

[Klauck et al.'07] Boolean Matrix-Matrix Multiplication requires $\mathbf{T^2S \geq \Omega(N^5)}$.

[Klauck et al.'07] Boolean Matrix-Vector Multiplication requires $\mathbf{T^2S \geq \Omega(N^3)}$.

[Ambainis et al.'09] Evaluating $Ax \geq (t,\ldots,t)$ requires $\mathbf{T^2S \geq \Omega(tN^3)}$ when $S < N/t$.

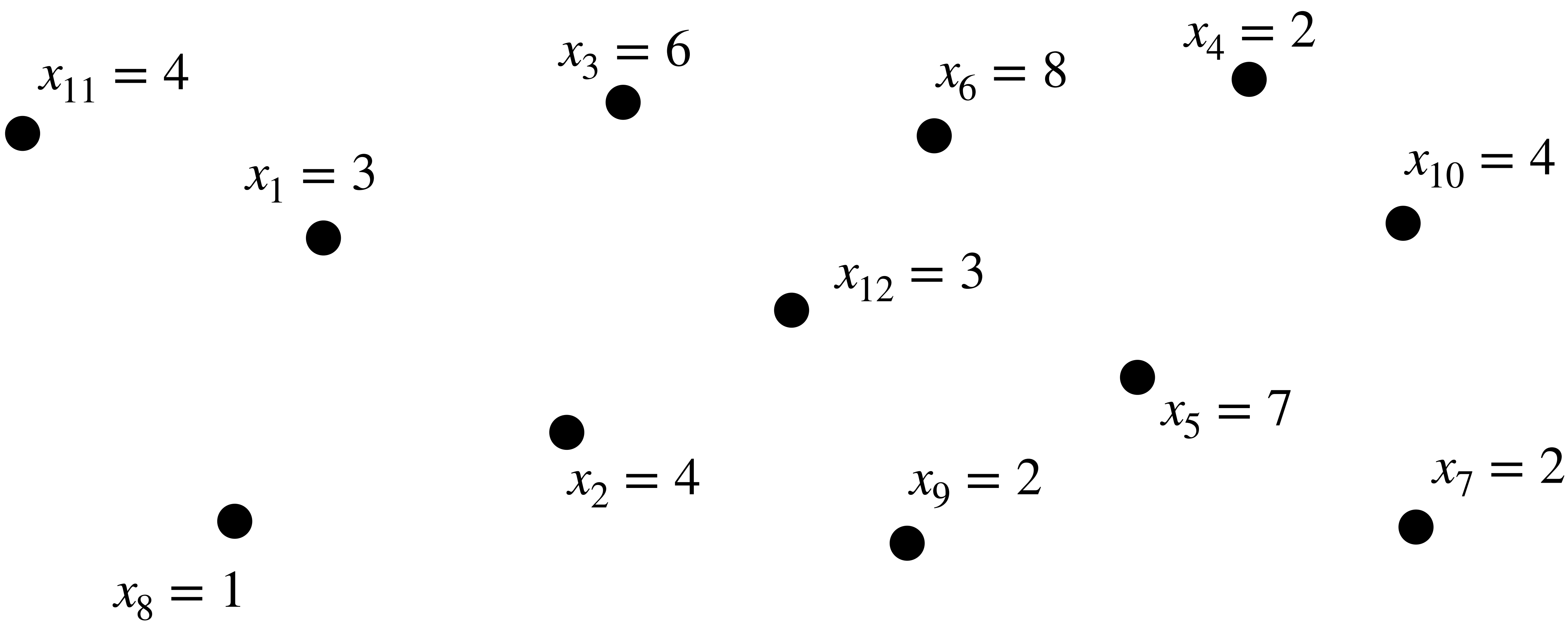
$\mathbf{TS \geq \Omega(N^2)}$ when $S > N/t$.

Our contribution: a new tradeoff for the **Collision Pairs Finding** problem.

1

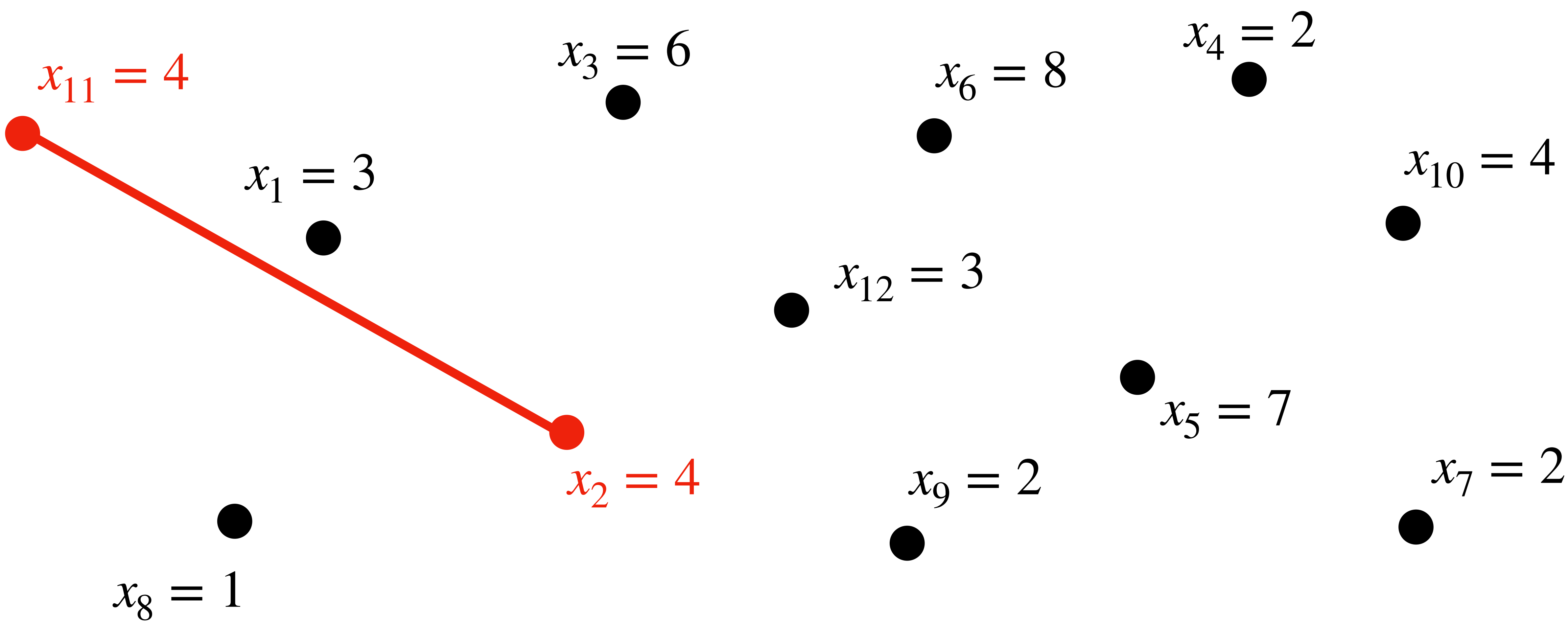
The Collision Pairs Finding Problem

Collision Pairs



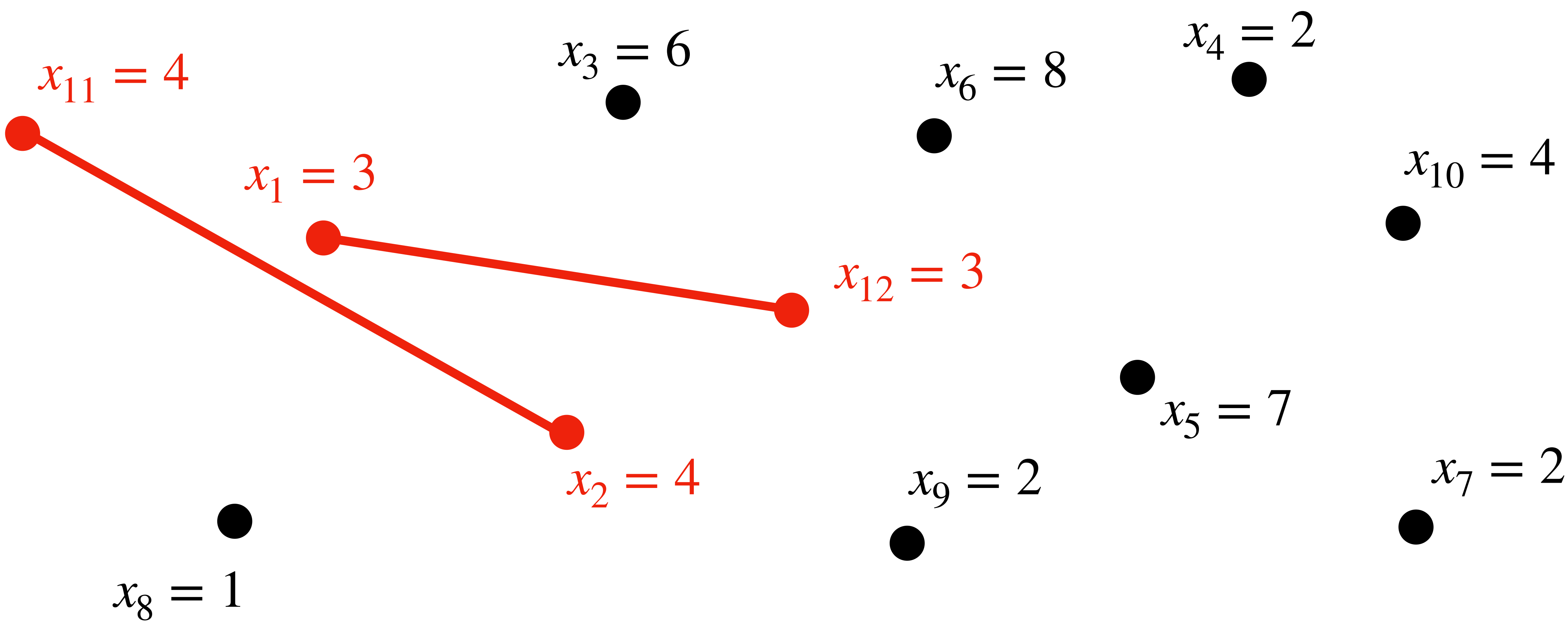
Collision pair: $x_i = x_j$

Collision Pairs



Collision pair: $x_i = x_j$

Collision Pairs



Collision pair: $x_i = x_j$

K-Collision Pairs

Find K collision pairs in a **random** input $x_1, \ldots, x_N$ ~ [N].

K-Collision Pairs

Find K collision pairs in a **random** input $x_1, \ldots, x_N$ ~ [N].

→ **A random input contains ~ Θ(N) collision pairs with high probability.**

K-Collision Pairs

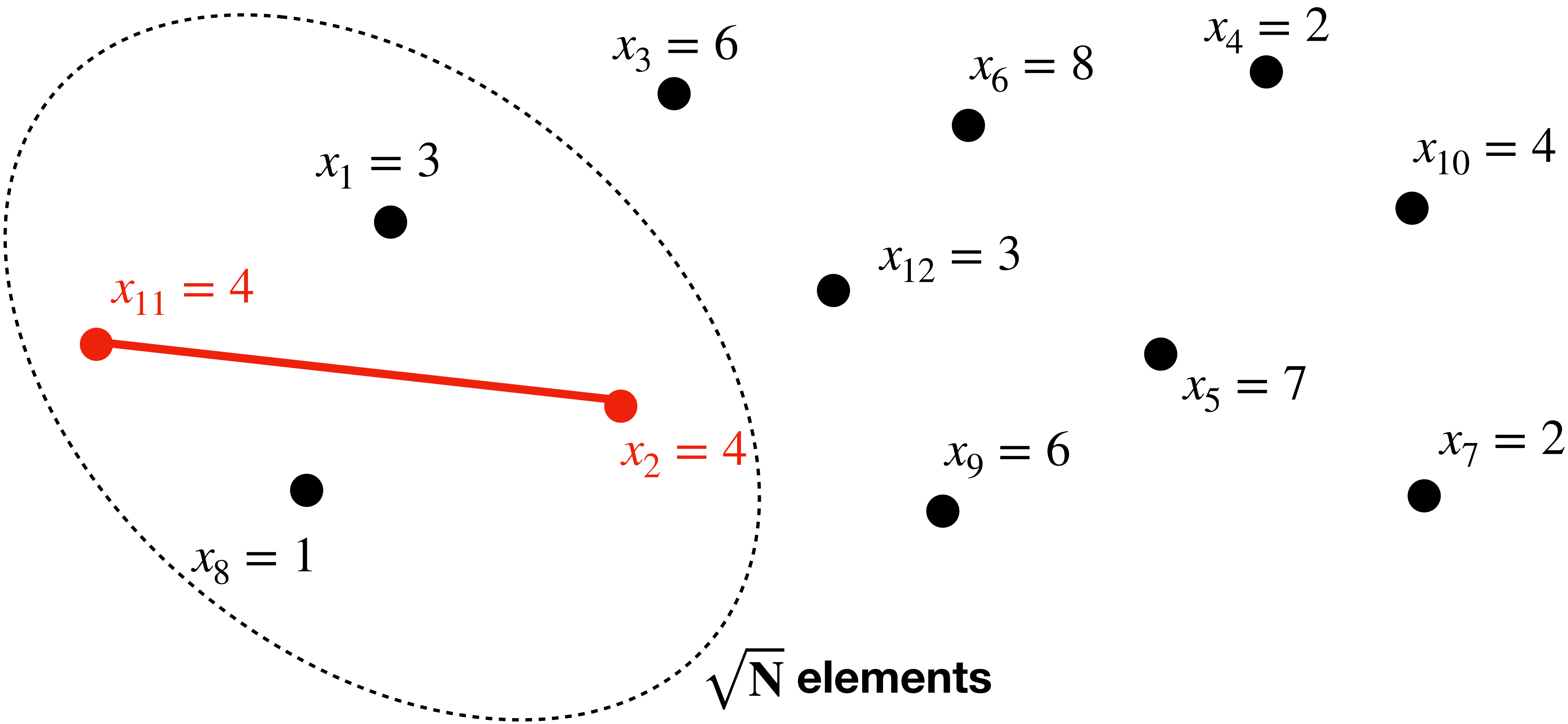
Find K collision pairs in a **random** input $x_1, \ldots, x_N$ ~ [N].

→ **A random input contains ~ Θ(N) collision pairs with high probability.**

→ **Finding collisions is an important problem in cryptanalysis:**

- preimage attacks on hash functions
- meet-in-the-middle attacks ← **requires to find many collisions**
- computing discrete logarithms
- …

Birthday attack (K = 1)

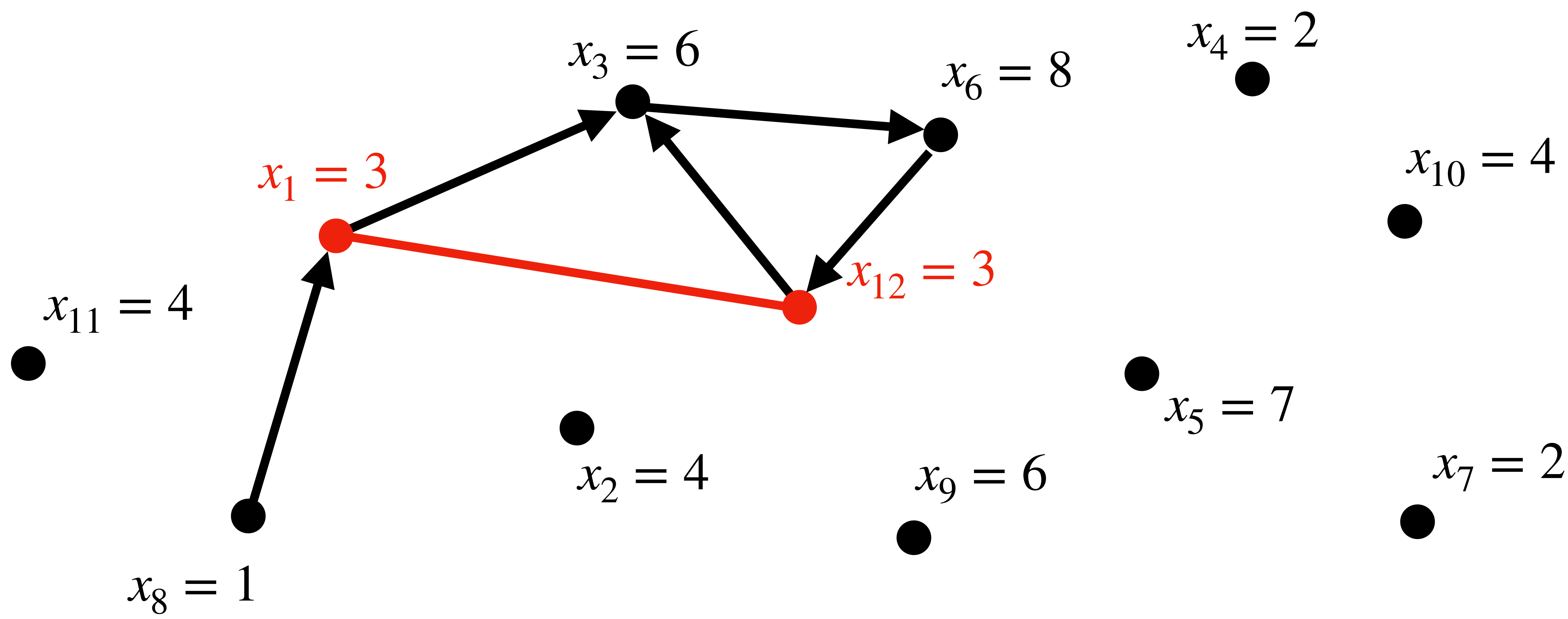


Birthday attack

$$T = O(\sqrt{N})$$

$$S = O(\sqrt{N})$$

Birthday attack (K = 1)



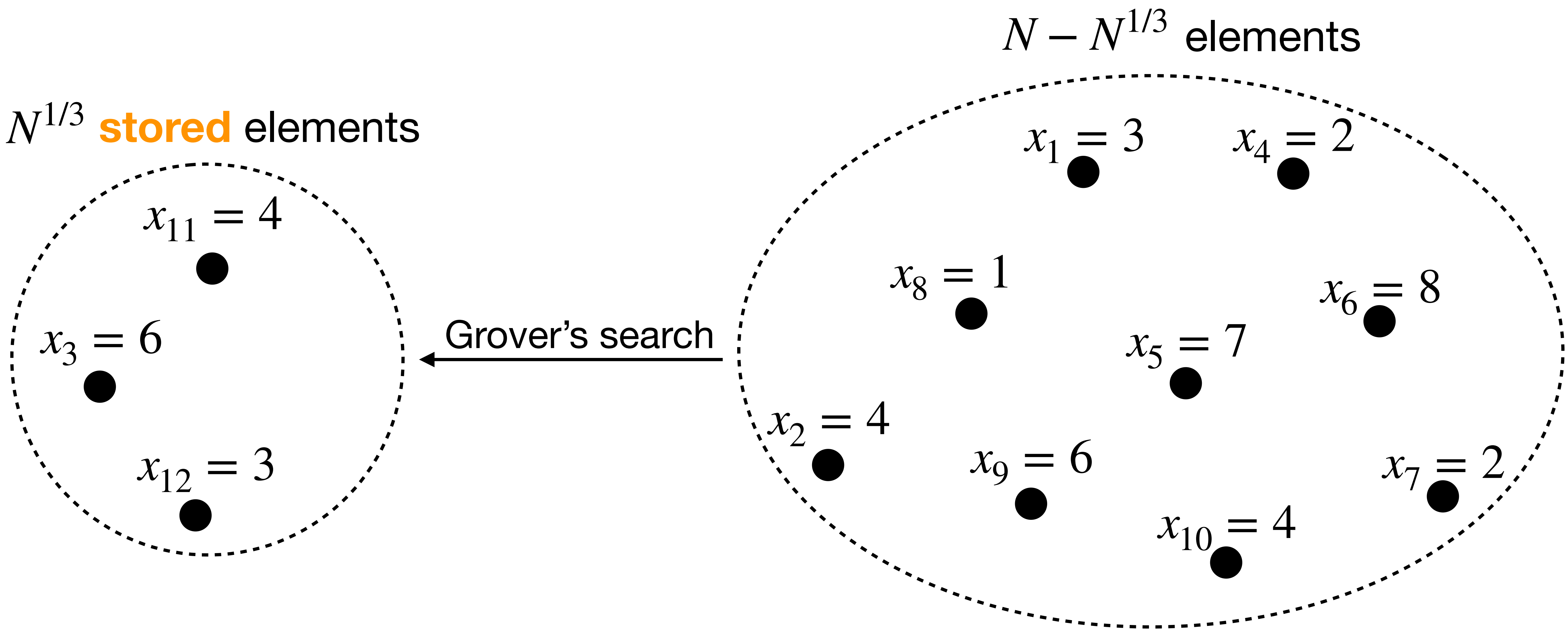
Birthday attack

$$T = O\left(\sqrt{N}\right)$$
$$S = O\left(\sqrt{N}\right)$$

Birthday attack + Floyd's cycle finding

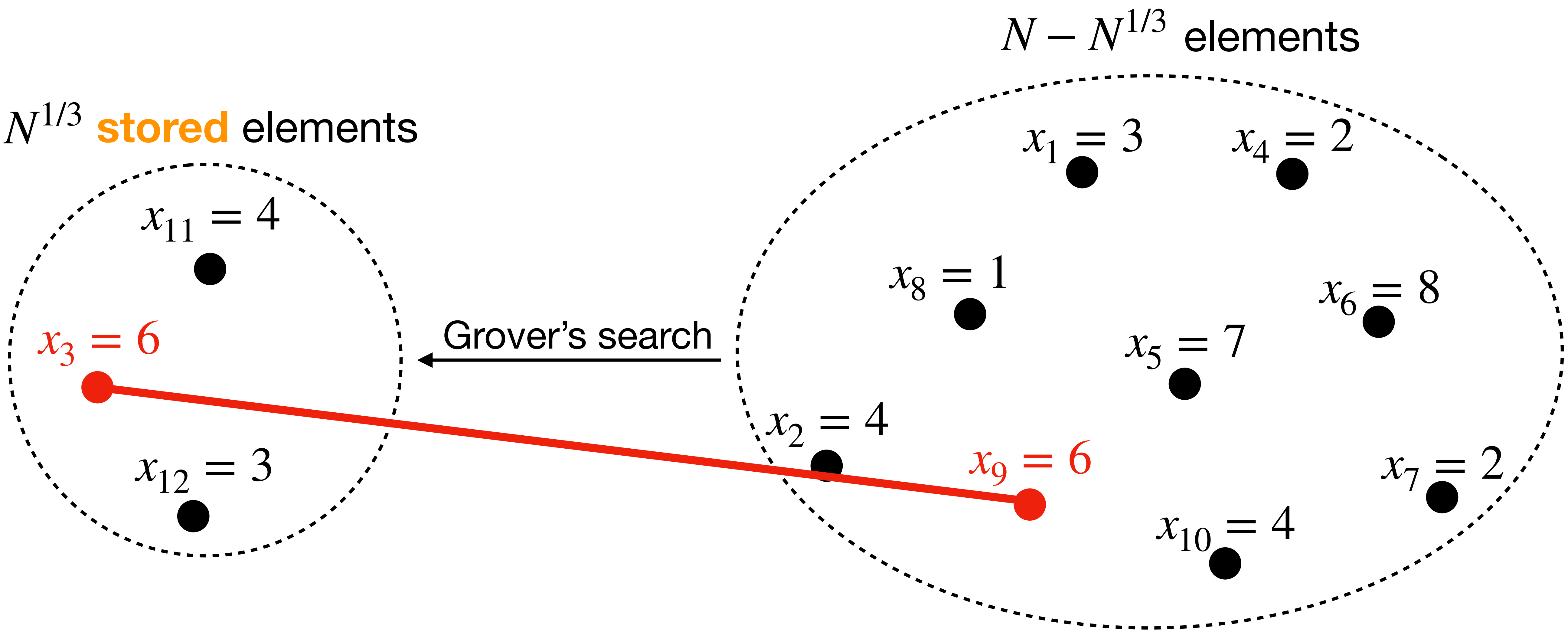
$$T = O\left(\sqrt{N}\right)$$
$$S = O(\log N)$$

Quantum BHT algorithm (K = 1)



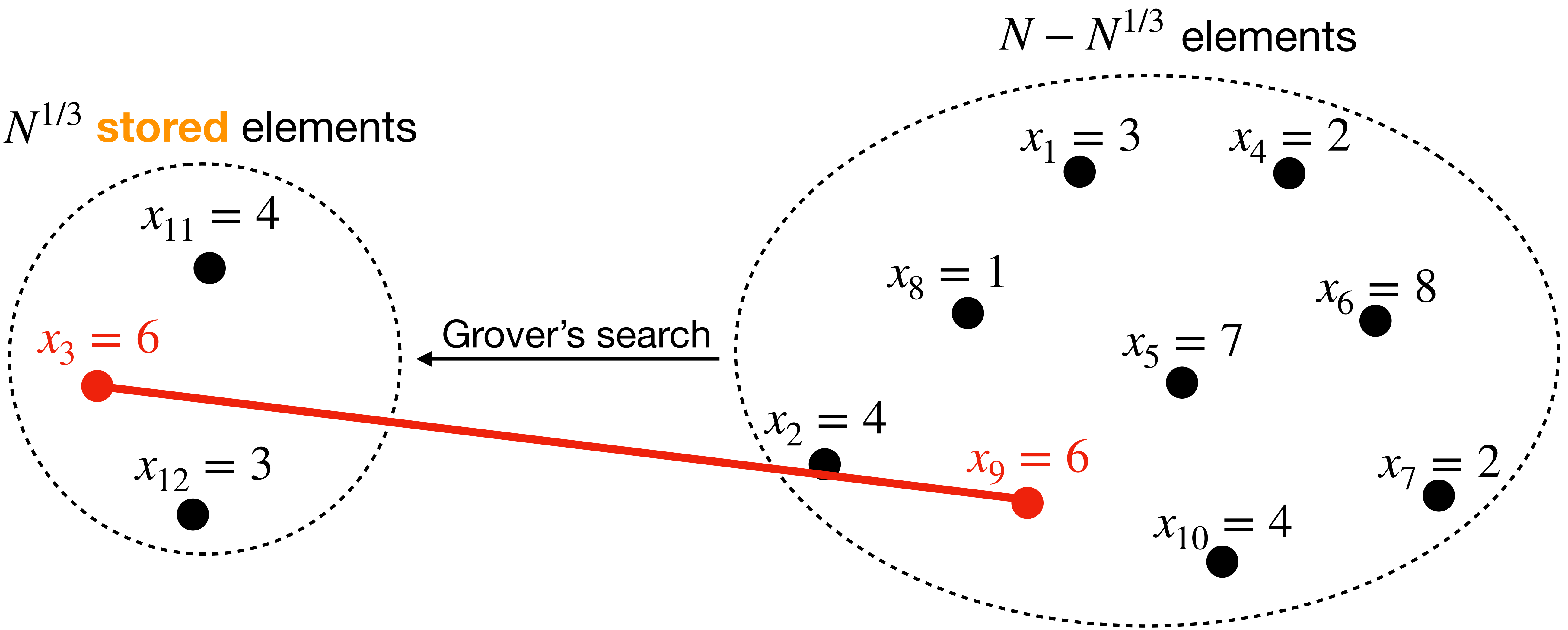
Quantum BHT algorithm

Quantum BHT algorithm (K = 1)



Quantum BHT algorithm

Quantum BHT algorithm (K = 1)

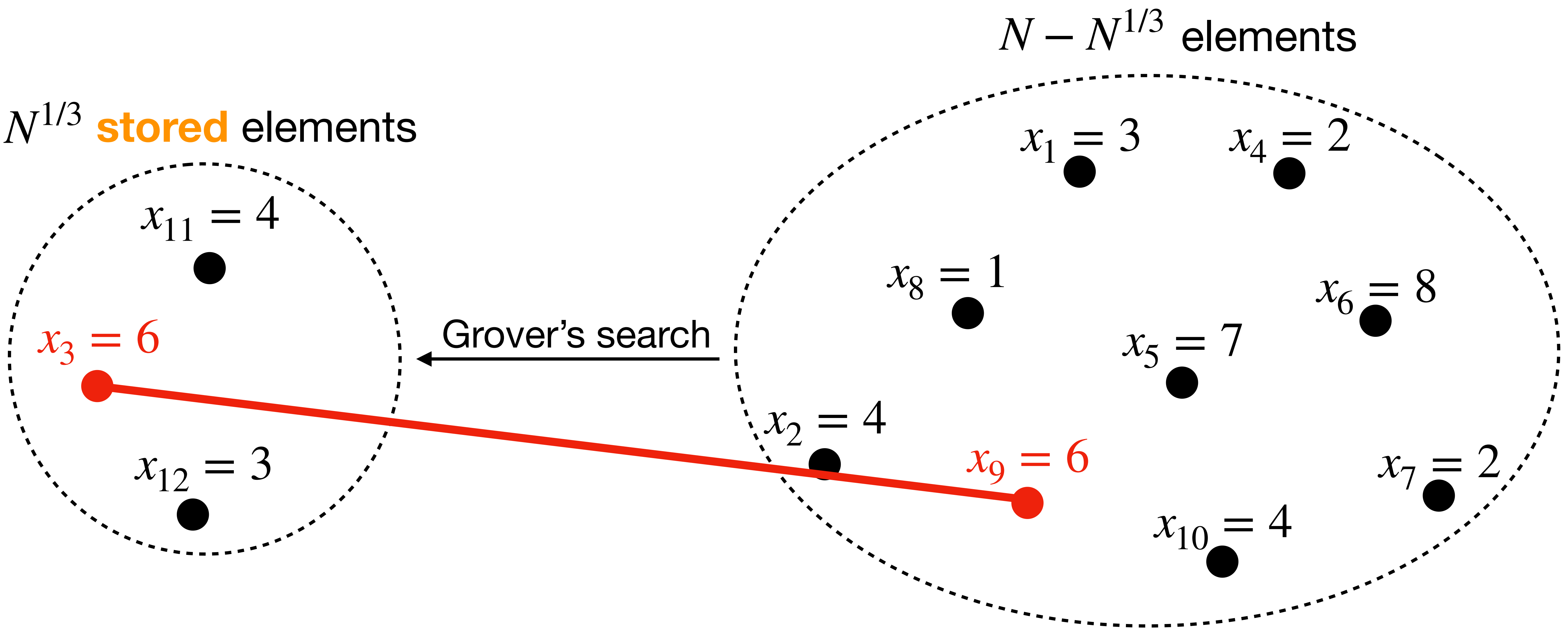


Quantum BHT algorithm

$$T = O\left(N^{1/3}\right)$$

$$S = O\left(N^{1/3}\right)$$

Quantum BHT algorithm (K = 1)



Quantum BHT algorithm

$$T = O\left(N^{1/3}\right)$$

$$S = O\left(N^{1/3}\right)$$

Open problem:

Is there a quantum algorithm with $T \leq o\left(\sqrt{N}\right)$ and $S = O(\log N)$?

Finding K collision pairs

	Classical Tradeoff
Upper bound	$T^2S \leq \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99]

Finding K collision pairs

	Classical Tradeoff	Quantum Tradeoff
Upper bound	**$T^2S \leq \tilde{O}(K^2N)$** when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99]	**$T^2S \leq \tilde{O}(K^2N)$** when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K^{2/3}N^{1/3})$ Adaptation of the BHT algorithm

Finding K collision pairs

	Classical Tradeoff	**Quantum Tradeoff**
Upper bound	$T^2S \leq \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99] → $T = \tilde{O}(K^{1/2}N^{1/2})$ at best	$T^2S \leq \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K^{2/3}N^{1/3})$ Adaptation of the BHT algorithm → $T = \tilde{O}(K^{2/3}N^{1/3})$ at best

Finding K collision pairs

	Classical Tradeoff	Quantum Tradeoff
Upper bound	$T^2S \le \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \le S \le \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99] $\rightarrow$ $T = \tilde{O}(K^{1/2}N^{1/2})$ at best	$T^2S \le \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \le S \le \tilde{O}(K^{2/3}N^{1/3})$ Adaptation of the BHT algorithm $\rightarrow$ $T = \tilde{O}(K^{2/3}N^{1/3})$ at best
Lower bound	$T^2S \ge \tilde{\Omega}(K^2N)$ [Chakrabarti,Chen'17] [Dinur'20]	

Finding K collision pairs

	Classical Tradeoff	**Quantum Tradeoff**
Upper bound	$T^2S \le \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \le S \le \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99] → $T = \tilde{O}(K^{1/2}N^{1/2})$ at best	$T^2S \le \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \le S \le \tilde{O}(K^{2/3}N^{1/3})$ Adaptation of the BHT algorithm → $T = \tilde{O}(K^{2/3}N^{1/3})$ at best
Lower bound	$T^2S \ge \tilde{\Omega}(K^2N)$ [Chakrabarti,Chen'17] [Dinur'20]	$T^3S \ge \Omega(K^3N)$ Our result

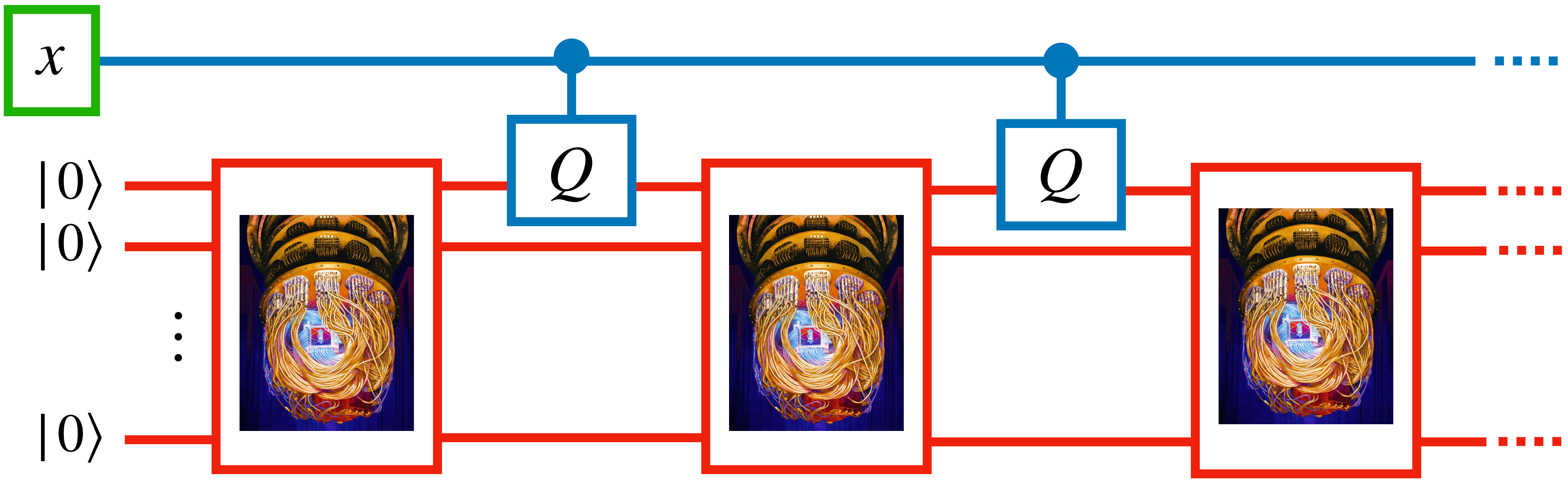
Finding K collision pairs

	Classical Tradeoff	**Quantum Tradeoff**
Upper bound	$T^2S \leq \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K)$ Parallel Collision Search [van Oorschot and Wiener'99] → $T = \tilde{O}(K^{1/2}N^{1/2})$ at best	$T^2S \leq \tilde{O}(K^2N)$ when $\tilde{\Omega}(\log N) \leq S \leq \tilde{O}(K^{2/3}N^{1/3})$ Adaptation of the BHT algorithm → $T = \tilde{O}(K^{2/3}N^{1/3})$ at best
Lower bound	$T^2S \geq \tilde{\Omega}(K^2N)$ [Chakrabarti,Chen'17] [Dinur'20]	$T^3S \geq \Omega(K^3N)$ Our result → $T \geq \tilde{\Omega}(KN^{1/3})$ when $S = \log(N)$

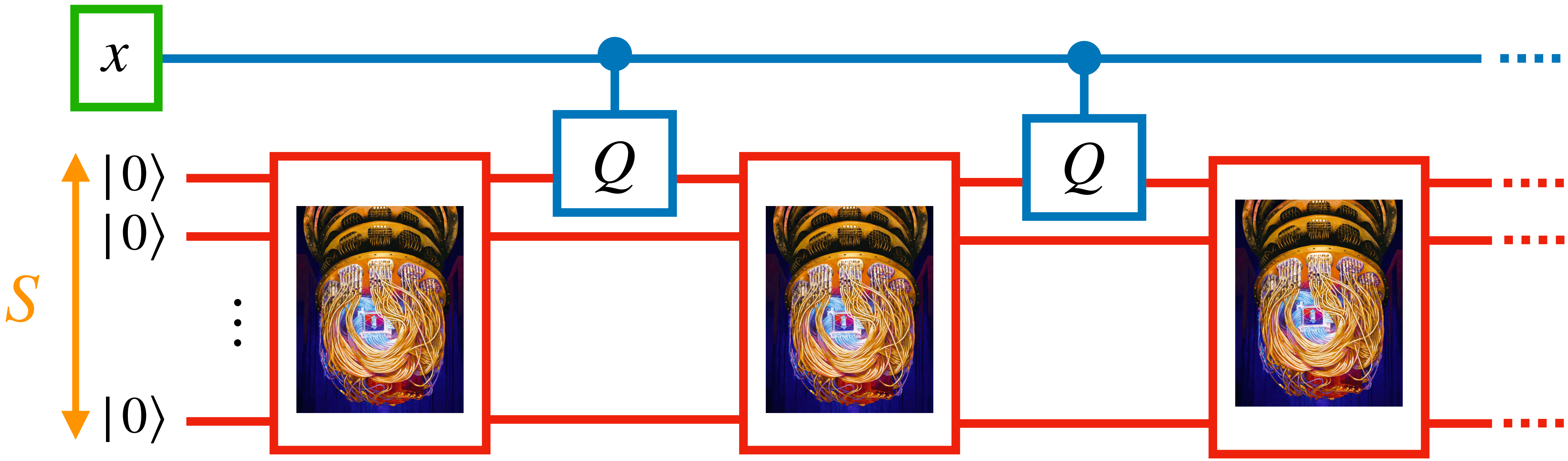
3

Lower Bound Method

Quantum Query Model

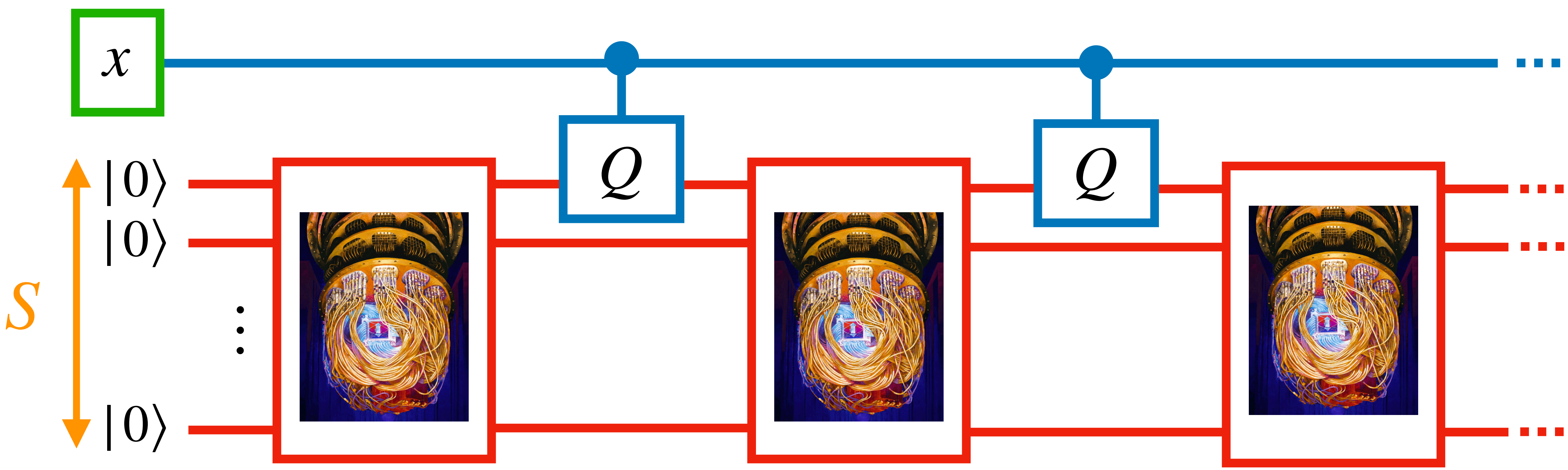


Quantum Query Model



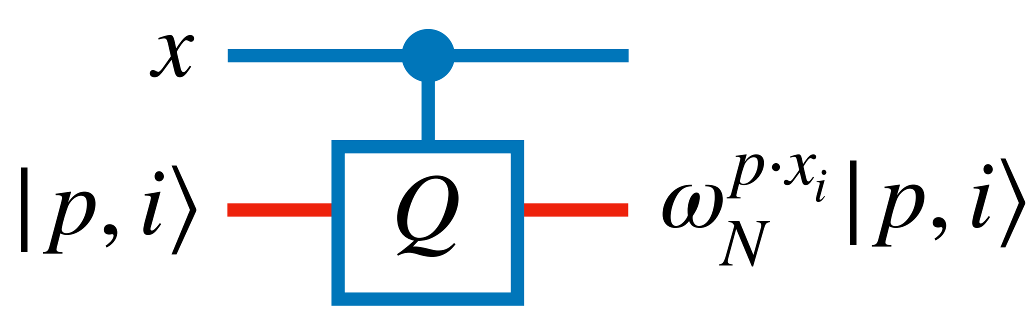
- The memory is made of **S qubits**, initially set to $|0\rangle$.

Quantum Query Model

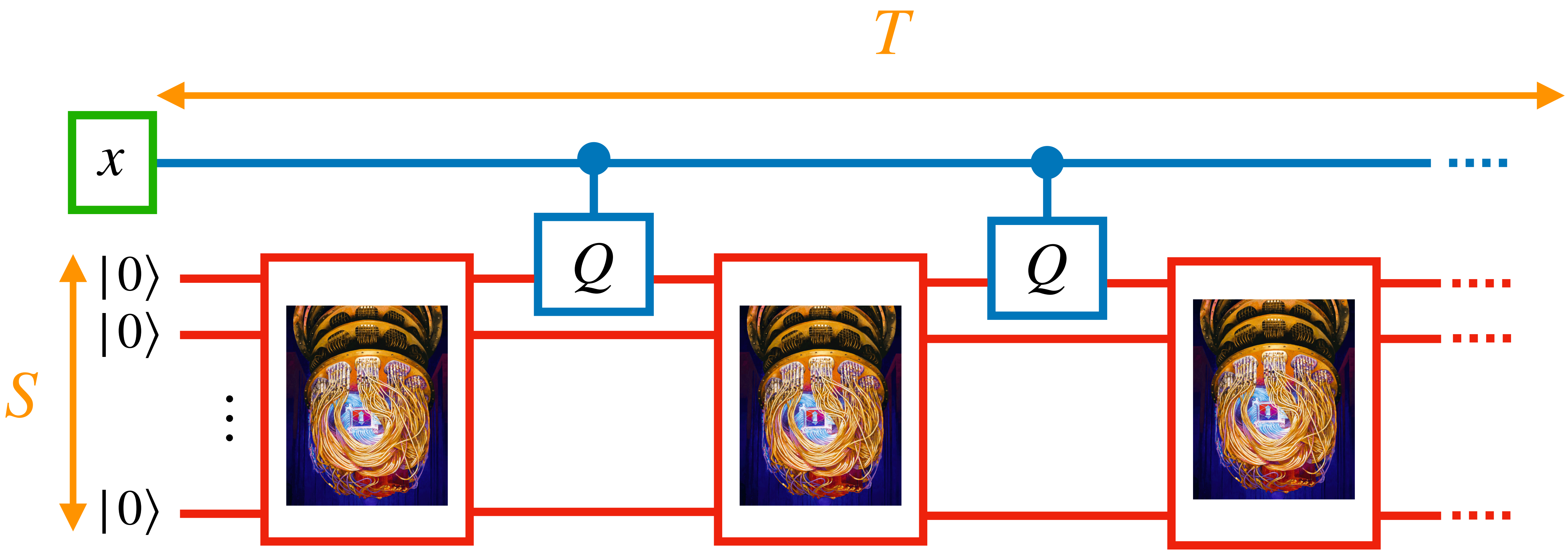


- The memory is made of **S qubits**, initially set to $|0\rangle$.

- The quantum "Query Operator" Q is:
  (when $x_i \in [N]$)

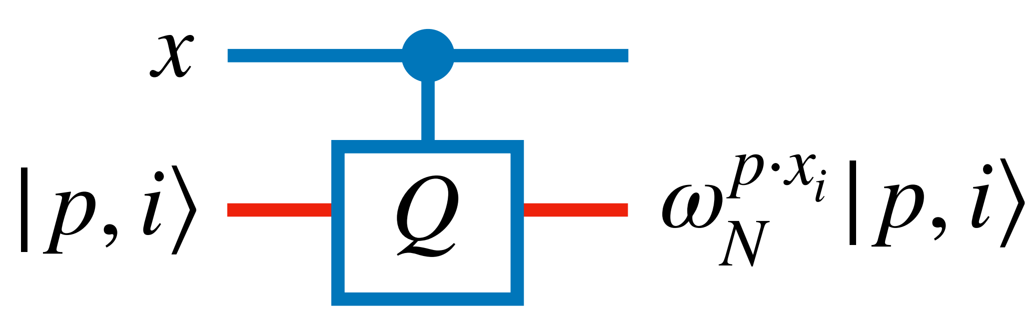


Quantum Query Model



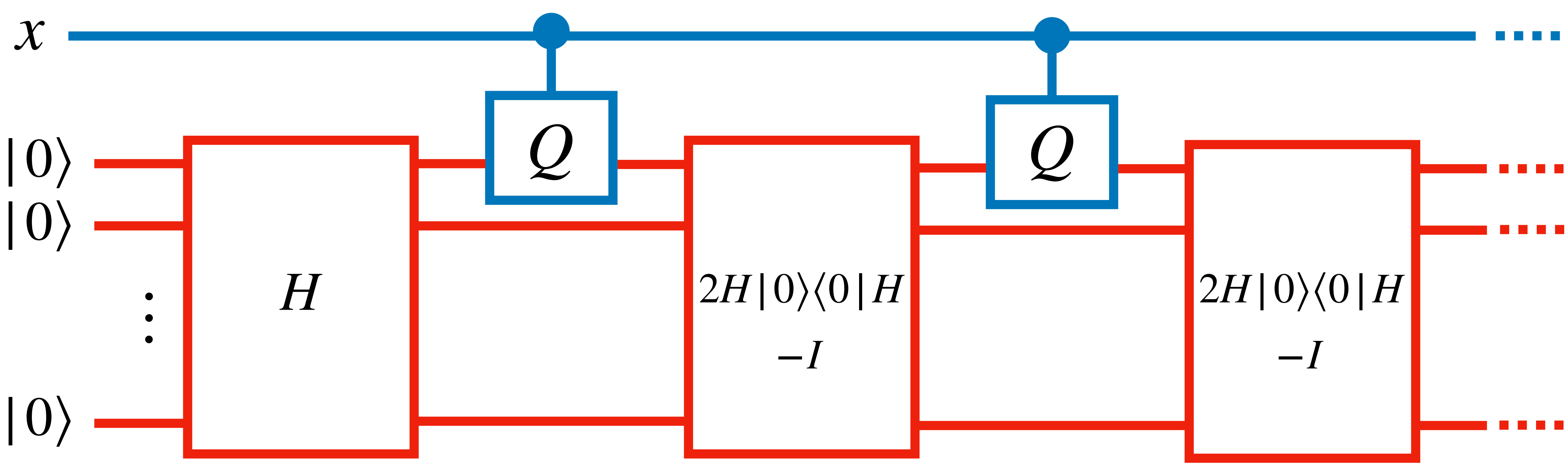
- The memory is made of **S qubits**, initially set to $|0\rangle$.

- The quantum "Query Operator" Q is:
  (when $x_i \in [N]$)



- The computation alternates between **T quantum queries** and **T quantum operations** on the memory.

Quantum Query Model



Example: Grover's Search

$$\begin{cases} T = O(\sqrt{N}) \\ S = O(\log N) \end{cases}$$

Time-Space Lower Bounds from Time Lower Bounds

- A method to deduce Time-Space lower bounds from **Time lower bounds**.
- Introduced by [Borodin et al.'81], and by [Klauck'03] for the quantum version.
- Applicable when the problem has a large **output of size K** (≠ decision problem).

Time-Space Lower Bounds from Time Lower Bounds

- A method to deduce Time-Space lower bounds from **Time lower bounds**.
- Introduced by [Borodin et al.'81], and by [Klauck'03] for the quantum version.
- Applicable when the problem has a large **output of size K** (≠ decision problem).

Time lower bound

For all K, it is impossible to compute K parts of the output with success probability $\geq 2^{-O(K)}$ in time $\tau(K)$.

Time-Space Lower Bounds from Time Lower Bounds

- A method to deduce Time-Space lower bounds from **Time lower bounds**.
- Introduced by [Borodin et al.'81], and by [Klauck'03] for the quantum version.
- Applicable when the problem has a large **output of size K** (≠ decision problem).

Time lower bound

For all K, it is impossible to compute K parts of the output with success probability **≥ $2^{-O(K)}$** in time $\boldsymbol{\tau(K)}$.

[Borodin et al.'81]
[Klauck'03]

$$\Rightarrow$$

$$K \geq S$$

Time-Space lower bound

$$T \geq \Omega\left(\tau(S) \cdot \frac{K}{S}\right)$$

Time-Space Lower Bounds from Time Lower Bounds

- A method to deduce Time-Space lower bounds from **Time lower bounds**.
- Introduced by [Borodin et al.'81], and by [Klauck'03] for the quantum version.
- Applicable when the problem has a large **output of size K** (≠ decision problem).

Time lower bound

For all K, it is impossible to compute K parts of the output with success probability $\geq 2^{-O(K)}$ in time $\tau(K)$.

[Borodin et al.'81]
[Klauck'03]

$\Rightarrow$

$K \geq S$

Time-Space lower bound

$$T \geq \Omega\left(\tau(S) \cdot \frac{K}{S}\right)$$

It is impossible to find **K collisions** with success probability $\geq 2^{-O(K)}$ in time $\tau(K) = O\left(K^{2/3}N^{1/3}\right)$.

Time-Space Lower Bounds from Time Lower Bounds

- A method to deduce Time-Space lower bounds from **Time lower bounds**.
- Introduced by [Borodin et al.'81], and by [Klauck'03] for the quantum version.
- Applicable when the problem has a large **output of size K** (≠ decision problem).

Time lower bound

For all K, it is impossible to compute K parts of the output with success probability **≥ 2^-O(K)** in time $\tau(K)$.

[Borodin et al.'81]
[Klauck'03]

$\Longrightarrow$

$K \geq S$

Time-Space lower bound

$$T \geq \Omega\left(\tau(S) \cdot \frac{K}{S}\right)$$

It is impossible to find **K collisions** with success probability ≥ $2^{-O(K)}$ in time $\tau(K) = O\left(K^{2/3} N^{1/3}\right)$.

$\Longrightarrow$

$$T \geq \Omega\left(K N^{1/3} / S^{1/3}\right)$$

Time Lower Bounds

It is impossible to find **K collisions**
with success probability ≥ $2^{-O(K)}$
in time $\tau(K) = O\left(K^{2/3} N^{1/3}\right)$.

+ We don't care about space anymore.
– We must deal with the exponentially small success probability regime.

Time Lower Bounds

It is impossible to find **K collisions** with success probability $\geq 2^{-O(K)}$ in time $\tau(K) = O\left(K^{2/3}N^{1/3}\right)$.

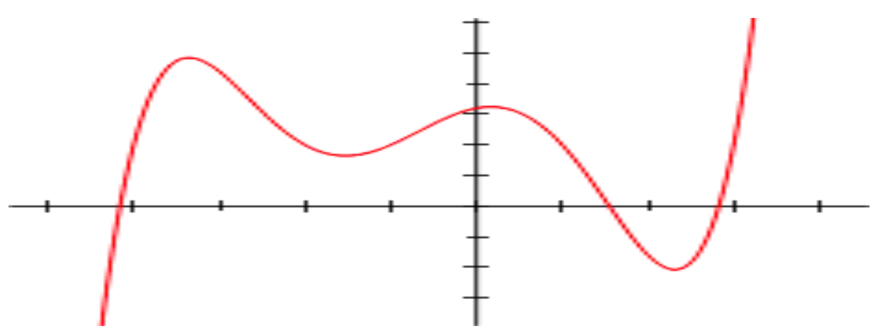
+ We don't care about space anymore.

– We must deal with the exponentially small success probability regime.

Two main methods for proving such lower bounds:

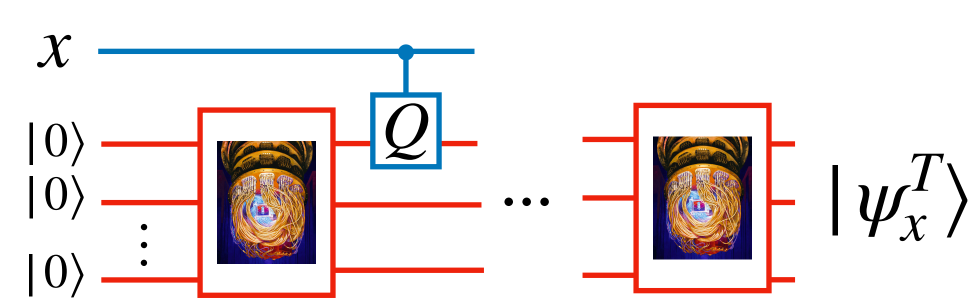
Polynomial Method

The acceptance probability of a T-query algorithm is a polynomial in x of degree at most 2T.



Adversary Method

Bound the progress $W^t = \sum_{x,y} w_{x,y} \langle \psi_x^t | \psi_y^t \rangle$.



Time Lower Bounds

It is impossible to find **K collisions**
with success probability ≥ $2^{-O(K)}$
in time $\tau(K) = O\left(K^{2/3}N^{1/3}\right)$.

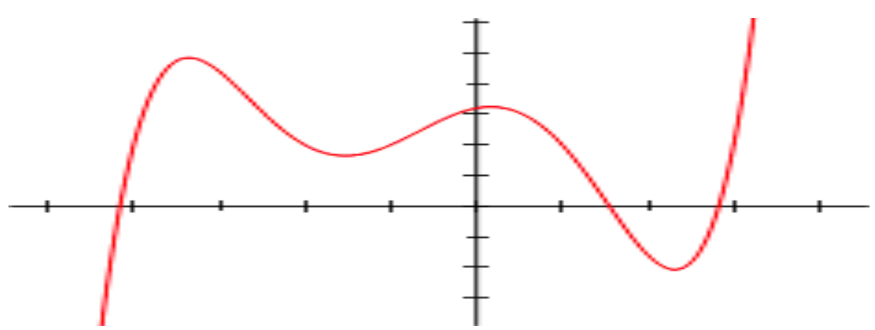
+ We don't care about space anymore.

– We must deal with the exponentially small success probability regime.

Two main methods for proving such lower bounds:

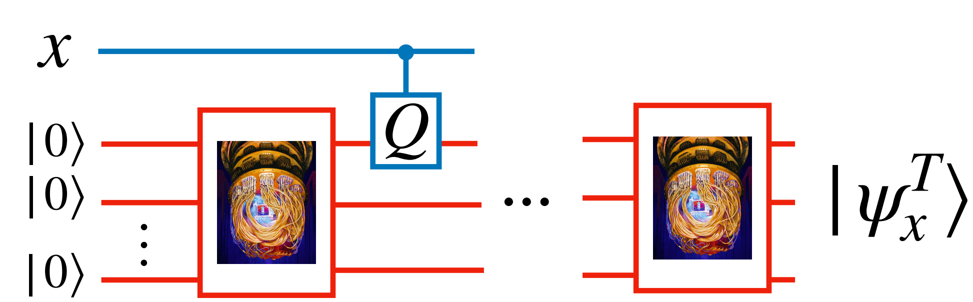
Polynomial Method

The acceptance probability of a T-query algorithm is a polynomial in x of degree at most 2T.



Adversary Method

Bound the progress $W^t = \sum_{x,y} w_{x,y} \langle \psi_x^t | \psi_y^t \rangle$.



Our approach: a refined version of Zhandry's **recording technique**

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm

Input

$$x = (\perp, \perp, \perp, \perp)$$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm

$\xrightarrow{x_2?}$

Input

$x = (\perp, \perp, \perp, \perp)$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm **Input**

$$x = (\perp, \perp, \perp, \perp)$$

$$\xrightarrow{x_2?}$$

$$x = (\perp, 4, \perp, \perp)$$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm **Input**

$$x = (\perp, \perp, \perp, \perp)$$

$$\xrightarrow{x_2?}$$

$$\xleftarrow{x_2 = 4}$$

$$x = (\perp, 4, \perp, \perp)$$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm | **Input**

$$x = (\perp, \perp, \perp, \perp)$$

$$\xrightarrow{x_2?}$$

$$\xleftarrow{x_2 = 4}$$

$$x = (\perp, 4, \perp, \perp)$$

$$\xrightarrow{x_1?}$$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm | **Input**

$x = (\perp, \perp, \perp, \perp)$

$\xrightarrow{x_2?}$

$\xleftarrow{x_2 = 4}$

$x = (\perp, 4, \perp, \perp)$

$\xrightarrow{x_1?}$

$x = (7, 4, \perp, \perp)$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm

Input

$x = (\perp, \perp, \perp, \perp)$

$\xrightarrow{x_2 ?}$

$\xleftarrow{x_2 = 4}$

$x = (\perp, 4, \perp, \perp)$

$\xrightarrow{x_1 ?}$

$\xleftarrow{x_1 = 7}$

$x = (7, 4, \perp, \perp)$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm | **Input**

$$x = (\perp, \perp, \perp, \perp)$$

$$\xrightarrow{x_2?}$$

$$\xleftarrow{x_2 = 4} \qquad x = (\perp, 4, \perp, \perp)$$

$$\xrightarrow{x_1?}$$

$$\xleftarrow{x_1 = 7} \qquad x = (7, 4, \perp, \perp)$$

$$\xrightarrow{x_2?}$$

$$\xleftarrow{x_2 = 4} \qquad x = (7, 4, \perp, \perp)$$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm | **Input**

$x = (\perp, \perp, \perp, \perp)$

$\xrightarrow{x_2 ?}$

$\xleftarrow{x_2 = 4}$ $\quad x = (\perp, 4, \perp, \perp)$

$\xrightarrow{x_1 ?}$

$\xleftarrow{x_1 = 7}$ $\quad x = (7, 4, \perp, \perp)$

$\xrightarrow{x_2 ?}$

$\xleftarrow{x_2 = 4}$ $\quad x = (7, 4, \perp, \perp)$

$\xrightarrow{x_4 ?}$

$\xleftarrow{x_4 = 4}$ $\quad x = (7, 4, \perp, 4)$

$\vdots$

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm **Input**

$x = (\perp, \perp, \perp, \perp)$

$\xrightarrow{x_2?}$

$\xleftarrow{x_2 = 4}$ $x = (\perp, 4, \perp, \perp)$

$\xrightarrow{x_1?}$

$\xleftarrow{x_1 = 7}$ $x = (7, 4, \perp, \perp)$

$\xrightarrow{x_2?}$

$\xleftarrow{x_2 = 4}$ $x = (7, 4, \perp, \perp)$

$\xrightarrow{x_4?}$

$\xleftarrow{x_4 = 4}$ $x = (7, 4, \perp, 4)$

$\vdots$

Progress measure: probability to have recorded at least **k** collisions after **t** queries

Classical Recording

Input: $x = (x_1, \ldots, x_N)$ where $x_i = y \in [N]$ with probability $1/N$.

Strategy: sample each entry only when it is queried, and record its value.

Algorithm **Input**

$x = (\perp, \perp, \perp, \perp)$

$\xrightarrow{x_2?}$

$\xleftarrow{x_2 = 4}$ $x = (\perp, 4, \perp, \perp)$

$\xrightarrow{x_1?}$

$\xleftarrow{x_1 = 7}$ $x = (7, 4, \perp, \perp)$

$\xrightarrow{x_2?}$

$\xleftarrow{x_2 = 4}$ $x = (7, 4, \perp, \perp)$

$\xrightarrow{x_4?}$

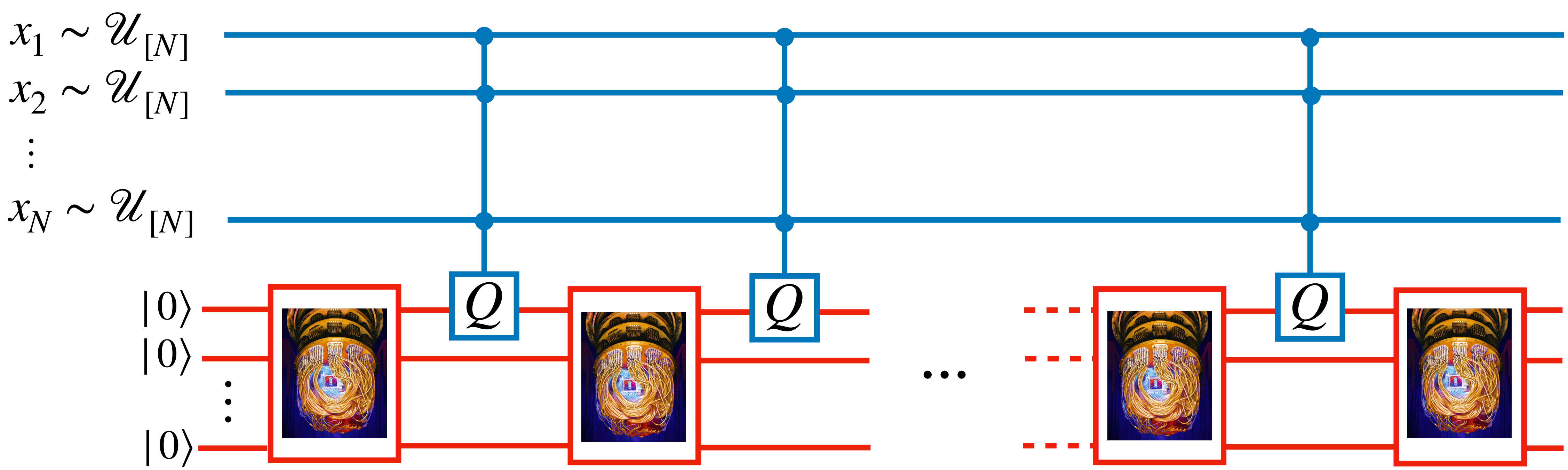
$\xleftarrow{x_4 = 4}$ $x = (7, 4, \perp, 4)$

$\vdots$

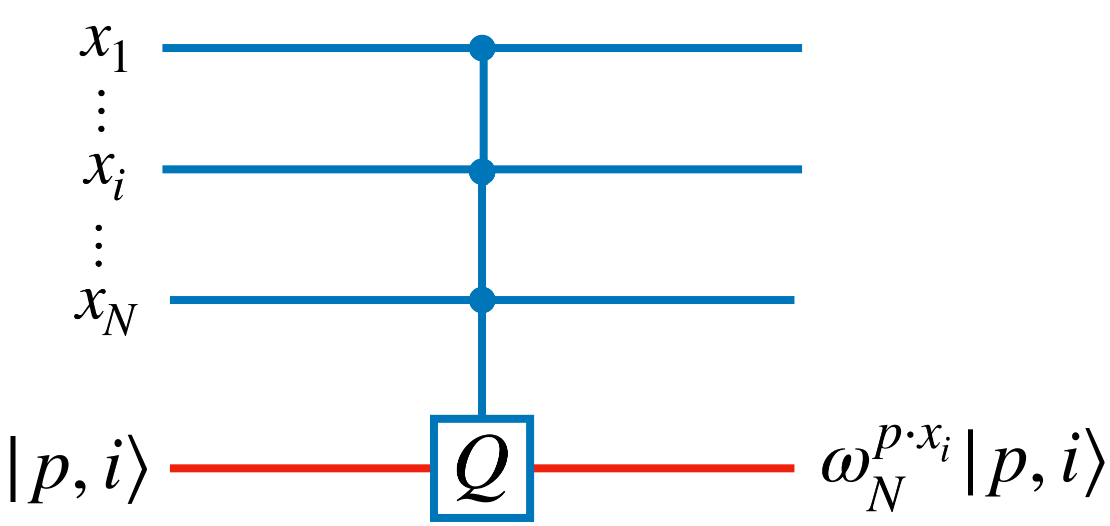
Progress measure: probability to have recorded at least **k** collisions after **t** queries

→ the algorithm must force the recording of many collisions to succeed with high probability

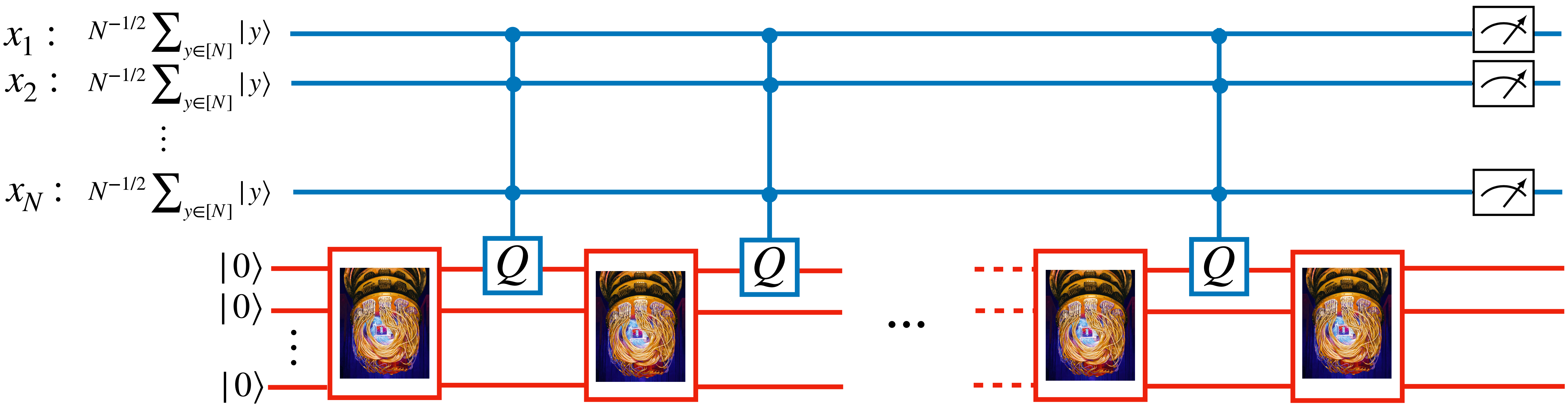
Quantum Recording



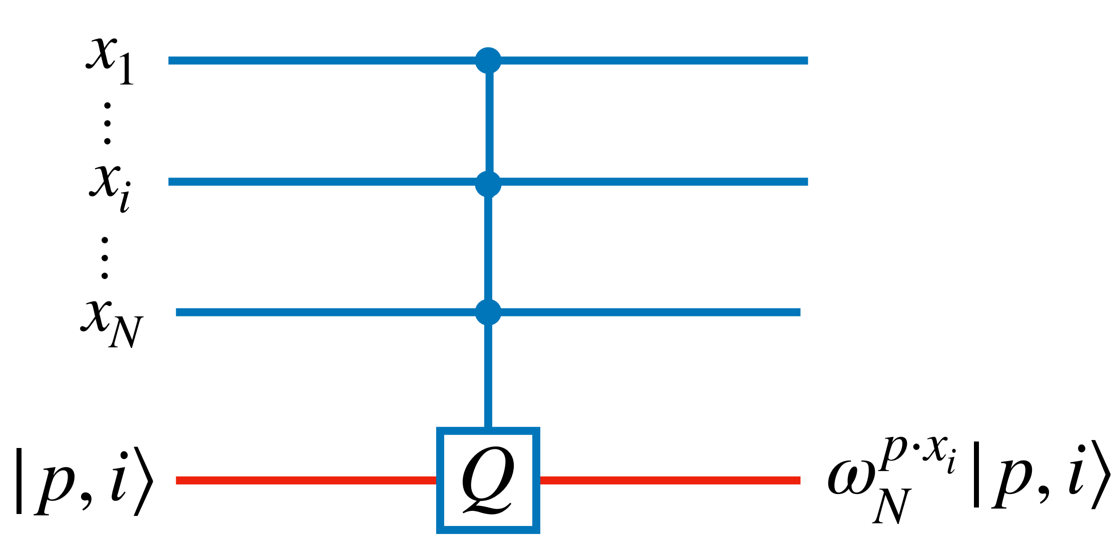
Query Operator



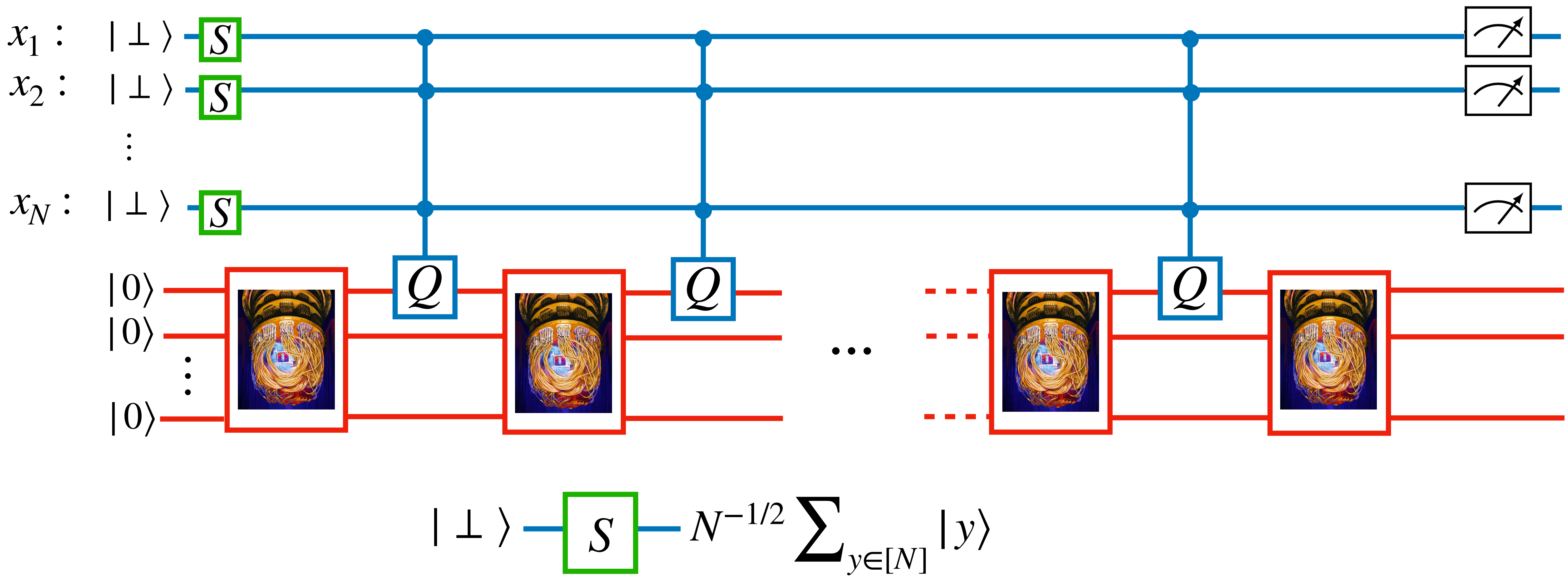
Quantum Recording



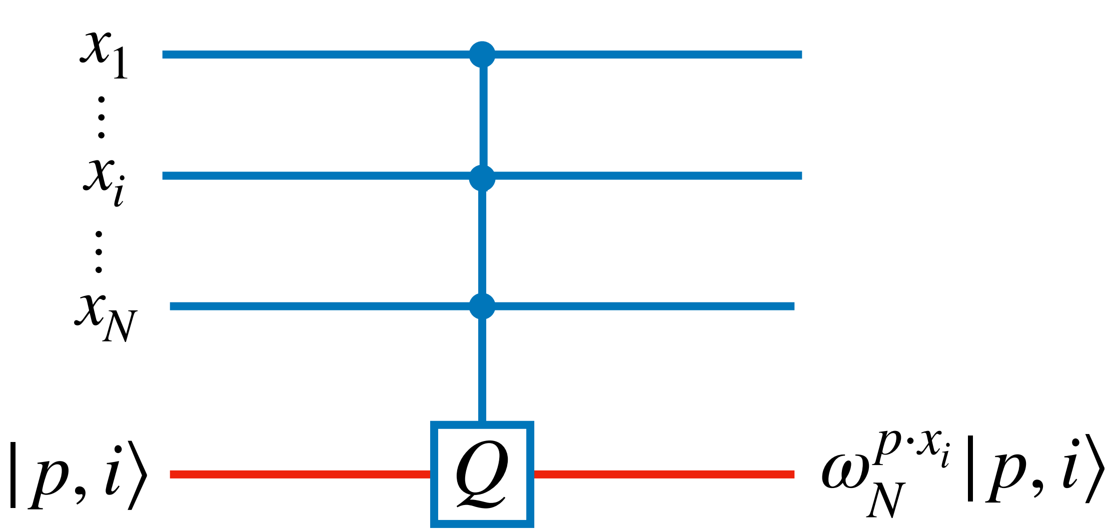
Query Operator



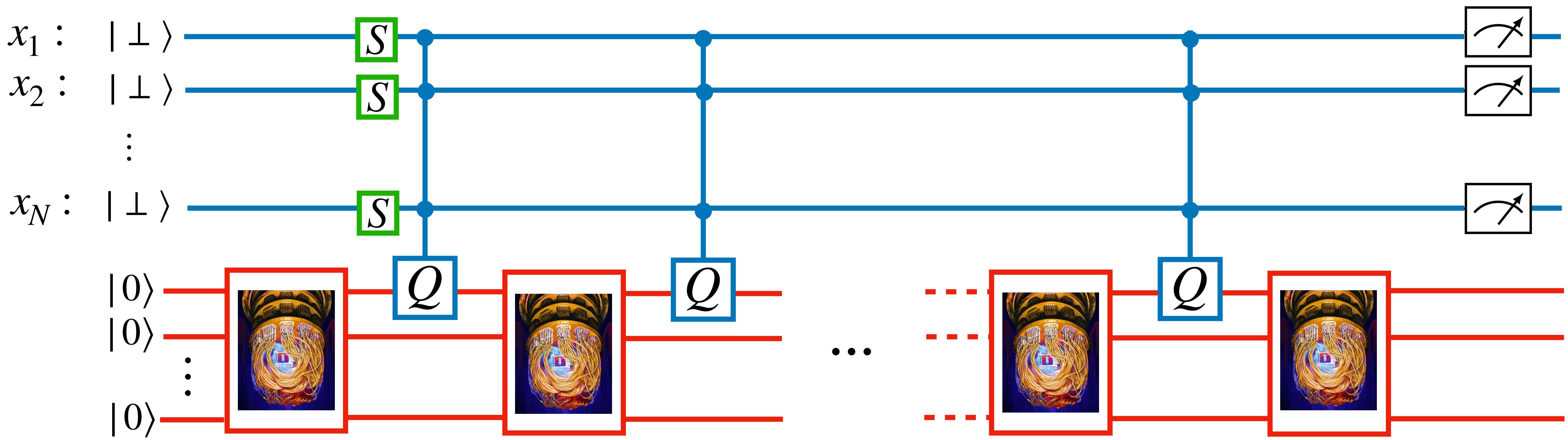
Quantum Recording

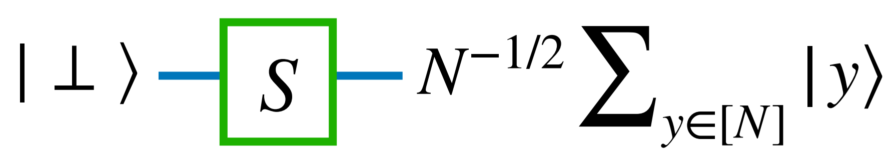


Query Operator

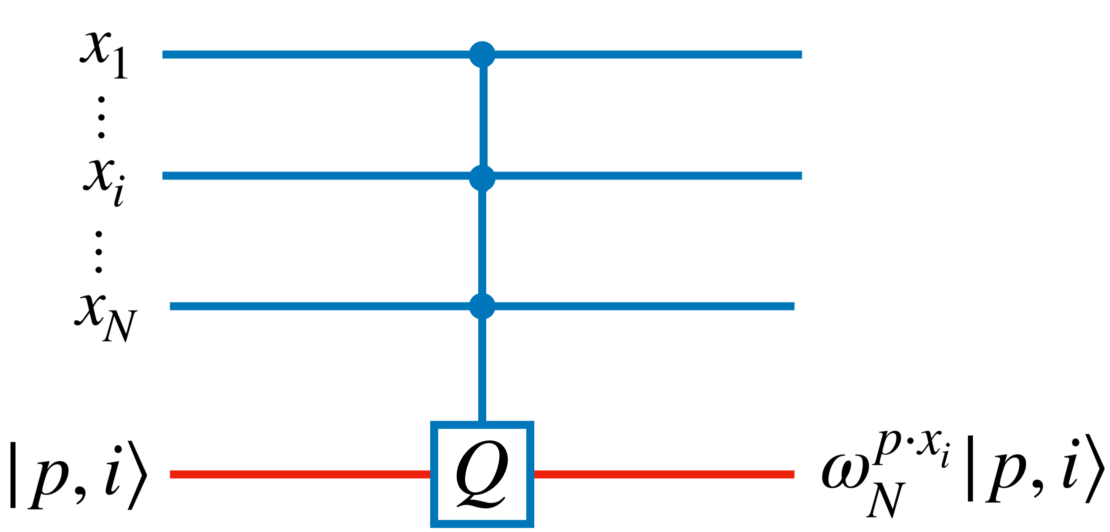


Quantum Recording

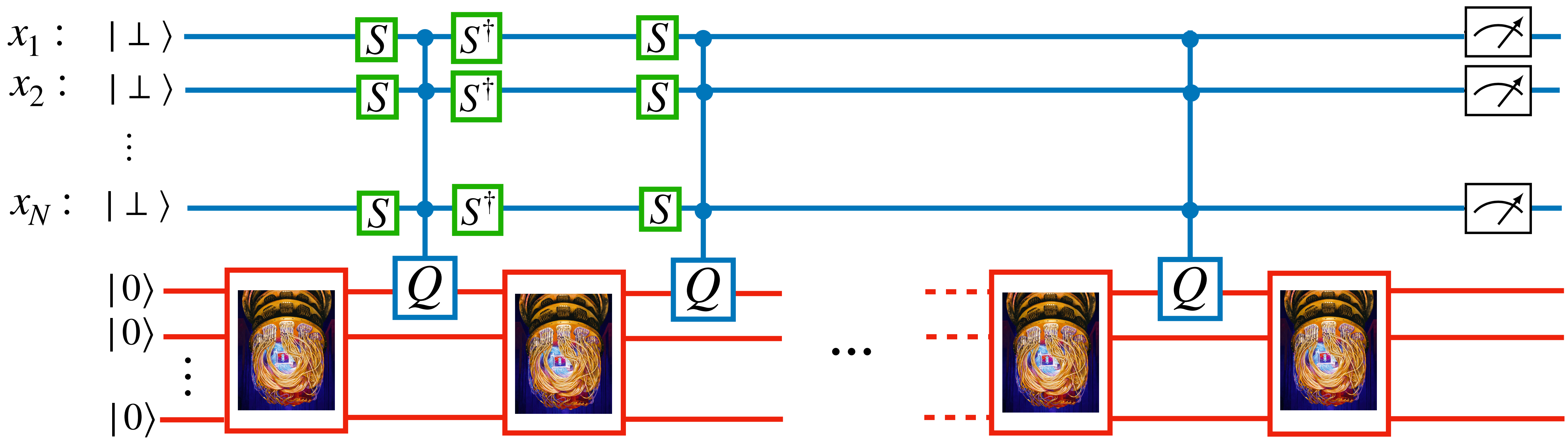


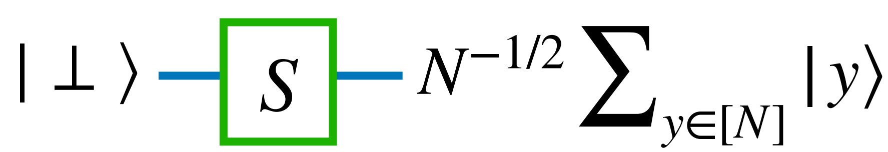


Query Operator

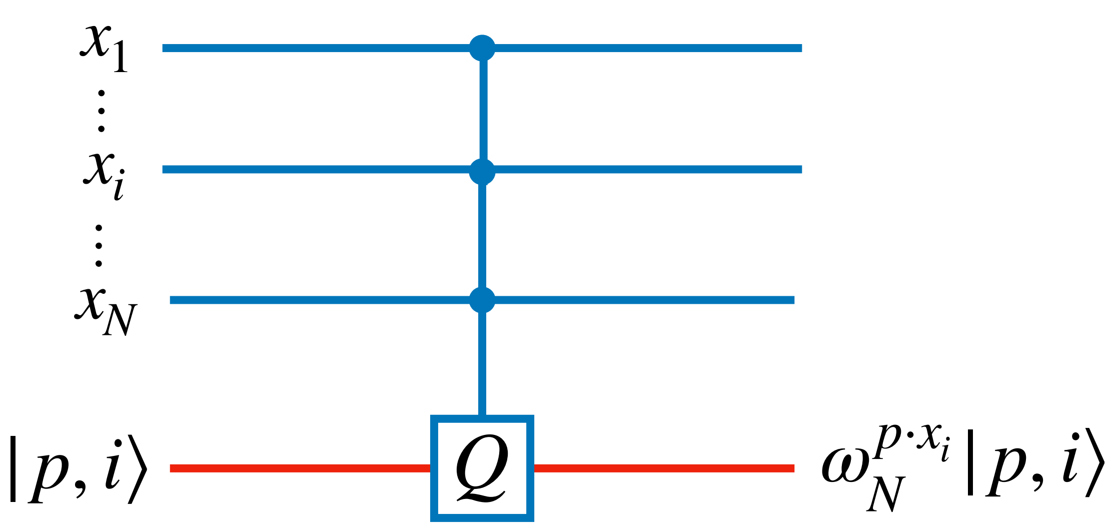


Quantum Recording

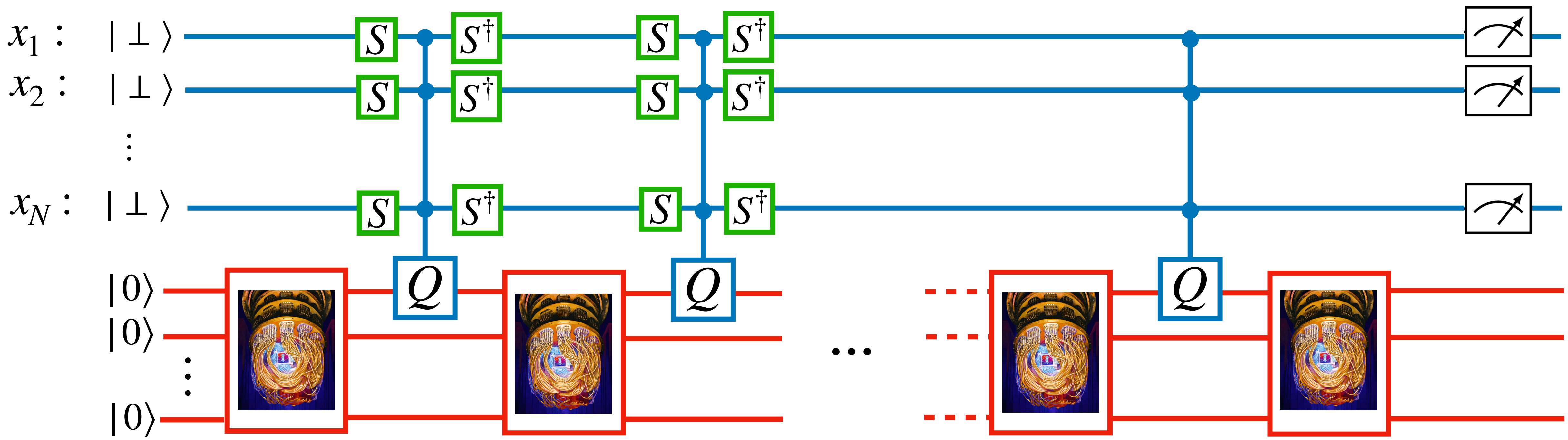


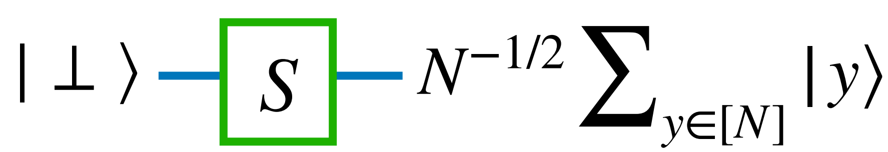


Query Operator

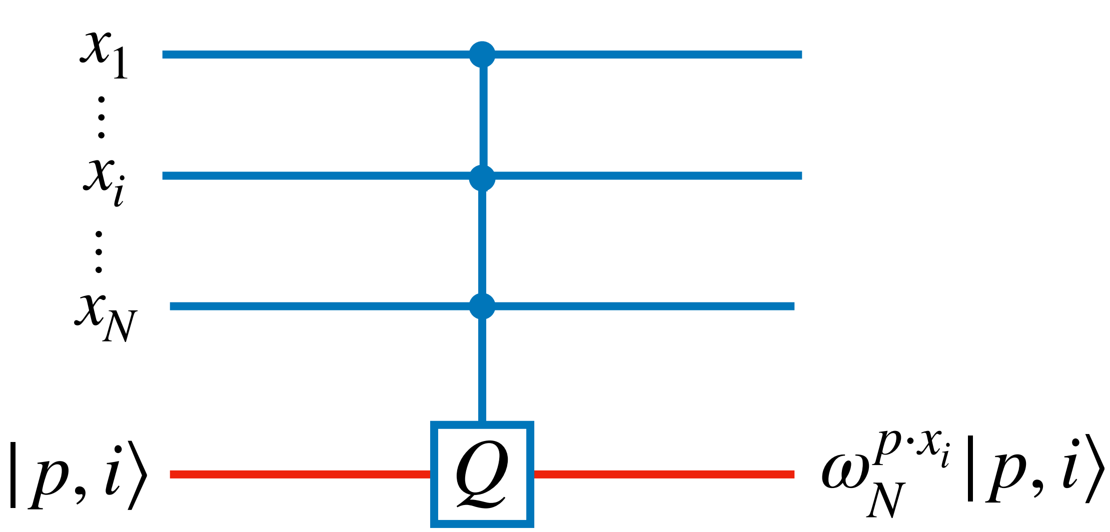


Quantum Recording

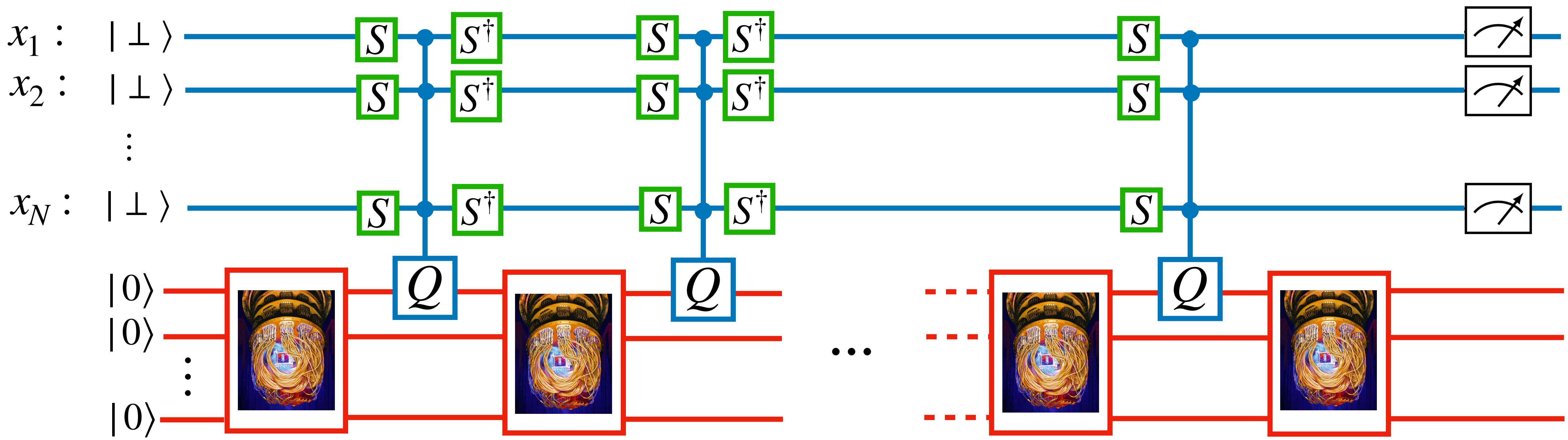




Query Operator

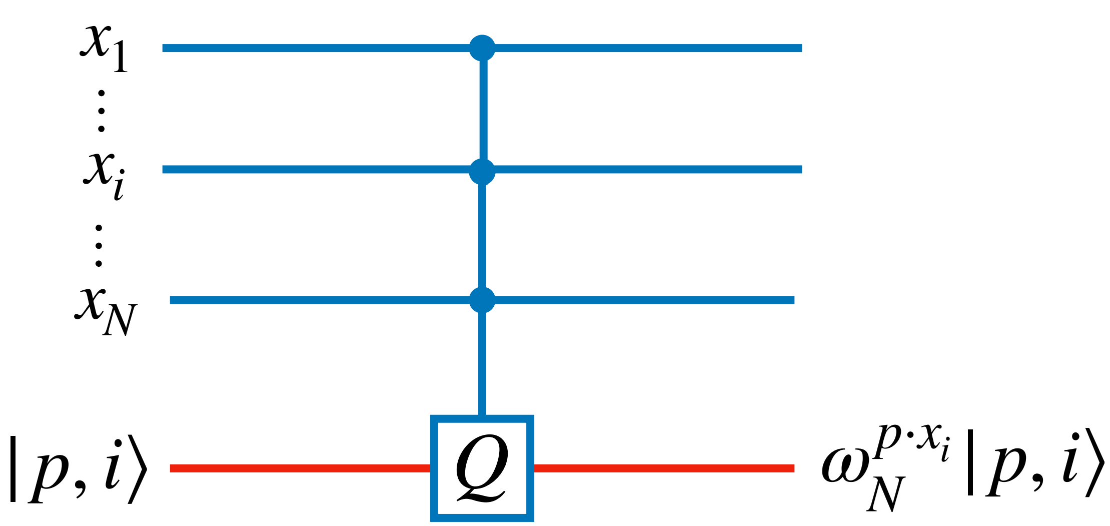


Quantum Recording

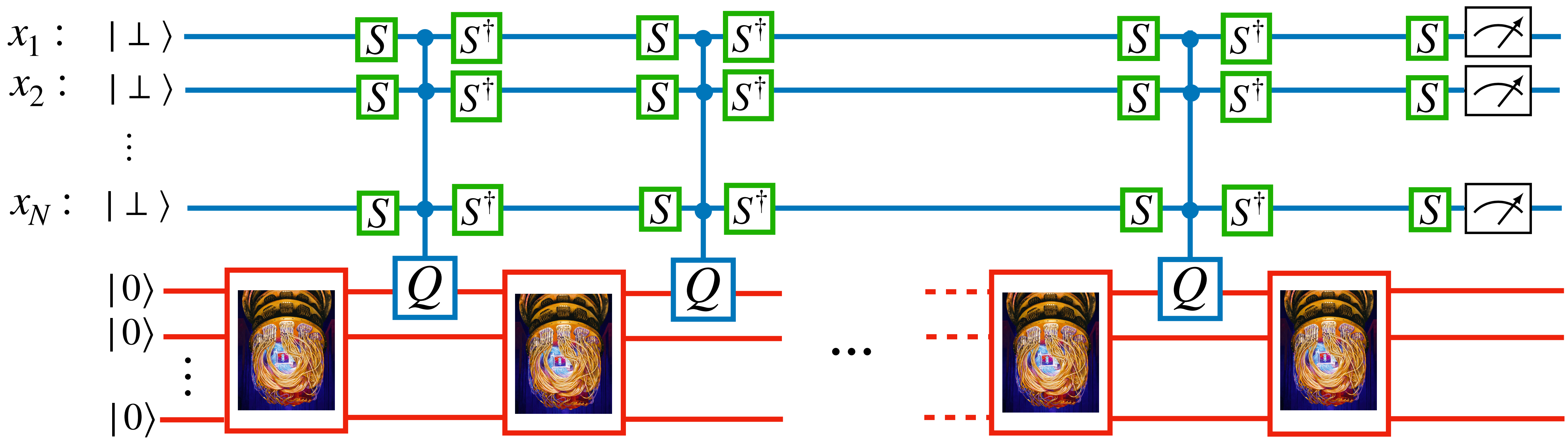


$$|\perp\rangle \; — \boxed{S} — \; N^{-1/2} \sum_{y \in [N]} |y\rangle$$

Query Operator

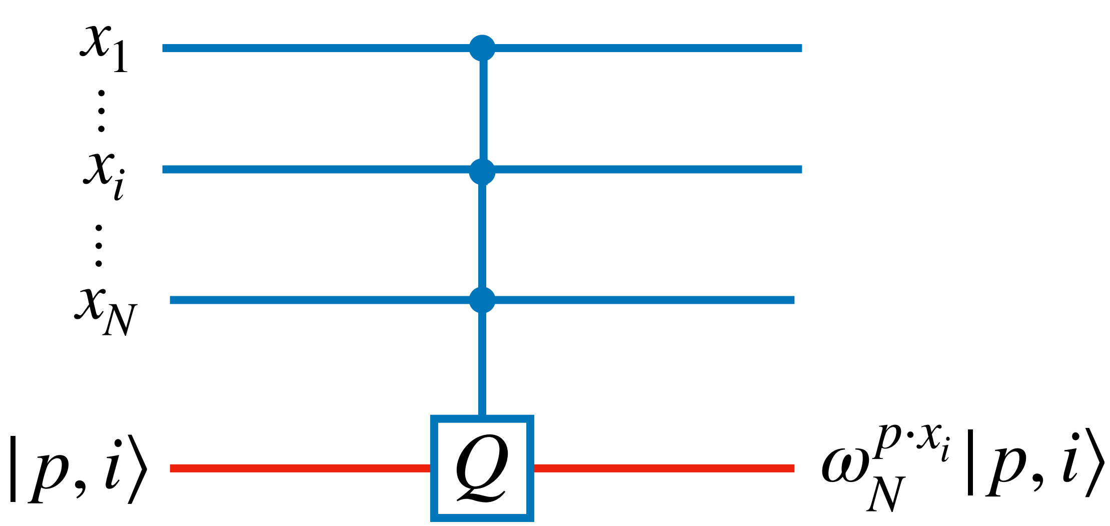


Quantum Recording

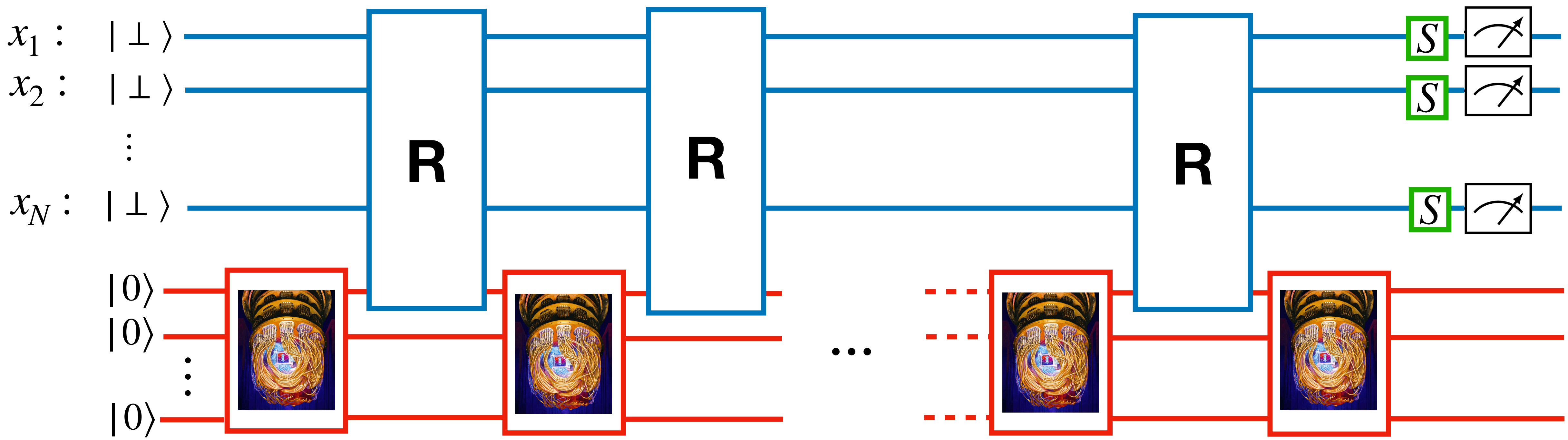


$$|\perp\rangle \; \boxed{S} \; N^{-1/2}\sum_{y\in[N]}|y\rangle$$

Query Operator

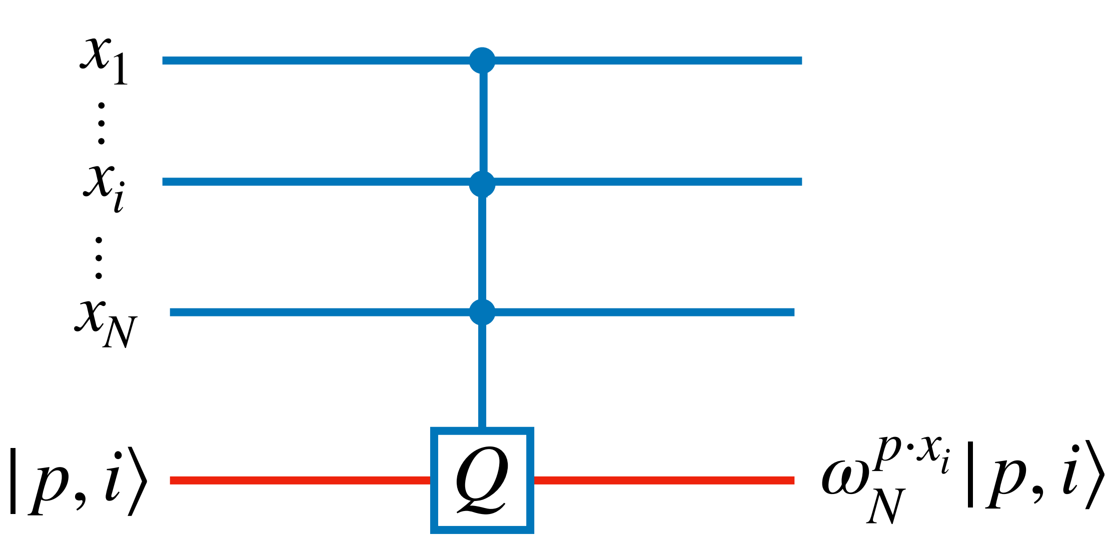


Quantum Recording

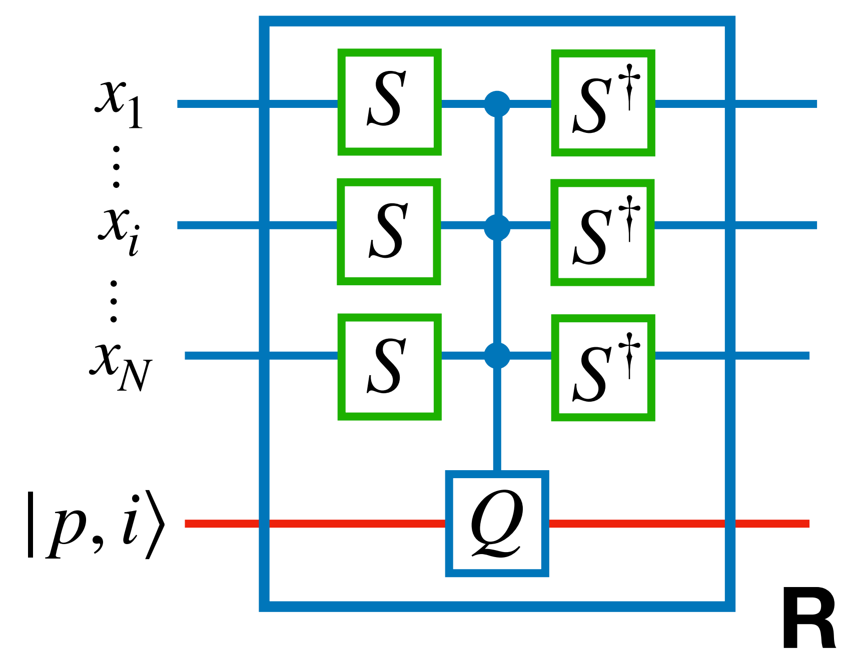


$$|\perp\rangle \xrightarrow{S} N^{-1/2}\sum_{y\in[N]}|y\rangle$$

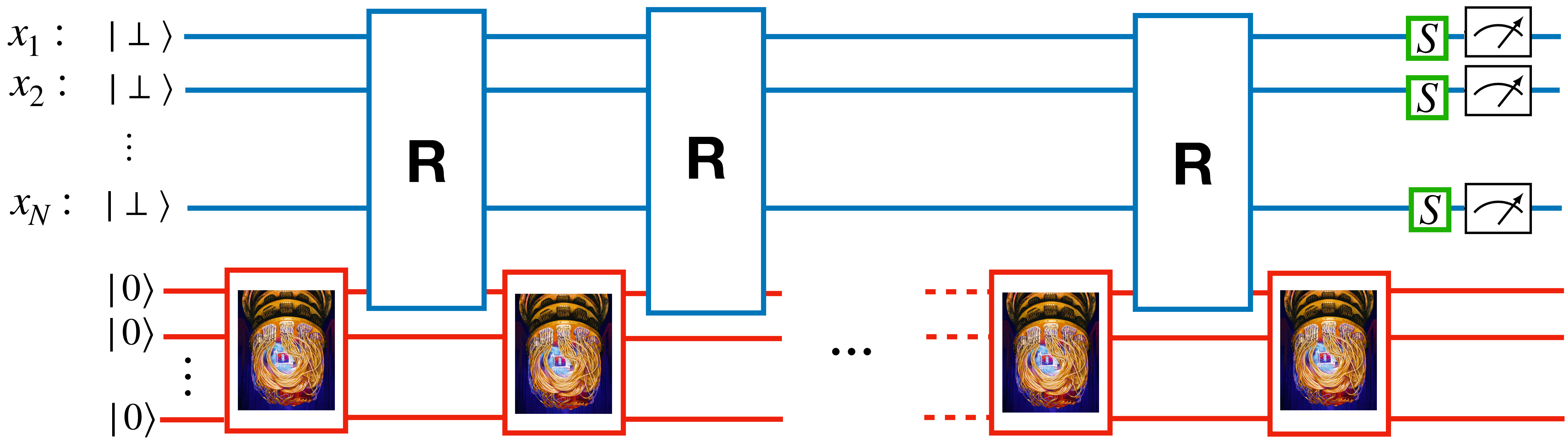
Query Operator



Recording Query Operator



Quantum Recording

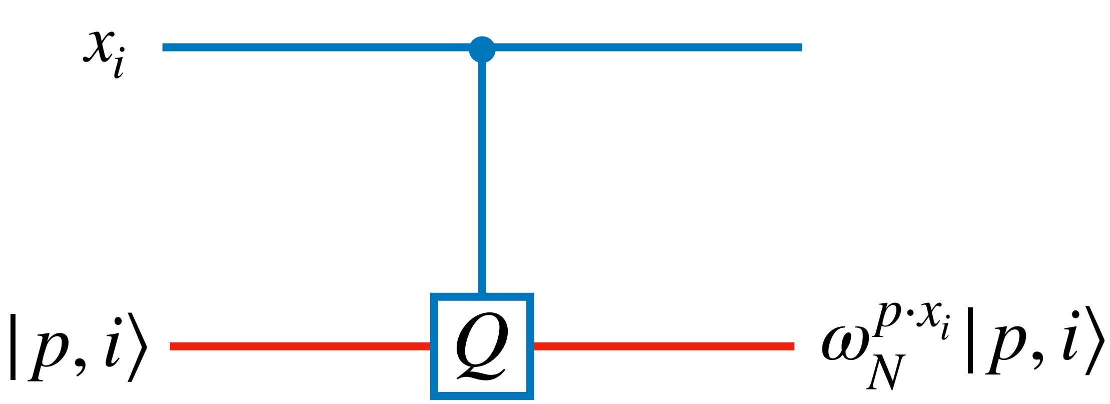


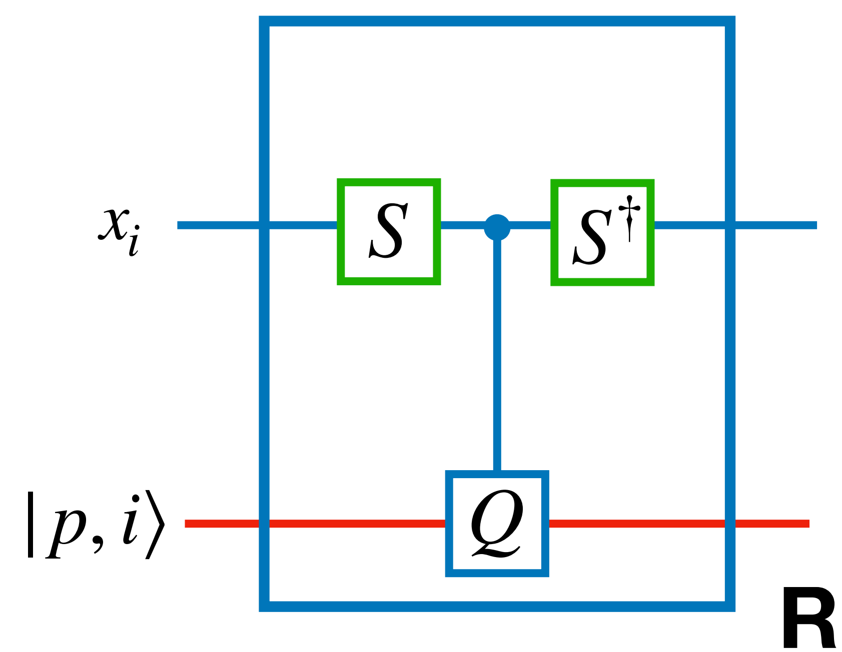
$|\perp\rangle$ — $\boxed{S}$ — $N^{-1/2}\sum_{y\in[N]}|y\rangle$

Query Operator

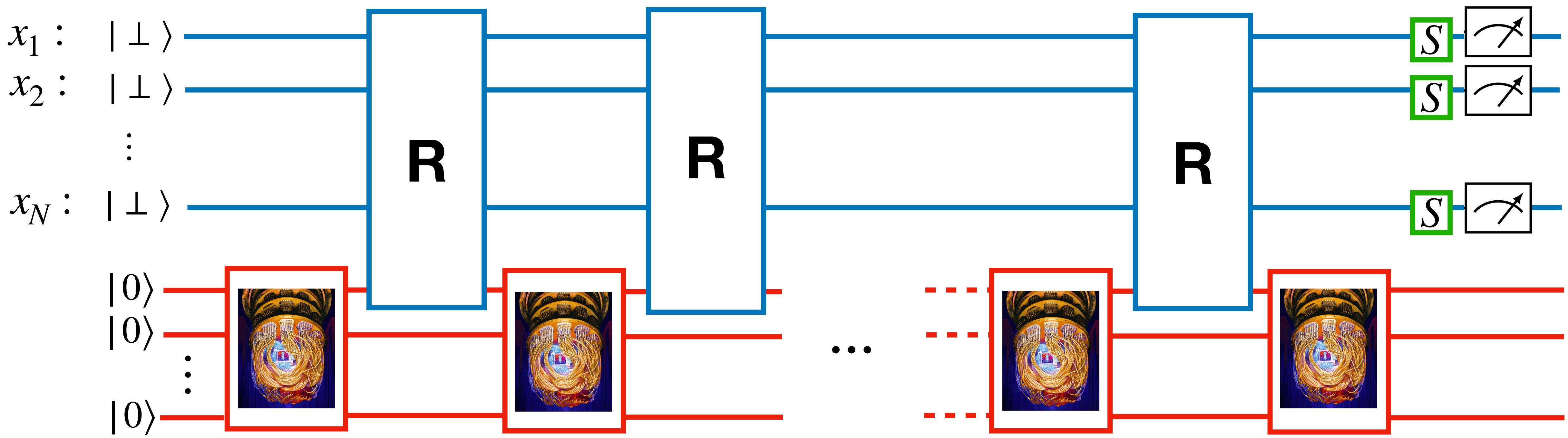
Recording Query Operator

✔ If x_j is not queried it stays unchanged.

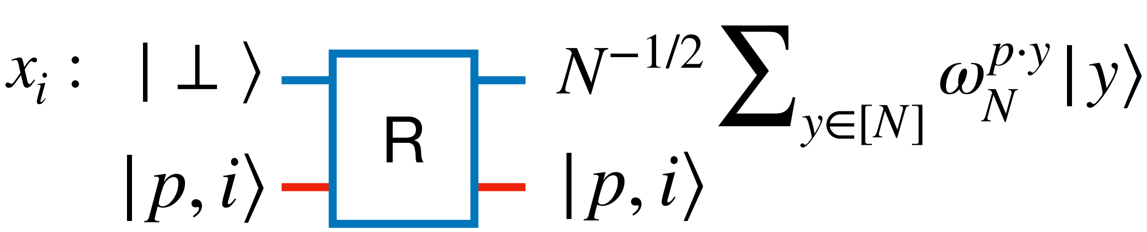




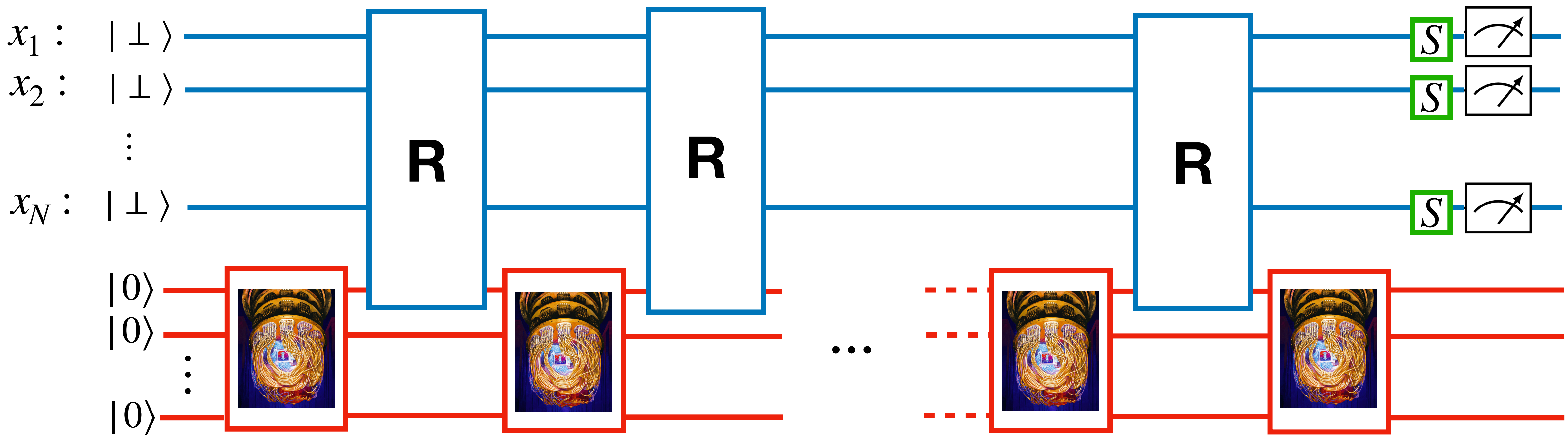
Quantum Recording



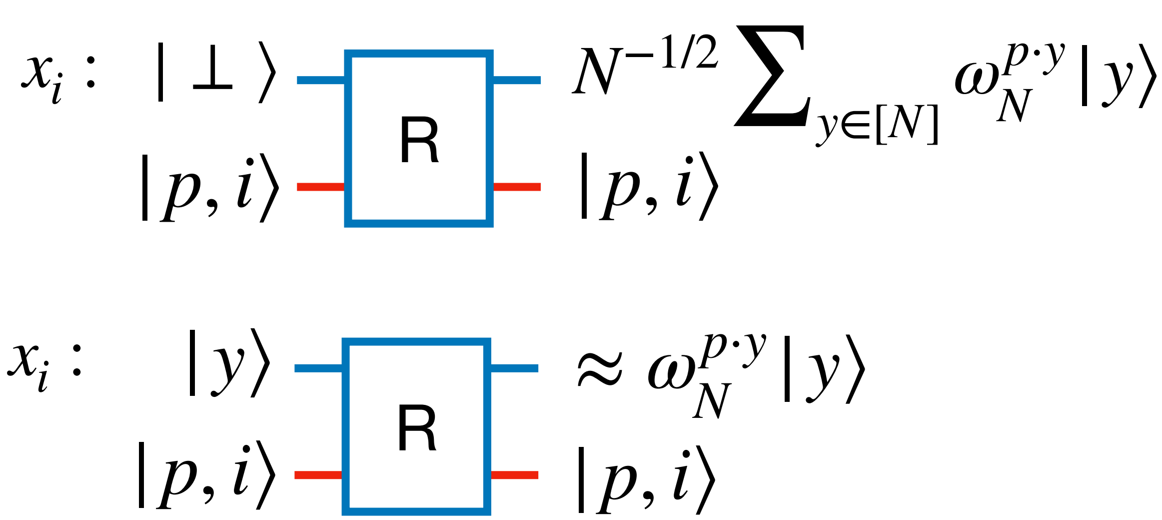
Recording Query Operator



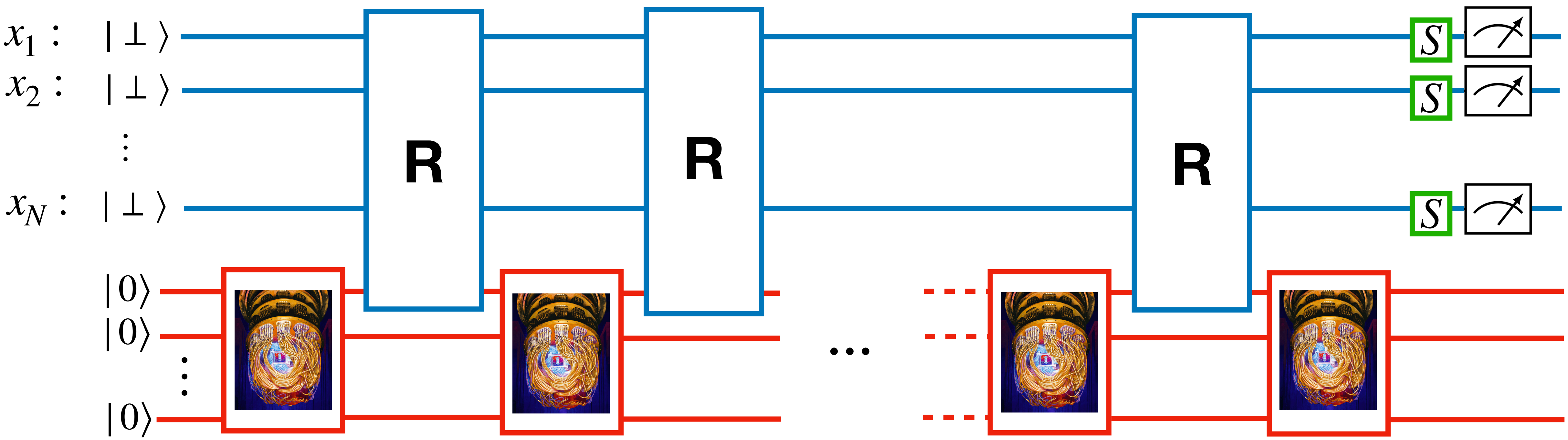
Quantum Recording



Recording Query Operator



Quantum Recording



Recording Query Operator

$$x_i: \quad |\perp\rangle \;\text{—}\boxed{\mathrm{R}}\text{—}\; N^{-1/2}\sum_{y\in[N]}\omega_N^{p\cdot y}|y\rangle$$

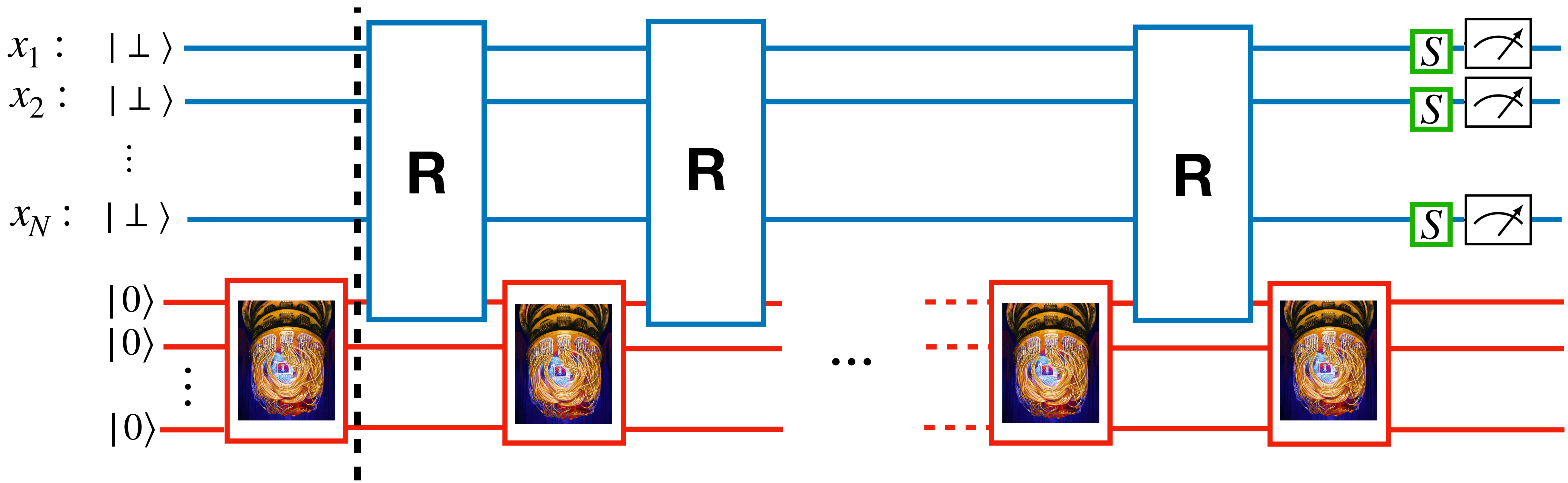
$$|p,i\rangle \;\text{—}\boxed{\mathrm{R}}\text{—}\; |p,i\rangle$$

$$x_i: \quad |y\rangle \;\text{—}\boxed{\mathrm{R}}\text{—}\; \approx \omega_N^{p\cdot y}|y\rangle$$

$$|p,i\rangle \;\text{—}\boxed{\mathrm{R}}\text{—}\; |p,i\rangle$$

$\Delta(t,k)$ = **amplitude** of the basis states containing $\geq$ $\boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

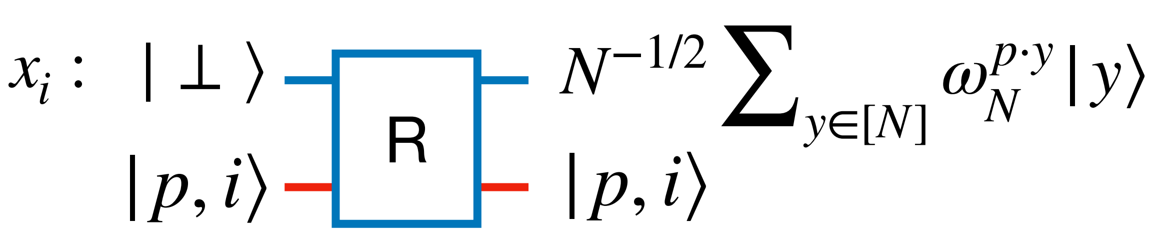
Quantum Recording

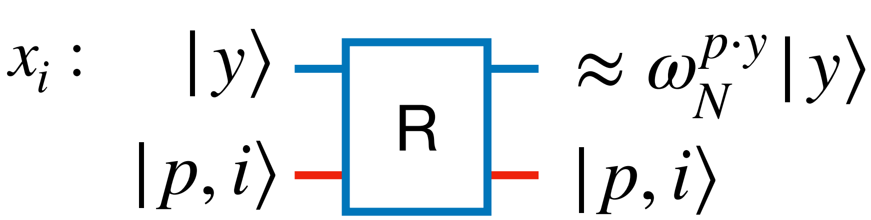


$\Delta(0,0) = 1$

$\Delta(0,k) = 0$ for $k > 0$

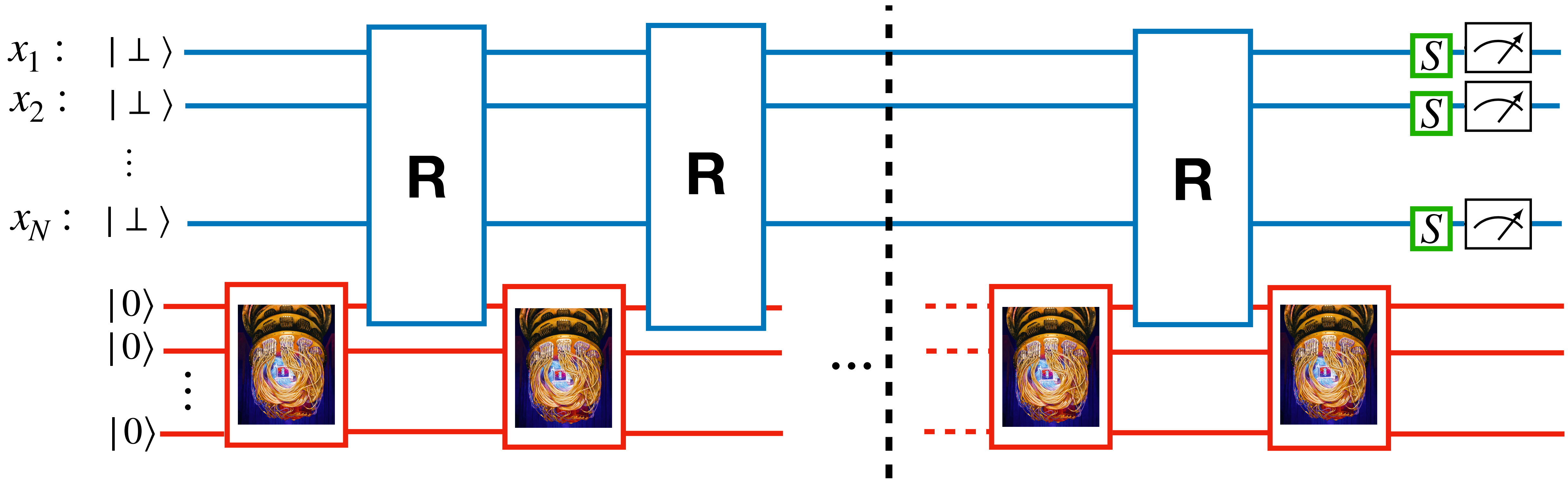
Recording Query Operator





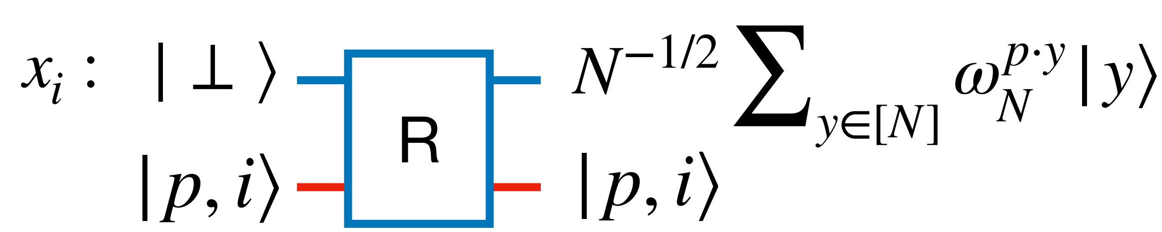
$\Delta(t, k)$ = **amplitude** of the basis states containing $\geq \boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

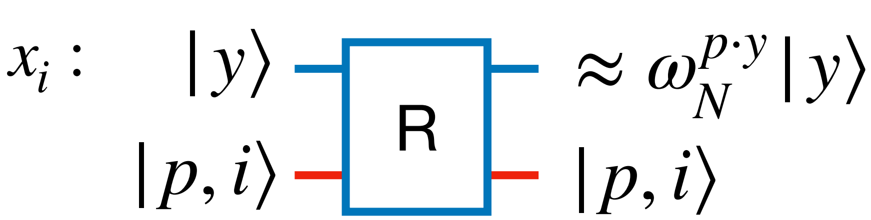
Quantum Recording



$$\Delta(t+1, k+1) \le \Delta(t, k+1) + \Delta(t,k) \cdot O\left(\sqrt{\frac{t}{N}}\right)$$

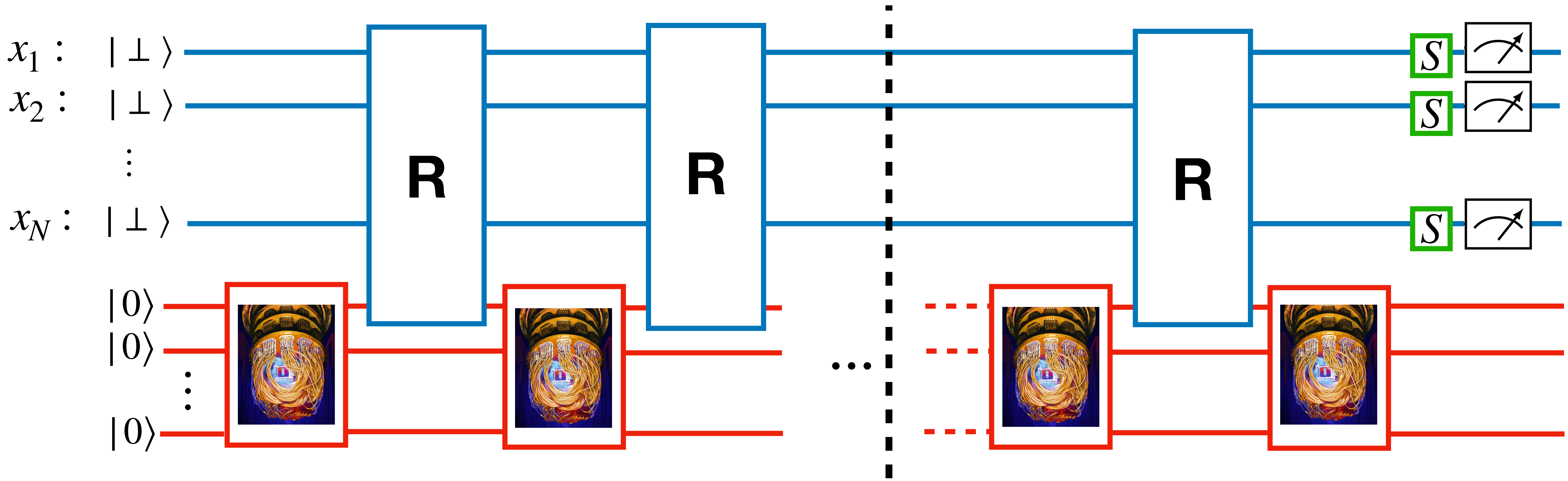
Recording Query Operator





$\Delta(t,k)$ = **amplitude** of the basis states containing $\ge$ $\boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Quantum Recording



$$\Delta(t+1, k+1) \leq \Delta(t, k+1) + \Delta(t,k) \cdot O\left(\sqrt{\frac{t}{N}}\right)$$

Recording Query Operator

Record

$x_i : \ |\perp\rangle$ —[R]— $N^{-1/2} \sum_{y \in [N]} \omega_N^{p\cdot y} |y\rangle$

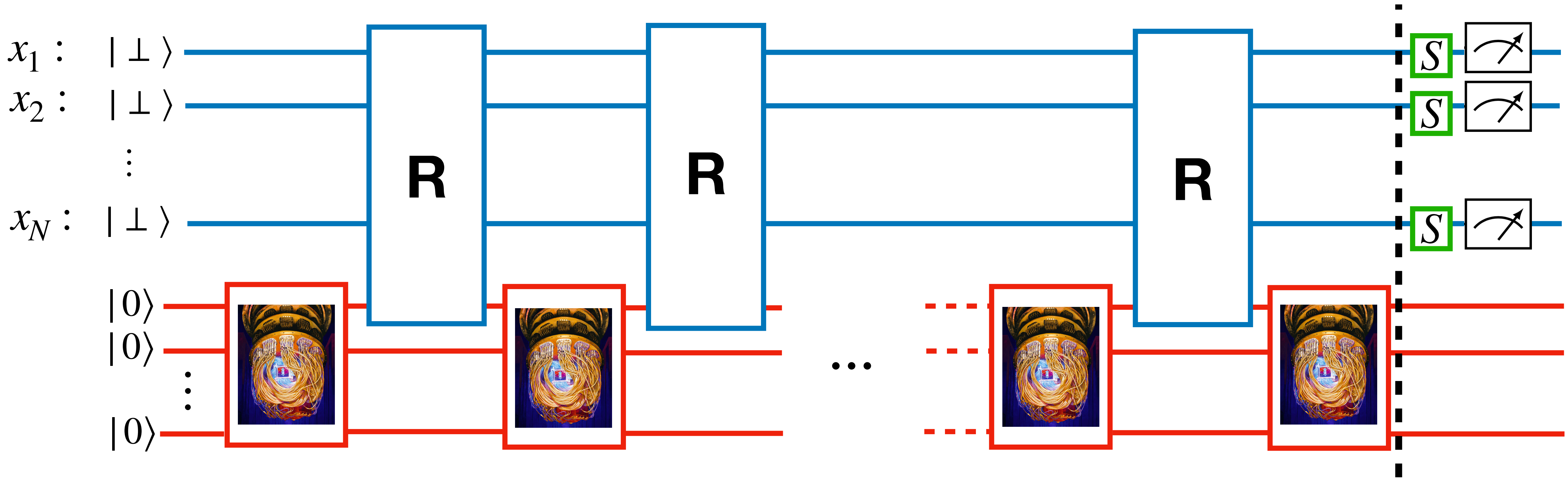
$|p, i\rangle$ —[R]— $|p, i\rangle$

$x_i : \ |y\rangle$ —[R]— $\approx \omega_N^{p \cdot y} |y\rangle$

$|p, i\rangle$ —[R]— $|p, i\rangle$

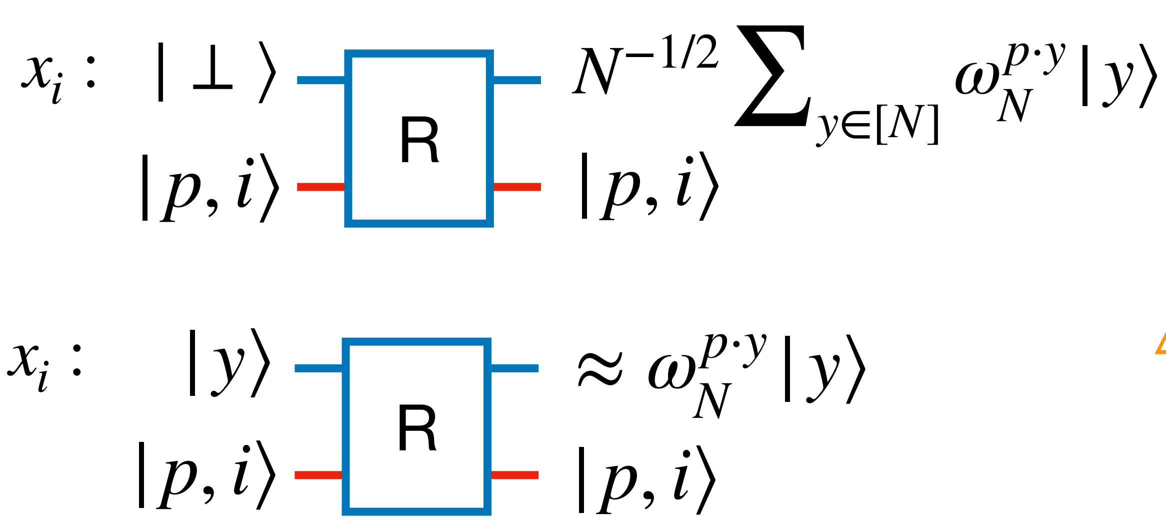
$\Delta(t, k)$ = **amplitude** of the basis states containing $\geq \boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Quantum Recording



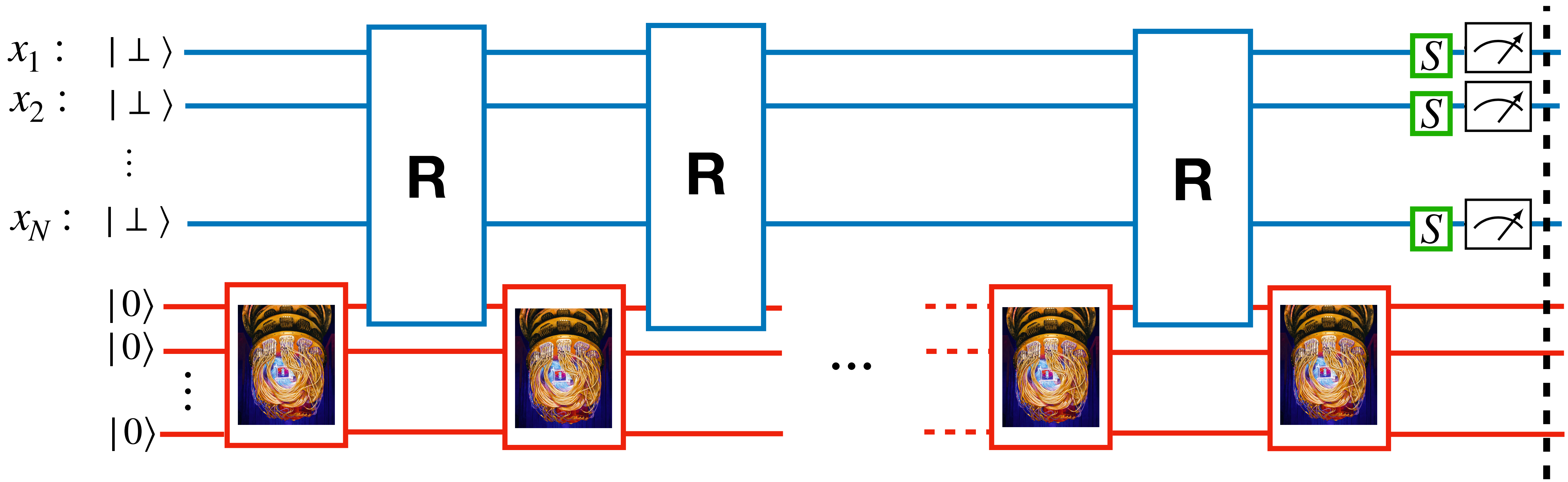
$\Delta(K^{2/3}N^{1/3}, K/2) \leq 2^{-K}$

Recording Query Operator



$\Delta(t, k)$ = **amplitude** of the basis states containing $\geq$ $\boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Quantum Recording



Success
≤
$\Delta(K^{2/3}N^{1/3}, K/2)$
$+\ O(K/N)^{K/2}$

Recording Query Operator

$x_i :\ |\perp\rangle$ — R — $N^{-1/2}\sum_{y\in[N]} \omega_N^{p\cdot y}\,|y\rangle$

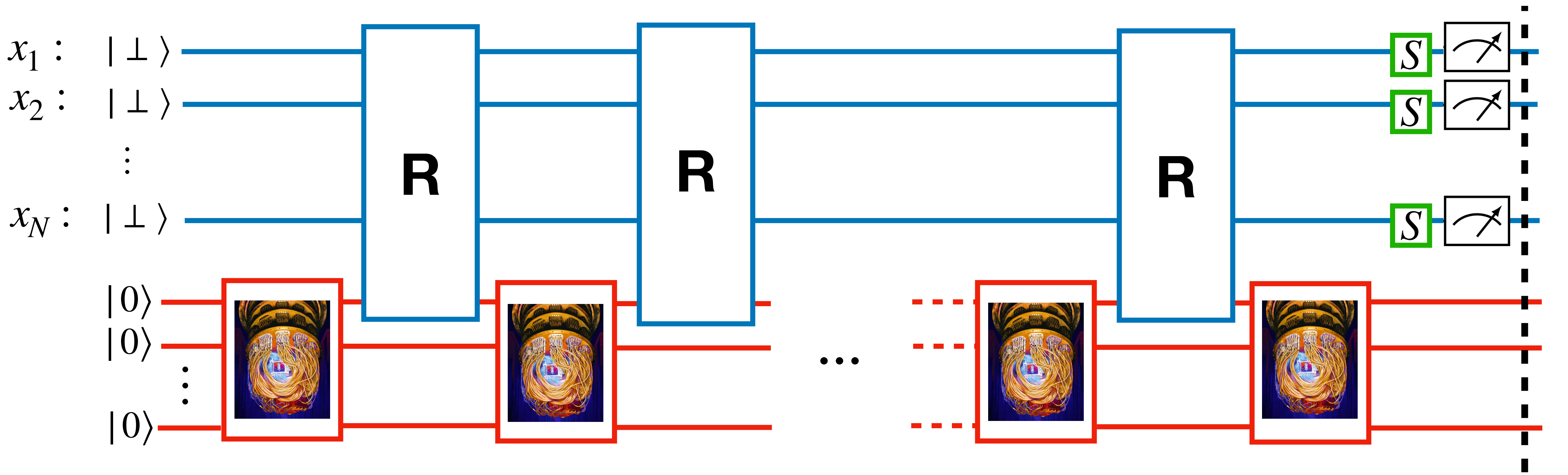
$|p,i\rangle$ — R — $|p,i\rangle$

$x_i :\ |y\rangle$ — R — $\approx \omega_N^{p\cdot y}\,|y\rangle$

$|p,i\rangle$ — R — $|p,i\rangle$

$\Delta(t,k)$ = **amplitude** of the basis states containing $\geq \boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Quantum Recording



Success
≤
$\Delta(K^{2/3}N^{1/3}, K/2)$
$+ \; O(K/N)^{K/2}$

≈ guess K/2 unrecorded collisions

Recording Query Operator

$x_i : \; |\perp\rangle \xrightarrow{\text{R}} N^{-1/2} \sum_{y \in [N]} \omega_N^{p \cdot y} |y\rangle$

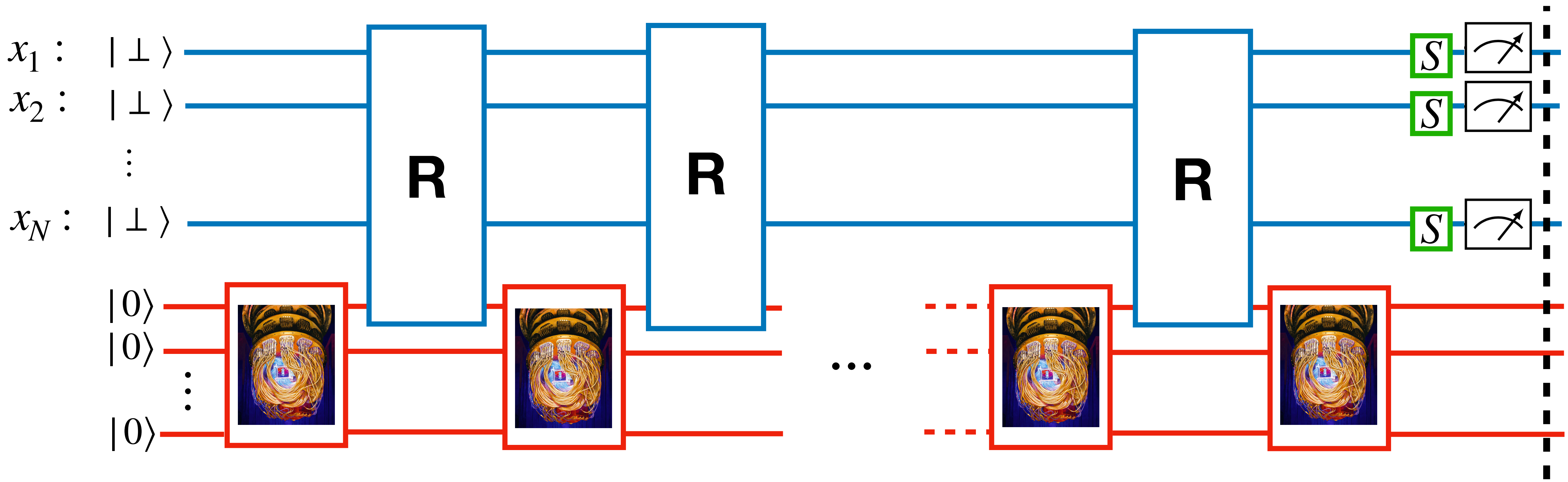
$|p, i\rangle \xrightarrow{\text{R}} |p, i\rangle$

$x_i : \; |y\rangle \xrightarrow{\text{R}} \approx \omega_N^{p \cdot y} |y\rangle$

$|p, i\rangle \xrightarrow{\text{R}} |p, i\rangle$

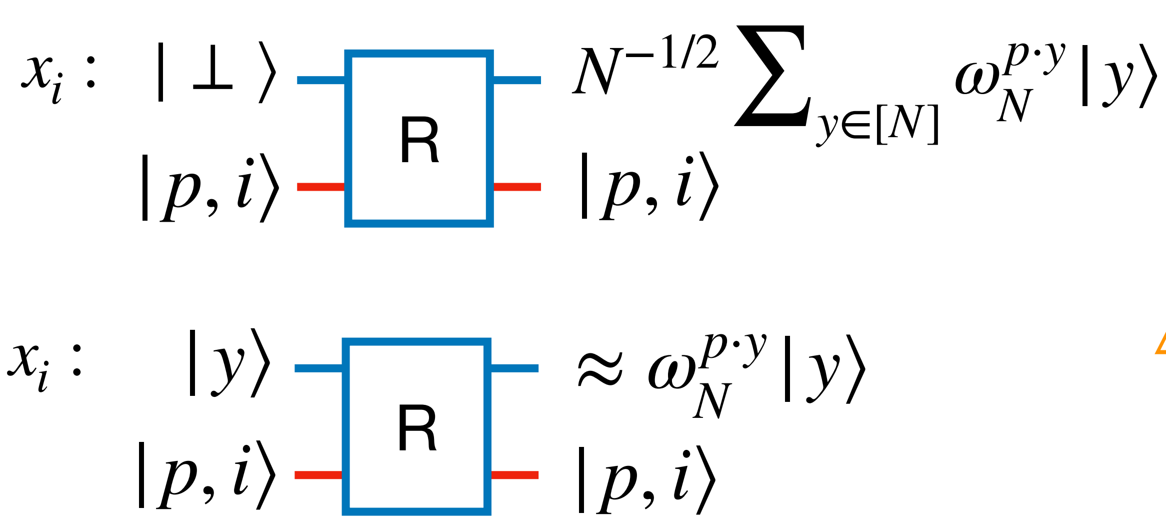
$\Delta(t, k)$ = **amplitude** of the basis states containing $\geq \boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Quantum Recording



Success
≤
$\Delta(K^{2/3}N^{1/3}, K/2)$
$+ \; O(K/N)^{K/2}$
$\lesssim 2^{-\Omega(K)}$

Recording Query Operator



$\Delta(t, k)$ = **amplitude** of the basis states containing $\geq \boldsymbol{k}$ (disjoint) collisions after $\boldsymbol{t}$ queries.

Conclusion

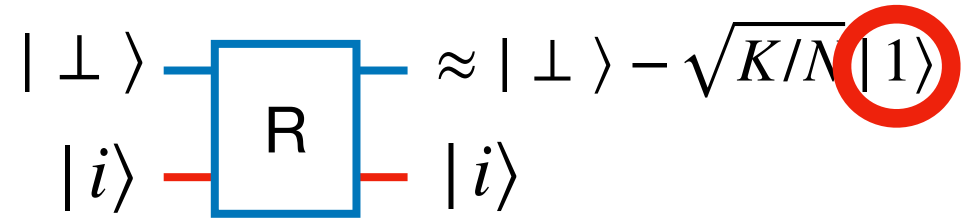
- Generalization to any product distribution

- Generalization to any product distribution

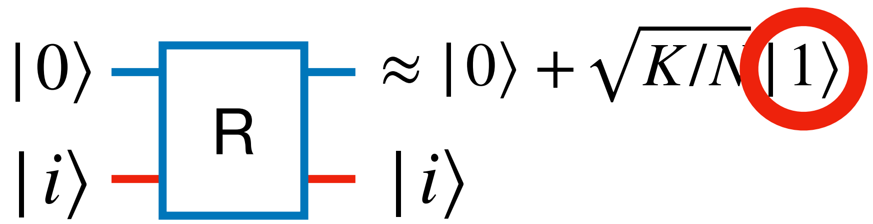
→ K-Search: find K ones in x where $x_i = 1$ with probability K/N

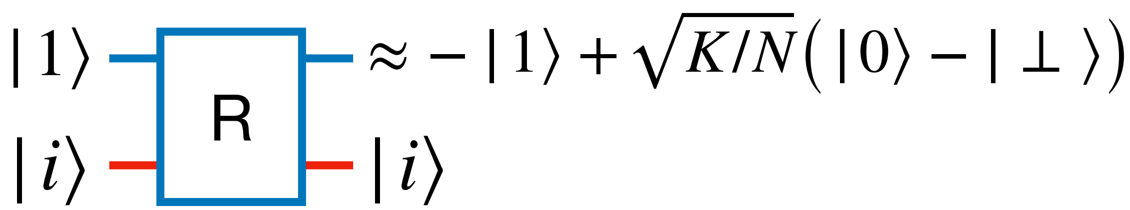
- Generalization to any product distribution

→ K-Search: find K ones in x where $x_i = 1$ with probability K/N



Record a new 1





- **Generalization to any product distribution**

→ K-Search: find K ones in x where $x_i = 1$ with probability K/N

$$\begin{array}{r} |\perp\rangle \\ |i\rangle \end{array} \boxed{\;R\;} \begin{array}{l} \approx |\perp\rangle - \sqrt{K/N}|1\rangle \\ |i\rangle \end{array}$$

Record a new 1

$$\begin{array}{r} |0\rangle \\ |i\rangle \end{array} \boxed{\;R\;} \begin{array}{l} \approx |0\rangle + \sqrt{K/N}|1\rangle \\ |i\rangle \end{array}$$

$$\begin{array}{r} |1\rangle \\ |i\rangle \end{array} \boxed{\;R\;} \begin{array}{l} \approx -|1\rangle + \sqrt{K/N}\left(|0\rangle - |\perp\rangle\right) \\ |i\rangle \end{array}$$

Solving the K-Search problem with success probability at least $2^{-O(K)}$ requires time $T \geq \Omega\left(\sqrt{NK}\right)$.

- **Generalization to any product distribution**

→ K-Search: find K ones in x where $x_i = 1$ with probability K/N

$$|\perp\rangle \;\text{—}\boxed{R}\text{—}\; \approx |\perp\rangle - \sqrt{K/N}|1\rangle$$
$$|i\rangle \;\text{—}\boxed{R}\text{—}\; |i\rangle$$

Record a new 1

$$|0\rangle \;\text{—}\boxed{R}\text{—}\; \approx |0\rangle + \sqrt{K/N}|1\rangle$$
$$|i\rangle \;\text{—}\boxed{R}\text{—}\; |i\rangle$$

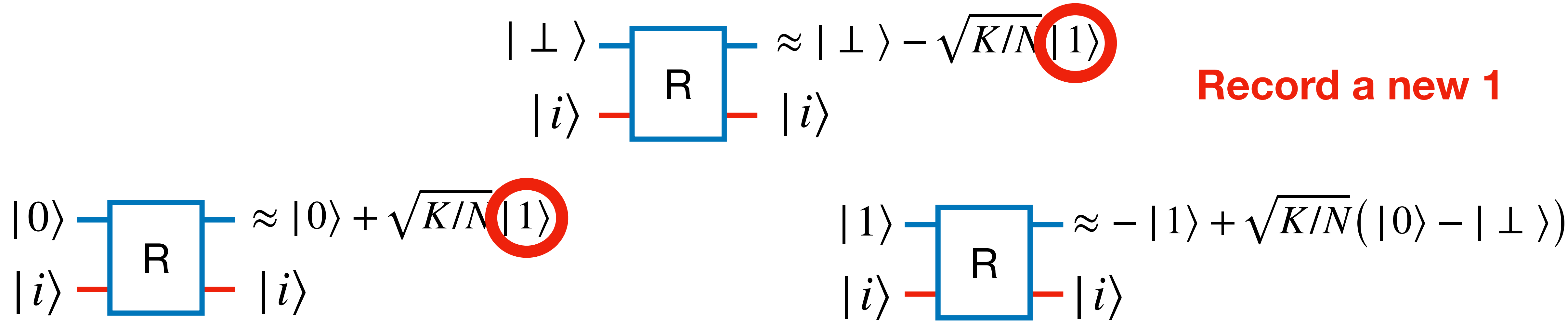
$$|1\rangle \;\text{—}\boxed{R}\text{—}\; \approx -|1\rangle + \sqrt{K/N}\big(|0\rangle - |\perp\rangle\big)$$
$$|i\rangle \;\text{—}\boxed{R}\text{—}\; |i\rangle$$

Solving the K-Search problem with success probability at least $2^{-O(K)}$ requires time $T \geq \Omega(\sqrt{NK})$.

+ Simpler proof than previous work [Klauck et al.'07, Ambainis'10, …]
+ Implies several quantum time-space tradeoffs (e.g. for Sorting).

- Generalization to any product distribution

→ K-Search: find K ones in x where $x_i = 1$ with probability K/N



Record a new 1

Solving the K-Search problem with success probability at least $2^{-O(K)}$ requires time $T \geq \Omega(\sqrt{NK})$.

+ Simpler proof than previous work [Klauck et al.'07, Ambainis'10, …]
+ Implies several quantum time-space tradeoffs (e.g. for Sorting).

- Recent work for non-product distributions [Czajkowski'21, Rosmanis'21]

- Generalization to any product distribution

→ K-Search: find K ones in x where $x_i = 1$ with probability K/N

$$|\perp\rangle|i\rangle \xrightarrow{R} \approx |\perp\rangle - \sqrt{K/N}|1\rangle, \quad |i\rangle$$

Record a new 1

$$|0\rangle|i\rangle \xrightarrow{R} \approx |0\rangle + \sqrt{K/N}|1\rangle, \quad |i\rangle$$

$$|1\rangle|i\rangle \xrightarrow{R} \approx -|1\rangle + \sqrt{K/N}\big(|0\rangle - |\perp\rangle\big), \quad |i\rangle$$

Solving the K-Search problem with success probability at least $2^{-O(K)}$ requires time $T \geq \Omega\big(\sqrt{NK}\big)$.

+ Simpler proof than previous work [Klauck et al.'07, Ambainis'10, …]

+ Implies several quantum time-space tradeoffs (e.g. for Sorting).

- Recent work for non-product distributions [Czajkowski'21, Rosmanis'21]

- Conjecture: $T^2S \geq \Omega(K^2N)$ for finding K collisions.