Advances in quantum algorithms for Hamiltonian simulation

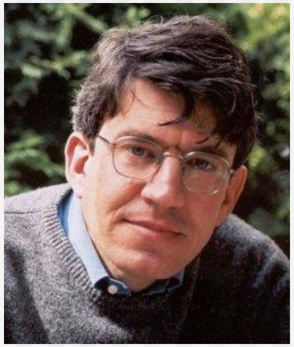
Dominic Berry

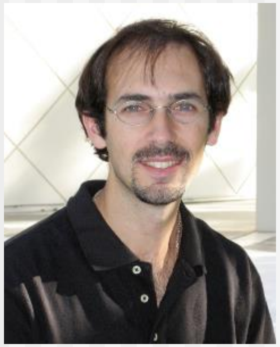
MACQUARIE UNIVERSITY

+

some collaborators:

Alán Aspuru-Guzik & Andrew Childs & Robin Kothari & Richard Cleve & Rolando Somma





HARVARD

UNIVERSITY OF MARYLAND

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

IQC Institute for Quantum Computing

Los Alamos NATIONAL LABORATORY EST. 1943

In the beginning…

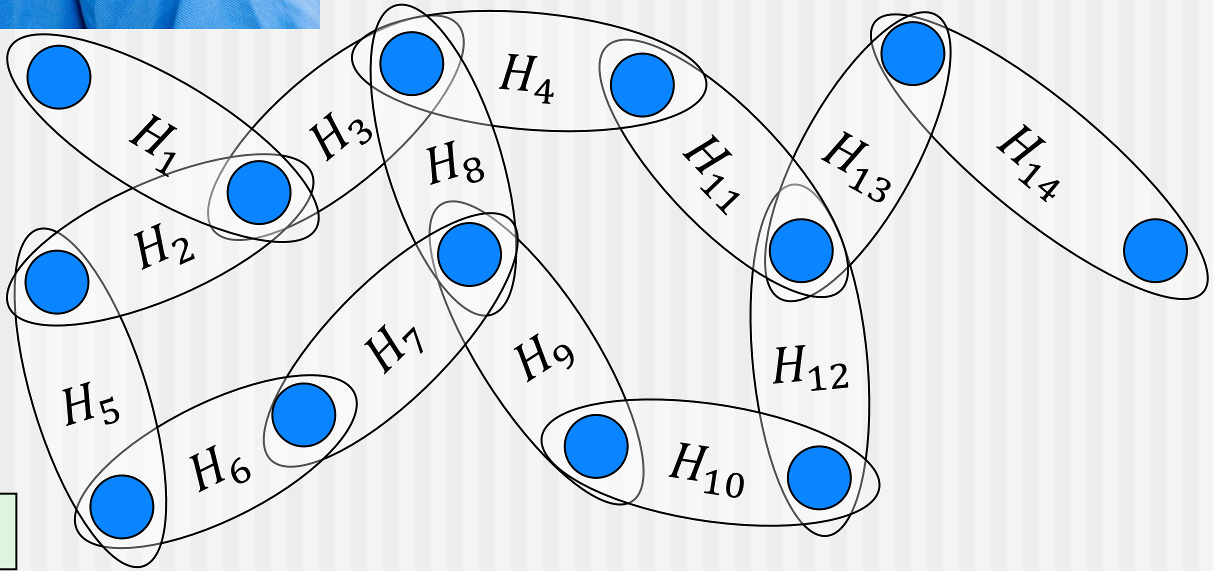
Simulating quantum mechanics is hard – so why not make our computers quantum???

Richard Feynman 1982

R. P. Feynman, Int. J. Theor. Phys. 21 467 (1982).

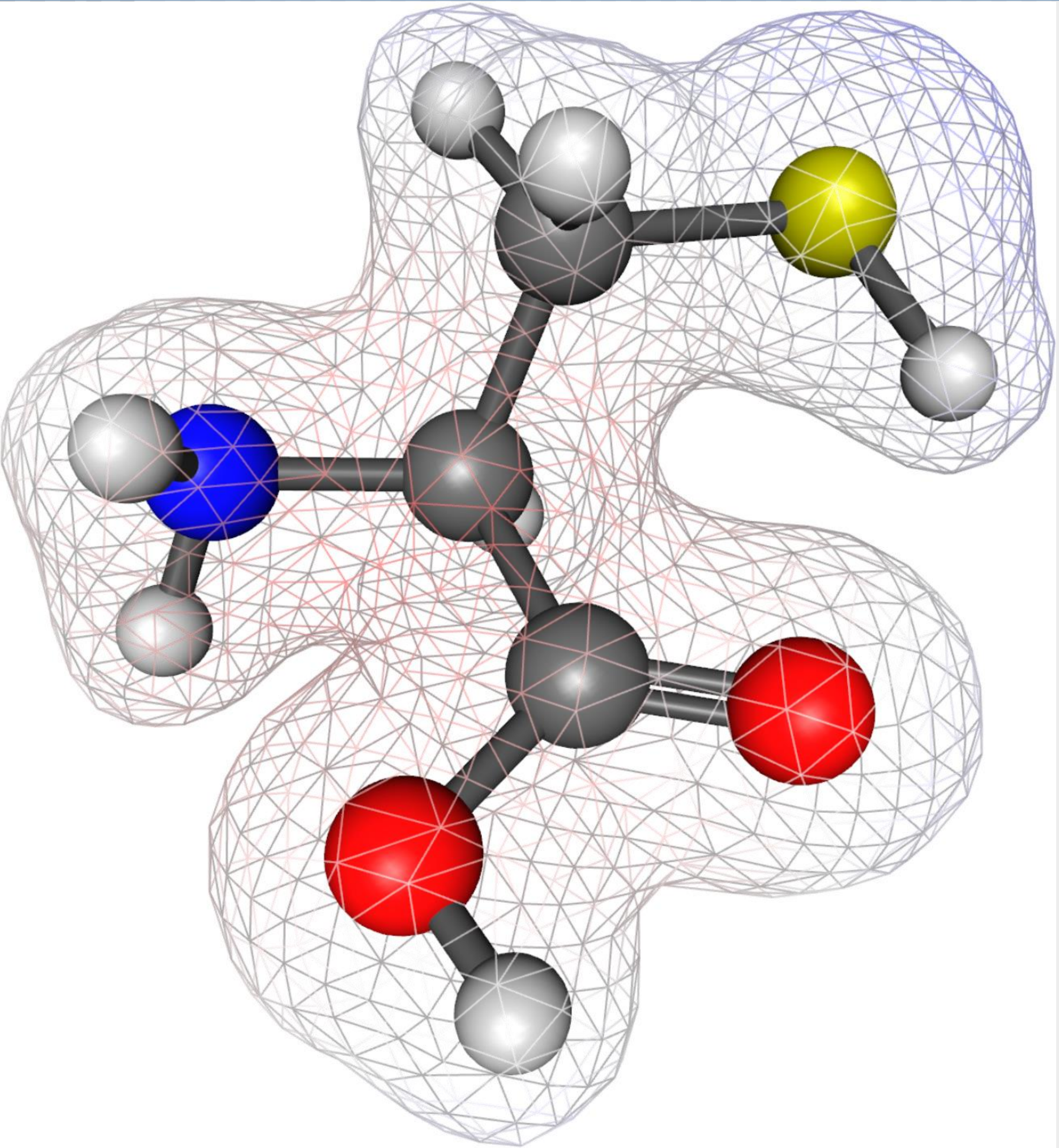
An actual algorithm

Seth Lloyd 1996



S. Lloyd, Science **273**, 1073 (1996).

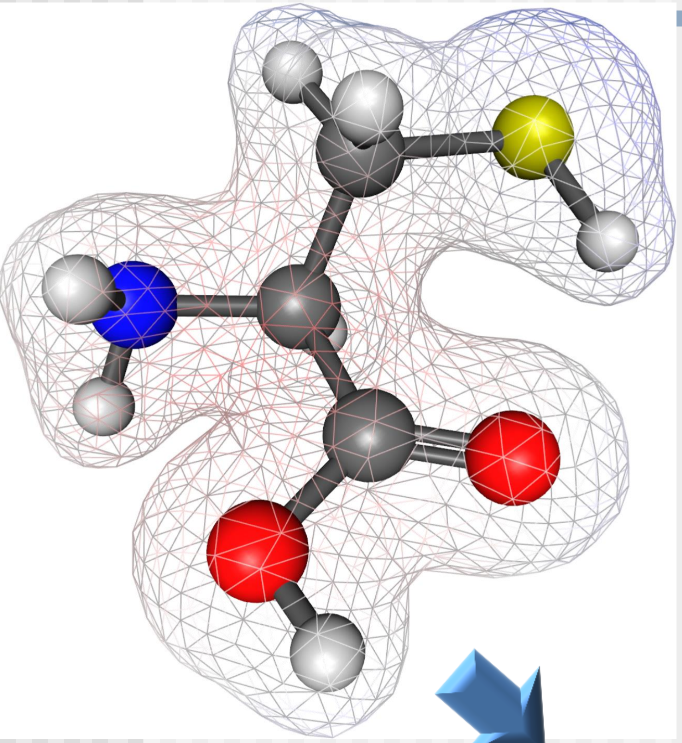
But what do we really want to do?



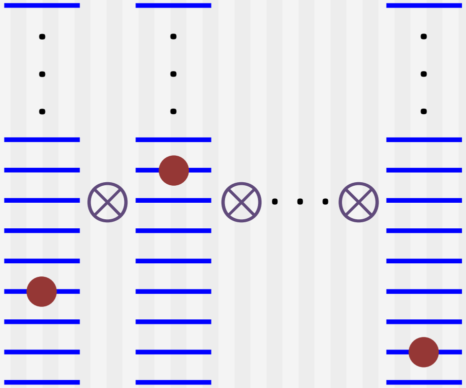
Design molecules, e.g. for

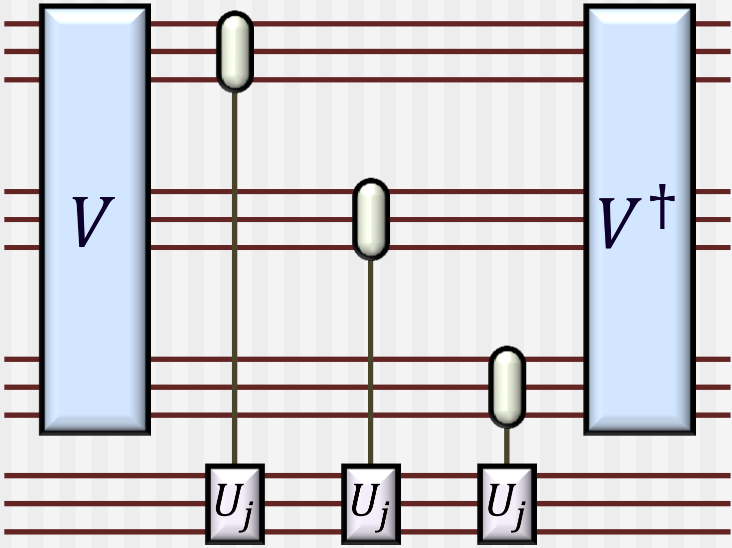
- solar cells
- medicine

Quantum chemistry



Alán Aspuru-Guzik 2005





A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, M. Head-Gordon, Science **309**, 1704 (2005).

Sparse Hamiltonians

$$H = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -\sqrt{3}+i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -\sqrt{3}-i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

Break up Hamiltonian into a sum of 1-sparse terms.

Aharonov & Ta-Shma 2003

arXiv:quant-ph/0301023v2

Product formulae

- Schrödinger's equation:

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle$$

- We want

$$e^{-iHt}$$

- Say $H = A + B$, so

$$e^{-iHt} \approx e^{-iAt}e^{-iBt}$$

- A better approximation:

$$e^{-iHt} \approx e^{-iAt/2}e^{-iBt/2}e^{-iAt/2}e^{-iBt/2}$$

- As many as you like:

$$e^{-iHt} \approx \left(e^{-iAt/r}e^{-iBt/r}\right)^r$$

Will it take too long?

Dave Wecker 2014

- With N orbitals product formulas would need $O(N^9)$ operations.

- Ferredoxin would need 10^{18} gates.

NOOOOOOOOOOOOOO

D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, M. Troyer, Phys. Rev. A **90**, 022305 (2014).

Advanced methods

1. Compressed product formulae
2. Implementing Taylor series
3. Quantum walks
4. Sum of quantum walk steps

1. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC ‘14; arXiv:1312.1414 (2013).
2. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
3. D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
4. D. W. Berry, A. M. Childs, R. Kothari, arXiv:1501.01715 (2015).

The simulation problem

Problem: Given a Hamiltonian H, simulate

$$\frac{d}{dt'}|\psi\rangle = -iH(t')|\psi\rangle$$

for time t and error no more than ε.

Inputs: H, t and ε.

Parameters of H:

- d – sparseness
- N – dimension
- $\|H\|$ – norm of the Hamiltonian
- $\|H'\|$ – norm of the time-derivative

Main result

$$O(\tau \times \mathrm{polylog})$$
$$\tau = d\|H\|_{\max} t$$

Queries:

$$O\left(\tau \frac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}\right)$$

Gates:

$$O\left(\tau \frac{\log^2(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}\right)$$

Comparison to prior work

$$O(\tau \times \mathrm{polylog})$$
$$\tau = d\|H\|_{\max} t$$

1. Lloyd 1996: $\mathrm{poly}(d, \log N) \times \|Ht\|^2/\varepsilon$
2. Aharonov & TaShma 2003: $\mathrm{poly}(d, \log N) \times \|Ht\|^{3/2}/\varepsilon^{1/2}$
3. Berry, Cleve, Ahokas, Sanders 2007: $(d^4\|Ht\| \log^* N)^{1+\delta}(1/\varepsilon)^\delta$
4. Childs & Kothari 2011: $(d^3\|Ht\| \log^* N)^{1+\delta}(1/\varepsilon)^\delta$
5. Berry & Childs 2012: $d\|H\|_{\max} t/\varepsilon^{1/2}$
6. Berry, Childs, Cleve, Kothari, Somma 2013: $d^2\|H\|_{\max} t \times \mathrm{polylog}$

Comparison to lower bound

Upper bound:

$$O(\tau \times \text{polylog})$$
$$\tau = d\|H\|_{\max} t$$

Lower bound:

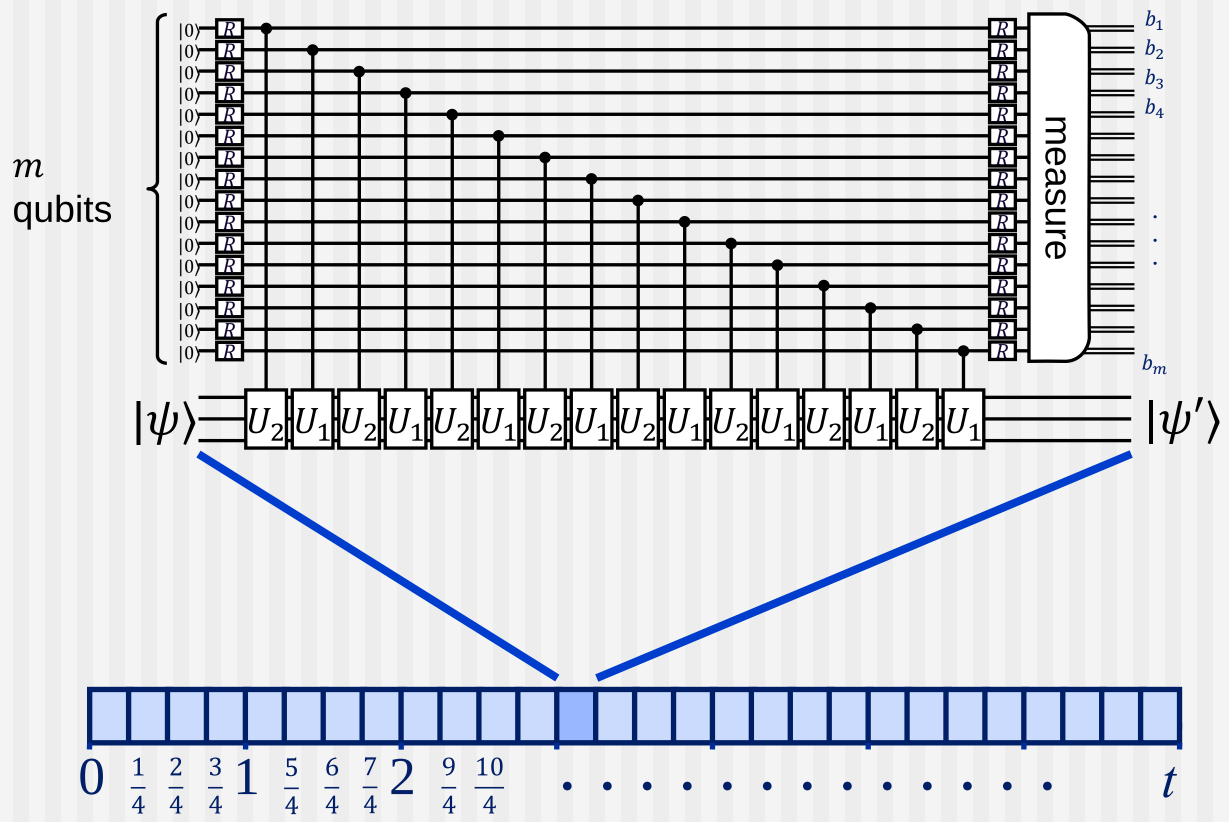
$$O(\tau + \text{polylog})$$
$$\tau = d\|H\|_{\max} t$$

Compressed product formulae

1. Decompose Hamiltonian into a sum of self-inverse Hamiltonians.

2. Approximate Hamiltonian evolution by Lie-Trotter formula, then compress it.

3. Use oblivious amplitude amplification.



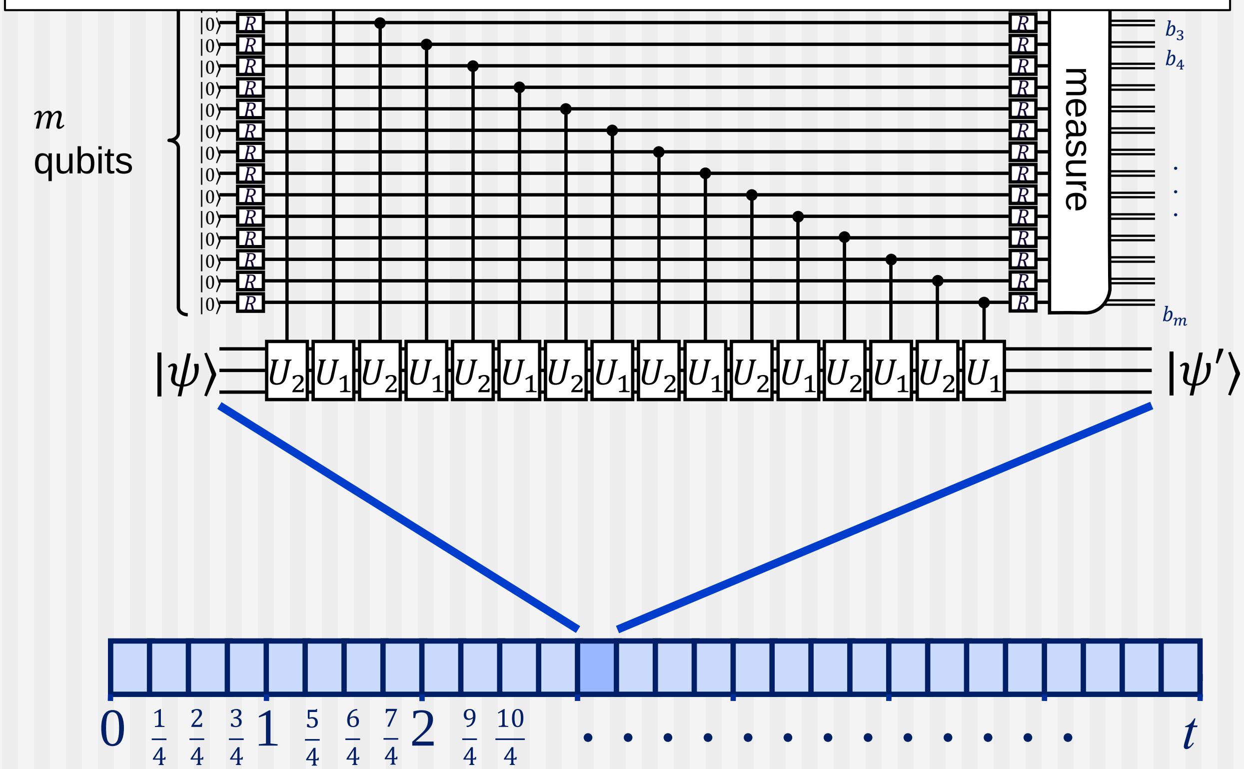
Compressed product formulae

1. Decompose Hamiltonian into a sum of self-inverse Hamiltonians.

2. Approximate Hami
formula, then comp

3. Use oblivious amplitude amplification.

1. Decompose Hamiltonian into 1-sparse.
2. Break 1-sparse into X, Y, Z parts.
3. Break X, Y, Z parts into self-inverse.



Decompose Hamiltonian to 1-sparse

- Decompose Hamiltonian into H_1 and H_2:

$$H = \begin{pmatrix} 0 & 0 & \boxed{2} & 0 & 0 & \boxed{\sqrt{2}i} & \cdots & 0 \\ 0 & \boxed{3} & 0 & 0 & 0 & \boxed{1/2} & \cdots & 0 \\ \boxed{2} & 0 & 0 & 0 & 0 & 0 & \cdots & \boxed{-\sqrt{3}+i} \\ 0 & 0 & 0 & \boxed{1} & \boxed{e^{i\pi/7}} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \boxed{e^{-i\pi/7}} & \boxed{2} & 0 & \cdots & 0 \\ \boxed{-\sqrt{2}i} & \boxed{1/2} & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \boxed{-\sqrt{3}-i} & 0 & 0 & 0 & \cdots & \boxed{1/10} \end{pmatrix}$$

- No more than d nonzero elements in any row or column.
- In general can decompose into d^2 parts.

Decompose Hamiltonian to 1-sparse

- Decompose Hamiltonian into H_1 and H_2:

$$H_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \boxed{\sqrt{2}i} & \cdots & 0 \\ 0 & \boxed{3} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \boxed{-\sqrt{3}+i} \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \boxed{2} & 0 & \cdots & 0 \\ \boxed{-\sqrt{2}i} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \boxed{-\sqrt{3}-i} & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

- No more than d nonzero elements in any row or column.
- In general can decompose into d^2 parts.

Decompose 1-sparse to X, Y, Z

- Break into X, Y and Z components:

off-diagonal real

$$H_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \boxed{\sqrt{2}i} & \cdots & 0 \\ 0 & \boxed{3} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \boxed{-\sqrt{3}}\boxed{+i} \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \boxed{2} & 0 & \cdots & 0 \\ \boxed{-\sqrt{2}i} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \boxed{-\sqrt{3}}\boxed{-i} & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

off-diagonal imaginary

on-diagonal real

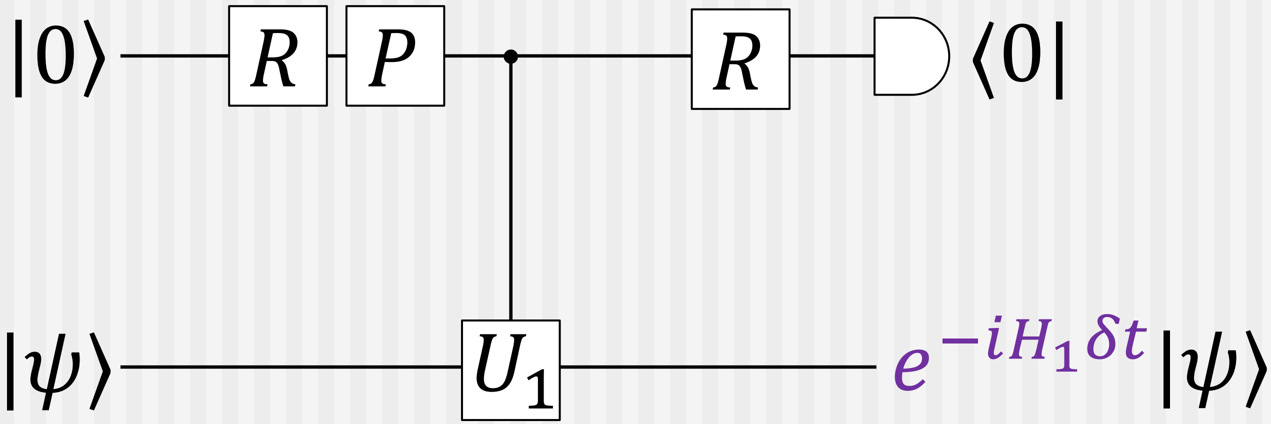
+ break into γ-size pieces to get self-inverse

Net result

$$H = \gamma \sum_{j=1}^{M} U_j$$

Evolution using control qubits

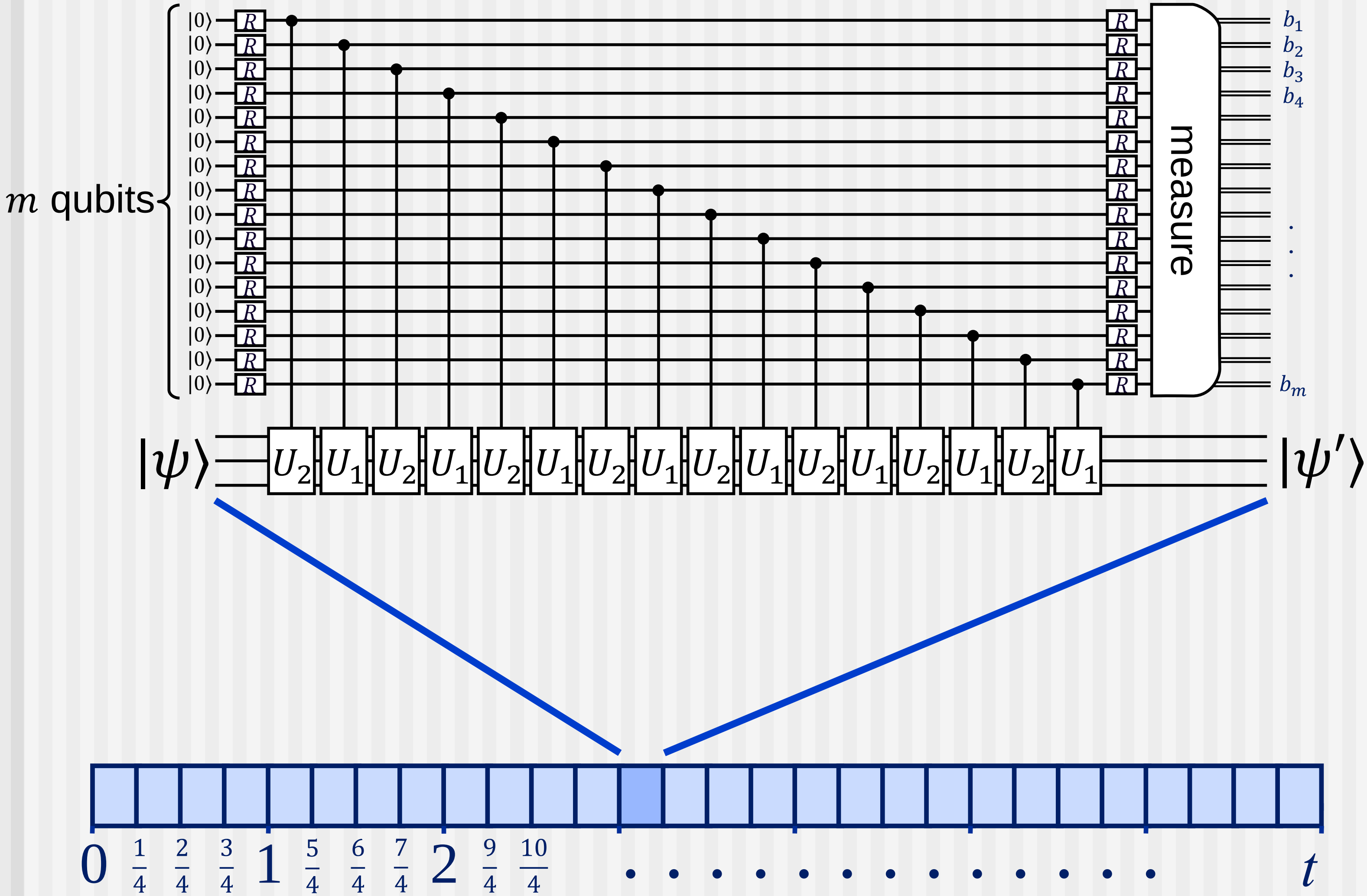
- $H_1 = \gamma U_1$
- U_1 is self-inverse



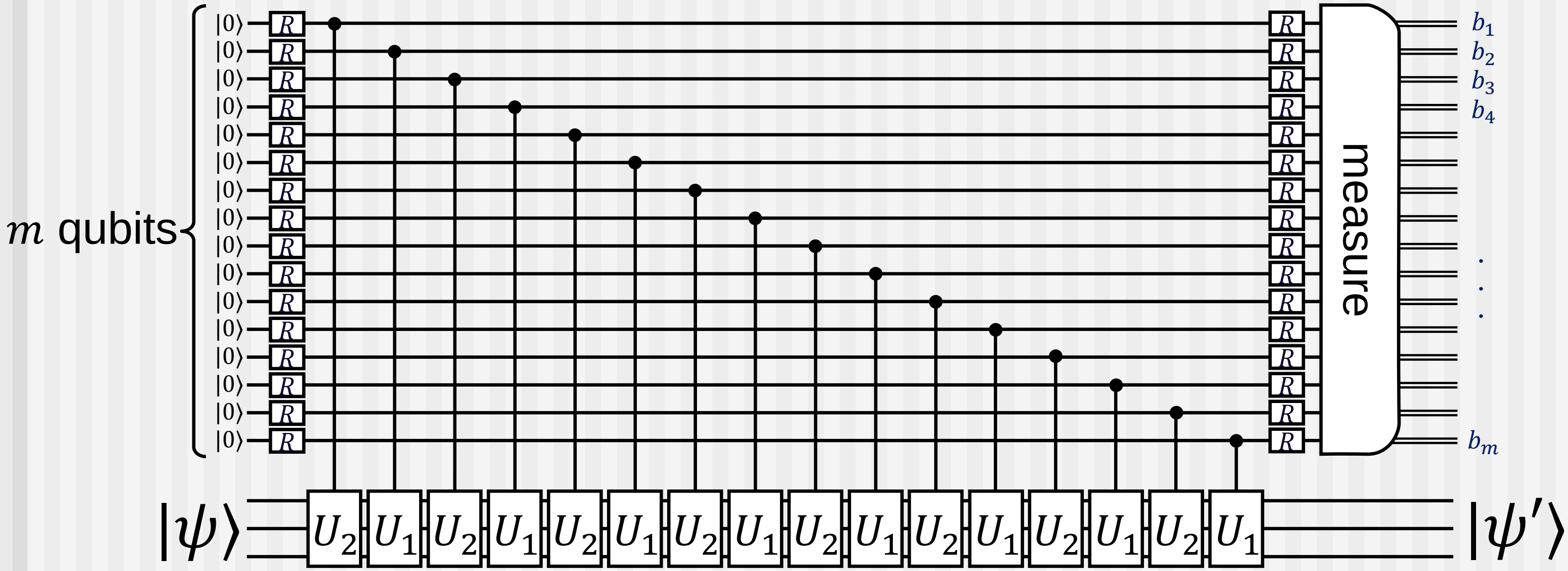
$$R = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$$

R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).

Simulation of segments

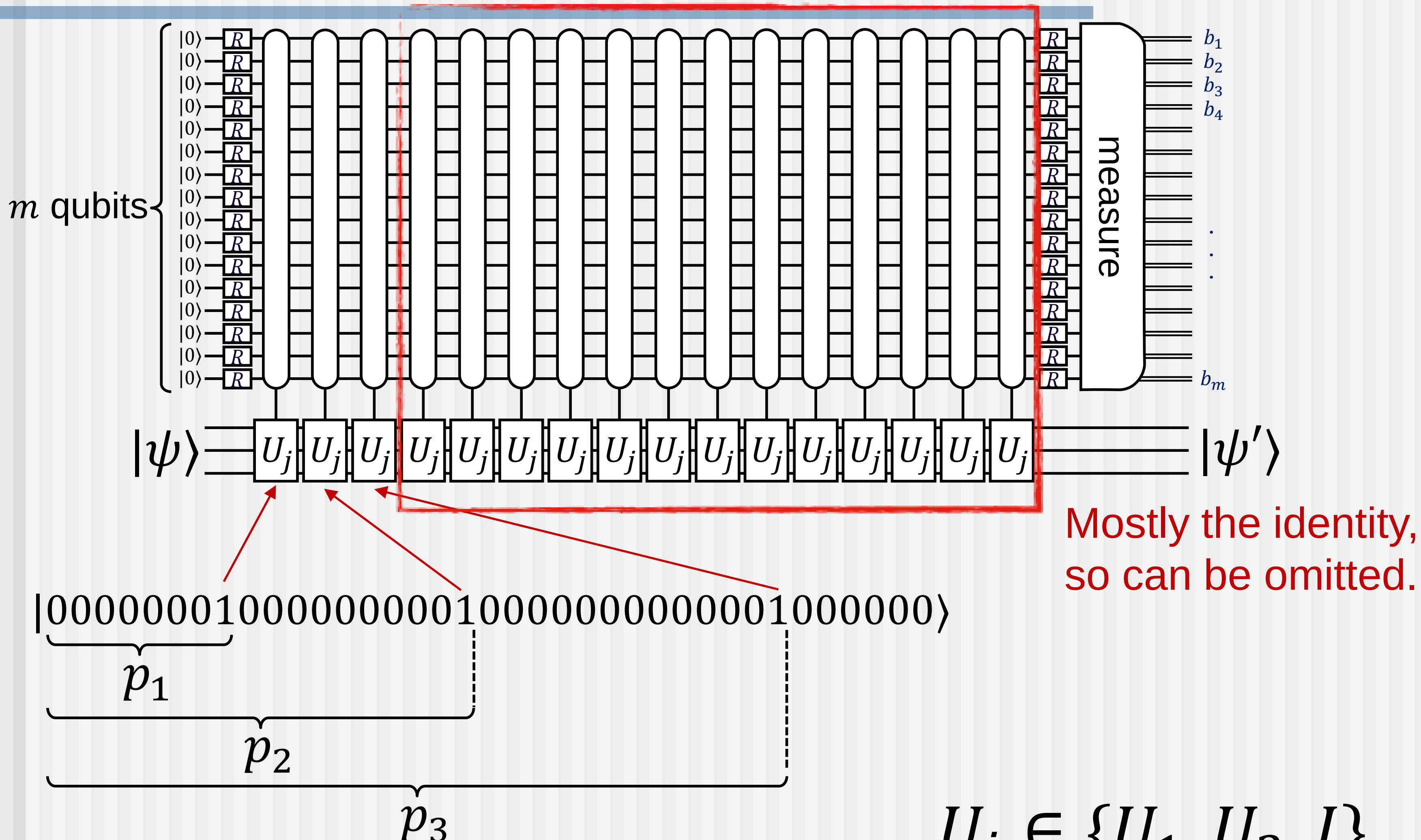


Simulation of segments



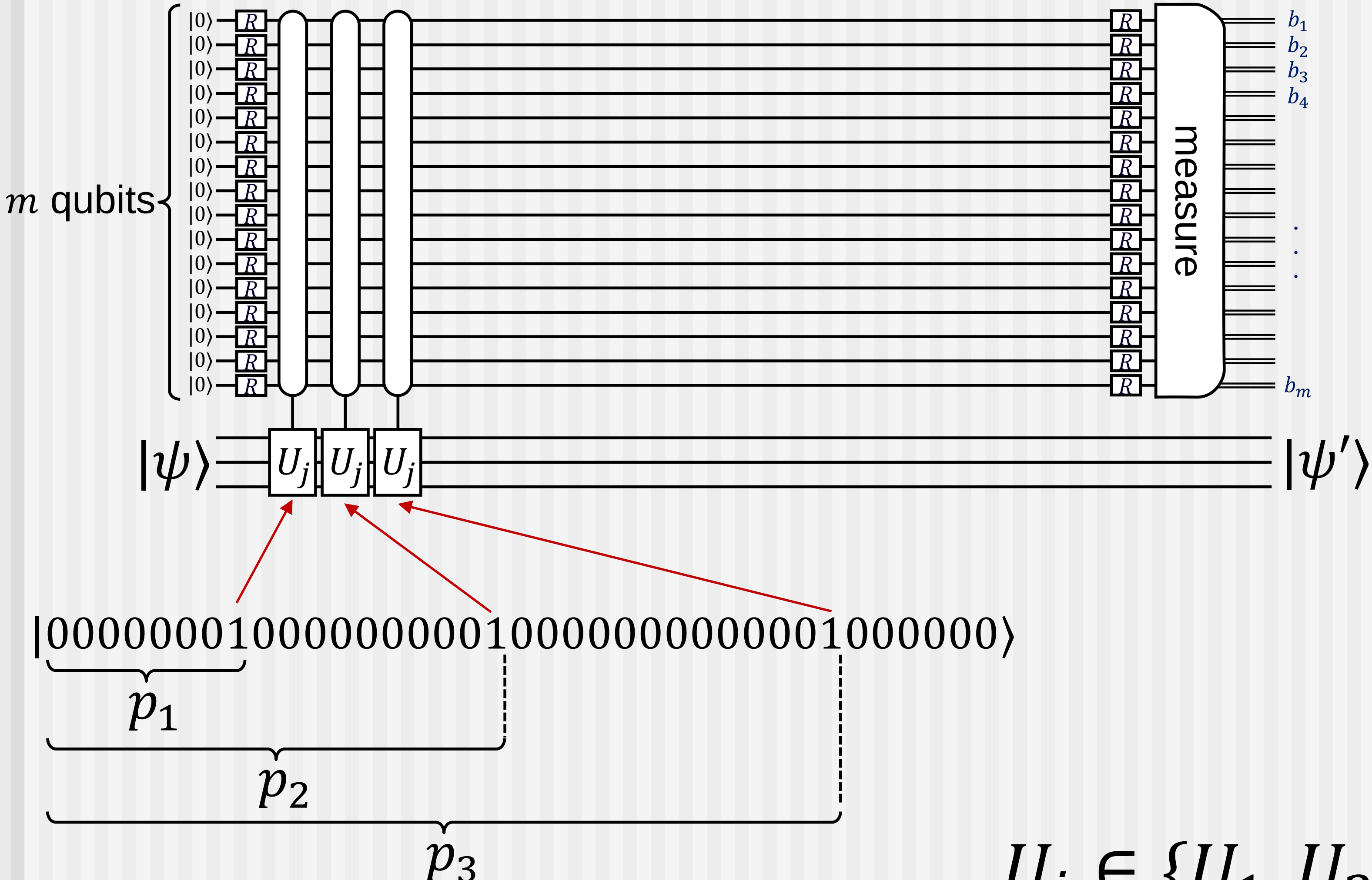
$$\left|\underbrace{\underbrace{\underbrace{00000001}_{p_1}0000000001}_{p_2}0000000000001}_{p_3}000000\right\rangle$$

Simulation of segments



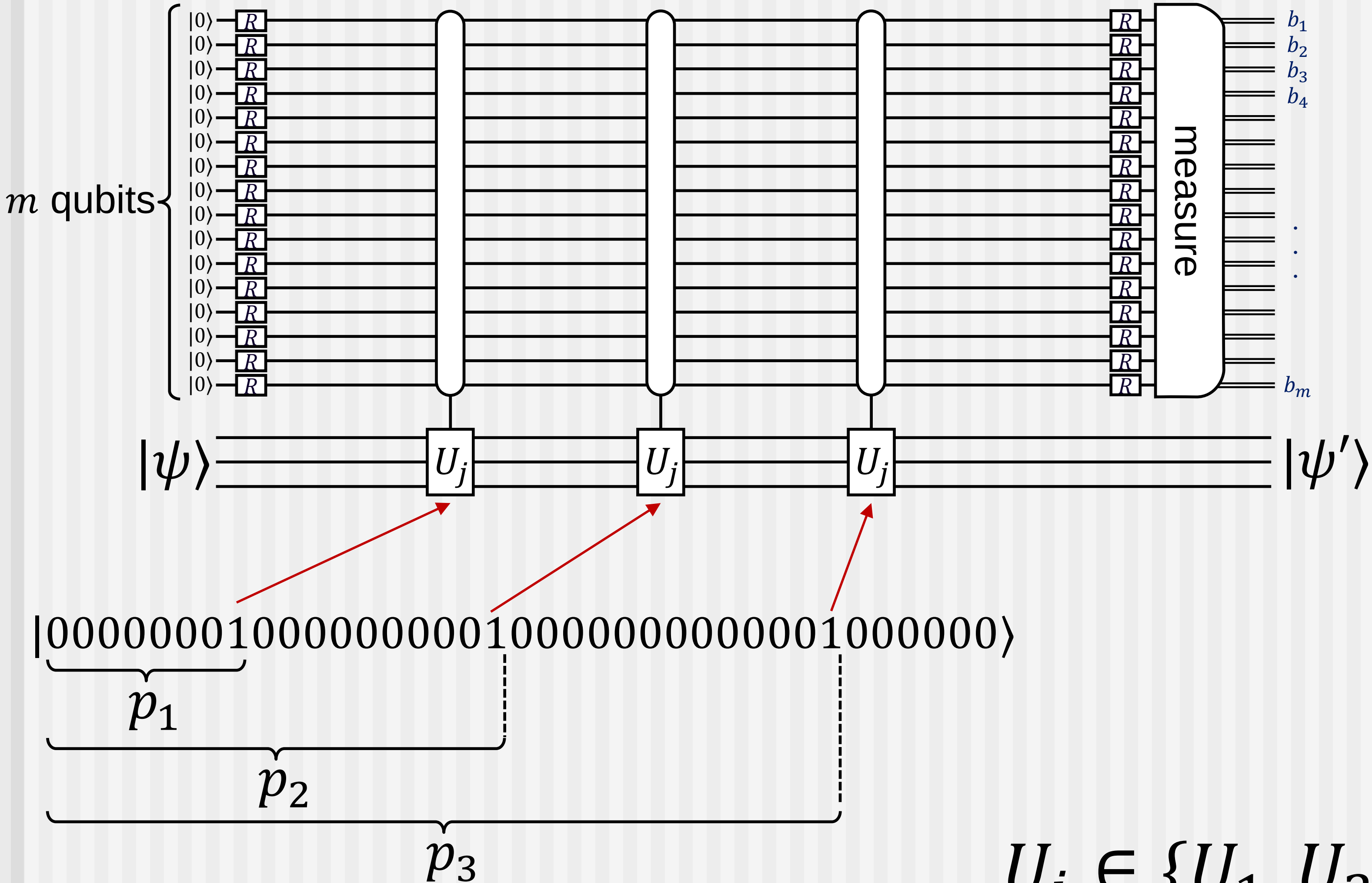
$$U_j \in \{U_1, U_2, I\}$$

Simulation of segments



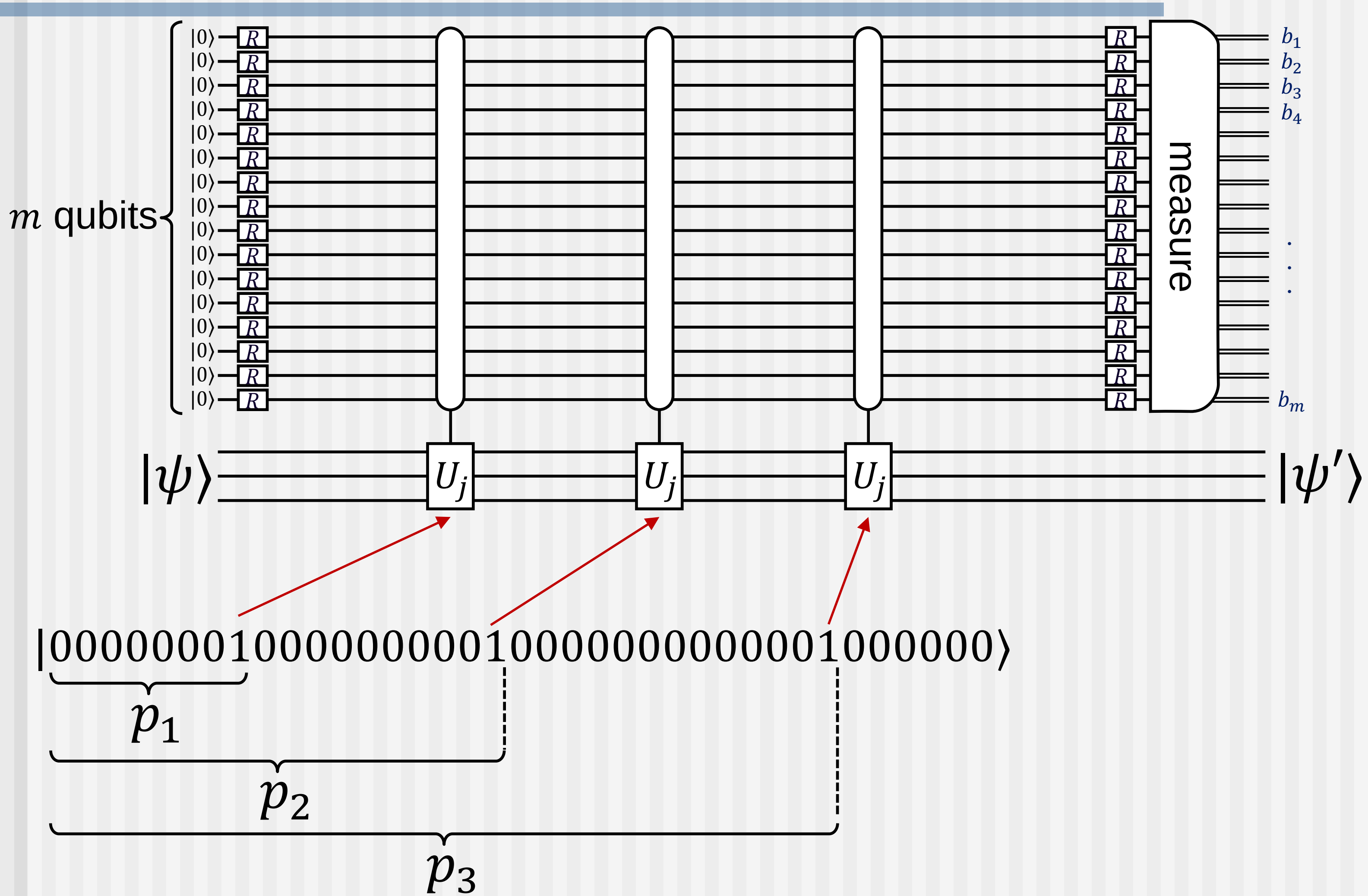
$$U_j \in \{U_1, U_2, I\}$$

Simulation of segments



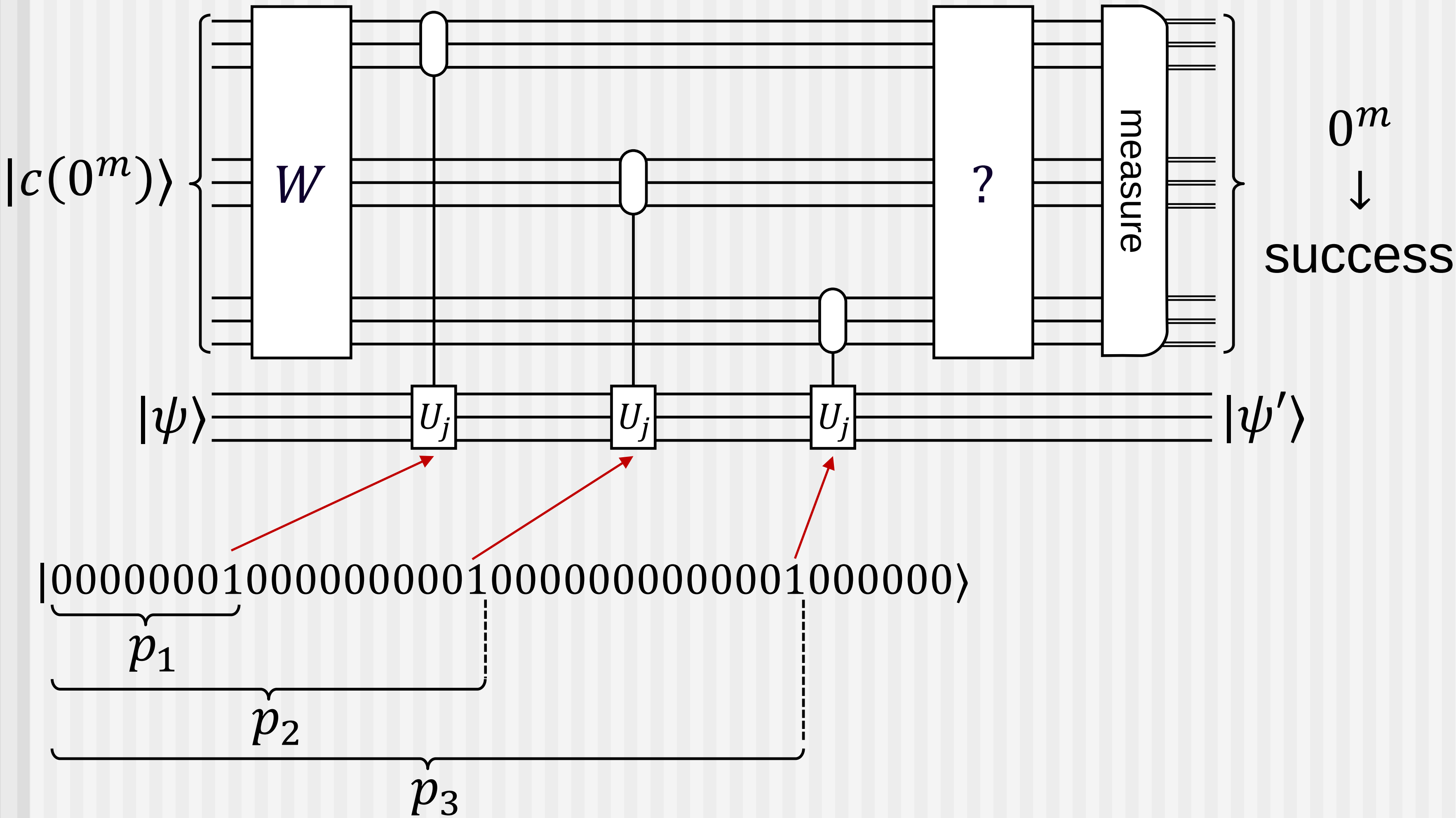
$$U_j \in \{U_1, U_2, I\}$$

Compression of control qubits



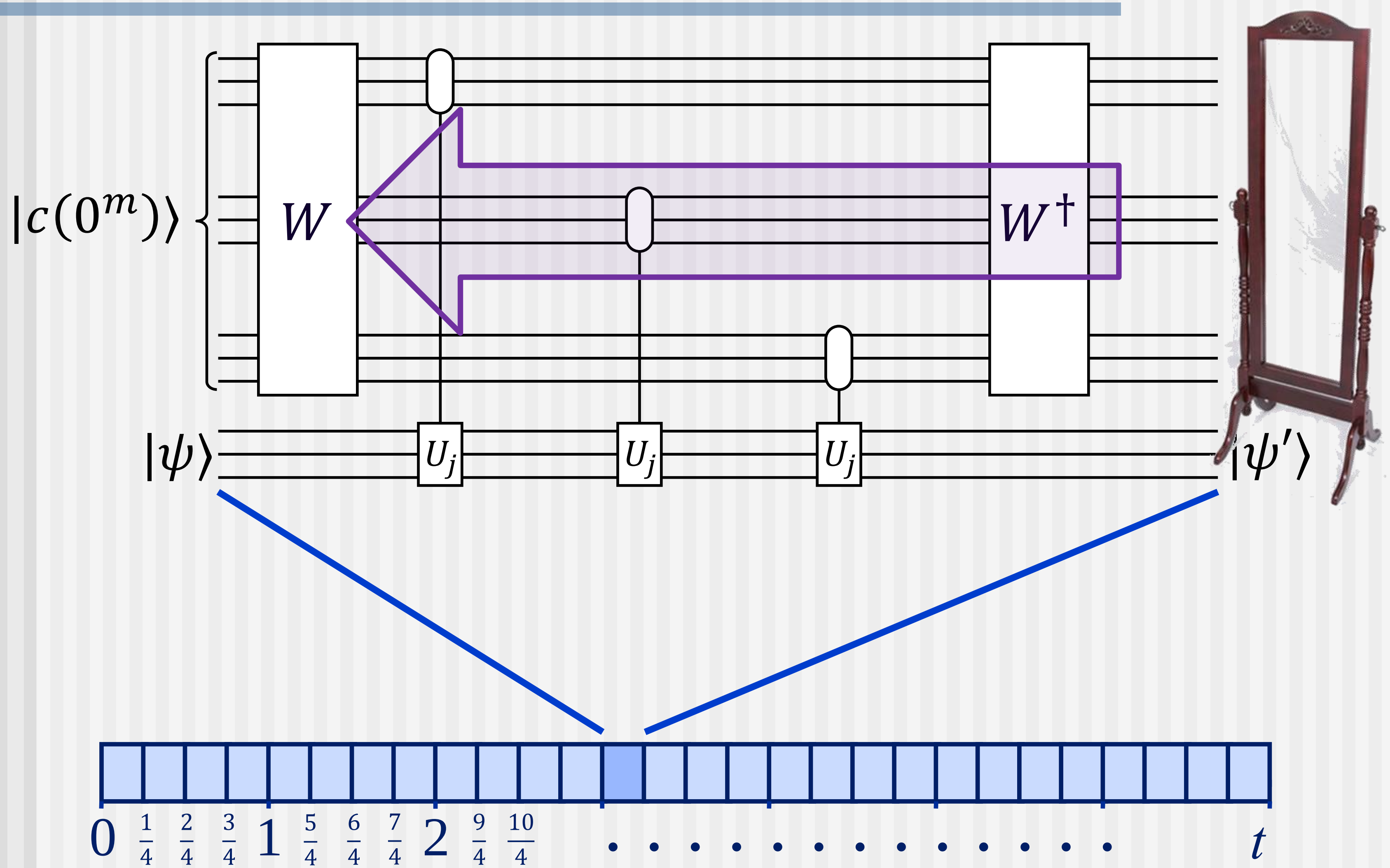
Compressed form: $|c(x)\rangle = |p_1\rangle \dots |p_{|x|}\rangle |m\rangle^{k-|x|}$

Compression of control qubits

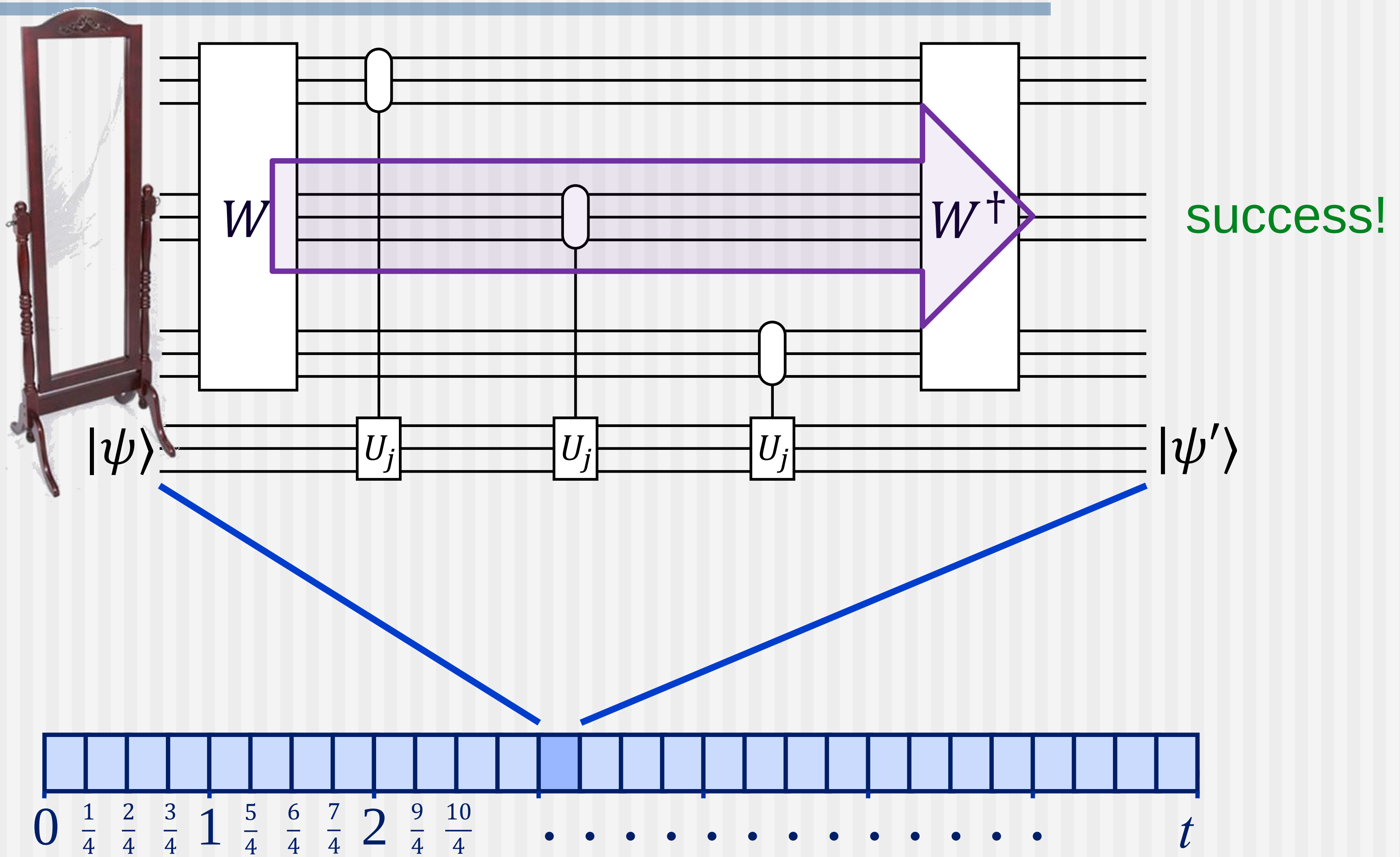


Compressed form: $|c(x)\rangle = |p_1\rangle \ldots |p_{|x|}\rangle |m\rangle^{k-|x|}$

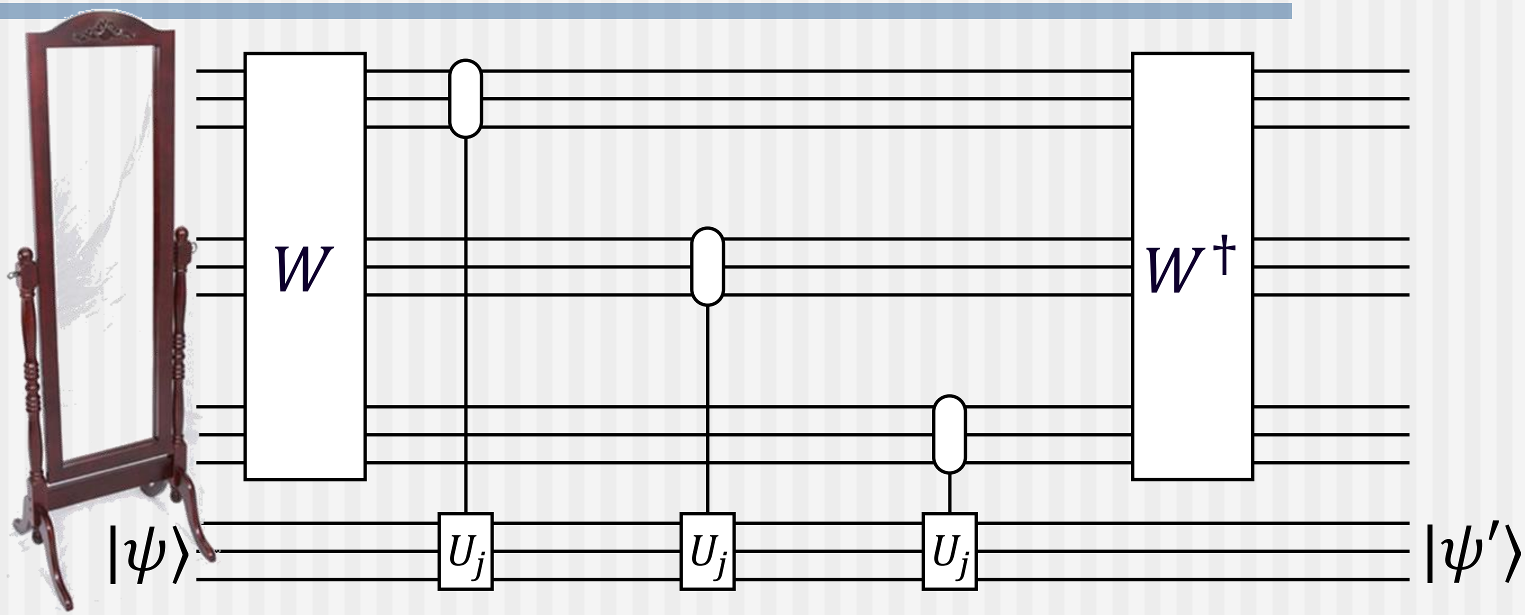
Oblivious amplitude amplification



Oblivious amplitude amplification



Oblivious amplitude amplification



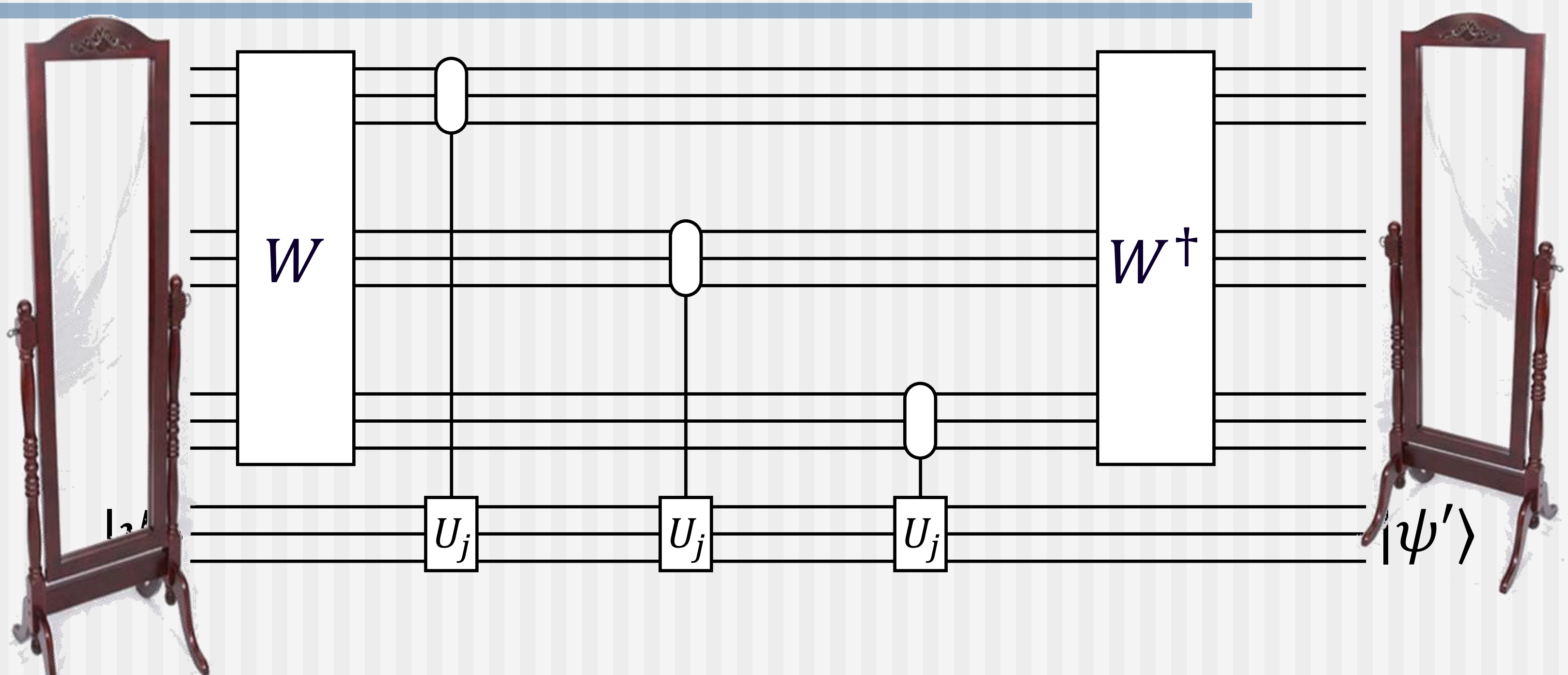
$$U|0\rangle|\psi\rangle = \sqrt{p}|0\rangle V|\psi\rangle + \sqrt{1-p}|1\rangle|\phi\rangle$$

Operation we know how to perform

Operation we want to perform

- Standard amplitude amplification: Need to reflect about $U|0\rangle|\psi\rangle$.

Oblivious amplitude amplification



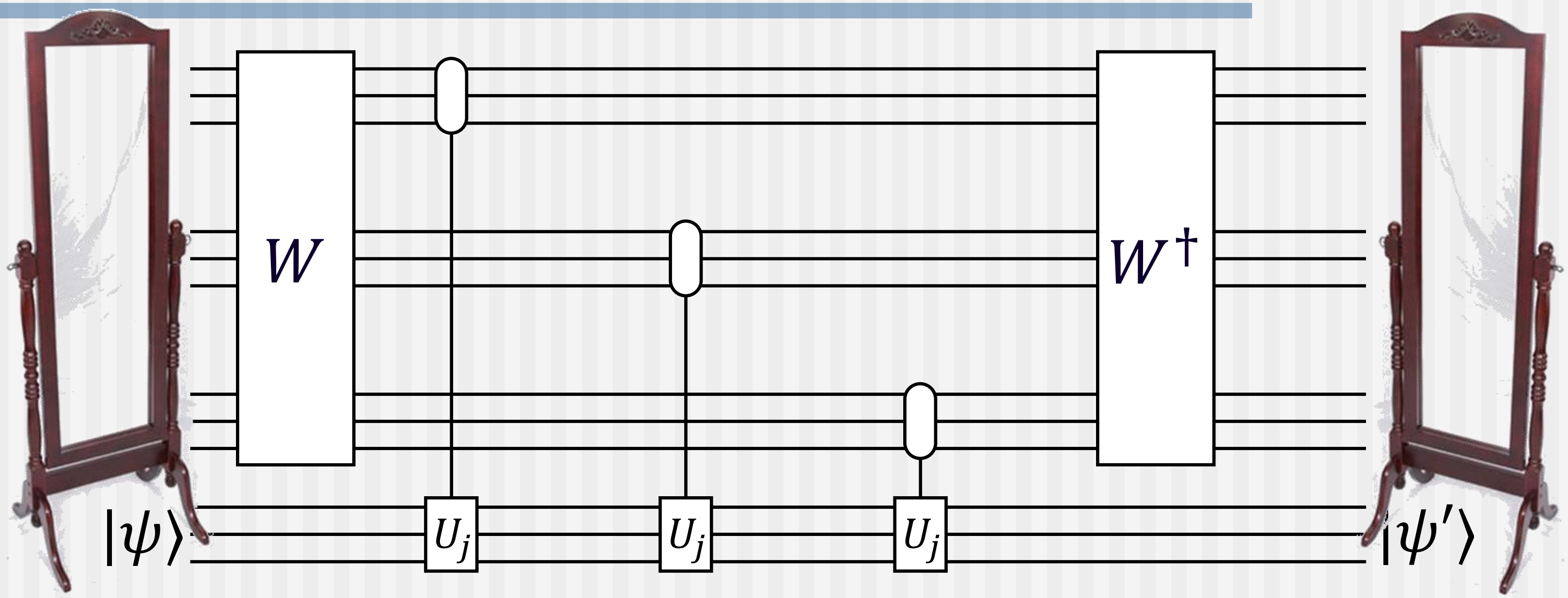
$$U|0\rangle|\psi\rangle = \sqrt{p}|0\rangle V|\psi\rangle + \sqrt{1-p}|1\rangle|\phi\rangle$$

Operation we know how to perform

Operation we want to perform

- Standard amplitude amplification: Need to reflect about $U|0\rangle|\psi\rangle$.

Oblivious amplitude amplification



$$U|0\rangle|\psi\rangle = \sqrt{p}|0\rangle V|\psi\rangle + \sqrt{1-p}|1\rangle|\phi\rangle$$

Operation we know how to perform

Operation we want to perform

- Oblivious amplitude amplification: Only do reflections on first register.

Advanced methods

1. Compressed product formulae ✓
2. Implementing Taylor series
3. Quantum walks
4. Sum of quantum walk steps

1. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC ‘14; arXiv:1312.1414 (2013).
2. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
3. D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
4. D. W. Berry, A. M. Childs, R. Kothari, arXiv:1501.01715 (2015).

Implementing Taylor series

- The Hamiltonian evolution can be expanded in Taylor series:

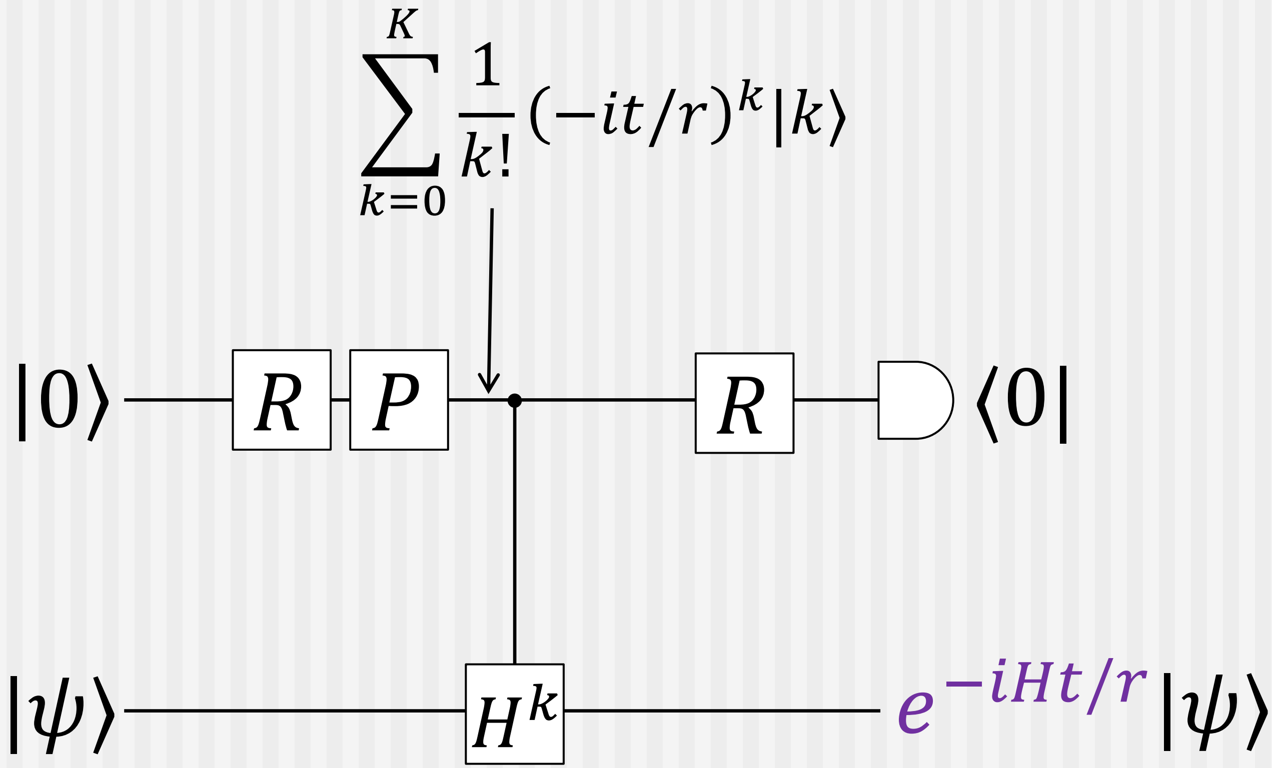
$$U = \exp(-iHt) = \sum_{k=0}^{\infty} \frac{1}{k!} (-iHt)^k$$

- For r segments, we would want

$$U_r = \exp(-iHt/r) \approx \sum_{k=0}^{K} \frac{1}{k!} (-iHt/r)^k$$

Implementing Taylor series

- If H is unitary, can probabilistically implement using controlled operation.



Implementing Taylor series

- In reality H is (approximately) a sum of unitaries

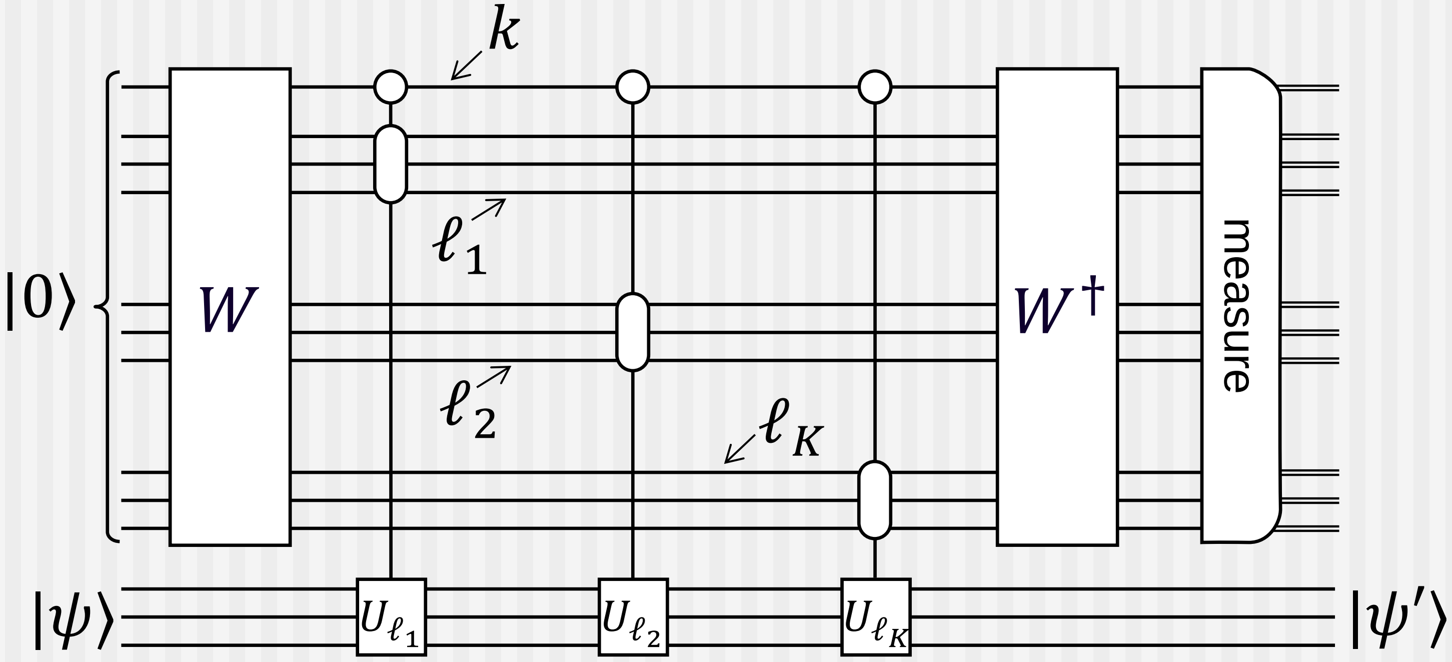
$$H \approx \gamma \sum_{\ell=1}^{M} U_\ell$$

- Exponential is then

$$\exp(-iHt/r) \approx \sum_{k}^{K} \sum_{\ell_1=1}^{M} \sum_{\ell_2=1}^{M} \cdots \sum_{\ell_k=1}^{M} \frac{(-it/r)^k}{k!} U_{\ell_1} U_{\ell_2} \cdots U_{\ell_k}$$

- We can again implement using controlled operations.

Implementing a Taylor series



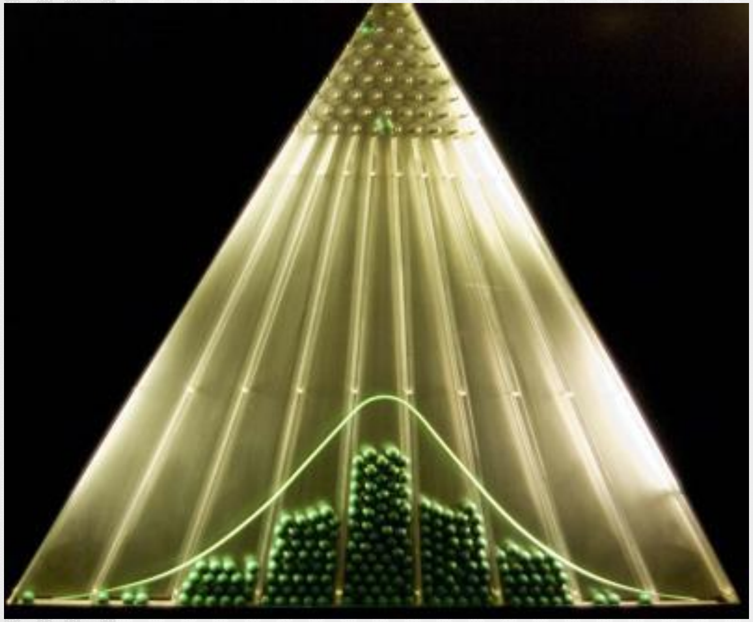
- A measurement result of 0 corresponds to success.
- This can be performed deterministically using oblivious amplitude amplification.

Advanced methods

1. Compressed product formulae ✓
2. Implementing Taylor series ✓
3. Quantum walks
4. Sum of quantum walk steps

1. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC '14; arXiv:1312.1414 (2013).
2. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
3. D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
4. D. W. Berry, A. M. Childs, R. Kothari, arXiv:1501.01715 (2015).

Quantum walks



- Classical walk: position x jumps either to the left or the right at each step.

- Quantum walk has position and coin values

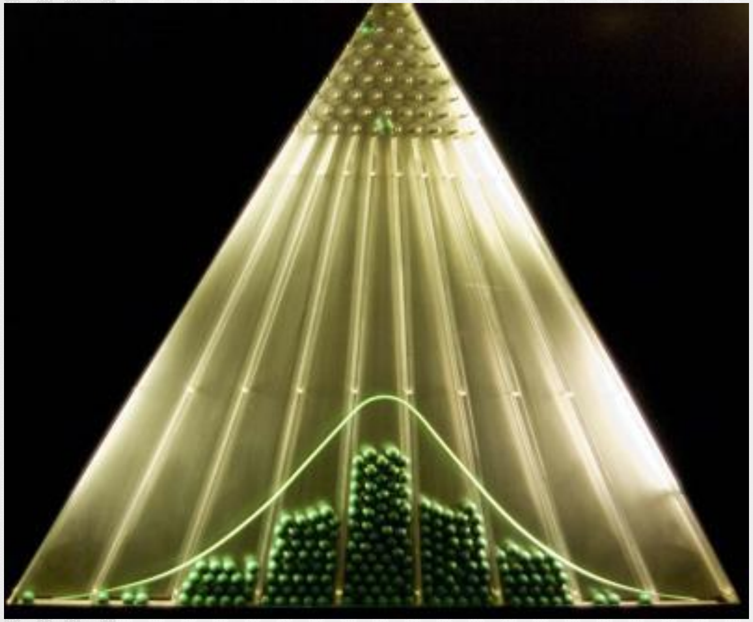
$$|x, c\rangle$$

- It then alternates coin and step operators,

$$C|x, \pm 1\rangle = (|x, -1\rangle \pm |x, 1\rangle)/\sqrt{2}$$
$$S|x, c\rangle = |x + c, c\rangle$$

- The position can progress linearly in the number of steps.

Quantum walks



- Classical walk: position x jumps either to the left or the right at each step.

- Quantum walk has position and coin values
$$|x, c\rangle$$
- It then alternates coin and step operators,
$$C|x, \pm 1\rangle = (|x, -1\rangle \pm |x, 1\rangle)/\sqrt{2}$$
$$S|x, c\rangle = |x + c, c\rangle$$

- The position can progress linearly in the number of steps.

- Szegedy quantum walk allows arbitrary dimensions, n and m on the two subsystems.

- Szegedy quantum walk uses more general controlled “diffusion” operators.

Szegedy quantum walk

- The "diffusion" operators are of the form

$$2CC^\dagger - \mathbb{I}$$
$$2RR^\dagger - \mathbb{I}$$

- C is controlled by the first register and acts on the second register.

- The operator C is a controlled reflection.

$$C = \sum_{i=1}^{n} |i\rangle\langle i| \otimes |c_i\rangle$$
$$|c_i\rangle = \sum_{j=1}^{m} \sqrt{c[i,j]}\,|j\rangle$$

- The diffusion operator $2RR^\dagger - \mathbb{I}$ is controlled by the second register and acts on the first.

M. Szegedy, quant-ph/0401053 (2004).

Szegedy walk for Hamiltonians

- Use symmetric system, with $n = m$ and
$$c[i,j] = r[i,j] = H_{ij}^*$$
- The step of the quantum walk is (S is swap)
$$V = iS(2CC^\dagger - \mathbb{I})$$
- Eigenvalues and eigenvectors are related to those of Hamiltonian.
- We need to modify to "lazy" quantum walk, with

$$|c_i\rangle = \sqrt{\frac{\delta}{\|H\|_1}} \sum_{j=1}^{N} \sqrt{H_{ij}^*}\,|j\rangle + \sqrt{1 - \frac{\sigma_i \delta}{\|H\|_1}}\,|N+1\rangle \qquad \sigma_i := \sum_{j=1}^{N} |H_{ij}|$$

extra component

A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

State preparation

- Grover state preparation starts from

$$\frac{1}{\sqrt{N}}\sum_{k=1}^{N}|k\rangle|0\rangle$$

- Rotate ancilla according to amplitude for state to be prepared

$$|\psi^b\rangle = \frac{1}{\sqrt{N}}\sum_{k=1}^{N}|k\rangle\left(\psi_k|0\rangle + \sqrt{1-|\psi_k|^2}|1\rangle\right)$$

- Amplitude amplification yields component where ancilla is zero.
- In comparison, state we wish to prepare is

$$|c_i\rangle = \sqrt{\frac{\delta}{\|H\|_1}}\sum_{j=1}^{N}\sqrt{H_{ij}^*}|j\rangle + \sqrt{1-\frac{\sigma_i\delta}{\|H\|_1}}|N+1\rangle$$

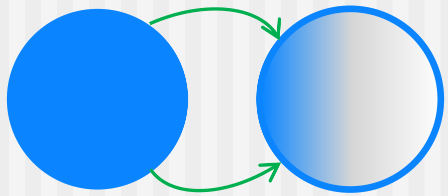
- We can just use one iteration!

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
L. K. Grover, PRL **85**, 1334 (2000).

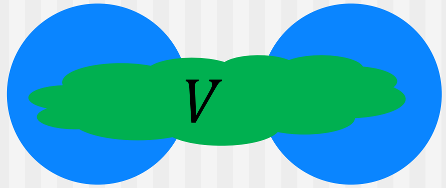
Szegedy walk for Hamiltonians

Three step process:

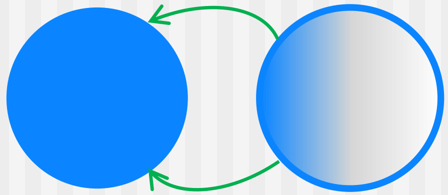
1. Start with state in one of the subsystems, and perform controlled state preparation.



2. Perform steps of quantum walk to approximate Hamiltonian evolution.



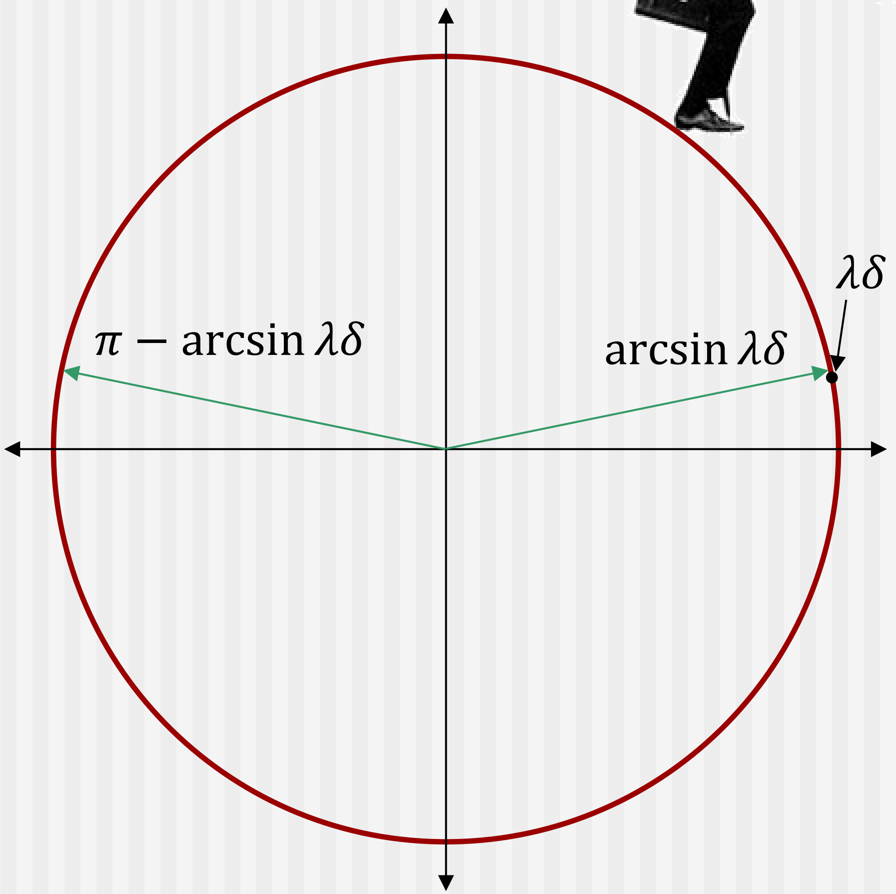
3. Invert controlled state preparation, so final state is in one of the subsystems.



A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

Szegedy walk for Hamiltonians

- A Hamiltonian H has eigenvalues λ.
- V is the step of a quantum walk, and has eigenvalues
$$\mu_{\pm} = \pm e^{\pm i \arcsin \lambda\delta}$$
- We aim to achieve evolution under the Hamiltonian. It has eigenvalues
$$e^{-i\lambda t}$$



A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

Advanced methods

1. Compressed product formulae ✓
2. Implementing Taylor series ✓
3. Quantum walks ✓
4. Sum of quantum walk steps

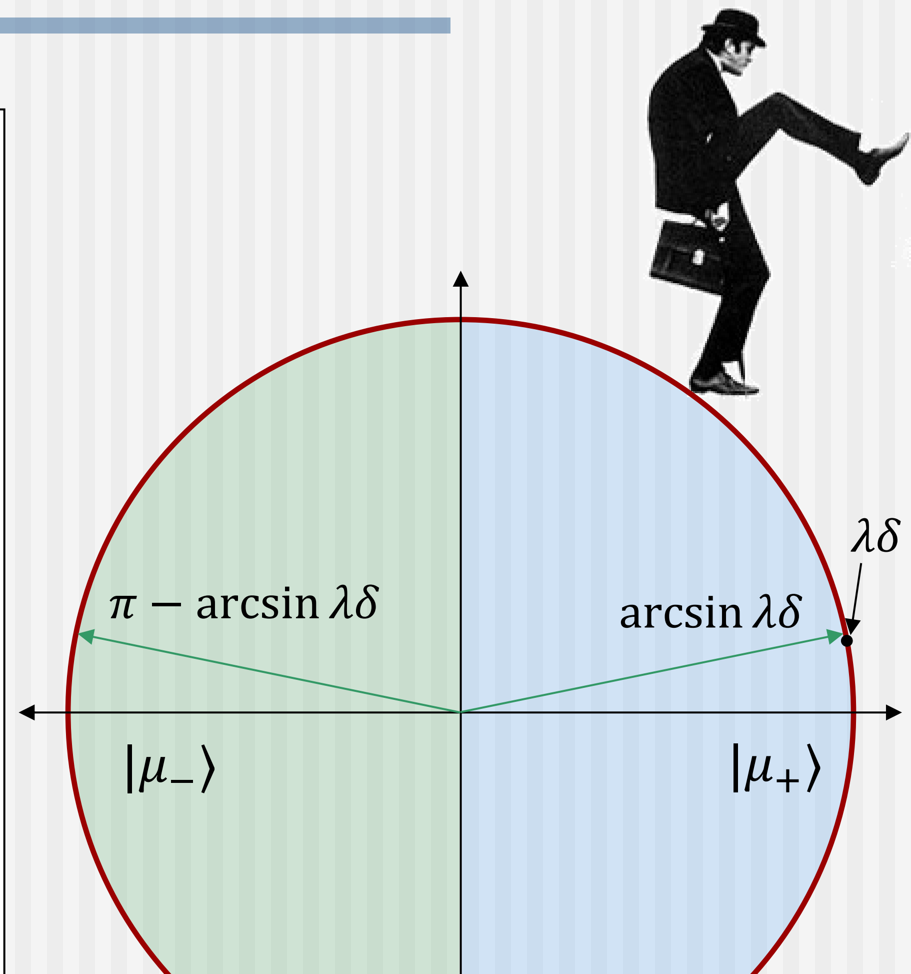
1. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC ‘14; arXiv:1312.1414 (2013).
2. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
3. D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
4. D. W. Berry, A. M. Childs, R. Kothari, arXiv:1501.01715 (2015).

Superposition of quantum walk

- A Hamiltonian H has eigenvalues λ.

- V is the step of a quantum walk, and has eigenvalues
$$\mu_\pm = \pm e^{\pm i \arcsin \lambda\delta}$$

- We aim to achieve evolution under the Hamiltonian. It has eigenvalues
$$e^{-i\lambda t}$$



- Corrected step V_c has eigenvalues
$$\mu = e^{-i \arcsin \lambda\delta}$$

Superposition of quantum walk

- We have

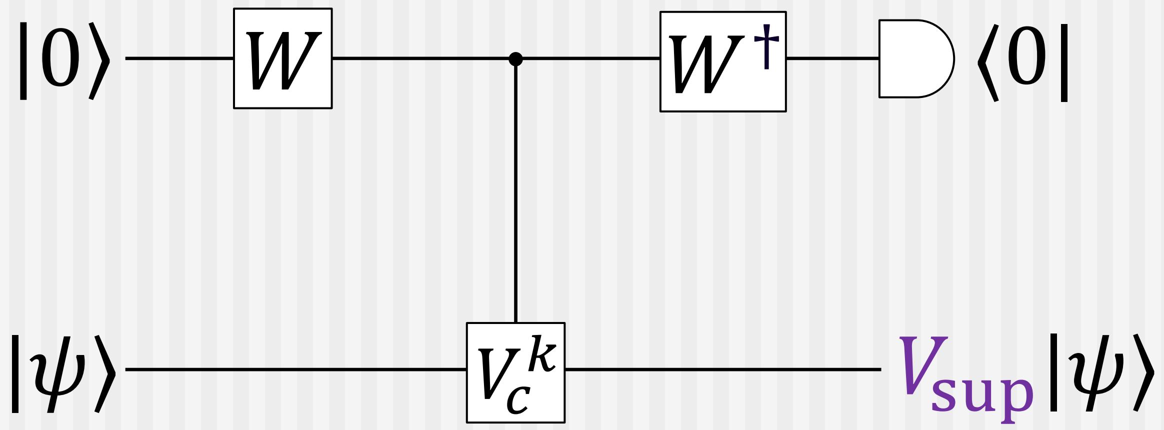
$$\mu = e^{-i \arcsin \lambda \delta}$$

- We aim for

$$e^{-i\lambda t}$$

- Try superposition of operations

$$V_{\mathrm{sup}} = \sum_{k=0}^{K} \alpha_k V_c^k$$



Solving for α_k

- We have

$$\mu = e^{-i \arcsin \lambda\delta}$$

- We aim for

$$e^{-i\lambda t}$$

- Try superposition of operations

$$V_{\text{sup}} = \sum_{k=0}^{K} \alpha_k V_c^k$$

- The eigenvalues of V_{sup} are

$$\mu_{\text{sup}} = \sum_{k=0}^{K} \alpha_k \mu^k$$

- We can solve for α_k such that $\mu_{\text{sup}} = e^{-it\lambda} + O((t\lambda)^{K+1})$

- Symmetry is better:

$$\mu_{\text{sup}} = \sum_{k=-K}^{K} \alpha_k \mu^k$$

- Then we can get $\mu_{\text{sup}} = e^{-it\lambda} + O((t\lambda)^{2K+1})$

Analytic formula for α_k

- We aim to find α_k such that

$$\sum_{k=-K}^{K} \alpha_k \mu^k \approx e^{-i\lambda t}$$

- The formula for μ gives

$$e^{-i\lambda t} = \exp\left[\frac{t}{2}\left(\mu - \frac{1}{\mu}\right)\right]$$

- But this is the generating function for Bessel functions!

$$\sum_{k=-\infty}^{\infty} J_k(t)\mu^k = \exp\left[\frac{t}{2}\left(\mu - \frac{1}{\mu}\right)\right]$$

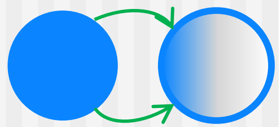
- We can choose α_k just from Bessel functions.

Without correcting the step

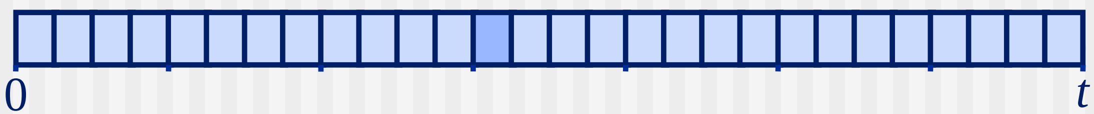
- We aim to find α_k such that
$$\sum_{k=-K}^{K} \alpha_k \mu_\pm^k \approx e^{-i\lambda t}$$
- The formula for $\mu_\pm$ gives
$$e^{-i\lambda t} = \exp\left[-\frac{t}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right)\right]$$
- But this is the generating function for Bessel functions!
$$\sum_{k=-\infty}^{\infty} J_k(-t)\mu^k = \exp\left[-\frac{t}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right)\right]$$
- We can choose α_k just from Bessel functions.
- We don't need to distinguish + from – or correct the step!

The complete algorithm

- Map into doubled Hilbert space.

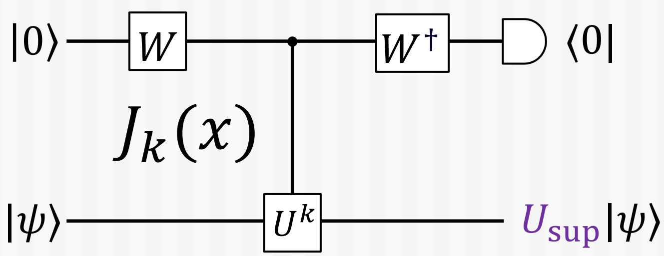


- Divide the time into $r = d\|H\|_{\max}t$ segments.

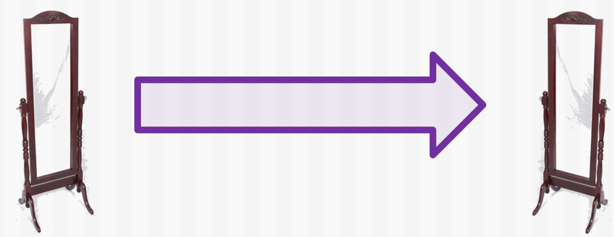


- For each segment:

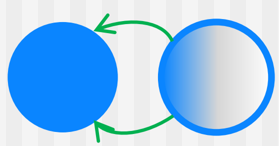
1. Perform the superposition.



2. Use amplitude amplification to obtain success deterministically.



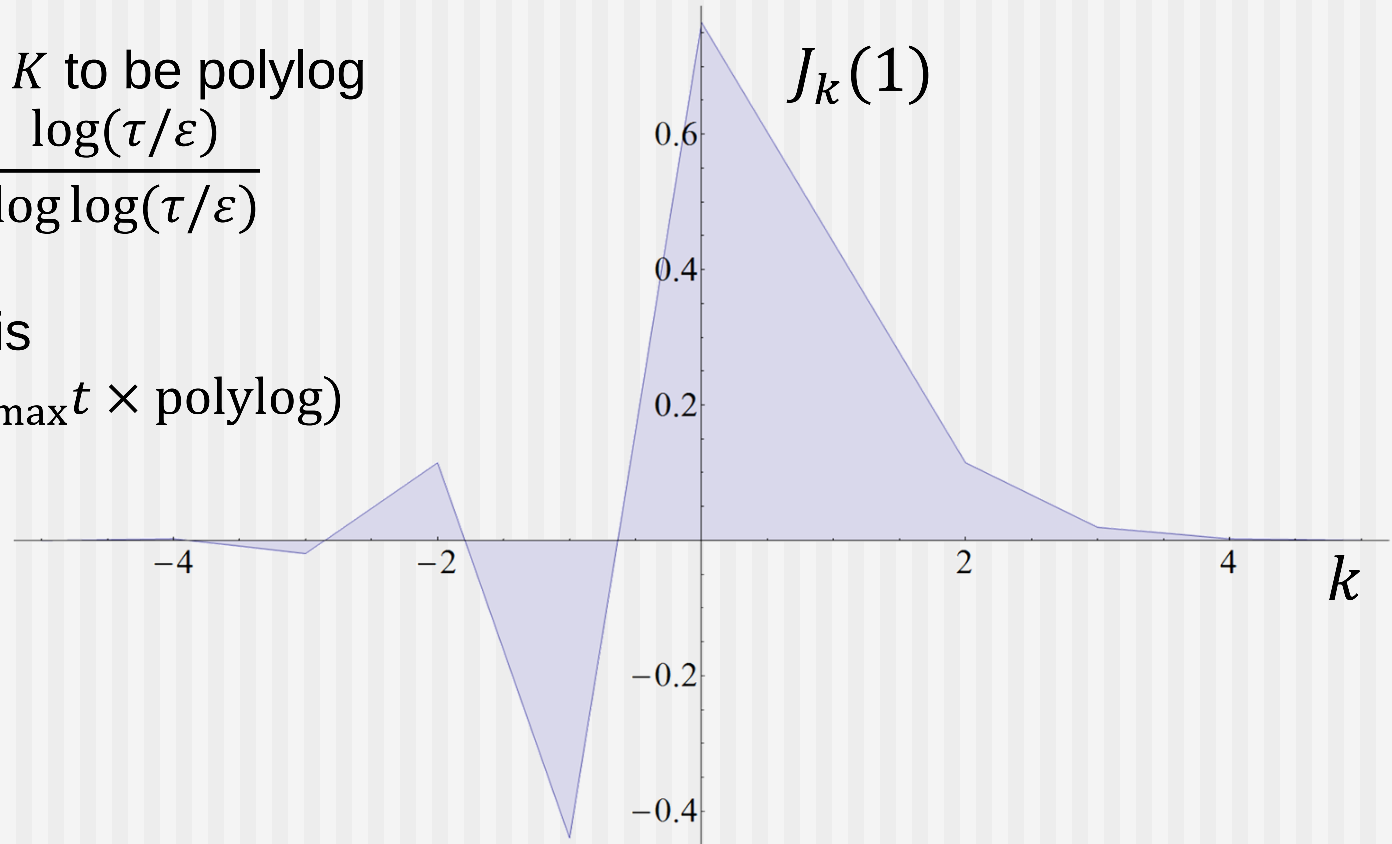
- Map back to original Hilbert space.



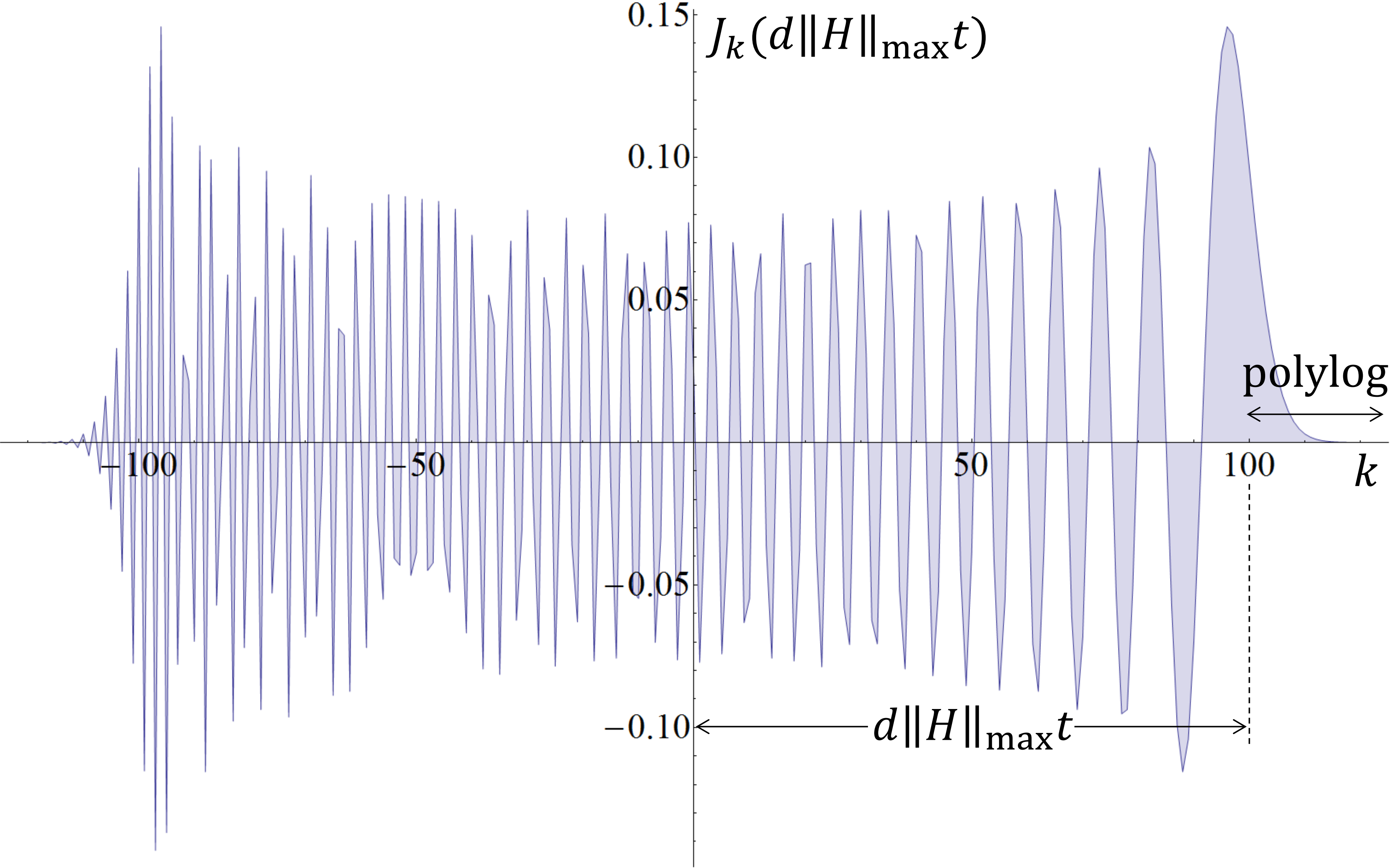
Total complexity: $d\|H\|_{\max}t \times K$

Choosing the value of K

- Bessel function may be bounded as
$$J_k(x) \le \frac{1}{k!}\left(\frac{x}{2}\right)^k$$
- Scaling is the same as for Taylor series!
- We can choose K to be polylog
$$K \sim \frac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}$$
- Overall scaling is
$$O(d\|H\|_{\max} t \times \text{polylog})$$



Single-segment approach



Conclusions

- We have complexity of sparse Hamiltonian simulation scaling as

$$O(d\|H\|_{\max}t \times \text{polylog})$$

- The lower bound is scaling as

$$O(d\|H\|_{\max}t + \text{polylog})$$

- The method combines the quantum walk and compressed product formula approaches.