Hamiltonian simulation with nearly optimal dependence on all parameters

Dominic Berry

MACQUARIE UNIVERSITY

+

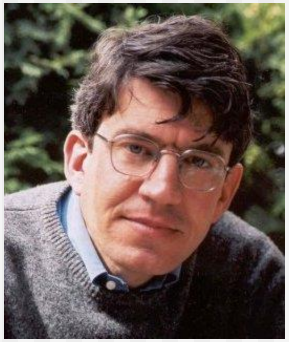
arXiv:1501.01715

Andrew Childs & Robin Kothari

UNIVERSITY OF MARYLAND 18 56

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

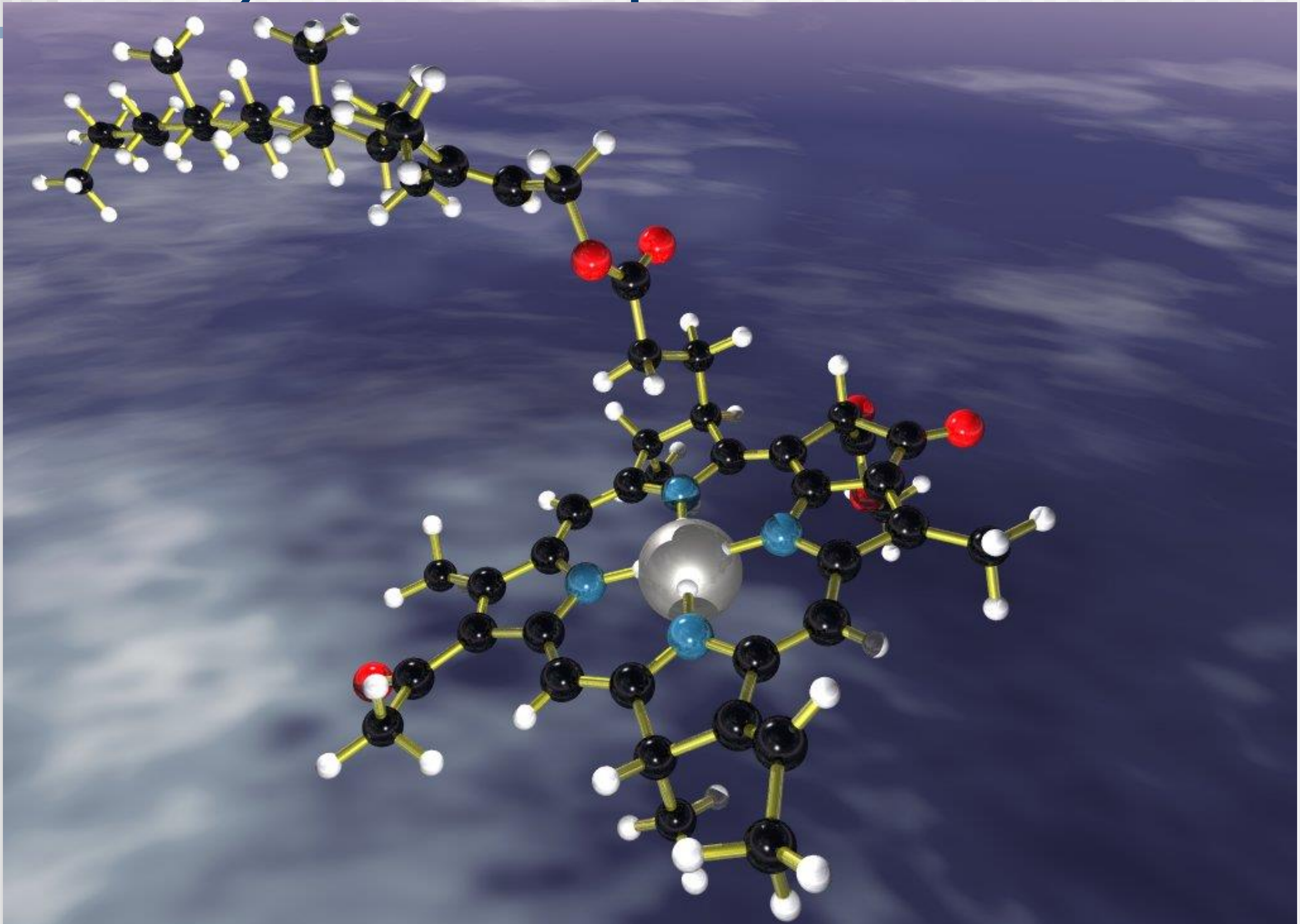
+ Richard Cleve & Rolando Somma



IQC Institute for Quantum Computing

Los Alamos NATIONAL LABORATORY EST. 1943

Why is this important?



Why is this important?

Aharonov & Ta-Shma
2003: Algorithm to simulate sparse Hamiltonians

Childs, Cleve, Jordan, Yonge-Mallo
2009: Quantum algorithm for NAND trees

Harrow, Hassidim, Lloyd
2009: Quantum algorithm to solve linear systems

Berry
2014: Quantum algorithm for differential equations

Wang
2014: Quantum algorithm for effective electrical resistance

Clader, Jacobs, Sprouse
2013: Quantum algorithm for scattering problems

The simulation problem

Problem: Given a Hamiltonian H, simulate

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

for time t and error no more than ε.

Inputs: H, t, ε

Parameters of H:

- d – sparseness
- N – dimension
- $\|H\|$ or $\|H\|_{\max}$ – norms of the Hamiltonian

Progression of results

Standard method:

Product formula
$O(d^4(\|H\|t)^{1+\delta}/\epsilon^\delta)$

Advanced methods:

Quantum walks
$O(d\|H\|_{\max}t/\sqrt{\epsilon})$

Compressed product formula or Taylor series
$O(d^2\|H\|_{\max}t \times \mathrm{polylog})$

New method:

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

Main results

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

- Near-linear in d, like quantum walk approach.
- Polylogarithmic in ε, like compressed product formulae.

What is the polylog factor?

Queries: polylog $\equiv \dfrac{\log(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}$

Gates: polylog $\equiv \dfrac{\log^2(\tau/\varepsilon)}{\log\log(\tau/\varepsilon)}$

$$\tau = d\|H\|_{\max}t$$

Lower bound: $\Omega(d\|H\|_{\max}t + \mathrm{polylog})$

Model

Sparse Hamiltonians

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

- **Query:** An efficient algorithm to determine the positions and values of non-zero entries.

Standard method

- Use decomposition as

$$H = \sum_{k=1}^{M} H_k$$

- Divide time into r intervals and use product formula:

$$e^{-iHt} \approx \left(\prod_{k=1}^{M} e^{-iH_k t/r} \right)^r$$

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

Advanced methods

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

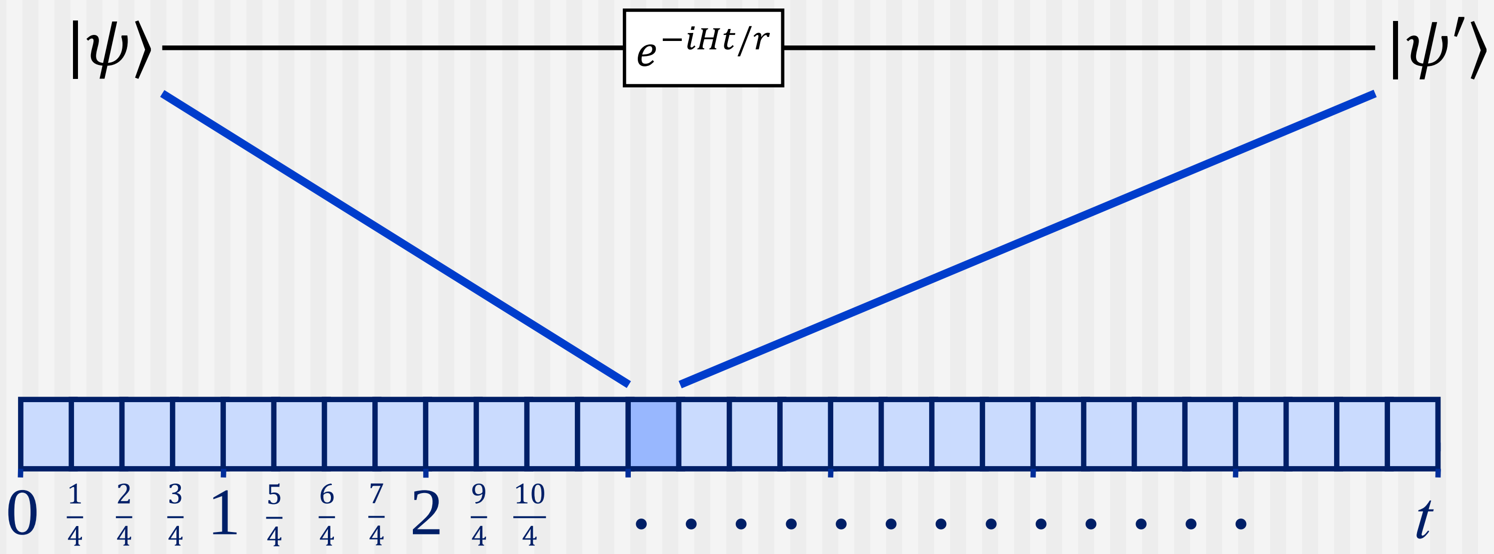
D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC '14; arXiv:1312.1414 (2013).
D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
D. W. Berry, A. M. Childs, R. Kothari,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

Compressed product formulae

Crucial ideas we use in new work:

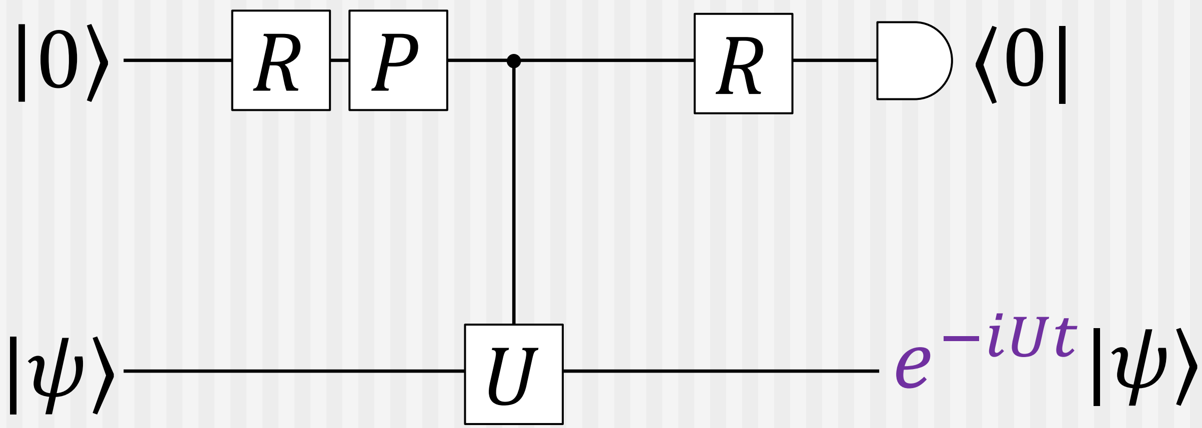
1. Break evolution into segments.
2. In each segment use controlled operations.
3. Apply oblivious amplitude amplification to achieve result deterministically.

Break into segments



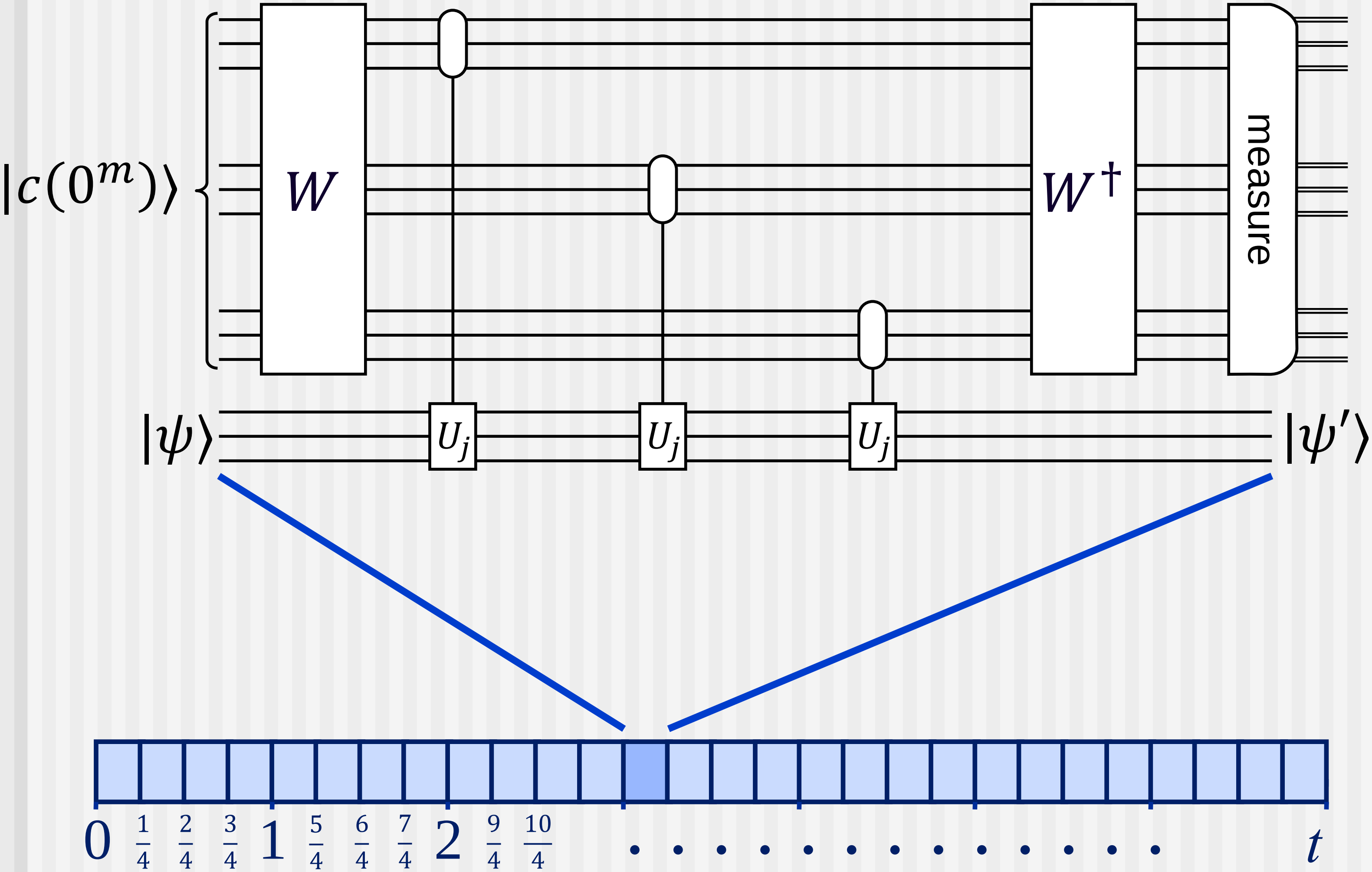
Evolution using control qubits

U is self-inverse

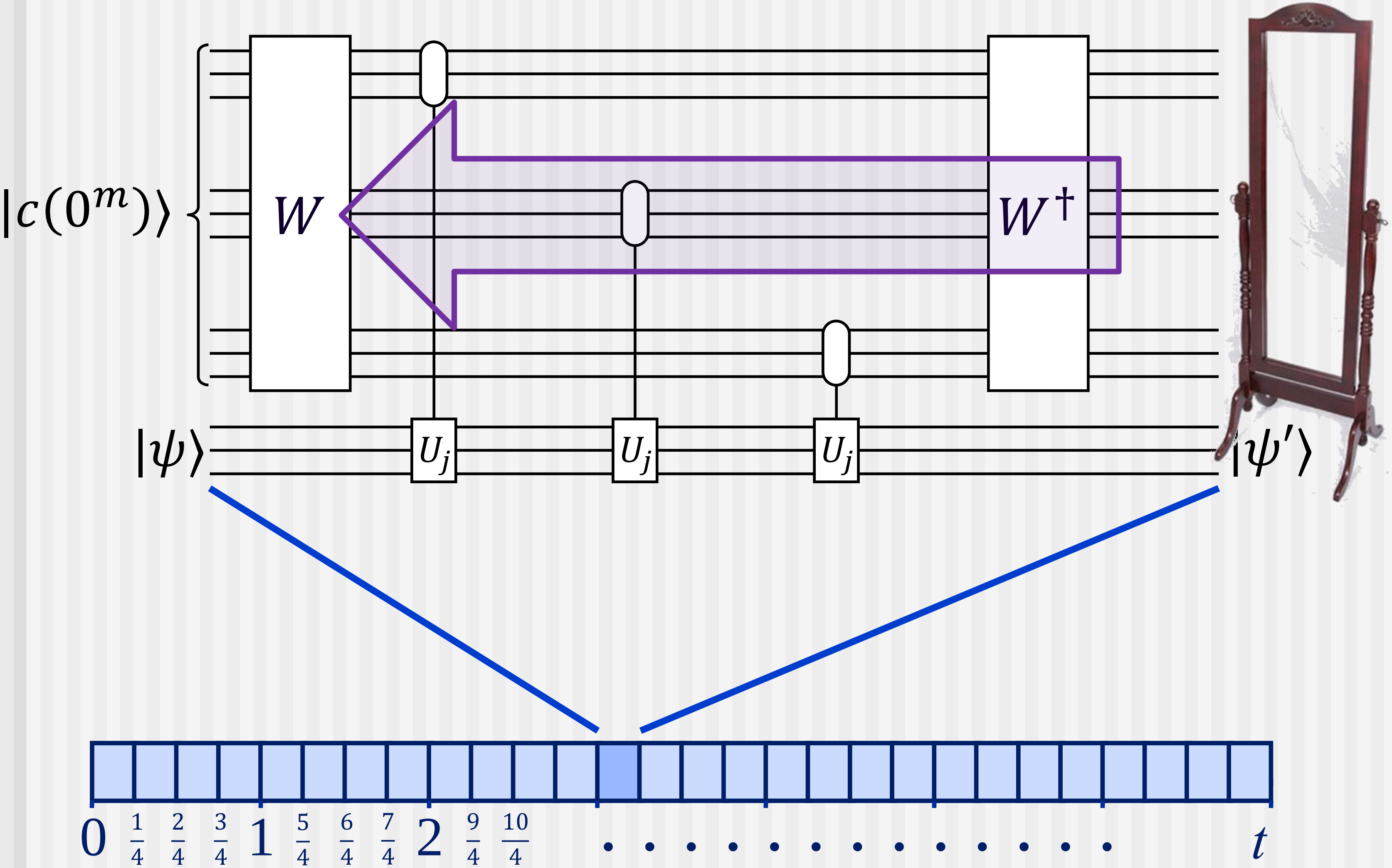


R. Cleve, D. Gottesman, M. Mosca, R. Somma,
D. Yonge-Mallo, STOC ‘09; arXiv:0811.4428 (2008).

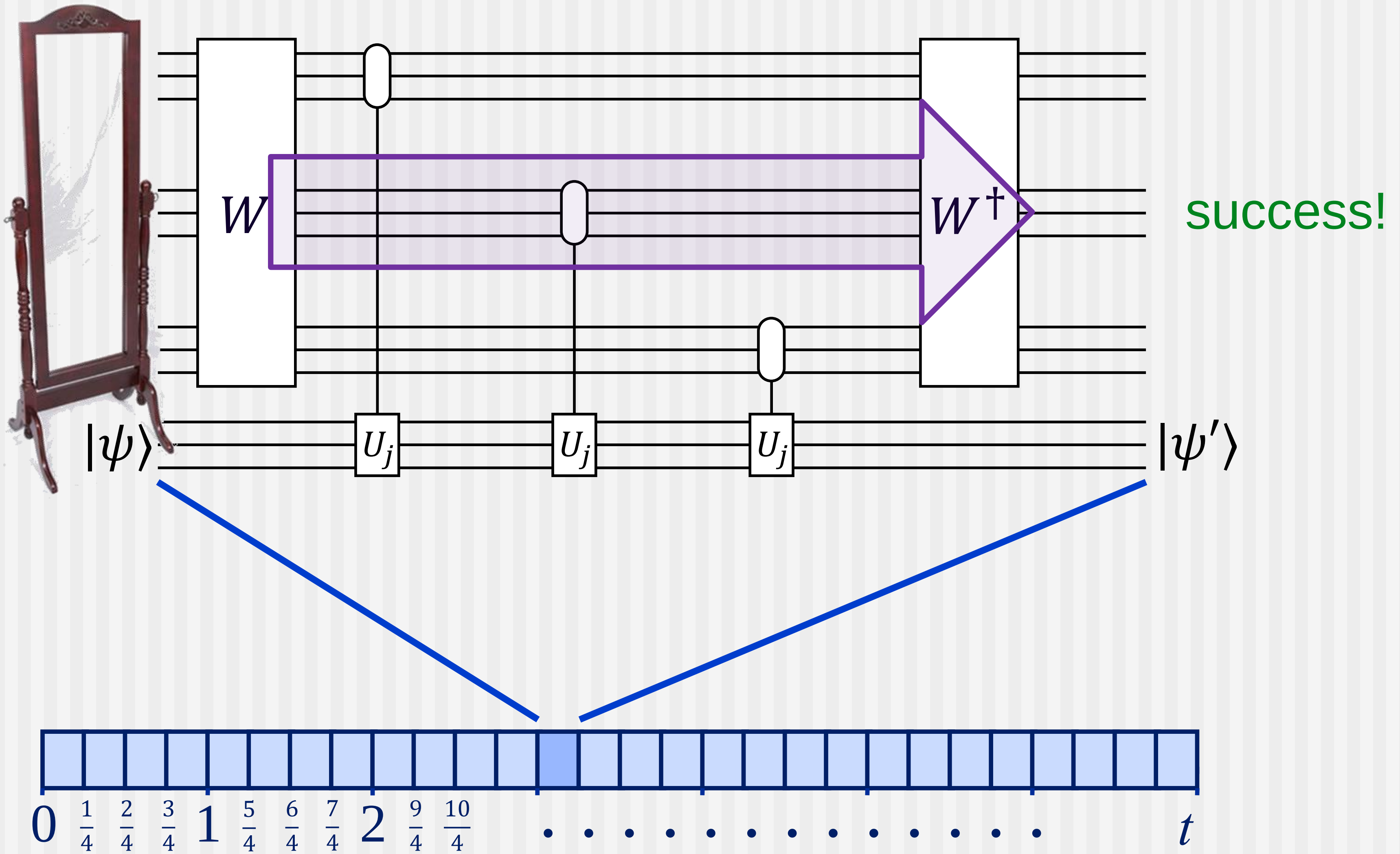
Oblivious amplitude amplification



Oblivious amplitude amplification



Oblivious amplitude amplification



Advanced methods

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

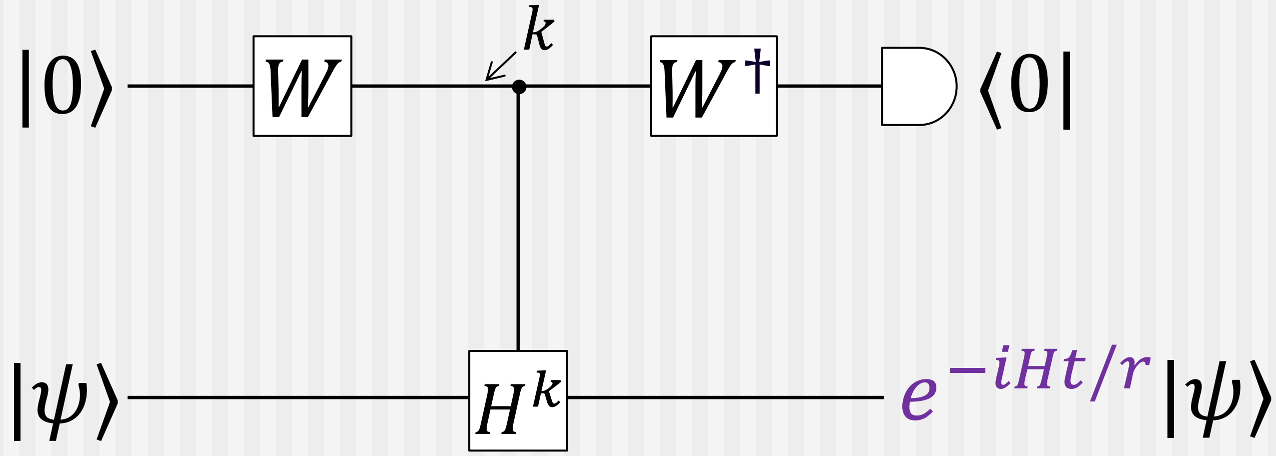
D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, STOC '14; arXiv:1312.1414 (2013).
D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, R. D. Somma, arXiv:1412.4687 (2014).
D. W. Berry, A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
D. W. Berry, A. M. Childs, R. Kothari,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

Implementing Taylor series

- Break Hamiltonian evolution into r segments and use

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

- Aim to perform using controlled operations.



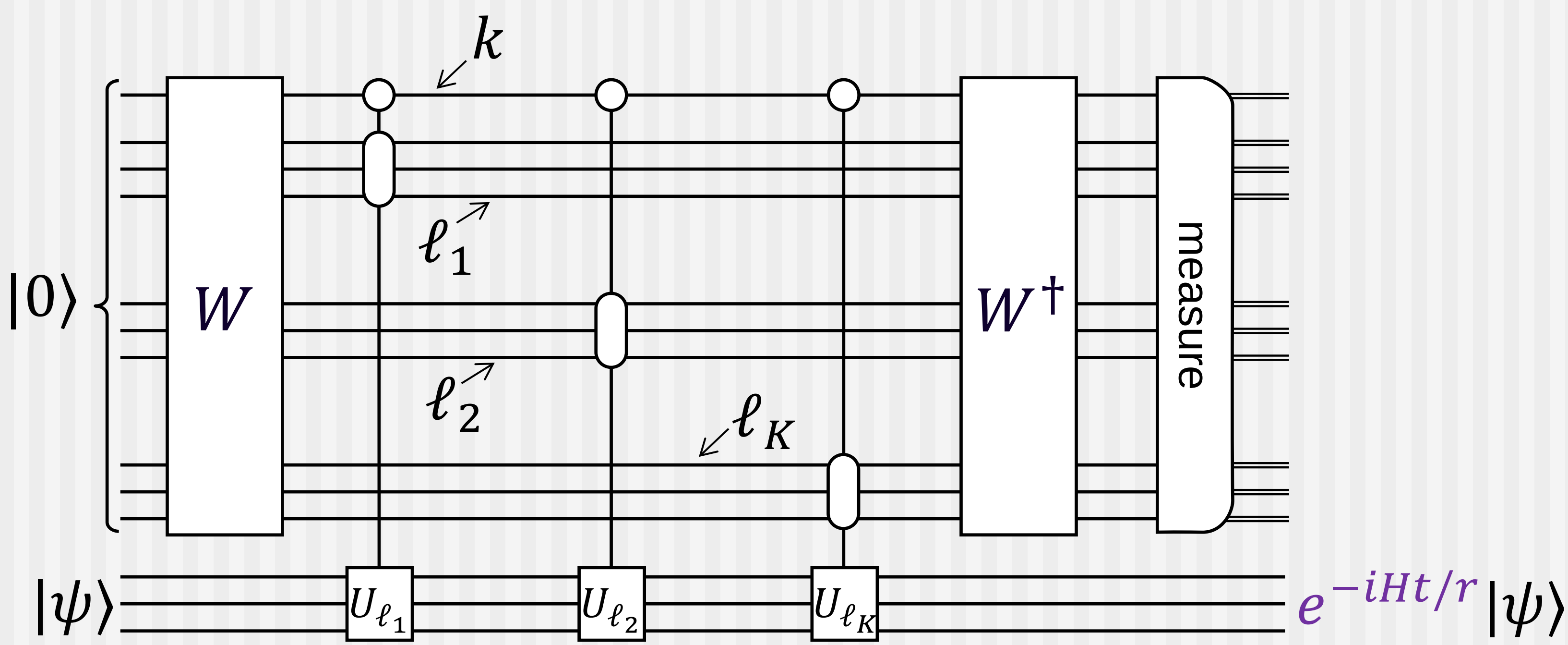
Implementing Taylor series

- Expand H as sum of unitaries

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

- Exponential is then

$$e^{-iHt/r} \approx \sum_{k=0}^{K} \sum_{\ell_1=1}^{M} \sum_{\ell_2=1}^{M} \cdots \sum_{\ell_k=1}^{M} \frac{(-it/r)^k}{k!} U_{\ell_1} U_{\ell_2} \cdots U_{\ell_k}$$

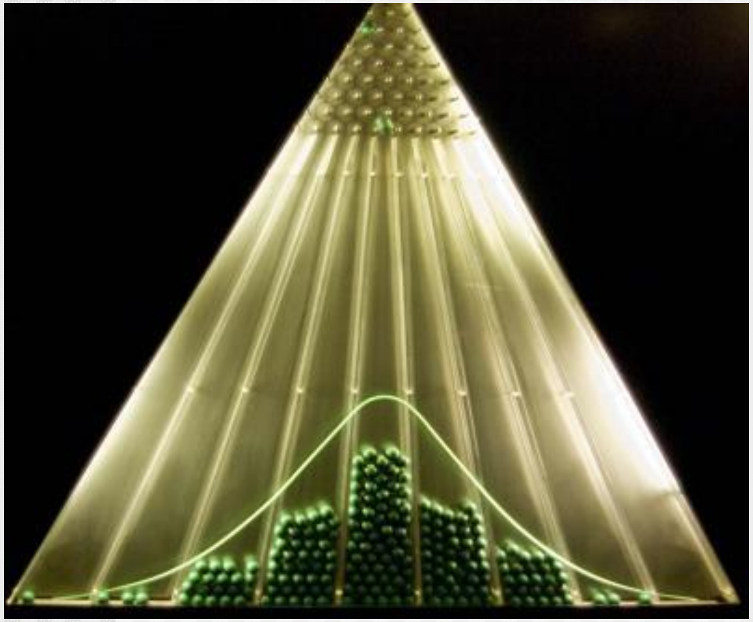


Advanced methods

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

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

Quantum walks



Classical walk

- Position is integer x.
- Step is map $x \to x \pm 1$ with equal probability.

Standard quantum walk

- Quantum position and coin registers $|x, c\rangle$.
- Alternates coin and step operators,

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

Szegedy quantum walk

- Two subsystems with arbitrary dimension.
- Step is controlled reflection.

Szegedy quantum walk

- Controlled reflections:

$$\sum_j |j\rangle\langle j| \otimes (2|c_j\rangle\langle c_j| - \mathbb{I})$$

controlled on j — reflect about $|c_j\rangle$

- After doing this we swap the two systems.
- Step operation is
$$U = i \times \mathrm{SWAP} \times \text{controlled reflection}$$
- Controlled reflection can be achieved with controlled preparation:

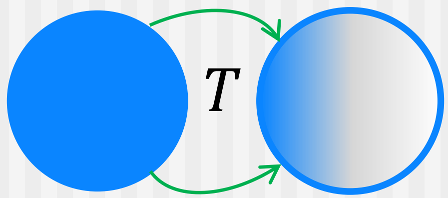
$$T = \sum_j |j\rangle\langle j| \otimes |c_j\rangle$$

M. Szegedy, FOCS '04; arXiv:quant-ph/0401053 (2004).

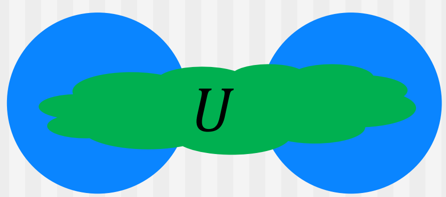
Szegedy walk for Hamiltonians

Three part process:

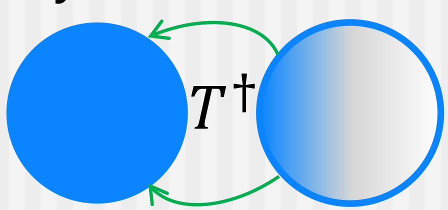
1. Start with state in one of the subsystems, and perform controlled state preparation T.



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



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

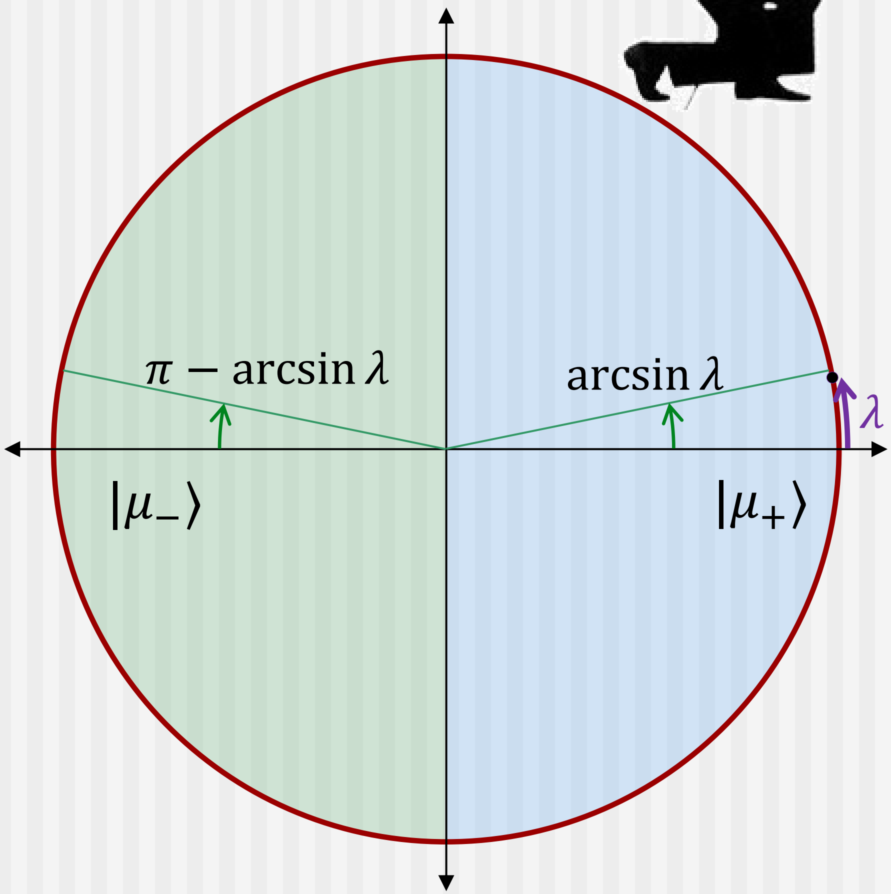


Each U or T uses $O(1)$ calls to H.

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

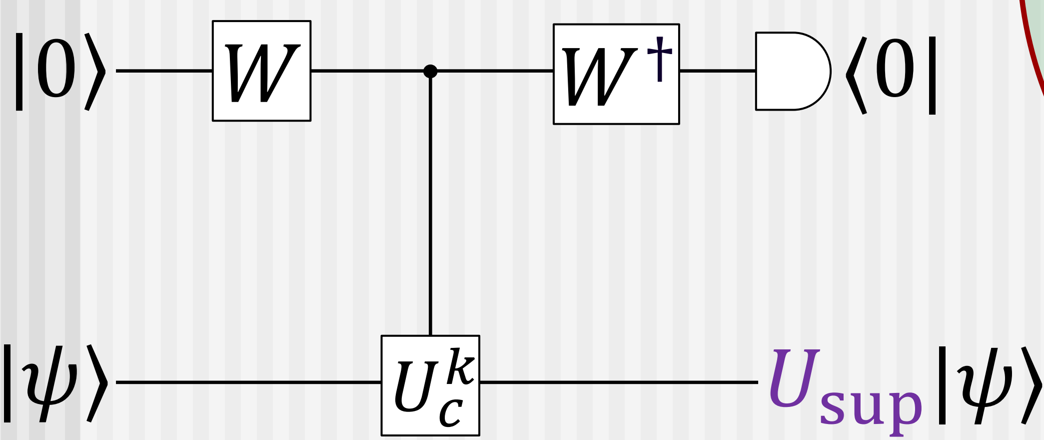
Eigenvalues of walk

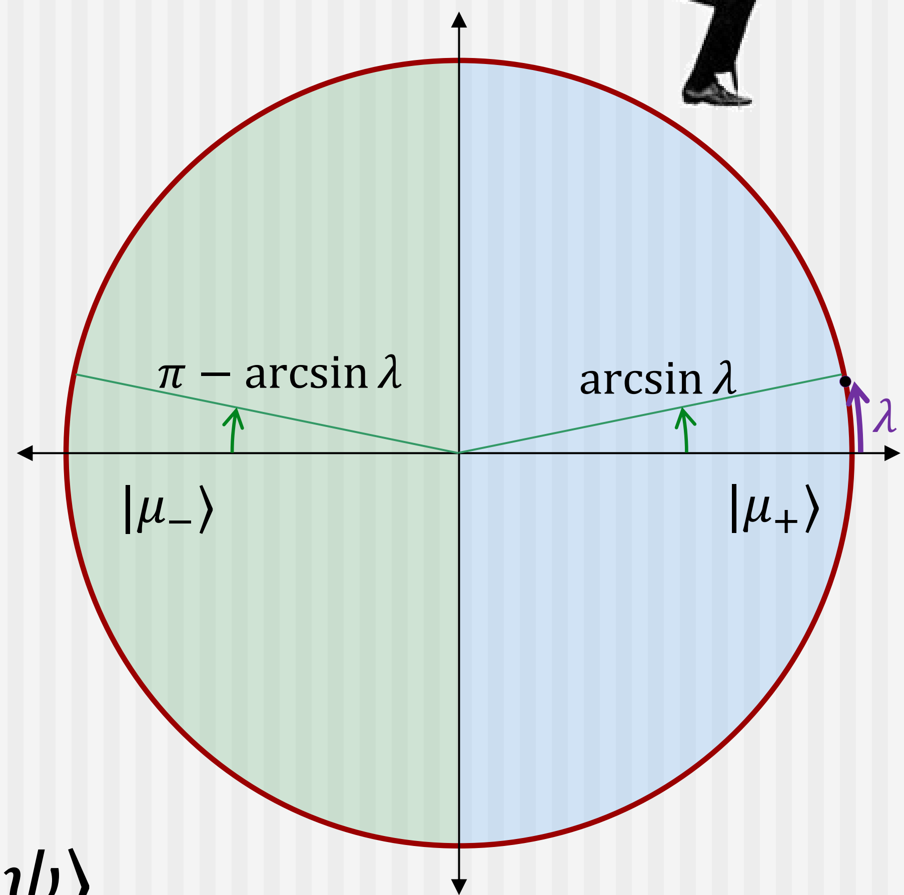
- Hamiltonian H has eigenvalues λ.
- Step U has eigenvalues
$$\mu_{\pm} = \pm e^{\pm i \arcsin \lambda}$$
- Evolution under the Hamiltonian has eigenvalues
$$e^{-i\lambda t}$$
- Given knowledge of + or − we can correct to U_c with eigenvalues
$$\mu = e^{-i \arcsin \lambda}$$



Eigenvalues of walk

- Step U_c has eigenvalues
$$\mu = e^{-i \arcsin \lambda}$$
- We aim for
$$e^{-i\lambda t}$$
- Try superposition of operations
$$U_{\text{sup}} = \sum_{k=-K}^{K} \alpha_k U_c^k$$





Choosing values 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!

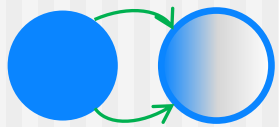
$$\sum_{k=-\infty}^{\infty} J_k(-t)\mu_{\pm}^{k} = \exp\left[-\frac{t}{2}\left(\mu_{\pm} - \frac{1}{\mu_{\pm}}\right)\right]$$

- We can choose α_k just from Bessel functions.

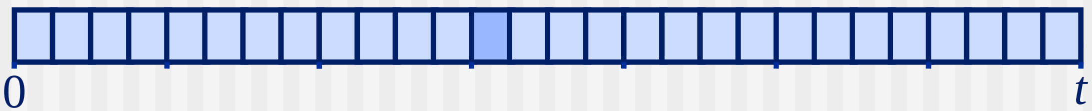
- We don't need to distinguish $+$ from $-$ or correct the step!

The complete algorithm

- Map into doubled Hilbert space.

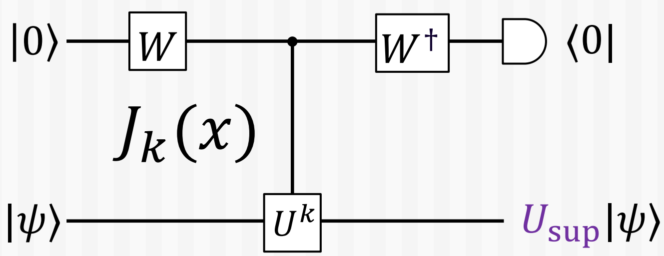


- Divide the time into $r = d\|H\|_{\max} t$ segments.

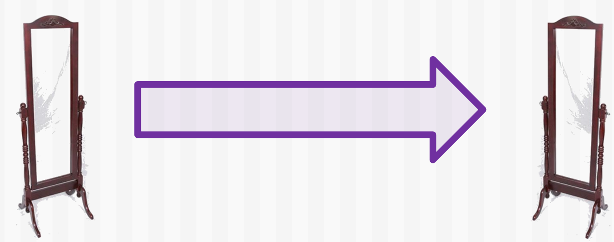


- For each segment:

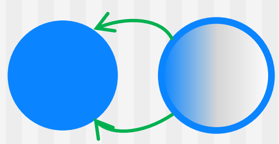
1. Perform the superposition.



2. Use amplitude amplification to obtain success deterministically.



- Map back to original Hilbert space.



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

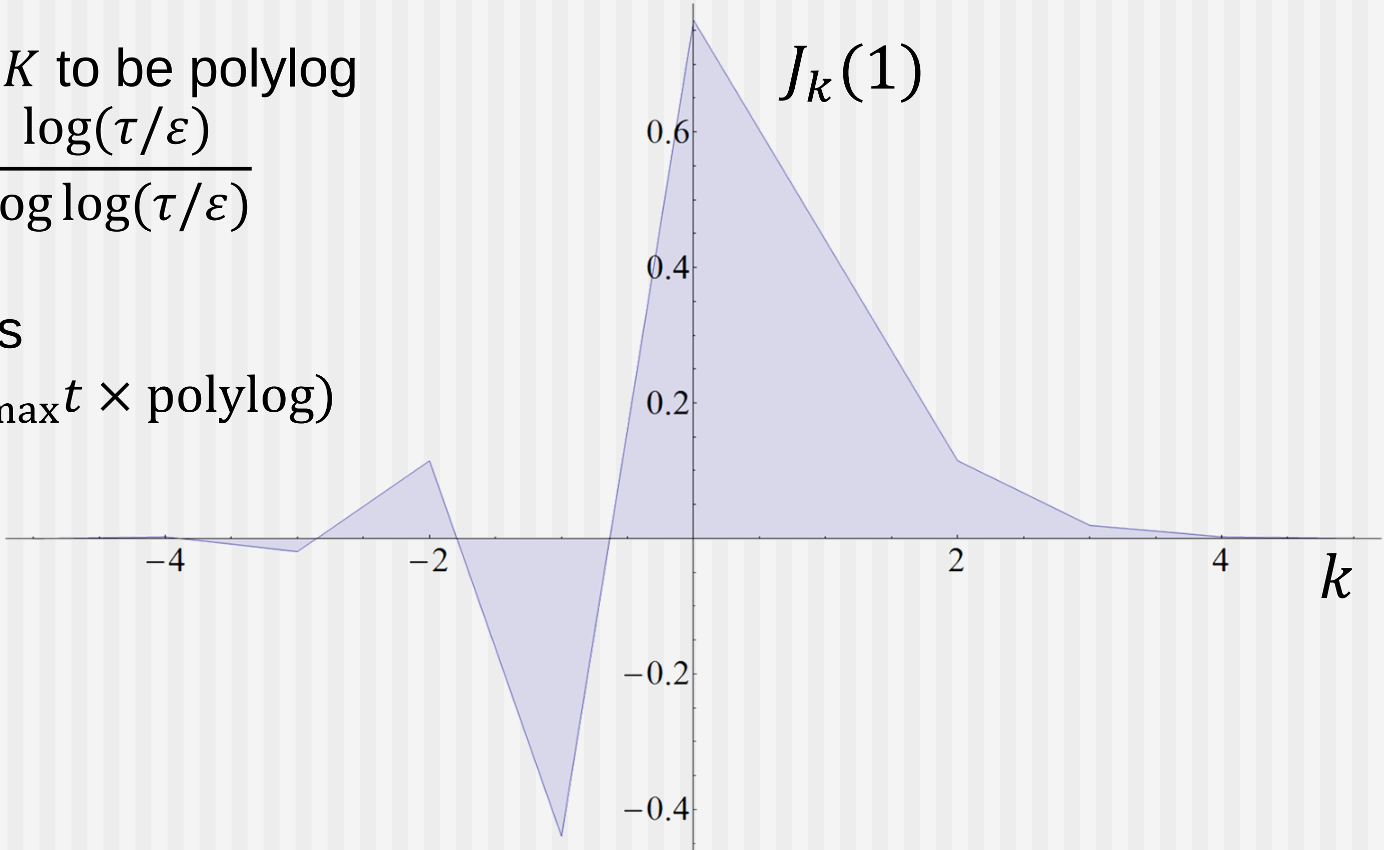
Choosing the value of K

- Bessel function may be bounded as
$$J_k(x) \le \frac{1}{k!}\left(\frac{x}{2}\right)^k$$

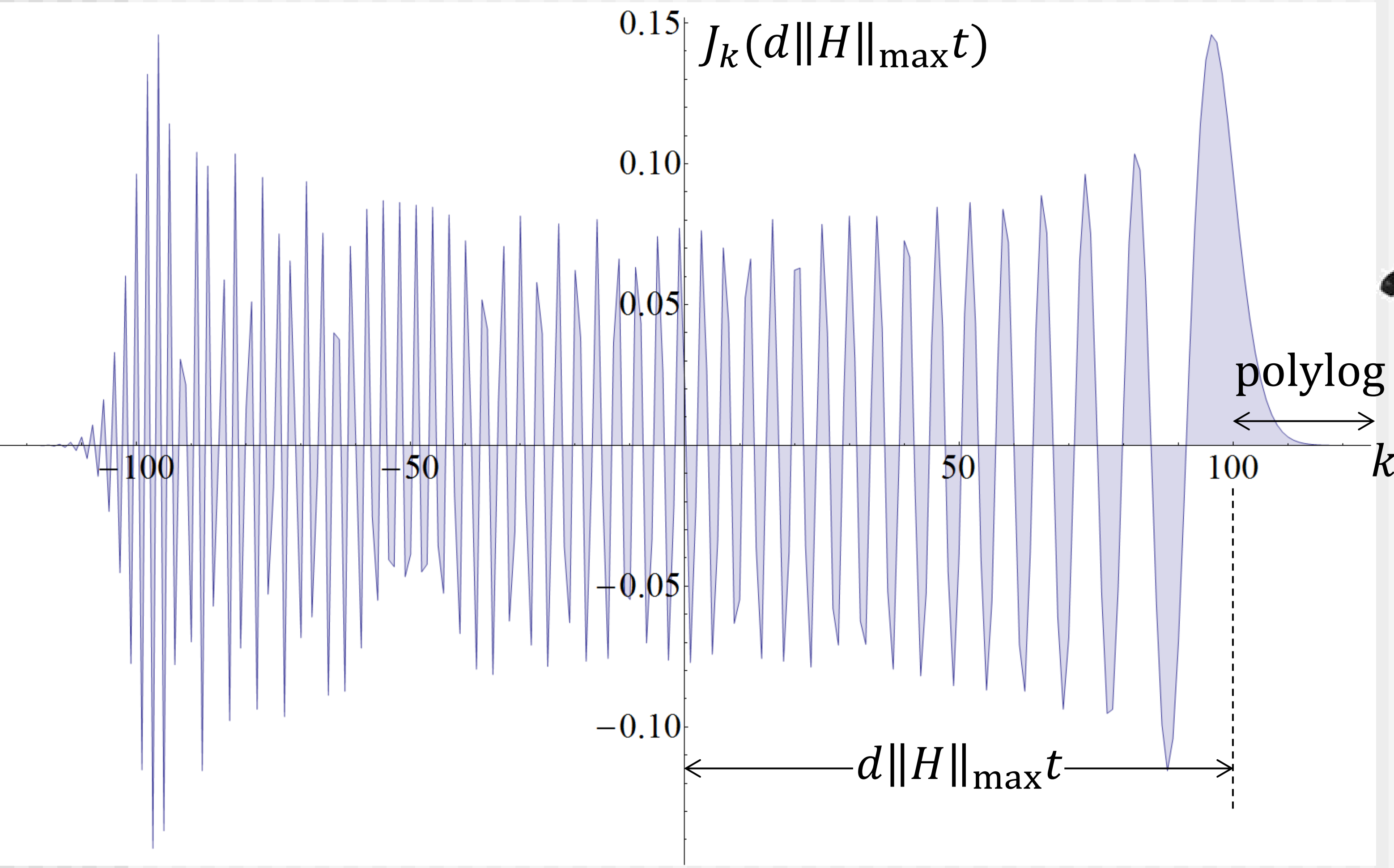
- Scaling is the same as for Taylor series!

- 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

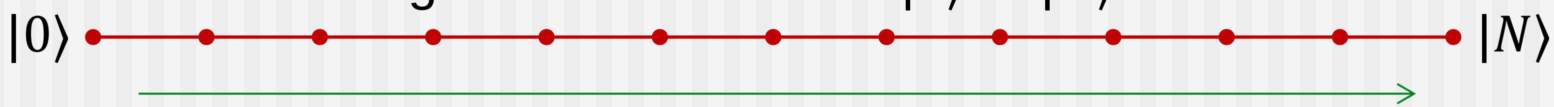


Choosing segment sizes τ^{α} gives complexity

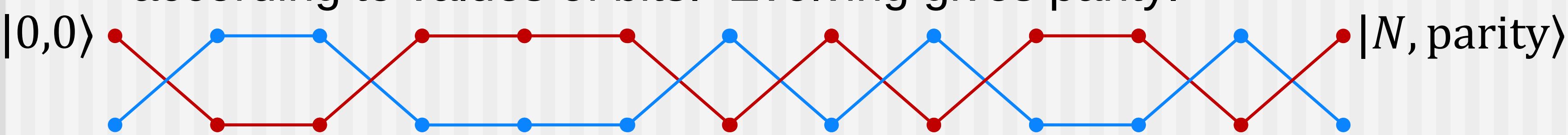
$$\tau^{1+\alpha/2} + \tau^{1-\alpha/2}\log(1/\varepsilon)$$

Lower bound

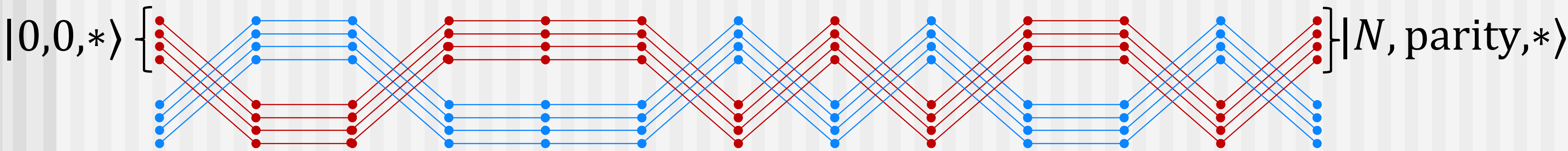
- Computing parity of a bit string $x_1, \ldots, x_N$ has complexity $\Omega(N)$.
- Can define a Hamiltonian such that evolving under the Hamiltonian gives transition from $|0\rangle$ to $|N\rangle$.



- Can define a Hamiltonian such that the states are connected according to values of bits. Evolving gives parity.



- $N \propto \|H\|_{\max} t$ gives $\Omega(\|H\|_{\max} t)$ lower bound.
- Take d copies of each node, and use superposition.



- $N \propto d\|H\|_{\max} t$ gives $\Omega(d\|H\|_{\max} t)$ lower bound.

Conclusions

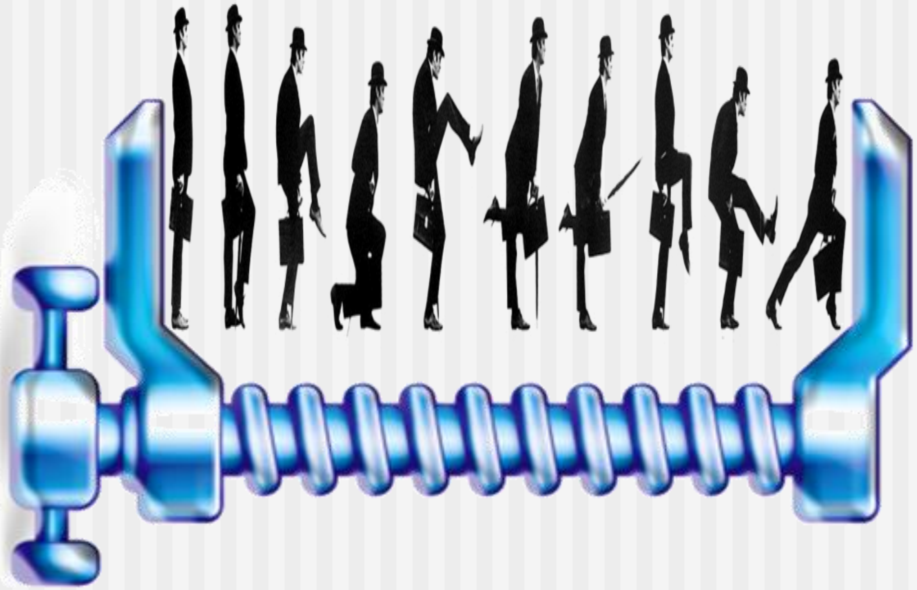
- We have complexity of sparse Hamiltonian simulation scaling as

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

- The lower bound is scaling as

$$\Omega(d\|H\|_{\max}t + \mathrm{polylog})$$

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



arXiv:1501.01715