# Double-bracket quantum algorithms

Nelly Ng
Nanyang Technological University
Singapore

# im.Q @ NTU

Quantum resource theories

Quantum algorithms

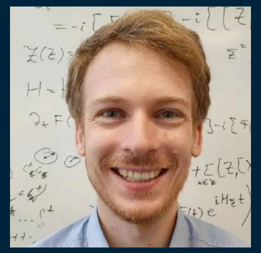
in.Q

Quantum field simulators

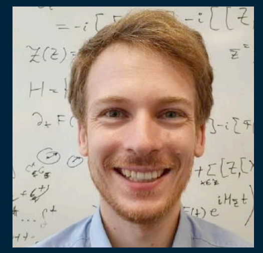
Quantum communications

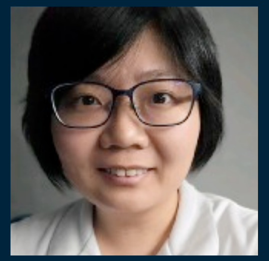
# An ongoing journey…

Double-bracket quantum algorithms for diagonalization
Quantum 8, 1316 (2024)



Quantum Dynamic Programming
Phys. Rev. Lett. 134, 180602 (2025)





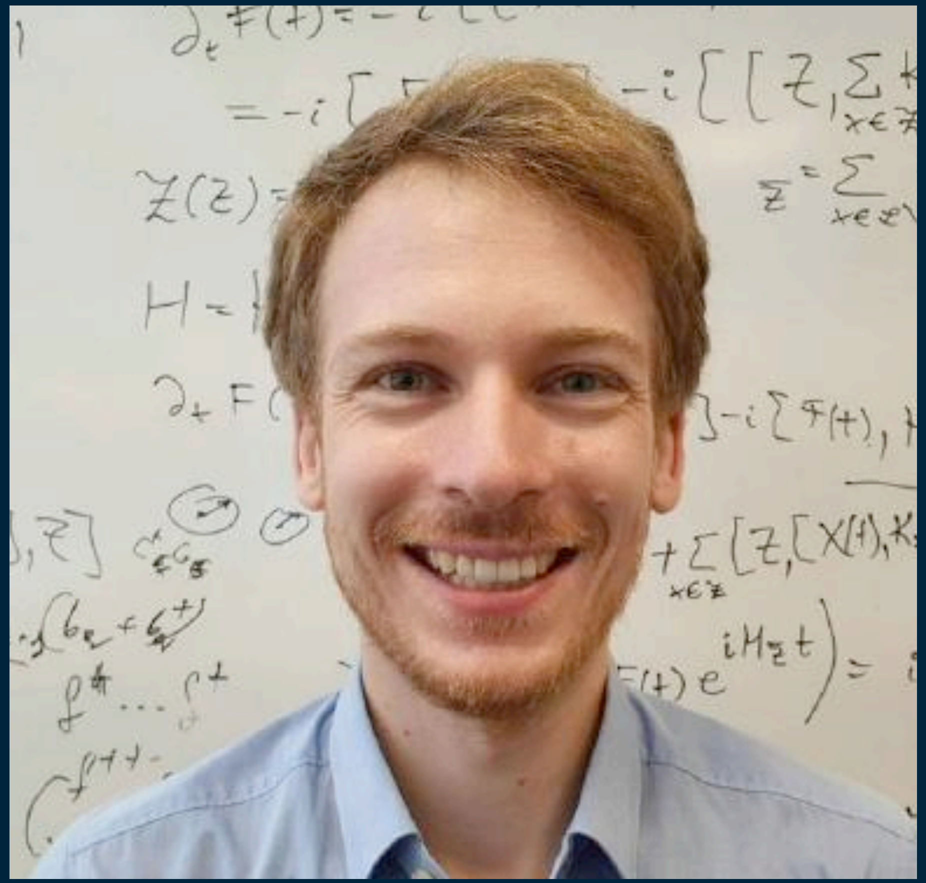
Double-bracket quantum algorithms for quantum imaginary-time evolution
arXiv:2412.04554

Double-bracket algorithm for quantum signal processing without post-selection
arXiv:2504.01077

Grover's algorithm is an approximation of imaginary-time evolution
arXiv:2507.15065

im.Q

Yours truly :)



Marek Gluza

Jeongrak Son

Tiang Bi Hong

NTU Singapore

Yudai Suzuki

Zoe Holmes

EPFL, Switzerland

René Zander

Raphael Seidel

Fraunhofer, Germany

# Outline

What is a double-bracket flow equation?

How do we implement a double-bracket flow algorithmically?

What is the steady state of a double-bracket flow?

What are the implications for the above?

# Outline

What is a double-bracket flow equation?

How do we implement a double-bracket flow algorithmically?

What is the steady state of a double-bracket flow?

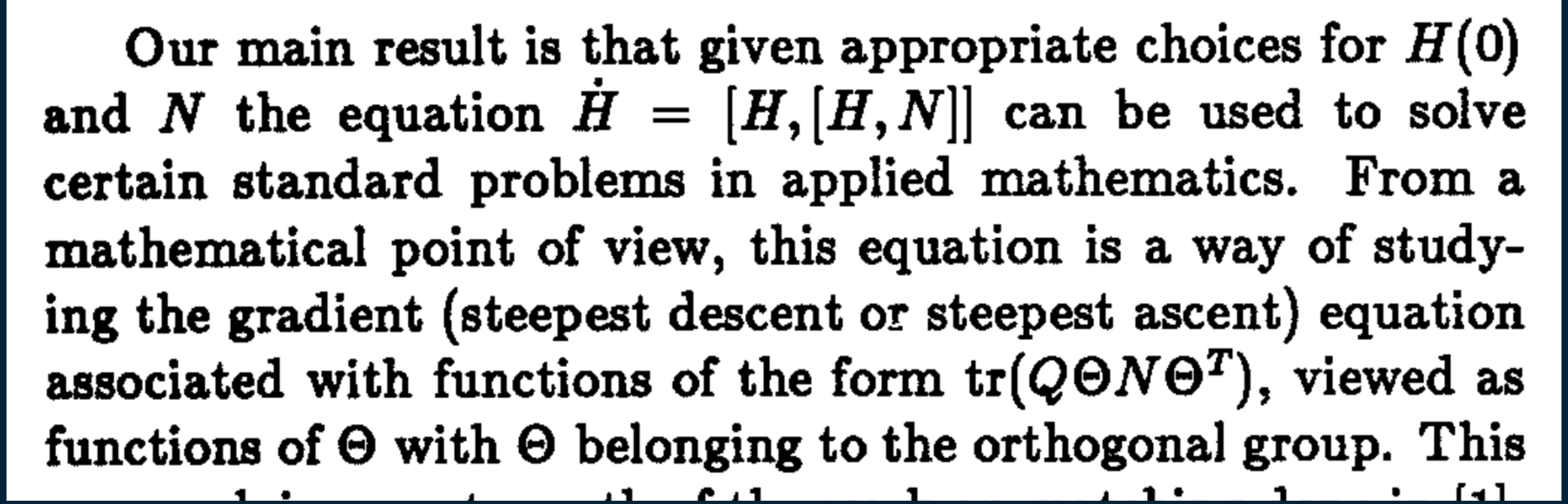
What are the implications for the above?

# The story starts with double-bracket flows by Brockett 1991

$$\dot{N}(t) = [N(t), [N(t), H]]$$

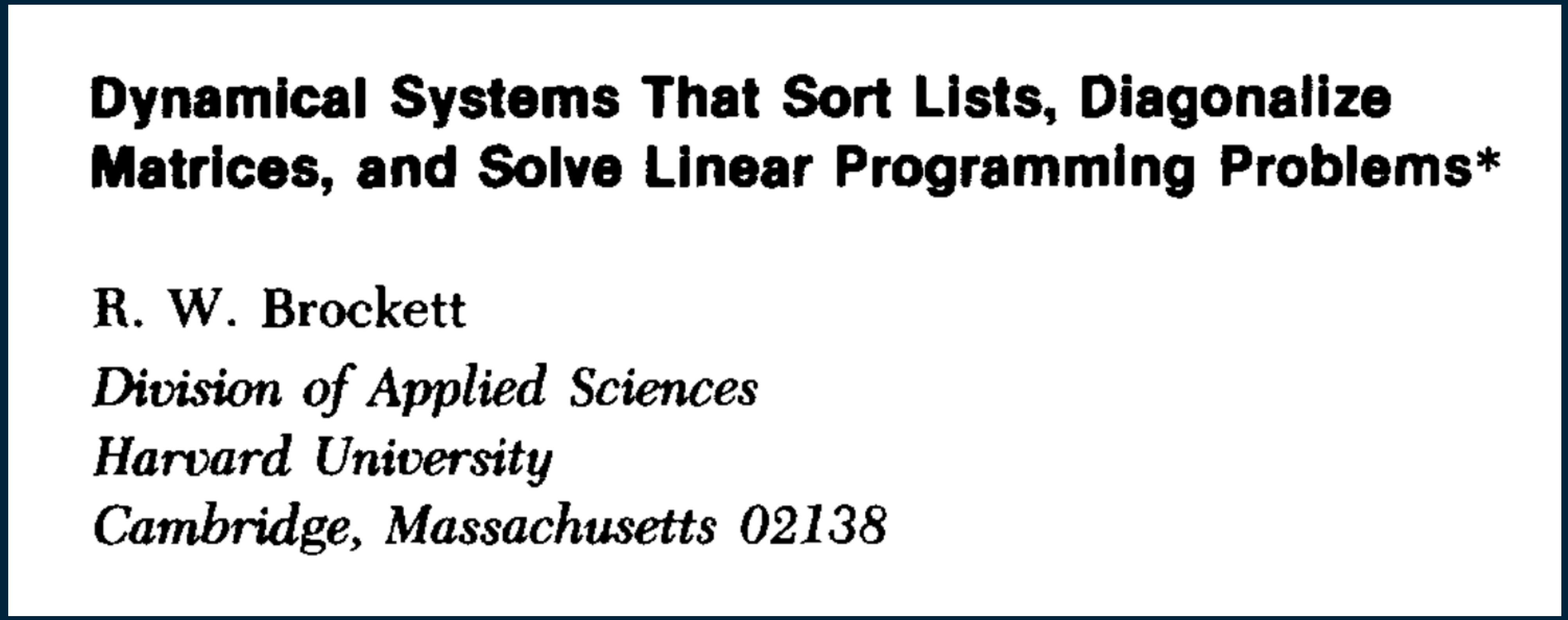
*A rate equation for the matrix $N(t)$, which is determined by a "double bracket (commutator)".*

$H, N$ are symmetric, real $n \times n$ matrices



Our main result is that given appropriate choices for $H(0)$ and $N$ the equation $\dot{H} = [H,[H,N]]$ can be used to solve certain standard problems in applied mathematics. From a mathematical point of view, this equation is a way of studying the gradient (steepest descent or steepest ascent) equation associated with functions of the form $\mathrm{tr}(Q\Theta N\Theta^T)$, viewed as functions of $\Theta$ with $\Theta$ belonging to the orthogonal group. This

*The flow equation is derived assuming that dynamics on $N(t)$ trace out a steepest descent that optimizes $\mathrm{tr}(HN(t))$, where $N(t) = O(t)N(0)O^T(t)$, $O(t)$ being a continuously parametrized orthogonal matrix*



**Dynamical Systems That Sort Lists, Diagonalize Matrices, and Solve Linear Programming Problems***

R. W. Brockett
*Division of Applied Sciences*
*Harvard University*
*Cambridge, Massachusetts 02138*

# The story starts with double-bracket flows by Brockett 1991

$$\dot{N}(t) = [N(t), [N(t), H]]$$

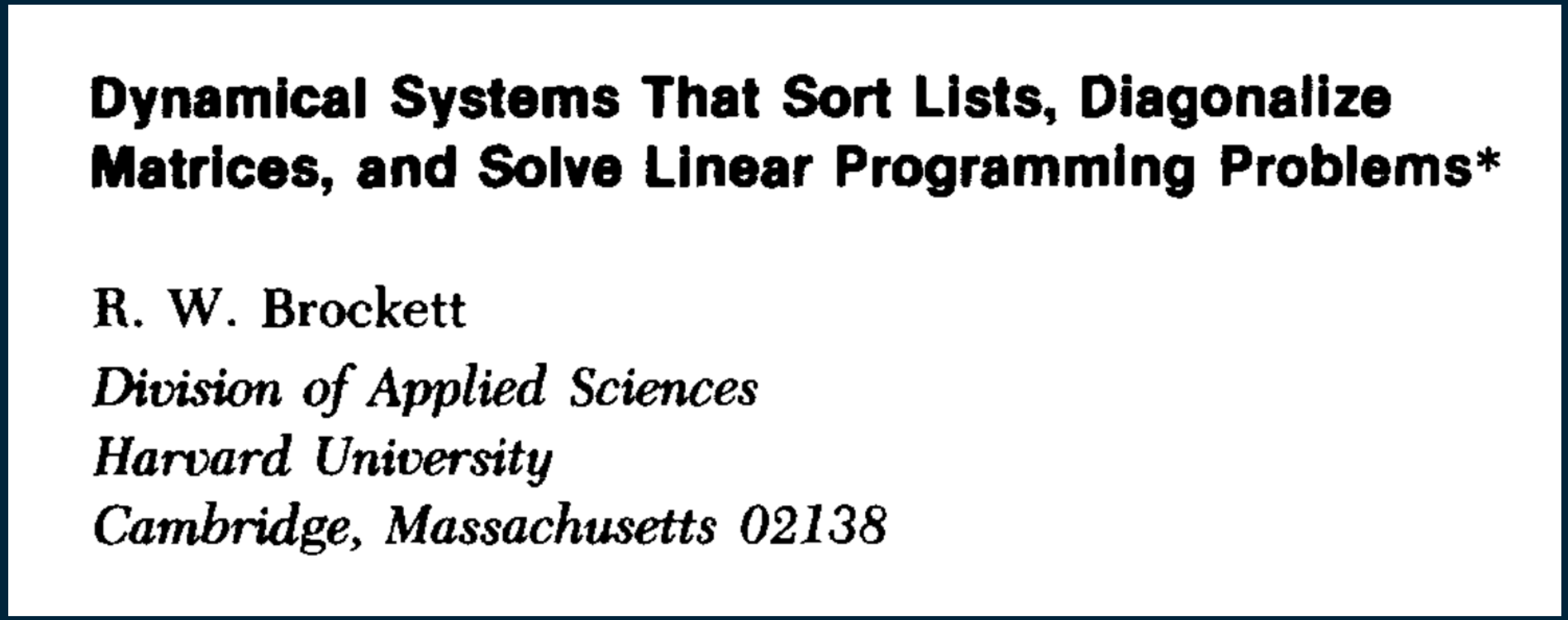
*A rate equation for the matrix $N(t)$, which is determined by a "double bracket (commutator)".*

$H, N$ are symmetric, real $n \times n$ matrices

Observation by Brockett: if $H$ is diagonal and non-degenerate, and $N(0)$ is symmetric, then

$$N(\infty) := \lim_{t \to \infty} N(t)$$

exists and is a diagonal matrix



**Dynamical Systems That Sort Lists, Diagonalize Matrices, and Solve Linear Programming Problems***

R. W. Brockett
*Division of Applied Sciences*
*Harvard University*
*Cambridge, Massachusetts 02138*

# The story starts with Brockett 1991

$$\dot{N}(t) = [N(t), [N(t), H]]$$

*A rate equation for the matrix $N(t)$, which is determined by a “double bracket (commutator)”*

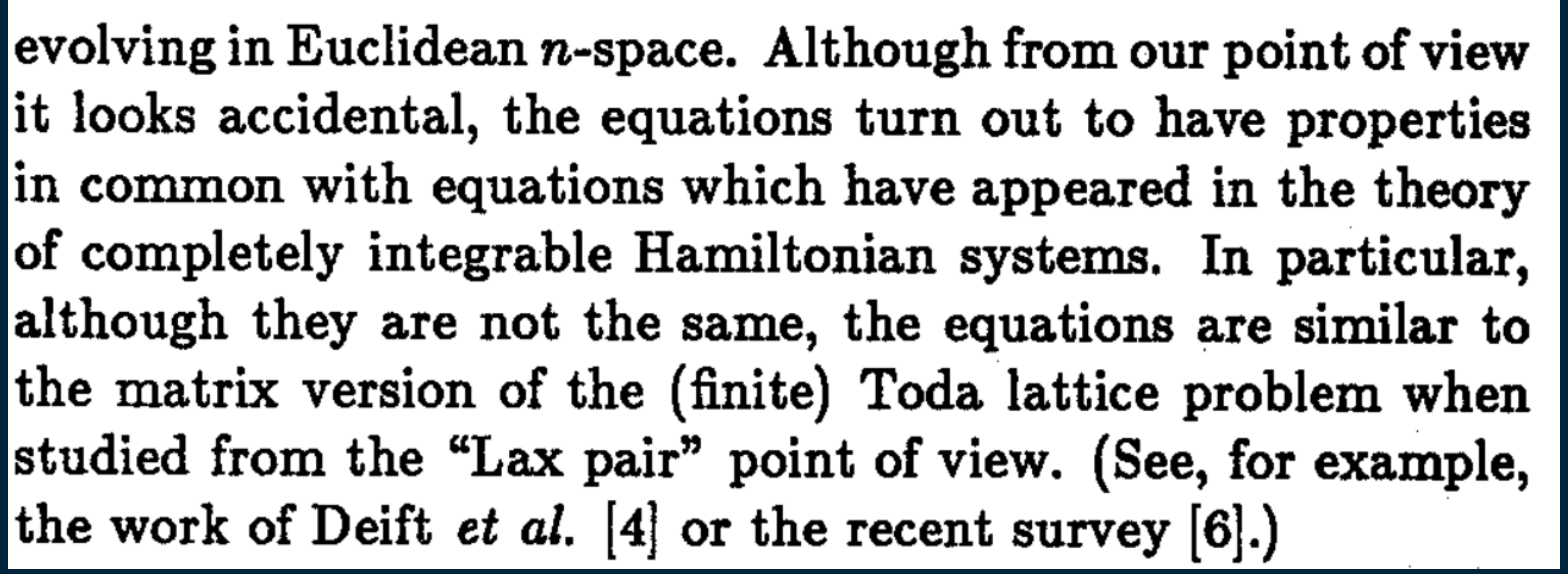
$H, N$ are symmetric, real $n \times n$ matrices

Proof sketch (generalization for non-degenerate H and complex matrices also possible):

1) consider $\dfrac{d}{dt}\operatorname{tr}(HN) = \operatorname{tr} N[H, [H, N]] = -\operatorname{tr}(HN - NH)^2$

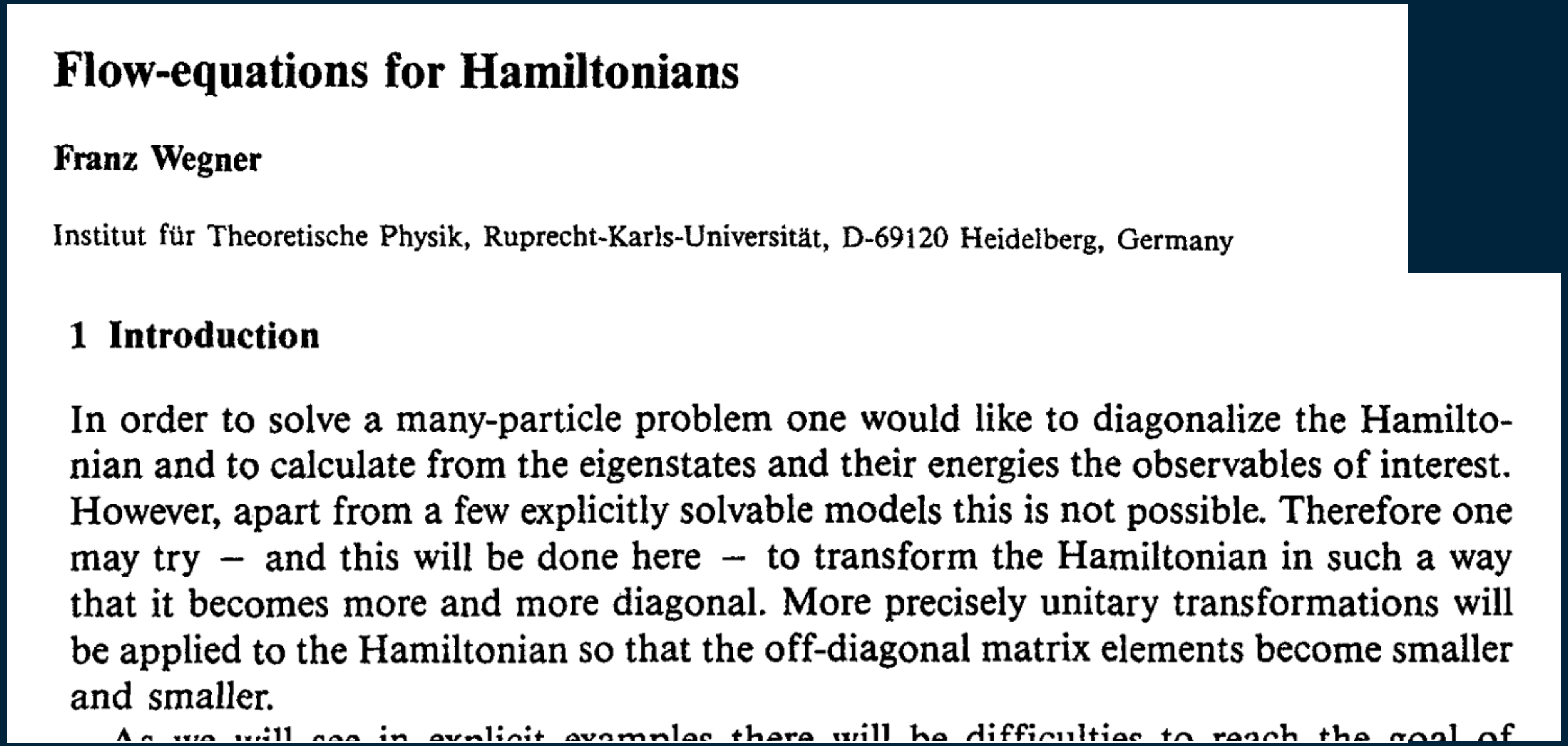
2) note that commutators are antisymmetric, so $\dfrac{d}{dt}\operatorname{tr}(HN) = \operatorname{tr}(HN - NH)(HN - NH)^T$, meaning that $\operatorname{tr}(HN)$ is monotonically increasing, and upper bounded, i.e. its derivative goes to 0. This can happen only if $N, H$ commute. Given that we assume $H$ non-degenerate, this means $N$ becomes diagonal.

# Implication of Brockett's formula for quantum many-body systems



evolving in Euclidean $n$-space. Although from our point of view it looks accidental, the equations turn out to have properties in common with equations which have appeared in the theory of completely integrable Hamiltonian systems. In particular, although they are not the same, the equations are similar to the matrix version of the (finite) Toda lattice problem when studied from the "Lax pair" point of view. (See, for example, the work of Deift *et al.* [4] or the recent survey [6].)

Głazek, Wilson, and Wegner, 1993-1994: applied flow equations for Hamiltonians of many-body systems



**Flow-equations for Hamiltonians**

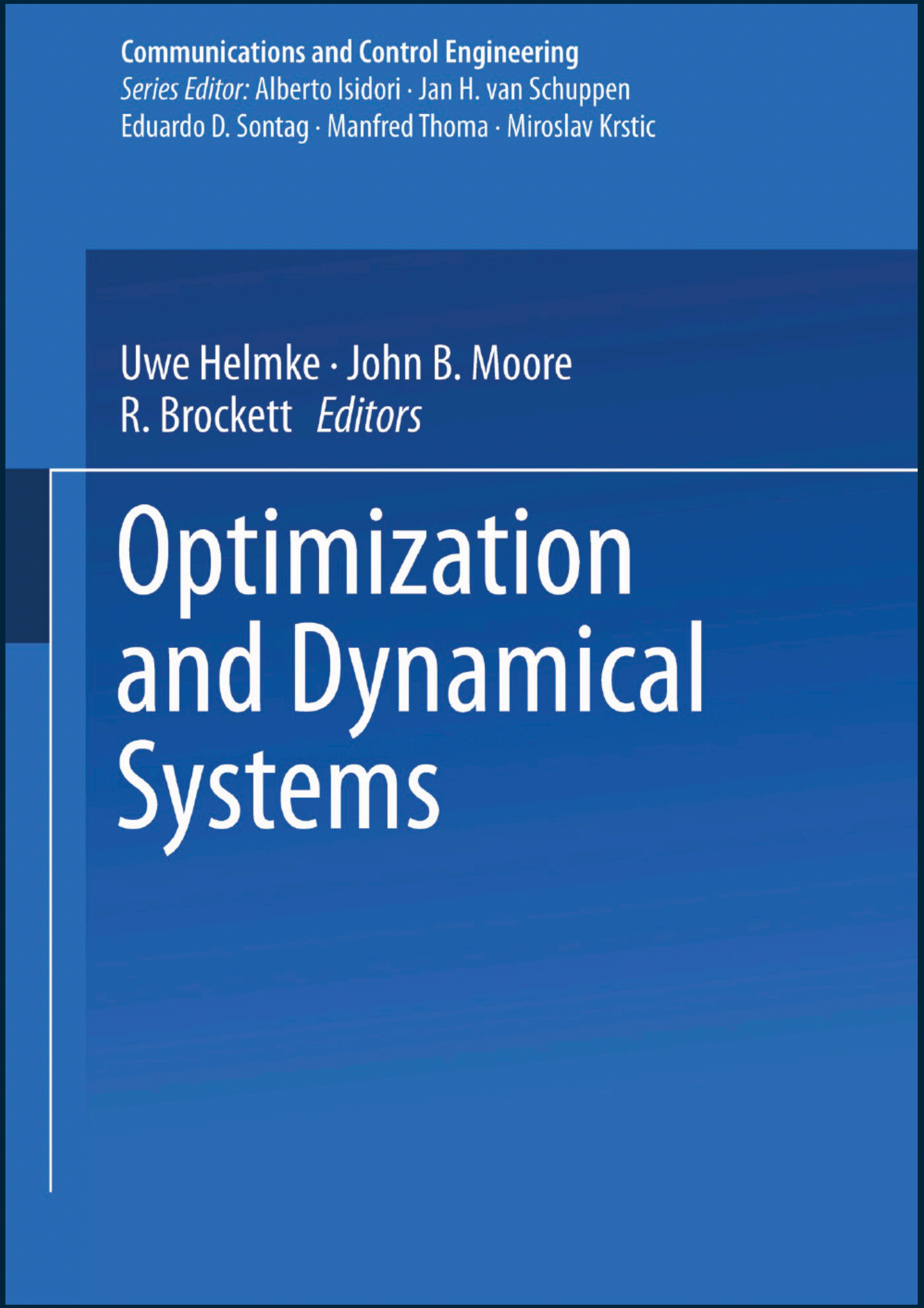
**Franz Wegner**

Institut für Theoretische Physik, Ruprecht-Karls-Universität, D-69120 Heidelberg, Germany

**1 Introduction**

In order to solve a many-particle problem one would like to diagonalize the Hamiltonian and to calculate from the eigenstates and their energies the observables of interest. However, apart from a few explicitly solvable models this is not possible. Therefore one may try – and this will be done here – to transform the Hamiltonian in such a way that it becomes more and more diagonal. More precisely unitary transformations will be applied to the Hamiltonian so that the off-diagonal matrix elements become smaller and smaller.

As we will see in explicit examples there will be difficulties to reach the goal of



Communications and Control Engineering
*Series Editor:* Alberto Isidori · Jan H. van Schuppen
Eduardo D. Sontag · Manfred Thoma · Miroslav Krstic

Uwe Helmke · John B. Moore
R. Brockett *Editors*

Optimization and Dynamical Systems

Systematic exposition of technical tools surrounding Brockett's flow equation

# A double-braket flow for quantum states

$$\frac{\partial \rho(t)}{\partial t} = [\rho(t), [\rho(t), -\hat{H}]]$$

Where $\hat{H}$ is the Hamiltonian of the system

Brockett, Glazek, Wilson and Wegner's results
collectively tell us that as $t$ increases,
$\rho(t)$ becomes increasingly diagonal in the basis of $\hat{H}$;
and eventually "equilibrates" to a steady, energy-incoherent state

# Double-bracket flow vs the von Neumann equation

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t), [\rho(t), \hat{H}]]$$

Non-linear

How can we **implement** this?

$$\frac{\partial \rho(t)}{\partial t} = i[\rho(t), \hat{H}]$$

Linear

Natural time-evolution

$$\rho(t) = e^{-i\hat{H}t}\rho(0)e^{i\hat{H}t}$$

# Outline

What is a double-bracket flow equation?

How do we implement a double-bracket flow algorithmically?

What is the steady state of a double-bracket flow?

What are the implications for the above?

# A recipe to approximately implement DBF

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t), [\rho(t), \hat{H}]]$$

*M. Gluza, Quantum 8, 1316 (2024)*

1) Assume first that $\rho(t) = |\Psi(t)\rangle\langle\Psi(t)|$ is pure. Note that we can then rewrite the double-bracket equation into its action on the pure state,

$$\partial_t |\Psi(t)\rangle = [\rho(t), \hat{H}] |\Psi(t)\rangle$$

2) For short times $\Delta t$, we have that

$$|\Psi(\Delta t)\rangle \approx \boxed{e^{\Delta t[\rho(0),\hat{H}]}} |\Psi(0)\rangle$$

Unitary, since $[\rho(0), \hat{H}]$ is anti-Hermitian

# A recipe to approximately implement DBF

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t), [\rho(t), \hat{H}]]$$

*M. Gluza, Quantum 8, 1316 (2024)*

3) Recall product formulas for small $s$,

$$e^{s[A,B]} \approx e^{i\sqrt{s}B}\ e^{i\sqrt{s}A}\ e^{-i\sqrt{s}B}\ e^{-i\sqrt{s}A}\ +\ \mathcal{O}(s^{3/2})$$

4) Note that the above, applied to our exponential map $e^{\Delta t[\rho,\hat{H}]}$ gives

Backward and forward usual time evolution

$$e^{\Delta t[\rho(0),\hat{H}]} \approx \boxed{e^{i\sqrt{\Delta t}\hat{H}}}\boxed{e^{i\sqrt{\Delta t}\rho(0)}}\boxed{e^{-i\sqrt{\Delta t}\hat{H}}}\boxed{e^{-i\sqrt{\Delta t}\rho(0)}}\ +\ \mathcal{O}(\Delta t^{3/2})$$

“Reflection operators around $\rho(0)$”

# A recipe to approximately implement DBF

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t),[\rho(t),\hat{H}]]$$

*M. Gluza, Quantum 8, 1316 (2024)*

5) In the next time step, the map needs to be updated to $e^{\Delta t[\rho(\Delta t),\hat{H}]}$ !
Generally, this gives us a recursive formula

$$|\Psi(t)\rangle \;—\boxed{R_{\rho(t)}\left(-\sqrt{\Delta t}\right)}—\boxed{R_H\left(-\sqrt{\Delta t}\right)}—\boxed{R_{\rho(t)}\left(\sqrt{\Delta t}\right)}—\boxed{R_H\left(\sqrt{\Delta t}\right)}—\; \approx |\Psi(t+\Delta t)\rangle$$

* freedom to choose time step size in every iteration

In fact, the first reflector is unnecessary:

$$|\Psi(t)\rangle \;—\boxed{R_H\left(-\sqrt{\Delta t}\right)}—\boxed{R_{\rho(t)}\left(\sqrt{\Delta t}\right)}—\boxed{R_H\left(\sqrt{\Delta t}\right)}—\; \approx |\Psi(t+\Delta t)\rangle$$

# Double-bracket algorithms are an example of quantum recursions

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

$$\rho_n = \hat{U}^{(\rho_{n-1})}\rho_{n-1}(\hat{U}^{(\rho_{n-1})})^\dagger$$

Assumptions:

- The $n_{\text{th}}$ recursive step $\hat{U}^{(\rho_{n-1})}$ is a **unitary** and depends on the **previous result** $\rho_{n-1}$.
- Initial state $\rho_0$ and $\hat{U}^{(\rho_0)}$ are known and given
- The exact form of $\hat{U}^{(\rho)}$ as a function of $\rho$ is known

# Naive implementation (unfolding) leads to exponential circuit depth

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

Let's look at the reflection $\hat{U}^{(\rho)} = e^{is\rho}$

*Remember: we assume we are given $\rho_0$ and $\hat{U}^{(\rho_0)}$, our goal is to implement $\hat{U}^{(\rho_1)}$*

- Observe that since we do know $\rho_0$, and furthermore $\rho_1 = \hat{U}^{(\rho_0)}\ \rho_0\ (\hat{U}^{(\rho_0)})^\dagger$, we also have that $e^{is\rho_1} = \hat{U}^{(\rho_0)} e^{is\rho_0} (\hat{U}^{(\rho_0)})^\dagger$, and $\hat{U}(\rho_1)$ is possible to implement without knowing $\rho_1$ — we just need to call the unitary $\hat{U}^{(\rho_0)}$ and its inverse.
- This example is trivial on its own….
- However, $\hat{U}^{(\rho)} = V_1 \cdot e^{is\rho} \cdot V_2$ is not so trivial, and yet we still have $\hat{U}^{(\rho_1)} = \hat{U}^{(\rho_0)} \cdot e^{is\rho_0} \cdot \hat{U}^{(\rho_0)^\dagger}$.
- For a more general unitary $\hat{U}^{(\rho)} = \hat{V}_L e^{is_L\rho} \hat{V}_{L-1} \cdots \hat{V}_1 e^{is_1\rho} \hat{V}_0$, we still can do this: $\hat{U}^{(\rho_1)}$ is possible if we call the unitary $\hat{U}^{(\rho_0)}$ a number of $2L$ times.

# Naive implementation (unfolding) leads to exponential circuit depth

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

Sticking with the more general example of $\hat{U}^{(\rho)} = \hat{V}_L e^{is_L\rho}\hat{V}_{L-1}\cdots\hat{V}_1 e^{is_1\rho}\hat{V}_0$, to execute $\hat{U}^{(\rho_2)}$,

- we call $\hat{U}^{(\rho_1)}$ a number of $2L$ times,
- each call for $\hat{U}^{(\rho_1)}$ is done by calling $\hat{U}^{(\rho_0)}$ for $2L$ times,
- Hence, the implementation of $\hat{U}^{(\rho_2)}$ requires $4L^2$ calls of $\hat{U}^{(\rho_0)}$. Similarly, $\hat{U}^{(\rho_3)}$ requires $8L^3$ calls of $\hat{U}^{(\rho_0)}$…
  - In other words, $e^{is\rho_n} = \hat{U}^{(\rho_{n-1})}\ldots\hat{U}^{(\rho_0)}e^{is\rho_0}(\hat{U}^{(\rho_0)})^\dagger\ldots(\hat{U}^{(\rho_{n-1})})^\dagger$
- $O((2L)^N)$ steps (i.e. circuit depth) is required for $\hat{U}^{(\rho_{N-1})}$

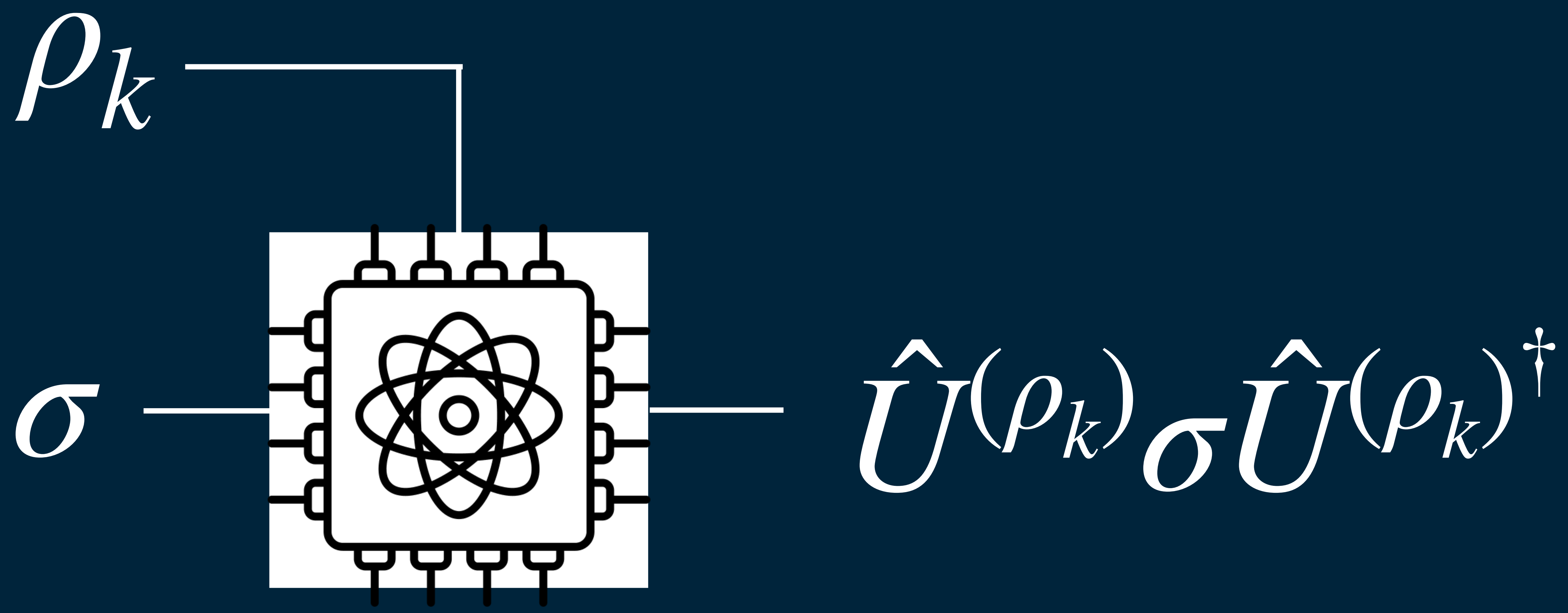
unfolding, reminiscent of classical naive methods without memory

Surely it'll be faster to compute this if we have a dynamic version of quantum computing, making use of quantum memory?

# Building a dynamic implementation via quantum instructions

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

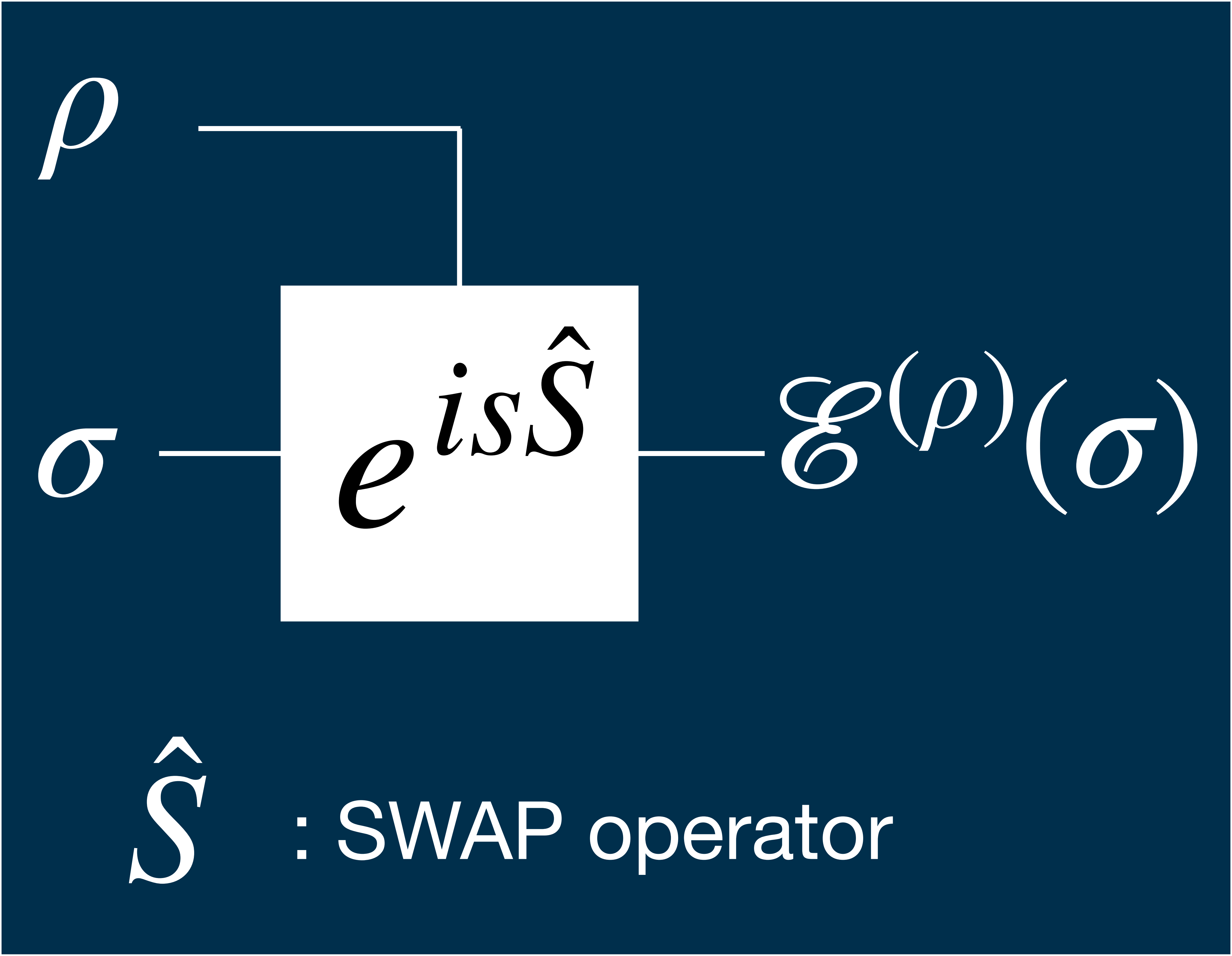
To do so, we need to consider circuits that allow us to take in **instructions encoded in the form of quantum states** $\rho_k$,



# Density Matrix Exponentiation

*Lloyd, Mohseni, and Rebentrost, Nat. Phys.* ***10****, 631 (2014)*

*Suppose we want* $\hat{U}^{(\rho)}\sigma\hat{U}^{(\rho)\dagger}$
*where* $\hat{U}^{(\rho)} = e^{is\rho}$



$\hat{S}$ : SWAP operator

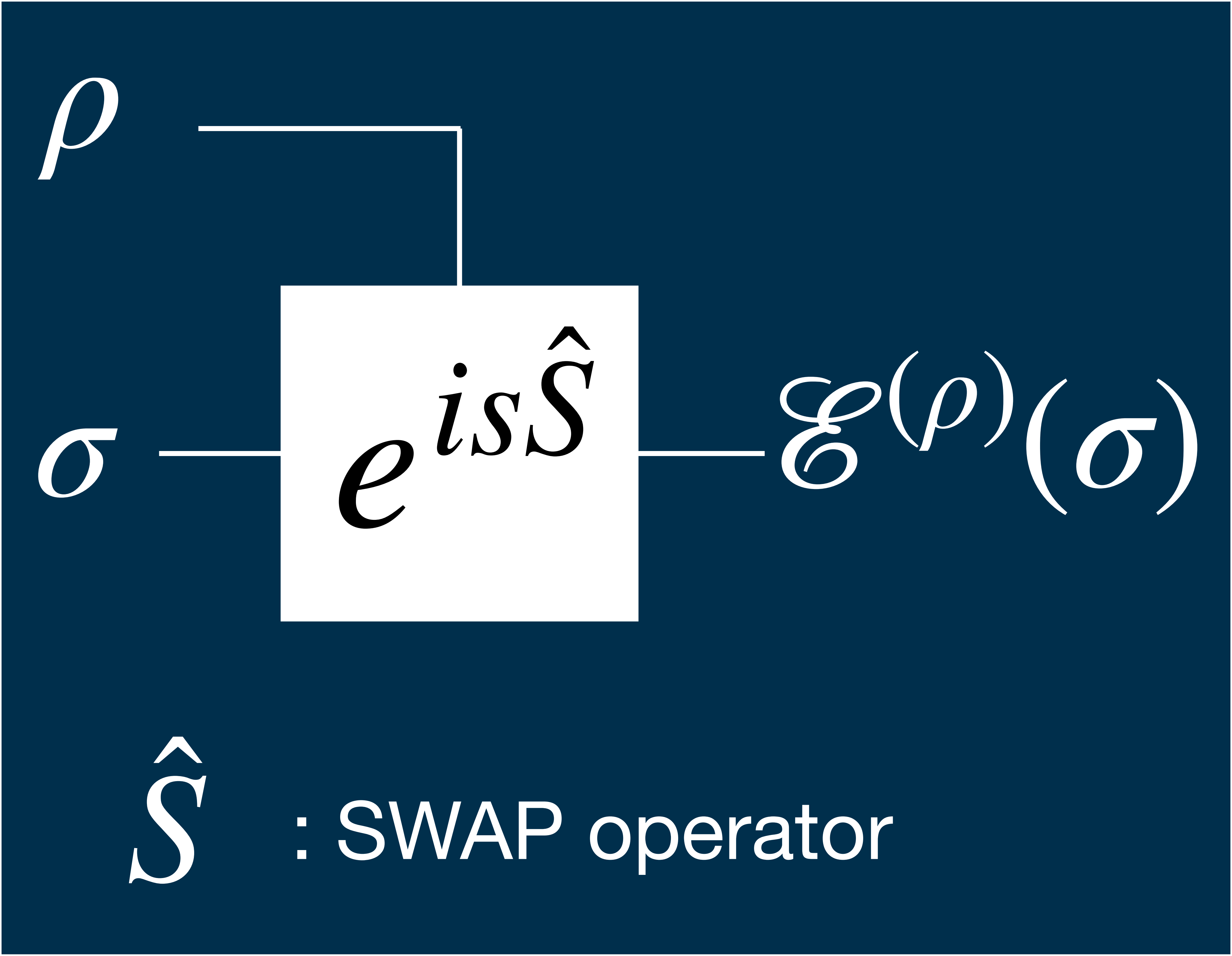
One can show that

$$\|\mathscr{E}_s^{(\rho)}(\,\bullet\,) - e^{-is\rho}(\,\bullet\,)e^{is\rho}\| = O(s^2)$$

# Density Matrix Exponentiation

*Lloyd, Mohseni, and Rebentrost, Nat. Phys.* ***10****, 631 (2014)*

*Suppose we want* $\hat{U}^{(\rho)}\sigma\hat{U}^{(\rho)^\dagger}$
*where* $\hat{U}^{(\rho)} = e^{is\rho}$



$\hat{S}$ : SWAP operator

Improvement via DME shown for sample complexity in Hamiltonian simulation, in comparison to tomographic methods

*Kimmel et al., npj QI* ***3****, 13 (2017)*

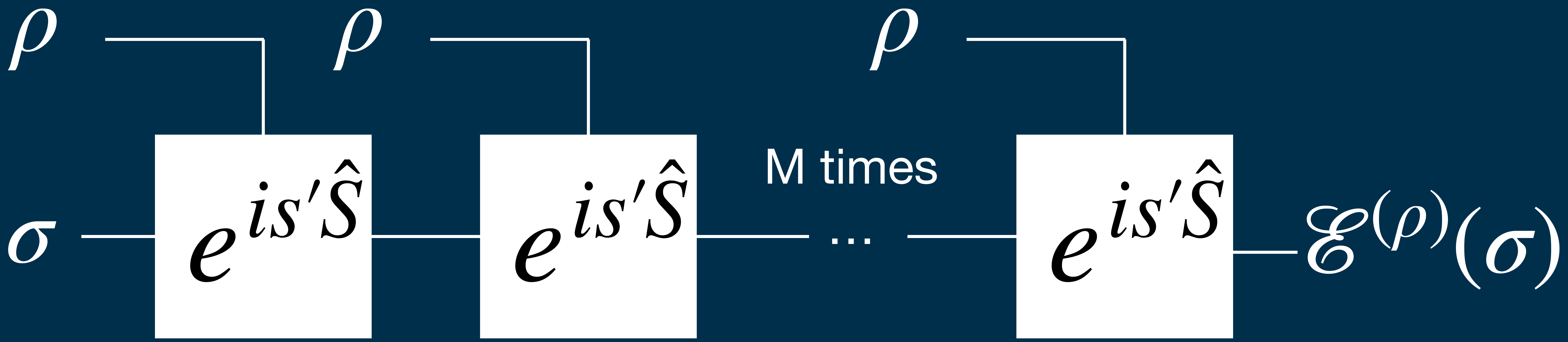
Generalizations to Hermitianicity-preserving maps exist

*Wei. et al, arXiv:2308.07956 (2023)*

# Density Matrix Exponentiation

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

To achieve a higher accuracy, one can use more copies, e.g. $M$ copies of $\rho$, with smaller values of $s' = s/M$:



$$\|\mathcal{E}^{(\rho)}_{s/M} \circ \cdots \circ \mathcal{E}^{(\rho)}_{s/M}(\,\bullet\,) - e^{-is\rho}(\,\bullet\,)e^{is\rho}\| = O(s^2/M)$$ , or $$\epsilon \propto M^{-1}$$

**Long story short:**

**Exponential circuit depth of double-brackets can be mitigated by a quantum notion of dynamic programming (QDP),**
**and this comes at the expense of a higher circuit width.**

**For quantum recursions, QDP provides flexibility to mitigate tradeoffs between depth vs width requirements**

*Son, Gluza, Takagi, N.Ng, Phys. Rev. Lett. 134, 180602 (2025)*

# Outline

What is a double-bracket flow equation?

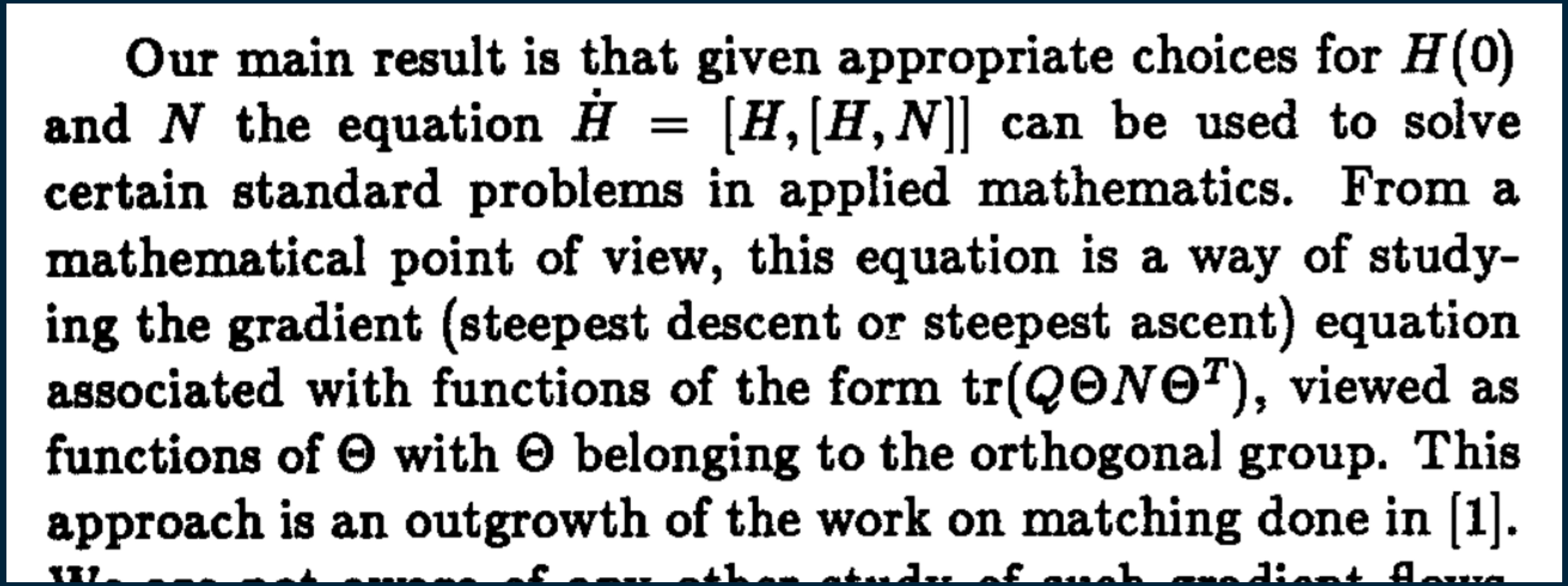
How do we implement a double-bracket flow algorithmically?

What is the steady state of a double-bracket flow?

What are the implications for the above?

# Double-bracket flow is a Riemannian gradient descent for minimizing energy

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*



Our main result is that given appropriate choices for $H(0)$ and $N$ the equation $\dot{H} = [H,[H,N]]$ can be used to solve certain standard problems in applied mathematics. From a mathematical point of view, this equation is a way of studying the gradient (steepest descent or steepest ascent) equation associated with functions of the form $\mathrm{tr}(Q\Theta N\Theta^T)$, viewed as functions of $\Theta$ with $\Theta$ belonging to the orthogonal group. This approach is an outgrowth of the work on matching done in [1]. We are not aware of any other study of such gradient flows

*Brocket, 1991*

The double-bracket flow equation

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t),[\rho(t),\hat{H}]]$$

is a gradient descent where the cost function is given as the average energy,

$$\frac{\partial \rho(t)}{\partial t} = -[\rho(t),[\rho(t),\hat{H}]] = -\,\mathrm{grad}_{\rho(t)}\frac{1}{2}\|\rho(t)-\hat{H}\|^2_{\mathrm{HS}} = -\,\mathrm{grad}_{\rho(t)}\,\langle \hat{H}\rangle_t$$

# Imaginary-time evolution is a framework for minimizing energy

*M.Gluza,, J.Son, B.H.Tiang,, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

$$|\Psi(t)\rangle = \frac{e^{-t\hat{H}}|\Psi_0\rangle}{\|e^{-t\hat{H}}|\Psi_0\rangle\|}, t \in \mathbb{R}$$

Are DBFs and ITE related to one another?

If so, how exactly?

Note that for $t \to \infty$, all higher energy populations become suppressed by $e^{-t\hat{H}}$, so the renormalized state should converge to the ground state of the system.

ITE is a standard paradigm for ground state preparation, but recipe for implementation is not always clear, and usually it is compiled variationally…

# Imaginary-time evolution is the solution for double-bracket flow equation

*M.Gluza,, J.Son, B.H.Tiang,, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

$$\frac{\partial \rho(t)}{\partial t} = [\rho(t), [\rho(t), -\hat{H}]]$$

→

Imaginary-time evolution is a solution!

$$|\Psi(t)\rangle = \frac{e^{-t\hat{H}}|\Psi_0\rangle}{\|e^{-t\hat{H}}|\Psi_0\rangle\|}, t \in \mathbb{R}$$

Assuming the initial state $\rho(0) = |\Psi(0)\rangle\langle\Psi(0)|$ is pure

Implications: The explicit recipe for DBQAs can be used to implement ITE.

# Guarantees for ITE energy decrease: fluctuation-refrigeration relation

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

Taking the ITE state

$$|\Psi(\tau)\rangle = \frac{e^{-\tau\hat{H}}|\Psi_0\rangle}{\|e^{-\tau\hat{H}}|\Psi_0\rangle\|}, \tau \in \mathbb{R}$$

*(We switch $t$ to $\tau$ to remind that this is not real “time” as we physically think of)*

A direct computation shows that

$$\frac{\partial E(\tau)}{\partial \tau} = -2V(\tau),$$

Where $E(\tau), V(\tau)$ are the average energy and energy fluctuations of $|\Psi(\tau)\rangle$

**The recipe for implementing ITE with double-brackets should give us something similar…**

# Fluctuation-refrigeration relation for DB-QITE

*M.Gluza, , J.Son, B.H.Tiang, , R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

Initialize our system in some $|\Psi_0\rangle$, and apply the recursive DB approximate implementation for ITE: in particular, in the $k$-th step, for a time step of $\Delta\tau = s_k$,

$$|\Psi_k\rangle \;\text{—}\; \boxed{R_H\left(-\sqrt{s_k}\right)} \;\text{—}\; \boxed{R_{\rho_k}\left(\sqrt{s_k}\right)} \;\text{—}\; \boxed{R_H\left(\sqrt{s_k}\right)} \;\text{—}\; |\Psi_{k+1}\rangle$$

We get $$E_{k+1} \leq E_k - 2s_kV_k + \mathcal{O}(s_k^2)$$

In particular, if the step sizes are chosen such that $s_k \leq 2V_k \cdot \left[5\epsilon_k||\hat{H}||^4\right]^{-1}$, then

$$E_{k+1} \leq E_k - s_kV_k$$

# Guarantees of fidelity convergence for DB-QITE

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

Given the system Hamiltonian $\hat{H}$, initialize system in some $|\Psi_0\rangle$, which has a fidelity $F_0$ to the ground state of $\hat{H}$.

Then there is a way to choose the time steps in the DB-QITE algorithm, such that the fidelity to ground state at step $k$,

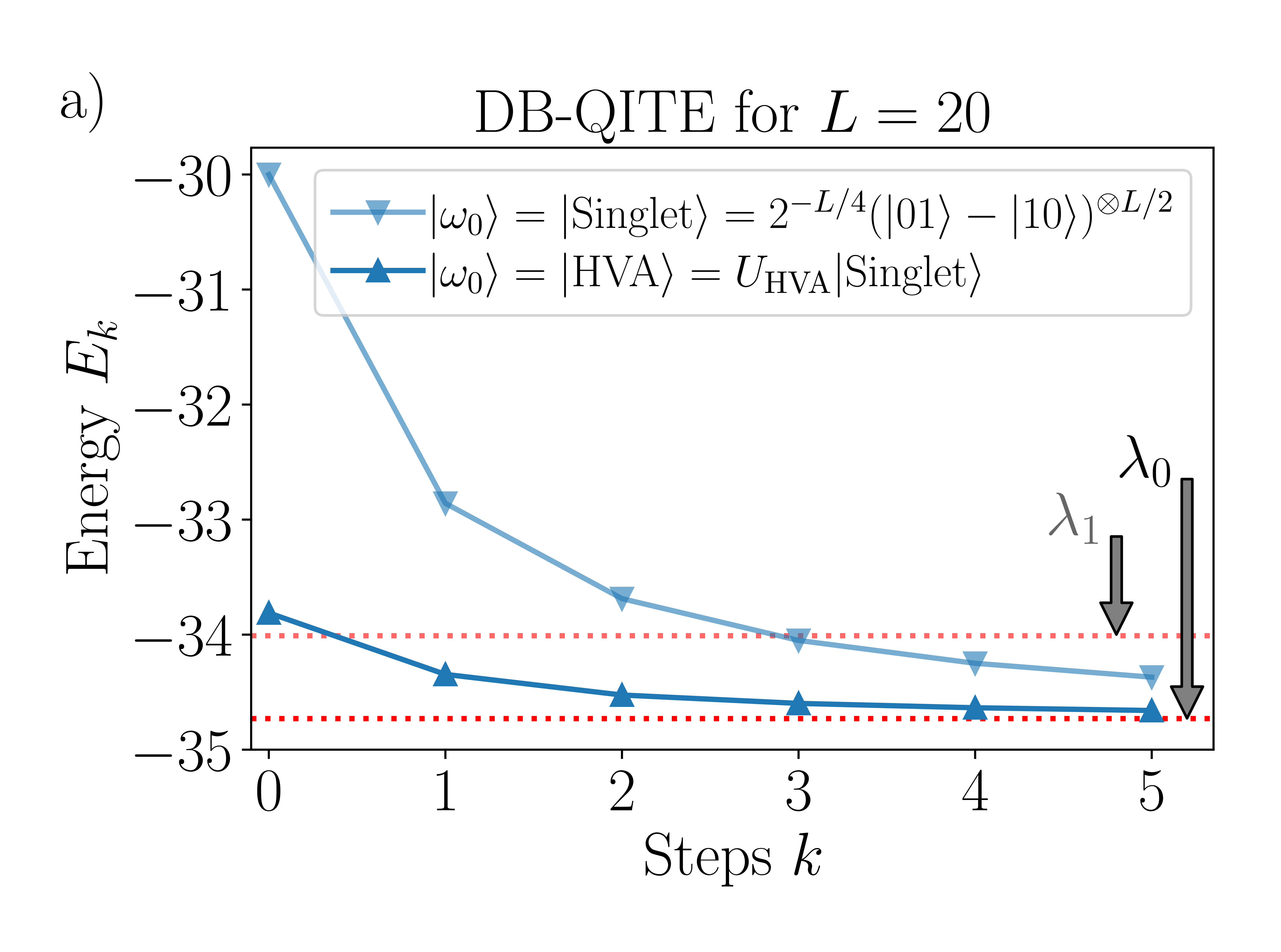
$$F_k \geq 1 - q^k, \qquad \text{for } q = 1 - \frac{\Delta^2 F_0}{12||\hat{H}||^3}$$

$\Delta$: Spectral gap

(Exponential convergence)

# Numerical performance of DB-QITE

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

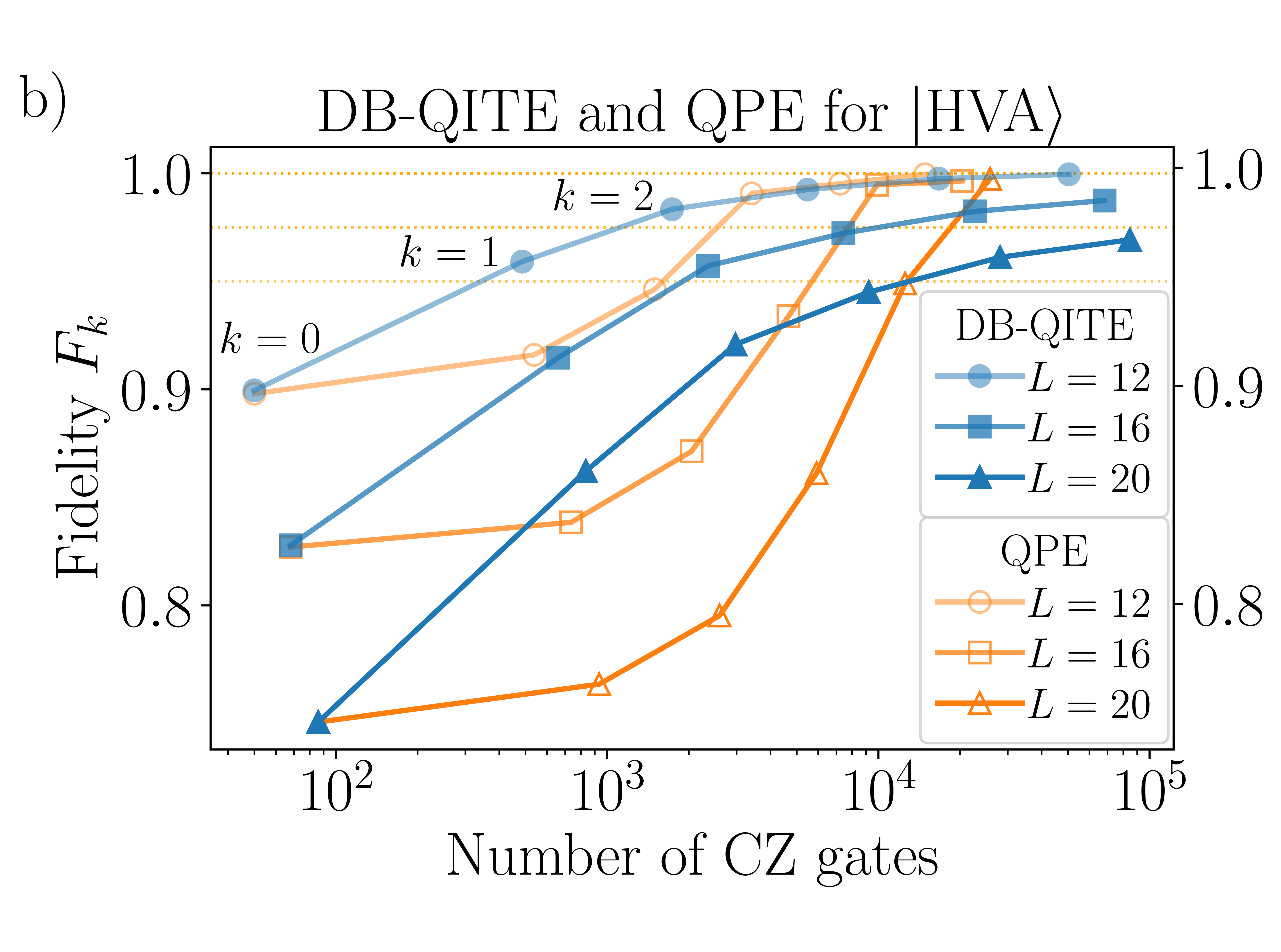


1d Heisenberg model for L=20 qubits, where $|\text{HVA}\rangle$ is a warm-start state output from a variational circuit. $\lambda_0, \lambda_1$ are ground state and first excited state energies.

- systematic decrease of energy observed as predicted, for small number of iterations.

# Pitting DB-QITE numerically against Quantum Phase Estimation

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*

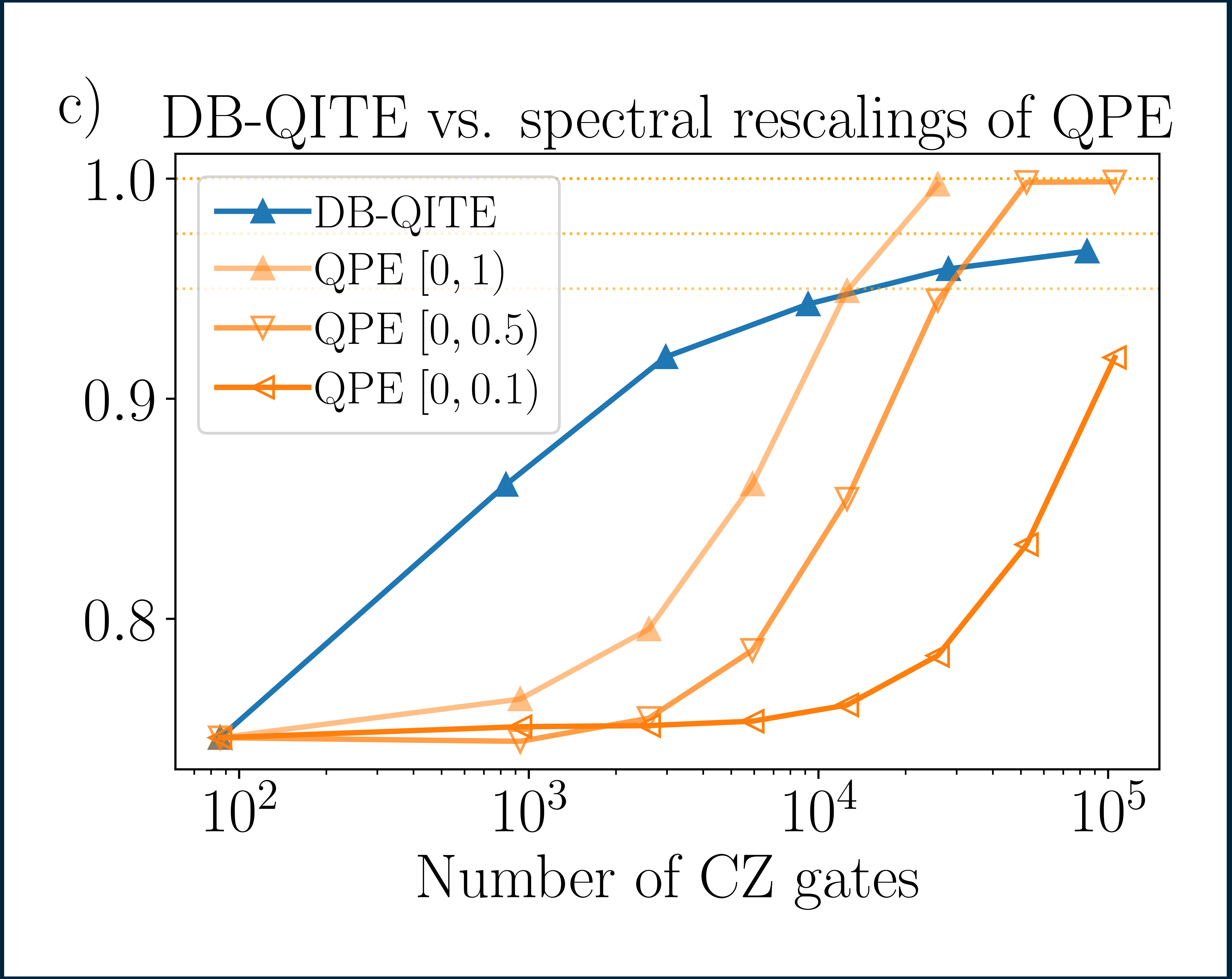


Observations:

- In the low CZ count regime (for near-term processors), DB-QITE outperforms QPE
- In high CZ count regime (e.g. fault-tolerant scenarios), QPE outperforms* DB-QITE
  - Assumptions*: favourable conditions for QPE, e.g. one knows the value of $\lambda_0$.

# Pitting DB-QITE numerically against Quantum Phase Estimation

*M.Gluza, J.Son, B.H.Tiang, R. Zander, R. Seidel, Y.Suzuki, Z.Holmes, N.Ng. arXiv:2412.04554*



Observations:

- Deterioration of QPE performance when ignorance of $\lambda_0$ leads to rescaling of the Hamiltonian spectrum
- DB-QITE unaffected.

# Outline

What is a double-bracket flow equation?

How do we implement a double-bracket flow algorithmically?

What is the steady state of a double-bracket flow?

What are the implications for the above?

Can DBQAs do other things?

How are they related to other known types of quantum algorithms?

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza,, Z.Holmes. arXiv:2504.01077*

Quantum signal processing: the action of a polynomial function $p(\hat{H})$,

$$|\Psi'\rangle = \frac{p(\hat{H})\,|\Psi\rangle}{\|p(\hat{H})\,|\Psi\rangle\|}$$

for some Hermitian $\hat{H}$

A general, $K$-degree polynomial $p(\hat{H})$ can be written as

$$p(\hat{H}) = \prod_{k=1}^{K} (H - \alpha_k I),$$

where $\alpha_k$ are complex numbers

Ex:

Real-time evolution: $p(\hat{H}) \approx \exp(i\hat{H}t)$

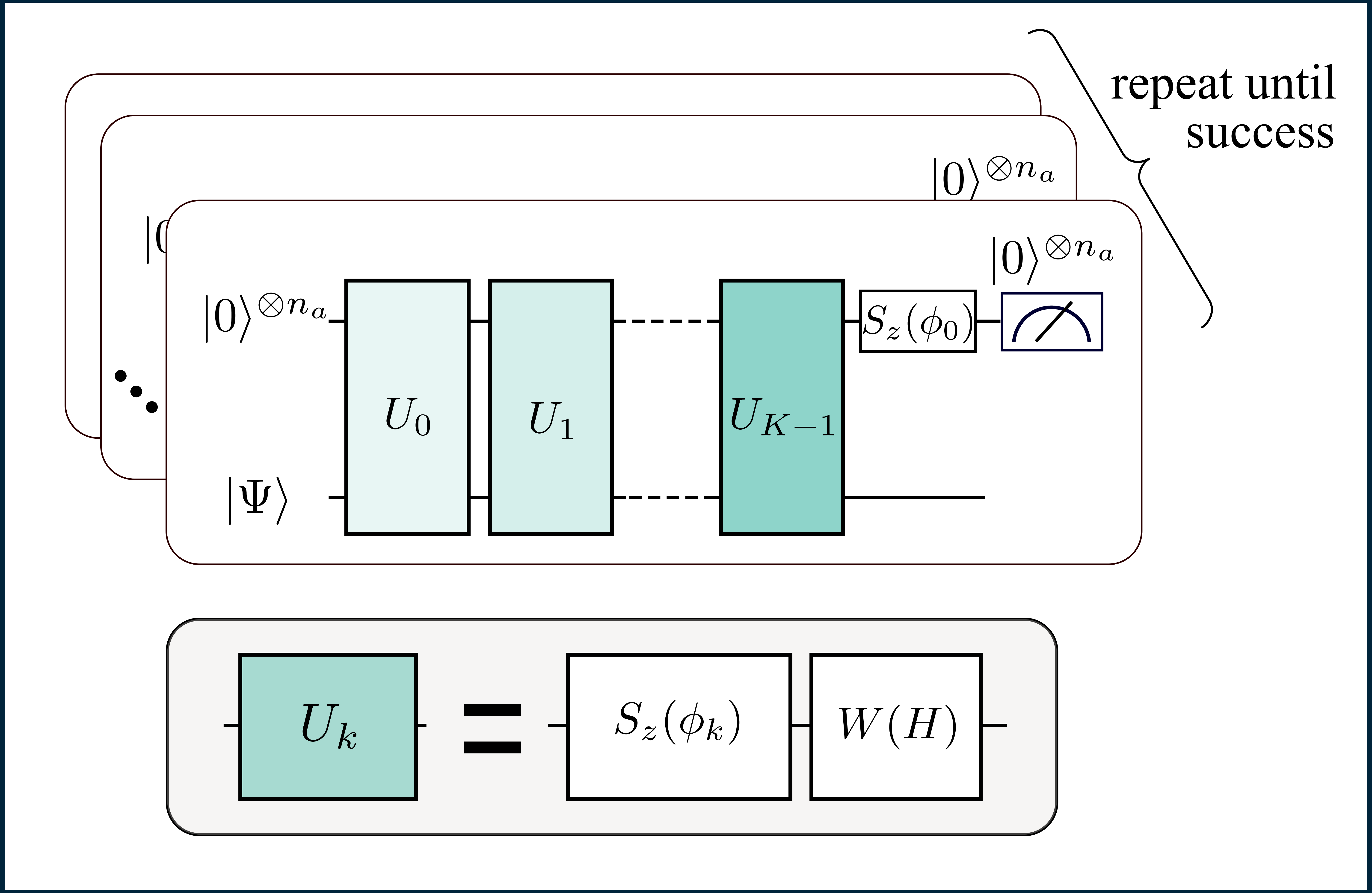
Imaginary-time evolution: $p(\hat{H}) \approx \exp(-\hat{H}\tau)$

Matrix inversion: $p(\hat{H}) \approx \hat{H}^{-1}$

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077*

Typical implementation of QSP : qubitization, linear combination of unitaries (LCU)



Signal operators $W(H)$: feed information about $H$

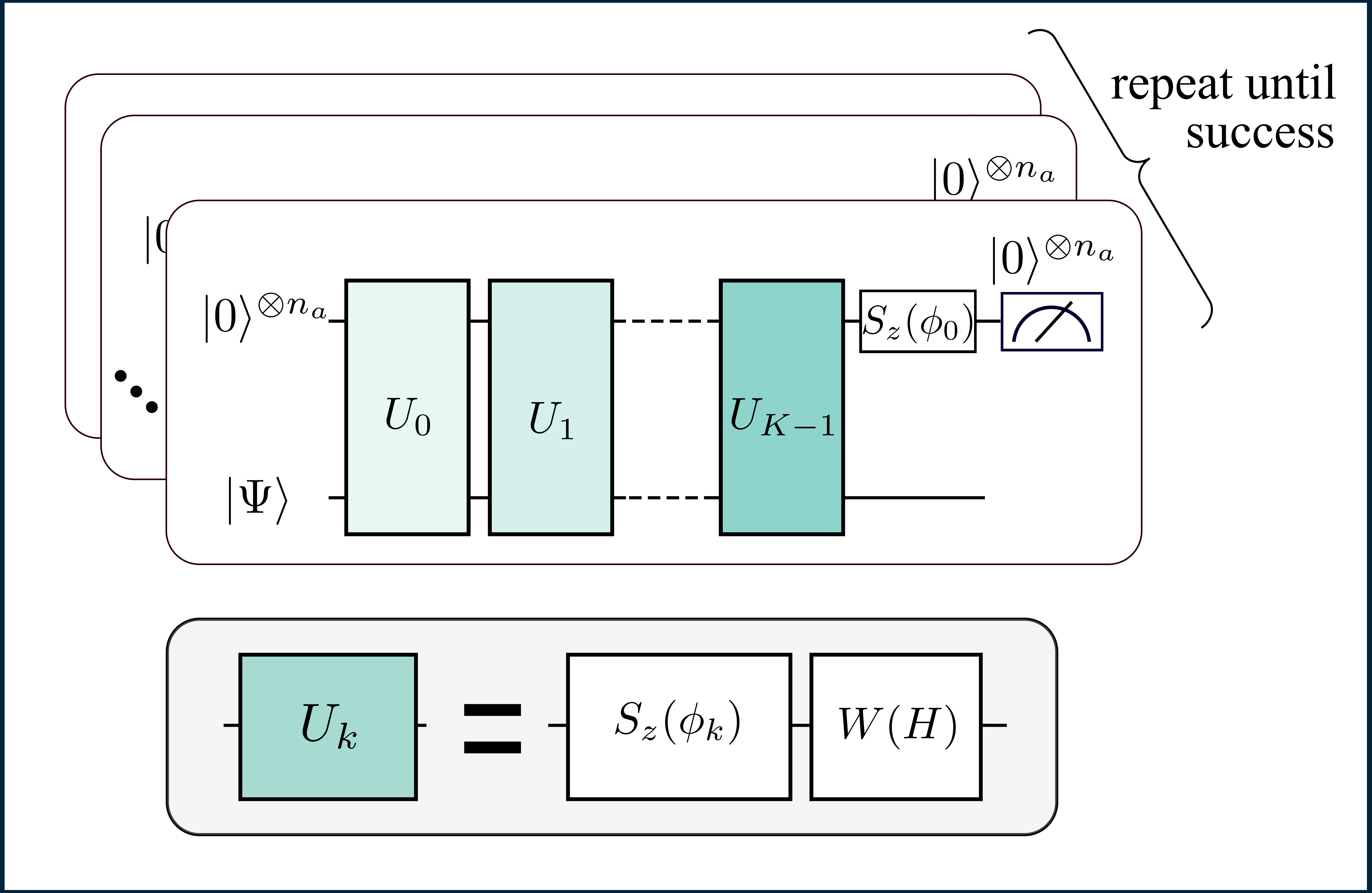
Signal processing operators $S_z(\phi_k)$: feed choice of phases that relate to roots of $p(H)$

Initialize system in $|\Psi\rangle$ with some ancillas, and at the end measure ancillas. Success can be associated with measuring 000…0 at the end.

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza,,Z.Holmes. arXiv:2504.01077*

Typical implementation of QSP : qubitization, linear combination of unitaries (LCU)



General challenge:

Post-selection often necessary, leading to probabilistic success (sometime exponentially small in system size)

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza,, Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi,\hat{H}]}|\Psi\rangle = \frac{p(\hat{H})|\Psi\rangle}{\|p(\hat{H})|\Psi\rangle\|}$$

for some value of $s_\Psi$!

In particular, $s_\Psi$ depends on the expectation value and variance of $\hat{H}$ for the state $\Psi$, i.e.

$$s_\Psi = \frac{-1}{\sqrt{V_\Psi}} \arccos\left(\frac{E_\Psi - \alpha}{\sqrt{V_\Psi + \left(E_\Psi - \alpha\right)^2}}\right)$$

Note that this is an equality, not an approximation!

In particular, $s_\Psi$ is not small, but is a value that depends on the state $\Psi$.

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi, \hat{H}]} | \Psi \rangle = \frac{p(\hat{H}) | \Psi \rangle}{\| p(\hat{H}) | \Psi \rangle \|}$$

for some value of $s_\Psi$

Proof sketch:

1) Expand the exponential,

$$e^{s_\Psi[\Psi, \hat{H}]} = \sum_{k=0}^{\infty} \frac{s^k}{k!} [\Psi, \hat{H}]^k$$

2) Look at the first order term acting on $\Psi$,

$$\begin{aligned} [\Psi, \hat{H}] | \Psi \rangle &= \Psi \hat{H} | \Psi \rangle - \hat{H} \Psi | \Psi \rangle \\ &= E_\Psi | \Psi \rangle - \hat{H} | \Psi \rangle \end{aligned}$$

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi,\hat{H}]}|\Psi\rangle = \frac{p(\hat{H})|\Psi\rangle}{\|p(\hat{H})|\Psi\rangle\|}$$ for some value of $s_\Psi$

Proof sketch:

3) Look at the second order term,

Although $([\Psi,\hat{H}])^2$ is in general arbitrary, its action on $|\Psi\rangle$ is "effectively" identity

$$[\Psi,\hat{H}]^2|\Psi\rangle = E_\Psi^2|\Psi\rangle - E_\Psi H|\Psi\rangle - \langle\Psi H^2\rangle\Psi|\Psi\rangle + E_\Psi H|\Psi\rangle$$
$$= -V_\Psi|\Psi\rangle\,.$$

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi,\hat{H}]}|\Psi\rangle = \frac{p(\hat{H})|\Psi\rangle}{\|p(\hat{H})|\Psi\rangle\|}$$

for some value of $s_\Psi$

Proof sketch:

4) This means that odd versus even terms in the exponential can be grouped into components of $|\Psi\rangle$ and $\hat{H}|\Psi\rangle$:

$$e^{s[\Psi,\hat{H}]}|\Psi\rangle = a(s, E_\Psi, V_\Psi)|\Psi\rangle + b(s, E_\Psi, V_\Psi)\hat{H}|\Psi\rangle$$

🎉

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza,,Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi,\hat{H}]} |\Psi\rangle = \frac{p(\hat{H})|\Psi\rangle}{\| p(\hat{H})|\Psi\rangle \|}$$

for some value of $s_\Psi$

To implement this exponential, we again use group commutators but don't (yet) need the recursive structure here as compared to DB-QITE, just:

$$e^{s[\Psi,\hat{H}]} = \left( e^{ia\Psi} e^{ia\hat{H}} e^{-ia\Psi} e^{-ia\hat{H}} \right)^N + O\left( s^{3/2}/\sqrt{N} \right)$$

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza,,Z.Holmes. arXiv:2504.01077*

Observation: if we want to implement a linear polynomial $p(\hat{H}) = \hat{H} - \alpha I$ for some $\alpha \in \mathbb{R}$, then we can implement this QSP task with the following unitary

$$e^{s_\Psi[\Psi,H]}|\Psi\rangle = \frac{p(\hat{H})|\Psi\rangle}{\|p(\hat{H})|\Psi\rangle\|}$$

for some value of $s_\Psi$

What if $\alpha \in \mathbb{C}$ ? Note that since $\Psi$ is pure, we have $e^{i\theta\Psi} = I + \underbrace{\boxed{(e^{i\theta} - 1)}}_{r \in \mathbb{C}}\Psi$.

I can then start with $(\hat{H} - aI)$ for some $a \in \mathbb{R}$, and observe that

$$(I - r\Psi)(\hat{H} - aI)|\Psi\rangle = \hat{H}|\Psi\rangle - \alpha|\Psi\rangle + r|\Psi\rangle\underbrace{\langle\Psi|\hat{H}|\Psi\rangle}_{E_\Psi} - \alpha r|\Psi\rangle$$

*An additional $e^{i\theta\Psi}$ allows us to address complex roots*

# An alternative framework to think about quantum computation

*Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077*

Extending to general polynomials, DBQAs provide us a recipe to implement any QSP protocol for $|\Psi'\rangle = \dfrac{p(\hat{H})|\Psi\rangle}{\|p(\hat{H})|\Psi\rangle\|}$:

*Recursiveness comes in here, in particular one needs to estimate $E_k, V_k$ in order to determine implementation parameters $s_{k+1}, \theta_{k+1}$*

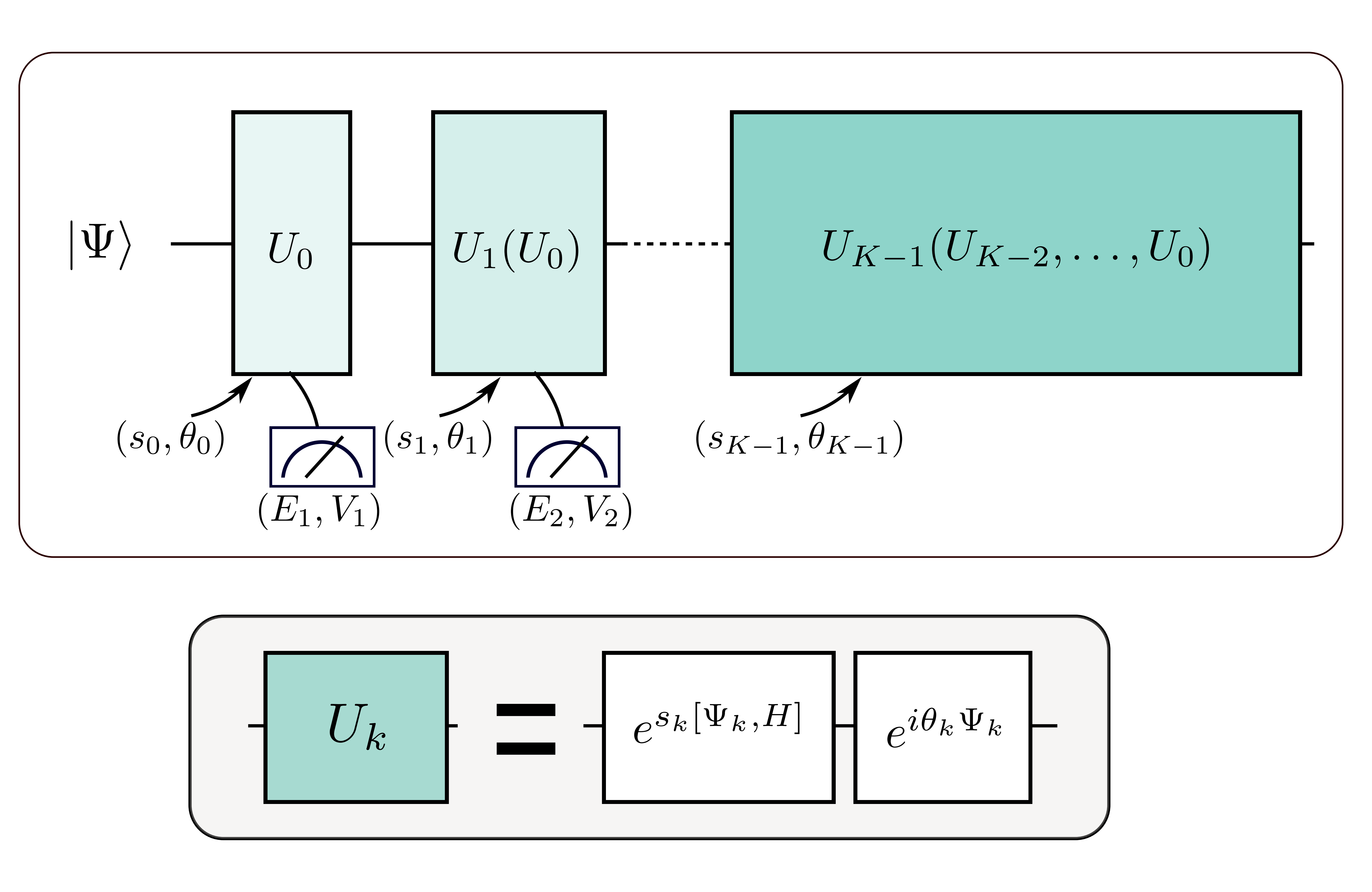
$$\prod_{k=0}^{K-1} \boxed{e^{i\theta_k \Psi_k} e^{s_k[\Psi_k, \hat{H}]}} |\Psi\rangle,$$

for some values of $\theta_k, s_k$

# An alternative framework to think about quantum computation

Y.Suzuki, B.H.Tiang, J.Son, N.Ng, M.Gluza, Z.Holmes. arXiv:2504.01077

## Summarizing DB method for QSP



No ancillas or post-selection

Instead, estimates of $E_k, V_k$ required at every turn so that $s_{k+1}, \theta_{k+1}$ can be chosen for the next iteration

Supplementary heuristics can also be used to choose $s_{k+1}, \theta_{k+1}$

# Grover's algorithm looks eeriely similar to DBQA reflectors…

*Y.Suzuki, M.Gluza, J.Son, B.H.Tiang, N.Ng, Z.Holmes. arXiv:2507.15065*

Recall that Grover's algorithm interleaves:

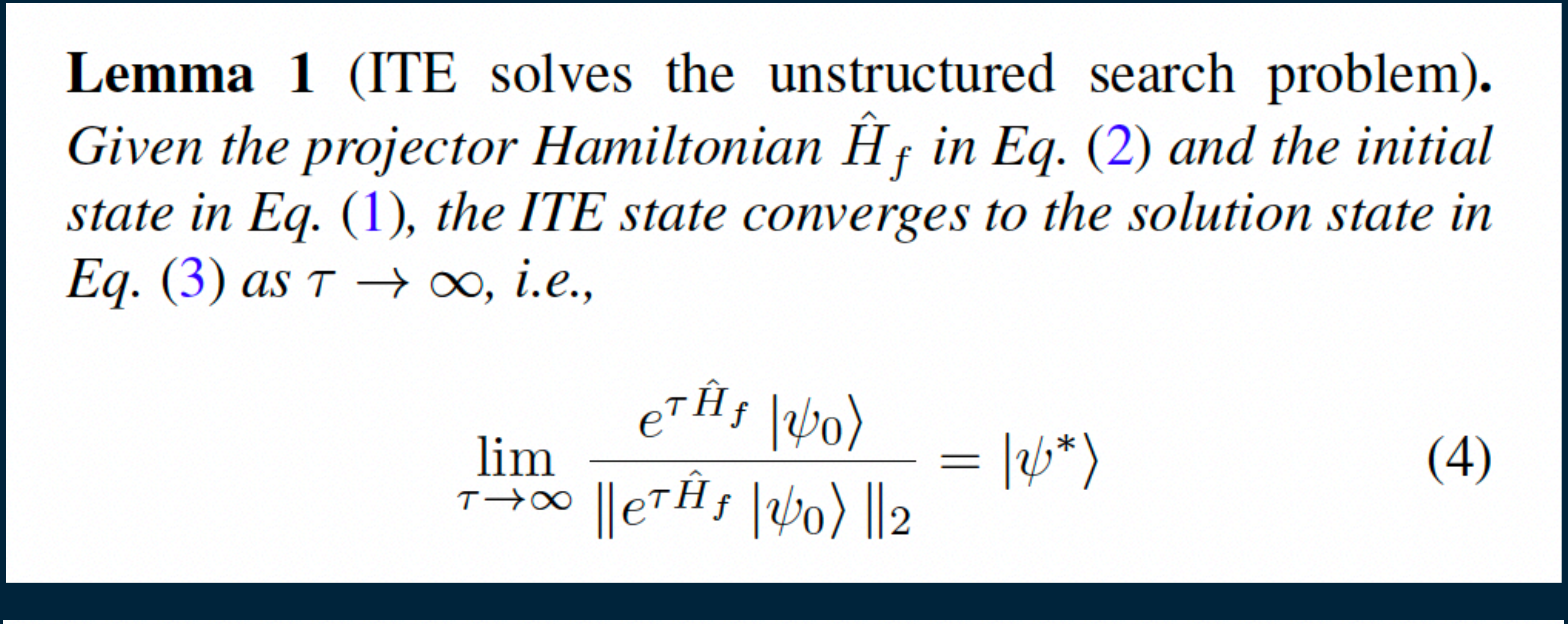
$$|\psi_0\rangle \;—\; \underbrace{\boxed{D\left(\alpha_k\right)}—\boxed{U_f(\beta_k)}}_{k\ \text{times}} \;—\; \approx |\psi^*\rangle$$

- **oracles** $U_f(\beta) = e^{i\beta H_f}$, where $H_f = \sum_{x \in \text{sol.}} |x\rangle\langle x|$ projects onto solution states, and
- **reflectors** $R(\alpha) = e^{i\alpha\psi_0}$ **around initial state**.

Nielsen and Chuang: can be seen as doing Hamiltonian simulation with $\hat{H} = H_f + H_i$ and trotterizing

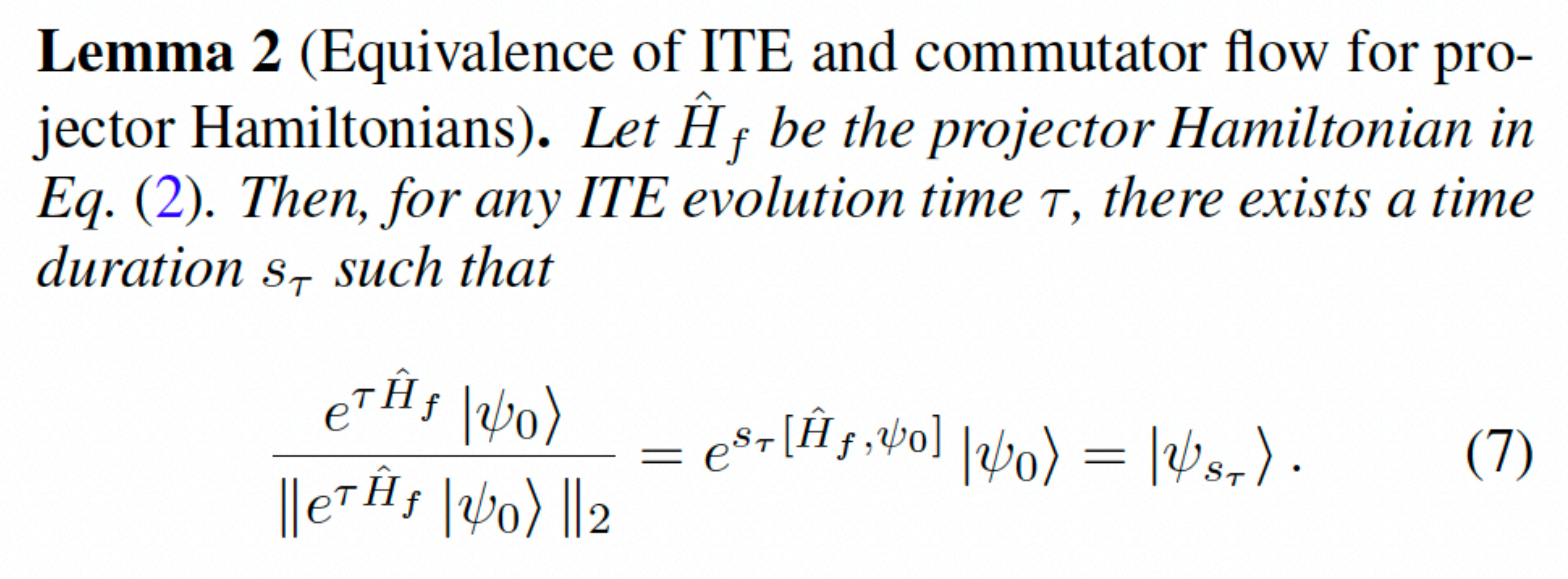
# … because it approximates DBFs!

*Y.Suzuki, M.Gluza, J.Son, B.H.Tiang, N.Ng, Z.Holmes. arXiv:2507.15065*



**Lemma 1** (ITE solves the unstructured search problem). *Given the projector Hamiltonian $\hat{H}_f$ in Eq. (2) and the initial state in Eq. (1), the ITE state converges to the solution state in Eq. (3) as $\tau \to \infty$, i.e.,*

$$\lim_{\tau\to\infty} \frac{e^{\tau \hat{H}_f} |\psi_0\rangle}{\| e^{\tau \hat{H}_f} |\psi_0\rangle \|_2} = |\psi^*\rangle \tag{4}$$



**Lemma 2** (Equivalence of ITE and commutator flow for projector Hamiltonians). *Let $\hat{H}_f$ be the projector Hamiltonian in Eq. (2). Then, for any ITE evolution time $\tau$, there exists a time duration $s_\tau$ such that*

$$\frac{e^{\tau \hat{H}_f} |\psi_0\rangle}{\| e^{\tau \hat{H}_f} |\psi_0\rangle \|_2} = e^{s_\tau [\hat{H}_f, \psi_0]} |\psi_0\rangle = |\psi_{s_\tau}\rangle \,. \tag{7}$$

# … because it approximates DBFs!

*Y.Suzuki, M.Gluza, J.Son, B.H.Tiang, N.Ng, Z.Holmes. arXiv:2507.15065*

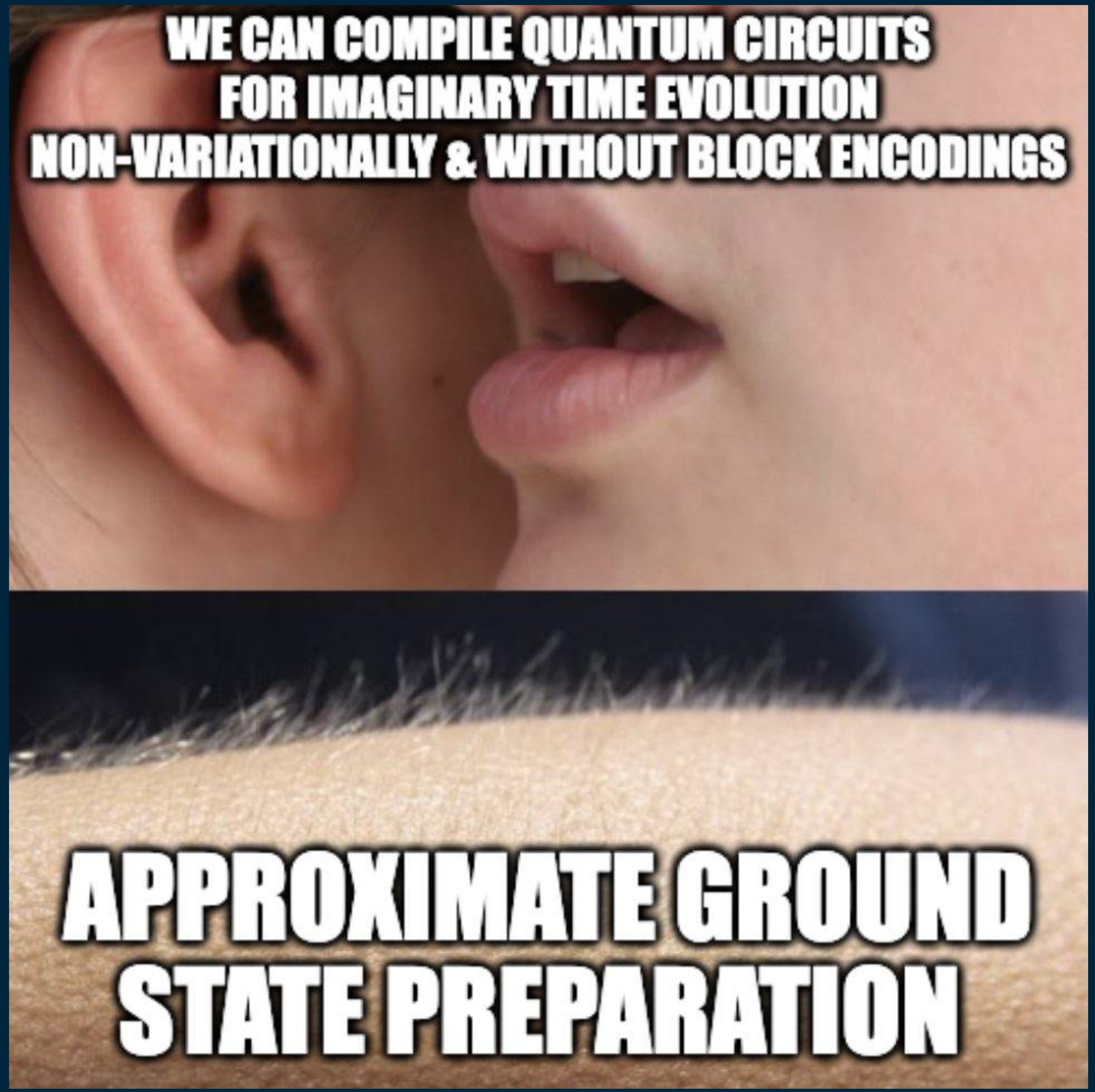
Brief recap of results:

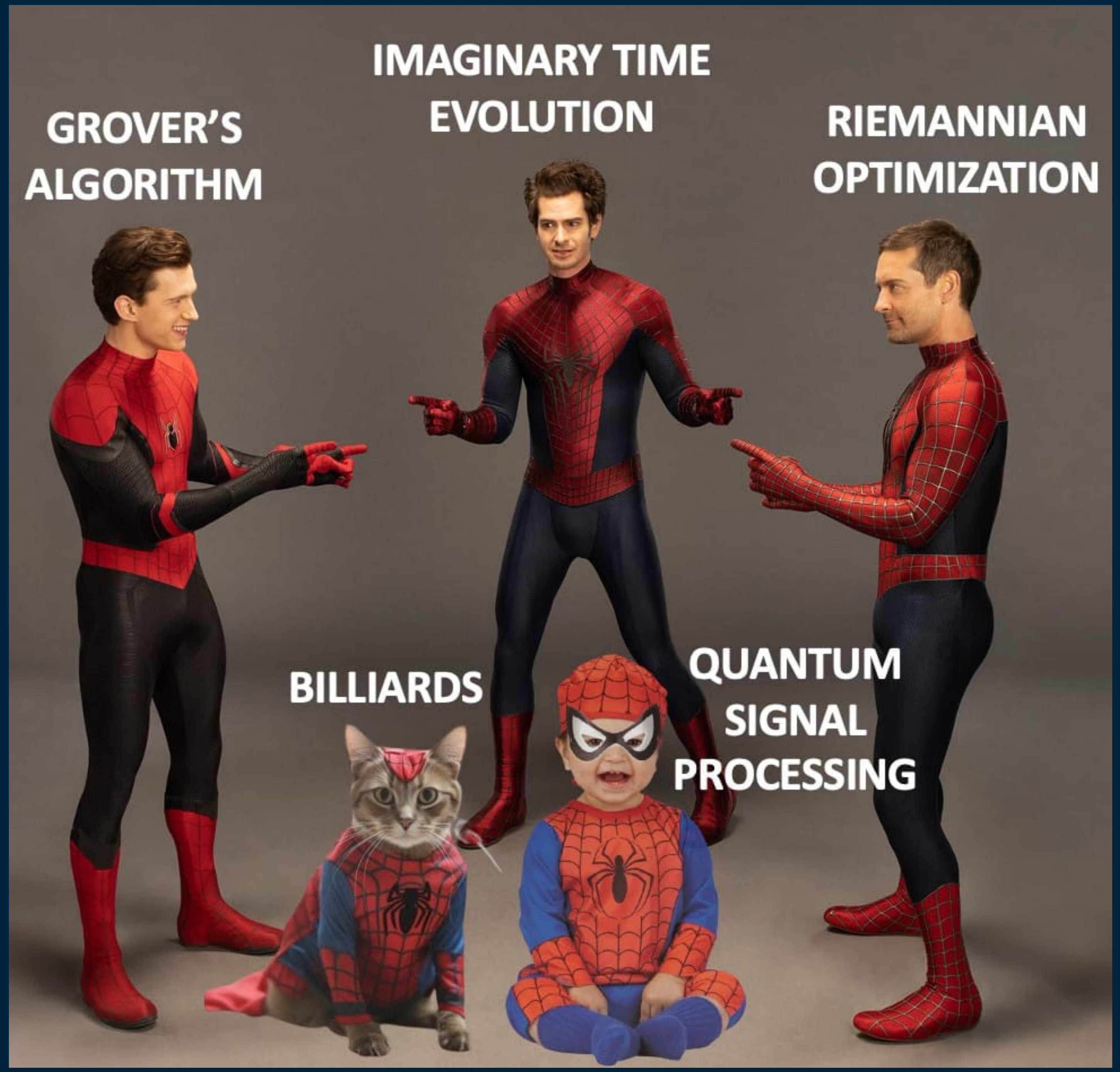
- ITE traces a geodesic for unstructured search,
- ITE's geodesic length determines query complexity of Grover's algorithm,
- Grover iterations implement a generic quantum signal processing sequence (with appropriate choices of reflection angles).

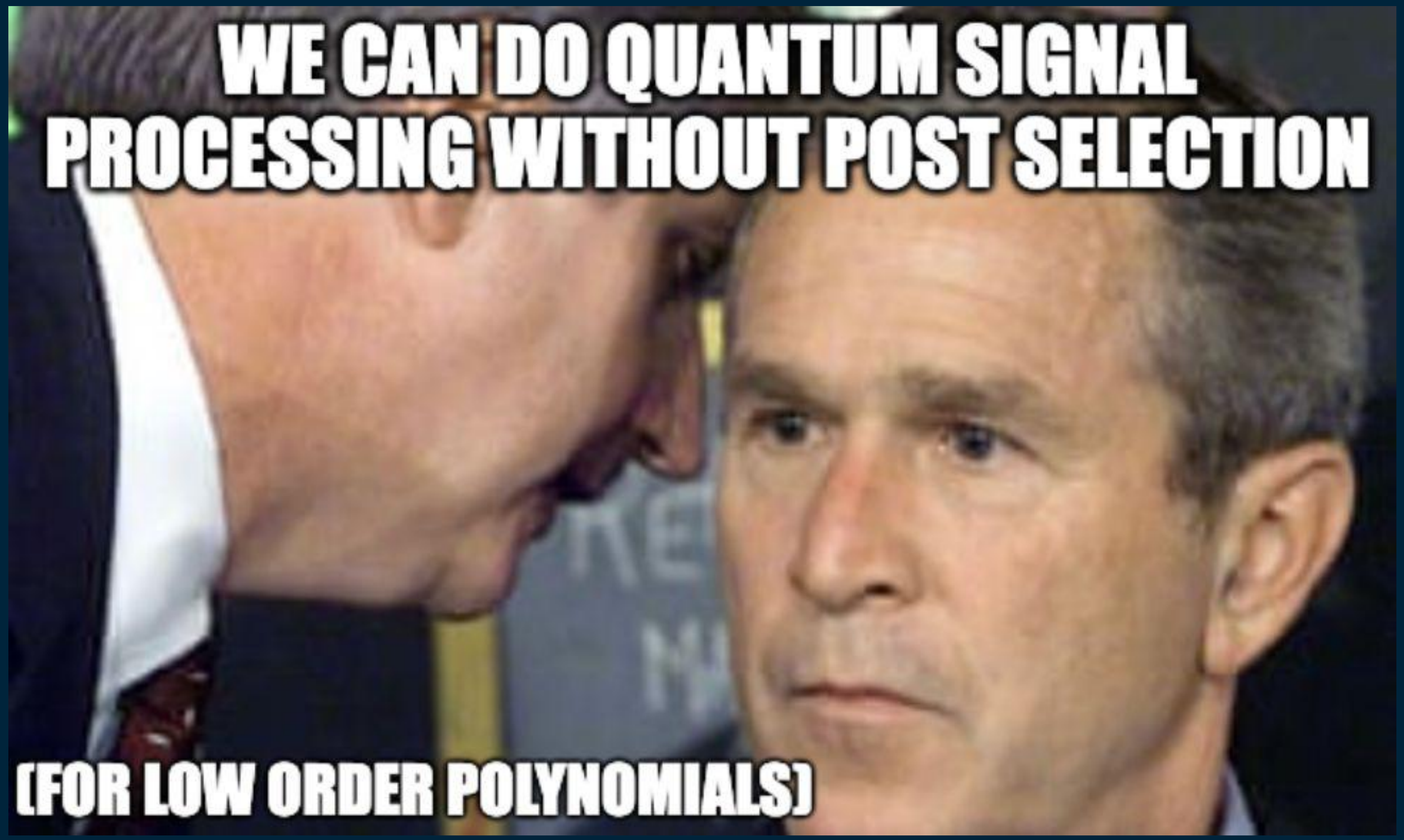
# Take home messages

ANALYTICALLY PROVE FIDELITY TENDS TOWARDS GROUND STATE
EXPONENTIALLY HARD COMPUTATION TIME
Perfectly balanced.
As all things should be.
imgflip.com

GROVER'S ALGORITHM
IMAGINARY TIME EVOLUTION
Corporate needs you to find the difference between this picture and this picture
They're the same picture [1ST ORDER]
imgflip.com


WE CAN COMPILE QUANTUM CIRCUITS FOR IMAGINARY TIME EVOLUTION NON-VARIATIONALLY & WITHOUT BLOCK ENCODINGS
APPROXIMATE GROUND STATE PREPARATION




WE CAN DO QUANTUM SIGNAL PROCESSING WITHOUT POST SELECTION
(FOR LOW ORDER POLYNOMIALS)

# Take home messages

**Thanks for listening!**
**Happy to take questions**

## What is a double-bracket flow equation?

*In the context of quantum information, it is a non-linear Schrodinger equation. A steady state solution exists and is energy-diagonal*

## How do we implement a double-bracket flow algorithmically?

*By approximating it with short-time Hamiltonian evolutions (back and forth), interleaving with reflections that depend on the state*

## What is the steady state of a double-bracket flow?

*It achieves imaginary-time evolution, and is also the steepest descent on the Riemannian manifold for minimizing system average energy.*

## What are the implications for the above?

*We can design explicit quantum algorithms that have theoretical guarantees on performance, and bypass methods such as variational strategies or post-selection. The cost (depth) can be flexibly mitigated by increasing width. Numerics demonstrate some promise. Bridges understanding between different forms of quantum algorithm design.*