# QUANTUM ALGORITHMS FOR THE QUANTUM SIMULATION OF CLOSED AND OPEN QUANTUM SYSTEMS

IAN JOEL DAVID

24 JUNE 2025

UNIVERSITY OF
KWAZULU-NATAL

NITheCS
National Institute for
Theoretical and Computational Sciences

fraqtal technologies

# WHAT IS QUANTUM SIMULATION?

- Quantum simulation involves using a controllable quantum system to study the behaviour of another quantum system that is difficult to study directly.
- Richard Feynman and Yuri Manin independently proposed the idea of using quantum simulators to simulate quantum systems.

Richard P. Feynman

Yuri I. Manin (Юрий Манин)

Yu. I. Manin, “Computable and Non-Computable,” Sovetskoe Radio, Moscow, 1980, 128 p.

R.P. Feynman, Simulating physics with computers, Internat. J. Theoret. Phys. 21 (6–7) (1982) 467–488, http://dx.doi.org/10.1007/bf02650179.

# WHY DO WE NEED QUANTUM SIMULATION?

- Unlike classical simulations, quantum simulations leverage the principles of quantum mechanics to model complex quantum phenomena that are infeasible for classical computers.

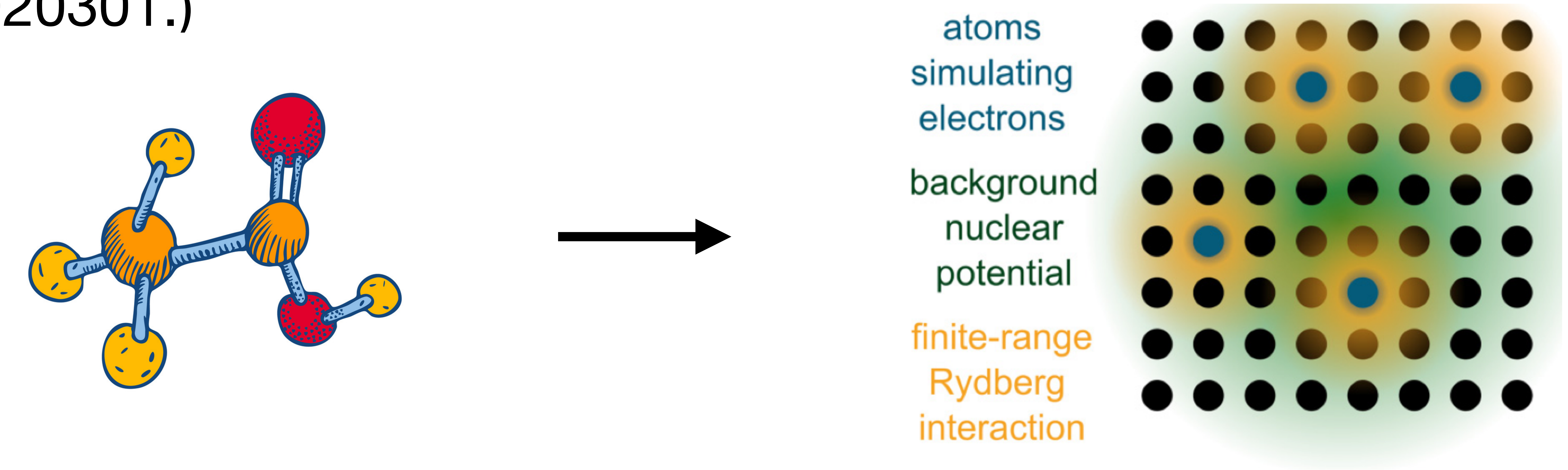
- Classical computers struggle with simulating quantum systems due to the exponential growth of the Hilbert space.

- Quantum simulations can provide insights into many-body physics, quantum chemistry, and material science.

# TYPES OF QUANTUM SIMULATION

## ANALOGUE VS DIGITAL

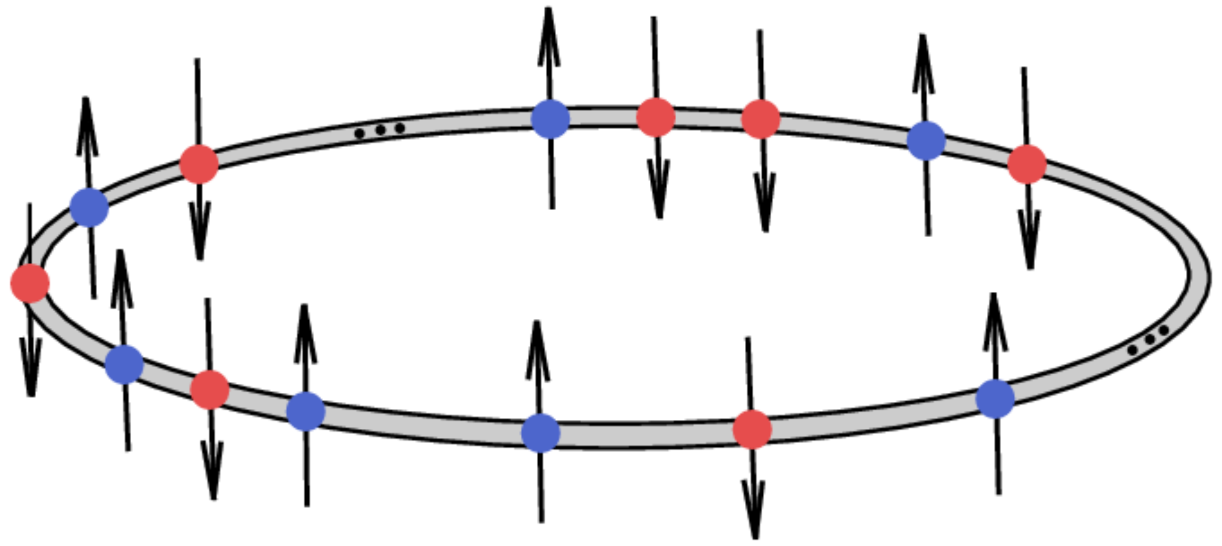
- **Analog Quantum Simulation:** Directly mimics the target quantum system using a physically similar system that is controllable.
  - This is done using analogue quantum computing which applies continuous transformations to continuous quantum states.
  - Example: Using fermionic atoms hopping in an optical latice to simulate quantum chemistry (Malz D, Cirac JI. PRX Quantum. 2023 Apr 1;4(2):020301.)

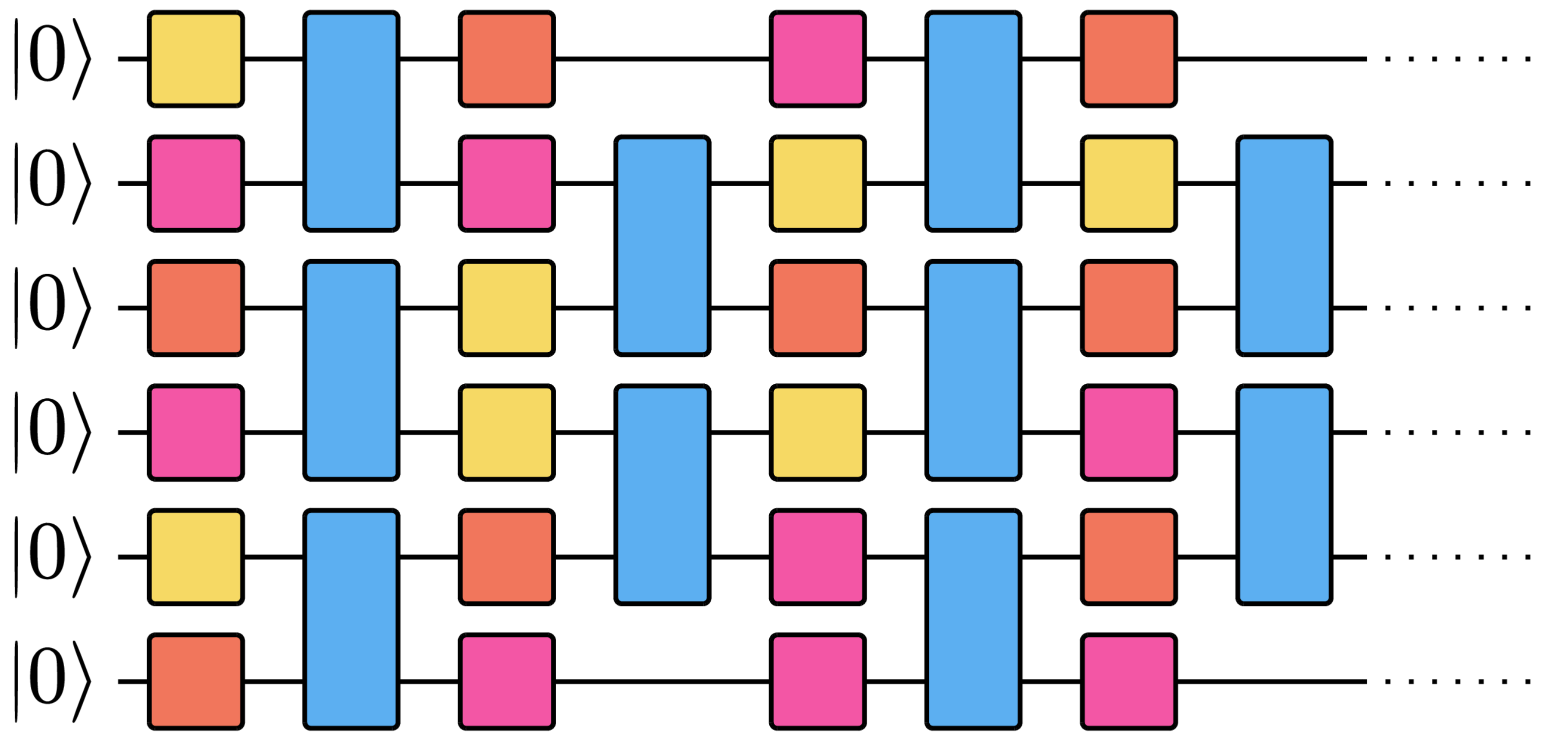


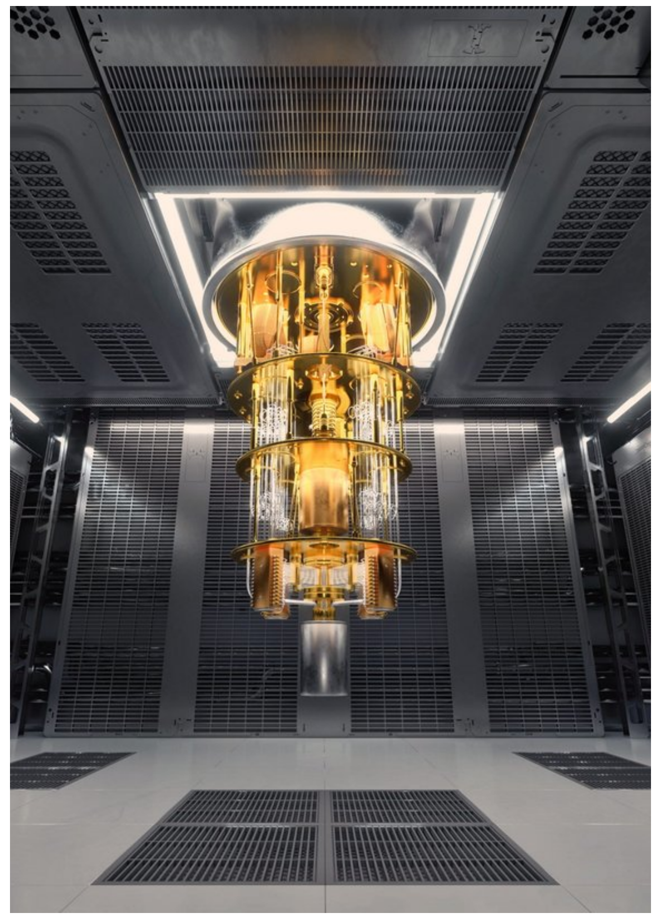
# TYPES OF QUANTUM SIMULATION

## ANALOGUE VS DIGITAL

- **Digital Quantum Simulation:** Uses a quantum computer to simulate the quantum system by breaking down the simulation into a sequence of quantum gates.
  - This is called digital because it relies on digital quantum computing which breaks down the computation into a discrete set of operations.
  - Example: Using a quantum computer and a method such as Trotter-Suzuki product formulas to simulate Heisenberg spin chains.



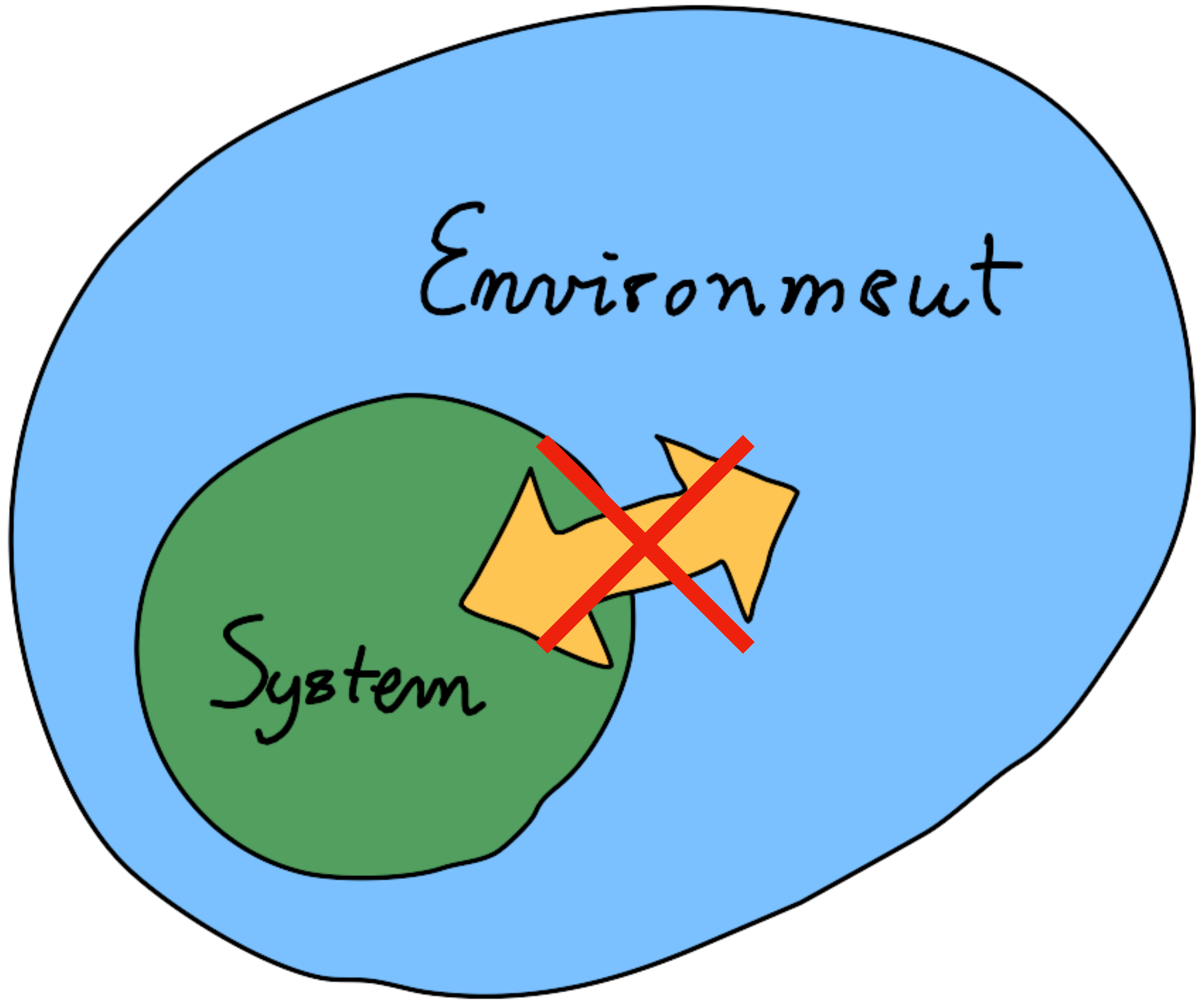




# WHAT TYPES OF QUANTUM SYSTEMS DO WE WANT TO SIMULATE?

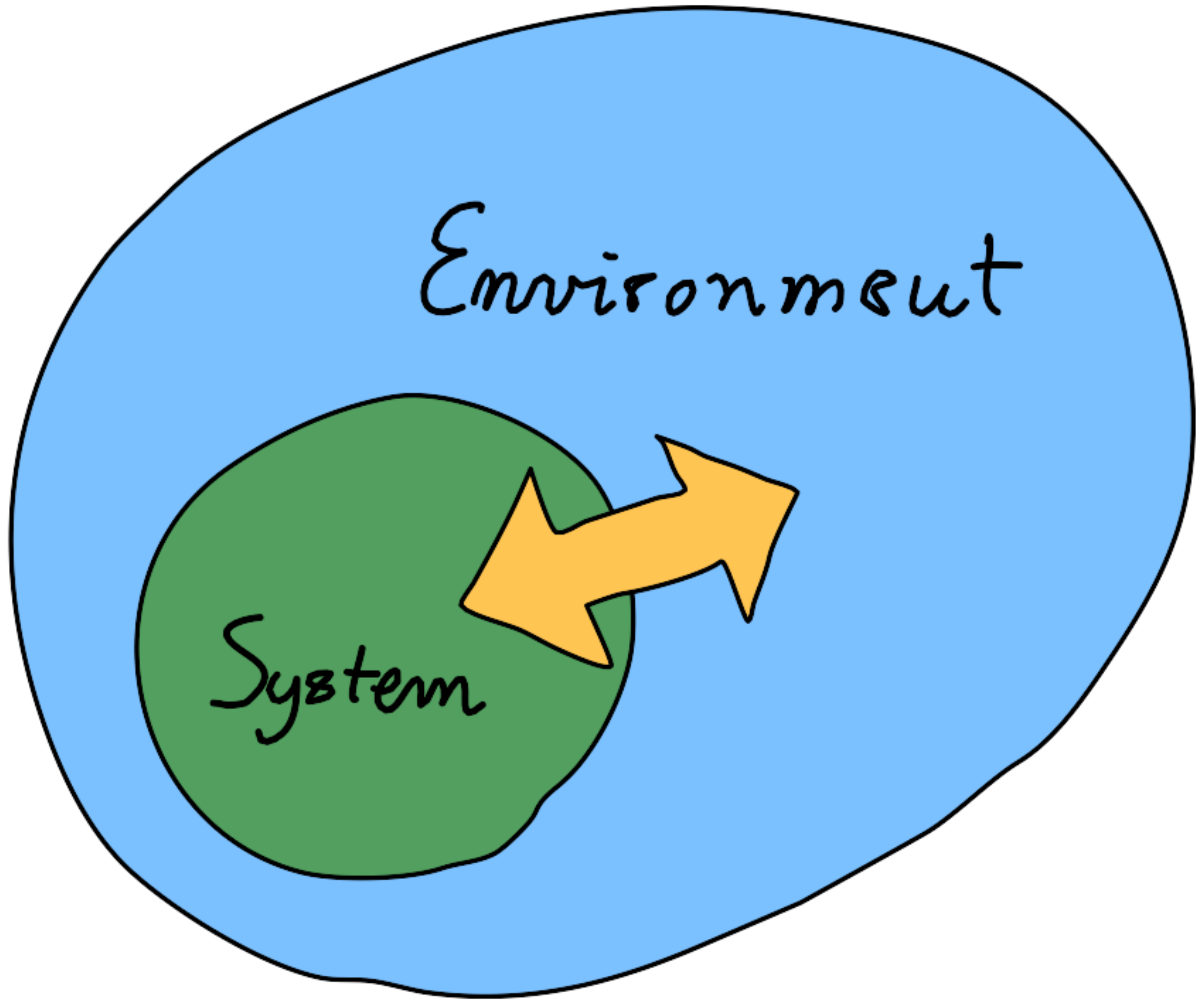
# WHAT TYPES OF QUANTUM SYSTEMS DO WE WANT TO SIMULATE?

- **Closed Quantum Systems:**
  - Isolated from their environment.
  - Evolve according to the Schrödinger equation with unitary time evolution.



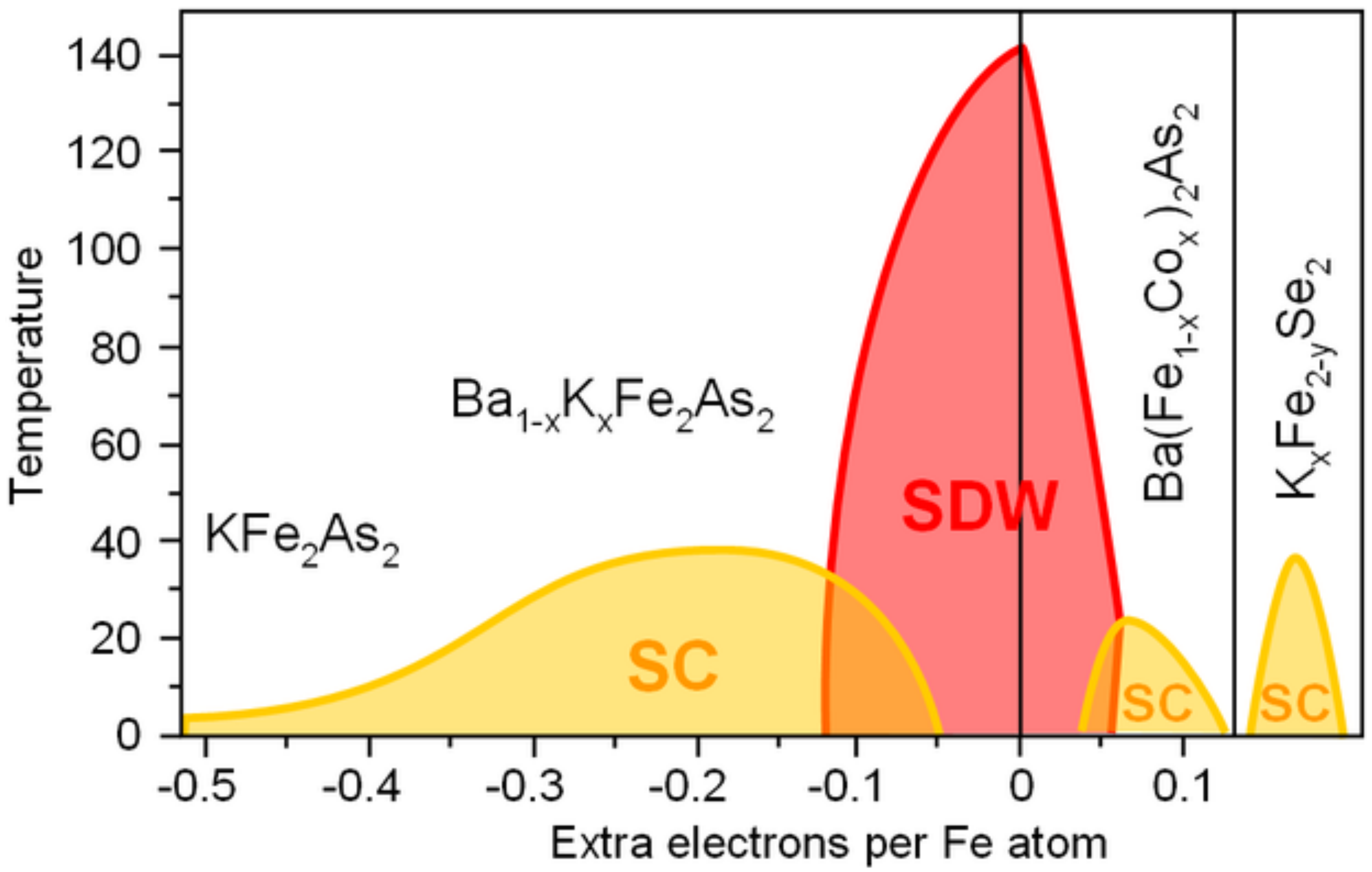
# WHAT TYPES OF QUANTUM SYSTEMS DO WE WANT TO SIMULATE?

- **Open Quantum Systems:**
  - The system is allowed to interact with an external environment so that it can exchange energy, matter and information.
  - Evolve according to the Lindblad master equation, incorporating dissipation and decoherence.

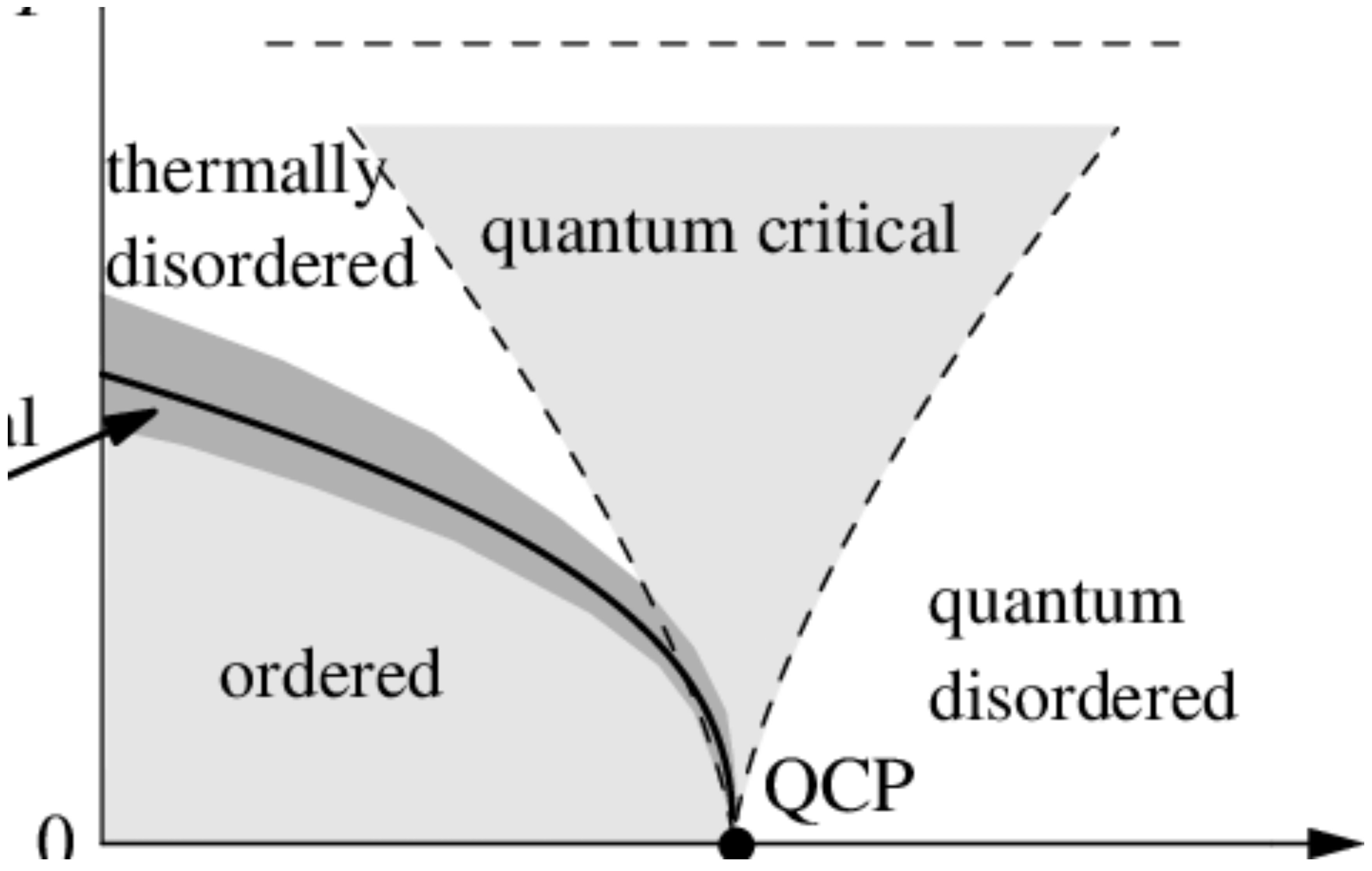


# APPLICATIONS OF QUANTUM SIMULATION

- **Fundamental Physics:**
  - Understanding high-temperature superconductivity.
  - Studying quantum phase transitions.

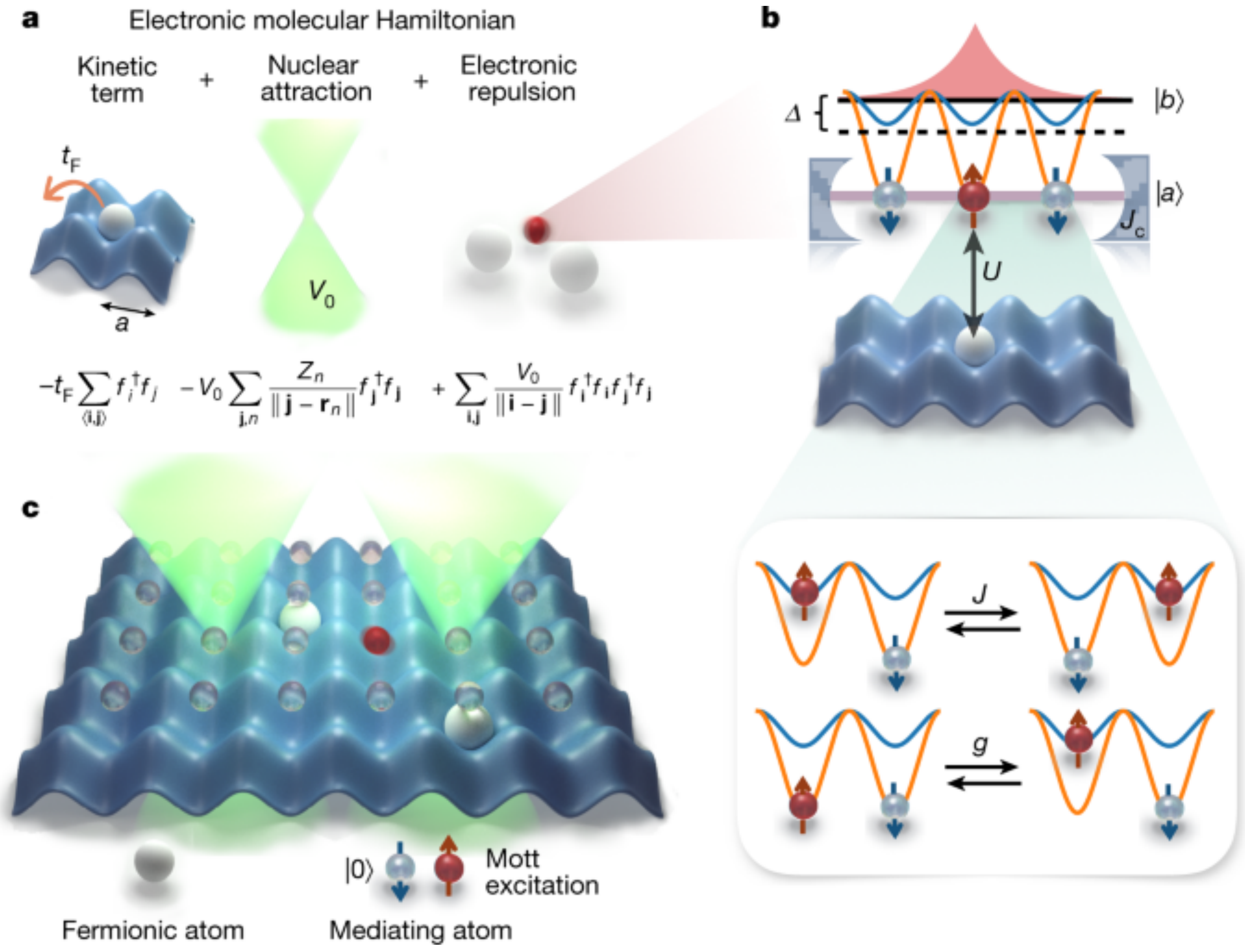


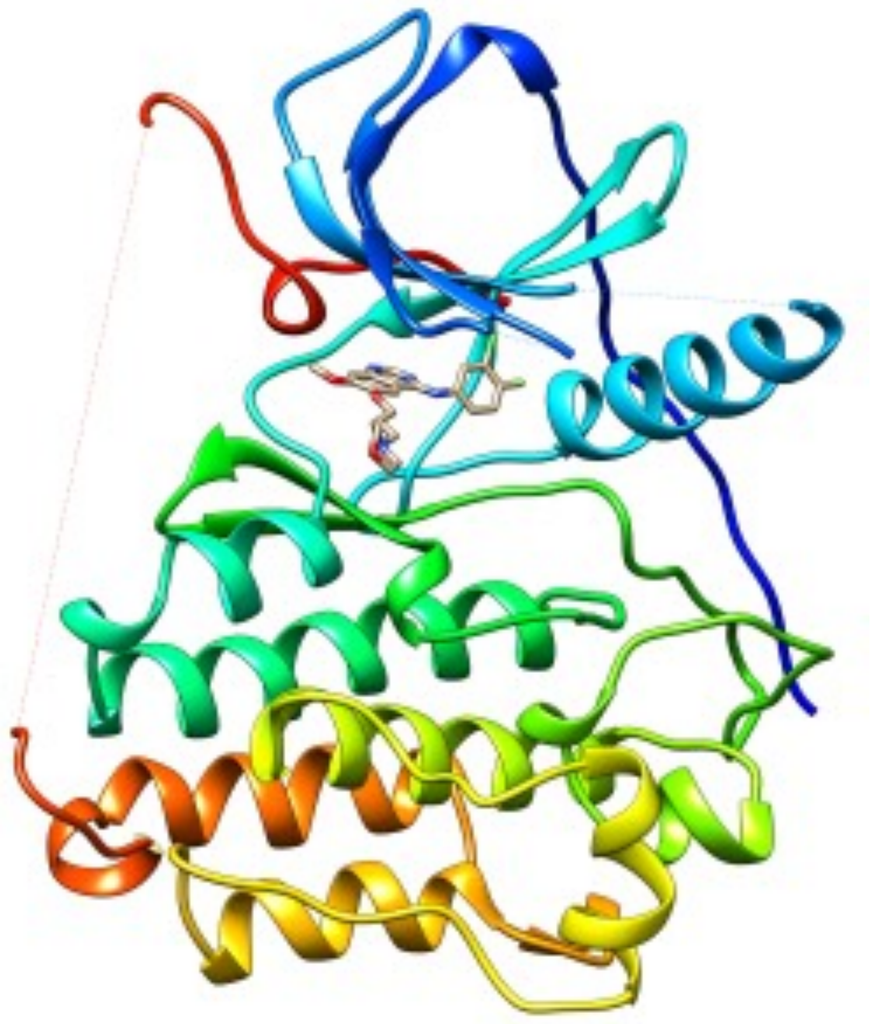
Phase diagram for high temperature super conductors based on iron

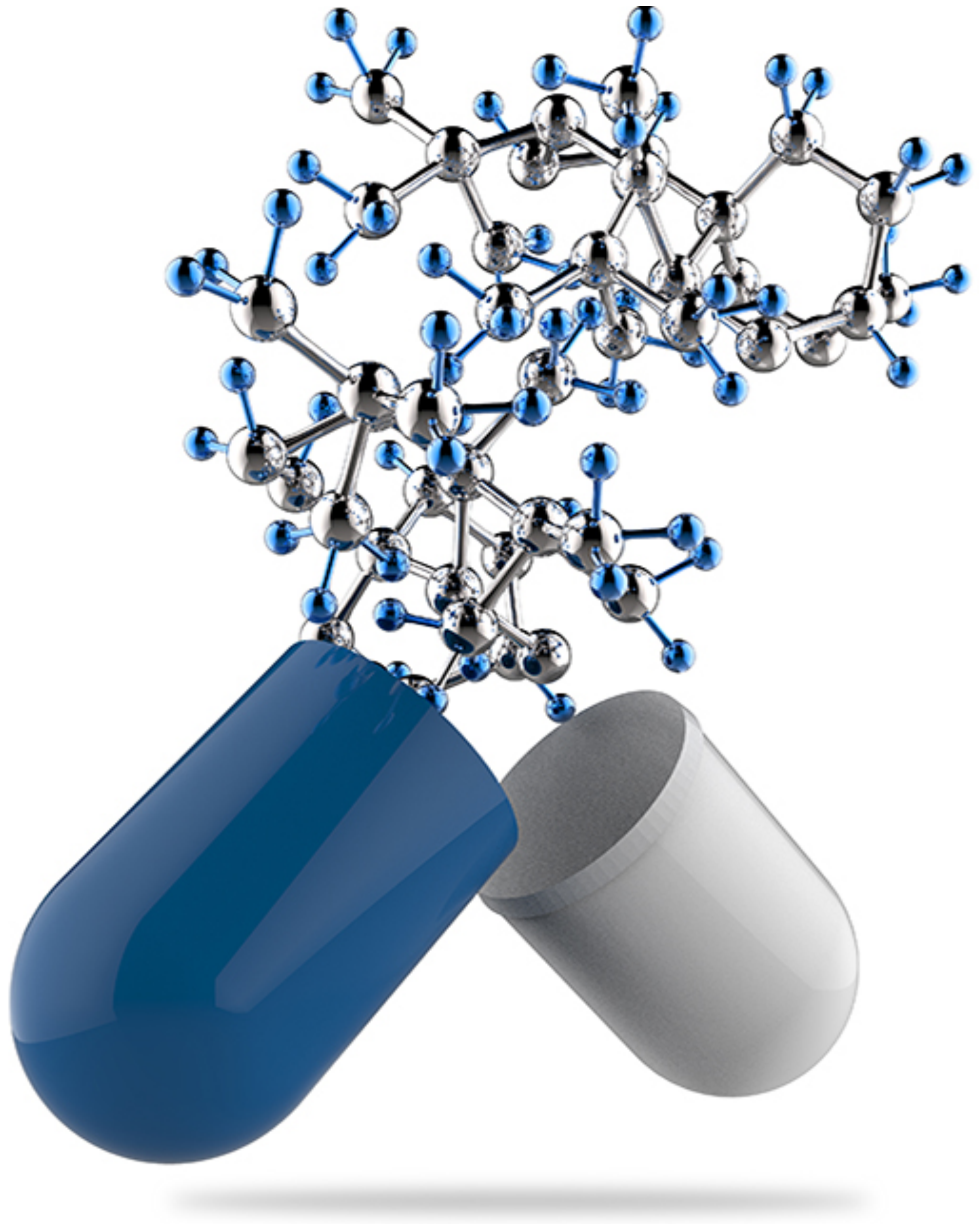


# APPLICATIONS OF QUANTUM SIMULATION

- **Quantum Chemistry:**
    - Simulating molecular structures and reactions.
    - Designing new drugs and materials.

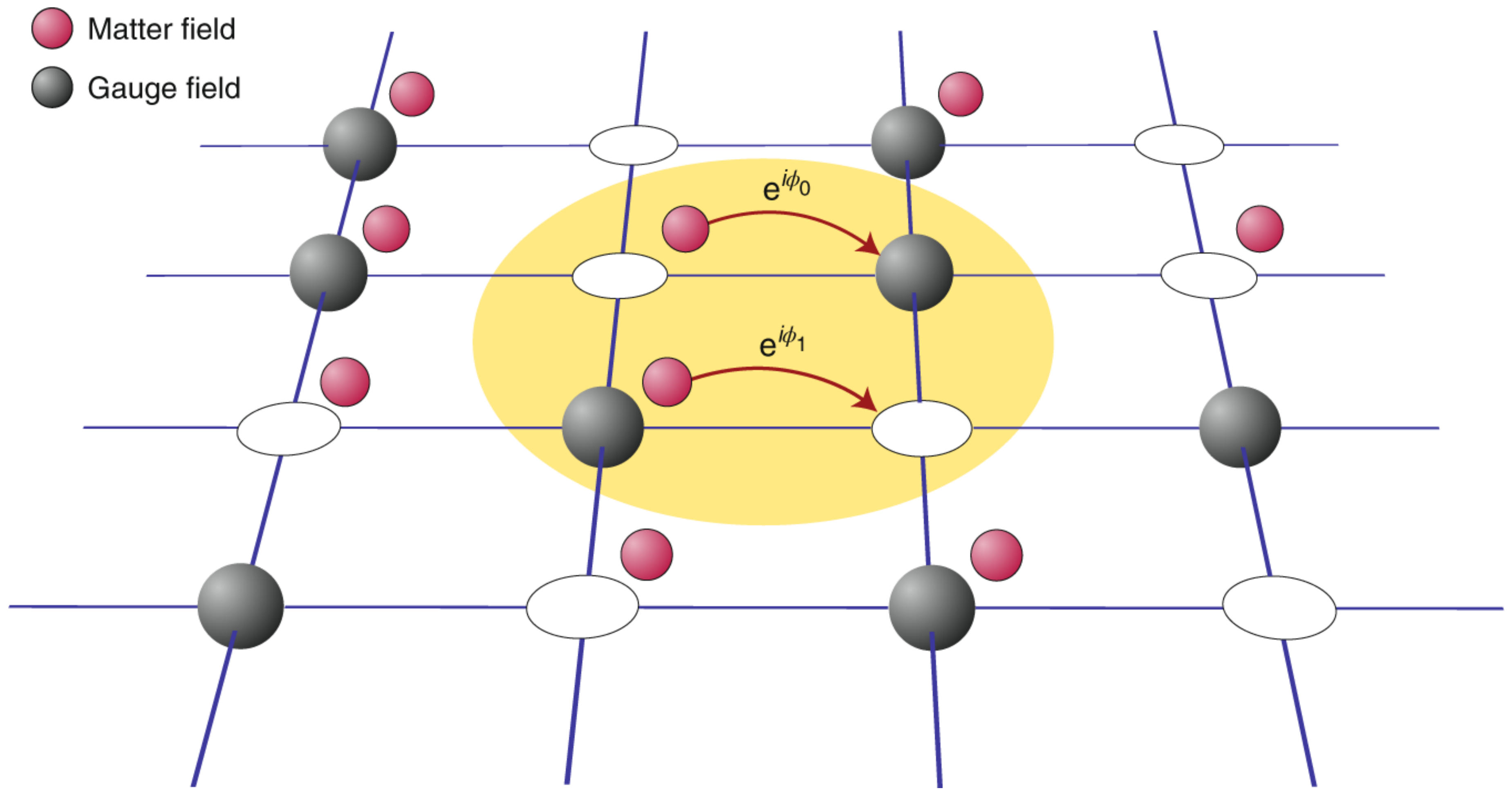

a
Electronic molecular Hamiltonian
Kinetic term
+
Nuclear attraction
+
Electronic repulsion
$t_F$
$a$
$V_0$
$-t_F \sum_{\langle i,j \rangle} f_i^\dagger f_j \quad -V_0 \sum_{\mathbf{j},n} \frac{Z_n}{\|\mathbf{j}-\mathbf{r}_n\|} f_\mathbf{j}^\dagger f_\mathbf{j} \quad + \sum_{\mathbf{i},\mathbf{j}} \frac{V_0}{\|\mathbf{i}-\mathbf{j}\|} f_\mathbf{i}^\dagger f_\mathbf{i} f_\mathbf{j}^\dagger f_\mathbf{j}$
b
$\Delta$
$|b\rangle$
$|a\rangle$
$J_c$
$U$
$J$
$g$
c
$|0\rangle$
Mott excitation
Fermionic atom
Mediating atom

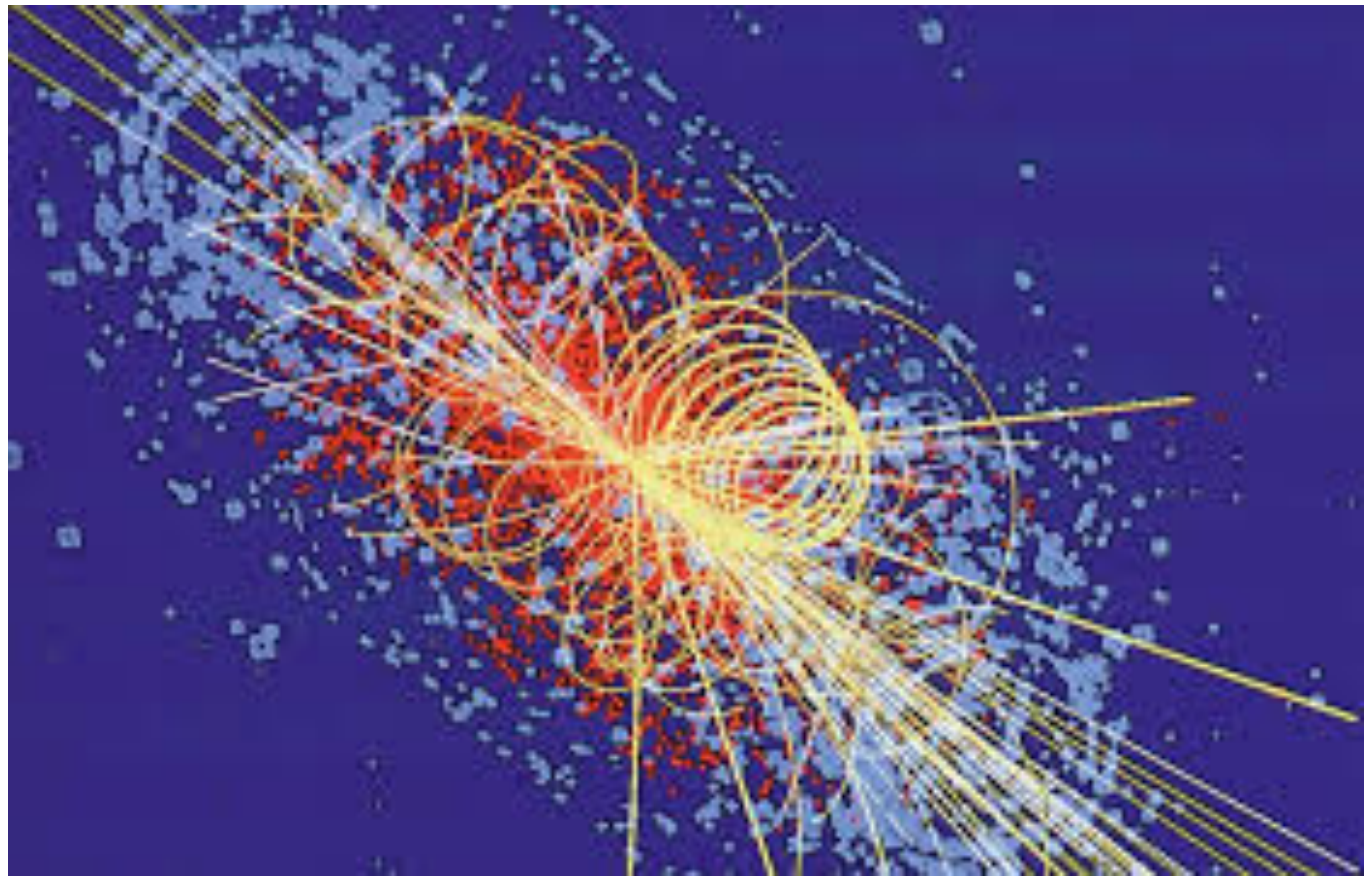




# APPLICATIONS OF QUANTUM SIMULATION

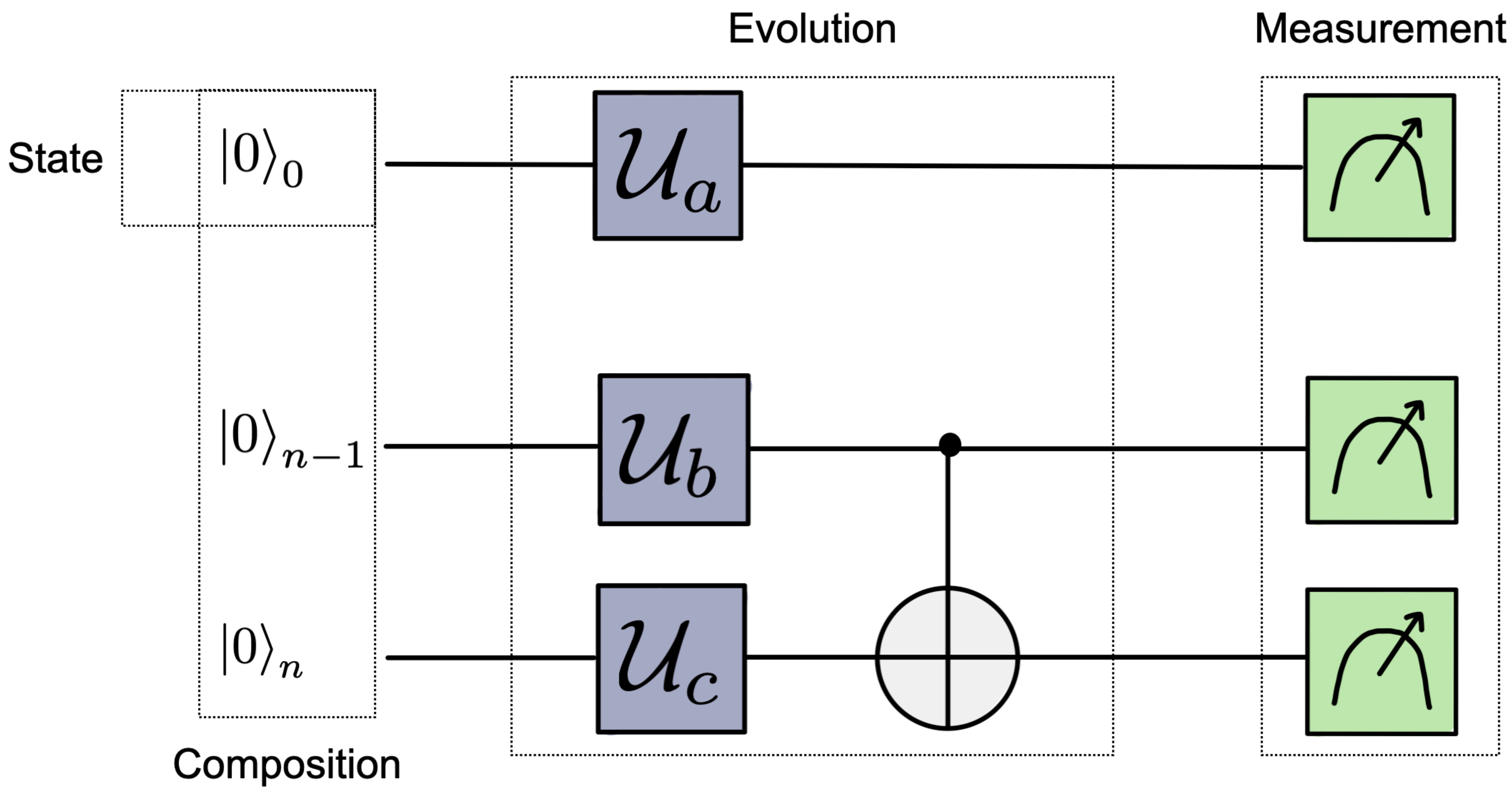
- **High Energy Physics:**
  - Simulating lattice gauge theories and particle interactions.





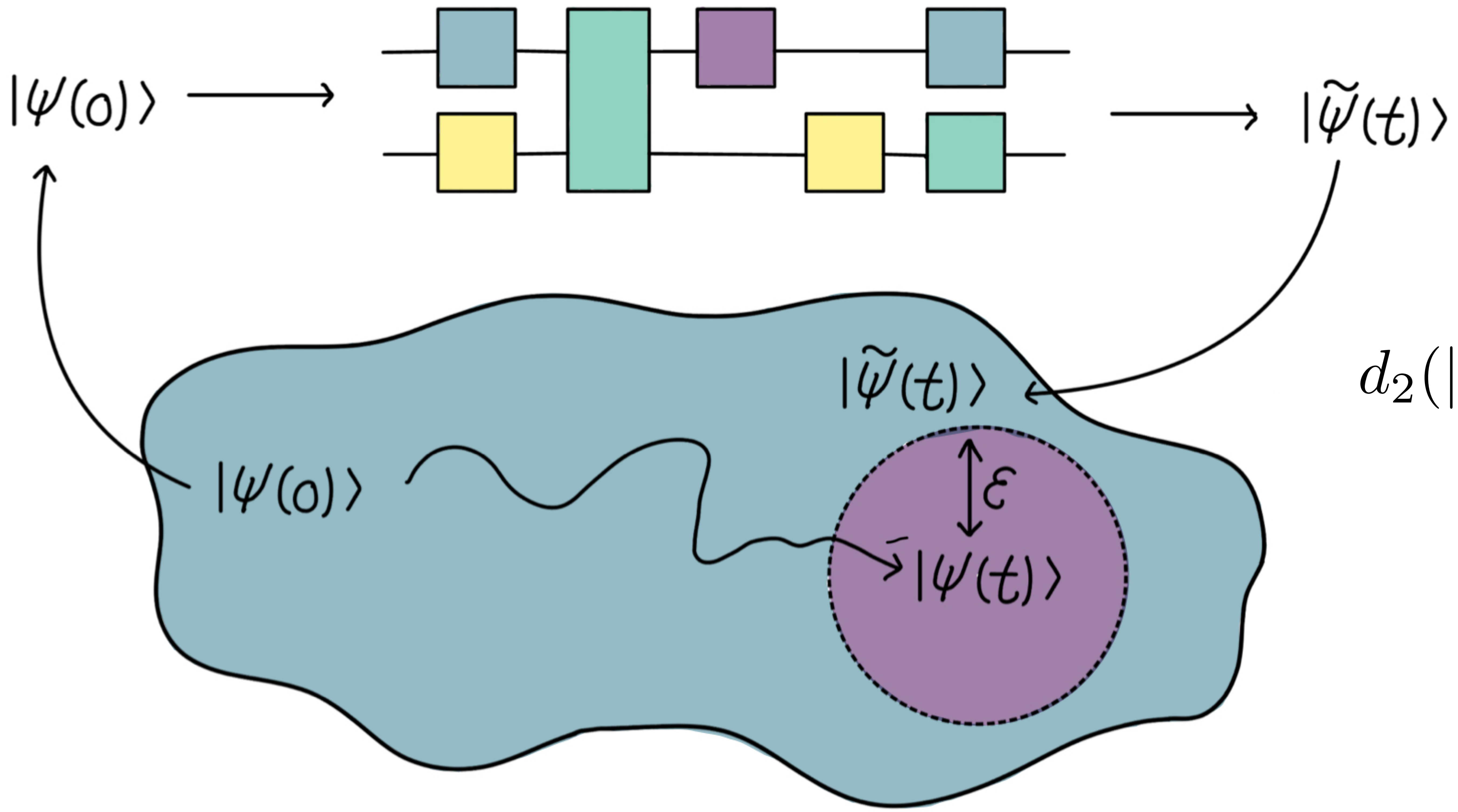
# BASIC QUANTUM COMPUTING

In quantum computing we make use of quantum systems called qubits and we perform computations by applying a sequence of quantum gates (unitary operations) to these qubits and then we perform measurements to obtain the output the computation.



# THE QUANTUM SIMULATION PROBLEM

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle \qquad |\tilde{\psi}(t)\rangle = \tilde{U}(t)\,|\psi(0)\rangle$$



$$\begin{aligned} d_2(|\psi(t)\rangle\,, |\tilde{\psi}(t)\rangle) &= \|\,|\psi(t)\rangle - |\tilde{\psi}(t)\rangle\|_2, \\ &\leq \|U(t) - \tilde{U}(t)\|_{op}\|\,|\psi(0)\rangle\,\|_2, \\ &\leq \|U(t) - \tilde{U}(t)\|_{op} \leq \epsilon \end{aligned}$$

# FORMAL: THE QUANTUM SIMULATION PROBLEM (CLOSED SYSTEM)

Given a Hamiltonian $H = \sum_j H_j$, a simulation time $t \geq 0$, a precision $\epsilon > 0$ and an initial state of the system $|\psi(0)\rangle$. Construct an approximation $\tilde{U}(t)$ to the time evolution operator $U(t) = e^{-iHt}$ such that,

$$\|U(t) - \tilde{U}(t)\| \leq \epsilon$$

and $\tilde{U}(t)$ can be efficiently implemented on a quantum computer using a polynomial number of quantum gates.

# WHY IS IT HARD TO SIMULATE QUANTUM DYNAMICS ON A CLASSICAL COMPUTER?

We can always represent $U(t)$ as a $d \times d$ matrix.

Consider a system of $n$ spins. The Hamiltonian describing a single spin is a $2 \times 2$ matrix.

So the Hamiltonian of $n$ spins is a $2^n \times 2^n$ matrix.

This means that the unitary operator describing the evolution will also be a $2^n \times 2^n$ matrix.

So the size of the matrix that a classical computer needs to store will grow exponentially with the number of spins.

Usually we want to construct $\tilde{U}(t)$ as a finite product of other simpler unitary operators $V_j(t)$, which can be easily implemented on a quantum computer i.e.

$$\tilde{U}(t) = \prod_j V_j(t)$$

We want to find $\tilde{U}(t)$ such that estimating bounds for the precision and analysis of the complexity are possible without specific knowledge of $H$.

# FIRST ORDER TROTTER-SUZUKI PRODUCT FORMULAS

Let us try to do this for a simple Hamiltonian we want to simulate,

$$H = H_1 + H_2$$

This means we need to construct an approximation to,

$$U(t) = e^{-i(H_1+H_2)t}$$

as a product of other unitaries. The natural thing to do is to try to write $\tilde{U}(t)$ in terms of $e^{-iH_1t}$ and $e^{-iH_2t}$, to do this we recall the Trotter Formula

$$e^{-i(H_1+H_2)t} = \lim_{n\to\infty}(e^{-iH_1t/n}e^{-iH_2t/n})^n.$$

The Trotter formula is an exact formula if we want to approximate $U(t)$ then we know that for large enough $n$ we have,

$$e^{-i(H_1+H_2)t} \approx (e^{-iH_1t/n}e^{-iH_2t/n})^n.$$

However to use this approximation we need a rigorous bound for the error as well as a bound on $n$ as saying large enough isn't precise enough.

Let us start by defining $\tau := t/n$ then we can write,

$$U(t) = (e^{-iHt/n})^n = (e^{-iH\tau})^n = U(\tau)^n$$

and,

$$\tilde{U}(\tau) = e^{-iH_1\tau}e^{-iH_2\tau}$$

Then we need to show that,

$$\|U(t) - U(\tau)^n\| \leq \epsilon$$

and we need to find lower bounds on the precision $\epsilon > 0$ and on $n \in \mathbb{N}$.

$$\Lambda := \max\{\|H_1\|, \|H_2\|\}$$

This allows us to arrive at the bound,

$$\left\|U(\tau) - \tilde{U}(\tau)\right\| \leq 2\frac{|-i\tau|^2(2\Lambda)^2}{2!}e^{2|-i\tau|\Lambda} = (2\tau\Lambda)^2 e^{2\tau\Lambda}$$

Since $\tau = t/n$ we have,

$$\left\|U(t) - \tilde{U}(\tau)^n\right\| \leq n\frac{(2t\Lambda)^2}{n^2}e^{2t\Lambda/n} = \frac{(2t\Lambda)^2}{n}e^{2t\Lambda/n}$$

If we choose $\epsilon \geq \frac{(2t\Lambda)^2}{n}e^{2t\Lambda/n}$ then $\left\|U(t) - \tilde{U}(\tau)^n\right\| \leq \epsilon$.

In a quantum simulation problem we usually want to specify a precision $\epsilon$ and then find the number of times $n$ we need to apply $\tilde{U}(\tau)$.

To find $n$ one either directly solves numerically $\epsilon = \frac{(2t\Lambda)^2}{n} e^{2t\Lambda/n}$, or if we choose some $n$ such that $e^{2t\Lambda/n} \approx 1$ then we can easily have,

$$n = \left\lceil \frac{(2t\Lambda)^2}{\epsilon} \right\rceil .$$

# HIGHER ORDER TROTTER-SUZUKI PRODUCT FORMULAS

So far we have looked at an approximation up to only first order but what about higher order approximations. This problem was addressed by Suzuki

To keep track of the order to which we approximate $U$ we will write down our product approximations as $S_k$ where $k$ will denote the order to of the approximation, for example

$$\tilde{U}(\tau) = S_1(\tau) = \prod_{j=1}^{M} e^{-iH_j\tau}.$$

Suzuki M. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. Physics Letters A. 1990 Jun 4;146(6):319-23.

As a starting point consider the second order approximation, which for a Hamiltonian $H = \sum_{j=1}^{M} H_j$ and a small time step $\tau \geq 0$ is,

$$S_2(\tau) = \prod_{j=1}^{M} e^{-iH_j\tau/2} \prod_{j'=M}^{1} e^{-iH_{j'}\tau/2}$$

The error bound for this formula is, $\|U(t) - S_2(\tau)^n\| \leq \frac{(2L\tau\Lambda)^3}{3n^2} e^{2L\tau\Lambda/n}$, which shows that it is a second order formula and a value for $n$ is found in a similar manner to the first order formula.

We also note that this approximation is symmetric meaning that under time reversal the structure is preserved.

Suzuki started with the second order product formula $S_2$ and recursively constructed higher order product formulas.

He did this by taking products of the second order product formulas in such a way that when you Taylor expand it you are able to cancel higher order terms in the expansion of $U(t)$.

As an example, Suzuki found that if you consider the Taylor expansion of the product,

$$S_{2k-2}(p_2\tau)^2 S_{2k-2}([1-4p_2]\tau) S_{2k-2}(p_k\tau)^2$$

where $p_k = 1/(4-4^{(1/2k-1)})$, then this product formula can approximate $U(\tau)$ to an order $2k$ where $k \in \mathbb{N}$. We will usually write this as $S_{2k}(\tau)$ i.e.

$$S_{2k}(\tau) = S_{2k-2}(p_2\tau)^2 S_{2k-2}([1-4p_2]\tau) S_{2k-2}(p_k\tau)^2.$$

Any formula of this form is referred to as a Trotter-Suzuki product formula.

The analysis for the error bounds with this product formula are quite complicated so we just state the bounds,

$$\|U(t) - S_{2k}(\tau)^n\| \leq \frac{(2L5^{k-1}\Lambda t)^{2k+1}}{3n^{2k}} \exp\left(\frac{2L5^{k-1}\Lambda t}{n}\right) \leq \epsilon$$

and we find $n$ to be,

$$n = \left\lceil \max\left\{2L5^{k-1}\Lambda t,\ \sqrt[2k]{\frac{e(2L5^{k-1}\Lambda t)^{2k+1}}{3\epsilon}}\right\}\right\rceil .$$

An important quantity that will aid us later in computing the number of gates in our circuit is the number of exponentials in the product formula $N_{exp}$ which for this formula is,

$$N_{exp} \leq 2L5^{k-1}n.$$

# TROTTER-SUZUKI QUANTUM SIMULATION ALGORITHM

**Step 1:** Given the Hamiltonian $H$ for our system write it in the form $H = \sum_j H_j$ such that each $H_j$ can be exponentiated easily and has an efficient quantum circuit implementation.

**Step 2:** Pick an order $k$ for the Trotter-Suzuki product formula $S_{2k}$ you wish to use to approximate $U(t)$.

**Step 3:** Compute $\Lambda = \max_j \|H_j\|$, specify a simulation time $t \geq 0$, a precision $\epsilon > 0$ and some initial state $|\psi(0)\rangle$. Use this to compute $n$.

**Step 4:** Construct quantum circuits for each $e^{-iH_j\tau}$.

**Step 5:** Construct a gate set using $S_{2k}(t/n)^n$ that will be implemented on a quantum computer using $|\psi(0)\rangle$ as an initial state.

The state of an open quantum system is described by a density matrix $\rho(t)$ whose evolution is described by a quantum channel $T_t = e^{t\mathscr{L}}$ i.e.

$$\rho(t) = T_t\rho(0)$$

where $\mathscr{L}$ is the Markovian GKSL generator

$$\mathscr{L}\rho = -i[H,\rho] + \sum_k \gamma_k(L_k\rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\})$$

For convenience we usually write the generator as,

$$\mathscr{L} = \sum_{k=1}^{M} \mathscr{L}_k$$

# PROBLEM AND SETTING

Given a quantum channel $T_t = \exp(t\mathscr{L})$ where

$$\mathscr{L}(\rho) = -i[H,\rho] + \sum_k \gamma_k \left( L_k \rho L_k^\dagger - \frac{1}{2}\left\{ L_k^\dagger L_k, \rho \right\} \right)$$

is the generator. We want to construct an approximation to the channel $T_t$ within some precision $\epsilon$, and be able to implement that approximation using a quantum circuit that uses fewer gates than standard Trotter-Suzuki product formulas.

# PROBLEM AND SETTING

To simplify our calculations we can re-write the generator as,

$$\mathscr{L}(\rho) = \sum_{k=1}^{M} \gamma_k \mathscr{L}_k(\rho)$$

where $\mathscr{L}_1(\rho) = -i[H, \rho]$, $\mathscr{L}_k(\rho) = L_k \rho L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho\right\}$ and $\gamma_1 = 1$.

Sometimes it will also be convenient to absorb the decay rates $\gamma_k$ into the generators $\mathscr{L}_k$ so we have,

$$\mathscr{L}(\rho) = \sum_{k=1}^{M} \hat{\mathscr{L}}_k(\rho)\,.$$

# DIAMOND NORM

We will make use of the diamond norm as a measure of the precision of our approximation. The diamond norm of a superoperator $V : \mathcal{B}(\mathcal{H}_s) \rightarrow \mathcal{B}(\mathcal{H}_s)$ is,

$$V = \sup_{A;\|A\|_1=1} \|(V \otimes \mathbb{1})(A)\|_1,$$

where $\mathbb{1} : \mathcal{B}(\mathcal{H}_s) \rightarrow \mathcal{B}(\mathcal{H}_s)$ is the identity superoperator, $A \in \mathcal{B}(\mathcal{H}_s)$ is an operator and $\| \cdot \|_1$ is the trace norm and it is defined as,

$$\|A\|_1 = \mathrm{tr}(\sqrt{AA^\dagger}),$$

for some operator $A$.

# IMPORTANT BOUND FOR DIAMOND NORM

Given two quantum channels $T$ and $V$ and some positive integer $N$,

$$\left\|T^N - V^N\right\|_\diamond \leq N \left\|T - V\right\|_\diamond .$$

# BOUND ON THE GENERATOR

Using the diamond norm we can immediately find an upper-bound on the generator $\mathcal{L}$,

$$\|\mathcal{L}\|_\diamond = \left\|\sum_{k=1}^{M} \hat{\mathcal{L}}_k\right\|_\diamond \leq \sum_{k=1}^{M} \left\|\hat{\mathcal{L}}_k\right\|_\diamond .$$

If we define $\Lambda := \max_k \left\{\left\|\hat{\mathcal{L}}_k\right\|_\diamond\right\}$ then,

$$\|\mathcal{L}\|_\diamond \leq \sum_{k=1}^{M} \left\|\hat{\mathcal{L}}_k\right\|_\diamond \leq \sum_{k=1}^{M} \Lambda = M\Lambda.$$

The diamond norm can be related to the trace norm via the following inequality. Given two superoperators $V$ and an operator $A$ we have by definition,

$$\|A\|_1 \leq \|V\|_\diamond .$$

# DETERMINISTIC TROTTER-SUZUKI PRODUCT FORMULAS

Using the Trotter-Suzuki (TS) product formulas we want to approximate the total evolution $T_t = \exp\left(t \sum_{k=1}^{M} \hat{\mathcal{L}}_k\right)$ as some product of simpler channels such that the total evolution is approximated up to a precision $\epsilon \geq 0$, when using the diamond norm as the distance measure for the quantum channels.

we can approximate the evolution $T_\tau$ up to first order by using the following deterministic TS product formula,

$$S_1^{(det)}(\tau) = \prod_{k=1}^{M} e^{\tau \hat{\mathcal{L}}_k},$$

where we shall refer to the exponentials of the form $\exp\left(\tau \hat{\mathcal{L}}_k\right) = \exp(\tau \gamma_k \mathcal{L}_k)$ as constituent channels. Similarly we can approximate $T_\tau$ up to second order using the formula,

$$S_2^{(det)}(\tau) = \prod_{k=1}^{M} e^{\frac{\tau}{2}\hat{\mathcal{L}}_k} \prod_{k'=M}^{1} e^{\frac{\tau}{2}\hat{\mathcal{L}}_{k'}}.$$

We refer to these product formulas as deterministic because the ordering of the exponentials in $S_1^{(det)}$ and $S_2^{(det)}$ is known before hand.

**Theorem:** Given the generator $\mathcal{L}$, of a quantum channel $T_t$ and some time $t \geq 0$. Define the first order deterministic TS product formula as,

$$S_1^{(det)}(\tau) = \prod_{k=1}^{M} e^{\tau \hat{\mathcal{L}}_k}.$$

Let $N \in \mathbb{N}$ and $\Lambda := \max_k \left\| \hat{\mathcal{L}}_k \right\|_\diamond$. Then,

$$\left\| T_t - S_1^{(det)} \left( \frac{t}{N} \right)^N \right\|_\diamond \leq \epsilon,$$

where,

$$\epsilon \geq \frac{t^2 \Lambda^2 M^2}{N}, \qquad N \geq \frac{t^2 \Lambda^2 M^2}{\epsilon}.$$

**Theorem:** Given the generator $\mathcal{L}$, of a quantum channel $T_t$ and some time $t \geq 0$. Define the second order deterministic TS product formula as,

$$S_2^{(det)}(\tau) = \prod_{k=1}^{M} e^{\frac{\tau}{2}\hat{\mathcal{L}}_k} \prod_{k'=M}^{1} e^{\frac{\tau}{2}\hat{\mathcal{L}}_{k'}}.$$

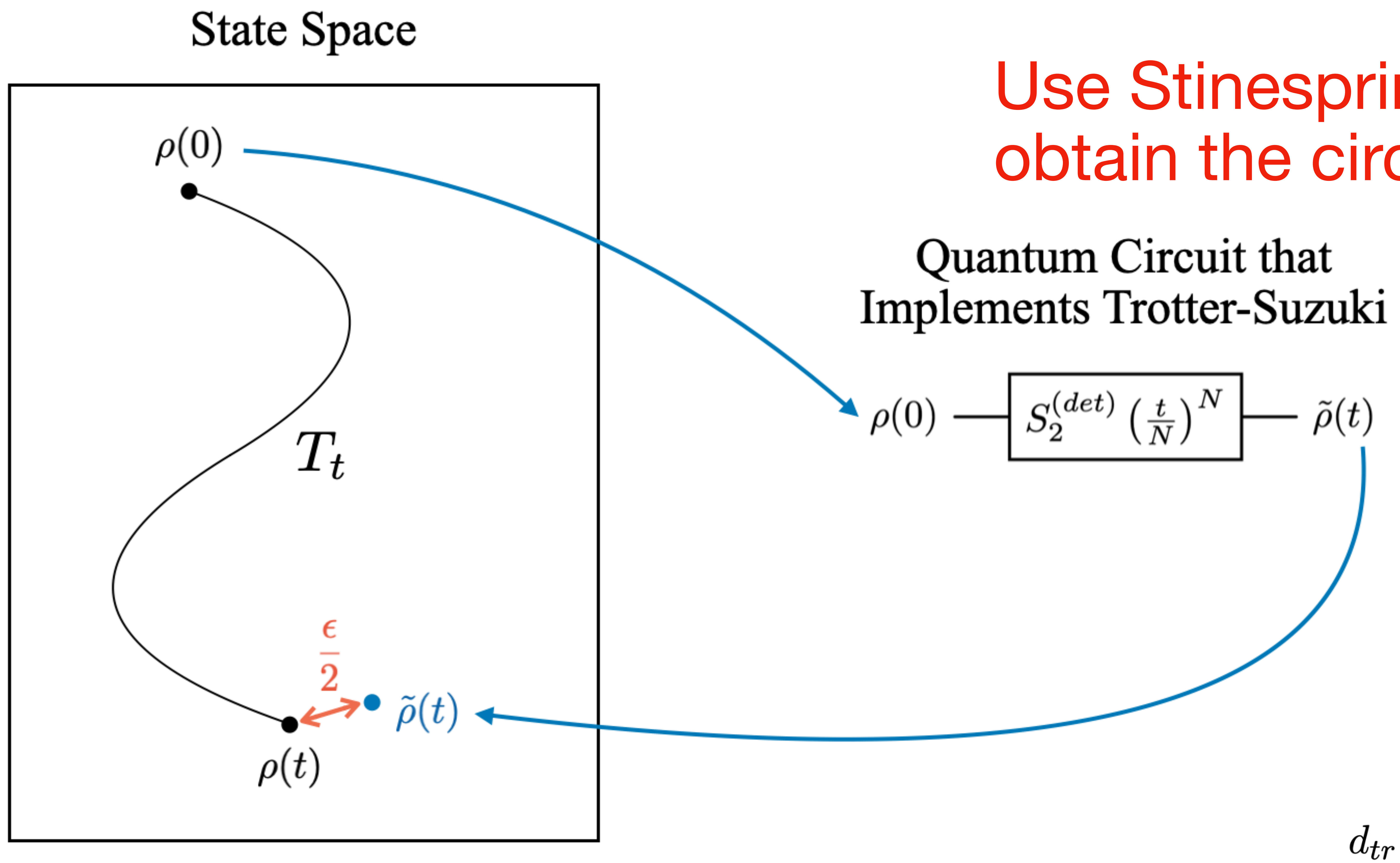
Let $N \in \mathbb{N}$ and $\Lambda := \max_k \left\|\hat{\mathcal{L}}_k\right\|_\diamond$. Then,

$$\left\|T_t - S_2^{(det)}\left(\frac{t}{N}\right)^N\right\|_\diamond \leq \epsilon,$$

where,

$$\epsilon \geq \frac{M^3 t^3 \Lambda^3}{3N^2}, \qquad N \geq \frac{M^{3/2} t^{3/2} \Lambda^{3/2}}{\sqrt{3\epsilon}}.$$

Use Stinespring dilation theorem to obtain the circuits for the channel



State Space

Quantum Circuit that Implements Trotter-Suzuki

$$
\begin{aligned}
d_{tr}(\rho(t), \tilde{\rho}(t)) &= \frac{1}{2}\|\rho(t) - \tilde{\rho}(t)\|_1, \\
&= \frac{1}{2}\left\|T_t(\rho(0)) - S_2^{(det)}(t/N)^N \rho(0)\right\|_1, \\
&\leq \frac{1}{2}\left\|T_t - S_2^{(det)}(t/N)^N\right\|_\diamond, \\
&\leq \frac{\epsilon}{2},
\end{aligned}
$$

where $\epsilon \geq (Mt\Lambda)^3/3N^2$

# GATE COMPLEXITIES FOR DETERMINISTIC TROTTER-SUZUKI PRODUCT FORMULAS

consider the product formula $S_1^{(det)}(\tau)$, this formula is the product of $M$ exponentials. If we consider the quantum circuit that implements $S_1^{(det)}(\tau)^N$ to approximate $\rho(t)$ to a precision $\epsilon$, then we have $N = \left\lceil t^2\Lambda^2 M^2/\epsilon \right\rceil$ applications of $S_1^{(det)}(\tau)$ which implies that we have to implement $\left\lceil t^2\Lambda^2 M^2/\epsilon \right\rceil M$ exponentials. Where $\lceil \cdot \rceil$ denotes the ceiling function and it is the smallest integer greater than its argument. If we denote the gate complexity for the circuit that implements $S_1^{(det)}$ as $g_1^{(det)}$ then the complexity is given by,

$$g_1^{(det)} = O\left(\frac{t^2\Lambda^2 M^3}{\epsilon}\right).$$

The second order formula $S_2^{(det)}(\tau)$, contains $2M$ exponentials. The quantum circuit that implements $S_2^{(det)}(\tau)^N$ to approximate $\rho(t)$ to a precision $\epsilon$ will contain $N = \lceil M^{3/2}t^{3/2}\Lambda^{3/2}/\sqrt{3\epsilon}\,\rceil$ applications of $S_2^{(det)}(\tau)$. This tells us that we will have to implement $2\lceil M^{3/2}t^{3/2}\Lambda^{3/2}/\sqrt{3\epsilon}\,\rceil M$ exponentials. By denoting the gate complexity of this formula by $g_2^{(det)}$ we see that,

$$g_2^{(det)} = O\left(\frac{M^{5/2}t^{3/2}\Lambda^{3/2}}{\sqrt{3\epsilon}}\right).$$

# HOW CAN WE IMPROVE THE GATE COMPLEXITIES DEPENDENCE ON THE NUMBER OF TERMS IN THE GENERATOR?

# ANSWER: USE RANDOMISATION

Based on: David I J, Sinayskiy I and Petruccione F. Faster Quantum Simulation of Markovian Open Quantum Systems via Randomisation. arXiv:2408.11683 (2024) (currently in review at Quantum Journal)

The key idea is to randomly apply exponentials in the TS product formulas of first and second order to achieve an improvement in the gate complexities dependence on the number of terms in the generator.

# QDRIFT ALGORITHM FOR OQS

We also construct an approximation inspired by the stochastic drift algorithm (qDRIFT) in [2]. We have the following theorem for the qDRIFT algorithm for open quantum systems.

**Theorem 3 (qDRIFT)**: Given a quantum channel $T_t = \exp(t\mathscr{L})$ with a generator $\mathscr{L} = \sum_{k=1}^{M} \gamma_k \mathscr{L}_k$ and for $\epsilon \geq 0$, there exists some $N \in \mathbb{N}$ such that,

$$||T_t - \mathscr{E}_\tau^{\circ N}||_\diamond \leq \epsilon,$$

with, $\epsilon \geq \dfrac{t^2\alpha^2\Lambda^2}{N}$ $\quad N \geq \dfrac{t^2\alpha^2\Lambda^2}{\epsilon}$ , and, $\mathscr{E}_\tau = \sum_{k=1}^{M} p_k \exp(\tau\mathscr{L}_k)$ , with $p_k = \gamma_k/\alpha$ , $\alpha = \sum_{k=1}^{M} \gamma_k$ such that $\sum_{k=1}^{m} p_k = 1$ and $\tau = t\alpha/N$.

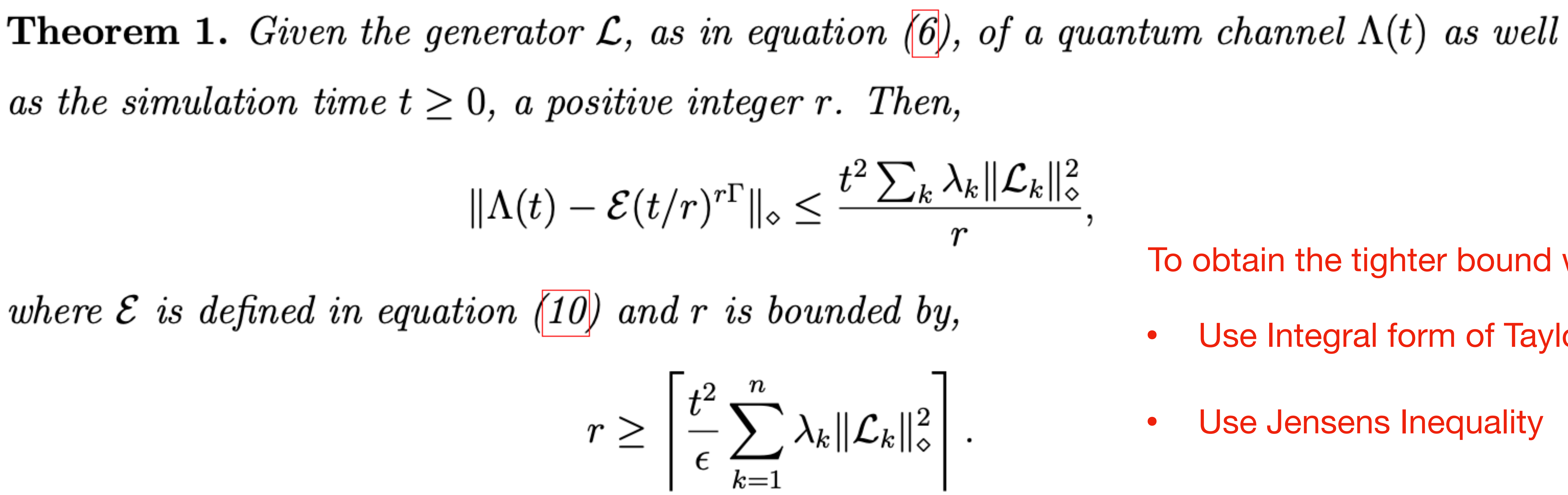
[2] Campbell E. Random compiler for fast Hamiltonian simulation. Physical review letters. 2019 Aug 14;123(7):070503.

| **Method Of Simulation** | **Gate Complexity** |
|---|---|
| First Order TS Deterministic | $O\left((t\Lambda)^2 M^3/\epsilon\right)$ |
| Second Order TS Deterministic | $O\left((t\Lambda)^{3/2} M^{5/2}/\sqrt{3\epsilon}\right)$ |
| First Order TS Randomised (CS) | $O\left((t\Lambda)^{3/2} M^{5/2}/\sqrt{3\epsilon}\right)$ |
| Second Order TS Randomised (CS) | $O\left((t\Lambda)^{3/2} M^{2}/\sqrt{\epsilon}\right)$ |
| QDRIFT (CS) | $O\left((t\Gamma\Omega)^2/\epsilon\right)$ |
| First Order TS Randomised (QF) | $O\left((t\Lambda)^{3/2} M^{5/2}/\sqrt{3\epsilon}\right)$ |
| QDRIFT (QF) | $O\left((t\Gamma\Omega)^2 M/\epsilon\right)$ |

# HOW CAN WE FURTHER IMPROVE THE QDRIFT METHOD FOR BOTH CLOSED AND OPEN QUANTUM SYSTEMS?

SOLUTION:David I. J. , Sinayskiy I. and Petruccione F. , Tighter Error Bounds for the qDRIFT algorithm, arXiv:2506.17199 (2025)

# MAIN RESULTS



**Theorem 1.** *Given the generator $\mathcal{L}$, as in equation (6), of a quantum channel $\Lambda(t)$ as well as the simulation time $t \geq 0$, a positive integer $r$. Then,*

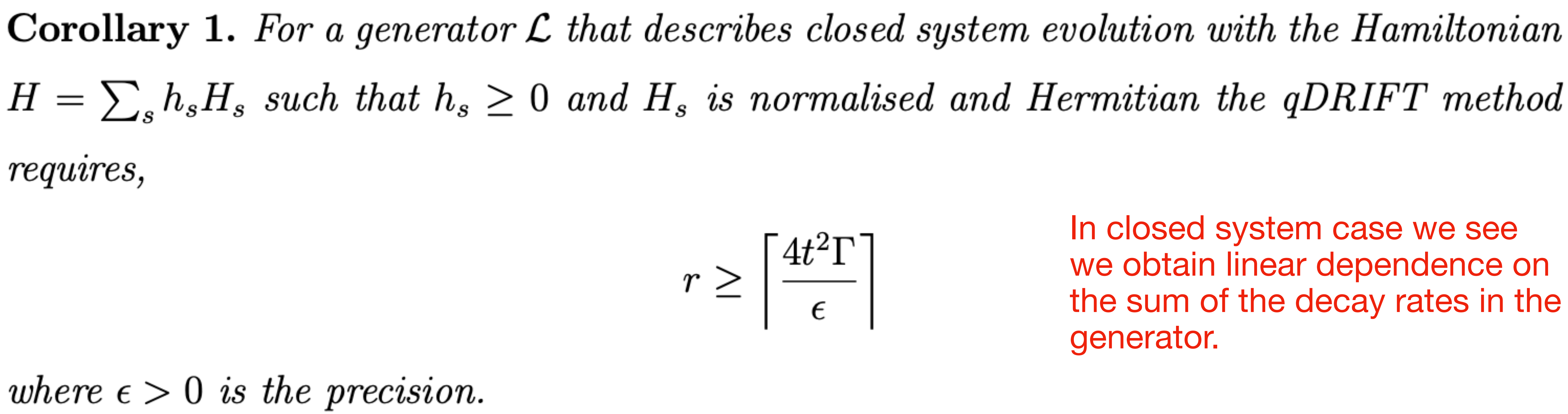
$$\|\Lambda(t) - \mathcal{E}(t/r)^{r\Gamma}\|_\diamond \leq \frac{t^2 \sum_k \lambda_k \|\mathcal{L}_k\|_\diamond^2}{r},$$

*where $\mathcal{E}$ is defined in equation (10) and $r$ is bounded by,*

$$r \geq \left\lceil \frac{t^2}{\epsilon} \sum_{k=1}^{n} \lambda_k \|\mathcal{L}_k\|_\diamond^2 \right\rceil .$$

To obtain the tighter bound we:

- Use Integral form of Taylor Expansion
- Use Jensens Inequality



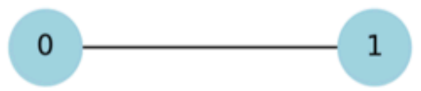
**Corollary 1.** *For a generator $\mathcal{L}$ that describes closed system evolution with the Hamiltonian $H = \sum_s h_s H_s$ such that $h_s \geq 0$ and $H_s$ is normalised and Hermitian the qDRIFT method requires,*

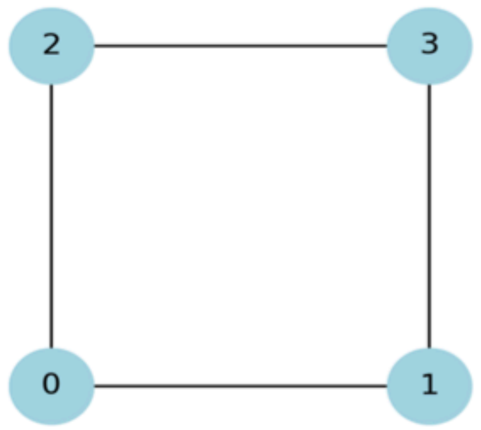
$$r \geq \left\lceil \frac{4t^2\Gamma}{\epsilon} \right\rceil$$

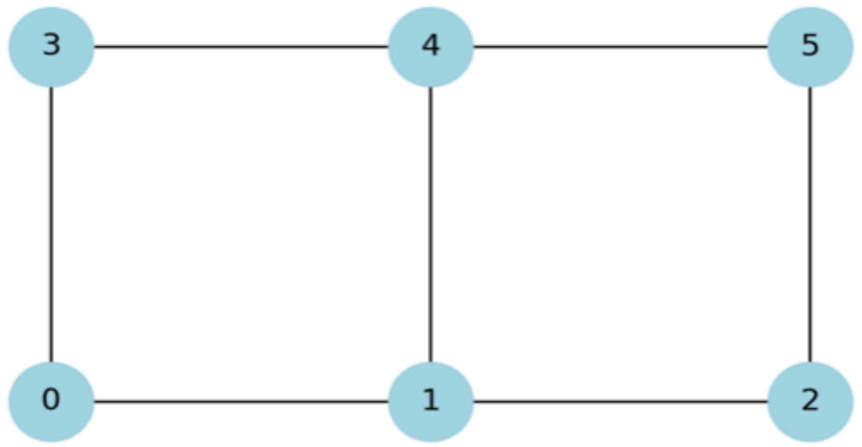
*where $\epsilon > 0$ is the precision.*

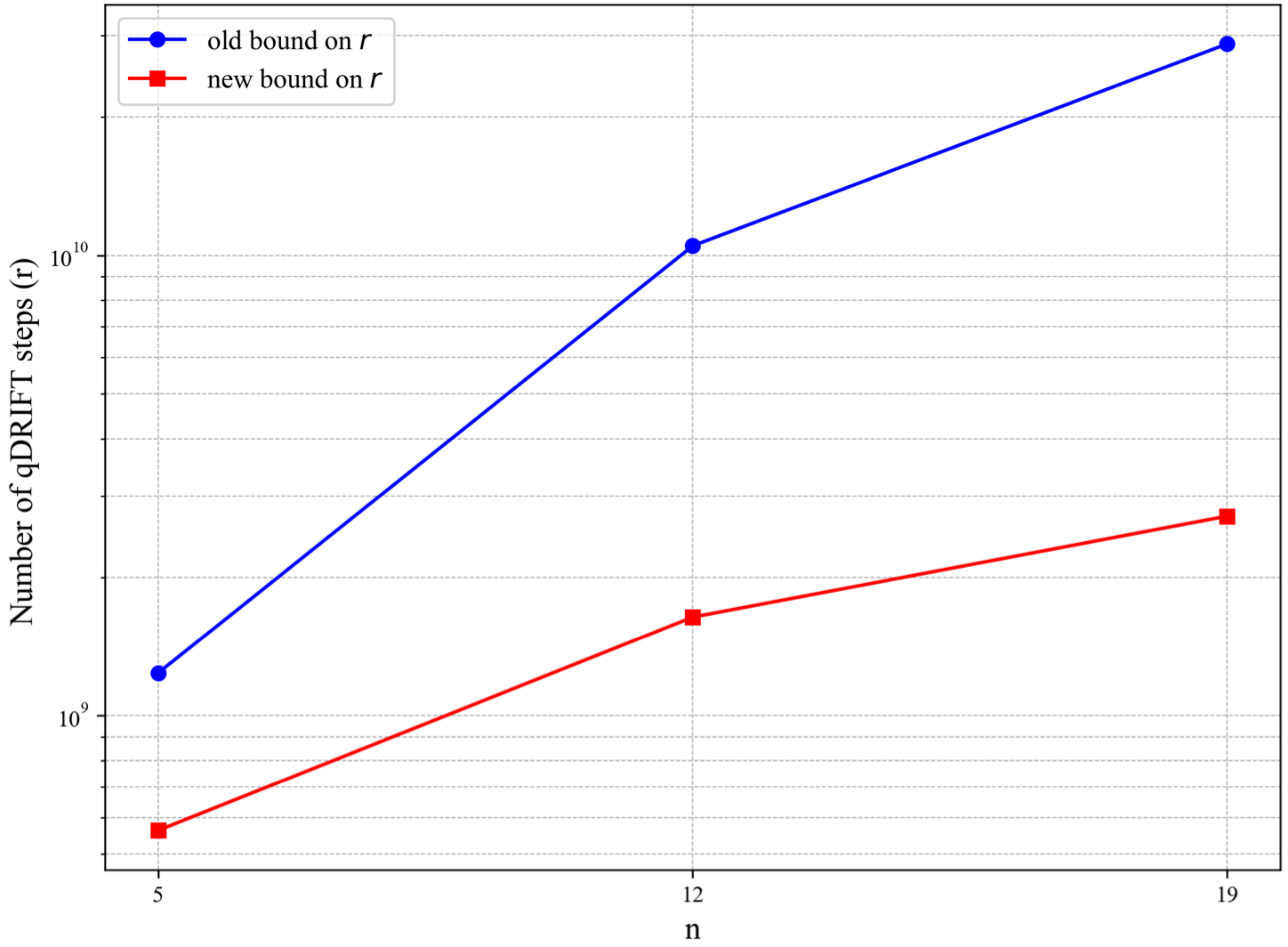
In closed system case we see we obtain linear dependence on the sum of the decay rates in the generator.

$$\mathcal{L}(\rho) = iJ\sum_{\langle i,j \rangle}[Z_i Z_j, \rho] + ih\sum_j [X_j, \rho] + \gamma\sum_j (Z_j \rho Z_j - \rho).$$









# THANK YOU!