



Fullqubit alchemist: *Quantum algorithm for alchemical free energy calculations*

Po-Wei Huang^{1,2}, Gregory Boyd¹, Gian-Luca R. Anselmetti³, Matthias Degroote³, Nikolaj Moll³,
Raffle Santagati³, Michael Streif³, Benjamin Ries³, Daniel Marti-Dafcik^{1,2}, Hamza Jnane^{1,2},
Sophia Simon⁴, Nathan Wiebe^{4,5,6}, Thomas R. Bromley², Bálint Koczor^{1,2}

¹UNIVERSITY OF OXFORD, ²QUANTUM MOTION, ³BOEHRINGER INGELHEIM,

⁴UNIVERSITY OF TORONTO, ⁵PNNL, ⁶CIFAR



UNIVERSITY OF
OXFORD

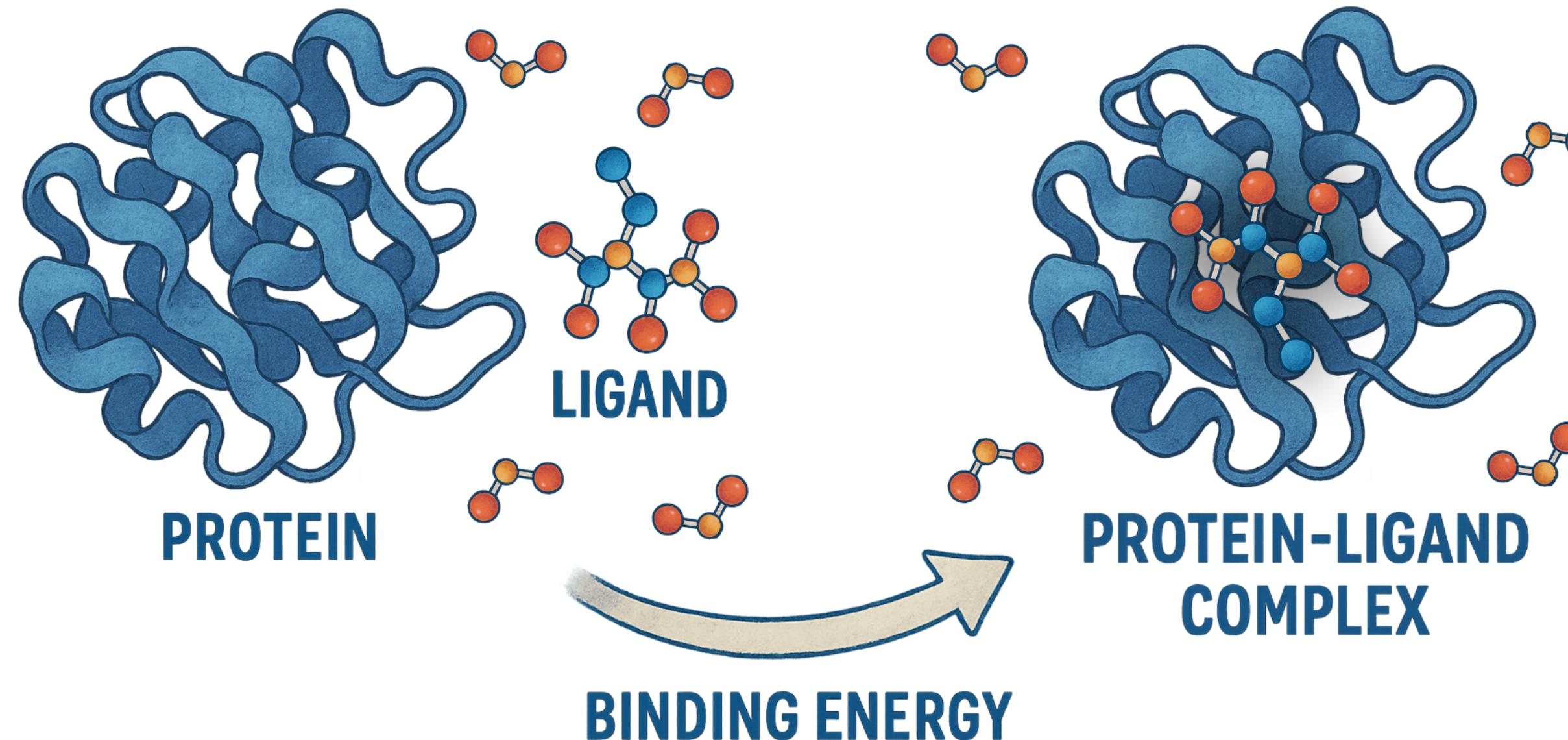


QUANTUM
MOTION

Drug design

Motivation and applications

- Binding free energy is used to rank drug candidates.
- Free energy is computed via thermal averages over many many different molecular configurations.



Talk questions and outline

PART I: Molecular dynamics on a quantum computer

How do we generate the various molecular configurations on the quantum computer?

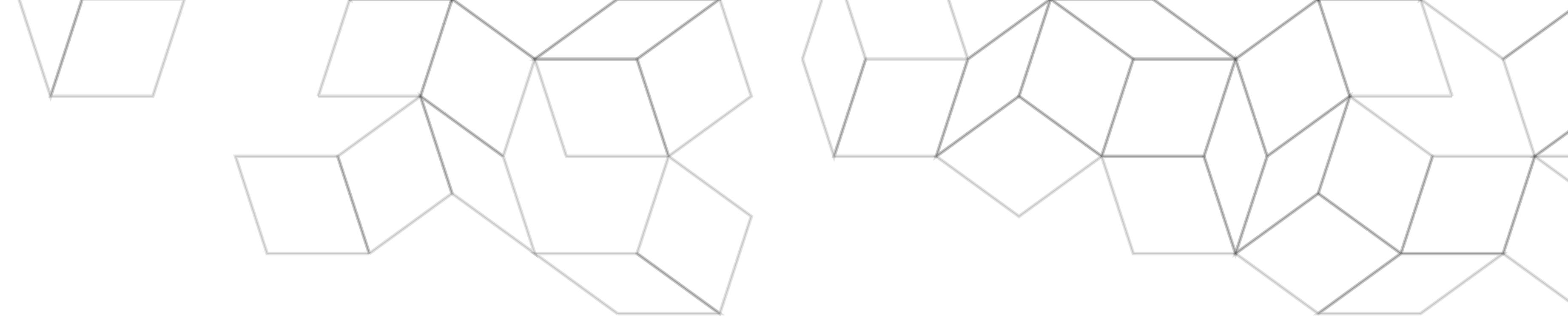
PART II: Alchemical free energy calculations

How do we calculate the *relative* free energy between two different systems?

Let's put alchemy and free energy aside for now...

And focus on how to generate the various molecular configurations





Part I:

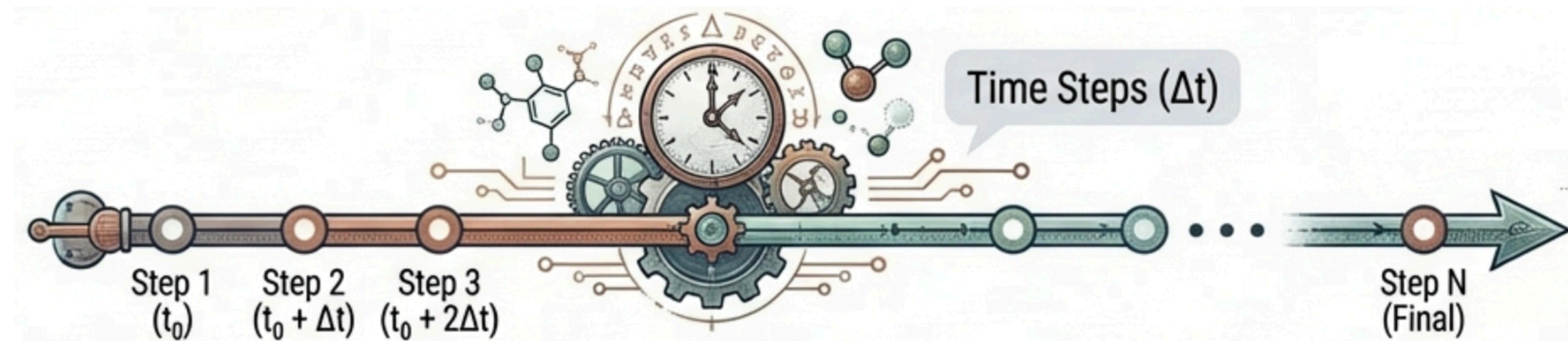
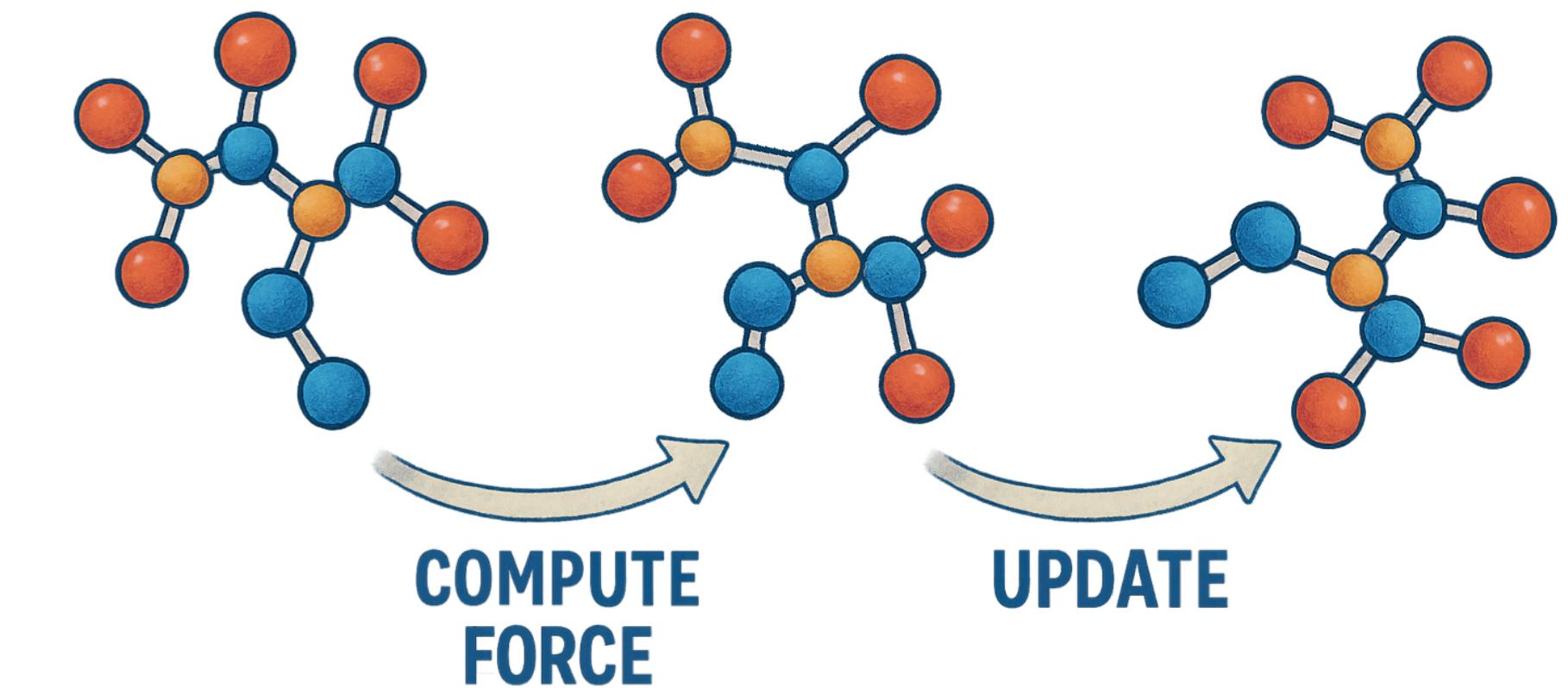
Molecular Dynamics

on a Quantum Computer

Molecular dynamics

In practice for classical computers

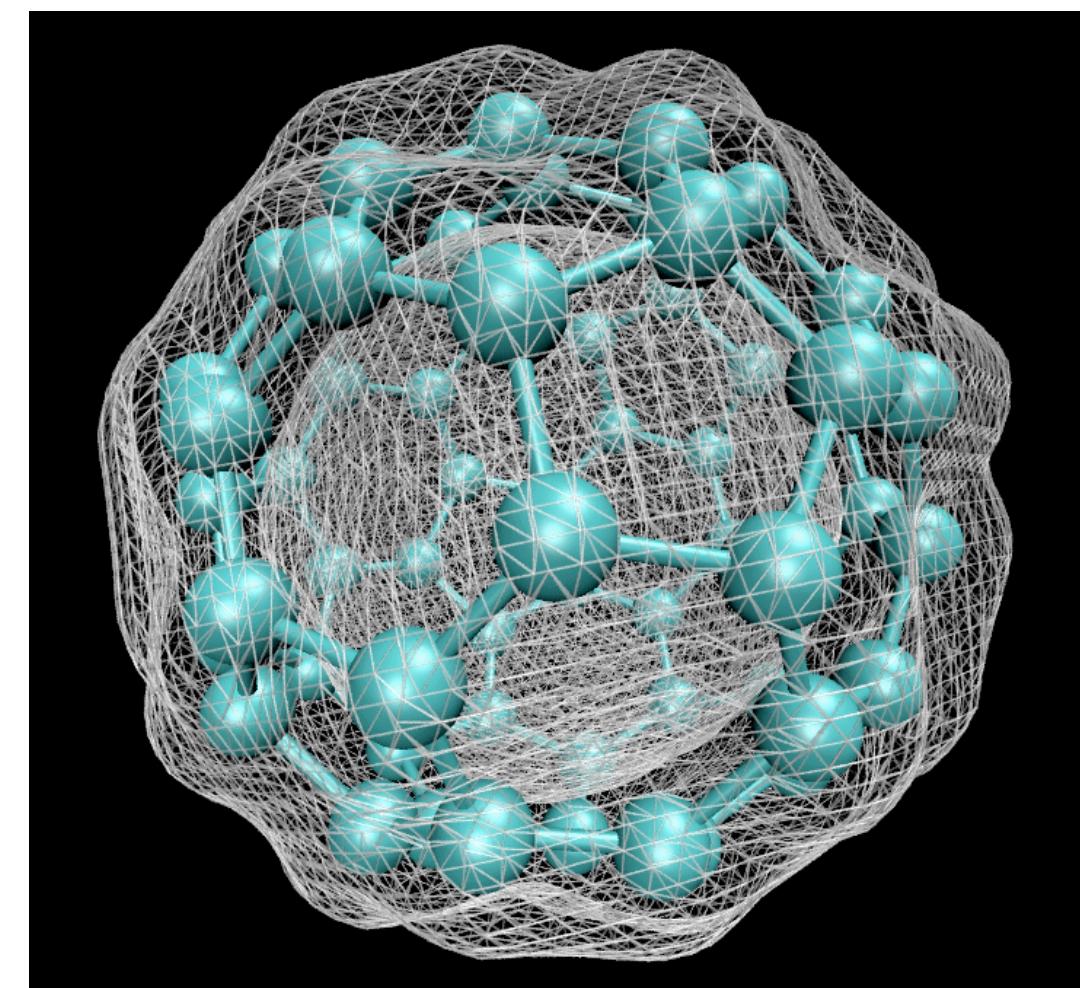
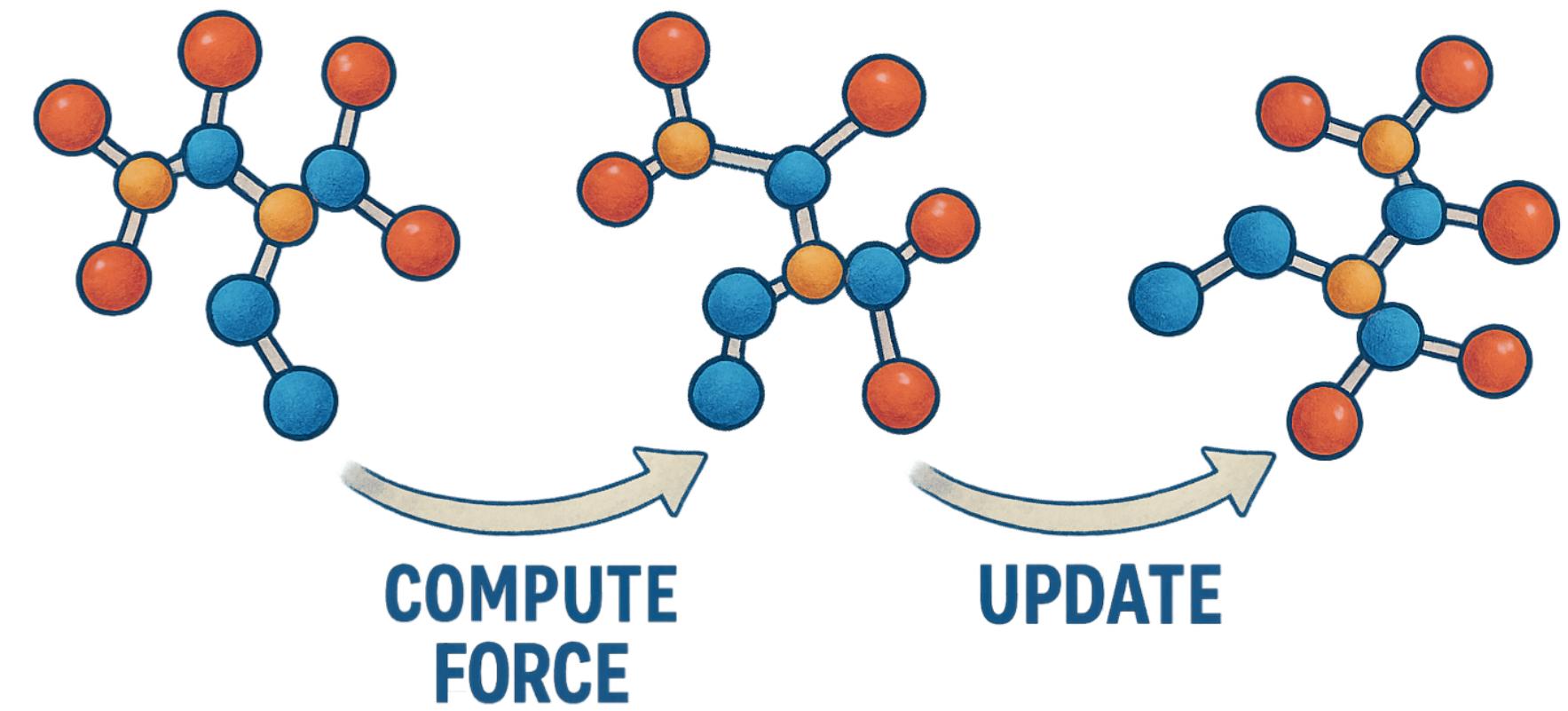
- Propagate particles with Hamiltonian dynamics
- Time average of evolution trajectory to replace ensemble average
 - Lots and lots of compute time!
- Used to compute thermal averages such as free energy



Hybrid quantum semi-classical modeling

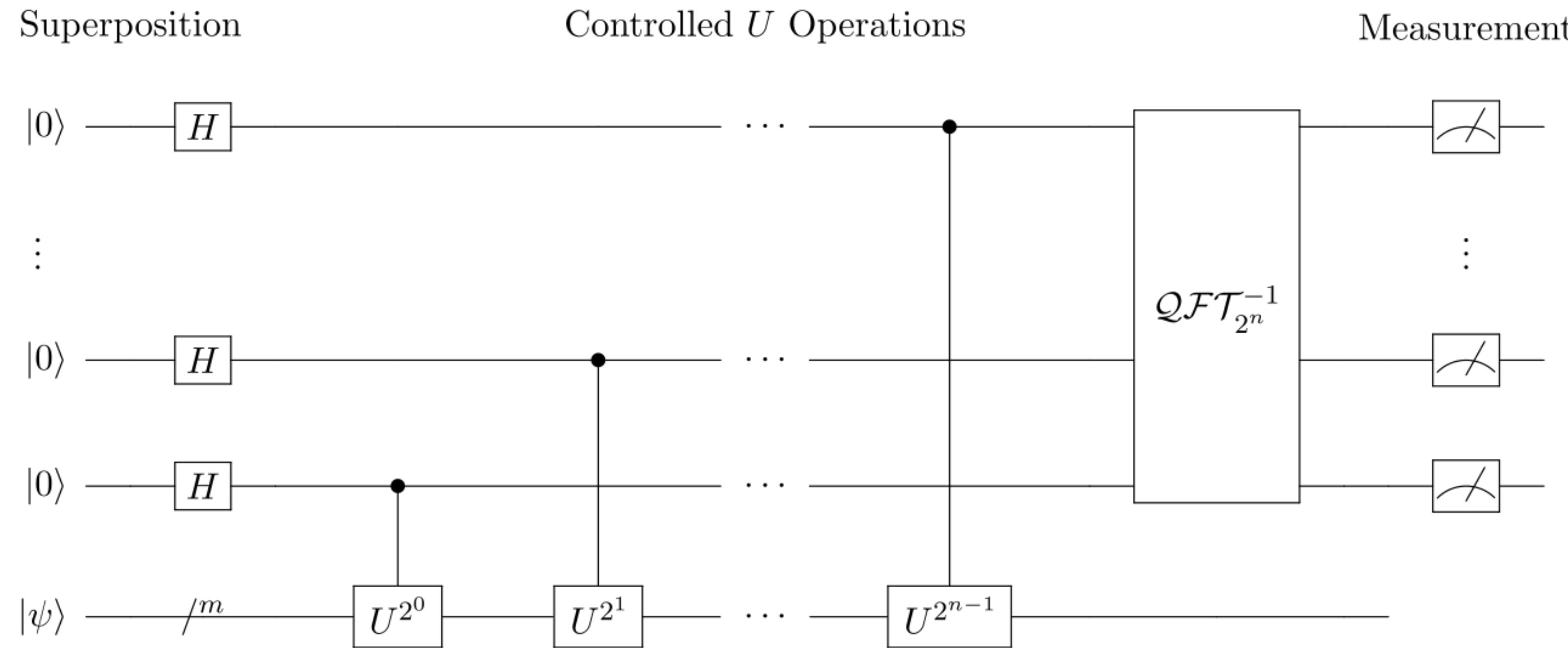
Not all particles are classical...

- While nuclei are heavy enough to be considered classical, electrons whizzing around definitely are not...
- Born-Oppenheimer Approximation:
 - The wave function of the *nuclei* and *electrons* are treated separately.
 - Treat nuclei *classically* → propagate by molecular dynamics
 - Treat electrons *quantumly* → electronic structure calculations (eg DFT, CCSD) of the ground state



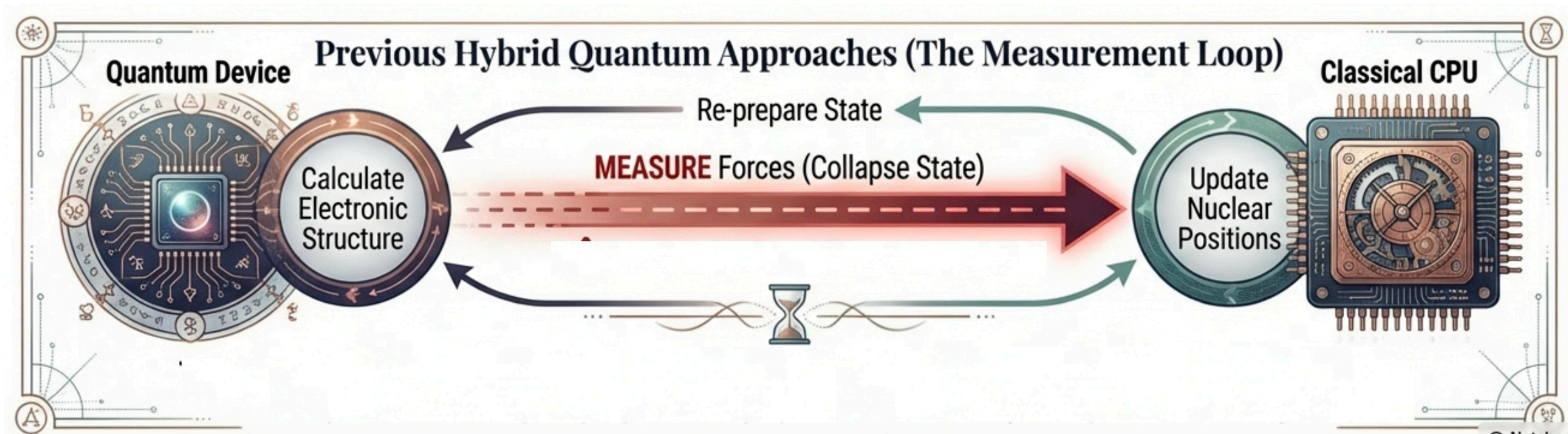
Calculating ground state properties...

Seem familiar?



- Offload ground state energy calculations to quantum computer
- Feed ground state energy calculations to MD step

Case closed?



Is there a coherent approach that doesn't require hybrid loops?
Can we implement molecular dynamics on a quantum computer?

Improved Precision Scaling for Simulating Coupled Quantum-Classical Dynamics

Sophia Simon^{1,*}, Raffaele Santagati,^{2,†} Matthias Degroote,² Nikolaj Moll^{1,‡}, Michael Streif,² and Nathan Wiebe^{3,4,5,‡}

¹*Department of Physics, University of Toronto, Toronto, Ontario M5S 1A7, Canada*

²*Quantum Lab, Boehringer Ingelheim, Ingelheim am Rhein 55218, Germany*

³*Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada*

⁴*Pacific Northwest National Laboratory, Richland, Washington 99354, USA*

⁵*Canadian Institute for Advanced Studies, Toronto, Ontario M4G 4G8, Canada*

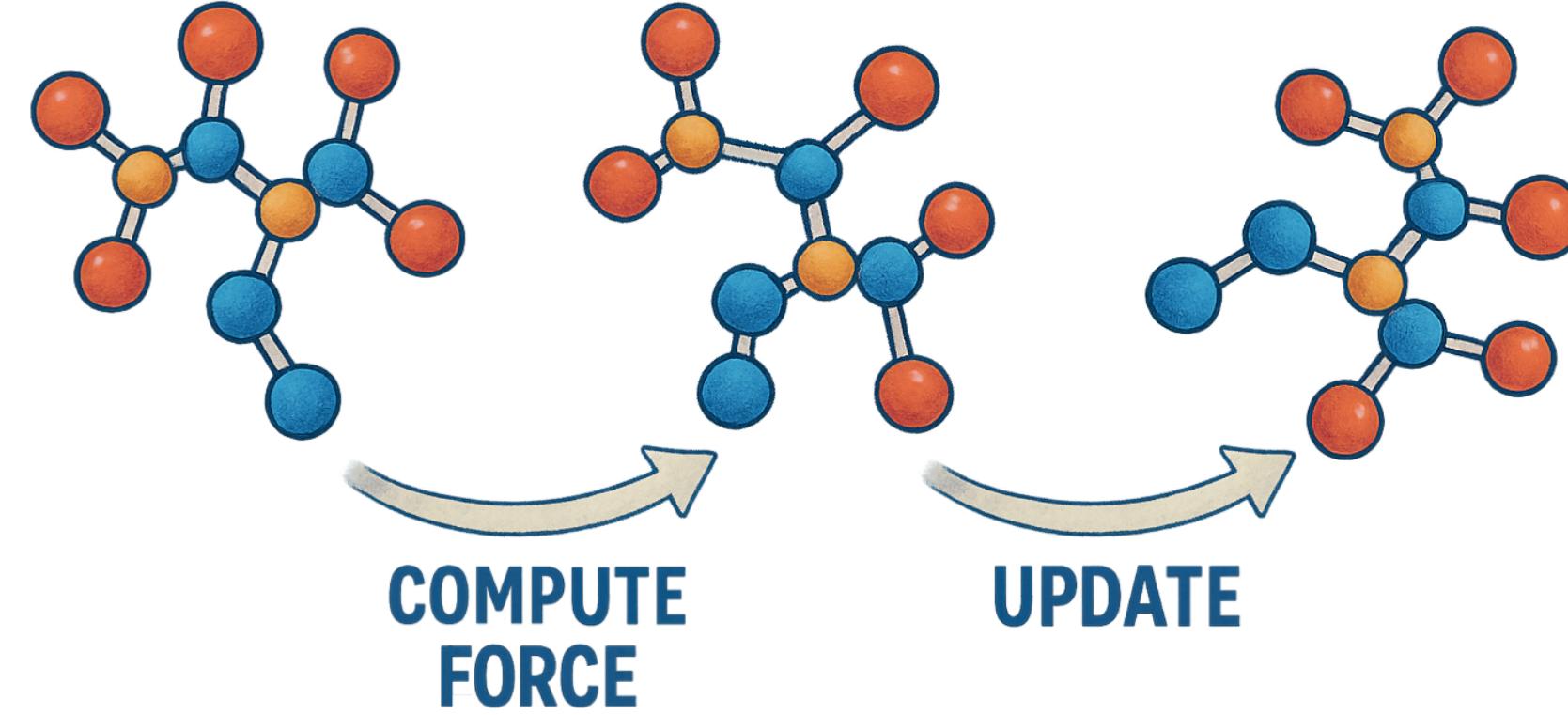


(Received 27 September 2023; accepted 12 February 2024; published 12 March 2024)

Molecular dynamics on a quantum computer

What works and what doesn't

- Large Hilbert space with small number of qubits
- Can simulate/discretize the entire phase-space
- A quantum state can be used to encode an entire probability distribution
- Time-based averages may be hard to implement



Koopman von Neumann mechanics

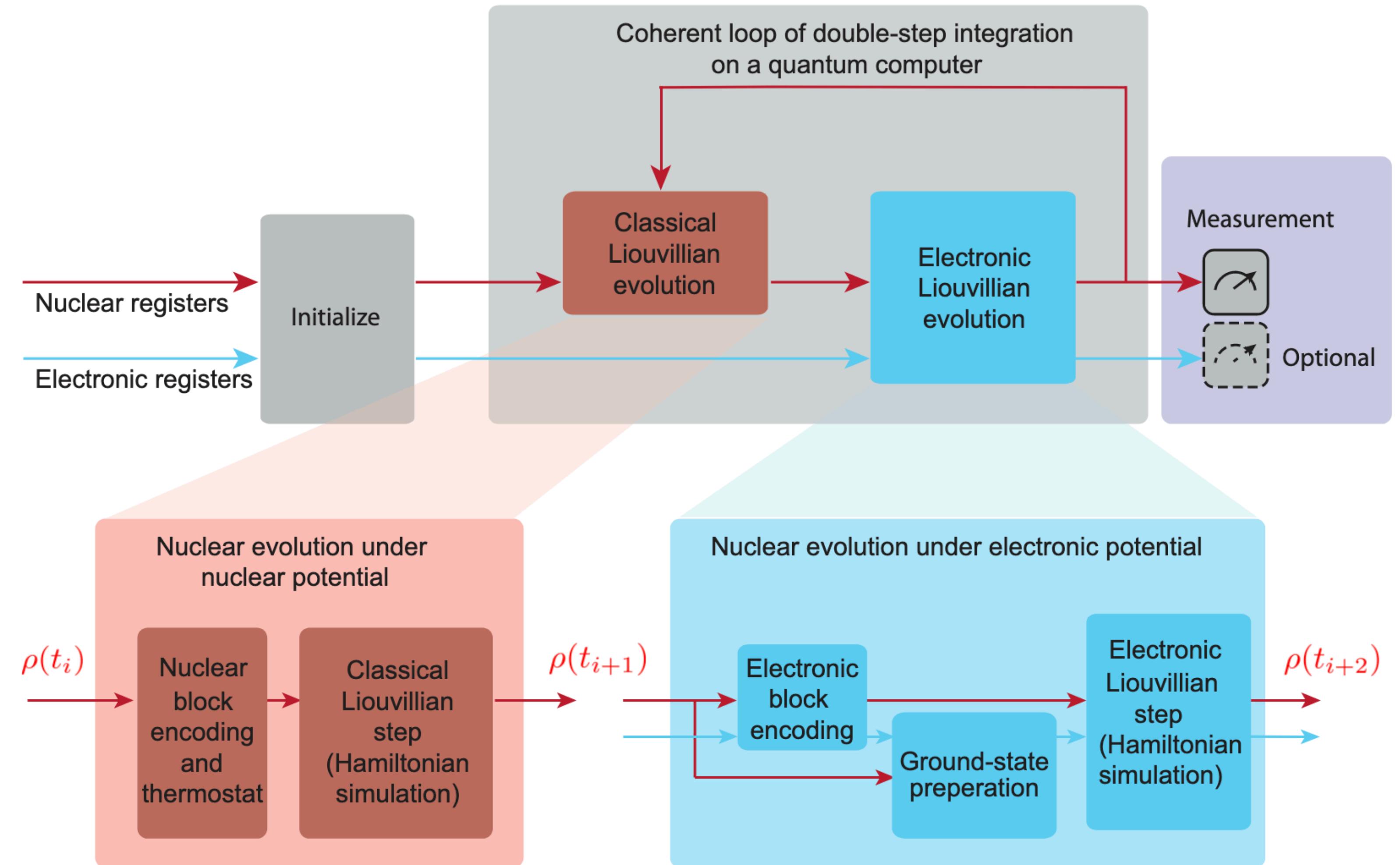
Under the Liouvillian picture

- Describes classical dynamics over a classical probability distribution
- Classical wavefunction $\rho = \psi_{KvN}^*(p, q, t) \psi_{KvN}(p, q, t)$ that encodes distribution
- Evolution of wavefunction by Liouvillian $L = -i \left(\frac{\partial H}{\partial p} \frac{\partial}{\partial x} - \frac{\partial H}{\partial x} \frac{\partial}{\partial p} \right)$
- Evolution dynamics governed by $i \frac{\partial}{\partial t} \psi_{KvN} = L \psi_{KvN}$

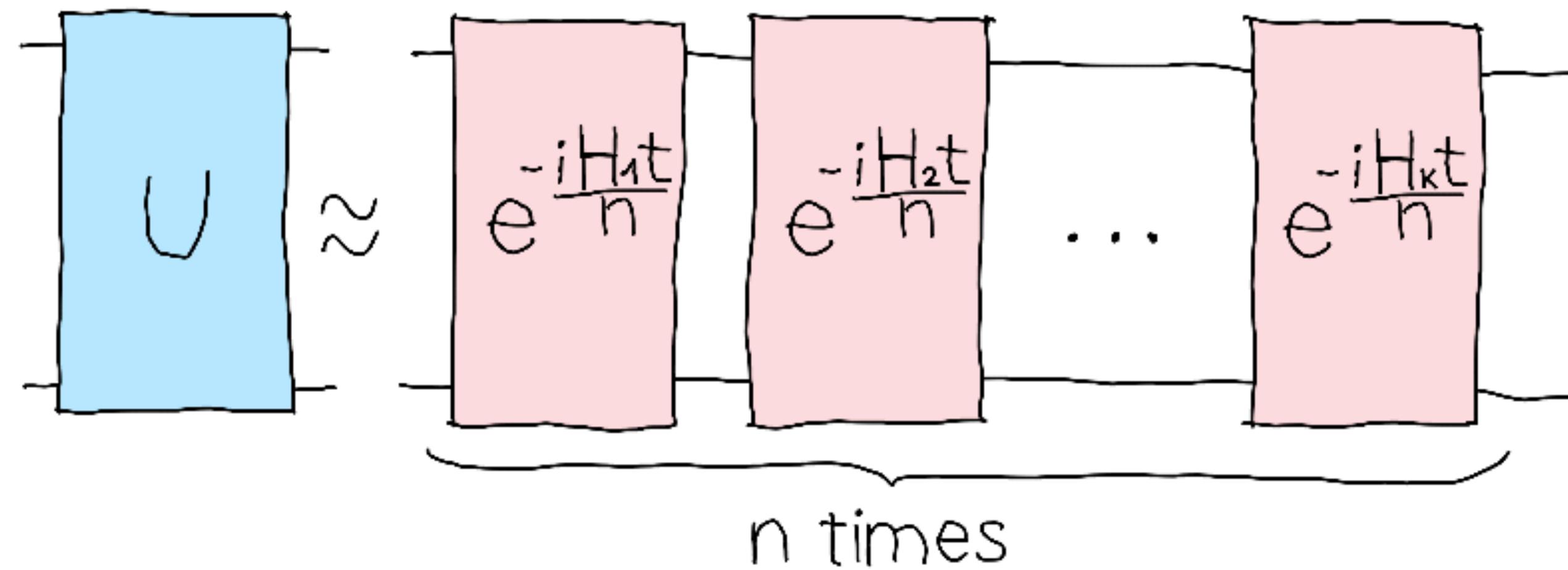
$$\int \rho_t = e^{-iLt} \int \rho_0$$

Simulating coupled quantum-classical dynamics

- Separate Liouvillian into classical and electronic parts, and evolve with small time steps individually.
- No more hybrid loop, but alternating time steps on a quantum computer



Case closed again?

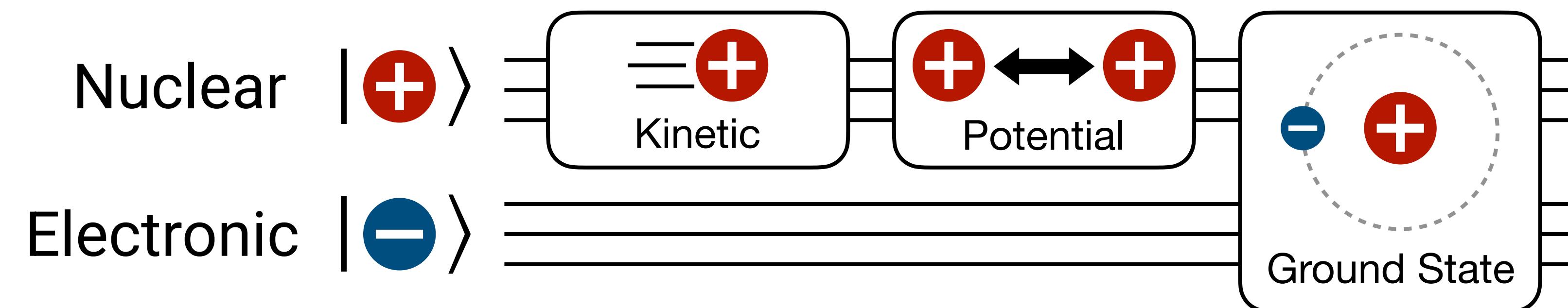


Can we do better than Trotterization?

Molecular dynamics on a quantum computer

Or, Liouvillian block-encodings and Liouvillian simulation

- Two register sets on quantum computer to implement the Liouvillian:
 - “Classical” nuclear register – Corresponds to classical MD in Liouvillian formalism on exponential sized grid
 - “Quantum” electronic register – Corresponds to ground state energy/forces calculations



Writing down the terms

Aka the slide with a lot of math

$$H_{\text{ext}}^{(\text{NVT})} = \sum_{n=1}^N \sum_{j=1}^3 \frac{p_{n,j}'^2}{2m_n s^2} + \sum_{k=1}^{N-1} \sum_{n=k+1}^N \frac{Z_n Z_k}{\|x_n - x_k\|} + \frac{p_s^2}{2Q} + N_f k_B T \ln(s) + E_{\text{el}}(\mathbf{x}),$$

$$L_{\text{cl,disc}} := -i \sum_{n=1}^N \sum_{j=1}^3 \left(D_{x_{n,j}} \otimes \frac{\partial H_{\text{cl}}}{\partial p'_{n,j}} - \frac{\partial H_{\text{cl}}}{\partial x_{n,j}} \otimes D_{p'_{n,j}} \right) - i \left(D_s \otimes \frac{\partial H_{\text{cl}}}{\partial p_s} - \frac{\partial H_{\text{cl}}}{\partial s} \otimes D_{p_s} \right), \quad (\text{A.11})$$

$$L_{\text{el}} = L - L_{\text{cl}} := i \sum_{n=1}^N \sum_{j=1}^3 \frac{\partial E_{\text{el}}(\mathbf{x})}{\partial x_{n,j}} \partial_{p'_{n,j}}$$

$$\frac{\partial H_{\text{cl}}}{\partial x_{n,j}} = \sum_{n' \neq n} \sum_{\bar{s}} \frac{-Z_n Z_{n'}}{\|x_n - \bar{x}_{n'}\|^{3/2}} \quad (\text{A.12})$$

$$\frac{\partial H_{\text{cl}}}{\partial p'_{n,j}} = \sum_{\bar{p}'_{n,j}} \sum_{\bar{s}} | \bar{s} \rangle \langle \bar{s} | \otimes | \bar{s} \rangle \langle \bar{s} | \quad (\text{A.13})$$

$$\frac{\partial H_{\text{cl}}}{\partial s} = \sum_{\bar{p}_{n,j}} \sum_{\bar{s}} | \bar{s} \rangle \langle \bar{s} | \otimes | \bar{s} \rangle \langle \bar{s} | \quad (\text{A.14})$$

$$\frac{\partial H_{\text{cl}}}{\partial p_s} = \sum_{\bar{p}_s} \frac{p_s}{Q} | \bar{p}_s \rangle \langle \bar{p}_s | \quad (\text{A.15})$$

Calculating ground state energy derivatives

Hellmann-Feynman Theorem

$$\frac{\partial E_n}{\partial \lambda} = \langle \psi_n | \frac{\partial H}{\partial \lambda} | \psi_n \rangle$$

PHYSICAL REVIEW RESEARCH **4**, 043210 (2022)

Near-optimal ground state preparation

Lin Lin^{1,2} and Yu Tong¹

¹Department of Mathematics, University of California, Berkeley, CA 94720, USA

²Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Efficient quantum computation of molecular forces and other energy gradients

Thomas E. O'Brien^{1,*,†} Michael Streif,^{2,†} Nicholas C. Rubin,^{1,‡} Raffaele Santagati^{1,§} Yuan Su,¹ William J. Huggins,¹ Joshua J. Goings¹ Nikolaj Moll¹ Elica Kyoseva,² Matthias Degroote¹ Christofer S. Tautermann^{1,||} Joonho Lee^{1,¶} Dominic W. Berry,⁵ Nathan Wiebe,^{6,7,||} and Ryan Babbush^{1,¶}

¹Google Research, Venice, California 90291, USA

²Quantum Lab, Boehringer Ingelheim, 55218 Ingelheim am Rhein, Germany

³Boehringer Ingelheim Pharma GmbH & Co KG, Birkendorfer Strasse 65, 88397 Biberach, Germany

⁴Department of Chemistry, Columbia University, New York, USA

⁵School of Mathematical and Physical Sciences, Macquarie University, New South Wales 2109, Australia

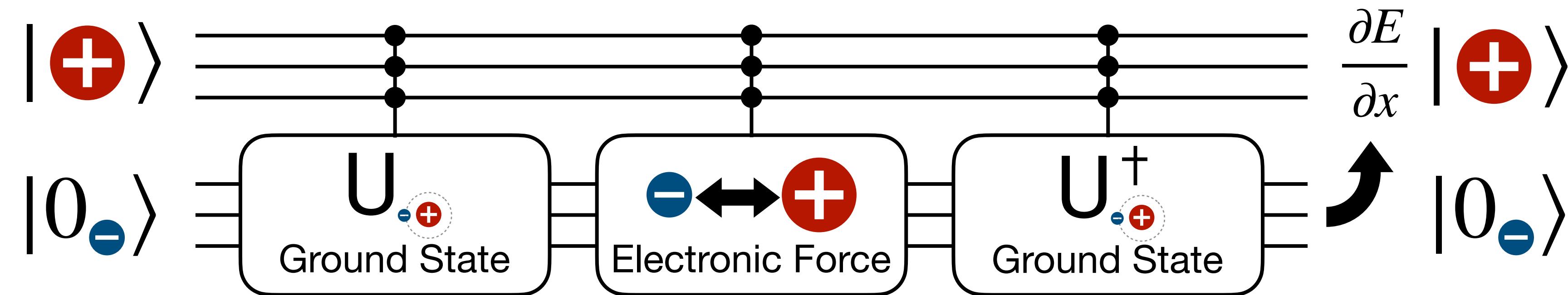
⁶Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4

⁷Pacific Northwest National Laboratory, Richland, Washington 99354, USA

Block-encoding energies without QPE

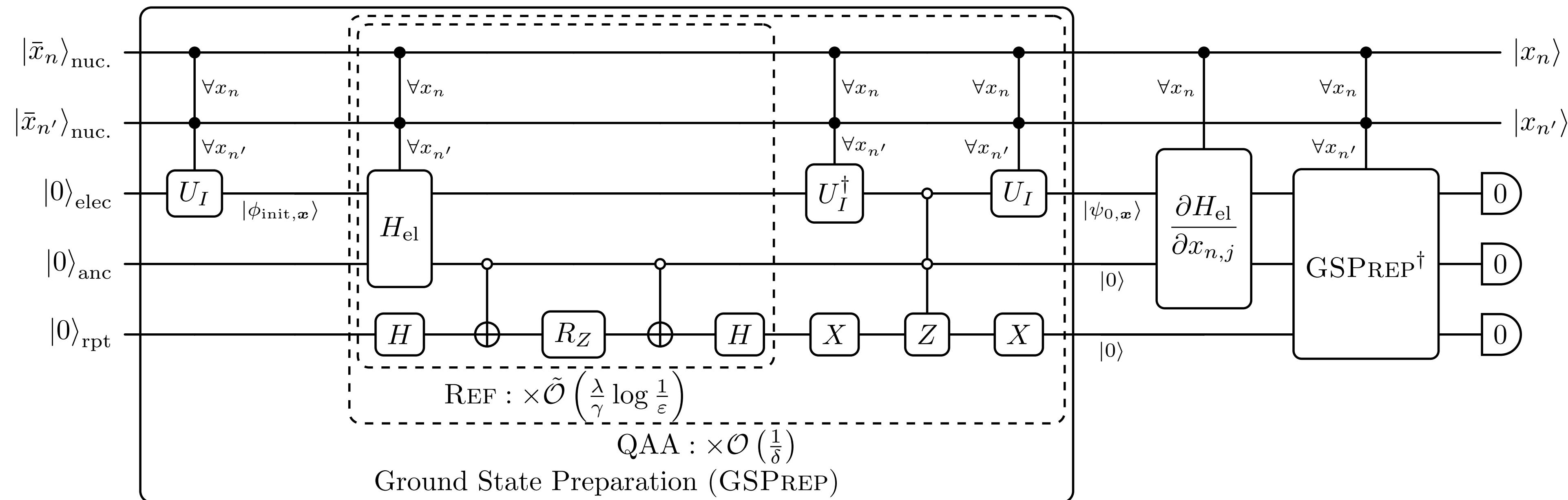
Another way to obtain ground state energies/forces

- Obtain quantum mechanical electronic forces via kicking back results from electronic register to nuclear register
- Compute ground state energy by preparing ground state, applying the Hamiltonian, then unpreparing the ground state
- Compute electronic forces with Hellmann-Feynman theorem and analytical derivative of first-quantized Hamiltonian

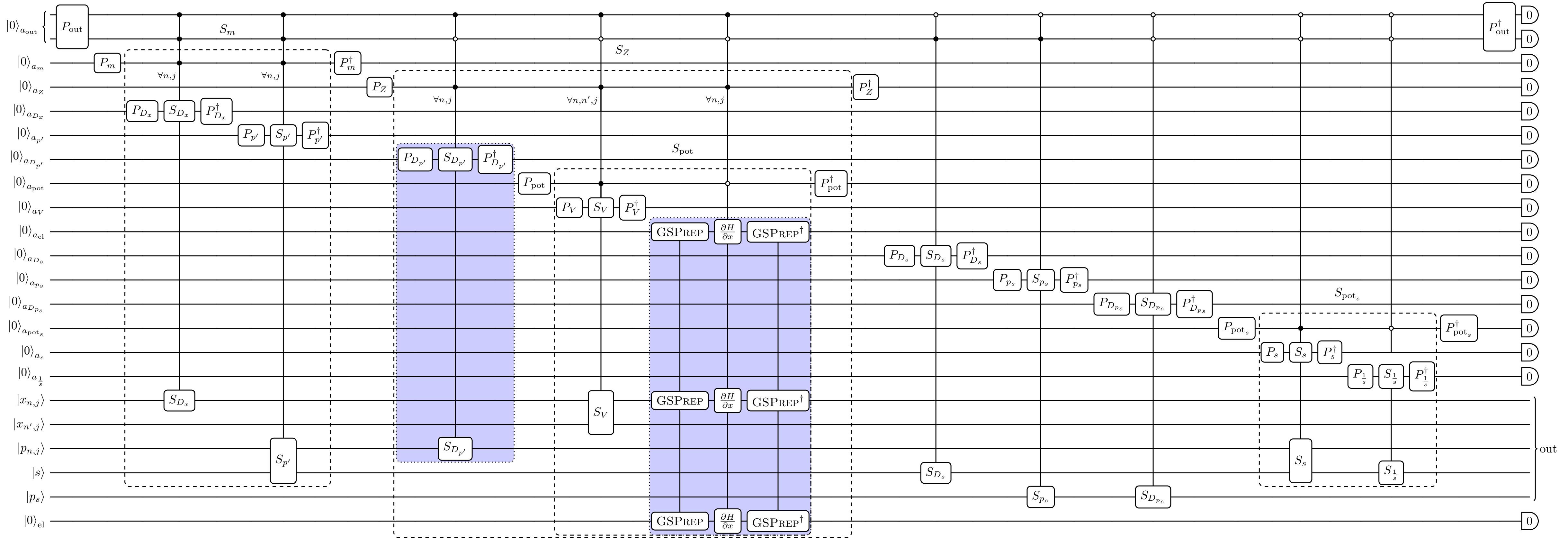


Implementation of energy derivatives

- Block-encoding of first-quantized Hamiltonian derivative found in [OSR+22]
- Ground state preparation algorithm by [LT20]: Eigenstate filtering+AA

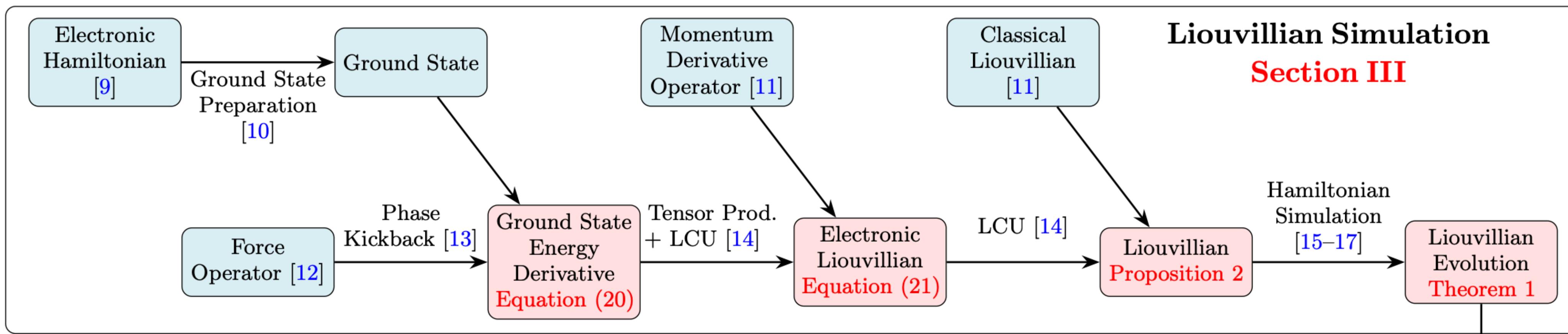
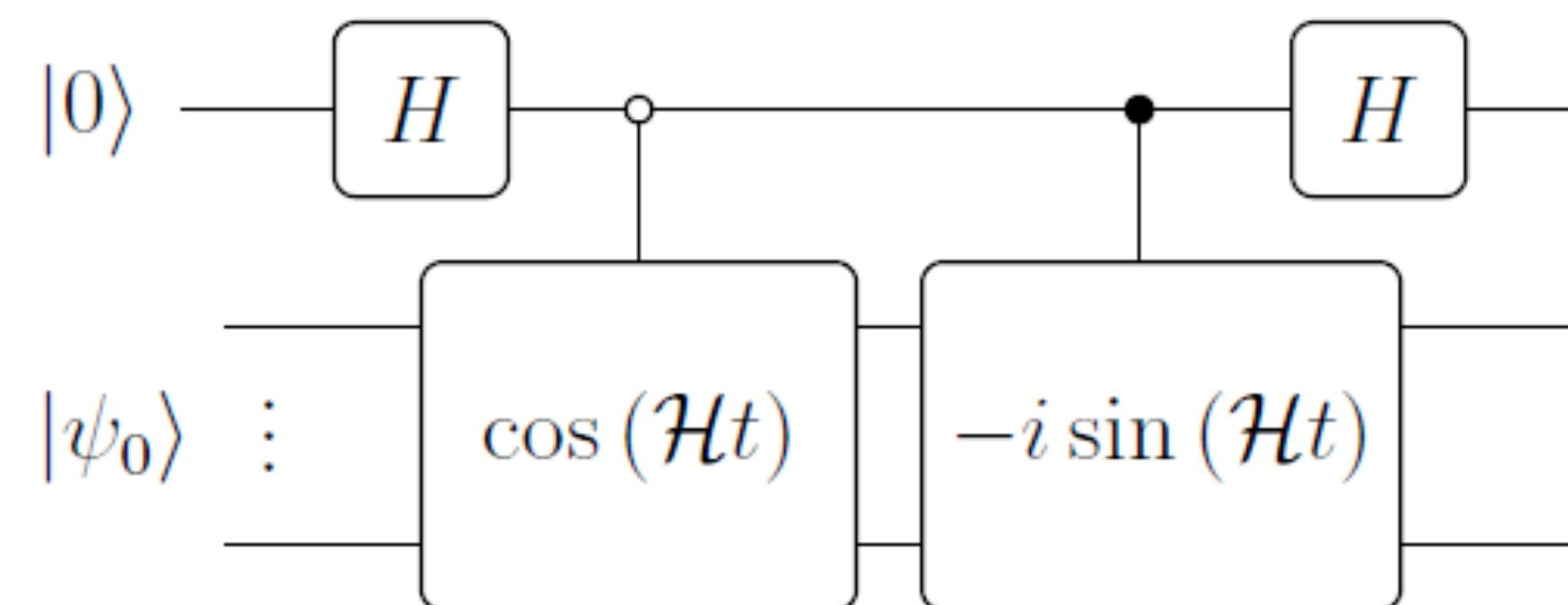


Full implementation of the Liouvillian



Tying everything together...

- Liouvillian simulation via QSVT-based Hamiltonian simulation





Part II:

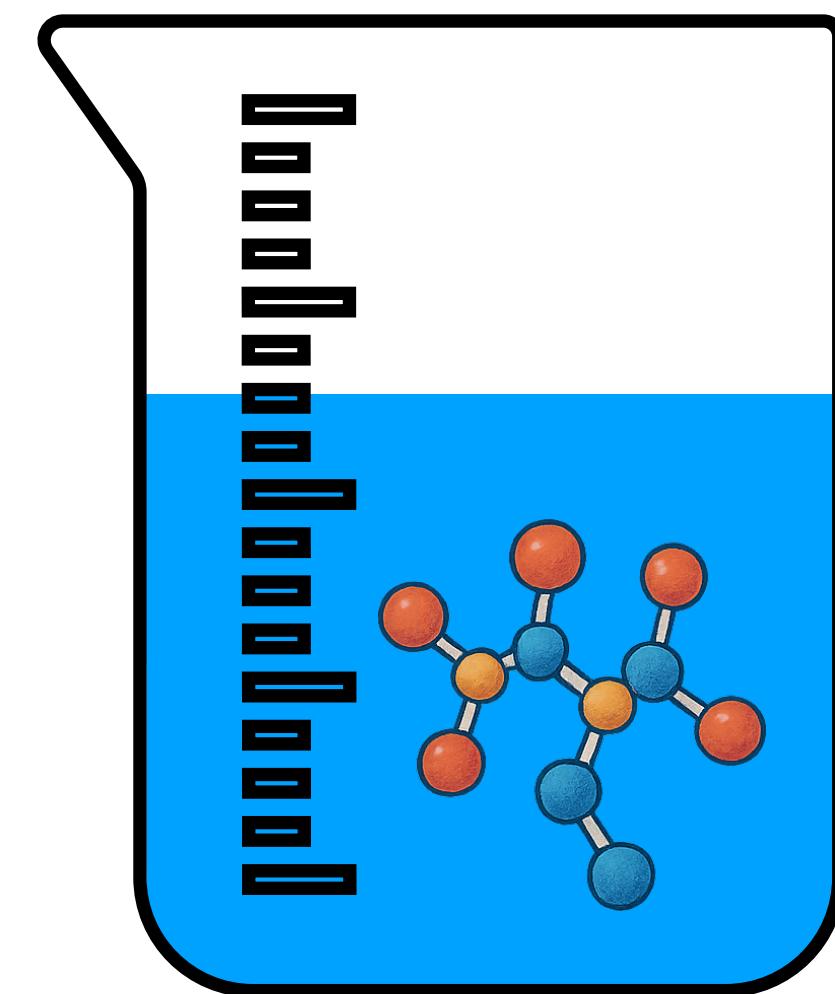
Alchemical Free Energy Calculation

Free Energy

Helmholtz and Gibbs

$$F = U - TS$$

Temperature
↓
Helmholtz Free Energy → F = U - TS ← Entropy
↑
Internal Energy



Gibbs Free Energy \rightarrow $G = U - TS + PV \leftarrow$ Volume
 \uparrow
Pressure

Thermodynamic Integration

The alchemical way to calculate free energy

$$F = U - TS$$

$$F = -k_B T \ln Z$$

- Absolute free energy difficult to compute
 - Either compute the entropy or partition function

$$\begin{aligned}\Delta F_{A \rightarrow B} &= \int_0^1 \frac{\partial F(\Lambda)}{\partial \Lambda} d\Lambda \\ &= -k_B T \int_0^1 \frac{1}{Z} \frac{\partial Z}{\partial \Lambda} d\Lambda \\ &= \int_0^1 \frac{1}{Z} \sum_S \frac{\partial E_S(\Lambda)}{\partial \Lambda} e^{-E_S(\Lambda)/k_B T} d\Lambda \\ &= \int_0^1 \left\langle \frac{\partial E(\Lambda)}{\partial \Lambda} \right\rangle_\Lambda d\Lambda, \\ &= \int_0^1 \langle E(B) - E(A) \rangle_\Lambda d\Lambda.\end{aligned}$$

Thermodynamic Integration: An sketch

System A



System B

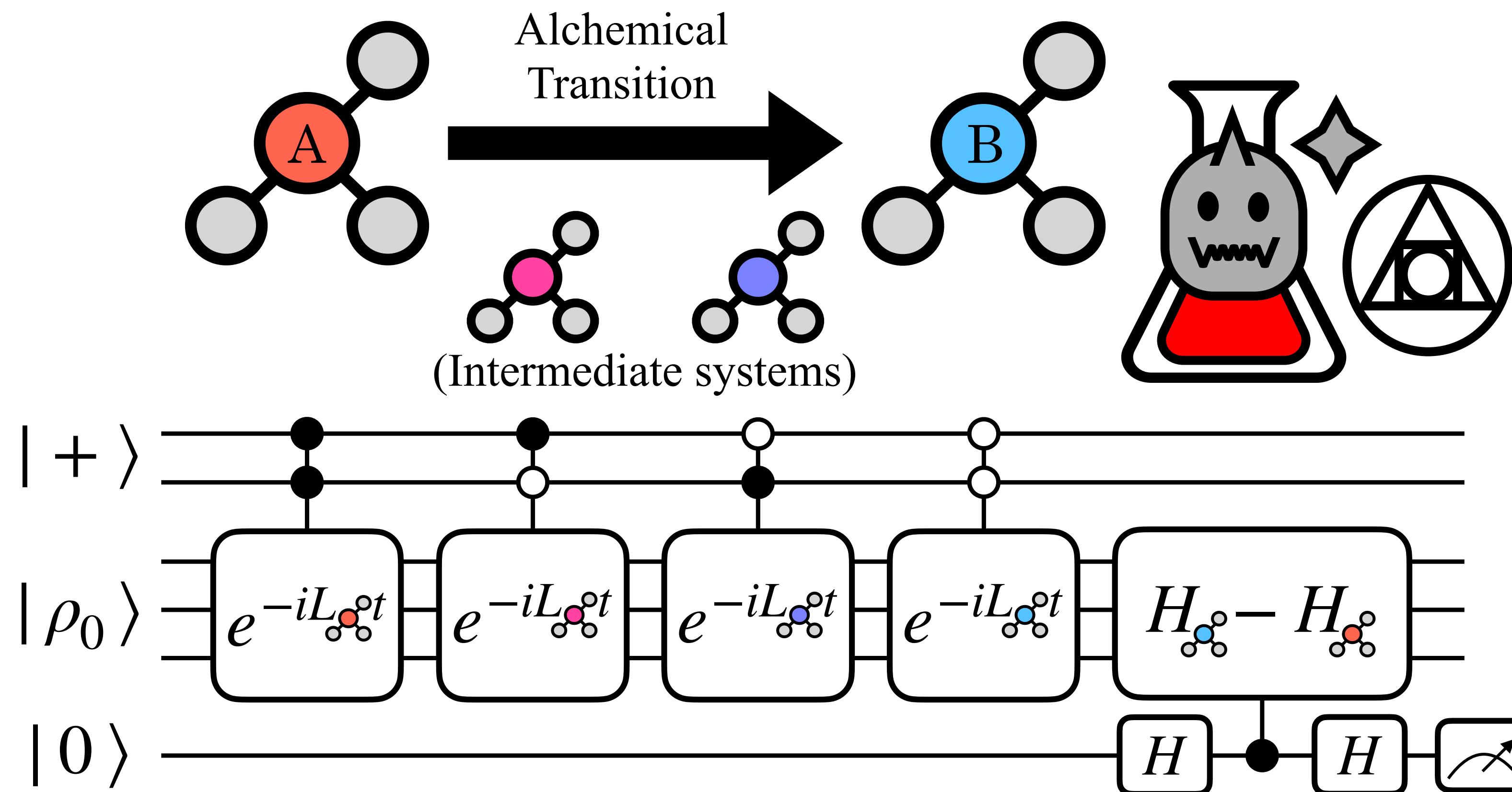


Non-physical intermediate state

Calculate the probability distributions under each (intermediate) system

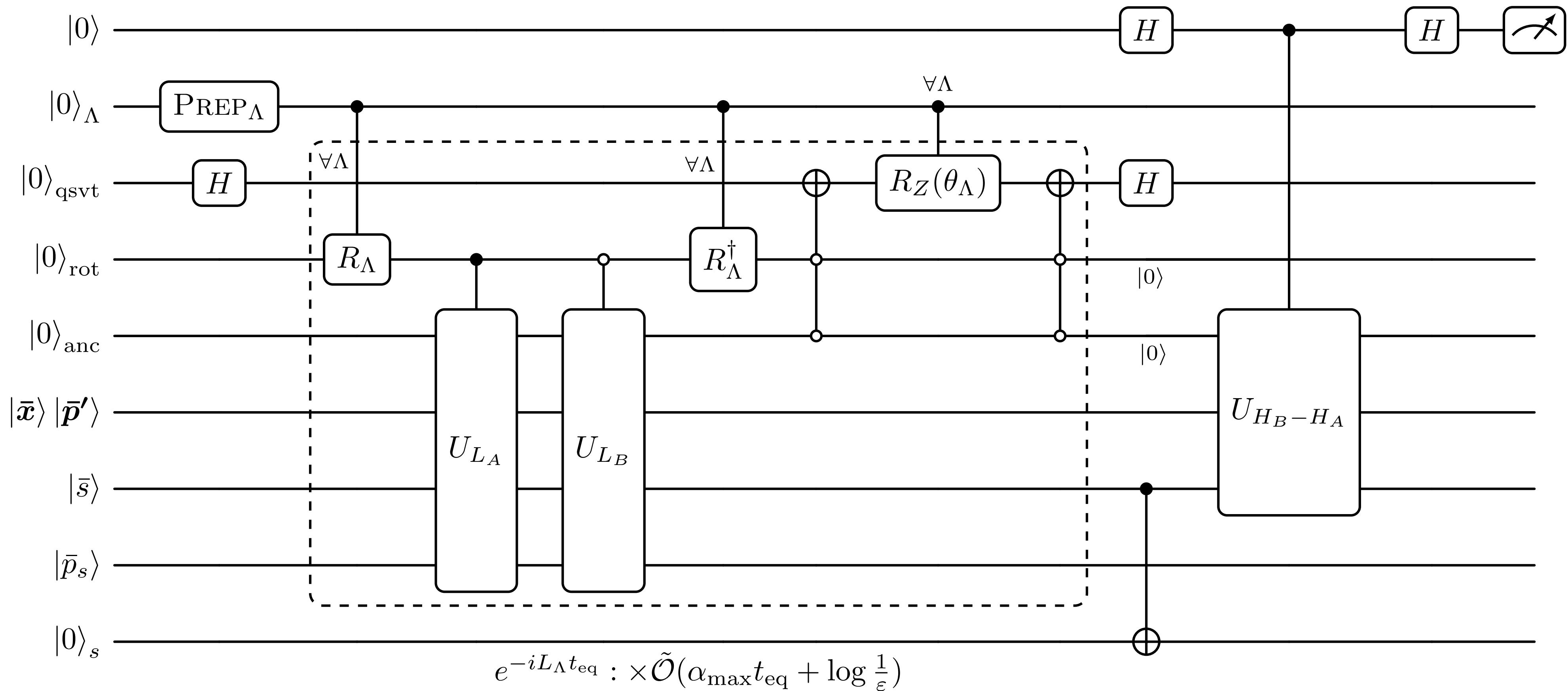
Calculating free-energy differences

Thermodynamic integration on a quantum computer

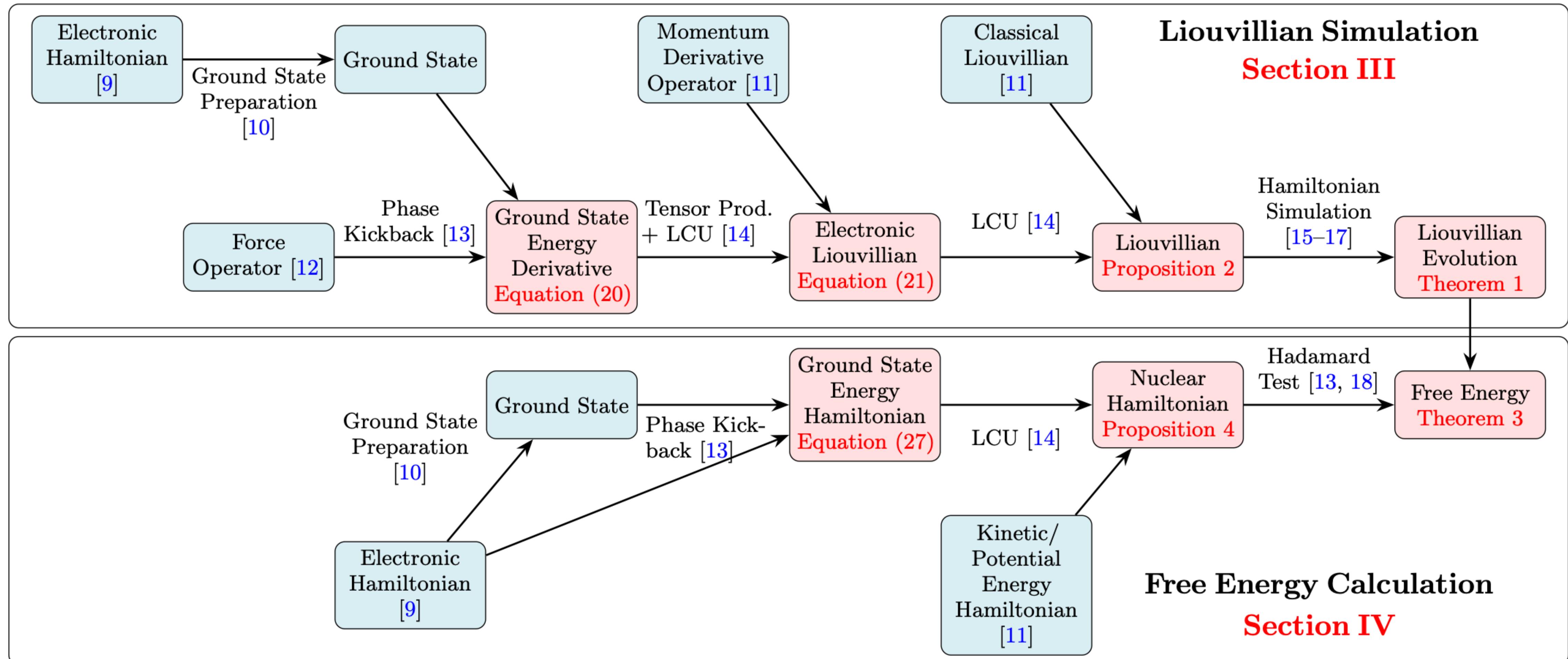


Controlled Liouvillian simulations

How the algorithm doesn't scale linear to the interpolation points



Algorithm breakdown



Performance comparison

With other quantum algorithms and advantages over classical

- Exponential improvements in precision over iterating between MD time step and electronic structure calculations (with both classical and quantum computing methods)
- Superpolynomial improvements over previous Trotterized Liouvillian approach
- Bypass entropy estimation via relative free energy calculations (no sampling over exponential phase-space)

	MD simulation	Free energy calculation
QPE + Euler Int. [1]	$\tilde{\mathcal{O}}(\varepsilon^{-1})$	—
Liouville + Trott. [2]	$\tilde{\mathcal{O}}(\varepsilon^{-o(1)})$	$\tilde{\mathcal{O}}(\eta^{1+o(1)}\varepsilon^{-1.5} + \varepsilon^{-2})$
Our work	$\mathcal{O}(\log^3 \varepsilon^{-1})$	$\tilde{\mathcal{O}}(\varepsilon^{-1})$