BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

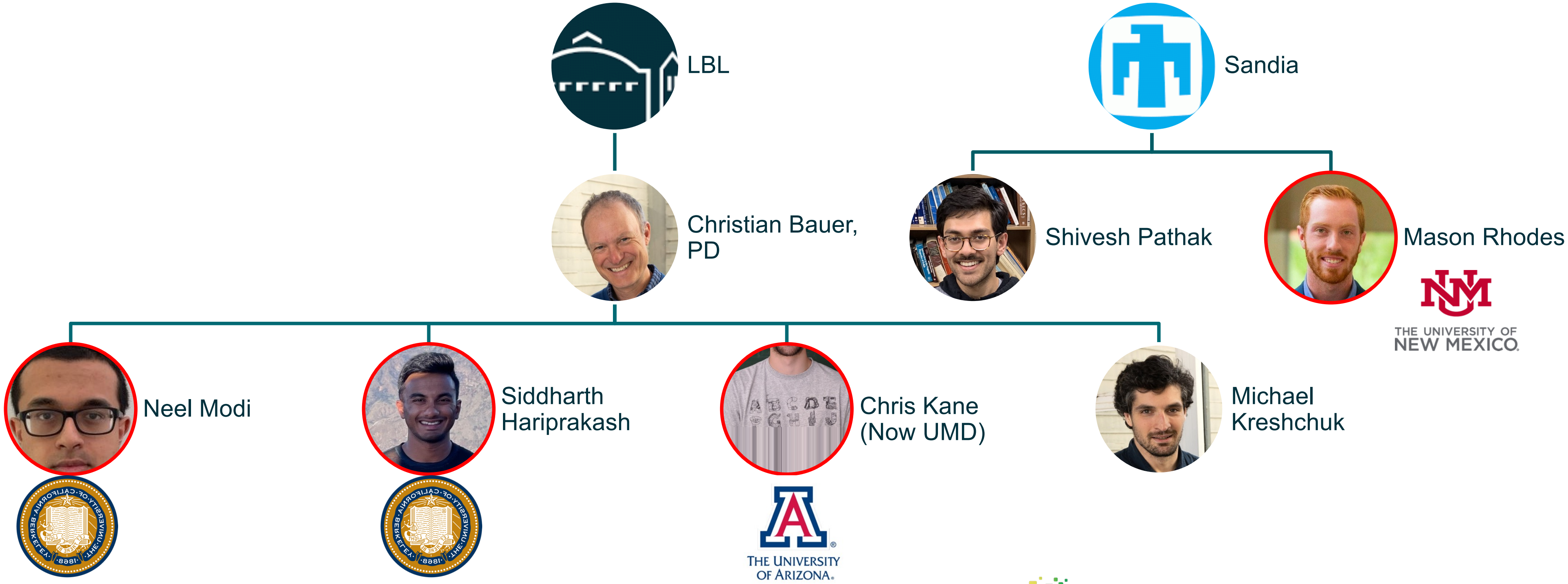
Near-Optimal Simulation of Quantum Field Theory

Michael Kreshchuk, Physics Division,

Lawrence Berkeley National Laboratory

QuantHEP,
September 2, 2024

Collaboration



QUANTUM SYSTEMS ACCELERATOR
Catalyzing the Quantum Ecosystem

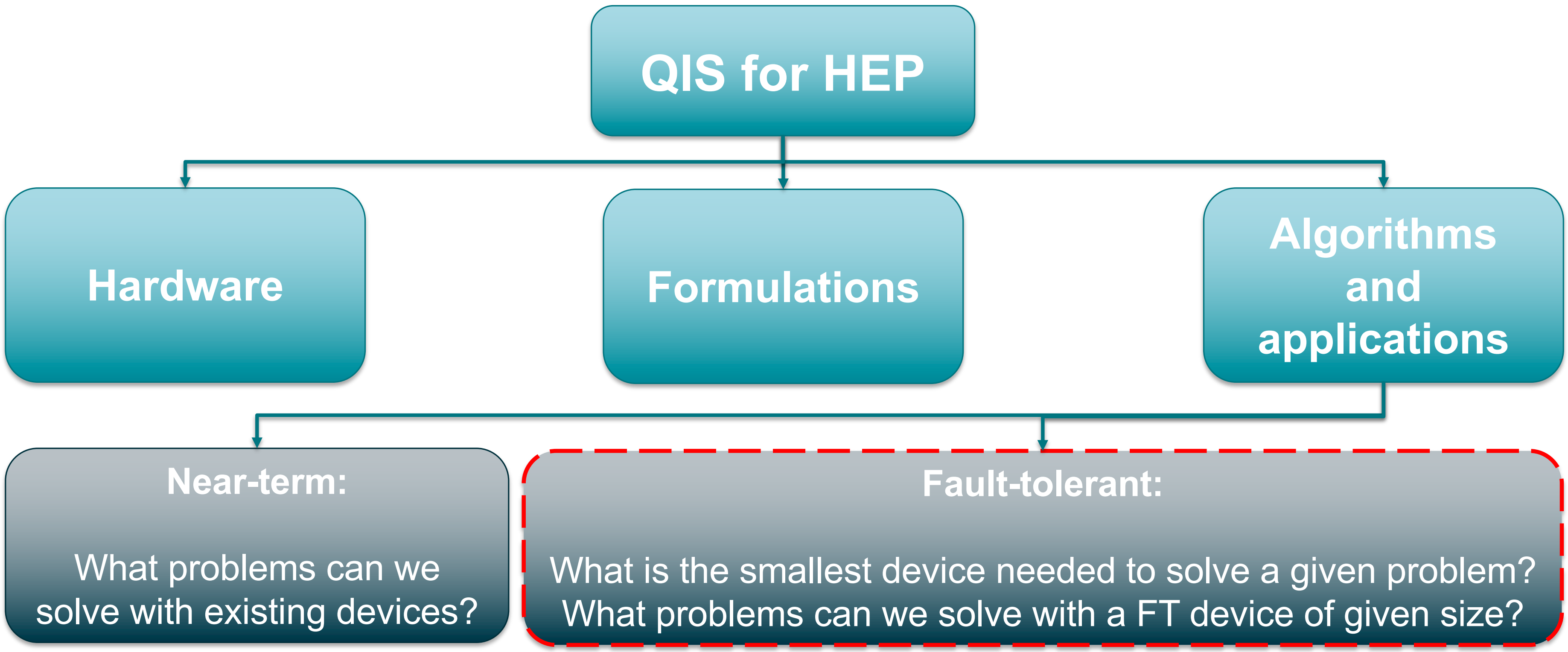
QNM-i

Outline

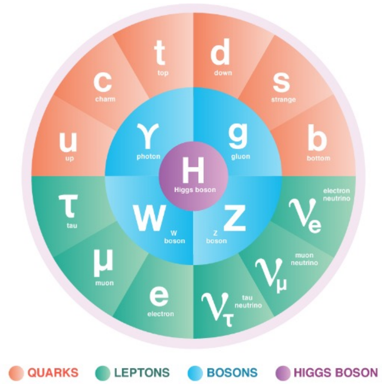
1. Motivation
2. Introduction to algorithms
3. Applications & Developments

1. Motivation

Focus of this talk



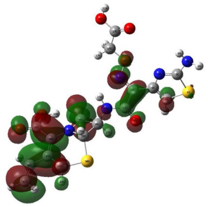
What makes QSim of HEP hard?

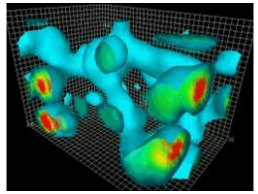


- Highly non-trivial model development.
- Large number of Degrees of Freedom (DOFs), both fermions and bosons:
  → requires a lot of qubits & long circuits.
- Various types of interactions, depending on the formulation.
- Naïve approaches lead to circuits with $\sim 10^{40\text{–}50}$ gates for realistic theories and problem sizes.

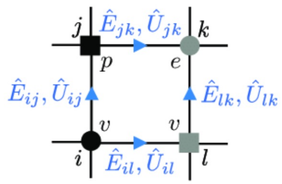
What makes QSim of HEP nice?

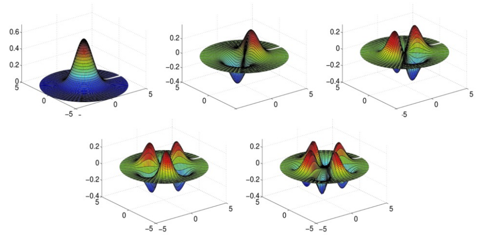
- Fundamental interactions are local
  → Problems are often well-structured.

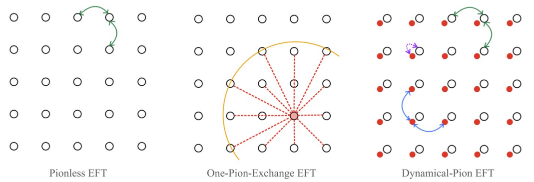


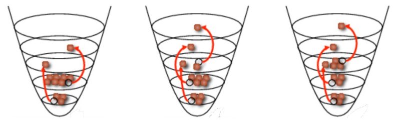


- Algorithms for Relativisitic QFTs can be readily utilized in other settings, e.g., for low-energy NP EFTs.









2. Algorithms

Why "near-optimal"

- Complexity of Trotter time evolution (1901.00564, 1912.08854):
  - Upper bound:

$$\#\text{gates} \sim \tilde{\alpha}\, t^{1+1/p} \epsilon^{-1/p},$$

    $p = 1,2, \ldots$ — Trotter order, $\tilde{\alpha}$ — commutator norm (😩).
  - Lower bound:

$$\#\text{gates} \sim (\|H\|\, t)^{1+1/p} \epsilon^{-1/p},$$

    $(\|H\|\, t)$ because $e^{-iHt} = e^{-i(H/\alpha)(\alpha t)}$.
  - Starting with $H_{qubit} = \sum^{N} c_\alpha P_\alpha$ implies $\#\text{gates} > O(N)$.

Questions to ask when developing algorithms for complex systems

- How far can the asymptotic dependence on $||H||$, t, ϵ be improved?

- How to improve the constant factors?
  - Bottleneck of many algorithms: *Block Encoding* (*BE*) subroutine.
  - Improve BE or consider alternative approaches.
  - Carefully study dependence on **all** model parameters.

- How do such methods compare to each other?
  - Depends on model, observables, and regime ($||H||$, t, ϵ, **...**).

Block encoding

- The Block Encoding (BE) construction allows one to implement *arbitrary* linear transformations of quantum states:

$$|\psi\rangle \mapsto A|\psi\rangle$$

- The idea is to embed A into a unitary matrix of larger size:

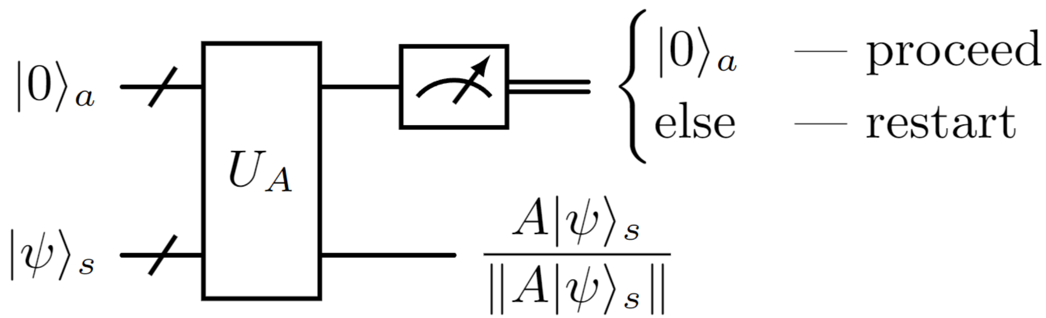
$$U_A = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$$

$$\begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}\begin{pmatrix} \psi \\ 0 \end{pmatrix} = \frac{1}{\alpha}\begin{pmatrix} A\psi \\ * \end{pmatrix} = \frac{1}{\alpha}\begin{pmatrix} A\psi \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ * \end{pmatrix}$$

- $1/\alpha$ is necessary because entries of U_A have to be < 1.

Block encoding circuit

- In a QC, extra dimensions come from adding more qubits:



- The extra ("ancillary") qubits have to be measured upon applying U_A, and the desired outcome is probabilistic.
- Constructing BEs is art/science, and is highly problem-dependent.

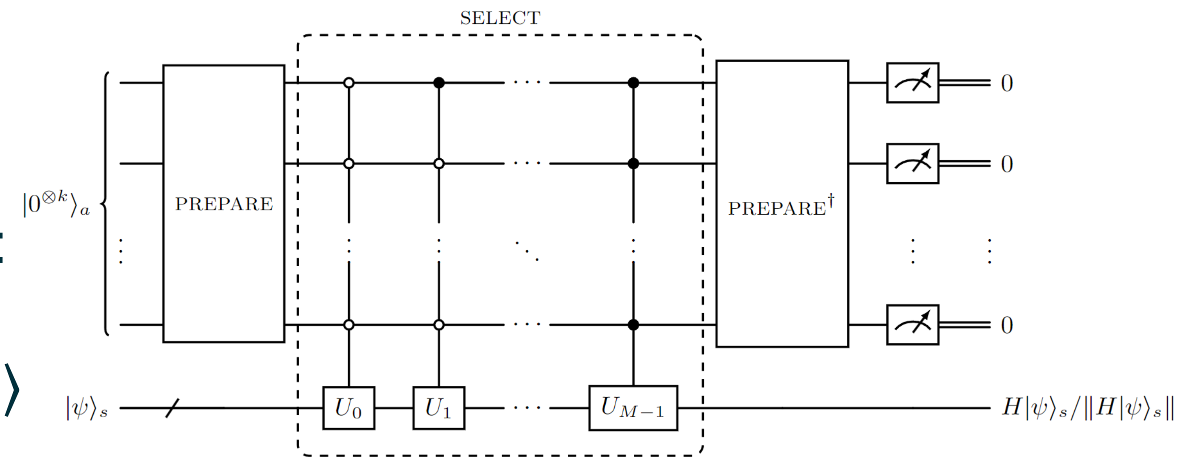
LCU

- Selectively apply U_l in $H = \sum_l \alpha_l U_l$ based on the index state value:

$$\text{SELECT}|l\rangle|\psi\rangle = |l\rangle U_l|\psi\rangle$$

- Coefficients $\vec{\alpha} \equiv \{\alpha_l\}$ are encoded in the index state:

$$\text{PREPARE}\,|0\rangle = \sum_l \sqrt{\frac{\alpha_l}{|||\vec{\alpha}|_1}}\,|l\rangle$$



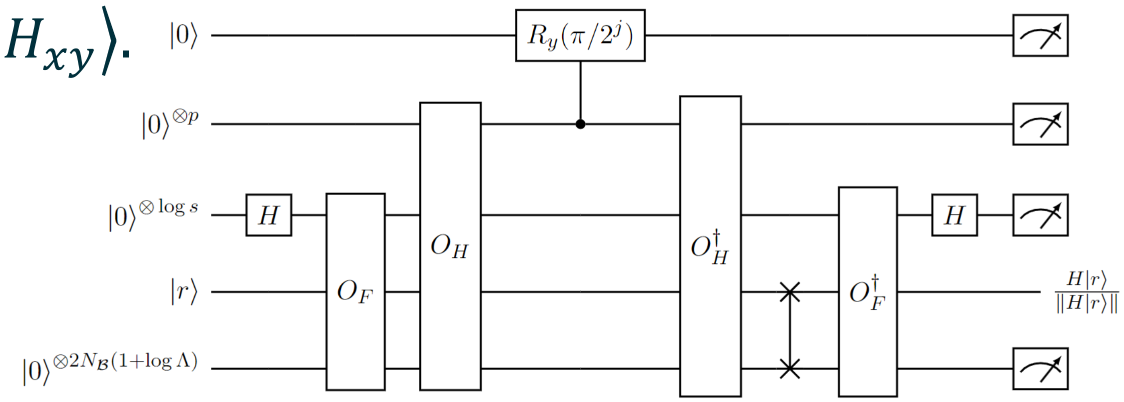
- $(1 \otimes \langle 0|)\ \text{PREPARE}^\dagger \cdot \text{SELECT} \cdot \text{PREPARE}\ (1 \otimes |0\rangle) \sim H$

Sparse Oracle Access

- Sparse oracle access, $O_F \& O_H \mapsto U_H$:

$$O_F|x\rangle|l\rangle = |x\rangle|y_l\rangle\,,$$

$$O_H|x\rangle|y\rangle|z\rangle = |x\rangle|y\rangle\big|z \oplus H_{xy}\big\rangle.$$

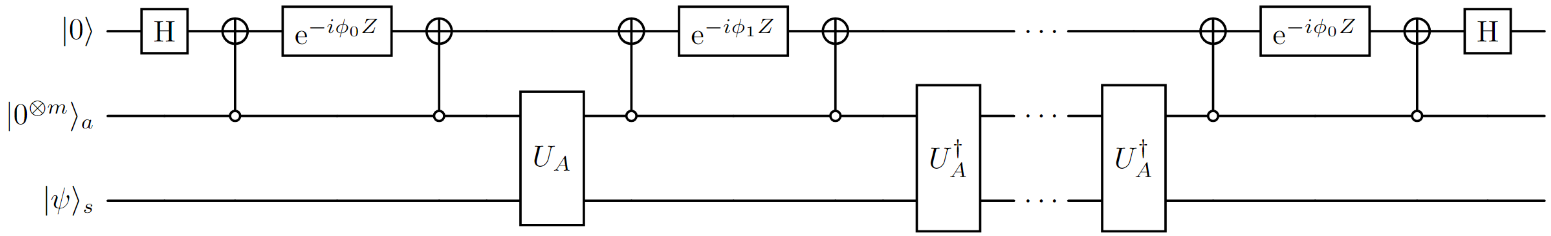


$$H = \begin{matrix} & |y\rangle & & \\ & \begin{pmatrix} \ddots & \cdots & \ddots \\ \vdots & \ddots & \vdots \\ H_{xy} & \cdots & \ddots \end{pmatrix} \\ |x\rangle & & & \end{matrix}$$

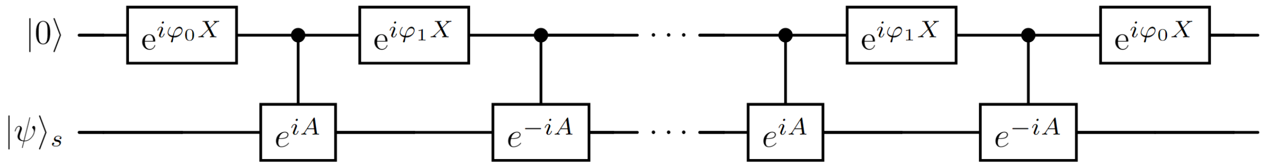
- $(1 \otimes \langle 0|)O_F \cdot O_H \cdot \ldots \cdot O_H^\dagger \cdot \ldots \cdot O_F^\dagger(1 \otimes |0\rangle) \sim H$

QSP-based algorithms: QSVT, QETU, etc.

- QSVT (1806.01838): **Implements $U_{P(A)}$ provided access to U_A**



- QETU (2204.05955): **Implements $U_{P(\cos(A/2))}$ provided access to e^{iA}**



- $P(x)$ approximates the desired function, e.g., e^{ix}, e^{-x}, $\operatorname{sgn} x$, etc.
- The angles of rotation gates are determined by the coefficients of $P(x)$.

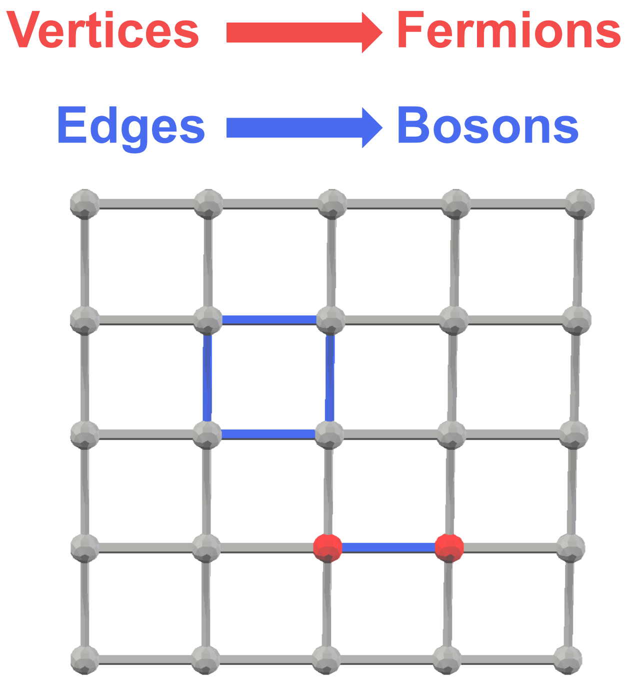
Part 3. Applications

3.1 Near-optimal simulation of Kogut-Susskind LGT

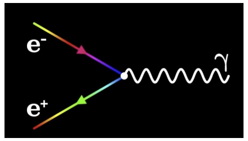
2405.10416

Nearly-optimal simulation of a local Lattice Gauge Theory

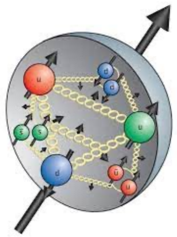
- Kogut-Susskind formulation of LGT:



- Can be used to simulate

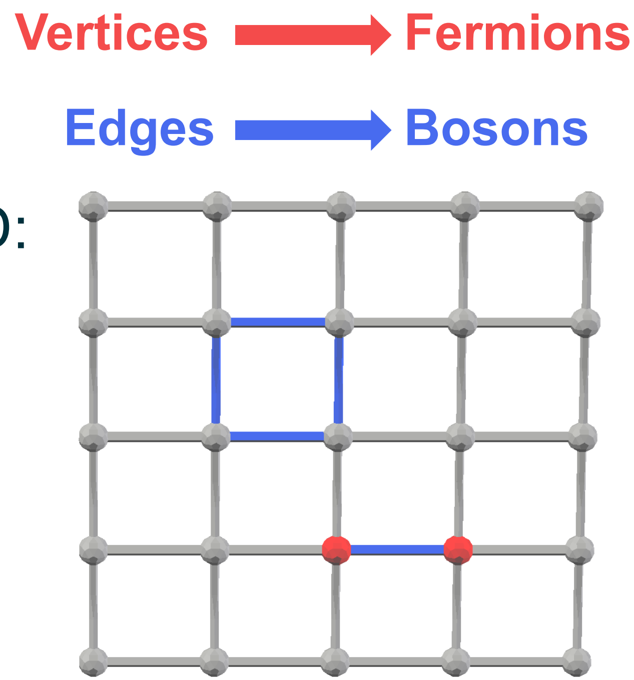


  - $\mathrm{U}(\mathbf{1})$ gauge theory
    "Quantum Electrodynamics":
    Electromagnetic interactions
    between electrons and photons.



  - $\mathrm{SU}(\boldsymbol{n_c})$ gauge theory
    "Quantum Chromodynamics":
    Strong interactions between quarks and gluons

Nearly-optimal simulation of a local Lattice Gauge Theory



- Kogut-Susskind formulation of SU(2) in 2+1D:

$$H = g_M H_M + g_{GM} H_{GM} + g_E H_E + g_B H_B$$

$$H_M = \sum_x (-1)^x \psi^\dagger(x) \cdot \psi(x)$$

$$H_{GM} = \sum_x \sum_{i=1}^{d} \psi^\dagger(x) \cdot U(x, n_i) \cdot \psi(x + n_i) + \text{h.c.}$$

$$H_E = \sum_x \sum_{i=1}^{d} E(x, n_i)^2$$

$$E(\vec{n}, \hat{l})|k\rangle = k|k\rangle \qquad U(\vec{n}, \hat{l})|k\rangle = |k-1\rangle$$

$$H_B = \sum_x \sum_{\substack{i,j=1 \\ i \neq j}}^{d} \text{tr}\left[U(x, n_i) \cdot U(x + n_i, n_j) \cdot U(x + n_j, n_i)^\dagger \cdot U(x, n_j)^\dagger\right] + \text{h.c.}$$



Nearly-optimal simulation of a local Lattice Gauge Theory

- Prior work: Algorithms for simulating LGTs using Trotterization:
  - Kan&Nam (2107.12769);
  - Davoudi et. al. (2212.14030).

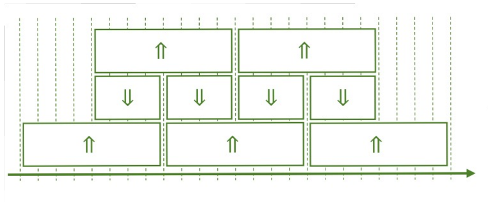
- Our work: Algorithms for near-optimal simulation of LGT, taking advantage of the Hamiltonian structure.

Nearly-optimal simulation of a local Lattice Gauge Theory

- General plan: Tong et. al. (2110.06942):

$$\tilde{O}(N^2; T^3; \log 1/\epsilon) \rightarrow \tilde{O}(N; T; \log 1/\epsilon)$$

via HHKL (1801.03922) and Interaction picture (1805.00675)

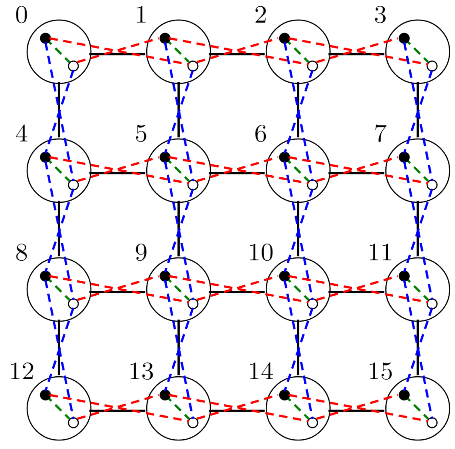


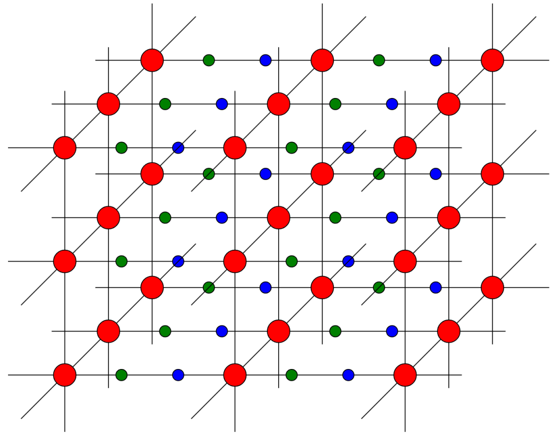
$$H_I^{\mathcal{B}}(t) = e^{itH_E^{\mathcal{B}}}(H_M^{\mathcal{B}} + H_{GM}^{\mathcal{B}} + H_B^{\mathcal{B}})e^{-itH_E^{\mathcal{B}}}$$

- Block encoding is required for all the terms.
- Devil in the details: Work out qubit mappings; Implement BEs using Linear Combination of Unitaries and Sparse Oracle approaches.

Local fermionic encodings

- VC (cond-mat/0508353) for U(1) in 2 dim: One auxiliary fermion per lattice site.
- GSE (1810.05274) for SU(3) in d dim: $d + n_c - 1$ qubits for a site of degree $2d$.
- For U(1) in 2+1D they the costs are same, GSE wins in more complex scenarios.

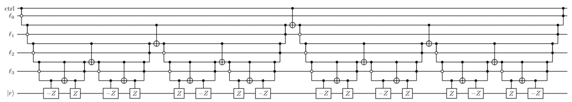




LCU

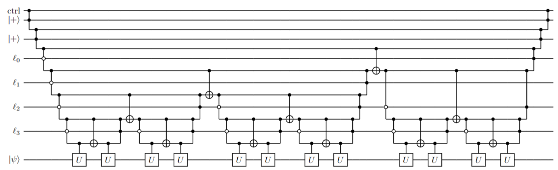
$$H_M = \sum_{\vec{n}} \sum_{a} (-1)^{\vec{n}} \psi_a^\dagger(\vec{n}) \psi_a(\vec{n})$$

- SELECT_{H_M}



$$H_{GM} = \sum_{\vec{n},\hat{l}} \sum_{a,b} \psi_a^\dagger(\vec{n}) U_{ab}(\vec{n},\hat{l}) \psi_b(\vec{n}+\hat{l}) + \text{h.c.}$$

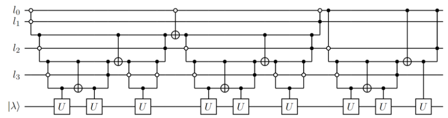
- $\mathrm{SELECT}_{H_{GM}}$



$$H_B = \sum_{\square} \mathrm{Tr}[P_\square + P_\square^\dagger]$$

$$\mathrm{Tr}[P_\square] = \sum_{abce} U_{ab}(\vec{n},\hat{\imath}) U_{bc}(\vec{n}+\hat{\imath},\hat{\jmath}) U_{ce}^\dagger(\vec{n}+\hat{\jmath},\hat{\imath}) U_{ea}^\dagger(\vec{n},\hat{\jmath})$$

- $\mathrm{SELECT}_{H_B} \approx O_{F_B}$



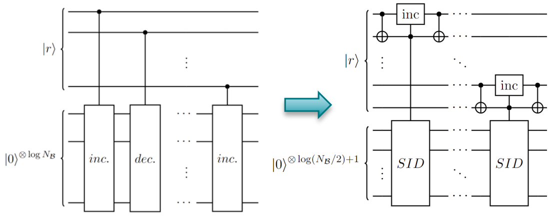
- Unlike in quantum chemistry, in lattice systems

$$\text{Cost SELECT} \gg \text{Cost PREPARE.}$$

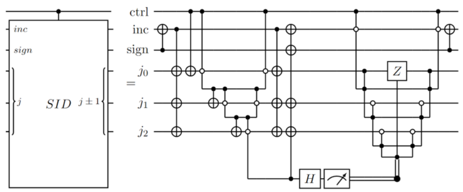
Sparse oracles

$$H_M = \sum_{\vec{n}} \sum_{a} (-1)^{\vec{n}} \psi_a^\dagger(\vec{n}) \psi_a(\vec{n})$$

- O_{F_M} <trivial>
- O_{H_M}

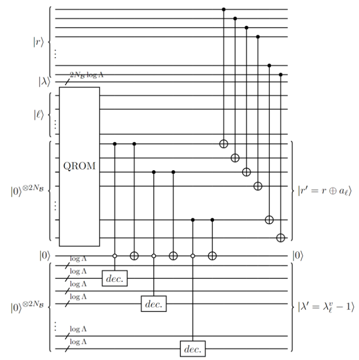


where

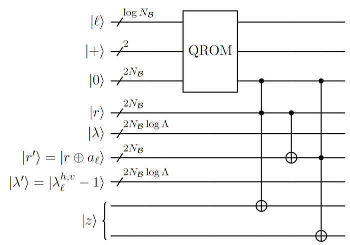


$$H_{GM} = \sum_{\vec{n},\hat{l}} \sum_{a,b} \psi_a^\dagger(\vec{n}) U_{ab}(\vec{n},\hat{l}) \psi_b(\vec{n}+\hat{l}) + \text{h.c.}$$

- $O_{F_{GM}}$



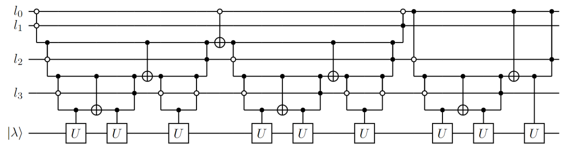
- $O_{H_{GM}}$



$$H_B = \sum_{\square} \mathrm{Tr}[P_\square + P_\square^\dagger]$$

$$\mathrm{Tr}[P_\square] = \sum_{abce} U_{ab}(\vec{n},\hat{\imath}) U_{bc}(\vec{n}+\hat{\imath},\hat{\jmath}) U_{ce}^\dagger(\vec{n}+\hat{\jmath},\hat{\imath}) U_{ea}^\dagger(\vec{n},\hat{\jmath})$$

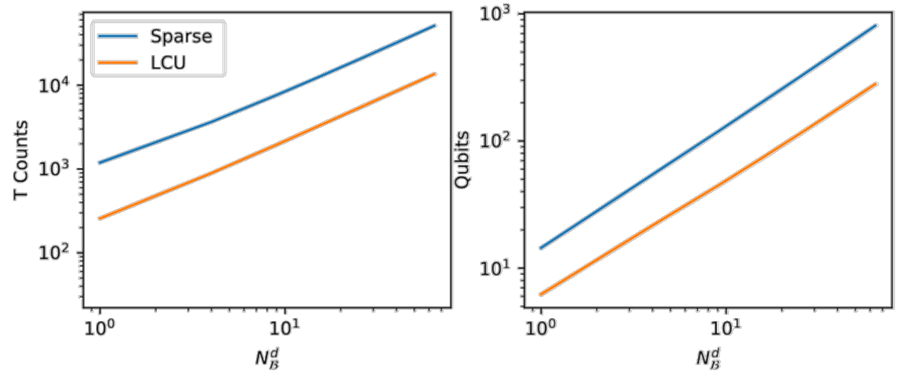
- O_{F_B}



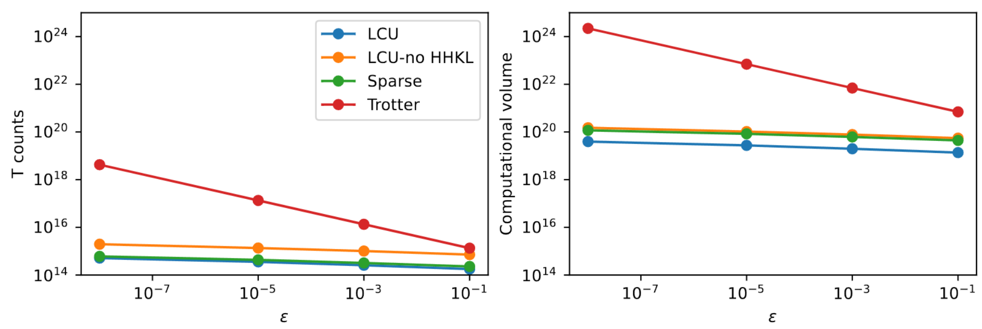
- O_{H_B} <trivial>

T-gate counts for $\mathrm{U}(1)$ LGT (Quantum Electrodynamics)

- T-counts for various terms in the Hamiltonian, $T = 1; \Lambda_0 = 5; \epsilon = 10^{-3}$:

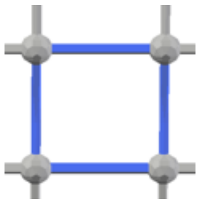


- Total T-counts and qubit counts:



Improvements for $\mathrm{SU}(n_c = 2)$ and $\mathrm{SU}(n_c = 3)$ (Strong forces)

- Better scaling than PF in *time & error*; better scaling in *size* — for large lattices.
- It outperforms conventional techniques for non-Abelian gauge theories:
  - Trotterization: $O\left(2^{8(n_c^2-1)}\right)$
  - BE-based simulation: $O(n_c^4)$



$$U_{ab}|j, m^L, m^R\rangle = \sum_{J=|j-1/2|}^{j+1/2} \sqrt{\frac{2j+1}{2J+1}} \langle J, M_L | j, m^L; 1/2, a'\rangle \langle J, M_R | j, m^R; 1/2, b'\rangle$$
$$\times \left| J, M_L = m^L + a', M_R = m^R + b' \right\rangle .$$

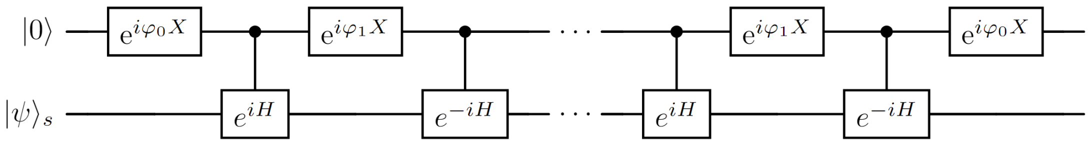
ϵ	N	a	T_{Trotter}	$T_{\text{Qubit.}}$	Q_{Trotter}	$Q_{\text{Qubit.}}$	Improvement
10^{-3}	10^3	10^0	1.2×10^{36}	1.7×10^{26}	10^{11}	2.4×10^{10}	2.9×10^{10}
		10^{-1}	3.9×10^{37}	2.5×10^{26}	10^{11}	2.4×10^{10}	6.0×10^{11}
		10^{-2}	1.2×10^{39}	1.1×10^{27}	10^{11}	2.4×10^{10}	4.4×10^{12}
	10^2	10^0	3.9×10^{31}	1.2×10^{20}	10^8	2.4×10^7	1.2×10^{12}
		10^{-1}	1.2×10^{33}	1.8×10^{20}	10^8	2.4×10^7	2.6×10^{13}
		10^{-2}	3.9×10^{34}	8.2×10^{20}	10^8	2.4×10^7	1.9×10^{14}
10^{-1}	10^3	10^0	1.2×10^{35}	7.8×10^{25}	10^{11}	2.4×10^{10}	6.4×10^9
		10^{-1}	3.9×10^{36}	1.1×10^{26}	10^{11}	2.4×10^{10}	1.3×10^{11}
		10^{-2}	1.2×10^{39}	5.0×10^{26}	10^{11}	2.4×10^{10}	9.9×10^{12}
	10^2	10^0	3.9×10^{30}	5.5×10^{19}	10^8	2.4×10^7	2.8×10^{11}
		10^{-1}	1.2×10^{32}	8.3×10^{19}	10^8	2.4×10^7	5.9×10^{12}
		10^{-2}	3.9×10^{33}	3.6×10^{20}	10^8	2.4×10^7	4.3×10^{13}

ϵ	N	a	T_{Trotter}	$T_{\text{Qubit.}}$	Q_{Trotter}	$Q_{\text{Qubit.}}$	Improvement
10^{-3}	10^3	10^0	1.0×10^{50}	2.0×10^{27}	2.6×10^{11}	6.0×10^{10}	2.1×10^{23}
		10^{-1}	3.2×10^{51}	2.9×10^{27}	2.6×10^{11}	6.0×10^{10}	4.6×10^{24}
		10^{-2}	1.0×10^{53}	1.2×10^{28}	2.6×10^{11}	6.0×10^{10}	3.5×10^{25}
	10^2	10^0	3.2×10^{45}	1.6×10^{21}	2.6×10^8	6.0×10^7	8.2×10^{24}
		10^{-1}	1.0×10^{47}	2.4×10^{21}	2.6×10^8	6.0×10^7	1.7×10^{26}
		10^{-2}	3.2×10^{48}	1.0×10^{22}	2.6×10^8	6.0×10^7	1.3×10^{27}
10^{-1}	10^3	10^0	1.0×10^{49}	1.7×10^{27}	2.6×10^{11}	6.0×10^{10}	2.5×10^{22}
		10^{-1}	3.2×10^{50}	2.5×10^{27}	2.6×10^{11}	6.0×10^{10}	5.4×10^{23}
		10^{-2}	1.0×10^{52}	1.0×10^{28}	2.6×10^{11}	6.0×10^{10}	4.1×10^{24}
	10^2	10^0	3.2×10^{44}	1.3×10^{21}	2.6×10^8	6.0×10^7	1.0×10^{24}
		10^{-1}	1.0×10^{46}	1.9×10^{21}	2.6×10^8	6.0×10^7	2.1×10^{25}
		10^{-2}	3.2×10^{47}	8.4×10^{21}	2.6×10^8	6.0×10^7	1.6×10^{26}

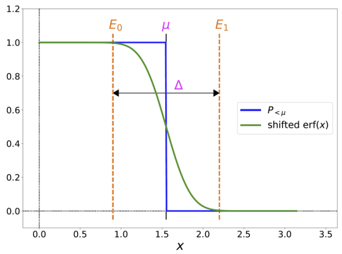
3.2 QETU for ground state preparation and beyond
2310.13757

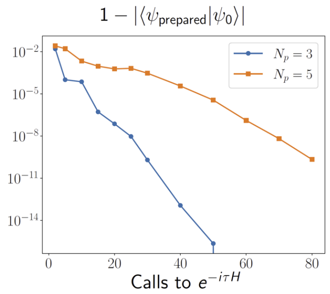
QETU: Ground state preparation for LGT
2310.13757

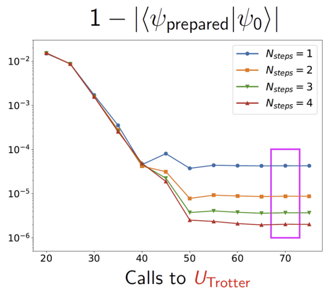
- QETU algorithm improvements and GS preparation in U(1) LGT:

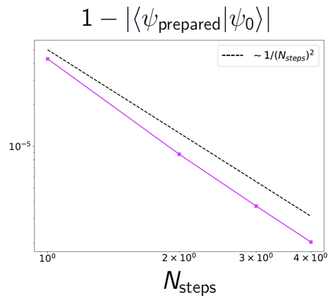


$P_{<\mu} \approx |\psi_0\rangle\langle\psi_0|$
$P_{<\mu}|\psi_{\text{init}}\rangle \approx |\psi_0\rangle$

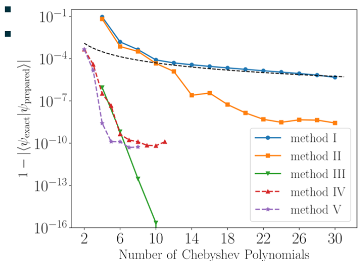


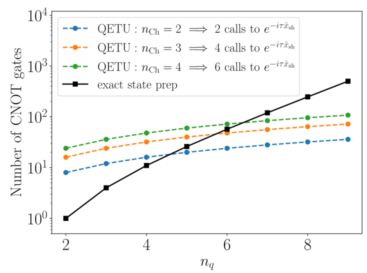






- Preparing Gaussian states with QETU:





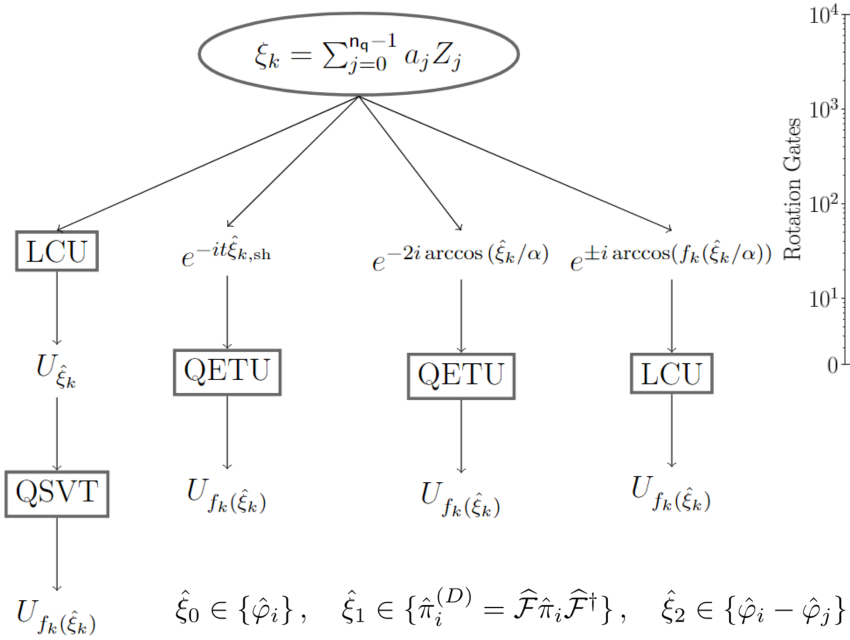
- What else is QETU good for?

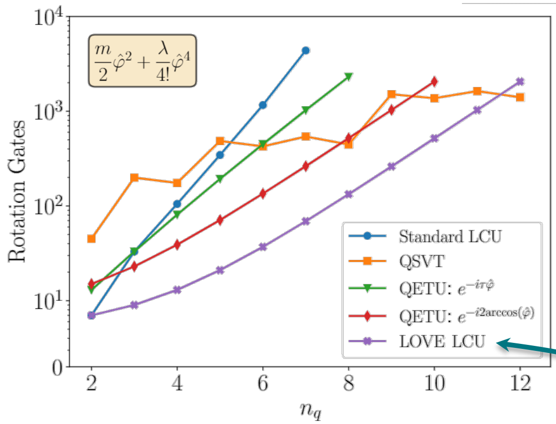
3.3 Block encodings for ϕ^4

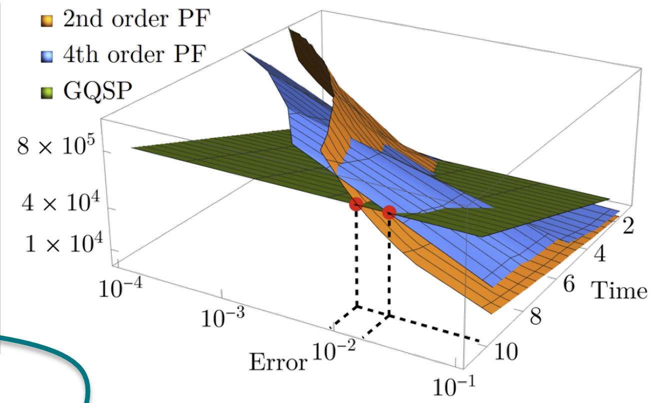
2312.11637, 2408.16824

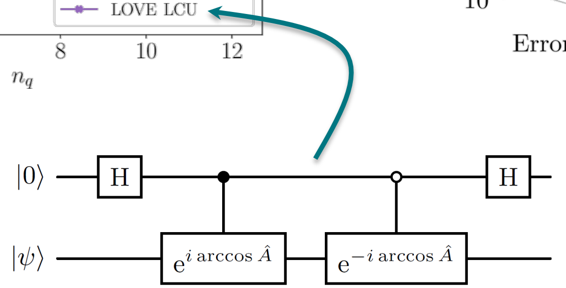
Block encodings for lattice ϕ^4 theory
[2312.11637](2312.11637), [2408.16824](2408.16824)

- Building blocks for local terms; using $\phi \sim \sum_{m=0}^{n_q-1} 2^m Z_m$ digitization.









Lessons

- Algorithms for simulating complex systems may rely on principles different from those upon which near-term simulations are based.

- Near-optimal algorithms may compete with those based on Product Formulae (Trotterization) in physically interesting regimes.

- This requires using state-of-the-art techniques.
  (qubit mappings, block encoding, interaction picture, HHKL, etc.)

Thank You!

#backup

Nearly-optimal simulation of a local Lattice Gauge Theory

- Tong et. al. 2110.06942:
  - Naïve approach (BE entire H + QSP) leads to $\tilde{O}(N^2; T^3; \log 1/\epsilon)$.
  - $\tilde{O}(N^2)$ which comes from:

$$(\text{BE cost})\times(||\text{H}|| \text{ in the query cost}) = \tilde{O}(N^2).$$

  - $\tilde{O}(T^3)$ which comes from

$$||\text{H}||\times T \text{ in the query cost} = \tilde{O}(T^3)$$

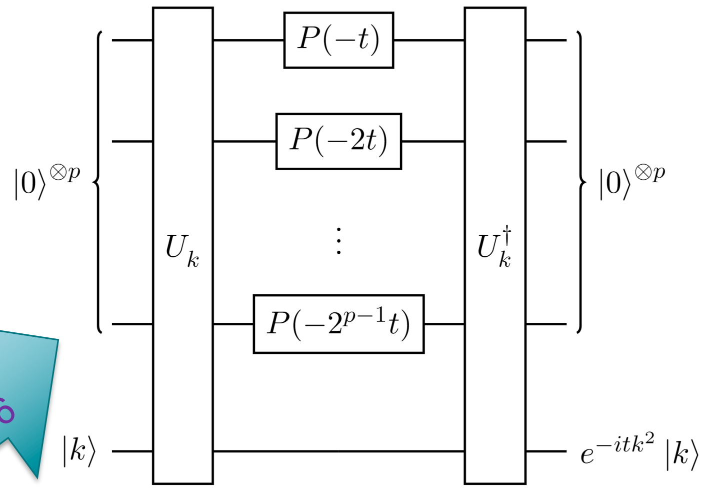
in the $H_E = \sum_{\vec{n},\hat{l}} [E_b(\vec{n},\hat{l})]^2$ term, since $||H_E|| \sim \Lambda^2 \sim T^2$:

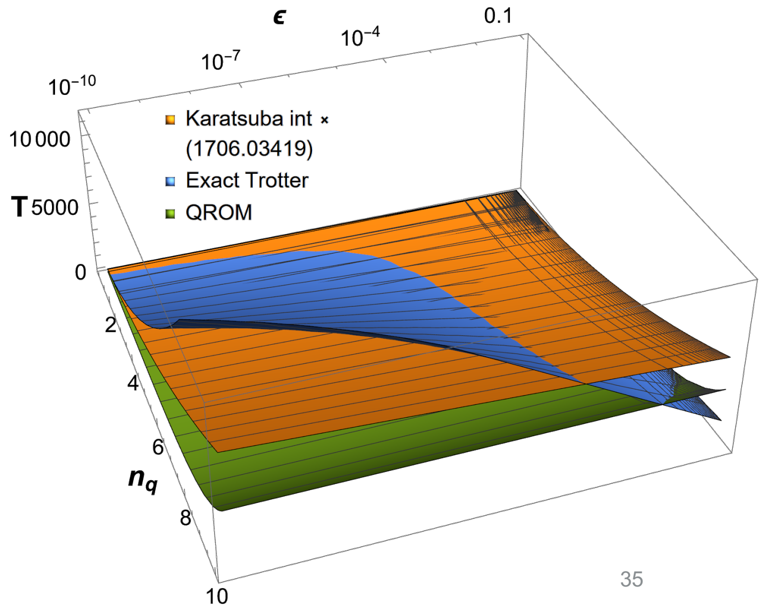
$$\Lambda = \Lambda_0 + O\big((\chi T + 1)\text{polylog}(N\epsilon^{-1})\big).$$

Fast-forwarding $H_E = \sum_{\vec{n},\hat{l}} [E_b(\vec{n},\hat{l})]^2$

- "Exact Trotter" implementation of $e^{i\tau k^2}$.
  - $\widehat{\boldsymbol{k}} \sim \sum_{j=0}^{n_q-1} \boldsymbol{Z_j}$,
    $e^{i\tau k^2} \sim \prod_j e^{Z_j} \prod_{jl} e^{Z_j Z_l}$,
    $\mathrm{T} = O\left(n_q^2 \log(1/\epsilon)\right)$.
- $U_k|k\rangle|z\rangle = |k\rangle|z \oplus k^2\rangle$ + Phase kickback.
  - U_k using arithmetic ([1706.03419](1706.03419)),
    $\mathrm{T} = O\left(n_q^2 + n_q \log 1/\epsilon\right)$.
  - U_k using QROM (i.e., LCU), $\mathrm{T} = O(2^{n_q})$.

Mike&Ike, Problem 4.1; Childs Thesis, Rule 1.6



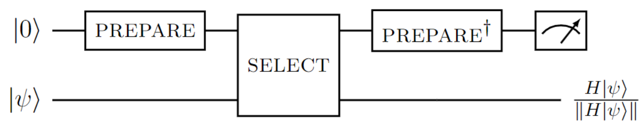


LCU vs Sparse: Takeaways from U(1)

- **$\mathrm{SELECT} \approx \boldsymbol{O_F}$ (applying $\boldsymbol{H}$)** and $\mathrm{PREPARE} \approx O_H$ (H_{xy}).

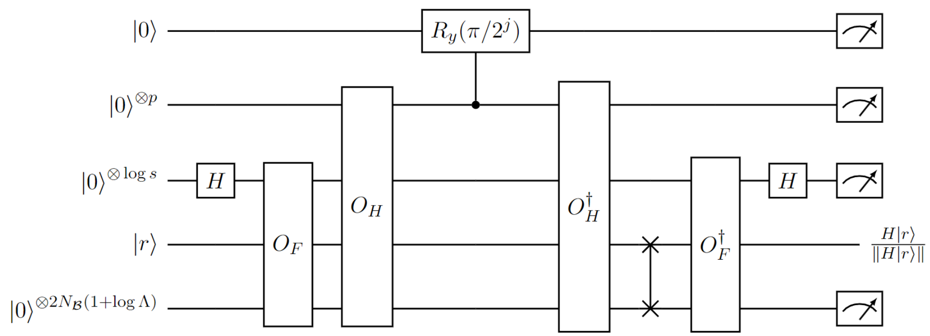
- LCU:

$\mathrm{T}\,[U_H] = \mathrm{T}[\,\mathrm{SELECT}\,] + 2\,\mathrm{T}[\,\mathrm{PREPARE}\,]$.



- Sparse:

$\mathrm{T}[\,U_H\,] = 2\,\mathrm{T}[\,O_F\,] + 2\,\mathrm{T}[\,O_H\,]$.



- LCU is cheaper than Sparse.

Interaction picture via Truncated Dyson Series

- Cost of implementing an elementary building block, the time-dependent block encoding:

$$\mathrm{T}[\text{HAM-T}^{\mathcal{B}}] = \mathrm{T}[e^{-iH_E^{\mathcal{B}}}] \log_2 \frac{16(\alpha + \alpha_E)}{\epsilon} + \mathrm{T}[H^{\mathcal{B}}_{M+GM+B}]$$

- Cost of simulating a single block in HHKL:

$$\mathrm{T}[e^{iH^{\mathcal{B}}}] = \alpha\Big(\mathrm{T}[e^{iH_E^{\mathcal{B}}}] + \lceil -1 + \frac{2\ln(2\alpha/\epsilon)}{\ln\ln(2\alpha/\epsilon) + 1}\rceil \mathrm{T}[\text{HAM-T}^{\mathcal{B}}]\Big)$$

- Cost of the entire simulation:

$$T_{Qubitization} = \frac{N^d T}{N^d_{\mathcal{B}}} \mathrm{T}[e^{iH^{\mathcal{B}}}]$$

Interaction picture via Truncated Dyson Series

- Cost of implementing an elementary building block, the time-dependent block encoding:

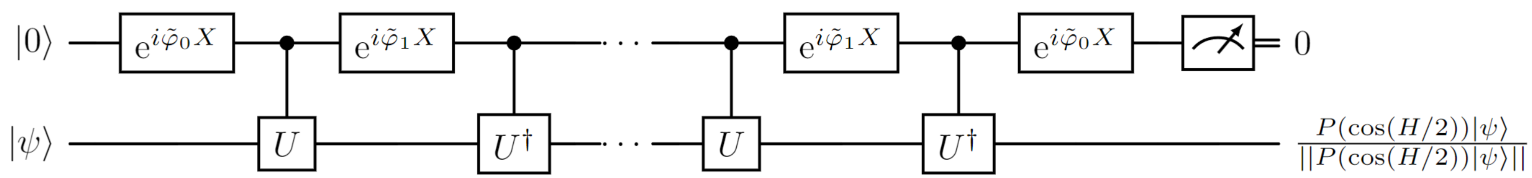
$$\mathrm{T}[\text{HAM-T}^{\mathcal{B}}] = \mathrm{T}[e^{-iH_E^{\mathcal{B}}}]\log_2\frac{16(\alpha+\alpha_E)}{\epsilon} + \mathrm{T}[H^{\mathcal{B}}_{M+GM+B}]$$

- Which terms in the Hamiltonian determine the simulation cost?

<table>
<tr><th></th><th>$T[e^{-iH_E^B}]$</th><th>$\log_2\frac{16(\alpha+\alpha_E)}{\epsilon}T[e^{-iH_E^B}]$</th><th>$T[H^B_{M+GM+B}]$</th></tr>
<tr><td>$U(1)$</td><td>|</td><td>|||||||||||||</td><td>|||</td></tr>
<tr><td>$SU(2)$</td><td>||</td><td>|||||||||||||||||||||||</td><td>|||||||</td></tr>
<tr><td>$SU(3)$</td><td>|||</td><td>||||||||||||||||||||||||||||||</td><td>|||||||||||||||||||||||||||||||||||||||||||||||||</td></tr>
</table>

QETU: a compromise between Nearly-optimal & Trotter

- Since BEs are so expensive, can one use alternative input models?
- What if we sacrifice *some* asymptotic properties?
- e^{-iHt} is a 0-ancilla BE of itself and $\approx e^{-iHt}$ for $t \ll 1$ is cheap.
- Dong et. al. (2204.05955): If $U_H \rightarrow e^{-iHt}$, QSVT turns into "QETU":



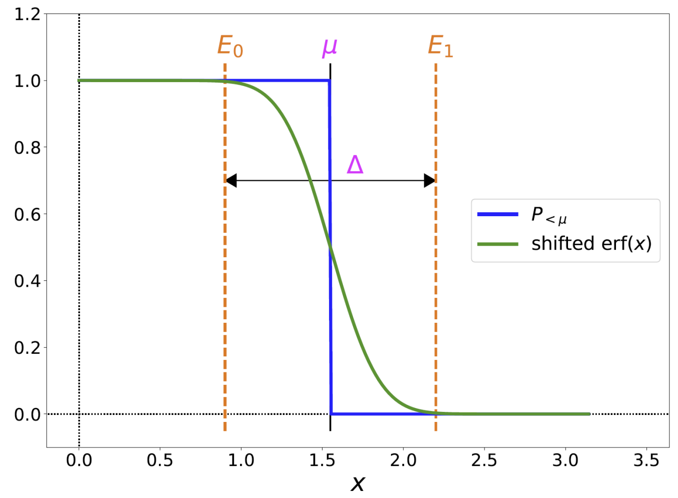
- While the QET circuit was implementing $P(H)$, the QETU circuit implements $P(\cos H/2)^*$.

Ground State Preparation via Projection (Filtering)
Lin & Tong ([2002.12508](2002.12508))

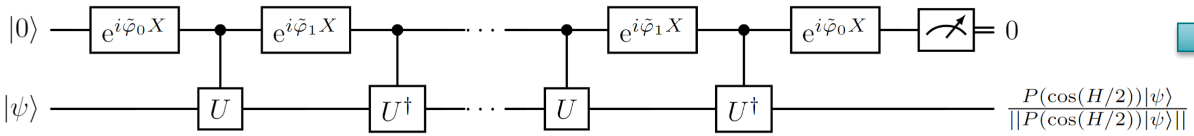
- Consider $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$ and assume access to:
  - Initial state $|\psi_{\text{init}}\rangle$; $|\langle\psi_{\text{init}}|\psi_0\rangle| = \gamma > 0$.
  - GS projector $\mathrm{P}_{<\mu} = |\psi_0\rangle\langle\psi_0|$,
- Then $\mathrm{P}_{<\mu}|\psi_{\text{init}}\rangle = |\psi_0\rangle$.
- Approximate implementation of $\mathrm{P}_{<\mu}$:

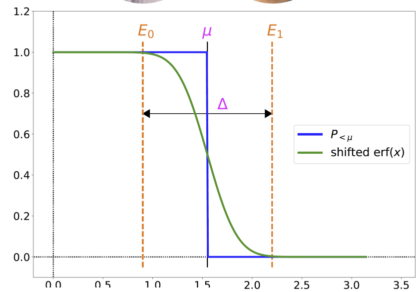
$$f(x) \approx \text{sign}(x) .$$

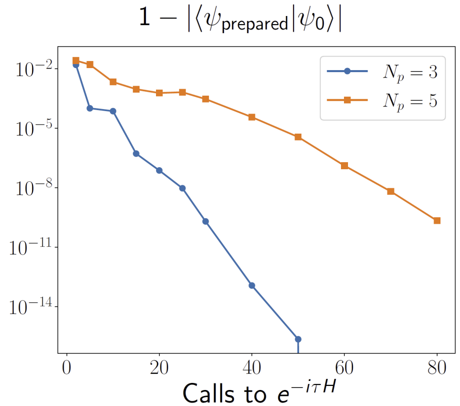
- Query complexity: $\tilde{O}(\gamma^{-1}\Delta^{-1}\log 1/(\gamma\epsilon))$ calls to U_H, where $\Delta = \lambda_1 - \lambda_0$.

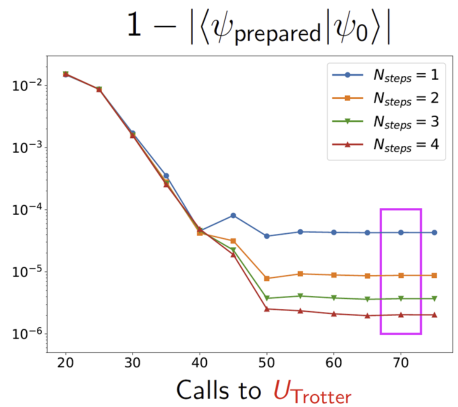


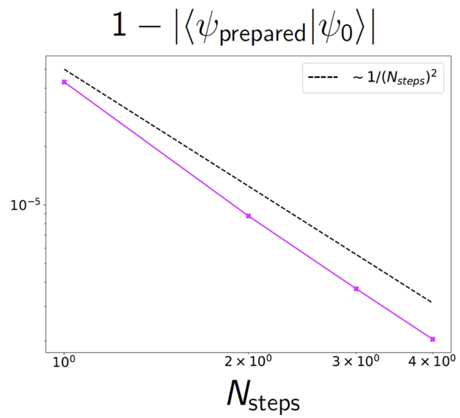
QETU ground state preparation in compact U(1) [2310.13757]











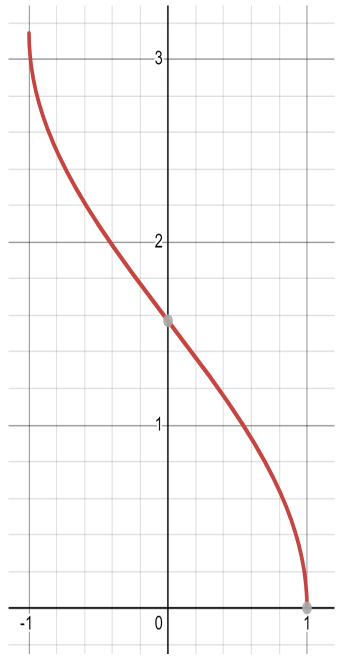
- Exact e^{iH} $\rightarrow$ exponential convergence.
- Trotterized e^{iH} $\rightarrow$ error saturates at $\epsilon \sim 1/N_{steps}^2$.

QETU: what else is it good for?

- Can QETU implement arbitrary functions of $A = A^\dagger$?
- Since QETU implements $P\left(\cos\left(\frac{x}{2}\right)\right)$, for a given $f(x)$, the polynomial $P(x)$ should approximate

$$P(x) \approx f(2\arccos(x)).$$

- But $\arccos(x)$ is not-so-nice at the interval ends: Its Chebyshev expansion has polynomial convergence. (vs. exp convergence for decent functions.)
- If $P\left(\cos\left(\frac{x}{2}\right)\right)$ doesn't have definite parity, LCU is required.



QETU for Gaussians

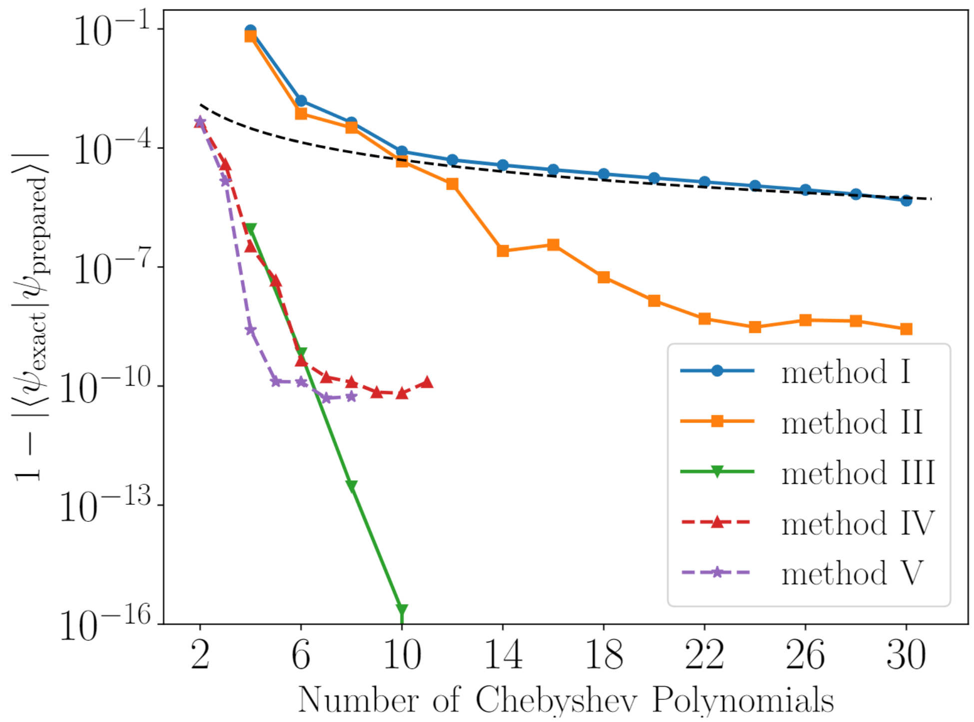
I. $\tau = 1$ and $\eta = 0$.

II. $\tau = 1$ and optimize η.

III. Optimize τ and η.

IV. $\tau = 2$, even $F(x)$, optimize η. (No LCU is required.)

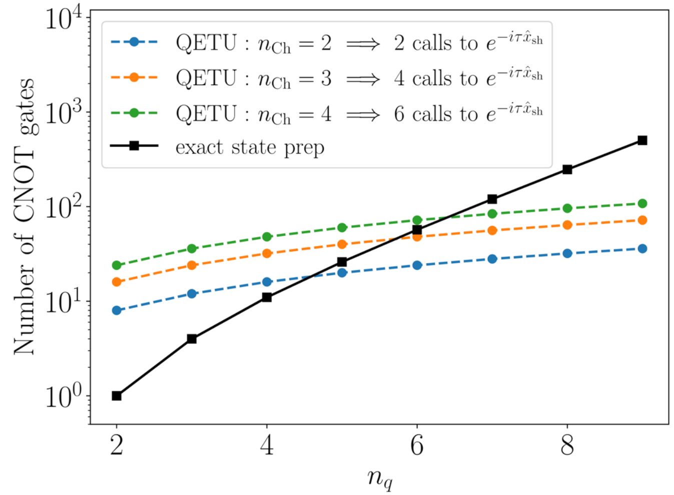
V. Same as IV, but requiring $P(x) \approx \sum_{k=0}^{d/2} c_{2k} T_{2k}(x)$ only at $x = x_j$.

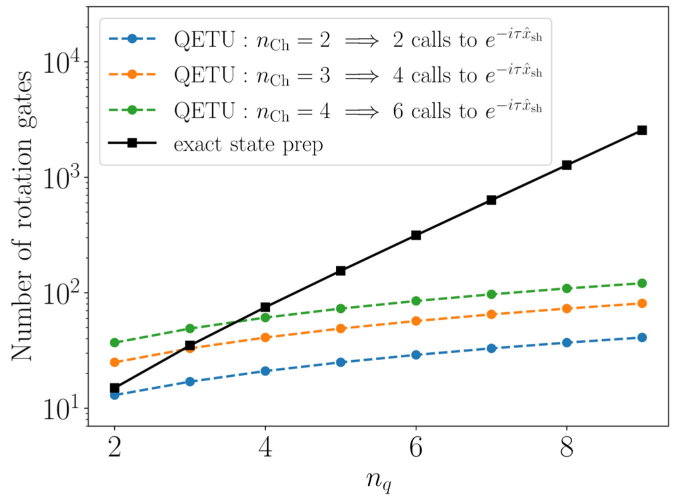


$n_q = 5$

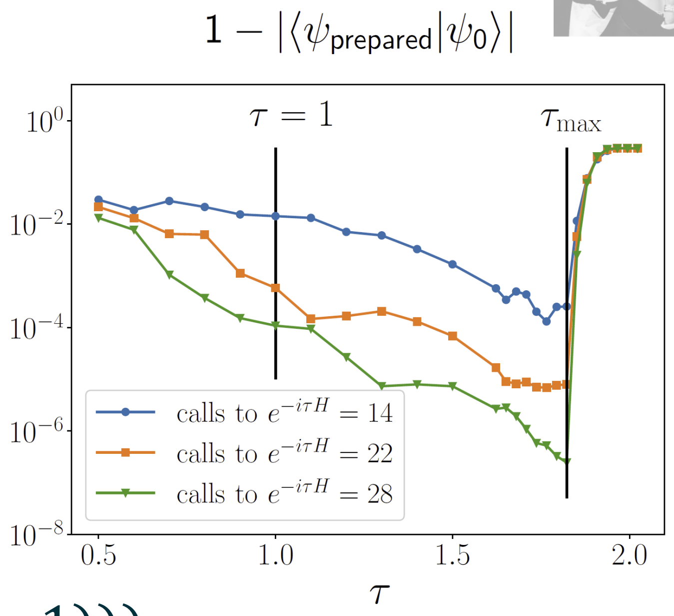
QETU for Gaussians

- Best known results (2109.10918):
  - For ≤ 16 qubits:
    Use direct state preparation.
  - For > 16 qubits:
    Use Kitaev-Webb.
- Improved QETU: wins over direct state preparation for $n_q > 2 - 5$ qubits for $\epsilon \sim 10^{-3}$.





QETU ground state preparation
Push it to the limit



- The original algorithm uses e^{iH}.
- We generalized this to $e^{i\tau H}$ and derived

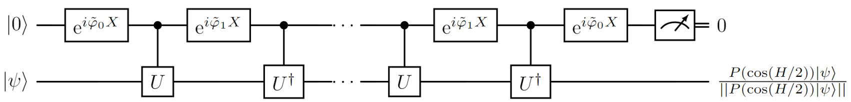
$$\tau_{max} = \frac{2\pi}{\pi - \eta + \mu + \Delta/2}$$

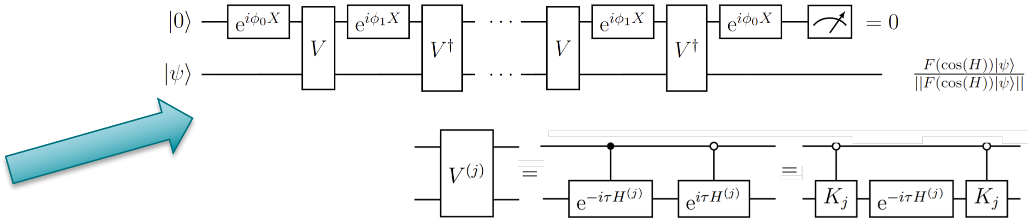
for exact implementation of $e^{i\tau H}$, where $\text{spec}\, H \subset [\eta, \pi - \eta]$ and $\mu = (\lambda_1 - \lambda_0)/2$.

- Precision improves by $\sim O(\exp(d\Delta(\tau_{max} - 1)))$.
- Also generalized the derivation to the case of Trotterization. (Minimized the number of elementary calls $N_{tot} = d \times N_{steps}$).

Bonuses

- Control-free QETU for bosons:





  - Fewer gates;
  - Better precision.

- Showed that, for a fixed ϵ, the $\delta\tau$ in Trotter $U = e^{i\delta\tau H}$ does not scale with the problem size (as one would expect from the $||H||$ growth…):

$$H \frac{\delta\tau}{E_{max}} = \delta\tau \frac{H}{E_{max}}.$$

Scalar and Abelian bosons on the lattice

- Scalar and Abelian fields on lattice — coupled anharmonic oscillators:

$$H = H_\phi + H_\pi \,, \qquad H_\phi \supset \{\phi_i^g ; \cos\phi_i \,; \phi_i\phi_j ; \ldots\} \,, \qquad H_\pi \supset \left\{\pi_i^2 ; \pi_i\pi_j ; \ldots\right\}.$$

- Computation-wise, a good toy model for non-Abelian gauge theories.
- "JLP digitization" = (eigenbases of ϕ and π) & (binary qubit mapping):

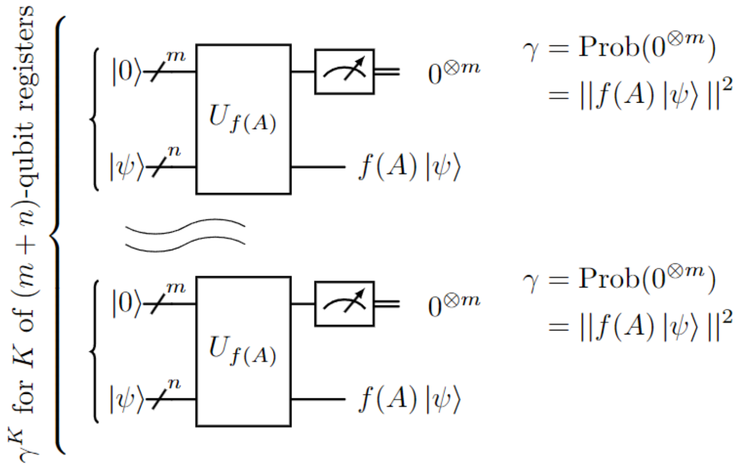
$$\phi \sim \sum_{m=0}^{n_q-1} 2^m \, Z_m \,, \qquad \pi^{(\pi)} \sim \sum_{m=0}^{n_q-1} 2^m \, Z_m \,.$$

- Since $H = H_\phi + \mathrm{FT}^\dagger H_\pi^{(\pi)} \mathrm{FT}$, the product formula is:

$$e^{i\tau H} \approx \left(e^{i\delta\tau H_\phi} \, \mathrm{FT}^\dagger e^{i\delta\tau H_\pi^{(\pi)}} \, \mathrm{FT}\right)^{N_{steps}}, \quad \delta\tau = \tau / N_{steps}.$$

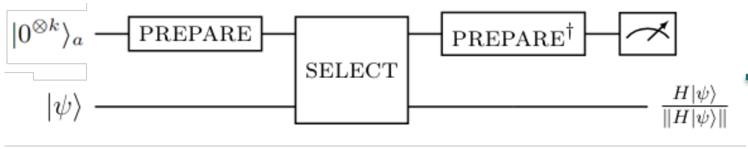
QETU for Gaussians
Disclaimer

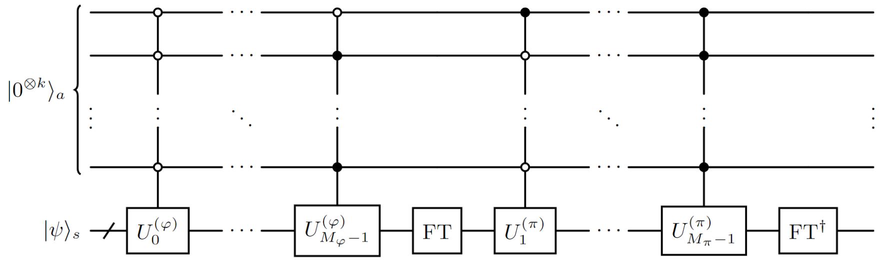
- Described approach is not directly applicable to preparing wavepackets in QFT: Since $f(H)$ is not unitary, the success probability cannot be made $O(1)$.
- But it proves an important point
- And raises the question whether QETU can be utilized in simulating time evolution.



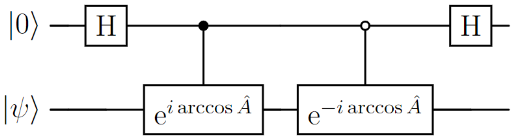
Block encodings for lattice ϕ^4 theory

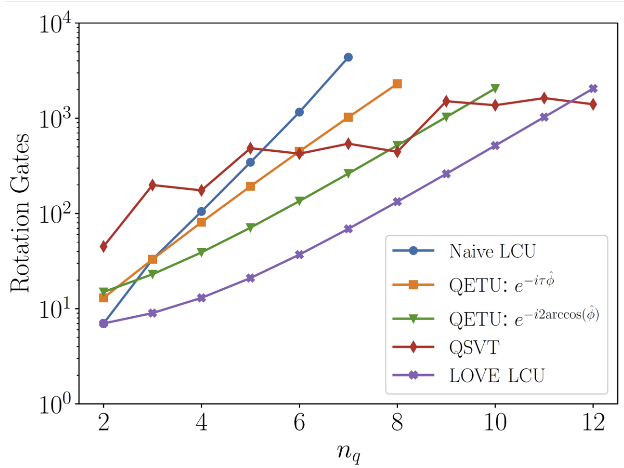
- Improving the naïve LCU using the Quantum Fourier Transform:





- "LOVE LCU"





QETU for Gaussians

- Work in the digitized eigenbasis of $\hat{x}$:

$$\hat{x}|x_j\rangle = x_j|x_j\rangle$$

- QETU circuits calls $U = e^{i\tau\hat{x}_{sh}}$, where $\text{spec}\,\hat{x}_{sh} \subset [\eta, \pi - \eta]$.
- The initial state:

$$\frac{1}{\sqrt{2^{n_q}}}\sum_{j=0}^{2^{n_q}-1}|x_j\rangle$$

- QETU implements Gaussian filter:

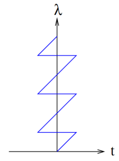
$$f(x) \sim \exp\left(-\frac{1}{\sigma^2}\left(\frac{2}{\tau}\arccos(x) - x_0\right)^2\right)$$

Quantum Simulation: Algorithms Zoo 🦕

Product Formulæ

Time evolution: a) Discretize time $e^{-iTH} = \left[e^{-i\left(\frac{T}{N}\right)H}\right]^N$; b) Each step $e^{A+B+C+\cdots} \approx e^{\frac{A}{2}}e^{\frac{B}{2}}e^{\frac{C}{2}} \ldots e^{\frac{C}{2}}e^{\frac{B}{2}}e^{\frac{A}{2}}$

Adiabatic state preparation: $|\psi\rangle \rightarrow \left[\, e^{-it(d\lambda/dt\ t)H}\ e^{+it(\lambda_1)H}\ e^{-it(d\lambda/dt\ t)H}\ e^{+it(\lambda_2)H} \ldots\right]|\psi\rangle$



Near-term

- **Variational** #vqe, #adapt-vqe, #qaoa
  - $E_{min} = \min_{\vec{\theta}} \langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle$, *where* $|\psi(\vec{\theta})\rangle = U(\vec{\theta})$.
- **Subspace** #krylov, #generalized-subspace
  - $\tilde{H}\vec{c} = \lambda\tilde{S}\vec{c}_{min}$, *where* $\tilde{H}_{ij} = \langle\psi_i|H|\psi_j\rangle$ and $\tilde{S}_{ij} = \langle\psi_i|\psi_j\rangle$.

Fault-tolerant #nearly-optimal

This talk

- **Block Encoding** #qsp, #qsvt, #gqsp, #dyson
  - *Given the ability to implement* $|\psi\rangle \mapsto H|\psi\rangle$, *construct* $|\psi\rangle \mapsto f(H)|\psi\rangle$.
- **Time Evolution Access** #qetu
  - *Given the ability to implement* $U = e^{-i\tau H}$, *construct* $|\psi\rangle \mapsto f(H)|\psi\rangle$.
- **ect.** #hhkl

These approaches can be combined:

- Variational real- & imaginary-time evolution.
- (Block encoding OR Time evolution access) to get E_{min} without $|\psi\rangle \mapsto f(H)|\psi\rangle$.
- (Subspace + Block Encoding) or (Subspace + Hamiltonian Time Evolution).