Randomized quantum singular value transformation

Changpeng Shao

Academy of Mathematics and Systems Science, Chinese Academy of Sciences
based on joint work with Shantanav Chakraborty, Soumyabrata Hazra,
Tongyang Li, Xinzhao Wang and Yuxin Zhang
(arXiv:2504.02385, 91 pages)

Quantum Information and Optimization (QIOP) workshop
13-19 April 2025

QSVT

- Quantum singular value transformation (QSVT) is a unifying framework that encapsulates most known quantum algorithms and serves as the foundation for new ones. (e.g. Grover, quantum phase estimation, Hamiltonian simulation, solving linear systems, etc.) [Gilyen-Su-Low-Wiebe, STOC'19]

QSVT

- Quantum singular value transformation (QSVT) is a unifying framework that encapsulates most known quantum algorithms and serves as the foundation for new ones. (e.g. Grover, quantum phase estimation, Hamiltonian simulation, solving linear systems, etc.) [Gilyen-Su-Low-Wiebe, STOC'19]
- It applies polynomial transformations on the singular values of an operator A, provided A is embedded in the top-left block of a unitary, known as block encoding. Namely, (I will assume A is Hermitian and $\|A\| \leq 1$ in this talk)

$$\underbrace{U := \begin{bmatrix} A & * \\ * & * \end{bmatrix}}_{\text{(unitary)}} \xrightarrow[\text{QSVT}]{\text{poly } f(x)} \underbrace{\widetilde{U} = \begin{bmatrix} f(A) & * \\ * & * \end{bmatrix}}_{\text{(unitary)}}$$

QSVT

- Quantum singular value transformation (QSVT) is a unifying framework that encapsulates most known quantum algorithms and serves as the foundation for new ones. (e.g. Grover, quantum phase estimation, Hamiltonian simulation, solving linear systems, etc.) [Gilyen-Su-Low-Wiebe, STOC'19]
- It applies polynomial transformations on the singular values of an operator A, provided A is embedded in the top-left block of a unitary, known as block encoding. Namely, (I will assume A is Hermitian and $\|A\| \leq 1$ in this talk)

$$\underset{\text{(unitary)}}{U := \begin{bmatrix} A & * \\ * & * \end{bmatrix}} \xrightarrow[\text{QSVT}]{\text{poly } f(x)} \underset{\text{(unitary)}}{\widetilde{U} = \begin{bmatrix} f(A) & * \\ * & * \end{bmatrix}}$$

- Usually, the most technical part is constructing U efficiently.

Previous works on QSVT

- The quantum circuit for QSVT: Assume $f(x) \in \mathbb{C}[x]$, degree d, even/odd, and $|f(x)| \le 1$ for all $x \in [-1, 1]$, then there exists $\Phi := (\phi_1, \ldots, \phi_d) \in \mathbb{R}^d$, s.t.

$$\begin{bmatrix} f(A) & * \\ * & * \end{bmatrix} = U_\Phi = \begin{cases} e^{\mathbf{i}\phi_1 Z} U \prod_{j=1}^{(d-1)/2} \left(e^{\mathbf{i}\phi_{2j} Z} U^\dagger e^{\mathbf{i}\phi_{2j+1} Z} U \right), & \text{if } d \text{ is odd,} \\ \prod_{j=1}^{d/2} \left(e^{\mathbf{i}\phi_{2j-1} Z} U^\dagger e^{\mathbf{i}\phi_{2j} Z} U \right), & \text{if } d \text{ is even.} \end{cases}$$

Previous works on QSVT

- The quantum circuit for QSVT: Assume $f(x) \in \mathbb{C}[x]$, degree d, even/odd, and $|f(x)| \leq 1$ for all $x \in [-1, 1]$, then there exists $\Phi := (\phi_1, \ldots, \phi_d) \in \mathbb{R}^d$, s.t.

$$\begin{bmatrix} f(A) & * \\ * & * \end{bmatrix} = U_\Phi = \begin{cases} e^{\mathbf{i}\phi_1 Z} U \prod_{j=1}^{(d-1)/2} \left(e^{\mathbf{i}\phi_{2j} Z} U^\dagger e^{\mathbf{i}\phi_{2j+1} Z} U \right), & \text{if } d \text{ is odd}, \\ \prod_{j=1}^{d/2} \left(e^{\mathbf{i}\phi_{2j-1} Z} U^\dagger e^{\mathbf{i}\phi_{2j} Z} U \right), & \text{if } d \text{ is even}. \end{cases}$$

- Assuming access to $U = e^{\mathbf{i}H}$, then there exists Φ such that

$$U_\Phi = \begin{bmatrix} P(\cos(A)) & * \\ * & * \end{bmatrix}, \quad H = \begin{bmatrix} & A^\dagger \\ A & \end{bmatrix}, \quad f(x) = P(\cos(x)).$$

See [Lloyd et al, arXiv:2104.01410; Dong-Lin-Tong, PRX Quantum 2022]

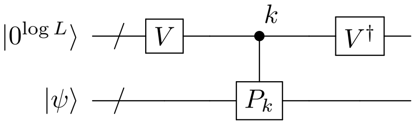
Motivations

- Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ is a unitary (e.g., Pauli) decomposition. Then how to do QSVT?

Motivations

- Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ is a unitary (e.g., Pauli) decomposition. Then how to do QSVT?

- Linear combination of unitaries (LCU): By LCU, we can construct a block encoding of A. Circuit depth is $O(L)$, number of ancilla qubits is $O(\log L)$.



Without loss of generality, we assume $\lambda_k > 0$ and $\sum_k \lambda_k = 1$, then

$$V|0\rangle = \sum_{k=1}^{L} \sqrt{\lambda_k}\,|k\rangle.$$

Motivations

Is that possible to do QSVT using 1 or $O(1)$ ancilla qubits?

Motivations

Is that possible to do QSVT using 1 or $O(1)$ ancilla qubits?

Theorem 1 (Ancilla qubits for LCU)

Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ *be a unitary decomposition of* A*, then* $\ell = \Omega(\log L)$ *ancilla qubits are required to* *exactly* *block-encoding* A*.*

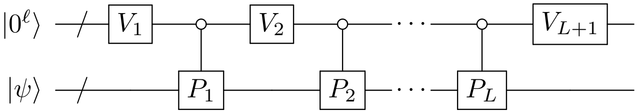
Motivations

Is that possible to do QSVT using 1 or $O(1)$ ancilla qubits?

Theorem 1 (Ancilla qubits for LCU)

Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ be a unitary decomposition of A, then $\ell = \Omega(\log L)$ ancilla qubits are required to exactly block-encoding A.

The above holds for a fairly general circuit model:



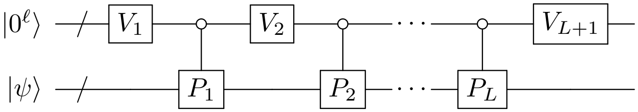
Motivations

Is that possible to do QSVT using 1 or $O(1)$ ancilla qubits?

Theorem 1 (Ancilla qubits for LCU)

Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ be a unitary decomposition of A, then $\ell = \Omega(\log L)$ ancilla qubits are required to exactly block-encoding A.

The above holds for a fairly general circuit model:



- Rmk. The above lower bound might give some hint about the number of ancillas needed for multivariate QSP/QSVT theory!?

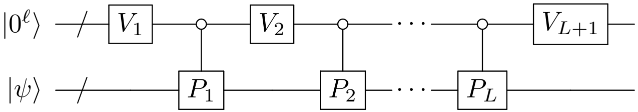
Motivations

Is that possible to do QSVT using 1 or $O(1)$ ancilla qubits?

Theorem 1 (Ancilla qubits for LCU)

Assume $A = \sum_{k=1}^{L} \lambda_k P_k$ be a unitary decomposition of A, then $\ell = \Omega(\log L)$ ancilla qubits are required to exactly block-encoding A.

The above holds for a fairly general circuit model:



- Rmk. The above lower bound might give some hint about the number of ancillas needed for multivariate QSP/QSVT theory!?
- Question: Can we construct an approximate block-encoding of A using 1 or $O(1)$ ancilla qubits? [Gilyén-Vasconcelos obtained such a result for multiplication of block encodings]

Motivations

- So we should do QSVT without block-encoding.

Motivations

- So we should do QSVT without block-encoding.

- Prior work: qDRIFT.

  If $f(x) = e^{\mathbf{i}xt}$, then we can use qDRIFT [Campbell, PRL'19]. Define distribution $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^{L}$. Sample ℓ Paulis $P_{j_1}, \ldots, P_{j_\ell}$, then on average (for some β)

$$e^{\mathbf{i}\beta P_{j_1}} e^{\mathbf{i}\beta P_{j_2}} \cdots e^{\mathbf{i}\beta P_{j_\ell}} \approx e^{\mathbf{i}Ht}$$

  - qDRIFT uses 0 ancilla qubit,
  - $\ell \approx t^2$ is independent of L.
  - It shows advantages over Trotter/QSVT when $t \ll L$.

Our results

Theorem 2

Let $f(x) \in \mathbb{C}[x]$ be a degree-d polynomial such that $|f(x)| \le 1$ for all $x \in [-1, 1]$ and has parity-($d \mod 2$). Assume A is Hermitian with $\|A\| \le 1$ and

$$U = \begin{bmatrix} cI & sA \\ sA^\dagger & -cI \end{bmatrix},$$

where

$$n = \widetilde{\Theta}(d), \quad s = 1/\sqrt{n}, \quad c = \sqrt{1 - 1/n}, \quad \alpha = \widetilde{\Theta}(\sqrt{n}).$$

Then there exists $\Phi \in \mathbb{R}^n$ such that

$$U_\Phi \approx \begin{bmatrix} * & f(A/\alpha) \\ * & * \end{bmatrix} \text{ if } d \text{ is odd, and } U_\Phi \approx \begin{bmatrix} * & * \\ * & f(A/\alpha) \end{bmatrix} \text{ if } d \text{ is even.}$$

Rmk. U and U_Φ may not be unitary. U is unitary iff A is unitary. The above is comparable to the key theorem for QSVT.

Randomized QSVT

- Recall (assume n is even for convenience)

$$U = \begin{bmatrix} cI & sA \\ sA^{\dagger} & -cI \end{bmatrix}, \quad U_{\Phi} = \prod_{j=1}^{n/2} \left(e^{\mathbf{i}\phi_{2j-1} Z} {\color{red}U^{\dagger}} e^{\mathbf{i}\phi_{2j} Z} {\color{red}U} \right).$$

Randomized QSVT

- Recall (assume n is even for convenience)

$$U = \begin{bmatrix} cI & sA \\ sA^\dagger & -cI \end{bmatrix}, \quad U_\Phi = \prod_{j=1}^{n/2} \left(e^{\mathbf{i}\phi_{2j-1} Z} U^\dagger e^{\mathbf{i}\phi_{2j} Z} U \right).$$

- For $A = \sum_{k=1}^{L} \lambda_k P_k$, we consider the distribution $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^{L}$. We independently sample n Paulis $P_{j_1}, \ldots, P_{j_n}$ and replace each U or $U^\dagger$ with

$$U_{j_k} = \begin{bmatrix} cI & sP_{j_k} \\ sP_{j_k}^\dagger & -cI \end{bmatrix}$$

Denote the resulting unitary as $U_\Phi^{(J)}$ with $J = (j_1, \ldots, j_n)$. Then

$$\mathbb{E}_J[U_\Phi^{(J)}] = U_\Phi.$$

Randomized QSVT

- Uses 1 ancilla qubit.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.

  - There exists a polynomial of degree $O(t)$ approximating it.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.
  - There exists a polynomial of degree $O(t)$ approximating it.
  - So $\alpha \approx \sqrt{t}$.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.
  - There exists a polynomial of degree $O(t)$ approximating it.
  - So $\alpha \approx \sqrt{t}$.
  - By our result, we can only implement $e^{\mathbf{i}At/\alpha} = e^{\mathbf{i}A\sqrt{t}}$.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.

  - There exists a polynomial of degree $O(t)$ approximating it.
  - So $\alpha \approx \sqrt{t}$.
  - By our result, we can only implement $e^{\mathbf{i}At/\alpha} = e^{\mathbf{i}A\sqrt{t}}$.
  - To implement $e^{\mathbf{i}At}$, we need to set $t \leftarrow t^2$ in the beginning, i.e., consider $e^{\mathbf{i}xt^2}$.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.
  - There exists a polynomial of degree $O(t)$ approximating it.
  - So $\alpha \approx \sqrt{t}$.
  - By our result, we can only implement $e^{\mathbf{i}At/\alpha} = e^{\mathbf{i}A\sqrt{t}}$.
  - To implement $e^{\mathbf{i}At}$, we need to set $t \leftarrow t^2$ in the beginning, i.e., consider $e^{\mathbf{i}xt^2}$.
  - Similar for $f(x) = x^d, (\kappa x)^{-1}, e^{xt}$.

Randomized QSVT

- Uses 1 ancilla qubit.
- Circuit depth is usually $\widetilde{O}(d^2)$ because $\alpha \approx \sqrt{d}$. [Note that $O(d)$ for the standard QSVT]

  For example, let $f(x) = e^{\mathbf{i}xt}$.

  - There exists a polynomial of degree $O(t)$ approximating it.
  - So $\alpha \approx \sqrt{t}$.
  - By our result, we can only implement $e^{\mathbf{i}At/\alpha} = e^{\mathbf{i}A\sqrt{t}}$.
  - To implement $e^{\mathbf{i}At}$, we need to set $t \leftarrow t^2$ in the beginning, i.e., consider $e^{\mathbf{i}xt^2}$.
  - Similar for $f(x) = x^d, (\kappa x)^{-1}, e^{xt}$.

- This has also been observed in several previous randomized quantum algorithms: qDRIFT, qSWIFT, and randomized LCU.

  [Nakaji et al., PRX Quantum'24; Wang et al. PRX Quantum'24; Chakraborty, Quantum'24]

Lower bounds

Theorem 3 (Sample complexity)

Given $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^{L}$, $\Omega(t^2/\varepsilon^2)$ *samples are required to estimate* $\langle \psi_1 | e^{\mathbf{i}At} | \psi_0 \rangle \pm \varepsilon$ *for any randimized quantum algorithms with only access to* $\mathcal{D}$.

Lower bounds

Theorem 3 (Sample complexity)

Given $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^{L}$, $\Omega(t^2/\varepsilon^2)$ samples are required to estimate $\langle\psi_1|e^{\mathbf{i}At}|\psi_0\rangle \pm \varepsilon$ for any randimized quantum algorithms with only access to $\mathcal{D}$.

A randomized quantum algorithm has the following form:

$$W_{Q_1,\ldots,Q_c} = U_1 Q_1 U_2 Q_2 \cdots U_c Q_c U_{c+1},$$

- $Q_1, \ldots, Q_c$ are randomly and independently generated unitaries, each depends on a random unitary from $\{P_1, \ldots, P_L\}$.
- $U_1, \ldots, U_{c+1}$ are unitaries independent of $\{P_1, \ldots, P_L\}$.
- We also assume $\mathbb{E}[W_{Q_1,\ldots,Q_c}] = e^{\mathbf{i}At}$.

Lower bounds

Theorem 3 (Sample complexity)

Given $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^L$, $\Omega(t^2/\varepsilon^2)$ *samples are required to estimate* $\langle\psi_1|e^{\mathbf{i}At}|\psi_0\rangle \pm \varepsilon$ *for any randimized quantum algorithms with only access to* $\mathcal{D}$.

A randomized quantum algorithm has the following form:

$$W_{Q_1,\ldots,Q_c} = U_1 Q_1 U_2 Q_2 \cdots U_c Q_c U_{c+1},$$

- $Q_1, \ldots, Q_c$ are randomly and independently generated unitaries, each depends on a random unitary from $\{P_1, \ldots, P_L\}$.
- $U_1, \ldots, U_{c+1}$ are unitaries independent of $\{P_1, \ldots, P_L\}$.
- We also assume $\mathbb{E}[W_{Q_1,\ldots,Q_c}] = e^{\mathbf{i}At}$.

By Hoeffding's inequality, $\Omega(1/\varepsilon^2)$ classical repetitions are required, so we have

circuit depth $c = \Omega(t^2)$.

Lower bounds

Theorem 3 (Sample complexity)

Given $\mathcal{D} = \{(\lambda_k, P_k)\}_{k=1}^L$, $\Omega(t^2/\varepsilon^2)$ samples are required to estimate $\langle \psi_1 | e^{\mathbf{i}At} | \psi_0 \rangle \pm \varepsilon$ for any randimized quantum algorithms with only access to $\mathcal{D}$.

A randomized quantum algorithm has the following form:

$$W_{Q_1,\ldots,Q_c} = U_1 Q_1 U_2 Q_2 \cdots U_c Q_c U_{c+1},$$

- $Q_1, \ldots, Q_c$ are randomly and independently generated unitaries, each depends on a random unitary from $\{P_1, \ldots, P_L\}$.
- $U_1, \ldots, U_{c+1}$ are unitaries independent of $\{P_1, \ldots, P_L\}$.
- We also assume $\mathbb{E}[W_{Q_1,\ldots,Q_c}] = e^{\mathbf{i}At}$.

By Hoeffding's inequality, $\Omega(1/\varepsilon^2)$ classical repetitions are required, so we have

circuit depth $c = \Omega(t^2)$.

Question: Can we prove a lower bound of $\Omega(d^2)$ for general polynomials?

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

$$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i} H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_\gamma^{(j)}.$$

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

  $$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_{\gamma}^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

  $$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_{\gamma}^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

- (QSVT with qDRIFT) We have another randomized algorithm for QSVT.

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

  $$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_{\gamma}^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

- (QSVT with qDRIFT) We have another randomized algorithm for QSVT.
  - QSVT with qDRIFT: uses 1 ancilla qubit as long as the input function has a Laurent polynomial approximation.

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

  $$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_\gamma^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

- (QSVT with qDRIFT) We have another randomized algorithm for QSVT.
  - QSVT with qDRIFT: uses 1 ancilla qubit as long as the input function has a Laurent polynomial approximation.
  - The Randomized QSVT: uses 3 ancilla qubits to implement a general real function. But it can implement any bounded function.

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

  $$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_\gamma^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

- (QSVT with qDRIFT) We have another randomized algorithm for QSVT.
  - QSVT with qDRIFT: uses 1 ancilla qubit as long as the input function has a Laurent polynomial approximation.
  - The Randomized QSVT: uses 3 ancilla qubits to implement a general real function. But it can implement any bounded function.

- As applications, we proposed near-optimal quantum algorithms for quantum linear systems and ground state problems without block-encoding.

Other results

- (QSVT with Trotter) We proposed an algorithm for QSVT with 1 ancilla qubit and circuit depth is $\widetilde{O}(d^{1+o(1)})$.

  Indeed, we have an algorithm for more general circuits

$$W = V_0 \prod_{j=1}^{d} e^{\mathbf{i}H^{(j)}} V_j, \quad H^{(j)} = \sum_{\gamma=1}^{\Gamma_j} H_\gamma^{(j)}.$$

  Question: Can we do in circuit depth $O(d)$?

- (QSVT with qDRIFT) We have another randomized algorithm for QSVT.
  - QSVT with qDRIFT: uses 1 ancilla qubit as long as the input function has a Laurent polynomial approximation.
  - The Randomized QSVT: uses 3 ancilla qubits to implement a general real function. But it can implement any bounded function.

- As applications, we proposed near-optimal quantum algorithms for quantum linear systems and ground state problems without block-encoding.

Thank you!