Variational Quantum Linear Solver

Carlos Bravo-Prieto[1,2,3], Ryan LaRose[4], M. Cerezo[1,5], Yiğit Subaşı[6], Lukasz Cincio[1], and Patrick J. Coles[1]

[1] Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA.

[2] Barcelona Supercomputing Center, Barcelona, Spain.

[3] Institut de Ciències del Cosmos, Universitat de Barcelona, Barcelona, Spain.

[4] Department of Computational Mathematics, Science, and Engineering & Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48823, USA.

[5] Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA

[6] Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Previously proposed quantum algorithms for solving linear systems of equations cannot be implemented in the near term due to the required circuit depth. Here, we propose a hybrid quantum-classical algorithm, called Variational Quantum Linear Solver (VQLS), for solving linear systems on near-term quantum computers. VQLS seeks to variationally prepare $|x\rangle$ such that $A|x\rangle \propto |b\rangle$. We derive an operationally meaningful termination condition for VQLS that allows one to guarantee that a desired solution precision ϵ is achieved. Specifically, we prove that $C \geqslant \epsilon^2/\kappa^2$, where C is the VQLS cost function and κ is the condition number of A. We present efficient quantum circuits to estimate C, while providing evidence for the classical hardness of its estimation. Using Rigetti's quantum computer, we successfully implement VQLS up to a problem size of 1024×1024. Finally, we numerically solve nontrivial problems of size up to $2^{50} \times 2^{50}$. For the specific examples that we consider, we heuristically find that the time complexity of VQLS scales efficiently in ϵ, κ, and the system size N.

1 Introduction

Linear systems of equations play an important role in many areas of science and technology, including machine learning [1, 2], solving partial differential equations [3], fitting polynomial curves [4], and analyzing electrical circuits [5]. In the past decade, significant attention has been given to the possibility of solving linear systems on quantum computers. Classically solving an $N \times N$ linear system (N equations for N unknowns) scales polynomially in N. In contrast, Harrow-Hassidim-Lloyd (HHL) introduced a quantum algorithm that scales logarithmically in N, suggesting that quantum computers may provide an exponential speedup for certain linear system problems [6]. More precisely, the HHL algorithm treats the Quantum Lin-

ear Systems Problem (QLSP), where the goal is to prepare a quantum state $|x\rangle$ that is proportional to a vector x that satisfies the equation $Ax = b$. If both A and b are sparse, then for a fixed precision ϵ in the solution, the complexity of HHL scales polynomially in $\log N$ and κ, where κ is the condition number of A, i.e, the ratio of the largest to the smallest singular value. Further improvements to HHL have reduced the complexity to linear κ scaling [7, 8] and polylogarithmic scaling in $1/\epsilon$ [9, 10], as well as improved the sparsity requirements [11].

The aforementioned quantum algorithms hold promise for the future, when large-scale quantum computers exist with enough qubits for quantum error correction. The timescale for such computers remains an open question, but is typically estimated to be on the order of two decades. On the other hand, commercial quantum computers currently exist with ~ 50 noisy qubits, with the number of qubits rapidly increasing. A crucial question is how to make use of such noisy intermediate-scale quantum (NISQ) computers [12]. In principle, one can implement the aforementioned quantum algorithms on NISQ devices, however noise limits the problem size to be extremely small. For example, the HHL algorithm has been implemented with superconducting qubits [13, 14], nuclear magnetic resonance (NMR) [15], and photonic devices [16, 17], but these experiments were limited to a problem size of 2×2. More recently, an alternative approach based on an adiabatic-inspired quantum algorithm [8] was implemented with NMR for an 8×8 problem, and this appears to be the current record for the largest linear system solved with a gate-based quantum computer [18].

An interesting strategy to make use of NISQ devices is to employ variational hybrid quantum-classical algorithms (VHQCAs). VHQCAs manage to reduce quantum circuit depth at the expense of additional classical optimization. Specifically, VHQCAs employ a short-depth quantum circuit to efficiently evaluate a cost function, which depends on the parameters of a quantum gate sequence, and then leverage
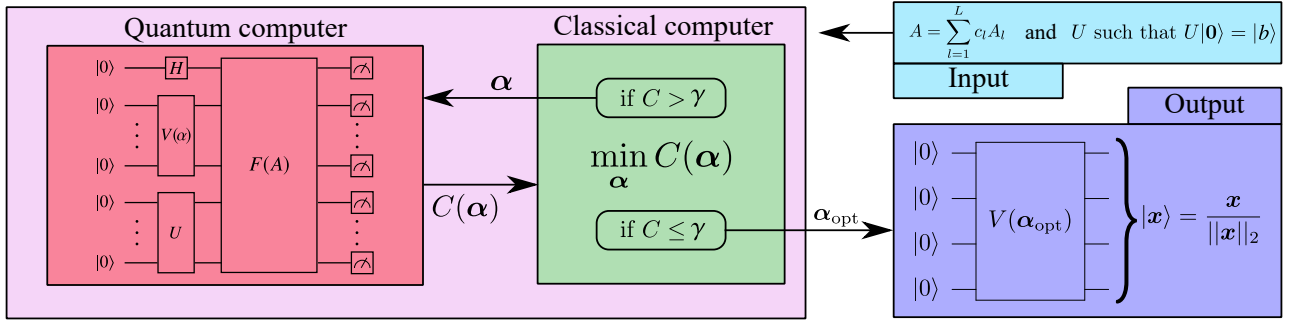
Accepted in 〈 〉uantum 2023-11-15, click title to verify. Published under CC-BY 4.0.

1

Figure 1: Schematic diagram for the VQLS algorithm. The input to VQLS is a matrix A written as a linear combination of unitaries A_l and a short-depth quantum circuit U which prepares the state $|b\rangle$. The output of VQLS is a quantum state $|x\rangle$ that is approximately proportional to the solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$. Parameters $\boldsymbol{\alpha}$ in the ansatz $V(\boldsymbol{\alpha})$ are adjusted in a hybrid quantum-classical optimization loop until the cost $C(\boldsymbol{\alpha})$ (local or global) is below a user-specified threshold. When this loop terminates, the resulting gate sequence $V(\boldsymbol{\alpha}_{\text{opt}})$ prepares the state $|x\rangle = \boldsymbol{x}/||\boldsymbol{x}||_2$, from which observable quantities can be computed. Furthermore, the final value of the cost $C(\boldsymbol{\alpha}_{\text{opt}})$ provides an upper bound on the deviation of observables measured on $|x\rangle$ from observables measured on the exact solution.

well-established classical optimizers to minimize this cost function. For example, while Shor's algorithm for factoring is not a near-term algorithm, recently a VHQCA for factoring was introduced potentially making factoring nearer term [19]. Other VHQCAs have been proposed for chemistry [20–23], simulation [24–28], data compression [29], state diagonalization [30–32], compiling [33, 34], quantum foundations [35], fidelity estimation [36], and metrology [37].

In this work, we propose a VHQCA for solving the QLSP. Our algorithm, called the Variational Quantum Linear Solver (VQLS), employs a cost function that quantifies either the global or local closeness of the quantum states $A|x\rangle$ and $|b\rangle$, where the latter is a normalized version of $\boldsymbol{b}$. We provide efficient quantum circuits to estimate our cost functions, and show under typical complexity assumptions that they cannot be efficiently estimated classically. Furthermore, we derive operational meaning for our cost functions, as upper bounds on ϵ^2/κ^2. This is crucial since it gives a termination criterion for VQLS that guarantees a desired precision ϵ.

It is important to emphasize that all VHQCAs are heuristic algorithms, making rigorous complexity analysis difficult. Nevertheless, our numerical simulations (without finite sampling, and both for specific A and for randomly chosen A) indicate that the run time of VQLS scales efficiently in κ, ϵ, and N. Namely, we find evidence of (at worst) linear scaling in κ, logarithmic scaling in $1/\epsilon$, and polylogarithmic scaling in N.

We employ Rigetti's Quantum Cloud Services to implement VQLS. With their quantum hardware, we were able to successfully solve a particular linear system of size 1024×1024. We are therefore optimistic that VQLS could provide a near-term approach to the QLSP.

2   Results

2.1   VQLS Algorithm

2.1.1   Overview

Figure 1 shows a schematic diagram of the VQLS algorithm. The input to VQLS is: (1) an efficient gate sequence U that prepares a quantum state $|b\rangle$ that is proportional to the vector $\boldsymbol{b}$, and (2) a decomposition of the matrix A into a linear combination of L unitaries of the form

$$A = \sum_{l=1}^{L} c_l A_l \,, \tag{1}$$

where the A_l are unitaries, and the c_l are complex numbers. The assumption that A is given in this form is analogous to the assumption that the Hamiltonian H in the variational quantum eigensolver [20] is given as a linear combination of Pauli operators $H = \sum_{l=1}^{L} c_l \sigma_l$, where naturally one makes the assumption that L is only a polynomial function of the number of qubits, n. Additionally, we assume $\kappa < \infty$ and $\|A\| \leqslant 1$, and that the A_l unitaries can be implemented with efficient quantum circuits. Appendix A describes an efficient method to decompose A in the form of (1) for the case when A is a sparse matrix.

With this input, the Quantum Linear Systems Problem (QLSP) is to prepare a state $|x\rangle$ such that $A|x\rangle$ is proportional to $|b\rangle$. To solve this problem, VQLS employs an ansatz for the gate sequence $V(\boldsymbol{\alpha})$ that prepares a potential solution $|x(\boldsymbol{\alpha})\rangle = V(\boldsymbol{\alpha})|\boldsymbol{0}\rangle$. The parameters $\boldsymbol{\alpha}$ are input to a quantum computer, which prepares $|x(\boldsymbol{\alpha})\rangle$ and runs an efficient quantum circuit that estimates a cost function $C(\boldsymbol{\alpha})$. The precise details of the cost function and its estimation are discussed below. We simply remark here that $C(\boldsymbol{\alpha})$ quantifies how much component $A|x\rangle$ has orthogonal to $|b\rangle$. The value of $C(\boldsymbol{\alpha})$ from the quantum computer

2

is returned to the classical computer which then adjusts $\boldsymbol{\alpha}$ (via a classical optimization algorithm) in an attempt to reduce the cost. This process is iterated many times until one reaches a termination condition of the form $C(\boldsymbol{\alpha}) \leqslant \gamma$, at which point we say that $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\text{opt}}$.

VQLS outputs the parameters $\boldsymbol{\alpha}_{\text{opt}}$, which can then be used to prepare the quantum state $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle = V(\boldsymbol{\alpha}_{\text{opt}})|\mathbf{0}\rangle$. One can then measure observables of interest on the state $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle$ in order to characterize the solution vector. Due to the operational meaning of our cost function (discussed below), one can upper bound the deviation of observable expectation values for $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle$ from those of the true solution, based on the value of the cost function. Hence, before running VQLS, one can decide on a desired error tolerance ϵ, where

$$\epsilon = (1/2)\text{Tr}|\ |x_0\rangle\langle x_0| - |x(\boldsymbol{\alpha}_{\text{opt}})\rangle\langle x(\boldsymbol{\alpha}_{\text{opt}})|\ | \quad (2)$$

is the trace distance between the exact solution $|x_0\rangle$ and the approximate solution $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle$. This ϵ then translates into a threshold value γ that the final cost $C(\boldsymbol{\alpha}_{\text{opt}})$ must achieve (see (9) for the relation between ϵ and γ).

2.1.2 Cost functions

For simplicity, we write $|x(\boldsymbol{\alpha})\rangle$ as $|x\rangle$ henceforth. Here we discuss several reasonable cost functions. A simple, intuitive cost function involves the overlap between the (unnormalized) projector $|\psi\rangle\langle\psi|$, with $|\psi\rangle = A|x\rangle$, and the subspace orthogonal to $|b\rangle$, as follows:

$$\widehat{C}_G = \text{Tr}(|\psi\rangle\langle\psi|(\mathbb{1} - |b\rangle\langle b|)) = \langle x|H_G|x\rangle. \quad (3)$$

We note that one can view this cost function as the expectation value of an effective Hamiltonian

$$H_G = A^\dagger(\mathbb{1} - |b\rangle\langle b|)A, \quad (4)$$

which is similar to the final Hamiltonian in Ref. [8]. The $\widehat{C}_G$ function is small if $|\psi\rangle$ is nearly proportional to $|b\rangle$ or if the norm of $|\psi\rangle$ is small. The latter does not represent a true solution, and hence to deal with this, one can divide $\widehat{C}_G$ by the norm of $|\psi\rangle$ to obtain

$$C_G = \widehat{C}_G/\langle\psi|\psi\rangle = 1 - |\langle b|\Psi\rangle|^2, \quad (5)$$

where $|\Psi\rangle = |\psi\rangle/\sqrt{\langle\psi|\psi\rangle}$ is a normalized state. As discussed in the Supplemental Material, C_G and $\widehat{C}_G$ have similar performance for the QLSPs that we considered.

We emphasize that global cost functions such as those in (3) and (5) can exhibit barren plateaus, i.e., cost function gradients that vanish exponentially in the number of qubits n, see Ref. [38]. To improve trainability for large n, one can introduce local versions of these costs, as follows:

$$\widehat{C}_L = \langle x|H_L|x\rangle, \qquad C_L = \widehat{C}_L/\langle\psi|\psi\rangle, \quad (6)$$
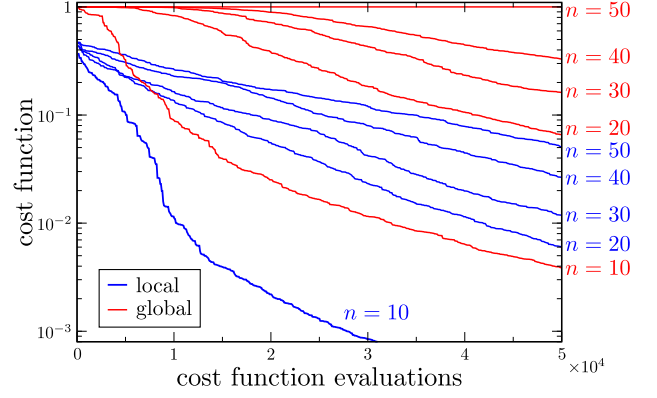

Figure 2: Comparison of local C_L and global C_G cost performance. Here we consider the QLSP of Eq. (26) for different system sizes. In all cases $\kappa = 20$. For each $n \in \{10, \ldots, 50\}$, we plot the cost value versus the number of cost function evaluations. As n increases it becomes increasingly hard to train to global cost function. At $n = 50$, our optimization cannot significantly lower C_G below a value of one. On the other hand, we are able to train C_L for all values of n considered.

where the effective Hamiltonian is

$$H_L = A^\dagger U \left(\mathbb{1} - \frac{1}{n}\sum_{j=1}^n |0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}} \right) U^\dagger A, \quad (7)$$

with $|0_j\rangle$ the zero state on qubit j and $\mathbb{1}_{\bar{j}}$ the identity on all qubits except qubit j. One can show that (see Appendix B)

$$\widehat{C}_L \leqslant \widehat{C}_G \leqslant n\widehat{C}_L, \qquad C_L \leqslant C_G \leqslant nC_L, \quad (8)$$

which implies that $\widehat{C}_L = 0 \leftrightarrow \widehat{C}_G = 0$ and $C_L = 0 \leftrightarrow C_G = 0$. We assume that κ is not infinite (i.e., that A is full rank) and hence that $\langle\psi|\psi\rangle \neq 0$. This implies that all four cost functions vanish under precisely the same conditions, namely, when $|\psi\rangle \propto |b\rangle$, which is the case when $|x\rangle$ is a solution to the QLSP.

As shown in Fig. 2, as n increases it becomes increasingly hard to optimize the global cost function C_G. On the other hand, the local cost function C_L performs significantly better, as we are able to train C_L for systems of size up to $2^{50} \times 2^{50}$ (i.e., with 50 qubits). These results show that the vanishing gradients of global cost functions could make them untrainable for large n, and hence we propose using our local cost functions for large-scale implementations.

2.1.3 Operational meaning of cost functions

Here we provide operational meanings for the aforementioned cost functions. These operational meanings are crucial since they allow one to define termination conditions for VQLS in order to achieve a desired precision. In particular, we find that the fol-

lowing bounds hold in general:

$$\widehat{C}_G \geqslant \frac{\epsilon^2}{\kappa^2}, \ \ C_G \geqslant \frac{\epsilon^2}{\kappa^2}, \ \ \widehat{C}_L \geqslant \frac{1}{n}\frac{\epsilon^2}{\kappa^2}, \ \ C_L \geqslant \frac{1}{n}\frac{\epsilon^2}{\kappa^2} . \tag{9}$$

Note that one can take the right-hand-sides of these inequalities as the γ quantity shown in Fig. 1.

We remark that, for C_G and C_L, the bounds in (9) can be tightened (by using the bounds on $\widehat{C}_G$ and $\widehat{C}_L$ in (9)) as follows:

$$C_G \geqslant \frac{\epsilon^2}{\kappa^2 \langle \psi | \psi \rangle} , \quad C_L \geqslant \frac{1}{n}\frac{\epsilon^2}{\kappa^2 \langle \psi | \psi \rangle} . \tag{10}$$

Here, $\langle \psi | \psi \rangle$ is experimentally computable (see (14) below) and satisfies $\langle \psi | \psi \rangle \leqslant 1$. Hence, when training C_G or C_L, one can employ the right-hand-sides of (10) as opposed to those of (9) as the termination condition γ.

Furthermore, one can employ the operational meaning of the trace distance [39] to note that, for any POVM element M, we have $\epsilon \geqslant D(M)$, where

$$D(M) = |\langle x|M|x\rangle - \langle x_0|M|x_0\rangle| \tag{11}$$

measures the difference between expectation values on $|x\rangle$ and $|x_0\rangle$. Relaxing to the general case where M is any Hermitian observable gives $\epsilon \geqslant D(M)/(2\|M\|)$, and hence (9) is a bound on observable differences.

Let us now provide a proof for (9). Consider first that $\widehat{C}_G = \langle H_G \rangle$, with the eigenstates and eigenvalues of H_G denoted by $\{|x_i\rangle\}$ and $\{E_i\}$, respectively for $i = 0, 1, \dots$. By construction $|x_0\rangle$ is the ground state of H_G with $E_0 = 0$. In what follows we assume for simplicity that $|x_1\rangle$ is non-degenerate, although the same proof approach works for the degenerate case.

It is clear that for a given ϵ, the smallest energy $\langle H_G \rangle$ (hence cost) is achieved if the state $|x\rangle$ is a superposition of $|x_0\rangle$ and $|x_1\rangle$ only. One can see this by expanding an arbitrary state $|x\rangle$ in the energy eigenbasis, $|x\rangle = \sum_i \chi_i |x_i\rangle$, and noting that ϵ depends only on the magnitude of χ_0. Hence for a fixed ϵ, one is free to vary the set of coefficients $\{\chi_i\}_{i \neq 0}$, and the set that minimizes the energy corresponds to choosing $\chi_i = 0$ for all $i > 1$.

So we take:

$$|x\rangle = \cos(\theta/2)|x_0\rangle + e^{i\phi}\sin(\theta/2)|x_1\rangle , \tag{12}$$

and the associated energy is given by

$$\langle x|H_G|x\rangle = E_1 \sin^2(\theta/2) \geqslant \frac{\sin^2(\theta/2)}{\kappa^2} , \tag{13}$$

where we used the fact that $E_0 = 0$, and that the first excited state energy satisfies $E_1 \geqslant 1/\kappa^2$ (which was shown in Ref. [8]). The trace distance between $|x\rangle$ and $|x_0\rangle$ can be easily computed as $\sqrt{1 - |\langle x|x_0\rangle|^2}$, which results in $\epsilon = |\sin(\theta/2)|$. Inserting this into (13) yields $\widehat{C}_G \geqslant \epsilon^2/\kappa^2$. The remaining inequalities in (9) follow from (8) and from the fact that $\langle \psi | \psi \rangle \leqslant 1$, which implies $C_G \geqslant \widehat{C}_G$.

2.1.4 Cost evaluation

In principle, all the aforementioned cost functions can be efficiently evaluated using the Hadamard Test circuit and simple classical post-processing. However, in practice, care must be taken to minimize the number of controlled operations in these circuits. Consider evaluating the term $\langle \psi | \psi \rangle$, which can be written as

$$\langle \psi | \psi \rangle = \sum_{ll'} c_l c_{l'}^* \beta_{ll'} , \tag{14}$$

with

$$\beta_{ll'} = \langle \mathbf{0}|V^\dagger A_{l'}^\dagger A_l V|\mathbf{0}\rangle. \tag{15}$$

There are $L(L-1)/2$ different $\beta_{ll'}$ terms that one needs to estimate, and which can be measured with Hadamard Tests. The Hadamard Test involves acting with V on $|\mathbf{0}\rangle$, and then using an ancilla as the control qubit, applying C_{A_l} followed by $C_{A_{l'}^\dagger}$, where C_W denotes controlled-W (see Appendix for precise circuits).

In addition, for $\widehat{C}_G$ and C_G, one needs to evaluate

$$|\langle b|\psi\rangle|^2 = |\langle \mathbf{0}|U^\dagger A V|\mathbf{0}\rangle|^2 = \sum_{ll'} c_l c_{l'}^* \gamma_{ll'} , \tag{16}$$

with

$$\gamma_{ll'} = \langle \mathbf{0}|U^\dagger A_l V|\mathbf{0}\rangle \langle \mathbf{0}|V^\dagger A_{l'}^\dagger U|\mathbf{0}\rangle. \tag{17}$$

The γ_{ll} terms are easily estimated by applying $U^\dagger A_l V$ to $|\mathbf{0}\rangle$ and then measuring the probability of the all-zeros outcome. For the $L(L-1)/2$ terms with $l \neq l'$, there are various strategies to estimate $\gamma_{ll'}$. For example, one could estimate the L terms of the form $\langle \mathbf{0}|U^\dagger A_l V|\mathbf{0}\rangle$ with a Hadamard Test, but one would have to control all of the unitaries: V, A_l, and $U^\dagger$. Instead, we introduce a novel circuit called the Hadamard-Overlap Test that directly computes $\gamma_{ll'}$ without having to control V or U at the expense of doubling the number of qubits. This circuit is schematically shown in Fig. 1 and explained in detail in Appendix C.

Finally, for $\widehat{C}_L$ and C_L, one needs to estimate terms of the form

$$\delta_{ll'}^{(j)} = \langle \mathbf{0}|V^\dagger A_{l'}^\dagger U(|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\mathbf{0}\rangle. \tag{18}$$

These terms can either be estimated with the Hadamard-Overlap Test or with the Hadamard Test, which are discussed in Appendix C.

2.1.5 Classical hardness of computing the cost functions

Here we state that computing the cost functions in (3), (5), and (6) is classically hard under typical complexity assumptions. As shown in Appendix D, the following proposition holds:
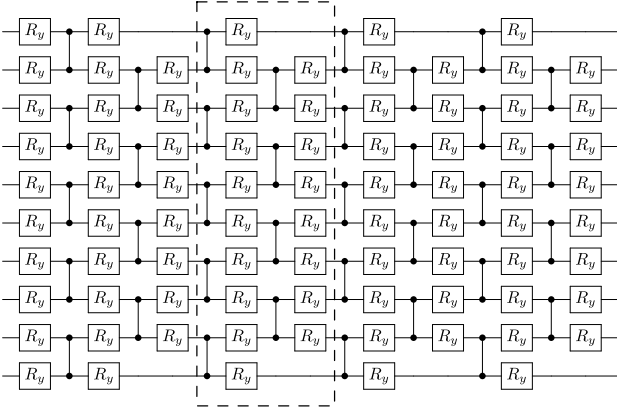
Figure 3: Fixed-structure layered Hardware-Eficient Ansatz for $V(\boldsymbol{\alpha})$. As indicated by the dashed box, each layer is composed of controlled-Z gates acting on alternating pairs of neighboring qubits which are preceded and followed by single qubit rotations around the y-axis, $R_y(\alpha_i) = e^{-i\alpha_i Y/2}$. Shown is the case of four layers and $n = 10$ qubits. The number of variational parameters and gates scales linearly with n: for $n = 50$, four layers of this ansatz consist of 640 gates and 440 variational parameters.

Proposition 1. *The problem of estimating the VQLS cost functions $\widehat{C}_G$, C_G, $\widehat{C}_L$, or C_L to within precision $\pm\delta = 1/\operatorname{poly}(n)$ is* DQC1-*hard.*

Recall that the complexity class Deterministic Quantum Computing with 1 Clean Qubit (DQC1) consists of all problems that can be efficiently solved with bounded error in the one-clean-qubit model of computation [40]. Moreover, classically simulating DQC1 is impossible unless the polynomial hierarchy collapses to the second level [41, 42], which is not believed to be the case. Hence, Proposition 1 strongly suggests that a classical algorithm cannot efficiently estimate the VQLS cost functions, and hence VQLS cannot be efficiently simulated classically.

2.1.6 Ansatz

In the VQLS algorithm, $|x\rangle$ is prepared by acting on the $|\mathbf{0}\rangle$ state with a trainable gate sequence $V(\boldsymbol{\alpha})$. Without loss of generality, $V(\boldsymbol{\alpha})$ can be expressed in terms of L gates from a gate alphabet $\mathcal{A} = \{G_k(\alpha)\}$ as

$$V(\boldsymbol{\alpha}) = G_{k_L}(\alpha_L) \dots G_{k_i}(\alpha_i) \dots G_{k_1}(\alpha_1). \quad (19)$$

Here $\boldsymbol{k} = (k_L, \dots, k_1)$ identifies the types of gates and their placement in the circuit (i.e., on which qubit they act), while $\boldsymbol{\alpha}$ are continuous parameters. When working with a specific quantum hardware, it is convenient to choose a Hardware-Efficient Ansatz [43], where $\mathcal{A}$ is composed of gates native to that hardware. This reduces the gate overhead that arises when implementing the algorithm in the actual device. We use the term "fixed-structure ansatz" when the gate structure of $V(\boldsymbol{\alpha})$ is fixed (i.e., when $\boldsymbol{k}$ is fixed), and

when one only optimizes over $\boldsymbol{\alpha}$. Figure 3 shows an example of such an ansatz, with $\mathcal{A}$ composed of single qubit y-rotations and controlled-Z gates. We employ the ansatz in Fig. 3 for the heuristics in Section 2.2.1. Let us remark that this ansatz can have trainability issues [38, 44] for large-scale problems.

Strategies such as layer-by-layer training [45] and correlating the $\boldsymbol{\alpha}$ parameters [46] have been shown to be effective to address these trainability issues. In addition, trainability could be further improved by combining these strategies with more advanced ansatz architectures, and we now consider two such architectures. First we discuss a "variable structure ansatz" [30, 47], where one optimizes over the gate angles and the gate placement in the circuit, i.e., where one optimizes over $\boldsymbol{\alpha}$ and also over $\boldsymbol{k}$. We employ such ansatz for our heuristics in Section 2.2.2. We refer the reader to Appendix E for a discussion of the optimization method employed for a variable structure ansatz.

In addition to the aforementioned ansatz, one can also employ the Quantum Alternating Operator Ansatz (QAOA) [48, 49] to construct the unitary $V(\boldsymbol{\alpha})$ and avoid trainability issues. The QAOA consists of evolving the $H^{\otimes n}|\mathbf{0}\rangle$ state (where H denotes the Hadamard unitary) by two Hamiltonians for a specified number of layers, or rounds. These Hamiltonians are conventionally known as *driver* and *mixer* Hamiltonians, and respectively denoted as H_D and H_M. Since the ground state of both H_G and H_L is $|x_0\rangle$, we can either use (4) or (7) as the driver Hamiltonian H_D. Evolving with H_D for a time α_i corresponds to the unitary operator $U_D(\alpha_i) := e^{-iH_D\alpha_i}$. Moreover, one can take the mixer Hamiltonian to be the conventional $H_M = \prod_{i=1}^{n} X_i$, where X_i denotes Pauli X acting on the ith qubit. Accordingly, evolving with H_M for a time α_j yields the unitary operator $U_M(\alpha_j) := e^{-iH_M\alpha_j}$. The trainable ansatz $V(\boldsymbol{\alpha})$ is then obtained by alternating the unitary operators $U_D(\alpha_i)$ and $U_M(\alpha_j)$ p times:

$$V(\boldsymbol{\alpha}) = e^{-iH_M\alpha_{2p}} e^{-iH_D\alpha_{2p-1}} \cdots e^{-iH_M\alpha_2} e^{-iH_D\alpha_1}. \quad (20)$$

In this ansatz, each α_i is a trainable continuous parameter. We note that QAOA is known to be universal as the number of layers p tends to infinity [48, 50], and that finite values of p have obtained good results for several problems [51–53]. In the Supplemental Material we present results of a small scale implementation of VQLS with a QAOA ansatz.

Let us remark that Ref. [6] showed that it is possible to efficiently generate an accurate approximation to the true solution $|x_0\rangle$, i.e., with a number of gates that is polynomial in n, assuming certain constraints on A and $\boldsymbol{b}$. Therefore, in principle, one may efficiently approximate these sort of solutions with a universal variational ansatz, such as the ones that we discussed above.

2.1.7 Training algorithm

There are several classical optimizers that may be employed to train $V(\boldsymbol{\alpha})$ and minimize the cost functions of VQLS. For example, our heuristics in Section 2.2.1 employ an optimization method that, at each iteration, chooses a random direction $\boldsymbol{w}$ in the parameter space along which to perform a line search, i.e., to solve $\min_{s \in \mathbb{R}} C(\boldsymbol{\alpha} + s\boldsymbol{w})$. On the other hand, in Section 2.2.2 we perform an optimization where all the parameters in $\boldsymbol{\alpha}$ are independently optimized at each iteration.

In addition, there has been an increasing interest in gradient-based methods for VHQCAs [54–56] as it has been shown that the first-order gradient information can be directly measured [57, 58] and can lead to faster rates of convergences to the optimum [59]. To enable gradient-based strategies, Appendix F derives explicit formulas for the gradients of the cost functions and shows that the same circuits used to compute the cost functions can be used to compute their gradients. Finally, when employing the QAOA ansatz we leverage literature on QAOA-specific training (for instance, Ref. [52]).

2.1.8 Resilience to noise

Recently, it was shown that certain VHQCAs, specifically those for compiling, can exhibit noise resilience in the sense that the optimal parameters are unaffected by certain noise models [60]. However, the generality of this Optimal Parameter Resilience (OPR) phenomenon is not clear. For the VQLS algorithm, the analysis of noise is made complicated by the fact that different quantum circuits are used to compute different terms in the cost function. However, perhaps surprisingly, we are able to prove that VQLS does exhibit the OPR phenomenon. Specifically, we show in Appendix G that our normalized cost functions C_G and C_L are both resilient to global depolarizing noise. We also show that C_L is resilient to measurement noise. This is encouraging since C_L is our proposed cost function of choice for large-scale implementations, although we leave an analysis of more complicated noise models for future work.

Because of OPR, we are able to train the variational ansatz and learn the correct parameters which prepare the solution $|x\rangle$. However, we still have a noisy estimate of the cost function which affects the termination conditions ((9) or (10)) used to achieve the desired precision. In what follows, we outline an error mitigation procedure known as Probabilistic Error Cancellation (PEC) [61] which allows us to certify the termination conditions in the presence of noise.

The basic idea of PEC is to represent an ideal gate in a basis of noisy gates which a given quantum computer can implement. Exact procedures to do so are known for simple noise models such as depolarizing and amplitude damping noise. Generally, as long as

the gates which a quantum computer can implement form a basis for unitary operations, an arbitrary unitary can be expressed in this basis by solving a linear program [61]. Using the notation of Ref. [61], an ideal circuit U_β can thus be written

$$U_\beta = \gamma_\beta \sum_{\alpha \in \Omega_T} \sigma_\alpha(\beta) p_\alpha(\beta) \mathcal{O}_\alpha \,. \tag{21}$$

Here, $p_\alpha(\beta)$ is a probability distribution, $\sigma_\alpha(\beta) = \pm 1$, and γ_β is the negativity of the quasi-probability distribution $a_\alpha(\beta) := \gamma_\beta \sigma_\alpha(\beta) p_\alpha(\beta)$. For any observable $A^\dagger = A$, the noise-free estimate is

$$\begin{aligned} E^*(\beta) &:= \mathrm{Tr}[A\mathcal{U}_\beta(|0\rangle\langle 0|)] \\ &= \gamma_\beta \sum_{\alpha \in \Omega_T} \sigma_\alpha(\beta) p_\alpha(\beta) \mathrm{Tr}[A\mathcal{O}_\alpha(|0\rangle\langle 0|)] \,. \end{aligned} \tag{22}$$

where T is the number of gates in the circuit, and Ω_T is the index set over all possible circuits which has size $|\Omega_T| = m^T$, where m is the number of gates in the noisy basis. This equation is exact, but the summation has exponentially terms in the number of gates T. However, we may define an estimate of the noise-free observable as the random variable

$$\hat{E}(\beta) := \frac{\gamma_\beta}{M} \sum_{i=1}^M \sigma_\beta(\alpha_i) \mathrm{Tr}[A\mathcal{O}_{\alpha_i}(|0\rangle\langle 0|)^{\otimes n}] \,, \tag{23}$$

where M is the number of samples. To achieve precision $|\hat{E}(\beta) - E^*(\beta)| \leq \delta$, it is shown in [61] that one can take $M = (\gamma_\beta/\delta)^2$ samples. Concretely, this means running $M = (\gamma_\beta/\delta)^2$ circuits sampled from the distribution $p_\alpha(\beta)$ and combining the results via (23).

We can apply this to VQLS as follows. Let $C \in \{C_G, C_L, \hat{C}_G, \hat{C}_L\}$ be a cost function and $\tilde{C}$ be its noisy estimate. The cost C is computed by evaluating $\mathrm{poly}(L)$ circuits (terms). Using PEC to compute each term with $M = (\gamma_\beta/\delta)^2$ circuits, we have $|\tilde{C} - C| \leq \mathrm{poly}(L)\delta$ since the error is additive. To achieve

$$|\tilde{C} - C| \leq \delta \tag{24}$$

for any $\delta > 0$, we thus may run

$$M = \frac{\gamma_\beta^2 \, \mathrm{poly}(L)}{\delta^2} \tag{25}$$

circuits.

In summary, OPR shows that we can obtain the optimal parameters for certain noise models, meaning that we will indeed obtain the optimal solution $|x\rangle$ to the QLSP. The above procedure based on PEC shows that, with polynomial overhead, we can verify if the solution is optimal by satisfying the termination conditions. Note that this overhead is not required during the training phase of the VQLS but rather only after the training phase has halted. If the verification fails, additional training may be done, or the training may be restarted.
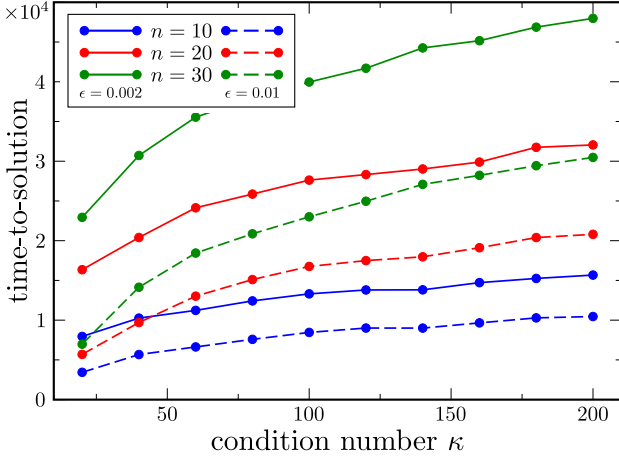
Figure 4: Scaling with κ for the Ising-inspired QLSP. The time-to-solution is the number of executions needed to guarantee a precision of $\epsilon = 0.002$ (solid line) and $\epsilon = 0.01$ (dashed line). Curves are shown for $n = 10, 20, 30$ qubits. In each case we averaged over 30 runs of the VQLS algorithm with four layers of the Layered Hardware-Efficient Ansatz of Fig. 3, and we trained the gate sequence by minimizing C_L of (6). While the the κ scaling appears to be sub-linear here, it is known that linear scaling is optimal in general [6], and hence the observed scaling is likely specific to this example.

2.2 Heuristic Scaling

Here we study the scaling of VQLS with the condition number κ, error tolerance ϵ, and number of qubits n. First we consider a specific QLSP for which $|x_0\rangle$ admits an efficient matrix-product-state representation, allowing us to simulate large values of n. We then consider QLSPs where the matrix A is randomly generated. In both cases we restrict A to be a sparse matrix, which is standard for QLSPs [6], and we simulate VQLS without finite sampling. Moreover, we quantify the run time of VQLS with the *time-to-solution*, which refers to the number of exact cost function evaluations during the optimization needed to guarantee that ϵ is below a specified value. In practice, for large-scale implementations where the true solution $|x_0\rangle$ is unknown, ϵ cannot be directly calculated. Rather, one can use the operational meaning of our cost function in (9) to upper-bound ϵ. Hence, we take this approach in all of our heuristics, i.e., we use the value of the cost, combined with (9), to determine the worst-case ϵ. We emphasize that, while it is tempting to directly compute ϵ from (2) in one's heuristics, this is essentially cheating since $|x_0\rangle$ is unknown, and this is why our certification procedure is so important.
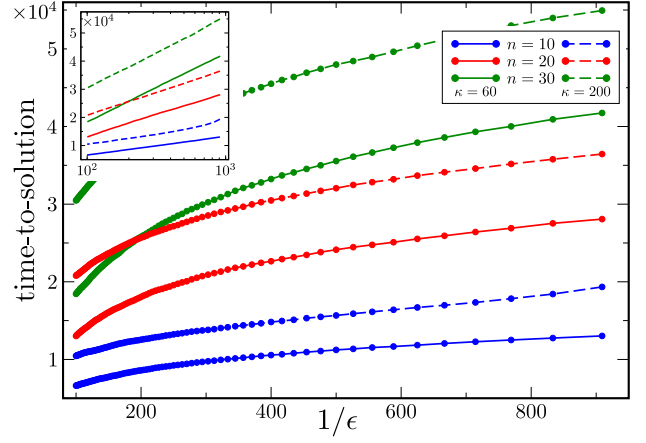

Figure 5: Scaling with $1/\epsilon$ for the Ising-inspired QLSP. Curves are shown for $n = 10, 20, 30$ qubits, with $\kappa = 60$ (solid line) or $\kappa = 200$ (dashed line). In all cases $V(\boldsymbol{\alpha})$ was composed of four layers of the Layered Hardware-Efficient Ansatz of Fig. 3, and we trained the local cost C_L. The time-to-solution was obtained by averaging over 30 runs of the VQLS algorithm. The inset depicts the same data in a logarithmic scale. The dependence on $1/\epsilon$ appears to be logarithmic, i.e., linear on a logarithmic scale.

2.2.1 Ising-inspired QLSP

Here we numerically simulate VQLS to solve the QLSP defined by the sparse matrix

$$A = \frac{1}{\zeta} \left(\sum_{j=1}^{n} X_j + J \sum_{j=1}^{n-1} Z_j Z_{j+1} + \eta \mathbb{1} \right) , \quad (26)$$

$$|b\rangle = H^{\otimes n} |\mathbf{0}\rangle ,$$

where the subscripts in (26) denote the qubits acted upon non-trivially by the Pauli operator. Here, we set $J = 0.1$. The parameters ζ and η are chosen such that the smallest eigenvalue of A is $1/\kappa$ and its largest eigenvalue is 1, which involves analytically computing [62] the smallest eigenvalue of the first two terms of A and then re-scaling A. As previously mentioned, this QLSP example is motivated from the fact that for $J = 0$ the solution is given by $|x_0\rangle = |b\rangle$. Hence for small J, $|x_0\rangle$ admits an efficient matrix-product-state representation.

Dependence on κ: Figure 4 shows our results, plotting time-to-solution versus κ for the QLSP in (26). Our numerical results were obtained by employing the layered Hardware-Efficient Ansatz of Fig. 3, and by training the local cost C_L for different values of n. Figure 4 shows that as the condition number κ is increased, the time-to-solution needed to achieve a given ϵ increases with a scaling that appears to be sub-linear. Hence VQLS scales efficiently with κ for this example. It is known that linear scaling is optimal [6]. Hence we expect that the scaling observed here is specific to this example, and indeed the example in the next subsection shows scaling that is closer to linear.
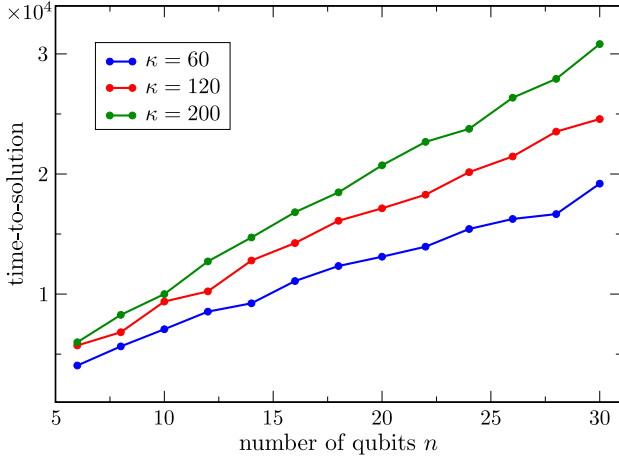
Figure 6: Scaling with n for the Ising-inspired QLSP. Curves are shown for $\epsilon = 0.01$ and for $\kappa = 60, 120, 200$. In all cases we trained the local cost C_L with four layers of the Layered Hardware-Efficient Ansatz of Fig. 3. The dependence on n appears to be linear (logarithmic in N) for this example.

Dependence on ϵ: To study the scaling of VQLS with ϵ, we numerically solved the QLSP in (26) for different values of κ and n. In all cases we trained the gate parameters by optimizing the C_L cost function. Figure 5 shows the time-to-solution versus $1/\epsilon$. These results show that as $1/\epsilon$ grows, the time-to-solution exhibits a logarithmic growth.

Dependence on n: The QLSP of (26) allows us to increase the number of qubits and analyze the scaling of VQLS with n. Here we implemented VQLS with $n = 6, 8, \ldots, 30$ and for $\kappa = 60, 120, 200$ by training the local cost function C_L. Figure 6 shows time-to-solution versus n. As the number of qubits increases the time-to-solution needed to guarantee a particular ϵ with κ fixed appears to increase linearly with n. This corresponds to logarithmic scaling in the linear system size N, analogous to that of the HHL algorithm [6].

2.2.2 Randomly generated QLSP

In this section we present scaling results for the case when the matrix A is randomly generated with the form

$$A = \xi_1 \Big(\mathbb{1} + \xi_2 \sum_j \sum_{k \neq j} p a_{j,k} \sigma_j^\alpha \sigma_k^\beta \Big). \qquad (27)$$

Here p is either 0 or 1 according to a fixed binomial distribution, $a_{j,k}$ are random weights in $(-1, 1)$, and σ_j^α is the Pauli matrix acting on qubit j with $\alpha = x, y, z$. For each $j, k = 1, \ldots, n$ in (27), α and β are randomly chosen. Finally, we remark that ξ_1, and ξ_2 are normalization coefficients that rescale the matrix so that its largest eigenvalue is 1 and its smallest is $1/\kappa$ (where κ is fixed).

For a given number of qubits n, we randomly created a matrix A according to (27), and we ran four independent instances of VQLS. We then selected the best run, i.e., the instance that required the small-

est number of cost function evaluations to reach a specified value of guaranteed ϵ (guarenteed via (9)). This procedure was then repeated for 10 independent random matrices A, and the time-to-solution was obtained as the average of the best run for each matrix.

Dependence on κ: In Fig. 7(a) we show the time-to-solution versus κ for matrices randomly generated according to (27), and for $n = 4$. Here we employed a variable-structure ansatz as described in Appendix E, and we trained the local cost in (6). Different curves represent different desired precision ϵ. The data were plotted in a log-log scale and each curve was fitted with a power function κ^m. In all cases we found $m < 1$, indicating that the scaling in κ for these examples is at worst linear. Linear scaling in κ is known to be optimal [6].

Dependence on ϵ: Let us now analyze the scaling of VQLS with respect to ϵ for matrices with different condition numbers. Figure 7(b) depicts the time-to-solution versus ϵ for matrices randomly generated according to (27), and for $n = 4$. All curves were fitted with a linear function, and since the x axis is in a logarithmic scale, the dependence on $1/\epsilon$ appears to be logarithmic. Upon examining Figures 7(a) and 7(b) collectively, the apparent scaling of the time-to-solution with κ and $\log(1/\epsilon)$ seems to exhibit a multiplicative behavior.

Dependence on n: In Fig. 7(c) we present the time-to-solution versus n needed to guarantee $\epsilon = 0.3$ for QLSPs with $n = 2, \ldots, 7$. All matrices A had condition number $\kappa = 10$. The data were fitted with a power function and we obtained the relation $y \sim n^{8.5}$. This corresponds to polylogarithmic scaling in N, which is the standard goal of quantum algorithms for the QLSP [7–10].

We refer the reader to the Supplemental Material for additional numerical simulations of VQLS for other QLSP examples, both with a Hardware-Efficient Ansatz and with a QAOA ansatz. These examples also exhibit efficient scaling behavior.

2.3 Implementation on quantum hardware

Here we present the results of a 1024×1024 (i.e., 10-qubit) implementation of VQLS using Rigetti's *16Q Aspen-4* quantum computer. Specifically, we solved the QLSP defined by the matrix A in (26), with $\zeta = \eta = 1$, and where the vector $|b\rangle = |0\rangle$ was the all zero state. The ansatz consisted of $R_y(\alpha_i)$ gates acting on each qubit. To adapt to hardware constraints, we computed the cost function C_G in (5) by expanding the effective Hamiltonian H_G in terms of Pauli operators and then employing Rigetti's quantum computer to estimate the expectation values of these terms.

The results of two representative VQLS runs are shown in Fig. 8. As shown, the cost function data obtained by training in a quantum computer closely matches the one obtained from training on a noiseless
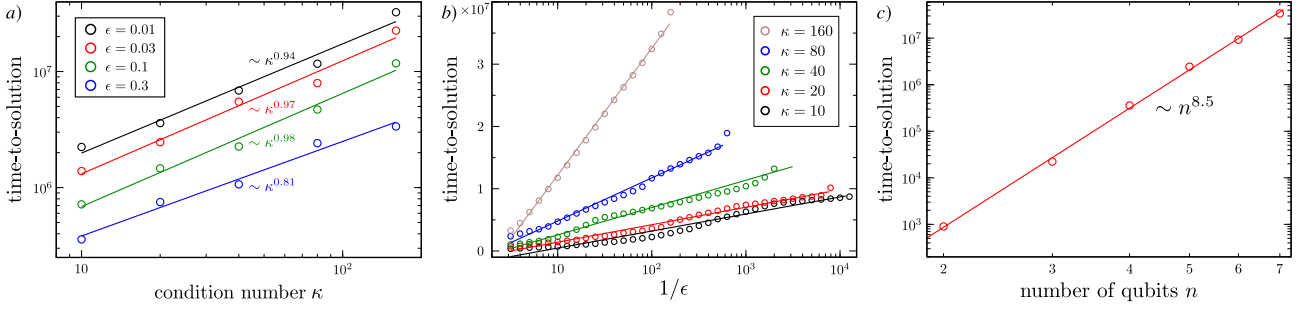
Figure 7: VQLS heuristic scaling for random matrices generated according to (27). The time-to-solution is the number of executions needed to guarantee a desired precision ϵ. In all cases we employed a variable-structure ansatz $V(\boldsymbol{\alpha})$ as described in Appendix E, and we trained the local cost C_L of (6). a) Time-to-solution versus κ for a system of $n = 4$ qubits. Axes are shown in a log-log scale. For each value of ϵ the data were fitted with a power function κ^m and in all cases $m < 1$, suggesting that the κ scaling appears to be sub-linear. b) Time-to-solution versus $1/\epsilon$ for a system of $n = 4$ qubits. The x axis is shown in a log scale. Each curve corresponds to a different condition number. For all values of κ the data were fitted with a linear function, implying that the $1/\epsilon$ scaling is logarithmic. c) Time-to-solution versus n needed to guarantee $\epsilon = 0.3$. All matrices had a condition number $\kappa = 10$. The plot employs a log-log scale. The data were fitted with a power function $y \sim n^{8.5}$, suggesting that the N dependence is polylogarithmic.

simulator. For each run on the QPU, the value of the cost function approaches zero, indicating that a good solution to the linear system was found.

Additional experiments performed on quantum hardware are presented in the Supplemental Material.

3 Discussion

In this work, we presented a variational quantum-classical algorithm called VQLS for solving the quantum linear systems problem. On the analytical side, we presented four different faithful cost functions, we derived efficient quantum circuits to estimate them while showing that they are difficult to estimate classically, and we proved operational meanings for them as upper bounds on ϵ^2/κ^2. On the numerical side, we studied the scaling of the VQLS run time by solving non-trivial problems of size up to $2^{50} \times 2^{50}$. For the examples considered, we found VQLS to scale efficiently, namely, at worst linearly in κ, logarithmically in $1/\epsilon$, and polylogarithmically in the linear system size N.

It remains to be seen how the VQLS training is affected by finite sampling, which is not accounted for in our heuristics. Our solution verification procedure in Sec. 2.1.3 will require the shot noise to appropriately scale with ϵ and κ as dictated by (9). Namely, the number of shots would need to scale as $(\kappa/\epsilon)^4$, although this complexity might be reduced if one does not require solution verification.

Furthermore, we utilized Rigetti's Quantum Cloud Services to implement VQLS for a particular problem up to a size of 1024×1024, which to our knowledge is the largest implementation of a linear system on quantum hardware. Interestingly, with our implementation on Rigetti's hardware, we noticed some preliminary evidence of noise resilience, along the same lines as those discussed in Ref. [60] for a different varia-
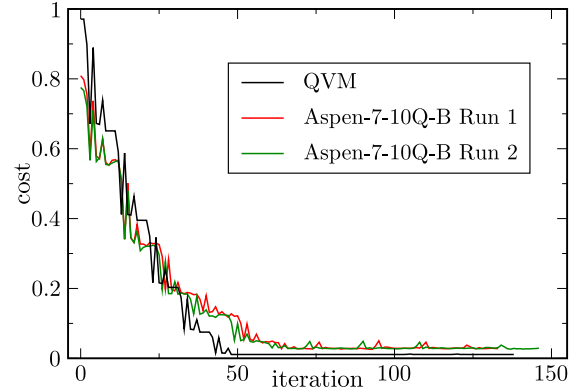

Figure 8: Implementation of VQLS on Rigetti's quantum hardware. Cost function C_G is plotted against the number of optimization steps, where A is defined in (26). One can observe that for each QPU run the cost function is reduced to a value below 10^{-1}. Due to noise present in the quantum device the cost does not go to zero.

tional algorithm. Namely, we noticed optimal parameter resilience, where VQLS learned the correct optimal parameters despite various noise sources (e.g., measurement noise, decoherence, gate infidelity) acting during the cost evaluation circuit. We will explore this in future work, including tightening our certification bound in (9) when accounting for noise.

Finally, we discuss how VQLS fits into the larger literature on quantum algorithms for linear systems. Most prior algorithms rely on time evolutions with the matrix A [6, 7, 9] or a simple function of it [8]. In these algorithms, the duration of the time evolution is $O(\kappa)$ in order to prepare a state $|x\rangle$ that is ϵ-close to the correct answer. In general, this can only be achieved with a quantum circuit of size linear in κ as per the "no fast-forwarding theorem" [63, 64]. This is even true if there exists a very short quantum circuit that prepares the desired state $|x\rangle$. The non-variational

algorithms simply cannot exploit this fact. On the other hand, a variational algorithm with a short-depth ansatz might be used to prepare such a state.

This does not mean, however, that the overall complexity of the variational algorithm does not depend on the condition number. This dependence enters through the stopping criteria given in (9). As the condition number increases, the cost has to be lowered further in order to guarantee an error of ϵ. This will undoubtedly require more iterations of the variational loop to achieve. In effect, our variational approach trades the gate complexity of non-variational algorithms with the number of iterations for a fixed circuit depth. This trade-off can be useful in utilizing NISQ devices without error correction.

We remark that other variational approaches to the QLSP distinct from ours were very recently proposed [65, 66]. Relatively speaking, the distinct aspects of our work include: (1) our quantitative certification procedure for the solution, (2) our clear approach to improve trainability for large-scale prob-
lems, (3) our novel circuits for efficient cost evalutaion, (4) our large-scale heuristics demonstrating efficient scaling, and (5) our large-scale implementations on quantum hardware. Finally, it exciting that, shortly after our paper was posted, two independent tutorials for the VQLS algorithm were created and added to IBM's open-source Qiskit textbook [67], and to Xanadu's PennyLane library [68].

A Sparse Matrices

In this section we describe how sparse matrices can be expressed as a linear combination of unitaries. Once this is achieved, the cost function can be computed using the methods described elsewhere in this paper. The construction below from Ref. [8] uses a version of Szegedy walks that applies to Hermitian matrices [69, 70]. Let A be a d-sparse matrix of dimension N and $n = \log_2 N$. We define unitary operations U_x, U_y, and S that act as follows:

$$U_x|j\rangle|0\rangle|0\rangle|0\rangle = \frac{1}{\sqrt{d}} \sum_{i \in F_j} |j\rangle|i\rangle|0\rangle \left(\sqrt{A_{ji}^*}|0\rangle + \sqrt{1 - |A_{ji}|}|1\rangle \right) , \tag{28}$$

$$U_y|0\rangle|j'\rangle|0\rangle|0\rangle = \frac{1}{\sqrt{d}} \sum_{i' \in F_{j'}} |i'\rangle|j'\rangle \left(\sqrt{A_{i'j'}}|0\rangle + \sqrt{1 - |A_{i'j'}|}|1\rangle \right) |0\rangle , \tag{29}$$

$$S|j\rangle|j'\rangle|\cdot\rangle|\cdot\rangle = |j'\rangle|j\rangle|\cdot\rangle|\cdot\rangle . \tag{30}$$

The first two registers have n qubits each, the last two registers have a single qubit each, and F_j is the set of indices i for which A_{ji} is nonzero. It follows that

$$A \otimes |\tilde{0}\rangle\langle\tilde{0}| = d \, |\tilde{0}\rangle\langle\tilde{0}| U_x^\dagger U_y S |\tilde{0}\rangle\langle\tilde{0}|$$
$$= \frac{d}{4} \left(\mathbb{1} - e^{i\pi P} \right) U_x^\dagger U_y S \left(\mathbb{1} - e^{i\pi P} \right) , \tag{31}$$

where we defined $|\tilde{0}\rangle := |0\rangle|0\rangle|0\rangle$ for the state of the last three registers and $P = |\tilde{0}\rangle\langle\tilde{0}|$. Equation (31) is a decomposition of A as a linear combination of 4 unitaries with equal weights of $d/4$.

We assume access to an oracle for A that acts as

$$|j\rangle|i\rangle|z\rangle \to |j\rangle|i\rangle|z \oplus A_{ji}\rangle , \tag{32}$$
$$|j\rangle|l\rangle \to |j\rangle|f(j,l)\rangle . \tag{33}$$

Here, j and i label the row and column of A, respectively, so that $j, i \in \{1, \dots, N\}$, and $f(j, l)$ is the column index of the l'th nonzero element of A in row j. We refer to this oracle as $\mathcal{O}_A$. This is the same as that used in previous works for the QLSP and Hamiltonian simulation such as Refs. [9, 70].

U_x can then be implemented in five steps as follows:

$$|j\rangle|0\rangle|0\rangle|0\rangle|0\rangle$$

$$\xrightarrow[\text{Hadamards}]{\log(d)} \frac{1}{\sqrt{d}} \sum_{l=0}^{d-1} |j\rangle|l\rangle|0\rangle|0\rangle|0\rangle$$

$$\xrightarrow{\mathcal{O}_A} \frac{1}{\sqrt{d}} \sum_{i \in F_j} |j\rangle|i\rangle|0\rangle|0\rangle|0\rangle$$

$$\xrightarrow{\mathcal{O}_A} \frac{1}{\sqrt{d}} \sum_{i \in F_j} |j\rangle|i\rangle|A_{ji}\rangle|0\rangle|0\rangle$$

$$\xrightarrow{M} \frac{1}{\sqrt{d}} \sum_{i \in F_j} |j\rangle|i\rangle|A_{ji}\rangle|0\rangle \left(\sqrt{A_{ji}^*}|0\rangle + \sqrt{1 - |A_{ji}|}|1\rangle \right)$$

$$\xrightarrow{\mathcal{O}_A} \frac{1}{\sqrt{d}} \sum_{i \in F_j} |j\rangle|i\rangle|0\rangle|0\rangle \left(\sqrt{A_{ji}^*}|0\rangle + \sqrt{1 - |A_{ji}|}|1\rangle \right) .$$

The third register is used to temporarily store the matrix elements of A and is discarded at the end. Its size depends on the precision with which the matrix elements of A are specified. A similar procedure can be followed to implement U_y.

Next, we briefly analyze the gate complexity of the unitaries in the decomposition of A given by Eq. (31).

Accepted in ⟨ ⟩uantum 2023-11-15, click title to verify. Published under CC-BY 4.0.

10

Since both U_x and U_y use 3 queries to $\mathcal{O}_A$, each unitary uses 6 queries. Other than the queries the main gate complexity comes from the implementation of the operations U_x and U_y. The gate complexity of both of these strongly depends on the form and precision with which the matrix elements of A are specified. The gate complexity of the unitaries $e^{i\pi P}$ is $O(\log N) = O(n)$.

Finally, we note that the decomposition in Eq. (31) has a prefactor proportional to d multiplying all the unitaries in the linear combination. This implies that in order to compute the cost function with a given precision, we have to estimate the expectation value of each unitary with a precision that is inversely proportional to d. For this reason we can only use this approach for sparse matrices where $d \ll N$.

B Faithfulness of the cost functions

We now prove (8), which is restated here:

$$\widehat{C}_L \leqslant \widehat{C}_G \leqslant n\widehat{C}_L, \qquad C_L \leqslant C_G \leqslant nC_L. \quad (34)$$

For the lower bound, let $\Pi_G = \mathbb{1} - |\mathbf{0}\rangle\langle\mathbf{0}|$ and $\Pi_L = \mathbb{1} - \frac{1}{n}\sum_{j=1}^{n}|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}$. Using the fact that $|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}} \geqslant |\mathbf{0}\rangle\langle\mathbf{0}|$, we have $\Pi_G \geqslant \Pi_L$ and hence $H_G \geqslant H_L$. This implies that $\widehat{C}_G \geqslant \widehat{C}_L$ and $C_G \geqslant C_L$.

For the upper bound, note that $\Pi_G = \sum_{\mathbf{z}\neq\mathbf{0}}|\mathbf{z}\rangle\langle\mathbf{z}|$. Let $\mathcal{S}_j$ denote the set of all bitstrings that have a one at position j, and let $\mathcal{S} = \bigcup_j \mathcal{S}_j$ denote the union of all of these sets. Then

$$n\Pi_L = \sum_j \sum_{\mathbf{z}\in\mathcal{S}_j}|\mathbf{z}\rangle\langle\mathbf{z}| \geqslant \sum_{\mathbf{z}\in\mathcal{S}}|\mathbf{z}\rangle\langle\mathbf{z}| = \sum_{\mathbf{z}\neq\mathbf{0}}|\mathbf{z}\rangle\langle\mathbf{z}| = \Pi_G. \quad (35)$$

Hence we have $nH_L \geqslant H_G$, which implies $n\widehat{C}_L \geqslant \widehat{C}_G$ and $nC_L \geqslant C_G$.

Equation (8) implies the faithfulness of the cost functions as follows. Because $\Pi_G \geqslant 0$ and $\Pi_L \geqslant 0$, we have that all four cost functions are non-negative. Furthermore, it is clear that if $A|x\rangle = |b\rangle$, then we have $\widehat{C}_G = C_G = \widehat{C}_L = C_L = 0$. Conversely, assuming that $A|x\rangle \neq |b\rangle$ implies that $C_G > 0$ and hence that all four cost functions are positive. Therefore, all four cost functions are faithful, vanishing if and only if $A|x\rangle = |b\rangle$.

C Cost evaluation circuits

In this section we present short-depth circuits for computing the cost functions of Eqs. (3), (5) and (6). In particular, we introduce the Hadamard-Overlap Test circuit, which should be of interest on its own as it is likely to have applications outside of the scope of VQLS.

C.1 Hadamard Test

Figure 9(a) shows a Hadamard Test which can be used to measure the coefficients $\beta_{ll'}$ defined in (15), and used to compute $\langle\psi|\psi\rangle$ as in (14). When the phase gate is excluded, the probability of measuring the ancilla qubit in the $|0\rangle^a$ state is $P(0) = (1 + \text{Re}[\beta_{ll'}])/2$, while the probability of measuring it in the $|1\rangle^a$ state is $P(1) = (1 - \text{Re}[\beta_{ll'}])/2$. Hence, by means of the Hadamard Test we can compute the real part of $\beta_{ll'}$ as

$$\text{Re}[\beta_{ll'}] = P(0) - P(1). \quad (36)$$

With a similar argument it can be easily shown that by including the phase gate one can compute $\text{Im}[\beta_{ll'}]$.

As we now show, in order to compute the coefficients $\delta_{ll'}^{(j)}$ in (18) we can use the previous result combined with those obtained by means of the Hadamard test of Figure 9(c). In particular, since $|0_j\rangle\langle 0_j| = (\mathbb{1}_j + Z_j)/2$, then we can express

$$\delta_{ll'}^{(j)} = \beta_{ll'} + \langle\mathbf{0}|V^\dagger A_{l'}^\dagger U(Z_j \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\mathbf{0}\rangle. \quad (37)$$

Hence, in order to calculate $\delta_{ll'}^{(j)}$ one only needs to measure the real and imaginary parts of the matrix elements $\langle\mathbf{0}|V^\dagger A_{l'}^\dagger U(Z_j \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\mathbf{0}\rangle$, which can be accomplished by means of the circuit in Fig. 9(c).

C.2 Hadamard-Overlap Test

Consider the circuit in Fig. 9(b), which we refer to as the Hadamard-Overlap Test. A nice feature of the Hadamard-Overlap Test is that it only requires one application of both U and V, and these unitaries do not need to be controlled, in contrast to the Hadamard Test. As explained below, the circuit for the Hadamard-Overlap Test can be obtained by combining the Hadamard Test with the Overlap circuit of Refs. [47, 71]. This circuit requires $2n + 1$ qubits and classical post-processing (which scales linearly with n) similar to that of the Overlap circuit.

Given that the Hadamard-Overlap Test combines two different ideas, let us describe its inner workings in detail. First, the circuit is initialized to the state $|0\rangle^a \otimes |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle$, with $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$. We will denote the first qubit, initialized to $|0\rangle^a$, as the ancilla, while we will refer to the next two sets of n-qubits as subsystems S_1 and S_2, respectively, as in Fig. 9(b). One then applies a Hadamard gate to the ancilla qubit, V to qubits in S_1, and the unitary U to the qubits in S_2, producing the state

$$|0\rangle^a \otimes |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle \rightarrow |+\rangle \otimes V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle.$$

The application of the controlled A_l and $A_{l'}^\dagger$ unitaries leads to

$$|+\rangle \otimes V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle$$
$$\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle \otimes V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle + |1\rangle \otimes A_l V|\mathbf{0}\rangle \otimes A_{l'}^\dagger U|\mathbf{0}\rangle\right).$$

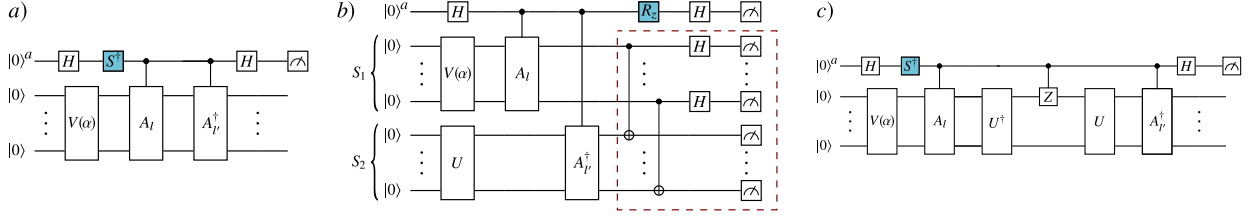Figure 9: a) Circuit for the Hadamard Test used to compute the coefficients $\beta_{ll'} = \langle\mathbf{0}|V^\dagger A_l^\dagger, A_l V|\mathbf{0}\rangle$ and calculate the inner product $\langle\psi|\psi\rangle$ of (14). The phase gate in the colored box is excluded when calculating the real part of $\beta_{ll'}$ and included when calculating its imaginary part. b) Hadamard-Overlap Test used to compute the coefficients $\gamma_{ll'}$ defined in (17). The Overlap circuit of Refs. [47, 71] is indicated in the dashed box. Here, the R_z gate in the colored box denotes a rotation about the z axis of an angle $-\pi/2$. Excluding (including) this rotation allows one to calculate the real (imaginary) part of $\gamma_{ll'}$. As explained in the text, additional post-processing is required. c) Hadamard Test circuit for computing $\delta_{ll'}^{(j)}$ as defined in (37). Shown here is case when $j = 1$.

Next, let us consider the case when the R_z gate in Fig. 9(b) is excluded. The next Hadamard gate on the ancilla leads to

$$\frac{1}{\sqrt{2}}\left(|0\rangle \otimes V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle + |1\rangle \otimes A_l V|\mathbf{0}\rangle \otimes A_{l'}^\dagger U|\mathbf{0}\rangle\right)$$

$$\to |\psi\rangle = \frac{1}{2}|0\rangle \otimes \left(V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle + A_l V|\mathbf{0}\rangle \otimes A_{l'}^\dagger U|\mathbf{0}\rangle\right)$$
$$+ \frac{1}{2}|1\rangle \otimes \left(V|\mathbf{0}\rangle \otimes U|\mathbf{0}\rangle - A_l V|\mathbf{0}\rangle \otimes A_{l'}^\dagger U|\mathbf{0}\rangle\right).$$

From here, one needs to perform a measurement over state $|\psi\rangle$. As shown in Fig. 9(b), we measure the ancilla on the computational basis and the S_1 and S_2 qubits on the Bell basis. More specifically, we can see that we are measuring the first qubit on S_1 and the first qubit in S_2 in the Bell basis, as well as the second qubit on S_1 and the second qubit in S_2 in the Bell basis, and so-on. At this point, we find it convenient to recall that the Bell basis on two qubits is composed of four states:

$$|\phi^{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\phi^{01}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\phi^{10}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |\phi^{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle),$$

Note that given a two-qubit state ρ, and denoting as $P(\phi^{pq}) = \text{Tr}[\rho|\phi^{pq}\rangle\langle\phi^{pq}|]$ the probability of obtaining the outcome $|\phi^{peq}\rangle$, by measuring ρ on the Bell basis, then

$$\sum_{pq=0,1} (-1)^{\delta_{pq,11}} P(\phi^{pq}) = P(\phi^{00}) + P(\phi^{01}) + P(\phi^{10}) - P(\phi^{11})$$
$$= \text{Tr}[\rho\text{SWAP}], \quad (38)$$

where here SWAP denotes the standard two-qubit SWAP gate. The previous result follows from the fact that the SWAP gate is diagonal in the Bell basis with $|\phi^{00}\rangle, |\phi^{01}\rangle, |\phi^{10}\rangle$ being eigenvectors of eigenvalue one, and $|\phi^{11}\rangle$ being an eigenvector of eigenvalue minus one. As such, a measurement on the Bell basis plus appropriate classical post-processing allows us to compute the expectation value of the SWAP operator.

From the previous, let us note that the measurement outcome from Fig. 9(b) can be expressed by the tuple $x\phi_1^{p_1 q_1}\phi_2^{p_2 q_2}\dots\phi_n^{p_n q_n}$ where $x = 0, 1$ denotes the measurement outcome of on the ancilla qubits, and $\phi_j^{p_j q_j}$ with $p_j, q_j = 0, 1$ and the Bell basis measurement outcome on the j-th qubits of S_1 and S_2. With this notation, $P(x\phi_1^{p_1 q_1}\phi_2^{p_2 q_2}\dots\phi_n^{p_n q_n})$ denotes the probability of outcome $x\phi_1^{p_1 q_1}\phi_2^{p_2 q_2}\dots\phi_n^{p_n q_n}$. If we compute the quantity

$$C(x) = \sum_{p_n q_n=0,1}\cdots\sum_{p_1 q_1=0,1}(-1)^{\delta_{p_j q_j, 11}} P(x\phi_1^{p_1 q_1}\phi_2^{p_2 q_2}\dots\phi_n^{p_n q_n}),$$

for $x = 0, 1$, one can readily see that

$$C(x) = \text{Tr}[|\psi\rangle\langle\psi|(|x\rangle\langle x|^a \otimes \text{SWAP}_n)], \quad (39)$$

where SWAP_n is the SWAP operator that exchanges the state of the qubits in S_1 and S_2. An explicit evaluation leads to

$$C(0) = \frac{1}{4}(\langle\mathbf{0}|U^\dagger V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger U|\mathbf{0}\rangle$$
$$+ \langle\mathbf{0}|U^\dagger A_{l'} A_l V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger A_l^\dagger A_{l'}^\dagger U|\mathbf{0}\rangle \quad (40)$$
$$+ \text{Re}\left[\langle\mathbf{0}|U^\dagger A_l V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger A_{l'}^\dagger U|\mathbf{0}\rangle\right]).$$

and

$$C(1) = \frac{1}{4}(\langle\mathbf{0}|U^\dagger V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger U|\mathbf{0}\rangle$$
$$+ \langle\mathbf{0}|U^\dagger A_{l'} A_l V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger A_l^\dagger A_{l'}^\dagger U|\mathbf{0}\rangle \quad (41)$$
$$- \text{Re}\left[\langle\mathbf{0}|U^\dagger A_l V|\mathbf{0}\rangle\langle\mathbf{0}|V^\dagger A_{l'}^\dagger U|\mathbf{0}\rangle\right]).$$

Then, combining (40) and (41) yields

$$\text{Re}[\gamma_{ll'}] = 2(C(0) - C(1)). \quad (42)$$

Following a similar procedure, it can be shown that including the R_z gate allows us to calculate $\text{Im}[\gamma_{ll'}]$.

Note that the Hadamard-Overlap test can also be used to compute the real and imaginary parts of $\delta_{ll'}^{(j)}$ in (18). In this case an additional random unitary R_j must be initially applied to the qubits in register S_2 in order to generate the input state $|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}$. Specifically, R_j randomly applies a bit-flips to all qubits except qubit j:

$$R_j = X_1^{r_1} \otimes X_2^{r_2} \otimes \cdots \otimes \mathbb{1}_j^{r_j} \otimes \cdots \otimes X_n^{r_n}, \quad (43)$$

with $\boldsymbol{r} = r_1, r_2\dots, r_n$ a random bitstring of length n.

D Proof of Proposition 1

Here we prove Proposition 1 of the main text, which we restate for convenience.

Proposition 1. *The problem of estimating the VQLS cost functions* $\widehat{C}_G$, C_G, $\widehat{C}_L$, *or* C_L *to within precision* $\pm\delta = 1/\operatorname{poly}(n)$ *is* DQC1-*hard.*

Proof. Let us first show that estimating $\widehat{C}_G$ and C_G is DQC1-hard. Our proof is a reduction from the problem of estimating the Hilbert-Schmidt inner-product magnitude Δ_{HS} between two quantum circuits $\tilde{U}$ and $\tilde{V}$ acting on n-qubits [33], where we have defined

$$\Delta_{\mathrm{HS}}(\tilde{U}, \tilde{V}) := \frac{1}{d^2}|\operatorname{Tr}(\tilde{V}^\dagger \tilde{U})|^2, \qquad (44)$$

with $d = 2^n$.

In particular, let us consider the following specific case of estimating Δ_{HS}, which in turn can be identified as a specific instance of approximating the cost functions $\widehat{C}_G$ or C_G. Let $A = \mathbb{1}$, and let $|x\rangle$ and $|b\rangle$ be $2n$-qubit states given by

$$|x\rangle = V|\mathbf{0}\rangle = (\tilde{V} \otimes \mathbb{1})E|\mathbf{0}\rangle, \qquad (45)$$

$$|b\rangle = U|\mathbf{0}\rangle = (\tilde{U} \otimes \mathbb{1})E|\mathbf{0}\rangle, \qquad (46)$$

where E is an efficient unitary gate that produces a maximally entangled state (e.g., a depth-two circuit composed of Hadamard and CNOT gates). Note that here $|x\rangle$ and $|b\rangle$ correspond to the Choi states of $\tilde{V}$ and $\tilde{U}$, respectively. The global cost function is given by

$$C_G = \widehat{C}_G = 1 - |\langle x|b\rangle|^2 \qquad (47)$$

$$= 1 - |\langle \mathbf{0}|E^\dagger(\tilde{V}^\dagger\tilde{U} \otimes \mathbb{1})E|\mathbf{0}\rangle|^2 \qquad (48)$$

$$= 1 - \frac{1}{d^2}|\operatorname{Tr}(\tilde{V}^\dagger\tilde{U})|^2 = 1 - \Delta_{\mathrm{HS}}(\tilde{U}, \tilde{V}). \qquad (49)$$

Moreover, it is known that approximating Δ_{HS} to within inverse polynomial precision is DQC1-hard [33], and hence the result follows. We additionally remark that estimating Eq. (47) can also be interpreted as estimating the Fidelity between two pure states, which was also shown to be DQC1-hard [36].

We now show that estimating $\widehat{C}_L$ and C_L is also DQC1-hard. In this case our proof is reduced to the problem of approximating the trace of a unitary matrix W. Let $|x\rangle = V|\mathbf{0}\rangle$ and $|b\rangle = U|\mathbf{0}\rangle$ be $2n + 1$-qubit states and let $A = \mathbb{1}$. Moreover, as depicted in Fig. 10, let us denote the first qubit as S_1, while qubits $2, \ldots, n+1$ compose system S_2, and qubits $n+2, \ldots, 2n+1$ compose S_3. Let

$$U^\dagger V = H^{S_1} C_W^{S_1 S_2} H^{S_1} E^{S_2 S_3}, \qquad (50)$$

where H^{S_1} is a Hadamard gate acting on S_1, $C_W^{S_1 S_2}$ denotes a controlled-W gate with S_1 the control and S_2 the target, and E is a maximally-entangling gate on S_2 and S_3.
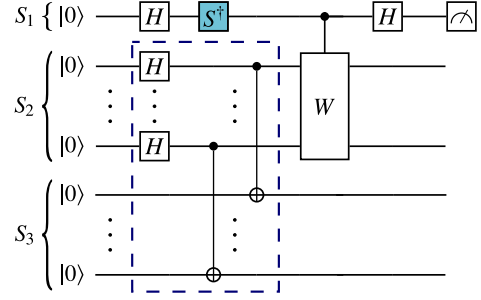

Figure 10: Schematic representation of the circuit used to compute C_L for the specific case when $A = \mathbb{1}$, and $|x\rangle = V|\mathbf{0}\rangle$, $|b\rangle = U|\mathbf{0}\rangle$ are $2n + 1$-qubit states such that $U^\dagger V$ is given by (50). Shown is the measurement of the qubit in S_1. The dashed box indicates the entangling gate E. The phase gate in the colored box is excluded when calculating the real part of $\operatorname{Tr} W$ and included when calculating the imaginary part.

The local cost is then

$$C_L = \widehat{C}_L$$
$$= 1 - \frac{1}{n}\sum_{j=1}^{n}\langle\mathbf{0}|V^\dagger U\left(|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}\right)U^\dagger V|\mathbf{0}\rangle. \qquad (51)$$

Consider first the case when $j = 1$ and we measure the qubit in S_1. It is straightforward to see that the probability of measuring this qubit in the $|0\rangle$ state is $P(0) = (1 + \operatorname{Re}(\operatorname{Tr} W))/2$. On the other hand, the probability $P(0) = 1/2$ for all qubits in S_2 or S_3. Hence, we find

$$C_L = \frac{n+1}{2n+1} - \frac{1 + \operatorname{Re}(\operatorname{Tr} W)}{4n+2}, \qquad (52)$$

which implies that the real part of $\operatorname{Tr} W$ can be computed as

$$\operatorname{Re}(\operatorname{Tr} W) = \frac{n+1}{2} - (4n+2)C_L - 1. \qquad (53)$$

Similarly, by adding a phase gate on S_1 (as indicated in Fig. 10) we can compute the imaginary part of $\operatorname{Tr} W$.

By choosing U and V according to (50) one finds that the problem of estimating the local cost function up to inverse polynomial precision is equivalent to approximating the real (or imaginary) part of $\operatorname{Tr} W$. Hence, computing C_L or $\widehat{C}_L$ is hard for DQC1, since all problems in DQC1 can be seen as estimating the real part of a trace of a unitary matrix [33]. $\qquad\square$

E Variable ansatz optimization

Here we discuss the optimization method employed for the heuristics in Section 2.2.2. As mentioned in the main text, we employed a variable-structure ansatz where the gate placement and the type of

gates in $V(\boldsymbol{\alpha})$ can change during the optimization. Our approach here is similar to the variable-structure ansatzes employed in Refs. [30, 47].

First, the gate structure and the angles of $V(\boldsymbol{\alpha})$ are randomly initialized. That is, one randomly chooses $\boldsymbol{k}$, and $\boldsymbol{\alpha}$ in (19). Then, the optimization is performed in two alternating loops: an inner loop and an outer loop. During the inner loop, $\boldsymbol{k}$ is fixed and one optimizes over $\boldsymbol{\alpha}$. Once a local minima is reached, the circuit layout is changed in the outer optimization loop. In this outer loop, the circuit is randomly grown by inserting into $V(\boldsymbol{\theta})$ a sequence of parametrized gates which compile to identity, such that they do not change the cost value. The previous process is then repeated by alternating between the inner and outer loops until the optimization termination condition is met.

Here we remark that the goal of the outer loop is to enhance the expressivity of $V(\boldsymbol{\alpha})$ and lead to smaller cost values during the next inner loop. However, it may happen that after growing the circuit, the optimizer is not able to minimize the cost function. This is due to the fact that some gate insertions do not lead to more expressive circuits. In order to avoid such unnecessary circuit growth, one can then accept the parametrized gate insertion conditioned to leading to smaller cost values.

F Gradient-based optimization

Here we derive analytical expressions for the gradients of our cost functions, and we show that these gradients can be computed with the same circuits as those introduced in Section C. This enables one to take a gradient-descent optimization approach to VQLS, and is inspired by gradient-based approaches in previous work [33, 57] for other VHQCAs.

For simplicity, let us first consider the global cost functions. Let us recall from (19) that the trainable unitary $V(\boldsymbol{\alpha})$ can be expressed as a sequence of gates $G(\alpha_i)$, where we have dropped the subscript on G. In turn, each gate $G(\alpha_i)$ can always be parametrized by a single-qubit rotation angle of the form $e^{-i\alpha_i\sigma_i/2}$. The gradient of $V(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ is then given by

$$\nabla_{\boldsymbol{\alpha}} V(\boldsymbol{\alpha}) = \left(\frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_1}, \dots, \frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_D} \right), \quad (54)$$

and each partial derivative is

$$\frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_i} = G(\alpha_L) \dots \frac{\partial G(\alpha_i)}{\partial \alpha_i} \dots G(\alpha_1)$$
$$= -\frac{i}{2} G(\alpha_L) \dots \sigma_i G(\alpha_i) \dots G(\alpha_1). \quad (55)$$

As we now show, the gradient of $\beta_{ll'}$ and $\gamma_{ll'}$ can be computed with the Hadamard-Overlap test and the Hadamard test, respectively. Let us consider the partial derivatives

$$\frac{\partial \gamma_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} = \langle\boldsymbol{0}|U^\dagger A_l \frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_i}|\boldsymbol{0}\rangle\langle\boldsymbol{0}|V(\boldsymbol{\alpha})^\dagger (A_{l'})^\dagger U|\boldsymbol{0}\rangle$$
$$+ \langle\boldsymbol{0}|U^\dagger A_l V(\boldsymbol{\alpha})|\boldsymbol{0}\rangle\langle\boldsymbol{0}|\frac{\partial V(\boldsymbol{\alpha})^\dagger}{\partial \alpha_i}(A_{l'})^\dagger U|\boldsymbol{0}\rangle,$$
$$(56)$$

$$\frac{\partial \beta_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} = \langle\boldsymbol{0}|\frac{\partial V(\boldsymbol{\alpha})^\dagger}{\partial \alpha_i} A_l (A_{l'})^\dagger V(\boldsymbol{\alpha})|\boldsymbol{0}\rangle$$
$$+ \langle\boldsymbol{0}|V(\boldsymbol{\alpha})^\dagger A_l (A_{l'})^\dagger \frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_i}|\boldsymbol{0}\rangle. \quad (57)$$

By means of the identity $i[\sigma_i, A] = G_i(-\frac{\pi}{2})A G_i(-\frac{\pi}{2})^\dagger - G_i(\frac{\pi}{2})A G_i(\frac{\pi}{2})^\dagger$, which is valid for any matrix A, we combine Eqs. (55)–(57) to obtain

$$\frac{\partial \gamma_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} = \frac{1}{2}(\langle\boldsymbol{0}|U^\dagger A_l V_+^i(\boldsymbol{\alpha})|\boldsymbol{0}\rangle\langle\boldsymbol{0}|V_+^i(\boldsymbol{\alpha})^\dagger (A_{l'})^\dagger U|\boldsymbol{0}\rangle$$
$$- \langle\boldsymbol{0}|U^\dagger A_l V_-^i(\boldsymbol{\alpha})|\boldsymbol{0}\rangle\langle\boldsymbol{0}|V_-^i(\boldsymbol{\alpha})^\dagger (A_{l'})^\dagger U|\boldsymbol{0}\rangle),$$
$$(58)$$

$$\frac{\partial \beta_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} = \frac{1}{2}(\langle\boldsymbol{0}|V_+^i(\boldsymbol{\alpha})^\dagger A_l (A_{l'})^\dagger V_+^i(\boldsymbol{\alpha})|\boldsymbol{0}\rangle$$
$$- \langle\boldsymbol{0}|V_-^i(\boldsymbol{\alpha})^\dagger A_l (A_{l'})^\dagger V_-^i(\boldsymbol{\alpha})|\boldsymbol{0}\rangle), \quad (59)$$

where we have defined

$$V_\pm^i(\boldsymbol{\alpha}) = G(\alpha_L) \dots G(\alpha_i \pm \frac{\pi}{2}) \dots G(\alpha_1). \quad (60)$$

Each term in (58) can be computed by means of the Hadamard-Overlap test, while the terms in (59) can be determined via the Hadamard test. These results imply that the gradient with respect to $\boldsymbol{\alpha}$ of C_G and $\widehat{C}_G$ are determined by

$$\frac{\partial C_G}{\partial \alpha_i} = -\frac{\sum_{l,l'm,m'} c_l c_{l'}^* c_m c_{m'}^* \frac{\partial \gamma_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} \beta_{mm'}(\boldsymbol{\alpha})}{\left(\sum_{l,l'=1}^L c_l c_{l'}^* \beta_{ll'}(\boldsymbol{\alpha})\right)^2}$$
$$+ \frac{\sum_{l,l'm,m'} c_l c_{l'}^* c_m c_{m'}^* \gamma_{ll'}(\boldsymbol{\alpha}) \frac{\partial \beta_{mm'}(\boldsymbol{\alpha})}{\partial \alpha_i}}{\left(\sum_{l,l'=1}^L c_l c_{l'}^* \beta_{ll'}(\boldsymbol{\alpha})\right)^2}, \quad (61)$$

and

$$\frac{\partial \widehat{C}_G}{\partial \alpha_i} = \sum_{l,l'=1}^L c_l c_{l'}^* \left(\frac{\partial \beta_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} - \frac{\partial \gamma_{ll'}(\boldsymbol{\alpha})}{\partial \alpha_i} \right), \quad (62)$$

and hence that they can be computed by means of the Hadamard-Overlap test and the Hadamard test.

A similar derivation can be used with

$$\frac{\partial \delta_{ll'}^{(j)}}{\partial \alpha_i} = \langle\boldsymbol{0}|\frac{\partial V(\boldsymbol{\alpha})^\dagger}{\partial \alpha_i} A_{l'}^\dagger U(|0_j\rangle\langle0_j| \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\boldsymbol{0}\rangle$$
$$+ \langle\boldsymbol{0}|V^\dagger A_{l'}^\dagger U(|0_j\rangle\langle0_j| \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l \frac{\partial V(\boldsymbol{\alpha})}{\partial \alpha_i}|\boldsymbol{0}\rangle. \quad (63)$$

Hence the gradient of the local cost functions, $\widehat{C}_L$ and C_L, can also be computed with the circuits in Sec. C.

G Resilience to simple noise models

Here we show that the normalized cost functions C_G and C_L are resilient to certain kinds of noise. In particular, we consider global depolarizing noise and measurement noise, which are relatively simple noise models. We leave the analysis of more complicated noise models for future work.

For noise resilience, we employ the definition in Ref. [60], where they define Optimal Parameter Resilience (OPR). A cost function exhibits OPR to a given noise model if this noise does not shift the global minima in parameter space, i.e., if the optimal parameters are not changed by the noise.

OPR is very important to the success of VQLS for the following reason. If VQLS exhibits OPR, then that means that optimizing the noisy cost function can still lead to the correct optimal parameters. Then, using these correct parameters, one can prepare the state $|x\rangle$ corresponding to the correct solution of the QLSP. One can then use standard error mitigation techniques, such as zero-noise extrapolation, to obtain noise-free estimates of observable expectation values of the form $\langle x|O|x\rangle$. Hence, OPR during the optimization process combined with error mitigation after the optimization is over can potentially lead to accurate estimation of observables on the true solution of the QLSP.

G.1 Global Cost

Let us write C_G as:

$$C_G = 1 - \frac{\nu}{\mu} \qquad (64)$$

where $\nu = |\langle\psi|b\rangle|^2$ and $\mu = \langle\psi|\psi\rangle$. For the noisy versions of these quantities, we write $\tilde{C}_G$, $\tilde{\nu}$, and $\tilde{\mu}$. Let us first consider global depolarizing noise, which transforms the state according to $\rho \to q\rho + (1-q)\mathbb{1}/d$. Note that if a circuit has L layers, with noise acting after each layer, then the final state is $q^L\rho + (1-q^L)\mathbb{1}/d$.

First consider $\tilde{\mu}$, which is estimated by the circuit in Fig. 9(a). We assume noise affects the estimation of the real and imaginary parts of $\tilde{\beta}_{ll'}$ to an equal extent, where $\tilde{\beta}_{ll'}$ is the noisy version of $\beta_{ll'}$. The maximally mixed state has zero expectation value for the circuit in Fig. 9(a), which measures the Pauli Z operator on the ancilla. Therefore, we obtain that $\tilde{\beta}_{ll'} = q^{L(\beta_{ll'})}\beta_{ll'}$. Here, $L(\beta_{ll'})$ is the number of layers in the circuit used to estimate $\beta_{ll'}$. Hence, we have $\tilde{\mu} = \sum_{ll'} c_l c_{l'}^* q^{L(\beta_{ll'})}\beta_{ll'}$. Finally, one can make an additional assumption that $L(\beta_{ll'}) = L_\beta$ is independent of l and l', which is a reasonable approximation since the depth of the circuit will likely be dominated by the ansatz V rather than by the controlled gates in Fig. 9(a). This gives:

$$\tilde{\mu} = q^{L_\beta}\mu. \qquad (65)$$

Next consider $\tilde{\nu}$, which we assume is estimated by the circuit in Fig. 9(b). We assume noise affects the estimation of the real and imaginary parts of $\tilde{\gamma}_{ll'}$ to an equal extent, where $\tilde{\gamma}_{ll'}$ is the noisy version of $\gamma_{ll'}$. The maximally mixed state has zero expectation value for the circuit in Fig. 9(b), which measures the Pauli Z operator on the ancilla. Therefore, we obtain that $\tilde{\gamma}_{ll'} = q^{L(\gamma_{ll'})}\gamma_{ll'}$. Here, $L(\gamma_{ll'})$ is the number of layers in the circuit used to estimate $\gamma_{ll'}$. Hence, we have $\tilde{\nu} = \sum_{ll'} c_l c_{l'}^* q^{L(\gamma_{ll'})}\gamma_{ll'}$. Finally, one can make an additional assumption that $L(\gamma_{ll'}) = L_\gamma$ is independent of l and l', which is a reasonable approximation since the depth of the circuit will likely be dominated by the ansatz V rather than by the controlled gates in Fig. 9(b). This gives:

$$\tilde{\nu} = q^{L_\gamma}\nu. \qquad (66)$$

Combining (65) and (66) gives

$$\tilde{C}_G = 1 - \frac{q^{L_\gamma}}{q^{L_\beta}}\frac{\nu}{\mu}. \qquad (67)$$

From this expression, we see that

$$\arg\min_{\boldsymbol{\alpha}} \tilde{C}_G = \arg\max_{\boldsymbol{\alpha}} \left(\frac{\nu}{\mu}\right). \qquad (68)$$

Furthermore, it is clear that

$$\arg\min_{\boldsymbol{\alpha}} C_G = \arg\max_{\boldsymbol{\alpha}} \left(\frac{\nu}{\mu}\right). \qquad (69)$$

Hence we arrive at

$$\arg\min_{\boldsymbol{\alpha}} \tilde{C}_G = \arg\min_{\boldsymbol{\alpha}} C_G. \qquad (70)$$

This proves our desired statement of OPR, since it shows that the optimal parameters are unaffected by the noise.

G.2 Local Cost

G.2.1 Global depolarizing noise

Now consider the local cost C_L. We can write

$$C_L = 1 - \frac{\omega}{\mu}, \qquad (71)$$

where $\omega = \frac{1}{n}\sum_{jll'} c_l c_{l'}^* \delta_{ll'}^{(j)}$. Let $\tilde{C}_L$, $\tilde{\omega}$, and $\tilde{\delta}_{ll'}^{(j)}$ denote the noisy versions of these quantities.

Recall that we can expand $\delta_{ll'}^{(j)}$ according to (37) as $\delta_{ll'}^{(j)} = \beta_{ll'} + \zeta_{ll'}^{(j)}$, where $\zeta_{ll'}^{(j)} = \langle\mathbf{0}|V^\dagger A_{l'}^\dagger U(Z_j \otimes \mathbb{1}_{\bar{j}})U^\dagger A_l V|\mathbf{0}\rangle$. Also, recall that we have $\tilde{\beta}_{ll'} = q^{L_\beta}\beta_{ll'}$ under the same assumptions as those considered above. This gives

$$\tilde{\omega} = \frac{1}{n}\sum_{jll'} c_l c_{l'}^* (q^{L_\beta}\beta_{ll'} + \tilde{\zeta}_{ll'}^{(j)}), \qquad (72)$$

where $\tilde{\zeta}_{ll'}^{(j)}$ is the noisy version of $\zeta_{ll'}^{(j)}$. Note that $\tilde{\zeta}_{ll'}^{(j)}$ is estimated by the circuit in Fig. 9(c). We assume noise affects the estimation of the real and imaginary parts of $\tilde{\zeta}_{ll'}^{(j)}$ to an equal extent. The maximally mixed state has zero expectation value for the circuit in Fig. 9(c), which measures the Pauli Z operator on the ancilla. Therefore, we obtain that $\tilde{\zeta}_{ll'}^{(j)} = q^{L(\zeta_{ll'}^{(j)})} \zeta_{ll'}^{(j)}$. Here, $L(\zeta_{ll'}^{(j)})$ is the number of layers in the circuit used to estimate $\zeta_{ll'}^{(j)}$. Finally, one can make an additional assumption that $L(\zeta_{ll'}^{(j)}) = L_\zeta$ is independent of l, l', and j, which is a reasonable approximation since the depth of the circuit will likely be dominated by either the ansatz V or the unitary U, rather than by the controlled gates in Fig. 9(c). This gives $\tilde{\zeta}_{ll'}^{(j)} = q^{L_\zeta} \zeta_{ll'}^{(j)}$, which implies

$$\tilde{\omega} = q^{L_\beta} \frac{1}{n} \sum_{jll'} c_l c_{l'}^* (\beta_{ll'} + \frac{q^{L_\zeta}}{q^{L_\beta}} \zeta_{ll'}^{(j)})$$

$$= q^{L_\beta} \frac{1}{n} \sum_j \text{Tr}(|\chi\rangle\langle\chi|(\rho_j \otimes \mathbb{1}_{\bar{j}})), \qquad (73)$$

where $|\chi\rangle = U^\dagger A |x\rangle$ is an unnormalized state, and $\rho_j = \frac{1}{2}(\mathbb{1}_j + \frac{q^{L_\zeta}}{q^{L_\beta}} Z_j)$ is a Hermitian (but not necessarily positive semidefinite) matrix. Note that we have $\tilde{\mu} = q^{L_\beta} \mu$, which gives:

$$\tilde{C}_L = 1 - \frac{1}{\mu} \frac{1}{n} \sum_j \text{Tr}(|\chi\rangle\langle\chi|(\rho_j \otimes \mathbb{1}_{\bar{j}}))$$

$$= 1 - \frac{1}{n} \sum_j \text{Tr}(|\hat{\chi}\rangle\langle\hat{\chi}|(\rho_j \otimes \mathbb{1}_{\bar{j}})), \qquad (74)$$

where $|\hat{\chi}\rangle = |\chi\rangle/\sqrt{\mu}$ is a normalized state.

Let us now argue that $\tilde{C}_L$ is a faithful cost function, reaching its minimum value iff $A|x\rangle \propto |b\rangle$. Let us first note that the minimum value of the cost can be found as follows. Note that $\text{Tr}(|\chi\rangle\langle\chi|(\rho_j \otimes \mathbb{1}_{\bar{j}}))$ is not larger than the largest eigenvalue of $(\rho_j \otimes \mathbb{1}_{\bar{j}})$, and this eigenvalue is $(1 + q^{L_\zeta}/q^{L_\beta})/2$. Hence, we have

$$\tilde{C}_L \geqslant (1 - q^{L_\zeta}/q^{L_\beta})/2 \,. \qquad (75)$$

The lower bound here is actually achievable and hence corresponds to the minimum value of $\tilde{C}_L$. Let us denote this minimum value as $\tilde{C}_L^{\min}$.

Namely, if we assume that $|x\rangle$ is a solution to the QLSP, then we have $A|x\rangle \propto |b\rangle$, which implies $|\chi\rangle \propto |\mathbf{0}\rangle$ and $|\hat{\chi}\rangle = |\mathbf{0}\rangle$. In turn, this implies $\text{Tr}(|\chi\rangle\langle\chi|(\rho_j \otimes \mathbb{1}_{\bar{j}})) = (1 + q^{L_\zeta}/q^{L_\beta})/2$ and hence $\tilde{C}_L = \tilde{C}_L^{\min}$.

Conversely, if we assume that $\tilde{C}_L = \tilde{C}_L^{\min}$, then this implies that $|\hat{\chi}\rangle$ must be an eigenvector of $(\rho_j \otimes \mathbb{1}_{\bar{j}})$ for all j, with an eigenvalue of $(1 + q^{L_\zeta}/q^{L_\beta})/2$. This implies that $\text{Tr}_{\bar{j}}(|\hat{\chi}\rangle\langle\hat{\chi}|) = |0\rangle\langle 0|$ for all j, i.e., that $|\hat{\chi}\rangle$ is locally the zero state on each qubit. The only state that satisfies this is $|\hat{\chi}\rangle = |\mathbf{0}\rangle$. This state corresponds to $A|x\rangle \propto |b\rangle$, and hence the QLSP is solved in this case.

This completes the argument that $\tilde{C}_L$ is a faithful cost function. Note that this faithfulness property corresponds to OPR, assuming the ansatz $V(\boldsymbol{\alpha})$ is capable of expressing the true solution. Hence we have proven OPR for $\tilde{C}_L$ under global depolarizing noise.

G.2.2 Measurement noise

Finally, we remark that the above proof has implications for measurement noise as well. Measurement noise that is symmetric with respect to the 0 and 1 outcomes can be modeled as a local depolarizing channel acting on the state just prior to the measurement. Moreover, if only a single qubit is being measured, then this local depolarizing channel can be replaced by a global depolarizing channel. We remark that this is the case for the circuits in Fig. 9(a) and (c), which are the circuits used to compute the local cost $\tilde{C}_L$. Hence, for this local cost, any symmetric measurement noise can mathematically be viewed as an additional global depolarizing channel acting just prior to measurement. This means that $\tilde{C}_L$ exhibits OPR to a noise model that includes both global depolarizing as well as symmetric measurement noise.

Acknowledgements

References

[1] E. Alpaydin, *Introduction to Machine Learning*, 4th ed. (The MIT Press, 2020).

[2] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

[3] L. C. Evans, *Partial differential equations* (American Mathematical Society, 2010).

[4] O. Bretscher, *Linear Algebra With Applications, 5th ed.* (Pearson, 2013).

[5] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," SIAM J. Comput. **40**, 1913–1926 (2011).

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. **103**, 150502 (2009).

[7] A. Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations," arXiv:1010.4458 [quant-ph] .

[8] Y. Subaşı, R. D. Somma, and D. Orsucci, "Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing," Phys. Rev. Lett. **122**, 060504 (2019).

[9] A. Childs, R. Kothari, and R. Somma, "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision," SIAM J. Computing **46**, 1920–1950 (2017).

[10] S. Chakraborty, A. Gilyén, and S. Jeffery, "The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation," in *46th International Colloquium on Automata, Languages, and Programming* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019) pp. 33:1-33:14.

[11] L. Wossnig, Z. Zhao, and A. Prakash, "Quantum linear system algorithm for dense matrices," Phys. Rev. Lett. **120**, 050502 (2018).

[12] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum **2**, 79 (2018).

[13] Y. Zheng, C. Song, M.-C. Chen, B. Xia, W. Liu, *et al.*, "Solving systems of linear equations with a superconducting quantum processor," Phys. Rev. Lett. **118**, 210504 (2017).

[14] Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on IBM quantum experience," Scientific Reports **9**, 4778 (2019).

[15] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, *et al.*, "Experimental realization of quantum algorithm for solving linear systems of equations," Phys. Rev. A **89**, 022313 (2014).

[16] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, Mile Gu, *et al.*, "Experimental quantum computing to solve systems of linear equations," Phys. Rev. Lett. **110**, 230501 (2013).

[17] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić, *et al.*, "A two-qubit photonic quantum processor and its application to solving systems of linear equations," Scientific Reports **4**, 6115 (2014).

[18] J. Wen, X. Kong, S. Wei, B. Wang, T. Xin, and G. Long, "Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing," Phys. Rev. A **99**, 012320 (2019).

[19] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, "Variational quantum factoring," in *International Workshop on Quantum Technology and Optimization Problems* (Springer, 2019) pp. 74–85.

[20] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," Nature Communications **5**, 4213 (2014).

[21] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, *et al.*, "Quantum chemistry in the age of quantum computing," Chemical Reviews **119**, 10856–10915 (2019).

[22] O. Higgott, D. Wang, and S. Brierley, "Variational Quantum Computation of Excited States," Quantum **3**, 156 (2019).

[23] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, "Variational quantum algorithms for discovering Hamiltonian spectra," Phys. Rev. A **99**, 062304 (2019).

[24] Y. Li and S. C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," Phys. Rev. X **7**, 021050 (2017).

[25] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, "Self-verifying variational quantum simulation of lattice models," Nature **569**, 355–360 (2019).

[26] K. Heya, K. M. Nakanishi, K. Mitarai, and K. Fujii, "Subspace variational quantum simulator," Phys. Rev. Research **5**, 023078 (2023).

[27] Cristina Cirstoiu, Zoe Holmes, Joseph Iosue, Lukasz Cincio, Patrick J Coles, and Andrew Sornborger, "Variational fast forwarding for quantum simulation beyond the coherence time," npj Quantum Information **6**, 82 (2020).

[28] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin, "Theory of variational quantum simulation," Quantum **3**, 191 (2019).

[29] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," Quantum Science and Technology **2**, 045001 (2017).

[30] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles, "Variational quantum state diagonalization," npj Quantum Information **5**, 57 (2018).

[31] C. Bravo-Prieto, D. García-Martín, and J. I. Latorre, "Quantum Singular Value Decomposer," Phys. Rev. A **101**, 062310 (2020).

[32] M. Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J Coles, "Variational quantum state eigensolver," npj Quantum Information **8**, 113 (2022).

[33] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, "Quantum-

assisted quantum compiling," Quantum **3**, 140 (2019).

[34] T. Jones and S. C Benjamin, "Robust quantum compilation and circuit optimisation via energy minimisation," Quantum **6**, 628 (2022).

[35] A. Arrasmith, L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles, "Variational consistent histories as a hybrid algorithm for quantum foundations," Nature communications **10**, 3438 (2019).

[36] Marco Cerezo, Alexander Poremba, Lukasz Cincio, and Patrick J Coles, "Variational quantum fidelity estimation," Quantum **4**, 248 (2020).

[37] Bálint Koczor, Suguru Endo, Tyson Jones, Yuichiro Matsuzaki, and Simon C Benjamin, "Variational-state quantum metrology," New Journal of Physics **22**, 083038 (2020).

[38] M Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," Nature Communications **12**, 1791 (2020).

[39] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, New York, NY, USA, 2011).

[40] E. Knill and R. Laflamme, "Power of one bit of quantum information," Phys. Rev. Lett. **81**, 5672–5675 (1998).

[41] K. Fujii, H. Kobayashi, T. Morimae, H. Nishimura, S. Tamate, and S. Tani, "Impossibility of Classically Simulating One-Clean-Qubit Model with Multiplicative Error," Phys. Rev. Lett. **120**, 200502 (2018).

[42] T. Morimae, "Hardness of classically sampling the one-clean-qubit model with constant total variation distance error," Phys. Rev. A **96**, 040302 (2017).

[43] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature **549**, 242 (2017).

[44] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nature communications **9**, 4812 (2018).

[45] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti, "An initialization strategy for addressing barren plateaus in parametrized quantum circuits," Quantum **3**, 214 (2019).

[46] Tyler Volkoff and Patrick J Coles, "Large gradients via correlation in random parameterized quantum circuits," Quantum Sci. Technol. **6**, 025008 (2021).

[47] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, "Learning the quantum algorithm for state overlap," New Journal of Physics **20**, 113022 (2018).

[48] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," arXiv:1411.4028 [quant-ph] .

[49] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," Algorithms **12**, 34 (2019).

[50] S. Lloyd, "Quantum approximate optimization is computationally universal," arXiv:1812.11075 [quant-ph] .

[51] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for MaxCut: A fermionic view," Phys. Rev. A **97**, 022304 (2018).

[52] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices," Phys. Rev. X **10**, 021067 (2020).

[53] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," arXiv preprint arXiv:1811.08419 (2018) .

[54] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, "An adaptive optimizer for measurement-frugal variational algorithms," Quantum **4**, 263 (2020).

[55] Andrew Arrasmith, Lukasz Cincio, Rolando D Somma, and Patrick J Coles, "Operator sampling for shot-frugal optimization in variational algorithms," arXiv preprint arXiv:2004.06252 (2020).

[56] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert, "Stochastic gradient descent for hybrid quantum-classical optimization," Quantum **4**, 314 (2020).

[57] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Phys. Rev. A **98**, 032309 (2018).

[58] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," Phys. Rev. A **99**, 032331 (2019).

[59] A. Harrow and J. Napp, "Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms," Phys. Rev. Lett. **126**, 140502 (2021).

[60] Kunal Sharma, Sumeet Khatri, Marco Cerezo, and Patrick Coles, "Noise resilience of variational quantum compiling," New Journal of Physics **22**, 043006 (2020).

[61] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," Phys. Rev. Lett. **119**, 180509 (2017).

[62] Y. He and H. Guo, "The boundary effects of transverse field ising model," Journal of Statistical Mechanics: Theory and Experiment **2017**, 093101 (2017).

[63] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," Communications in Mathematical Physics **270**, 359–371 (2007).

[64] Y. Atia and D. Aharonov, "Fast-forwarding of hamiltonians and exponentially precise measurements," Nature communications **8**, 1572 (2017).

[65] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, "Variational algorithms for linear algebra," Science Bulletin **66**, 2181–2188 (2021).

[66] H.-Y. Huang, K. Bharti, and P. Rebentrost, "Near-term quantum algorithms for linear systems of equations with regression loss functions," New Journal of Physics **23**, 113021 (2021).

[67] A. Asfaw, L. Bello, Y. Ben-Haim, S. Bravyi, L. Capelluto, *et al.*, "Learn quantum computation using qiskit." (2019).

[68] A. Mari, "Variational quantum linear solver." (2019).

[69] M. Szegedy, "Quantum speed-up of markov chain based algorithms," in *Proceedings of the 45th Annual IEEE Symposium on FOCS.* (IEEE, 2004) pp. 32–41.

[70] D. W. Berry, A. M. Childs, and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *Proceedings of the 56th Symposium on Foundations of Computer Science* (2015).

[71] J. C. Garcia-Escartin and P. Chamorro-Posada, "Swap test and Hong-Ou-Mandel effect are equivalent," Phys. Rev. A **87**, 052330 (2013).

[72] M. J. D. Powell, "A fast algorithm for nonlinearly constrained optimization calculations," in *Numerical analysis* (Springer, 1978) pp. 144–157.

Accepted in 〈Quantum 2023-11-15, click title to verify. Published under CC-BY 4.0.

19

Supplemental Material for
"Variational Quantum Linear Solver"

A   Supplementary Note 1 – Additional Implementations

A.1   Scaling Heuristics for 8×8 Systems

Here we study the scaling of the VQLS algorithm for 8×8 systems. For this purpose we employed the ansatz of Fig. S1 (with randomly initialized parameters) and we numerically implement VQLS to solve the three different QLSPs with different degeneracy g in the minimum eigenvalue of A defined by:

$$A_1 = \frac{1}{8\kappa}[4(\kappa+1)\mathbb{1} + (\kappa-1)(Z_3 + Z_2 + 2Z_1)] \tag{S76}$$

$$A_2 = \frac{1}{4\kappa}[2(\kappa+1)\mathbb{1} + (\kappa-1)(Z_3 + Z_2)] \tag{S77}$$

$$A_3 = \frac{1}{2\kappa}[(\kappa+1)\mathbb{1} + (\kappa-1)Z_3] . \tag{S78}$$

The degeneracy of each matrix is $g = 1, 2, 4$, respectively. We remark that we considered different values of g to analyze if this parameter affects the VQLS performance. The state $|b\rangle$ is

$$|b\rangle = H^{\otimes 3}|000\rangle . \tag{S79}$$

Figure S2 shows results plotting time-to-solution versus κ for the aforementioned A matrices by training $\widehat{C}_G$, C_G, $\widehat{C}_L$, and C_L. In all cases we employed the operational meanings of our cost functions in (9) and (10) of the main text for our certification procedure, i.e., to upper-bound the quantity ϵ. For each data point in Fig. S2, we implemented and averaged over 1000 runs of VQLS.

These results show that the κ scaling is efficient for the problems considered (regardless of the value of g considered). This is in agreement with the scaling observed in the main text. It is worth noting that this efficient scaling holds for all of our cost functions. The unnormalized cost functions have slightly better performance for the A_1 and A_2 matrices, although all four cost functions perform similarly for A_3, indicating that the performances of different cost functions is problem dependent.

A.2   Implementation with QAOA Ansatz

Here we numerically analyze the VQLS scaling with κ when employing the QAOA ansatz. Since poorly conditioned matrices (i.e., large κ) are more difficult to invert, we expect that for fixed ϵ the number of layers p must increase with κ. While this is generally true, we can also alleviate this issue by evolving with the driver Hamiltonian H_D for a longer time. This corresponds to scaling the parameters α_i for odd i
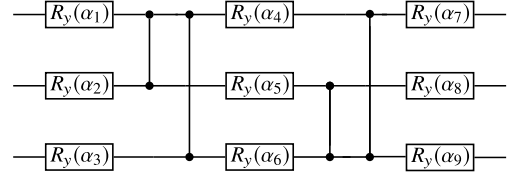
Supplementary Figure S1: Hardware-Efficient Ansatz used to solve the QLSPs in (S76)–(S79). Since A and $|b\rangle$ are real, $V(\boldsymbol{\alpha})$ contains only rotation around the y-axis $R_y(\alpha_i) = e^{-i\alpha_i Y/2}$, and control-$Z$ gates.

in (20) of the main text by some value that grows with κ. As shown in Fig. S3(a) and (b), this scaling can indeed transform the cost landscape such that it contains more regions of low cost and thus makes optimization more likely to be successful.

In Fig. S3(c), we show the time-to-solution versus the condition number. Here, we consider the QLSP on three qubits defined by the A_2 matrix of (S77) and with $|b\rangle$ given by (S79). For this small scale-implementation we obtained the time-to-solution be exactly computing ϵ. The condition number was varied from $\kappa = 10^0$ to $\kappa = 10^3$. For each κ, VQLS was implemented 100 times with the parameters randomly initialized. For each of the three values of ϵ considered, the scaling with κ is sub-exponential. Hence, these results indicate that VQLS with QAOA also scales efficiently in the condition number κ. Finally, we emphasize that these results were obtained with only $p = 1$ round of QAOA, and remark that additional rounds $p > 1$ may lead to better performance.

A.3   Implementations on Rigetti's quantum computer

Here we present additional implementations performed on Rigetti's quantum device *16Q Aspen-4*. We have considered different problem sizes, from 2×2 up to 32×32. We additionally recall that the matrices A and states $|b\rangle$ in these QLSP are such that the ansatz and the cost computing circuits are simplified.

First we present the results of a 32×32 (i.e., 5-qubit) implementation of VQLS using Rigetti's quantum chip *16Q Aspen-4*. We considered the QLSP defined by

$$A = \mathbb{1} + 0.2X_1Z_2 + 0.2X_1 , \tag{S80}$$

and $|b\rangle = H_1H_3H_4H_5|0\rangle^{\otimes 5}$. This particular choice of A and $|b\rangle$ is motivated from the fact that they lead to simplified ansatz and cost evaluation circuits. In particular, the ansatz considered consists of $R_y(\alpha_i)$ gates acting on each qubit.

The results are shown in Fig. S4. At each run of the VQLS algorithm the parameters were initialized to random angles, and the classical optimization
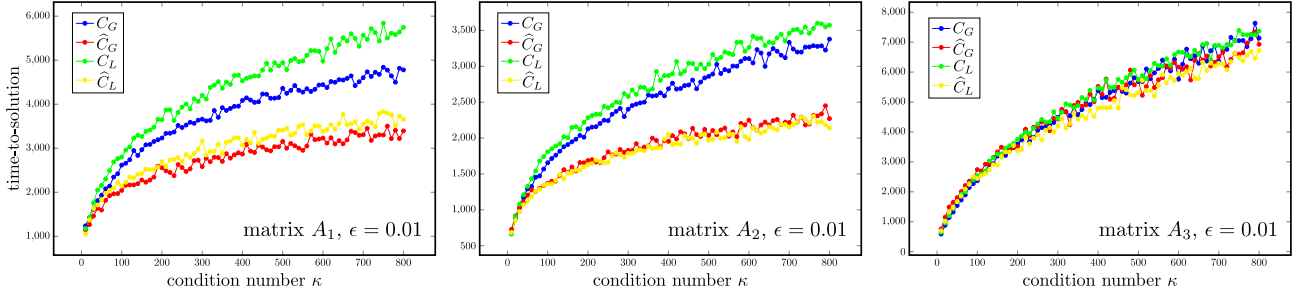
Supplementary Figure S2: Time-to-solution versus condition number κ. The time-to-solution is the mean number of executions needed to guarantee a desired precision ϵ. The QLSP is determined by $|b\rangle$ of (S79), and A given by: *Left:* Matrix A_1 of (S76). *Center:* Matrix A_2 of (S77). *Right:* Matrix A_3 of (S78). For each data point we ran and averaged 1000 instances of the VQLS algorithm. In all cases we trained the gate sequence by minimizing $\widehat{C}_G$, C_G, $\widehat{C}_L$, and C_L. As can be seen, the scaling in terms of the condition number κ appears to be efficient for all A matrices and for all cost functions.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z	0	0.04 ± 0.02	0.002 ± 0.002

Supplementary Table S1: Expectation value of an observable M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{2\times 2} = H$, and $|b\rangle = X|0\rangle$.

was performed with the Powell method [72]. For every run, the local cost function C_L of (6) achieved a value of $\sim 7 \times 10^{-2}$ (hardware noise prevented further cost reduction). While this cost value led to a trivial bound on ϵ via (9), we nevertheless found the solution $|x\rangle$ to be of high quality. We verified this by measuring the expectation value of different Hermitian observables M on the state $|x\rangle$ prepared on the quantum computer. According to (11), we can use $D(M)^2$ as a figure of merit to quantify the quality of our solution. For all M we considered, $D(M)^2$ was no larger than 0.01, and hence the results have a good agreement with the exact solution. See Table S7 for all values of $D(M)^2$.

Figure S5 shows the value of the cost function versus the number of optimization steps for different linear systems and for several runs. It is worth mentioning that the cost function is reduced to values $\lesssim 0.1$ for every example, except for the case depicted in panel (b). In this particular case, the solution of the 2×2 linear system is $|x_0\rangle = |1\rangle$. Therefore, one may note the effect of relaxation to the state $|0\rangle$ in the quantum device, which likely significantly affected the result quality. The Tables S1, S2, S3, S4, S5 and S6 correspond to the examples shown in Figure S5. In the tables we show the expectation values of several observables M, obtained from the output of the VQLS and we compare them to the exact ones.
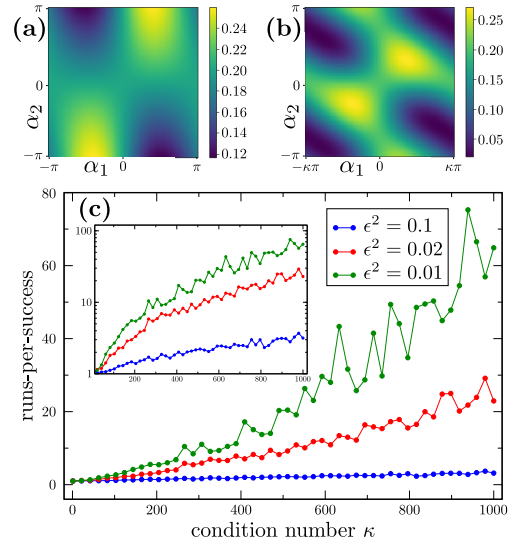
Supplementary Figure S3: (a) Landscape for $\widehat{C}_G$ with a QAOA ansatz of $p = 1$ layer and unscaled parameters α_1 and α_2. Here, α_1 (α_2) corresponds the the parameter in the driver (mixer) Hamiltonian. (b) Landscape for $\widehat{C}_G$ with a QAOA ansatz of $p = 1$ layer where α_1 was scaled by the condition number κ. In both cases the QLSP is defined by a randomly generated 4×4 matrix with condition number $\kappa \approx 11$, and with $|b\rangle$ given by (S79). The scaled landscape contains more regions of low cost and thus makes optimization more likely to be successful. (c) Time-to-solution versus condition number κ for the QLSP on three qubits defined by the A_2 matrix of (S77) and with $|b\rangle$ given by (S79). Three curves are shown for $\epsilon^2 = 0.10, 0.02$, and 0.01. The inset depicts the same data on a logarithmic scale. As can be seen from the inset, the scaling in κ is sub-exponential for each ϵ considered.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z	-1	-0.819 ± 0.005	0.032 ± 0.002

Supplementary Table S2: Expectation value of an observable M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{2\times 2} = \mathbb{1} + 0.25 \, Z$, and $|b\rangle = X|0\rangle$.
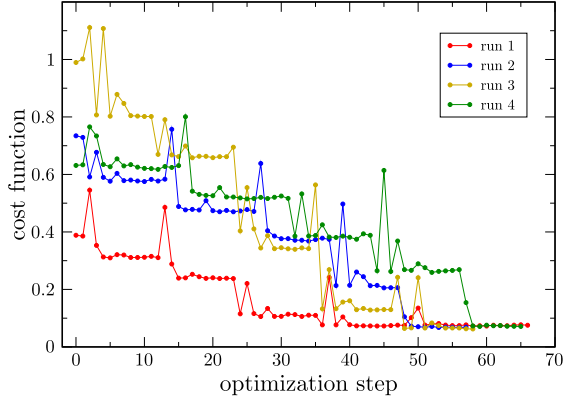
Supplementary Figure S4: Implementation of VQLS on Rigetti's quantum hardware. Cost function C_L is plotted versus number of optimization steps, where A is given by (S80). One can observe that for every run the cost function is reduced to a value of $\sim 7 \times 10^{-2}$. Due to noise present in the quantum device the cost does not go to zero.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z_2	1	0.943 ± 0.003	0.0032 ± 0.0003
Z_1	0	0.02 ± 0.04	0.000 ± 0.002
$Z_1 Z_2$	0	0.02 ± 0.04	0.000 ± 0.002

Supplementary Table S3: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{4\times4} = X_1 H_2$, and $|b\rangle = H_1 H_2 |\mathbf{0}\rangle$.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z_2	1	0.930 ± 0.004	0.0047 ± 0.0005
Z_1	0	0.00 ± 0.02	0.00000 ± 0.00006
$Z_1 Z_2$	0	0.00 ± 0.02	0.00000 ± 0.00009

Supplementary Table S4: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{4\times4} = \mathbb{1} + 0.25 Z_2$, and $|b\rangle = H_1 |\mathbf{0}\rangle$.

| M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z_3	1	0.88 ± 0.01	0.013 ± 0.002
Z_2	0	0.00 ± 0.04	0.0000 ± 0.0001
$Z_2 Z_3$	0	0.00 ± 0.03	0.0000 ± 0.0002
Z_1	0	0.04 ± 0.05	0.002 ± 0.003
$Z_1 Z_3$	0	0.04 ± 0.04	0.002 ± 0.004
$Z_1 Z_2$	0	-0.002 ± 0.009	0.00000 ± 0.00004
$Z_1 Z_2 Z_3$	0	-0.004 ± 0.009	0.00002 ± 0.00009

Supplementary Table S5: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{8\times8} = \mathbb{1} + 0.25 Z_3$, and $|b\rangle = H_1 H_2 |\mathbf{0}\rangle$.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z_5	0	0.180 ± 0.030	0.030000 ± 0.01000
Z_4	0	0.000 ± 0.100	0.000000 ± 0.00400
$Z_4 Z_5$	0	0.000 ± 0.020	0.000000 ± 0.00040
Z_3	0	0.000 ± 0.100	0.000000 ± 0.00900
$Z_3 Z_5$	0	0.000 ± 0.020	0.000000 ± 0.00040
$Z_3 Z_4$	0	-0.006 ± 0.009	0.000000 ± 0.00010
$Z_3 Z_4 Z_5$	0	0.000 ± 0.001	0.000000 ± 0.000001
Z_2	0	0.100 ± 0.020	0.010000 ± 0.00500
$Z_2 Z_5$	0	0.019 ± 0.009	0.000300 ± 0.00030
$Z_2 Z_4$	0	0.000 ± 0.010	0.000000 ± 0.00004
$Z_2 Z_4 Z_5$	0	-0.001 ± 0.009	0.000000 ± 0.00003
$Z_2 Z_3$	0	0.000 ± 0.010	0.000000 ± 0.00002
$Z_2 Z_3 Z_5$	0	0.005 ± 0.004	0.000020 ± 0.00004
$Z_2 Z_3 Z_4$	0	-0.007 ± 0.006	0.000050 ± 0.00009
$Z_2 Z_3 Z_4 Z_5$	0	-0.002 ± 0.008	0.000000 ± 0.00004
Z_1	0	0.010 ± 0.020	0.000200 ± 0.00070
$Z_1 Z_5$	0	0.005 ± 0.008	0.000030 ± 0.00009
$Z_1 Z_4$	0	0.003 ± 0.005	0.000010 ± 0.00003
$Z_1 Z_4 Z_5$	0	-0.002 ± 0.007	0.000000 ± 0.00004
$Z_1 Z_3$	0	0.000 ± 0.010	0.000000 ± 0.00001
$Z_1 Z_3 Z_5$	0	-0.002 ± 0.005	0.000000 ± 0.00002
$Z_1 Z_3 Z_4$	0	0.000 ± 0.008	0.000000 ± 0.00001
$Z_1 Z_3 Z_4 Z_5$	0	-0.001 ± 0.004	0.000000 ± 0.00004
$Z_1 Z_2$	0	0.006 ± 0.004	0.000040 ± 0.00005
$Z_1 Z_2 Z_5$	0	0.000 ± 0.010	0.000010 ± 0.00007
$Z_1 Z_2 Z_4$	0	-0.002 ± 0.001	0.000000 ± 0.00001
$Z_1 Z_2 Z_4 Z_5$	0	0.002 ± 0.005	0.000000 ± 0.00002
$Z_1 Z_2 Z_3$	0	0.001 ± 0.007	0.000000 ± 0.00002
$Z_1 Z_2 Z_3 Z_5$	0	0.000 ± 0.006	0.000000 ± 0.00001
$Z_1 Z_2 Z_3 Z_4$	0	0.001 ± 0.002	0.000001 ± 0.000006
$Z_1 Z_2 Z_3 Z_4 Z_5$	0	0.002 ± 0.001	0.000004 ± 0.000007

Supplementary Table S6: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{32\times32} = \mathbb{1} + 0.25 X_5$, and $|b\rangle = H^{\otimes 5} |\mathbf{0}\rangle$.

M	$\langle M \rangle_{\text{exact}}$	$\langle M \rangle_{\text{exp}}$	$D(M)^2$
Z_5	0	0.1 ± 0.1	0.01 ± 0.04
Z_4	0	0.00 ± 0.04	0.0 ± 0.0005
$Z_4 Z_5$	0	0.00 ± 0.01	0.0 ± 0.00007
Z_3	0	0.0 ± 0.1	0.0 ± 0.004
$Z_3 Z_5$	0	0.01 ± 0.02	0.0002 ± 0.0009
$Z_3 Z_4$	0	-0.002 ± 0.007	0.0 ± 0.00003
$Z_3 Z_4 Z_5$	0	0.000 ± 0.005	0.0 ± 0.000009
Z_2	1	0.971 ± 0.002	0.0008 ± 0.0001
$Z_2 Z_5$	0	0.1 ± 0.1	0.01 ± 0.04
$Z_2 Z_4$	0	0.00 ± 0.04	0.0 ± 0.0004
$Z_2 Z_4 Z_5$	0	0.00 ± 0.01	0.0 ± 0.00006
$Z_2 Z_3$	0	0.0 ± 0.1	0.0 ± 0.003
$Z_2 Z_3 Z_5$	0	0.01 ± 0.02	0.0002 ± 0.0009
$Z_2 Z_3 Z_4$	0	-0.002 ± 0.006	0.0 ± 0.00003
$Z_2 Z_3 Z_4 Z_5$	0	0.001 ± 0.005	0.0 ± 0.00001
Z_1	0	-0.02 ± 0.04	0.0 ± 0.001
$Z_1 Z_5$	0	0.000 ± 0.007	0.0 ± 0.0006
$Z_1 Z_4$	0	0.000 ± 0.004	0.0 ± 0.0000006
$Z_1 Z_4 Z_5$	0	0.000 ± 0.006	0.0 ± 0.00001
$Z_1 Z_3$	0	-0.002 ± 0.006	0.0 ± 0.00003
$Z_1 Z_3 Z_5$	0	-0.001 ± 0.004	0.0 ± 0.00001
$Z_1 Z_3 Z_4$	0	0.004 ± 0.005	0.00001 ± 0.00004
$Z_1 Z_3 Z_4 Z_5$	0	0.003 ± 0.004	0.00001 ± 0.00003
$Z_1 Z_2$	0	-0.02 ± 0.03	0.0 ± 0.001
$Z_1 Z_2 Z_5$	0	0.000 ± 0.008	0.0 ± 0.0000004
$Z_1 Z_2 Z_4$	0	0.000 ± 0.005	0.0 ± 0.000004
$Z_1 Z_2 Z_4 Z_5$	0	0.000 ± 0.006	0.0 ± 0.000006
$Z_1 Z_2 Z_3$	0	-0.001 ± 0.007	0.0 ± 0.00002
$Z_1 Z_2 Z_3 Z_5$	0	0.000 ± 0.004	0.0 ± 0.000008
$Z_1 Z_2 Z_3 Z_4$	0	0.004 ± 0.003	0.00001 ± 0.00003
$Z_1 Z_2 Z_3 Z_4 Z_5$	0	0.004 ± 0.003	0.00001 ± 0.00003

Supplementary Table S7: Expectation value of observables M computed with the exact solution, and with the output solution of VQLS. $D(M)$ measures the difference between these two results. The linear system considered is $A_{32 \times 32} = \mathbb{1} + 0.2 X_1 Z_2 + 0.2 X_1$, and $|b\rangle = H_1 H_3 H_4 H_5 |\mathbf{0}\rangle$.
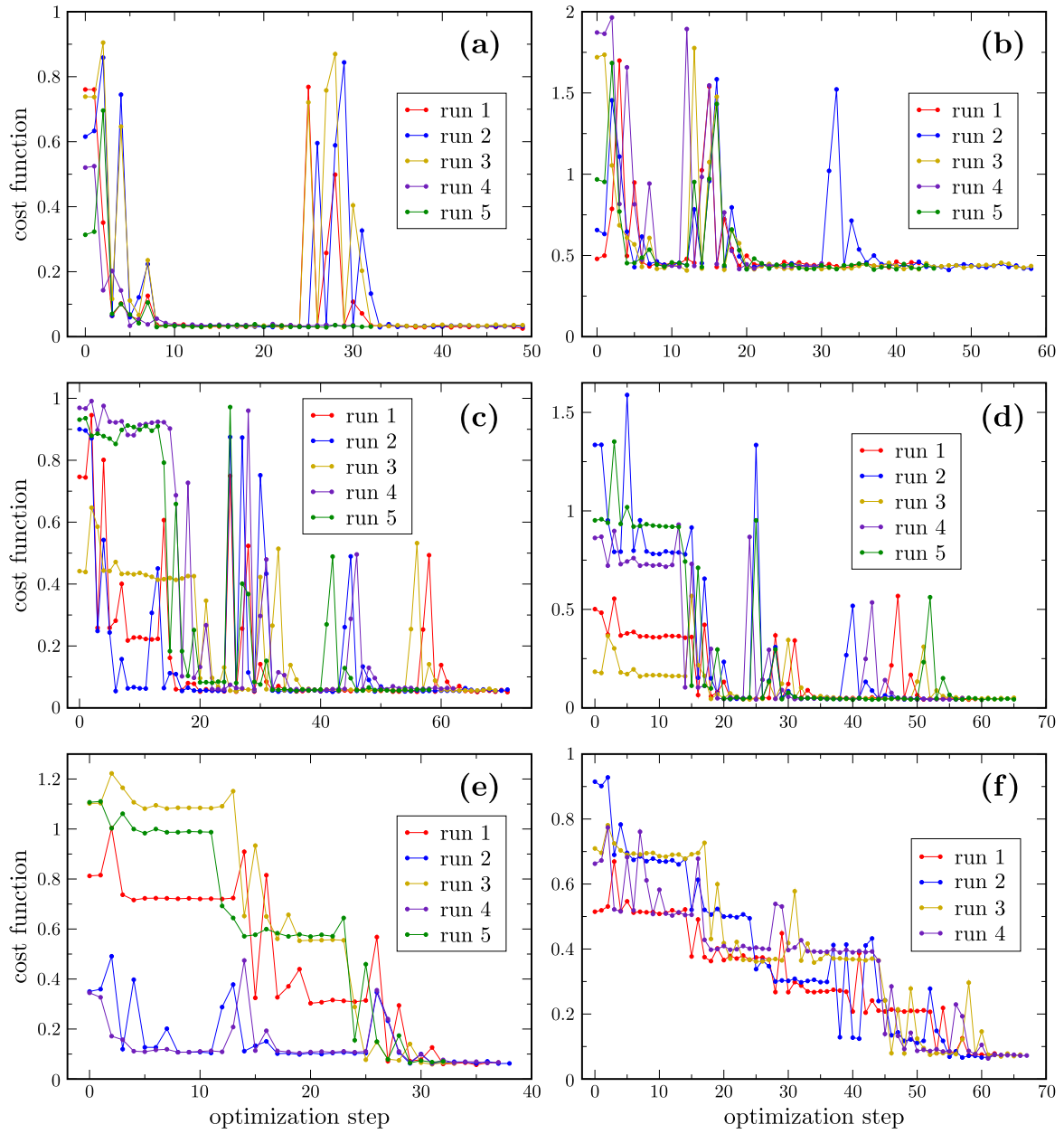
Supplementary Figure S5: Cost function versus number of optimization steps. The classical optimization algorithm employed is the Powell method which uses an unconstrained bi-directional search. Randomization in this algorithm occasionally leads to spikes in the cost function, visible in the plots, which quickly deteriorate as the optimizer reverts back towards better parameters. (a) $A_{2\times2} = H$, and $|b\rangle = X|0\rangle$. (b) $A_{2\times2} = \mathbb{1}+0.25Z$, and $|b\rangle = X|0\rangle$. (c) $A_{4\times4} = X_1 H_2$, and $|b\rangle = H_1 H_2|\mathbf{0}\rangle$. (d) $A_{4\times4} = \mathbb{1} + 0.25Z_2$, and $|b\rangle = H_1|\mathbf{0}\rangle$. (e) $A_{8\times8} = \mathbb{1} + 0.25Z_3$, and $|b\rangle = H_1 H_2|\mathbf{0}\rangle$. (f) $A_{32\times32} = \mathbb{1} + 0.25X_5$, and $|b\rangle = H^{\otimes5}|\mathbf{0}\rangle$.