

# Two-Unitary Decomposition Algorithm and Open Quantum System Simulation

Nishchay Suri<sup>1,2,3</sup>, Joseph Barreto<sup>1,2,4</sup>, Stuart Hadfield<sup>1,2</sup>, Nathan Wiebe<sup>5,6</sup>, Filip Wudarski<sup>1,2</sup>, and Jeffrey Marshall<sup>1,2</sup>

<sup>1</sup>QuAIL, NASA Ames Research Center, Moffett Field, California 94035, USA

<sup>2</sup>USRA Research Institute for Advanced Computer Science, Mountain View, California 94043, USA

<sup>3</sup>Department of Physics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

<sup>4</sup>QuTech, Delft University of Technology, Delft, The Netherlands

<sup>5</sup>Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3E1, Canada

<sup>6</sup>Pacific Northwest National Laboratory, Richland, Washington 99352, USA

Simulating general quantum processes that describe realistic interactions of quantum systems following a non-unitary evolution is challenging for conventional quantum computers that directly implement unitary gates. We analyze complexities for promising methods such as the Sz.-Nagy dilation and linear combination of unitaries that can simulate open systems by the probabilistic realization of non-unitary operators, requiring multiple calls to both the encoding and state preparation oracles. We propose a quantum two-unitary decomposition (TUD) algorithm to decompose a  $d$ -dimensional operator  $A$  with non-zero singular values as  $A = (U_1 + U_2)/2$  using the quantum singular value transformation algorithm, avoiding classically expensive singular value decomposition (SVD) with an  $\mathcal{O}(d^3)$  overhead in time. The two unitaries can be deterministically implemented, thus requiring only a single call to the state preparation oracle for each. The calls to the encoding oracle can also be reduced significantly at the expense of an acceptable error in measurements. Since the TUD method can be used to implement non-unitary operators as only two unitaries, it also has potential applications in linear algebra and quantum machine learning.

## 1 Introduction

In the early 1980s Manin [1] and Feynman [2] independently proposed that in order to circumvent the prohibitive scaling in the simulation of quantum systems, one would need a device that operates according to the principles of quantum mechanics. These considerations gave birth to the field that we now call *quantum computing* [3]. Since then an entire zoo of quantum algorithms for simulating quantum processes has emerged, some of them promise spectacular improvements over their classical alternatives, even exponential speedups [3, 4, 5, 6, 7, 8]. However, most of these prominent quantum algorithms are

---

Nishchay Suri: [nsuri@usra.edu](mailto:nsuri@usra.edu)

Joseph Barreto: [joeybarreto4@gmail.com](mailto:joeybarreto4@gmail.com)

Filip Wudarski: [filip.a.wudarski@nasa.gov](mailto:filip.a.wudarski@nasa.gov)

Jeffrey Marshall: [jeffrey.s.marshall@nasa.gov](mailto:jeffrey.s.marshall@nasa.gov)

solely tailored for unitary simulations of isolated quantum phenomena. Since such a quantum device is universal in its computational nature, one may expect that its applicability reaches beyond the standard unitary framework.

In general, quantum systems of interest are rarely isolated as they are exposed to the inevitable influence of their environments. The environment consists of inaccessible degrees of freedom of the system's natural surroundings as well as from the control devices that can become entangled with the system, ultimately leading to a loss of information. Therefore, the dynamics of a general quantum process must be described by a non-unitary, open quantum evolution [3, 9, 10, 11]. Understanding open quantum dynamics is indispensable to the study of dissipation and decoherence in quantum systems and therefore, noise in quantum circuits [12, 13, 14, 15, 16, 17], and is fundamental to a wide variety of phenomena such as thermalization [18, 19], non-equilibrium steady states [20, 21, 22], transport in strongly correlated systems [23, 24] as well as applications in quantum biology [25, 26, 27, 28]. Moreover, open quantum dynamics can be harnessed to perform universal quantum computation [29, 30] and prepare topological [31, 32, 33] as well as entangled states [34, 35, 36, 37]. Simulating the dynamics of open quantum systems is vital to understanding these processes.

Unlike isolated quantum evolution, open quantum evolution is non-unitary and thus presents a challenge for conventional quantum computers that are capable of directly implementing unitary gates. Quantum algorithms such as trotterization of Lindbladians [38, 39, 40, 41] and simulation of Markovian open dynamics from a set of universal channels [42, 43] rely on dilation techniques of encoding a non-unitary operator in a larger unitary. Recent advances have led to promising methods such as the Sz.-Nagy encoding, which provides a minimal dilation [44, 26, 45, 46]. Another useful method for realizing a non-unitary operator is by implementing it as a linear combination of unitaries (LCU) [47, 48], recently used to simulate Lindblad evolution [49]. In Ref. [50], the authors propose implementing an arbitrary operator as a linear combination of four unitaries created by approximate operator exponentiation. A recent work [51] implements the non-unitary operators in the singular basis by calculating the singular value decomposition (SVD) classically, thus incurring significant overhead for large systems. Additionally, non-unitary operators may be implemented via a combination of unitary operations and measurements requiring a feedback loop [52, 53], imaginary time evolution algorithms [54, 55, 56] that need quantum tomography at each step, and even with methods that use the intrinsic dissipation of the quantum computer [57].

In this article, we first analyze prominent techniques such as the Stinespring dilation, the Sz.-Nagy dilation and the LCU method and discuss their implementation of non-unitary operators to simulate open quantum dynamics and calculate expectation values of observables. The Sz.-Nagy and LCU methods implement the dynamical map by the probabilistic realization of each non-unitary Kraus operator individually. We estimate the number of encoding oracle calls, state preparation oracle calls, and auxiliary qubits required for successful implementation of individual operators in the dynamical map. We provide a fully quantum method to estimate the expectation value of an observable to a specified confidence level, improving on previous methods that use Cholesky decomposition, which incurs  $\mathcal{O}(d^3)$  of classical overhead, where  $d$  is the dimension. In particular, given an initial state  $|\psi\rangle$  prepared by the state preparation oracle  $\mathcal{S}$  and a contraction operator  $A$  block encoded as an isometry (unitary)  $U_A$ , the state  $A|\psi\rangle/\sqrt{p}$  where  $p = \langle\psi|A^\dagger A|\psi\rangle$ ,  $0 < p \leq 1$ , can be successfully implemented with probability at least  $1 - \beta$  using  $\mathcal{O}(1/p \log(1/\beta))$  calls to both  $U_A$  and  $\mathcal{S}$ . For realistic cases of interest such as the amplitude damping channel, a particular Kraus operator can have  $p \ll 1$  which

leads to a large number of calls to both  $U_A$  and  $\mathcal{S}$ , which can be especially limiting if the initial state is difficult to prepare.

We provide a quantum algorithm to decompose any arbitrary  $d$ -dimensional contraction operator  $A$  ( $\|A\| \leq 1$ )<sup>1</sup>, with non-zero singular values into two unitaries such that  $A = (U_1 + U_2)/2$  using the quantum singular value transformation (QSVT) algorithm [58, 59, 60, 5], without explicitly computing the SVD, which has  $\mathcal{O}(d^3)$  of classical overhead. Given a state  $|\psi\rangle$  prepared by  $\mathcal{S}$  and a block encoding  $U_A$  of an arbitrary operator  $A$ , our two-unitary decomposition (TUD) algorithm implements  $\tilde{U}_1|\psi\rangle, \tilde{U}_2|\psi\rangle$  which are  $\epsilon$  approximations of  $U_1|\psi\rangle, U_2|\psi\rangle$  with probability at least  $1 - \beta$  using  $\mathcal{O}(\log(1/\beta))$  calls to  $\mathcal{S}$ , and  $\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$  calls to  $U_A$  and  $U_A^\dagger$ , where the singular values  $\sigma_i$  of  $A$  are such that  $\sigma_i \in [\delta, 1 - \delta] \forall i$ . Therefore, instead of implementing  $A|\psi\rangle$  directly which is probabilistic, the TUD algorithm implements  $\tilde{U}_1|\psi\rangle$  and  $\tilde{U}_2|\psi\rangle$  deterministically with each requiring approximately one use of state preparation oracle  $\mathcal{S}$  or a single shot of  $|\psi\rangle$ . This is a considerable improvement from the previously needed  $\mathcal{O}(1/p \log(1/\beta))$  calls to  $\mathcal{S}$ . We show that the unitary decomposition can be used to estimate the expectation value of an observable with respect to  $A|\psi\rangle$  which can be done with Hadamard tests. Since the TUD method can be used to implement non-unitary operators as only two unitaries, it also has potential applications in linear algebra and quantum machine learning [61, 62, 63].

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Open Quantum Systems and Kraus Operators</b>	<b>4</b>
<b>3</b>	<b>Analysis for Previous Methods</b>	<b>5</b>
3.1	Stinespring Dilation . . . . .	5
3.2	Parallel Simulation of a Dynamical Map . . . . .	7
3.2.1	Sz.-Nagy Dilation . . . . .	9
3.2.2	Linear Combination of Unitaries . . . . .	10
<b>4</b>	<b>Two-Unitary Decomposition Algorithm</b>	<b>11</b>
4.1	Four-Unitary Decomposition . . . . .	15
4.2	Application: Dynamical Map Simulation . . . . .	16
4.2.1	Given General Encoding . . . . .	17
4.2.2	Given Sz.-Nagy Encoding . . . . .	19
<b>5</b>	<b>Conclusion</b>	<b>20</b>
<b>A</b>	<b>Total Number of Expected Runs</b>	<b>22</b>
<b>B</b>	<b>Error Estimation</b>	<b>22</b>
<b>C</b>	<b>Measuring Expectation Value: Hadamard Test and Variance</b>	<b>25</b>
<b>D</b>	<b>Quantum Singular Value Transformation</b>	<b>27</b>

---

<sup>1</sup>Throughout  $\|A\|$  denotes the operator norm of  $A$ , or equivalently, the maximum singular value of  $A$ .

## 2 Open Quantum Systems and Kraus Operators

The evolution of an isolated system in a mixed quantum state is described by the von Neumann equation

$$\dot{\rho}_t = -i[H(t), \rho_t], \quad (1)$$

where  $H(t)$  is the system's Hamiltonian and  $\rho_t := \rho(t)$  is the density matrix (in  $\hbar = 1$  units). The formal solution to Eq. (1) is a unitary transformation

$$\rho_t = \mathcal{U}_t(\rho_0) = U_t \rho_0 U_t^\dagger, \quad (2)$$

where  $U_t = \mathcal{T} \exp\left(-i \int_0^t H(t') dt'\right)$  and  $\mathcal{T}$  is the time-ordering operator, simplifying to  $U_t = \exp(-iHt)$  when  $H$  is time-independent.

The system and environment together undergo the unitary evolution of an isolated quantum process. To describe the effective dynamics of the system, we trace out the environmental degrees of freedom obtaining the open quantum evolution. Without loss of generality, one can assume that the system is initially decoupled from its environment such that the entire density matrix is  $\rho_0 = \omega_E \otimes \rho_S$ , where  $\omega_E$ ,  $\rho_S$  are the initial density matrices of the environment and system, respectively. The total evolution  $U_t = \mathcal{T} \exp\left(-i \int_0^t H_{\text{Total}}(t') dt'\right)$  is described by a Hamiltonian of the general form  $H_{\text{Total}}(t) = I_E \otimes H_S(t) + H_E(t) \otimes I_S + H_I(t)$ , where  $H_S$ ,  $H_E$  act only on the system and the environment respectively, while  $H_I$  is responsible for their interaction. This allows us to describe the effective dynamics of the  $d$ -dimensional system in terms of a dynamical map  $\Lambda_t: \mathcal{B}(\mathcal{H}_S) \rightarrow \mathcal{B}(\mathcal{H}_S)$  given by

$$\Lambda_t(\rho_S) = \text{Tr}_E \left( U_t \omega_E \otimes \rho_S U_t^\dagger \right) = \sum_{k=1}^m A_k(t) \rho_S A_k^\dagger(t), \quad (3)$$

where  $\mathcal{H}_S$  is the Hilbert space of the system,  $\text{Tr}_E$  is the partial trace over the Hilbert space of the environment, and  $m \leq d^2$ . The dynamical map  $\Lambda_t$  is a completely positive trace preserving (CPTP) map which describes physically valid transformation from quantum states to quantum states [9]. Following the partial trace, one obtains a set of Kraus operators  $A_k \in \mathcal{B}(\mathcal{H}_S)$ , where  $\mathcal{B}(\mathcal{H}_S)$  is the space of bounded linear operators. The Kraus operators are in general time-dependent, non-unitary, and also depend on the initial environmental state. For example, if  $\omega_E = |\nu\rangle\langle\nu|$ , then  $A_k(t) \equiv (\langle k| \otimes \mathbb{1}_S) U_t (|\nu\rangle \otimes \mathbb{1}_S)$ , where  $|k\rangle$  is a basis for the Hilbert space of the environment. Furthermore, the Kraus representation of the dynamical map is not unique as the partial trace over the environment is independent of the choice of basis. The Kraus operators themselves lack any special structure apart from the following trace-preservation constraint:  $\sum_k A_k(t)^\dagger A_k(t) = I$  for all  $t \geq 0$ . For notational clarity we shall suppress any explicit dependence on time or other parameters unless stated otherwise:  $A_k(t) \equiv A_k$ .

Since a quantum computer only has access to unitary operations and measurements, the implementation of dynamical maps comprised of non-unitary Kraus operators is a challenge. Implementing the transformation in Eq. (3) by implementing the entire unitary  $U_t$  that describes the system and its environment together quickly becomes infeasible as the environment can often have large or even infinite dimensionality. Therefore it is necessary to develop quantum algorithms that can emulate the effect of environmental influence indirectly. Given Kraus operators  $A_k$  block encoded in unitaries  $U_k$ , we estimate the number of resources to implement each non-unitary Kraus operator individually on a quantum computer as well as the entire map given by Eq. (3). In many practical cases,

$m$  can be assumed to be  $\mathcal{O}(\text{poly log } d)$  and therefore, the implementation of the map becomes practically feasible. For brevity we restrict further analysis to  $\rho_S = |\psi\rangle\langle\psi|$ , where  $|\psi\rangle$  is created by the oracle  $\mathcal{S}$ , with the extension to mixed states being straightforward. Additionally, we analyze the complexity to estimate the expectation value of a Hermitian observable  $O$ ,

$$\langle O \rangle = \text{Tr} \left( O \Lambda(|\psi\rangle\langle\psi|) \right) = \sum_{k=1}^m \langle \psi | A_k^\dagger O A_k | \psi \rangle, \quad (4)$$

such that the estimator is  $\widehat{\langle O \rangle} := 1/N \sum_{i=1}^N x_i$  where  $x_i$  is the  $i^{\text{th}}$  measured (randomly sampled) value of the random variable  $X$ , where  $\mathbb{E}(X) = \langle O \rangle$  and  $N$  is the total number of measurements. In practice  $N$  will be selected so that the variance of the estimate (or standard error of the mean) is below a certain pre-chosen threshold  $\text{Var}(\widehat{\langle O \rangle}) = \text{Var}(X)/N \leq v$ , as discussed in more detail in App. C.

### 3 Analysis for Previous Methods

#### 3.1 Stinespring Dilation

Simulating the dynamical map of Eq. (3) by realizing the entire evolution of the system and its environment together as a unitary is often infeasible. A natural and intuitive first approach is to instead emulate the degrees of freedom of the environment by a set of auxiliary qubits that may be traced out to obtain the desired map. The Stinespring dilation theorem, proposed long before the advent of quantum simulation, offers a way to achieve this ‘mimicking’ of the environmental degrees of freedom.

**Theorem 1 (Stinespring Dilation theorem)** *Let  $\Lambda: \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H})$  be a quantum CPTP map over a finite dimensional Hilbert space  $\mathcal{H}$ . Then there exists a Hilbert space  $\mathcal{E}$ , a unitary operator  $U_{\text{St}}$  over the joint Hilbert space  $\mathcal{E} \otimes \mathcal{H}$ , and a quantum state  $\omega \in \mathcal{B}(\mathcal{E})$ , such that*

$$\Lambda(\rho) = \text{Tr}_{\mathcal{E}} \left( U_{\text{St}} \omega \otimes \rho U_{\text{St}}^\dagger \right), \quad \forall \rho \in \mathcal{H}, \quad (5)$$

where  $\dim(\mathcal{E}) \leq \dim(\mathcal{H})^2$ , and the representation is unique up to a unitary equivalence.

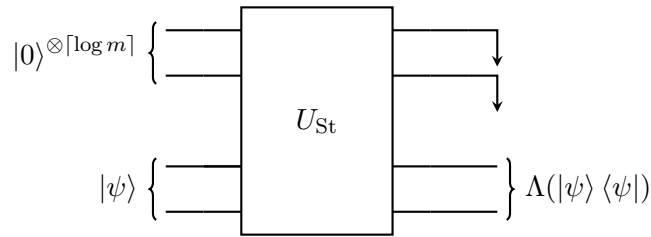


Figure 1: The Stinespring dilation unitary  $U_{\text{St}}$  with  $\lceil \log m \rceil$  auxiliary qubits that are traced out (indicated by downward-pointing arrows) to obtain the Kraus map.

The Kraus operators can be embedded in the first block-column of the Stinespring dilation unitary  $U_{\text{St}}$  that acts on the system and a set of auxiliary qubits emulating the

environment

$$U_{\text{St}} = \sum_{k=1}^m |k\rangle\langle 0| \otimes A_k + \dots = \begin{bmatrix} A_1 & \cdots & \cdots \\ A_2 & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ A_m & \cdots & \cdots \end{bmatrix}, \quad (6)$$

where  $m \leq d^2$ . We apply the unitary

$$U_{\text{St}} |0\rangle \otimes |\psi\rangle = \sum_{k=1}^m |k\rangle \otimes A_k |\psi\rangle, \quad (7)$$

and trace out the auxiliary qubits—analogueous to tracing out the environment—to obtain the dynamical map  $\Lambda(|\psi\rangle\langle\psi|) = \sum_k A_k |\psi\rangle\langle\psi| A_k^\dagger$ . Given the Kraus operators  $A_k$ , the  $md$  dimensional matrix  $U_{\text{St}}$  may be created by filling in the rest of  $(m-1)d$  columns via the Gram-Schmidt process, incurring a classical overhead of  $\mathcal{O}(m^3 d^3)$ . The quantum circuit for realizing the unitary  $U_{\text{St}}$  in Fig. 1 uses  $\mathcal{O}(\log(m))$  auxiliary qubits, and may be implemented with  $\mathcal{O}(m^2 d^2)$  one and two-qubit gates in the worst case [3]. The Stinespring method implements the entire dynamical map in one go and therefore, one can simply proceed to measure the observable  $O$ . The number of shots  $N$  is chosen such that  $\text{Var}(\langle O \rangle) \leq v$ .

One way to possibly reduce the number of gates is by completing the Stinespring unitary to minimize the number of gates in its decomposition rather than by a naive application of Gram-Schmidt procedure. While the above analysis does not assume any knowledge about the Kraus operators, the gate decomposition complexity can also improve greatly for some sparse or specially-structured Kraus operators.

**Example 1** *Let's consider the amplitude damping channel, with Kraus operators  $A_0 = |0\rangle\langle 0| + \sqrt{1-p}|1\rangle\langle 1|$  and  $A_1 = \sqrt{p}|0\rangle\langle 1|$ . The Stinespring unitary may be filled in as follows*

$$U_S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-p} & -\sqrt{p} & 0 \\ 0 & \sqrt{p} & \sqrt{1-p} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{array}{c} |0\rangle \\ | \psi \rangle \end{array} \begin{array}{c} \text{---} \boxed{R_y(\theta)} \text{---} \\ \text{---} \bullet \text{---} \\ \text{---} \bullet \text{---} \oplus \text{---} \end{array}$$

The channel is implemented with the aid of a single auxiliary qubit initialized in state  $|0\rangle$  which is traced out at the end of the evolution. The last two columns of the unitary are easy to obtain by inspection, due to the relative sparsity of the operators, and it is also straightforward to show  $\theta = 2 \sin^{-1}(\sqrt{p})$  given that  $R_y(\theta) = e^{-i\theta\sigma_y/2}$ .

**Example 2** *Simulating a Unitary Ensemble: This is inspired by the observation that many noise sources in controllable quantum systems stem from imperfect control, leading to a unitary implementation which can vary from realization to realization, e.g. disordered evolution [64]. One example is that of a continuous control pulse used to enact a particular unitary gate, where the pulse itself is prone to statistical fluctuations [65]. Moreover, Pauli errors are captured by the same framework. For concreteness, we consider the case of random unitary channels [66] also known as random external fields [67], where the quantum map is described by a discrete convex combination of unitaries  $U_i$*

$$\Lambda(\rho) = \sum_{i=1}^m q_i U_i \rho U_i^\dagger, \quad (8)$$

where  $q_i > 0$  are probabilities ( $\sum_i q_i = 1$ ).

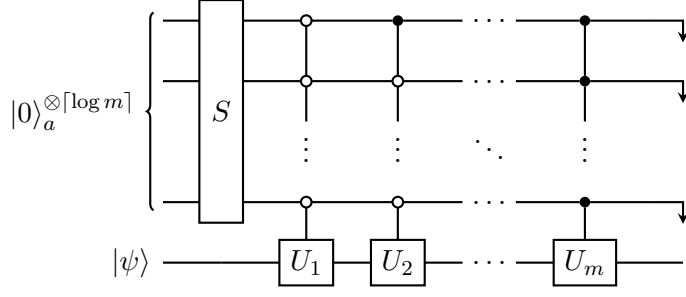


Figure 2: Circuit construction for simulating the unitary ensemble (8). The trailing symbols  $\downarrow$  on the auxiliary wires indicate trace-out.

An illustration of the proposed scheme to enact Eq. (8) is shown in Fig. 2, which proceeds in two steps. First, a state preparation unitary  $S$  is implemented to prepare with  $\mathcal{O}(\log m)$  auxiliary qubits the state  $\sum_{i=1}^m \sqrt{q_i} |i\rangle$ . Next, multi-controlled unitaries  $CU_i$  are applied, conditioned on the state of the auxiliary qubits, i.e.  $CU_i|i\rangle|\psi\rangle = |i\rangle U_i|\psi\rangle$ . The state of the original system plus auxiliary qubits is given by

$$|\Psi\rangle := \sum_i \sqrt{q_i} |i\rangle \otimes U_i|\psi\rangle, \quad (9)$$

where for ease of notation we assume the initial state is pure  $|\psi\rangle$ , though nothing in the above scheme is modified if it is an arbitrary mixed state. Finally, by tracing out (i.e. leaving unmeasured) the auxiliary qubits, one can see Eq. (9) becomes precisely Eq. (8):

$$\text{Tr}_a[|\Psi\rangle\langle\Psi|] = \text{Tr}_a \left[ \sum_{i,j} \sqrt{q_i q_j} |i\rangle\langle j| \otimes U_i|\psi\rangle\langle\psi| U_j^\dagger \right] = \sum_i q_i U_i|\psi\rangle\langle\psi| U_i^\dagger. \quad (10)$$

At this point, any expectation value can be measured as desired using this output state.

### 3.2 Parallel Simulation of a Dynamical Map

A natural attempt at overcoming the difficulty of creating and implementing the entire Stinespring unitary is to implement each Kraus operator individually in parallel [44], which we will see below can reduce the gate complexity. Since in general the Kraus operator  $A_k$  is not unitary, it can itself be encoded within a larger unitary  $U_k$  which can be implemented on a quantum computer. The Kraus operator can then be accessed via measurement of the auxiliary encoding qubits leading to its probabilistic implementation. We define a general block encoding as below.

**Definition 1 (Block Encoding [58])** Assuming that  $A$  is an  $n$ -qubit operator<sup>2</sup>,  $U$  is a unitary acting on  $n + \ell$  qubits, and  $\alpha \in \mathbb{R}^+$  is an upper bound on the maximum singular value of  $A$ , then we say that  $U$  is a  $(\alpha, \ell, \Delta)$  block-encoding of  $A$  if

$$\|A - \alpha(\langle 0| \otimes \mathbb{1})U(|0\rangle \otimes \mathbb{1})\| \leq \Delta. \quad (11)$$

Without loss of generality, we assume that  $A/\alpha$  is the top-left block of  $U$

$$U = \begin{bmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{bmatrix}, \quad (12)$$

<sup>2</sup>If  $A$  has dimensionality  $d < 2^n$ , then  $A$  can be completed to act on  $n$  qubits by padding with zeros.

where  $\|A/\alpha\| \leq 1$ .

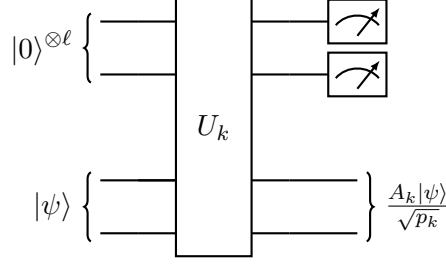


Figure 3: A unitary from the set of dilation unitaries  $\{U_k\}$  such that  $k \in \{1, \dots, m\}$ , where each implements the state  $A_k |\psi\rangle / \sqrt{p_k}$  conditioned on measuring the auxiliary qubit in state  $|0\rangle^{\otimes \ell}$ .

The trace preservation constraint implies that all Kraus operators are contractions such that  $\|A_k\| \leq 1 \forall k$ . Therefore they may be embedded with  $\alpha = 1$  for any general block encoding [44]. We assume that each  $A_k$  is block encoded in its respective unitary  $U_k$  with  $\ell$  encoding auxiliary qubits denoted by a  $(1, \ell, 0)$  block encoding. The action of each unitary on the state  $|\psi\rangle$  is given as

$$U_k |0\rangle^{\otimes \ell} \otimes |\psi\rangle = |0\rangle^{\otimes \ell} \otimes A_k |\psi\rangle + |\Psi_{\perp}\rangle, \quad (13)$$

where  $(\langle 0^{\otimes \ell} | \otimes \mathbb{1}) |\Psi_{\perp}\rangle = 0$ . The state  $A_k |\psi\rangle / \sqrt{p_k}$  is successfully implemented with probability  $p_k = \langle \psi | A_k^{\dagger} A_k |\psi\rangle = \|A_k |\psi\rangle\|^2$  when the auxiliary qubits are measured in state  $|0\rangle^{\otimes \ell}$ . To implement the state  $A_k |\psi\rangle / \sqrt{p_k}$  successfully with at least probability  $1 - \beta$ , we require  $\mathcal{O}(1/p_k \log(1/\beta))$  calls to the block encoding oracle  $U_k$ . Since we also need to prepare  $|\psi\rangle$  each time after the failure, we require the same number of calls to the state preparation oracle  $\mathcal{S}$ . The total number of expected calls to both  $U_k$  and  $\mathcal{S}$  to implement all  $m$  Kraus operators in parallel is  $E = \sum_{i=1}^m 1/p_k$  where  $E \geq m^2$  as derived in App. A. If we know the initial state  $|\psi\rangle$ , we can use amplitude amplification to achieve the square root advantage [68, 69] to implement each state in  $\mathcal{O}(1/\sqrt{p_k} \log(1/\beta))$  calls to  $U_k$  and  $\mathcal{S}$ . Similarly the total number of expected calls to  $U_k$  and  $\mathcal{S}$  is  $E' = \sum_{k=1}^m 1/\sqrt{p_k}$  where  $E' \geq m^{3/2}$ .

We can measure the expectation value of the observable  $O$  in parallel which is given by  $\langle O \rangle = \sum_{k=1}^m \langle A_k^{\dagger} O A_k \rangle$ . Each time any of the states  $A_k |\psi\rangle / \sqrt{p_k}$  is successfully implemented, we can proceed to perform a measurement of  $O$ . The observable  $O$  may be measured directly if one knows the eigenbasis of  $O$ , or when  $O$  is provided as a linear combination of unitaries (e.g. Pauli strings) which may be measured separately and totaled (see Sect. 3.2.2). More generally if no information is provided, one may instead use the Hadamard test to calculate the expectation value given a block-encoding of  $O$ . We detail such a calculation and its associated variance in Appendix C. This is repeated to obtain the expectation value of each term  $\langle A_k^{\dagger} O A_k \rangle$  in the sum to calculate  $\langle O \rangle$ . Finally a summation over the  $m$  terms is performed to obtain the expectation value of  $O$ . The condition  $\text{Var}(\langle \widehat{O} \rangle) \leq v$ , implies that  $\sum_k (p_k - \langle A_k^{\dagger} O A_k \rangle^2) / N_k \leq v$ , where  $N_k$  is the number of shots or measurements for each term  $\langle A_k^{\dagger} O A_k \rangle$ .

A completely arbitrary block encoding  $U_k$  uses  $\mathcal{O}(L^2 d^2)$  one and two qubit gates for its decomposition in the worst case where  $L = 2^{\ell}$ . The total gate complexity is  $\mathcal{O}(m L^2 d^2)$  and total number of additional qubits  $\mathcal{O}(m \log(L))$  for the  $m$  Krauses. Our process of measuring each expectation value only requires the implementation of block encoding  $U_k$



and the  $(1, q, 0)$  block encoding  $U_O$  of observable  $O$  using at most  $\mathcal{O}((L^2 + Q^2)d^2)$  one and two qubit gates for decomposition, where  $Q = 2^q$ . Arbitrary Kraus operators  $A_k$  and thus arbitrary unitaries require an exponential number (with respect to the number of qubits) of one and two-qubit gates to decompose. In many realistic scenarios, each  $A_k$  is a sparse matrix describing transitions between  $s$  quantum states where  $s = \mathcal{O}(\text{poly log } d)$ . In such cases, the number of one and two qubit gates for decomposition reduces to  $\mathcal{O}(L^2 s^2)$ . We discuss below that for special oracles like Sz.-Nagy and LCU encodings, this complexity can be reduced significantly.

### 3.2.1 Sz.-Nagy Dilation

The Sz.-Nagy dilation theorem guarantees a minimal block encoding of any arbitrary  $d$ -dimensional contraction  $A$  using only one auxiliary qubit.

**Theorem 2 (Sz.-Nagy theorem [70, 71, 44])** *For every contraction operator  $A: \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H})$  ( $\|A\| \leq 1$ )<sup>3</sup> there exists a unitary dilation operator  $U: \mathcal{B}(\mathcal{H}) \oplus \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H}) \oplus \mathcal{B}(\mathcal{H})$  where*

$$U^{SN} = \begin{bmatrix} A & \sqrt{I - AA^\dagger} \\ \sqrt{I - A^\dagger A} & -A^\dagger \end{bmatrix}, \quad (14)$$

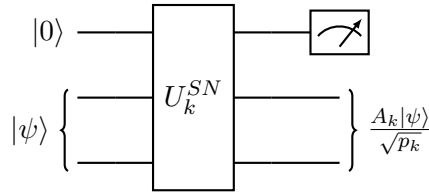


Figure 4: A unitary from the set of Sz.-Nagy's dilation unitaries  $\{U_k^{SN}\}$  such that  $k \in \{1, \dots, m\}$ , where each implements the state  $A_k |\psi\rangle / \sqrt{p_k}$  conditioned on measuring the auxiliary qubit in state  $|0\rangle$ .

Hu, Xia and Kais [44] give a quantum algorithm where each Kraus operator  $A_k$  is encoded in its respective Sz.-Nagy unitary denoted by  $U_k^{SN}$  which is a  $(1,1,0)$  block encoding. The action of each unitary on a state and an auxiliary qubit is defined as

$$U_k^{SN} |0\rangle \otimes |\psi\rangle = |0\rangle \otimes A_k |\psi\rangle + |1\rangle \otimes \sqrt{I - A_k^\dagger A_k} |\psi\rangle. \quad (15)$$

Since  $U_k$  is a  $2d$  dimensional matrix, its decomposition uses  $\mathcal{O}(d^2)$  one and two qubit gates for the worst case. The total space complexity for simulating the full map of  $m$  Kraus operators, in the worst case, is therefore  $\mathcal{O}(md^2)$ , and total number of additional qubits is  $\mathcal{O}(m)$ . The same results for general block encodings apply here except that the number of gates required for decomposition is  $\mathcal{O}(d^2)$  which is a great reduction as it removes the dependence on  $L$  which in general can depend on  $d$ .

Hu, Xia and Kais [44] provide an innovative method to calculate the expectation value by converting the observable  $O$  to a positive-semidefinite form  $\tilde{O}$  and performing Cholesky decomposition which can be incorporated into the dilation itself. This decomposition uses at most a classical overhead of  $\mathcal{O}(d^3)$  per Kraus operator. We do not require the additional step of Cholesky decomposition using our method to estimate expectation value

<sup>3</sup>We use  $\|A\|$  to denote the operator norm (i.e., the maximum singular value) throughout.

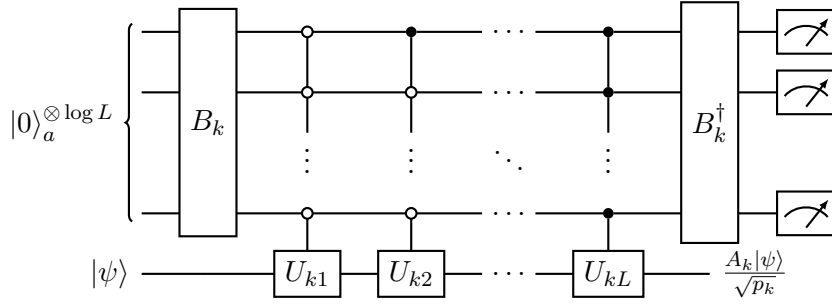


Figure 5: Circuit construction to implement the LCU block encoding unitaries  $\{(B_k \otimes I)U_k(B_k^\dagger \otimes I)\}$  such that  $k \in \{1, \dots, m\}$ , where each implements the state  $A_k |\psi\rangle / \sqrt{p_k}$  conditioned on measuring the auxiliary qubit in state  $|0\rangle^{\otimes \ell}$ .

mentioned in Sec. 3.2 which reduces to  $\mathcal{O}(d^2)$  complexity in the worst case given the Sz.-Nagy encodings. Similarly as above, in the case when  $A_k$  are  $s$ -sparse, the number of one and two qubit gates for decomposition reduces to  $\mathcal{O}(s^2)$  and each Sz.-Nagy circuit becomes efficient to implement. However there are still  $m$  circuits to be implemented, which can be done in parallel.

The Sz.-Nagy dilation needs only one auxiliary qubit regardless of the dimension of the encoded  $A_k$ , making it the most economical block-encoding. However, constructing the dilation unitary might not be efficient as implementing the off-diagonal blocks  $\sqrt{I - A^\dagger A}, \sqrt{I - AA^\dagger}$  is non-trivial. Moreover, we show in Sec 4.2 that the knowledge of the off-diagonal components of the Sz.-Nagy unitaries is equivalent to the knowledge of expressing any  $d$ -dimensional Hermitian matrix in two unitaries or a  $d$ -dimensional general matrix in four unitaries, instead of the  $d^2$  unitaries that span the basis.

### 3.2.2 Linear Combination of Unitaries

The Linear Combination of Unitaries (LCU) method allows for a probabilistic implementation of an arbitrary operator in a quantum circuit realized as a linear combination of unitaries [47, 59, 72]. Using the LCU method we can block encode an arbitrary matrix into a larger unitary given that we know its decomposition as a sum of unitaries. Let an arbitrary operator  $A_k$  be represented as a linear combination of  $L$  unitaries

$$A_k = \sum_{i=1}^L \alpha_{ki} U_{ki}, \quad (16)$$

where  $\|A_k\| \leq \alpha_k = \sum_{i=1}^L |\alpha_{ki}|$  and  $\alpha_{ki} \geq 0, \forall k, i$  without loss of generality. It is assumed that information about each Kraus operator  $A_k$  is provided in terms of  $\alpha_{ki}$  as a list of  $L$  numbers and each  $U_{ki}$  is assumed to be implemented with at most  $c$  one and two qubit gates. Examples of such  $U_{ki}$  include Pauli operators or more generally 1-sparse matrices.

**Lemma 1 (LCU Lemma [47, 59])** *Given a set of  $\alpha_{ki} \geq 0$  and  $U_{ki}$ , with  $U_{ki}U_{ki}^\dagger = I$ , such that  $A_k = \sum_{i=1}^L \alpha_{ki} U_{ki}$ , there exists a quantum circuit (Fig. 5) that implements*

$$\frac{A_k}{\alpha_k} = (|0\rangle^{\otimes \ell} \otimes \mathbb{1})(B_k \otimes I)U_k(B_k^\dagger \otimes I)(|0\rangle^{\otimes \ell} \otimes \mathbb{1}) \quad (17)$$

where  $\ell = \log L$ ,  $B_k$  is a state preparation unitary such that  $B_k |0\rangle^{\otimes \ell} = \sum_{i=1}^L \sqrt{\frac{\alpha_{ki}}{\alpha_k}} |i\rangle$  and  $U_k = \sum_{i=1}^L |i\rangle \langle i| \otimes U_{ki}$ . The unitary  $(B_k \otimes I)U_k(B_k^\dagger \otimes I)$  is a  $(\alpha_k, \ell, 0)$  block encoding of  $A_k$ .

The LCU circuit in Fig. 5 described by Eq. (17) implements the Kraus operator  $A_k$ , requiring  $\log L$  auxiliary qubits with at most  $Lc$  number of decomposition gates. The total gate complexity is at most  $mLc$  and the total number of qubits at most  $m[\log(L)]$ . The LCU method implements the state  $A_k |\psi\rangle / (\alpha_k \sqrt{p_k})$  with probability  $p_k = \langle \psi | A_k^\dagger A_k | \psi \rangle / \alpha_k^2$ . The previous discussion and all results about the number of oracle calls to  $U_k$  and  $\mathcal{S}$  hold with the modification  $A_k \rightarrow A_k / \alpha_k$ , or equivalently  $p_k \rightarrow p_k / \alpha_k^2$ .

Cleve and Wang in Ref. [49] show that all Kraus operators can be implemented in a single circuit by using an additional set of qubits applying the LCU method to sum over the Kraus operator index  $k$ . Their method can implement  $\epsilon$  approximation of a quantum channel comprising of  $m$  Kraus operators in a  $2^n$  dimensional space, with a quantum circuit of size  $\mathcal{O}(q^2 m^2 \frac{(\log(mq/\epsilon) + n) \log(1/\epsilon)}{(\log \log(1/\epsilon))})$ , where each Kraus is given in a linear combination of  $q$  Paulis.

## 4 Two-Unitary Decomposition Algorithm

A non-unitary operator given in a block encoding unitary such as the Sz.-Nagy or LCU unitary has a probabilistic implementation. Each time the implementation fails, the block encoding unitary as well as the initial state preparation oracle must be applied again. In this section, we provide a quantum algorithm to decompose an arbitrary  $d$ -dimensional contraction operator  $A$  into two unitaries such that  $A = (U_1 + U_2)/2$  using the quantum singular value transformation (QSVT) algorithm [58, 59, 60, 5]. Given a state  $|\psi\rangle$  prepared by  $\mathcal{S}$  and a  $(1, \ell, 0)$  block encoding  $U_A$  of an arbitrary operator  $A$ , the TUD algorithm implements  $\tilde{U}_1 |\psi\rangle, \tilde{U}_2 |\psi\rangle$  such that  $\|U_1 - \tilde{U}_1\|, \|U_2 - \tilde{U}_2\| \leq \epsilon$ , with probability at least  $1 - \beta$  using  $\mathcal{O}(\log(1/\beta))$  calls to  $\mathcal{S}$ , and  $\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$  calls to  $U_A$  and  $U_A^\dagger$ , where the singular values  $\sigma_i$  of  $A$  are such that  $\sigma_i \in [\delta, 1 - \delta] \forall i$ . Any operator can be alternatively realized by implementing the two unitaries separately with each requiring one use of the state preparation oracle  $\mathcal{S}$  (i.e. a single shot of  $|\psi\rangle$ ). This is a considerable improvement from the previously needed  $\mathcal{O}(1/p \log(1/\beta))$  calls to  $\mathcal{S}$ . We apply the TUD algorithm to implement non-unitary Kraus operators to simulate the dynamical map and calculate the expectation value of an observable in Sec. 4.2. The queries to the block encoding unitaries  $U_k$  can also be reduced significantly at the expense of an acceptable error in measurements.

We assume access to a  $(1, \ell, 0)$  block encoding  $U_A$  of an arbitrary operator  $A = (\langle 0 |^{\otimes \ell} \otimes \mathbb{1}) U_A (|0\rangle^{\otimes \ell} \otimes \mathbb{1})$  where  $\ell = \log(L)$ . Information about a sparse  $A$  may typically be given in terms of its Pauli expansion,  $A = \sum_{i=1}^L \alpha_i P_i$ , where  $P_i$  are the products of  $d$ -Pauli matrices. The oracle can then be created with techniques such as LCU using Ref. [47, 59] or Lemma 1 as shown in Sec. 3.2.2. Mathematically, any arbitrary operator  $A$  can be expressed in two unitaries as  $A = (U_1 + U_2)/2$  which is guaranteed by the following theorem.

**Theorem 3 (Two Unitary Decomposition [73, 74])** *For every contraction operator  $A: \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H})$  ( $\|A\| \leq 1$ ), there exist two unitary operators  $U_1$  and  $U_2$  such that  $A = \frac{1}{2}(U_1 + U_2)$ . The operators  $U_1 = \sum_i (\sigma_i + i\sqrt{1 - \sigma_i^2}) |w_i\rangle \langle v_i|$  and  $U_2 = \sum_i (\sigma_i - i\sqrt{1 - \sigma_i^2}) |w_i\rangle \langle v_i|$  are obtained from the singular value decomposition of  $A: A = \sum_i \sigma_i |w_i\rangle \langle v_i|$ , where the  $|w_i\rangle$  and  $|v_i\rangle$  form orthonormal bases.*

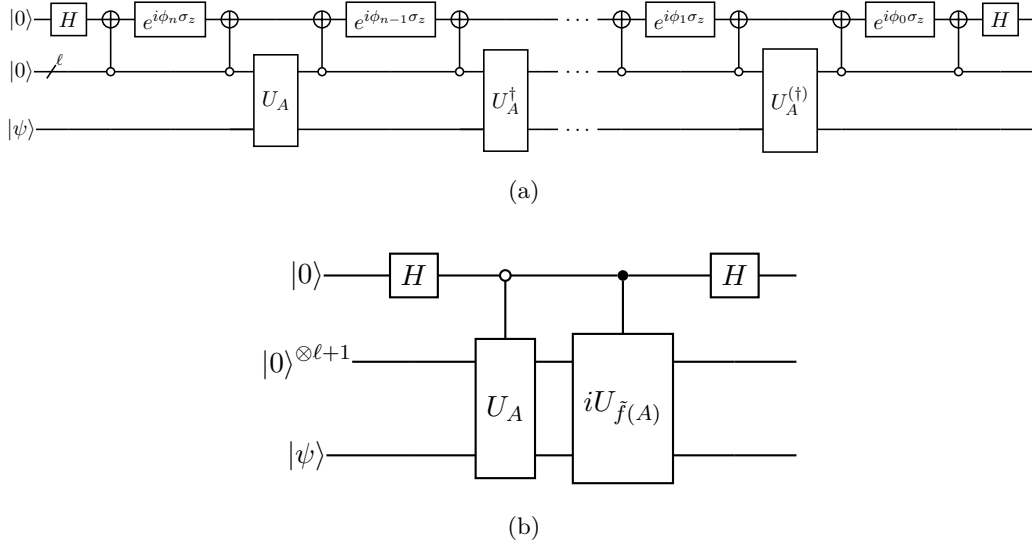


Figure 6: (a) The QSVT circuit to implement  $U_{\tilde{f}(A)}$  such that  $\tilde{f}(A) = (|0\rangle^{\otimes \ell+1} \otimes \mathbb{1})U_{\tilde{f}(A)}(|0\rangle^{\otimes \ell+1} \otimes \mathbb{1})$ . The final application of the block-encoding oracle is  $U_A^{(\dagger)}$ , which is  $U_A$  or  $U_A^\dagger$  for an odd or even polynomial of degree  $n$  respectively. (b) Circuit  $C$  in Eq. (21) to obtain the two decomposition unitaries  $\tilde{U}_1$  and  $\tilde{U}_2$ .

The above theorem relies on performing a SVD to obtain  $U_1$  and  $U_2$  which costs  $\mathcal{O}(d^3)$  on a classical computer and is therefore prohibitively expensive for large quantum systems. We avoid this cost in our algorithm as we do *not* perform SVD. In essence, our algorithm relies on the insight that if the SVD of  $A = \sum_i \sigma_i |w_i\rangle \langle v_i|$  then,  $U_1 = A + if(A)$  and  $U_2 = A - if(A)$ , where the function  $f(A) := \sum_i \sqrt{1 - \sigma_i^2} |w_i\rangle \langle v_i|$  is defined over the singular values of  $A$ .

Our algorithm implements  $\tilde{U}_1 |\psi\rangle$ ,  $\tilde{U}_2 |\psi\rangle$  where  $\tilde{U}_1 = A + i\tilde{f}(A)$  and  $\tilde{U}_2 = A - i\tilde{f}(A)$  such that  $\|U_1 - \tilde{U}_1\|, \|U_2 - \tilde{U}_2\| \leq \epsilon$ . We create the function  $\tilde{f}(A)$  which is an  $n$ -degree polynomial approximation of the function  $f(A)$  such that  $\|\tilde{f}(A) - f(A)\| \leq \epsilon$  using the QSVT algorithm as shown in Fig. 6 (a). The function  $\tilde{f}(A)$  is implemented with  $n/2 + 1$  calls to  $U_A$  and  $n/2$  calls to  $U_A^\dagger$  such that  $n$  is  $\mathcal{O}(1/\delta \log(1/\epsilon))$  where we assume that the singular values of  $A$ ,  $\sigma_i \in [\delta, 1 - \delta] \forall i$ , such that  $\delta > 0$  is known. We then use the LCU-addition circuit in Fig. 6 (b) to add  $A$  and  $\pm i\tilde{f}(A)$  to obtain  $\tilde{U}_{1,2}/2$ . Using oblivious amplitude amplification (OAA) then boosts the probability of implementation arbitrarily close to one [48, 72], finally obtaining  $\tilde{U}_1$  and  $\tilde{U}_2$ .

The QSVT algorithm can implement the transformation  $\tilde{g}(A) = \sum_i \tilde{g}(\sigma_i) |v_i\rangle \langle v_i|$  for even degree polynomial of  $\sigma_i$  or  $\tilde{f}(A) = \sum_i \tilde{f}(\sigma_i) |w_i\rangle \langle v_i|$  for an odd degree polynomial of  $\sigma_i$ . We must use an odd degree polynomial approximation as we require the basis of transformation to be  $|w_i\rangle \langle v_i|$  to ensure that  $\tilde{U}_{1,2} = \sum_i (\sigma_i \pm i\tilde{f}(\sigma_i)) |w_i\rangle \langle v_i|$  are unitaries. Since the even function  $\sqrt{1 - \sigma^2}$  cannot be approximated by an odd degree polynomial, we implement the polynomial approximation of the odd function  $\text{sign}(\sigma)\sqrt{1 - \sigma^2}$ . Both functions are identical in the relevant domain  $\sigma \in (0, 1]$  as singular values are always non-negative. We present a slightly modified algorithm for the case when  $\delta = 0$  in the next section by implementing  $A$  as a four-unitary decomposition.

To obtain the odd polynomial approximation of the function  $f(x) = \text{sign}(x)\sqrt{1 - x^2}$ ,

we apply the QSVT transformation shown in Fig. 6 (a)

$$U_{\tilde{f}(A)} = \Pi_{\phi_0} U_A \left[ \prod_{k=1}^{(n-1)/2} \Pi_{\phi_{2k-1}} U_A^\dagger \Pi_{\phi_{2k}} U_A \right] \Pi_{\phi_n}, \quad (18)$$

where  $n$  is an odd number,  $\Pi_\phi = e^{i\phi(2\Pi-I)}$ ,  $\Pi = (|0\rangle\langle 0|)^{\otimes \ell} \otimes I$ , and the set of phases  $\{\phi_i\}$  in Eq. (18) are efficiently computable [75, 76, 77]. We project it to the block-encoded subspace to obtain  $\tilde{f}(A) = (\langle 0|^{\otimes \ell+1} \otimes \mathbb{1}) U_{\tilde{f}(A)} (|0\rangle^{\otimes \ell+1} \otimes \mathbb{1})$ . Since each application of the encoding oracle  $U_A$  increases the degree of the polynomial by one, the required circuit depth such that  $|f(x) - \tilde{f}(x)| \leq \epsilon$  where  $x \in [-1 + \delta, -\delta] \cup [\delta, 1 - \delta]$  can be found by setting degree  $n = \mathcal{O}(1/\delta \log(1/\epsilon))$ , see App. D. Figures 7 (a) and (b) show the QSVT approximation for the scalar function  $f(x) = \text{sign}(x)\sqrt{1-x^2}$  and the relative error using a set of 51 angles  $\{\phi_i\}$ , respectively, computed using the `pyqsp` open-source repository [78]. After the previous step, we have

$$U_A |0\rangle^{\otimes \ell} |\psi\rangle = |0\rangle^{\otimes \ell} A |\psi\rangle + |\Psi_A^\perp\rangle, \quad (19)$$

$$U_{\tilde{f}(A)} |0\rangle^{\otimes \ell+1} |\psi\rangle = |0\rangle^{\otimes \ell+1} \tilde{f}(A) |\psi\rangle + |\Psi_{\tilde{f}(A)}^\perp\rangle, \quad (20)$$

where  $(\langle 0|^{\otimes \ell} \otimes \mathbb{1}) |\Psi_A^\perp\rangle = (\langle 0|^{\otimes \ell+1} \otimes \mathbb{1}) |\Psi_{\tilde{f}(A)}^\perp\rangle = 0$ . We add  $U_A$  and  $\pm i U_{\tilde{f}(A)}$  by implementing a LCU circuit to obtain  $\tilde{U}_1$  and  $\tilde{U}_2$  as shown in Fig. 6 (b) obtaining

$$C |0\rangle |0\rangle^{\otimes \ell+1} |\psi\rangle = \frac{1}{2} \{ |0\rangle (|0\rangle^{\otimes \ell+1} \tilde{U}_1 |\psi\rangle + |\Psi_+^\perp\rangle) + |1\rangle (|0\rangle^{\otimes \ell+1} \tilde{U}_2 |\psi\rangle + |\Psi_-^\perp\rangle) \}, \quad (21)$$

where  $|\Psi_\pm^\perp\rangle = |\Psi_A^\perp\rangle \pm i |\Psi_{\tilde{f}(A)}^\perp\rangle$ . The decomposition unitaries  $\tilde{U}_1$  and  $\tilde{U}_2$  are each flagged by auxiliary qubits  $|0\rangle |0\rangle^{\otimes \ell+1}$  and  $|1\rangle |0\rangle^{\otimes \ell+1}$ . Each of the unitaries  $\tilde{U}_1$  or  $\tilde{U}_2$  can now be implemented deterministically by only using oblivious amplitude amplification once [48, 72] for each case. We show in App. B that the probability of successfully implementing  $\tilde{U}_1, \tilde{U}_2$  is  $\sim 1 - 3\epsilon^2/4$ . We shall neglect the dependence on  $\epsilon^2$  moving forward by choosing it small enough. A high level description of the TUD algorithm is given in Alg. 1.

We demonstrate our algorithm by decomposing a set of two-dimensional matrices  $A$  such that  $A = U \text{diag}(\sigma, \sigma) V^\dagger$ , where  $U$  and  $V^\dagger$  are both independently randomized<sup>4</sup>. We decompose  $A$  and implement  $\tilde{U}_1$  and  $\tilde{U}_2$  by running our two-unitary decomposition algorithm for an approximating polynomial or query complexity of  $n = 51$ . Figure 7 (c) shows the approximation error  $\|U_1 - \tilde{U}_1\|$  and Fig. 7 (d) shows the probability of success of implementing  $\tilde{U}_1$  with respect to singular values of  $A$ . We see that when the singular values of  $A$  fall within the range  $\sim [0.1, 0.9]$ , the approximation error remains below  $10^{-2}$  and the unitaries are deterministically implemented.

The algorithm produces the correct output with  $n = \mathcal{O}(1/\delta \log(1/\epsilon))$  uses of the oracle with the promise that singular values  $\sigma_i \in [\delta, 1 - \delta] \forall i$ . In practice we control  $n$  and can increase it to obtain a better approximation. When given the minimum singular value  $\delta$ , we can increase the degree of fitting polynomial  $\tilde{f}(x)$  in Fig. 7 (a),(b) until  $|f(x) - \tilde{f}(x)| \leq \epsilon, \forall x > \delta$ .

<sup>4</sup>We use the `qutip` library (version 4.6.2 [79]) and its method `rand_unitary` with all default parameters which draws random Hermitian operators, multiplies them by “ $-i$ ” and then exponentiates to form a unitary matrix. We make all singular values the same so that only the error at each unique value of  $\sigma$  contributes for each sampled matrix  $A$ .

---

**Algorithm 1: Two Unitary Decomposition**


---

- Input** : Arbitrary matrix  $A$ ,  $\delta, \epsilon > 0$  such that singular values of  $A$  are  $\sigma_i \in [\delta, 1 - \delta] \forall i$  block encoded in a unitary  $U_A$  such that  $A = (\langle 0 |^{\otimes \ell} \otimes \mathbb{1}) U_A (|0 \rangle^{\otimes \ell} \otimes \mathbb{1})$ . The state  $|\psi\rangle$  prepared by a call to the oracle  $\mathcal{S}$ .
- Output** : Flagged states  $|0\rangle |0\rangle^{\otimes \ell+1}, |1\rangle |0\rangle^{\otimes \ell+1}$  indicating implementation of  $\tilde{U}_1 |\psi\rangle, \tilde{U}_2 |\psi\rangle$  with success probability at least  $1 - \beta$  such that  $\|U_1 - \tilde{U}_1\|, \|U_2 - \tilde{U}_2\| \leq \epsilon$  respectively where  $A = \frac{1}{2}(U_1 + U_2)$ .
- Runtime** :  $\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$  queries to  $U_A, U_A^\dagger$  using 2 additional auxiliary qubits.  $\mathcal{O}(\log(1/\beta))$  queries to the state preparation oracle  $\mathcal{S}$ .

**Procedure:**

- 1 Form the QSVT circuit of Eq. (18) for  $U_{\tilde{f}(A)}$  as shown in Fig. 6 (a) by choosing  $n = \mathcal{O}(1/\delta \log(1/\epsilon))$  to ensure  $\tilde{f}(\sigma_i)$  is an odd  $n$ -degree  $\epsilon$  approximation of the function  $f(\sigma) = \text{sign}(\sigma_i) \sqrt{1 - \sigma_i^2}, \forall i$ .
  - 2 Implement the addition of  $U_A$  and  $\pm i U_{\tilde{f}(A)}$  by using the LCU circuit in Fig. 6 (b) and perform oblivious amplitude amplification once for each  $\tilde{U}_1, \tilde{U}_2$ .
- 

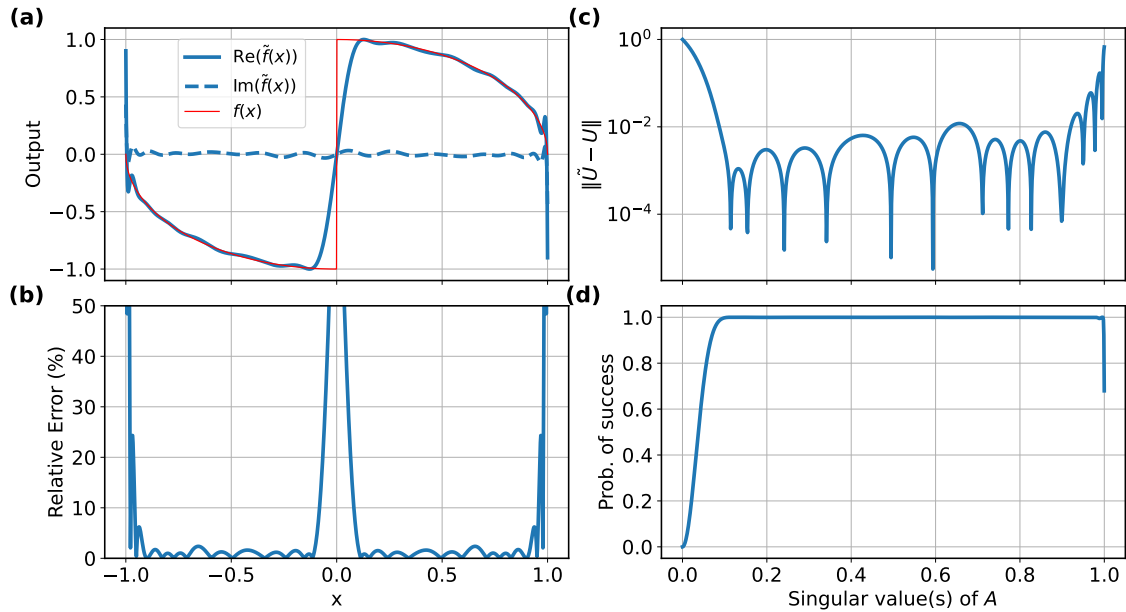


Figure 7: (a) Approximation of  $f(x) = \text{sign}(x) \sqrt{1 - x^2}$  using an odd degree-51 polynomial. Real and imaginary components of the approximation are plotted in comparison with the ideal function. (b) Relative percent error between the real and ideal curves of plot (a) as a function of the scalar input. (c) The operator norm (or max singular value) of the difference between the approximate QSVT unitary  $\tilde{U}$  and the ideal unitary  $U = A \pm i f(A)$ , plotted vs. the singular value(s) of the input matrix  $A$ . We generated 1000 two-dimensional random matrices with both singular values being identical such that  $\sigma \sim U(0, 1)$ . (d) The norm of the block-encoded matrix  $\tilde{U}$  after oblivious amplitude amplification, where deviations from 1 correspond to regions of higher error in (c).

The algorithm however fails when  $\sigma_{\min} = 0$  which can happen in many realistic cases. This problem can be addressed if we first shift all the singular values to fall within the region  $[\delta', 1 - \delta']$  such that  $\delta' > \delta$  for our choice of  $\delta'$ , perform the decomposition that

only costs  $\sim \mathcal{O}(1/\delta' \log(1/\epsilon))$  and undo the shifting. The reduction in circuit depth will come at the cost of an increase in error tolerance. However, a controlled shift of this kind requires knowing the singular vectors in advance which is expensive which we therefore avoid<sup>5</sup>. We show below that, with a slight modification in the formalism and by performing transformations on the eigenvalues, we can apply arbitrary eigenvalue shifts without explicit knowledge of the eigenbasis.

#### 4.1 Four-Unitary Decomposition

---

##### Algorithm 2: Two Unitary Decomposition for Hermitian Matrix

---

- Input** : Hermitian matrix  $H$ ,  $\delta, \epsilon > 0$  with eigenvalues of  $H$  are  $\lambda_i \in [-1 + \delta, 1 - \delta] \forall i$ , block encoded in a Unitary such that  $H = (|0\rangle^{\otimes \ell} \otimes \mathbb{1})U_H(|0\rangle^{\otimes \ell} \otimes \mathbb{1})$ . The state  $|\psi\rangle$  prepared by a call to the oracle  $\mathcal{S}$ .
- Output** : Flagged states  $|0\rangle|0\rangle^{\otimes \ell+1}, |1\rangle|0\rangle^{\otimes \ell+1}$  indicating implementation of  $\tilde{U}|\psi\rangle, \tilde{U}^\dagger|\psi\rangle$  with probability at least  $1 - \beta$  such that  $\|U - \tilde{U}\|, \|U^\dagger - \tilde{U}^\dagger\| \leq \epsilon$  respectively where  $A = \frac{1}{2}(U + U^\dagger)$ .
- Runtime** :  $\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$  queries to  $U_H, U_H^\dagger$  using 2 additional auxiliary qubits.  $\mathcal{O}(\log(1/\beta))$  queries to the state preparation oracle  $\mathcal{S}$ .

**Procedure:**

- 1 Form the QSVT circuit of Eq. (18) for  $U_{\tilde{f}(A)}$  as shown in Fig. 6 (a) by choosing  $n = \mathcal{O}(1/\delta \log(1/\epsilon))$  with modification that  $\tilde{f}(\sigma_i)$  is an even  $n$ -degree  $\epsilon$  approximation of the function  $f(\lambda_i) = \text{sign}(\lambda_i)\sqrt{1 - \lambda_i^2}, \forall i$ .
  - 2 Implement the addition of  $U_H$  and  $\pm iU_{\tilde{f}(H)}$  by using the LCU circuit in Fig. 6 (b) and perform oblivious amplitude amplification once for each  $\tilde{U}, \tilde{U}^\dagger$ .
- 

Any arbitrary matrix  $A$  can be decomposed such that  $A = H_1 + iH_2$  where  $H_1 = (A + A^\dagger)/2$  and  $H_2 = -i(A - A^\dagger)/2$  are Hermitian matrices. We show below that if we perform the two-unitary decomposition on each of the Hermitian matrices  $H_1$  and  $H_2$  such that  $H_1 = (U_1 + U_1^\dagger)/2$  and  $H_2 = (U_2 + U_2^\dagger)/2$ , we can control the shifting of the eigenvalues and therefore can control the query complexity to the oracle as desired.

A Hermitian matrix  $H$  given in a  $(1, \ell, 0)$  block encoding  $U_H$  can be decomposed into two-unitaries such that  $H = (U + U^\dagger)/2$ , where  $U = \sum_j (\lambda_j + i\sqrt{1 - \lambda_j^2}) |\lambda_j\rangle \langle \lambda_j|$  and  $\lambda_j, |\lambda_j\rangle$  are the eigenvalues and eigenvectors of  $H$ . We use a simplified version of our two-unitary decomposition algorithm, Alg. 2, to implement the decomposition unitaries  $\tilde{U}, \tilde{U}^\dagger$  such that  $\|U - \tilde{U}\|, \|U^\dagger - \tilde{U}^\dagger\| \leq \epsilon$ , with at least probability  $1 - \beta$  using  $\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$  calls to the block encoding oracle  $U_H$  where the eigenvalues  $\lambda_j \in [-1 + \delta, 1 - \delta] \forall j$ . Particularly, when given a Hermitian matrix, the QSVT algorithm will always implement both the even or odd polynomial approximations of any function in the same basis  $|\lambda_j\rangle \langle \lambda_j|$ , thus removing the restriction of choosing an odd degree polynomial to ensure the unitarity of  $U$ . Therefore, we can choose the originally-intended even function  $f(x) = \sqrt{1 - x^2}$  to be approximated by an even degree polynomial  $\tilde{f}(x)$  created by the circuit in Fig. 6 (a) with the final call to  $U_A^\dagger$  instead of  $U_A$ . Figure 8

---

<sup>5</sup>To shift each singular value  $\sigma \rightarrow (\sigma + \lambda)/\alpha$  we have to perform the transformation  $(A + \lambda WV^\dagger)/\alpha = \sum_i \frac{\sigma_i + \lambda}{\alpha} |w_i\rangle \langle v_i|$  where  $W, V$  are rows of left and right singular vectors respectively.

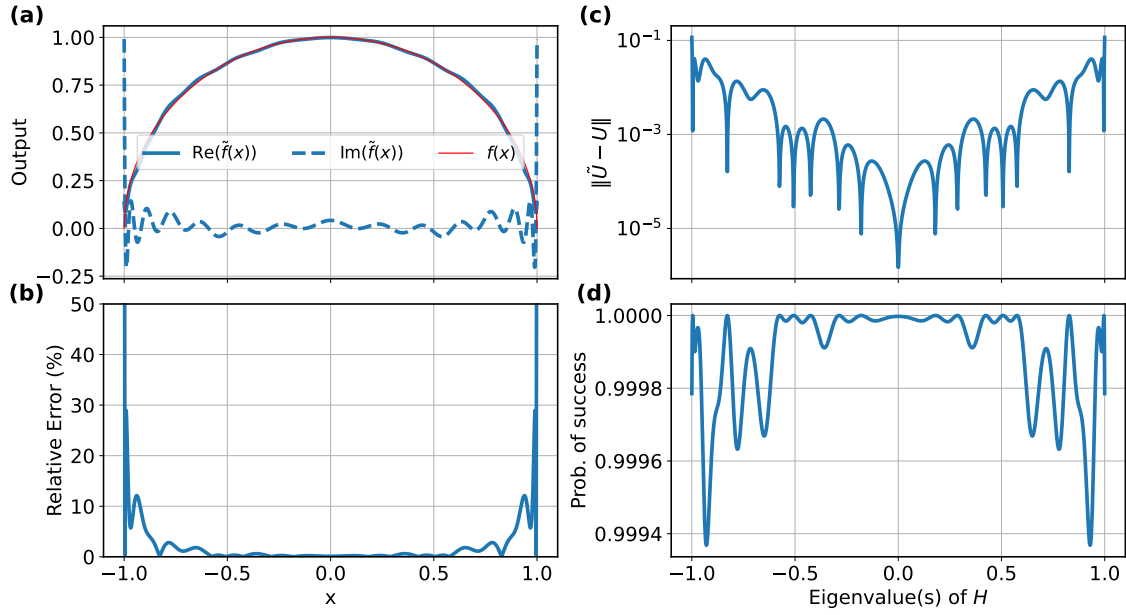


Figure 8: (a) Approximation of  $\sqrt{1-x^2}$  using an even degree-30 polynomial. Real and imaginary components of the approximation are plotted in comparison with the ideal function. (b) Relative percent error between the real and ideal curves of plot (a). as a function of the scalar input. (c) The operator norm (or max singular value) of the difference between the approximate QSVT unitary  $\tilde{U}$  and the ideal unitary  $U = H \pm i\sqrt{1-H^2}$ , plotted vs. the eigenvalue(s) of the input matrix  $H$ . We generated 1000 two-dimensional random matrices with both eigenvalues being identical such that  $\lambda \sim U(0,1)$ . (d) The norm of the block-encoded matrix  $\tilde{U}$  after oblivious amplitude amplification, where deviations from 1 correspond to regions of higher error in (c).

(a),(b) show the QSVT approximation  $\tilde{f}(x)$  for the scalar function  $f(x) = \sqrt{1-x^2}$  and the relative error using a set of 30 angles  $\{\phi_i\}$ . We can see that there is large error when  $x \rightarrow \pm 1$ . This error can be avoided by performing the two-unitary decomposition for  $H/\alpha$ ;  $H/\alpha = (U' + U'^{\dagger})/2$  for  $\alpha > 1$  to obtain  $\tilde{U}'$  such that  $\|U' - \tilde{U}'\| \leq \epsilon$ . The effect of scaling is to compress all eigenvalues from  $\lambda_i \in [-1, 1] \rightarrow \lambda_i/\alpha \in [-1/\alpha, 1/\alpha]$  and to avoid the error close to  $-1, 1$ . This method thus avoids the problem when a singular value of  $A$  is zero by using a different polynomial approximation whose error does *not* blow up when an eigenvalue is 0. We discuss the dependence of the query complexity on the scaling factor  $\alpha$  in App. D. For the example shown in Fig. 8 where  $n = 30$ , the scaling  $\alpha$  can be chosen such that the relative error shown in Fig. 8 (b) for the approximation is within the acceptable range. One could choose  $\alpha = 2$  in the decomposition of  $H/\alpha$ , and obtain  $\|U' - \tilde{U}'\| \sim 10^{-2}$  in the scaled domain  $[-1/2, 1/2]$  as shown in Fig. 8 (c).

## 4.2 Application: Dynamical Map Simulation

In summary, a Kraus operator  $A$  with non-zero singular values can be implemented as two separate unitaries using Alg. 1 or four unitaries if it has a vanishing singular value as shown in Sec 4.1 using Alg. 2. If  $A$  is Hermitian, Alg. 2 can be used directly to implement it as two-unitaries. In this subsection we show the application of the TUD algorithm to implement non-unitary Kraus operators to simulate the dynamical map and calculate expectation value of an observable. We assume access to Kraus operators in a general block encoding such as the LCU in Sec. 4.2.1, and show the calculation for generalized amplitude damping channel using four-unitary decomposition. If one assumes access to



block-encodings specifically of the Sz.-Nagy form, we show in Sec. 4.2.2 that the four-unitary decomposition can be obtained without using QSVT.

#### 4.2.1 Given General Encoding

We assume access to the block encoding of  $m$  Kraus operators  $A_k$  such that  $A_k = (\langle 0 |^{\otimes \ell} \otimes \mathbb{1}) U_{A_k} (|0\rangle^{\otimes \ell} \otimes \mathbb{1})$  where  $\ell = \log(L)$ , which may be created by techniques such as LCU as shown in Sec. 3.2.

We demonstrate our algorithm by taking the generalized amplitude damping channel as an example. The channel describes the effect of dissipation on a single qubit at a finite temperature. Let the temperature be  $T$ , with the steady-state probabilities  $p = e^{-E_0/kT}/\mathcal{Z}$  and  $1-p = e^{-E_1/kT}/\mathcal{Z}$ , where  $E_0$  and  $E_1$  are the energies of states  $|0\rangle$  and  $|1\rangle$  respectively, and where the partition function  $\mathcal{Z} = e^{-E_0/kT} + e^{-E_1/kT}$ . The parameter  $\gamma$  controls the probability of decay which in general is a function of time. The system is described by the four Kraus operators

$$A_0 = \sqrt{p} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \quad A_1 = \sqrt{p} \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}, \quad (22)$$

$$A_2 = \sqrt{1-p} \begin{bmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{bmatrix}, \quad A_3 = \sqrt{1-p} \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{bmatrix}. \quad (23)$$

We take  $E_0 = 0, E_1 = 5$  GHz,  $T = 50$  mK, which are typical values for superconducting qubits resulting in  $p = 0.982$ . Since the Kraus operators  $A_1, A_3$  have a zero singular value for all values of  $\gamma$ , we use the four unitary decomposition to implement them. Following the discussion in Sec. 4.1, each Kraus operator  $A_k$  can be written as  $A_k = H_{1k} + iH_{2k}$ . Given  $H_{1k}, H_{2k}$  in the form of  $(\alpha, \ell, 0)$  block encodings  $U_{H_{1k}}, U_{H_{2k}}$ , we can use the four unitary decomposition to express  $A_k : A_k = \frac{\alpha}{2}(U_{1k} + U_{1k}^\dagger + iU_{2k} + iU_{2k}^\dagger)$  where the scaling  $\alpha = 1.61$  in order to restrict the input eigenvalues to a sub-region having a low error profile. We chose  $n = 30$  degree polynomial approximation (or equivalently the number of oracle calls) and show that the error in Fig. 9 (a) and (b) is bounded by  $10^{-2}$ . Figure 9 (c) and (d) show the expected number of oracle calls to implement the state  $A_k |\psi\rangle / \sqrt{p_k}$  for  $k \in \{0, 1, 2, 3\}$  respectively as a function of  $\gamma$ . The expected number of calls to the block encoding  $U_{A_k}$  of each Kraus operator  $A_k$  as well as the state preparation oracle  $\mathcal{S}$  to prepare  $|\psi\rangle$  is  $\mathcal{O}(1/p_k)$  shown by the orange solid line. Instead of implementing  $A_k$ , if we implement the individual decomposition unitaries  $\tilde{U}_{1k} |\psi\rangle, \tilde{U}_{1k}^\dagger |\psi\rangle, \tilde{U}_{2k} |\psi\rangle, \tilde{U}_{2k}^\dagger |\psi\rangle$ , we see that we only require 93 calls<sup>6</sup> to the block encoding oracles  $U_{H_{1k}}, U_{H_{2k}}$  and one call to the state preparation oracle  $\mathcal{S}$  for each which is shown by the solid and dashed blue line in Fig. 9 (c-f) respectively. We show that these unitaries can be used to calculate expectation values of observables.

Given that the Kraus operator  $A_k$  has non-zero singular values, it can be implemented as two separate unitaries  $\tilde{U}_{1k}, \tilde{U}_{2k}$  using Alg. 1, which can be used to compute each term  $\langle A_k^\dagger O A_k \rangle$  separately and then summed to obtain  $\langle O \rangle$ . Each term in the sum is written as

$$\langle \widetilde{A_k^\dagger O A_k} \rangle = \frac{1}{4} \{ \langle \tilde{U}_{1k}^\dagger O \tilde{U}_{1k} \rangle + \langle \tilde{U}_{2k}^\dagger O \tilde{U}_{2k} \rangle + 2 \operatorname{Re} \langle \tilde{U}_{1k}^\dagger O \tilde{U}_{2k} \rangle \}, \quad (24)$$

where the first two terms are expectation values of  $O$  in the states  $\tilde{U}_{1k} |\psi\rangle$  and  $\tilde{U}_{2k} |\psi\rangle$  respectively. The third term can be calculated by performing Hadamard test given  $O$  as a

<sup>6</sup>We used 31 calls to obtain  $\tilde{U}_1/2$ , and additional  $2 \times 31$  to perform OAA.

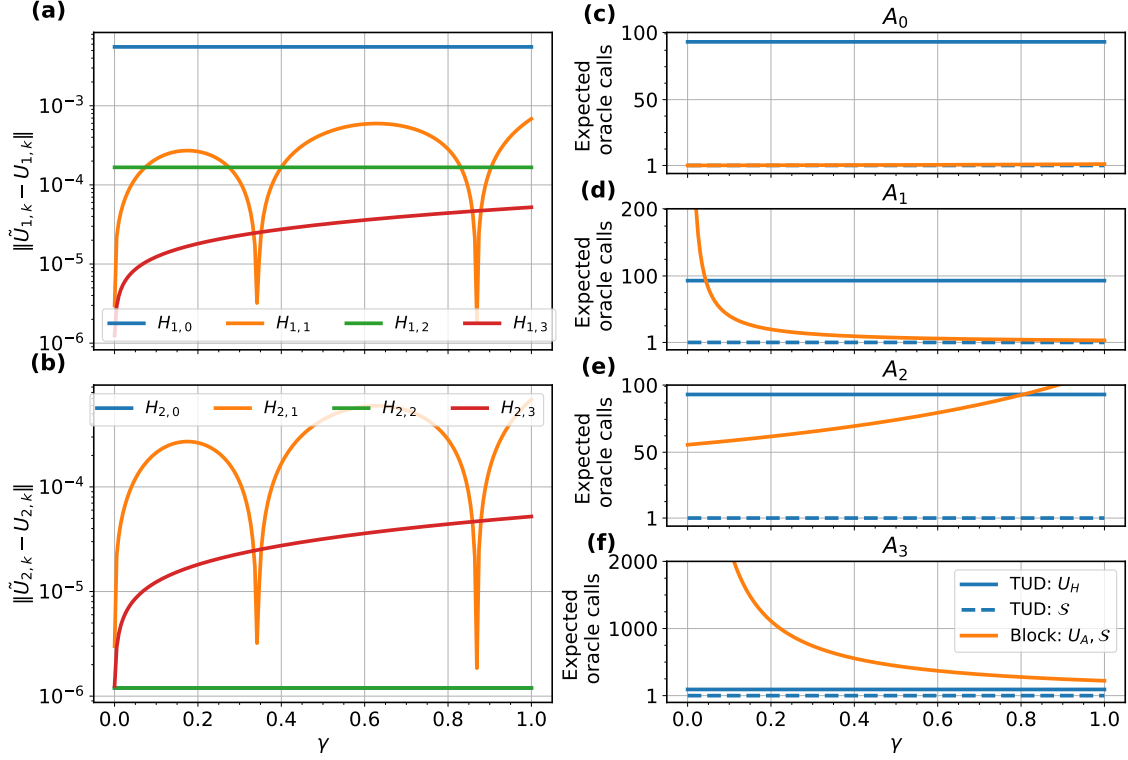


Figure 9: Error behavior and query complexity for the four Kraus operators of the generalized amplitude damping channel, as a function of the decay/excitation event probability  $\gamma$ . **(a-b)** The operator norm (or max singular value) of the difference between the approximate QSVT unitary  $\tilde{U}_{jk}$  and the ideal unitary  $U_{jk} = H_{jk} \pm i\sqrt{1-H_{jk}^2}$ , where (a)  $H_{1k} = \frac{1}{2}(A_k + A_k^\dagger)$ , (b)  $H_{2k} = \frac{i}{2}(A_k - A_k^\dagger)$ , and  $k$  indexes the Kraus operators of the channel. **(c-f)** The expected number of calls to both the state preparation and block-encoding oracles needed to obtain a successful run for each Kraus operator. To implement each decomposition unitary, the number of queries to  $U_H$  (either  $U_{H_1}, U_{H_2}$ ) and  $S$  is shown with blue solid and dashed line respectively using the TUD method. The orange curve shows expected number of queries to both  $U_A$  and  $S$  using any block encoding which is a probabilistic method.

string of Paulis or block encoded in a unitary  $U_O$ . We show in the App. B that the error in estimation is  $|\langle A_k^\dagger O A_k \rangle - \langle A_k^\dagger O A_k \rangle| \leq \epsilon$ , which comes from the use of OAA and vanishes without the use of OAA. This error can be made arbitrarily small by increasing the degree of polynomial approximation by querying the block encoding  $U_k$  for  $\mathcal{O}(\log(1/\epsilon))$  times and therefore is ignored as it has a negligible effect in the estimation of number of runs to estimate the expectation of  $\langle A_k^\dagger O A_k \rangle$ . The detailed procedure of calculation of expectation value and estimation of variance can be found in Appendix C. The condition  $\text{Var}\langle \hat{O} \rangle \leq v$ , implies that  $\sum_k (1 - (\frac{1}{4}\langle U_{1k}^\dagger O U_{1k} \rangle^2 + \frac{1}{4}\langle U_{2k}^\dagger O U_{2k} \rangle^2 + \frac{1}{2}\text{Re}\langle U_{1k}^\dagger O U_{2k} \rangle^2))/N_k \leq v$ , where  $N_k$  are the number of shots for each expectation value  $\langle A_k^\dagger O A_k \rangle$ . The number of runs however, is higher than for what we proposed in probabilistic implementation of Sec. 3.2.

If the decomposition of  $O$  is unknown, then it can also be further decomposed into two unitaries using the same algorithm:  $O = \frac{1}{2}(U_O + U_O^\dagger)$ . Each term can now be calculated as

$$\langle A_k^\dagger O A_k \rangle = \frac{1}{4} \{ \text{Re}\langle \tilde{U}_{1k}^\dagger \tilde{U}_O \tilde{U}_{1k} \rangle + \text{Re}\langle \tilde{U}_{2k}^\dagger \tilde{U}_O \tilde{U}_{2k} \rangle + \text{Re}\langle \tilde{U}_{1k}^\dagger \tilde{U}_O \tilde{U}_{2k} \rangle + \text{Re}\langle \tilde{U}_{2k}^\dagger \tilde{U}_O \tilde{U}_{1k} \rangle \} \quad (25)$$

where each term in the curly bracket is calculated by a Hadamard test, thus requiring four

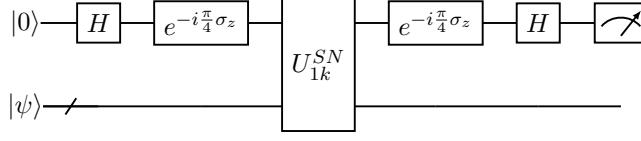


Figure 10: The circuit implements  $U_{1k} |\psi\rangle$  deterministically.

Hadamard tests regardless of the dimension of Hilbert space.

If a singular value of a matrix is zero or very close to it, we must simulate the map by decomposing each  $A_k$  into four unitaries as shown in Sec. 4.1. Each expectation value is then similarly calculated

$$\begin{aligned} \langle \widetilde{A_k^\dagger O A_k} \rangle &= \frac{1}{4} \{ \langle \widetilde{U}_{1k}^\dagger O \widetilde{U}_{1k} \rangle + \langle \widetilde{U}_{1k} O \widetilde{U}_{1k}^\dagger \rangle + \langle \widetilde{U}_{2k}^\dagger O \widetilde{U}_{2k} \rangle + \langle \widetilde{U}_{2k} O \widetilde{U}_{2k}^\dagger \rangle \} + \frac{1}{2} \text{Re} \langle \widetilde{U}_{1k} O \widetilde{U}_{1k} \rangle \\ &+ \frac{1}{2} \text{Re} \langle \widetilde{U}_{2k} O \widetilde{U}_{2k} \rangle - \frac{1}{2} \text{Im} \{ \langle \widetilde{U}_{1k} O \widetilde{U}_{2k} \rangle + \langle \widetilde{U}_{1k} O \widetilde{U}_{2k}^\dagger \rangle + \langle \widetilde{U}_{1k}^\dagger O \widetilde{U}_{2k} \rangle + \langle \widetilde{U}_{1k}^\dagger O \widetilde{U}_{2k}^\dagger \rangle \}, \end{aligned} \quad (26)$$

where each term is separately computed in parallel. The first four terms in the first curly bracket can be measured directly whereas the rest can be measured via Hadamard tests. We verify the above formula for the example of generalized amplitude damping channel by plotting the expectation value and error in App. B. The variance for the four-unitary case can be similarly derived to the two-unitary decomposition case presented before.

#### 4.2.2 Given Sz.-Nagy Encoding

In the special case where we assume access to a minimal dilation of the Sz.-Nagy form [44], we show that we do not need QSVT to obtain the two-unitary decomposition and can instead obtain it with only one call to the encoding oracle deterministically and without any error. This assumption permits the calculation of expectation values using only a single query to the Sz.-Nagy block-encoding. Each  $A_k = H_{1k} + iH_{2k}$  where  $H_{1k} = (A_k + A_k^\dagger)/2$  and  $H_{2k} = -i(A_k - A_k^\dagger)/2$  are Hermitian matrices. We assume that both the Hermitian operators  $H_{1k}, H_{2k}$  are encoded in their respective Sz.-Nagy unitaries

$$U_{1k}^{SN} = \begin{bmatrix} H_{1k} & \sqrt{I - H_{1k}^2} \\ \sqrt{I - H_{1k}^2} & -H_{1k} \end{bmatrix}, \quad U_{2k}^{SN} = \begin{bmatrix} H_{2k} & \sqrt{I - H_{2k}^2} \\ \sqrt{I - H_{2k}^2} & -H_{2k} \end{bmatrix}. \quad (27)$$

We aim to obtain the decomposition  $H_{1k} = \frac{1}{2}(U_{1k} + U_{1k}^\dagger)$ . Let the eigenvalues and eigenvectors of  $H_{1k}$  be  $\lambda_{ki}, |\lambda_{ki}\rangle$ , then we can see that the decomposition unitaries can be written as

$$U_{1k} = \bigoplus_i e^{i \arccos \lambda_{ki}} |\lambda_{ki}\rangle \langle \lambda_{ki}|, \quad (28)$$

$$U_{1k}^\dagger = \bigoplus_i e^{-i \arccos \lambda_{ki}} |\lambda_{ki}\rangle \langle \lambda_{ki}|. \quad (29)$$

One can then easily obtain the decomposition unitaries by performing rotations on the Sz.-Nagy encoding qubit as shown in Fig. 10, such that

$$U_{1k} = ((\langle 0| H e^{-i\frac{\pi}{4}\sigma_z} \otimes \mathbb{1}) U_k^{SN} ((e^{-i\frac{\pi}{4}\sigma_z} H |0\rangle) \otimes \mathbb{1})), \quad (30)$$

$$U_{1k}^\dagger = ((\langle 1| H e^{-i\frac{\pi}{4}\sigma_z} \otimes \mathbb{1}) U_k^{SN} ((e^{-i\frac{\pi}{4}\sigma_z} H |1\rangle) \otimes \mathbb{1})). \quad (31)$$

Method	State Preparation $\mathcal{S}$	Block Encoding $U_A$	Aux. Qubit
Block Encoding			$\ell$
LCU	$\mathcal{O}(1/p \log(1/\beta))$	$\mathcal{O}(1/p \log(1/\beta))$	$\ell$
Sz.-Nagy			1
TUD	$\mathcal{O}(\log(1/\beta))$	$\mathcal{O}(1/\delta \log(1/\epsilon) \log(1/\beta))$	$\ell + 2$

Table 1: Given an initial state  $|\psi\rangle$  prepared by the state preparation oracle  $\mathcal{S}$  and the operator  $A$  block encoded in  $U_A$  using  $\ell$  auxiliary qubits, the first row shows the complexity of successfully implementing the state  $A|\psi\rangle/\sqrt{p}$  with probability at least  $1 - \beta$ , for general, LCU, and Sz.-Nagy block encodings. The second row shows the complexity of implementing  $\tilde{U}_1|\psi\rangle, \tilde{U}_2|\psi\rangle$  which are  $\epsilon$  approximations of  $U_1|\psi\rangle, U_2|\psi\rangle$  such that  $A = (U_1 + U_2)/2$  with probability at least  $1 - \beta$ , where the singular values of  $A$  are  $\sigma_i \in [\delta, 1 - \delta], \forall i$ .

with  $H$  being the Hadamard gate. A similar process is done to decompose  $H_{2k} = \frac{1}{2}(U_{2k} + U_{2k}^\dagger)$  to obtain the decomposition of the arbitrary matrix  $A_k = \frac{1}{2}(U_{1k} + U_{1k}^\dagger + iU_{2k} + iU_{2k}^\dagger)$ . Therefore, given the Sz.-Nagy encoding, we decomposed an arbitrary operator into four unitaries without QSVT, and with only one oracle call and zero error. We can measure each expectation in parallel using Eq. (26) by replacing QSVT approximation unitaries  $\tilde{U}_{ik}$  with the exact unitaries  $U_{ik}$ .

Encoding the matrix in Sz.-Nagy unitary circumvents the need for QSVT due to its special structure. When given a Hermitian matrix  $H$ , the off-diagonal blocks of the Sz.-Nagy unitary already provide access to the transformation  $f(H) = \sqrt{1 - H^2}$ , which can be accessed. This special structure is even present in any other general oracle but the basis required to access the off-diagonal block is unknown in general. Additionally, we use the four-unitary decomposition and not the two-unitary decomposition here because Sz.-Nagy dilation of  $A$  has  $\sqrt{I - A^\dagger A} = \sum_i \sqrt{1 - \sigma_i^2} |v_i\rangle \langle v_i|$  and  $\sqrt{I - AA^\dagger} = \sum_i \sqrt{1 - \sigma_i^2} |w_i\rangle \langle w_i|$  as the off-diagonal blocks, whereas the desired transformation is  $f(A) = \sum_i \sqrt{1 - \sigma_i^2} |w_i\rangle \langle v_i|$ . When  $A$  is Hermitian, the Sz.-Nagy encoding provides the desired transformation as the input and output bases are the same.

## 5 Conclusion

The simulation of open quantum systems is challenging due to nature of the dynamical map that describes the evolution. In particular, the dynamical map can be represented in terms of Kraus operators, that, in general, fail to display special operator properties such as unitarity or hermiticity. Therefore, direct simulation on a standard (unitary based) quantum computer is impossible, i.e., to do so generally one must sacrifice some degree of accuracy, efficiency, or success probability. In order to circumvent those shortcomings, one needs to employ indirect methods that use a combination of auxiliary qubits, tracing-out or subsystem measurements. In this work, we analyzed prominent approaches such as the Stinespring dilation, the Sz.-Nagy dilation and the LCU technique. Additionally, we introduced a two-unitary decomposition algorithm as an alternative method for simulating Kraus operators, and hence, the entire dynamical map in parallel. A comparison between the algorithms is summarized in Table 1. All discussed algorithms share a common feature of embedding a non-unitary Kraus operator into a larger unitary matrix. This procedure requires expanding the Hilbert space with extra auxiliary qubits to accommodate the embedding, which is usually referred to as a dilation or block encoding.

We examined previously-introduced algorithms, that are of a probabilistic nature, by

providing an estimation for both classical and quantum resources. Moreover, we introduce a quantum method to calculate the expectation value of an observable and compute the variance of its estimator, necessary for understanding the total number of runs of a given algorithm to reach certain precision. We provided the TUD algorithm, which compared to previous methods allowed one to express the non-unitary Kraus operator with non-zero singular values as a sum of two unitaries which can be deterministically implemented. The TUD approach relies on the quantum singular value transformation and avoids expensive  $O(d^3)$  classical SVD overhead. Since each Kraus is expressed as a combination of two unitaries, one can implement each unitary independently without failure, which uses only a single access to the state preparation oracle. The two-unitary decomposition suffers a singularity in its error profile for matrices with vanishing singular values. We therefore provided a remedy in the form of a four-unitary decomposition, which decomposes a non-Hermitian Kraus operator into a sum of two Hermitian operators, that are then decomposed in two unitaries each. This procedure also allows to arbitrarily scale and shift eigenvalues. Even though in this paper we focused mainly on applications related to the simulation of open quantum systems, the basic algorithms discussed here rely on simulating arbitrary contractions ( $\|A\| \leq 1$ ). Thus, one can readily use the TUD algorithm in other settings that require implementing matrices without particular structure, as is often the case with classical input data accessed via an oracle. For example, methods in linear algebra such as the HHL algorithm [61], or in machine learning [62, 63] more generally, can potentially benefit from our approach. However, further work is required to identify in which cases the TUD algorithm can provide advantages over existing techniques.

The introduced TUD algorithm and reviewed methods are currently state-of-the-art techniques for simulation of open quantum systems. However, we treat this contribution not as the last word, but yet another step towards more efficient ways for simulating these complex systems. We hope that quantum computing can reveal new phenomena related to environment-system interactions that cannot be studied by other means - where the analytical treatment is confined to a handful of small-scale systems, and numerical techniques fail to scale with system's size. In particular, biological and chemical systems that are embedded in a non-trivial environment (a solution, a protein complex, etc.) so far have been treated with various approximation and semi-classical techniques or limited to small-size systems. Once we can tap into full scale OQS simulations, these restrictions can be lifted and a wealth of new physics (e.g. reaction mechanisms) could be potentially explored.

## Acknowledgements

NS did theory and JB did simulations, with equal contribution. We thank Sathyawageeswar Subramanian for helpful discussions on an early version of this work. This work was supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C2QA) under contract number DE-SC0012704 including through NASA-DOE interagency agreement SAA2-403601. NS, JB, SH, FW, JM are thankful for support from the NASA Ames Research Center, and NASA Academic Mission Services, Contract No. NNA16BD14C.

## A Total Number of Expected Runs

The number of expected runs to implement each Kraus operator  $A_k$  and therefore the state  $\frac{A_k|\psi\rangle}{\sqrt{p_k}}$  is  $\mathcal{O}(1/p_k)$ , where  $p_k = \langle\psi|A_k^\dagger A_k|\psi\rangle$ . We wish to find the minimum value of the total number of expected runs to implement all Kraus operators satisfying the constraint  $\sum_{k=1}^m p_k = 1$ . We write the cost function as

$$E = \sum_{k=1}^m \frac{1}{p_k} + \lambda \left( \sum_{k=1}^m p_k - 1 \right), \quad (32)$$

$$\frac{\partial E}{\partial p_k} = -\frac{1}{p_k^2} + \lambda = 0, \quad (33)$$

$$\frac{\partial E}{\partial \lambda} = \sum_{k=1}^m p_k - 1 = 0. \quad (34)$$

Solving the above we obtain  $p_k = p_{\min} = 1/m \forall k$ . These are the minima as  $\frac{\partial^2 E}{\partial p_k^2}|_{p_{\min}} > 0$ . Putting this back in the expected number of runs, we obtain  $E_{\min} = m^2$ . Therefore  $E \geq m^2$ . When amplitude amplification is used each  $A_k$  is implemented in  $\mathcal{O}(1/\sqrt{p_k})$ . Carrying out a similar calculation gives  $E = \sum_{i=1}^m 1/\sqrt{p_k} \geq m^{3/2}$  with amplitude amplification.

## B Error Estimation

### B.1 Probability of Success

We calculate the probability of successfully implementing  $\tilde{U}_1$  and  $\tilde{U}_2$  after OAA. The output of circuit  $C$  in the main text Fig. 6 (b) is

$$C|0\rangle|0^{\ell+1}\rangle|\psi\rangle = \frac{1}{2} \{ |0\rangle (|0^{\ell+1}\rangle\tilde{U}_1|\psi\rangle + |\Psi_+^\perp\rangle) + |1\rangle (|0^{\ell+1}\rangle\tilde{U}_2|\psi\rangle + |\Psi_-^\perp\rangle) \}. \quad (35)$$

We only have to use OAA once (which calls the QSVT routine three times) to implement either  $\tilde{U}_1$  or  $\tilde{U}_2$ , which we denote by  $\tilde{U}$  in the calculation below.

$$P_{\text{success}} = \frac{1}{4} \|(3\tilde{U}^\dagger - \tilde{U}^\dagger\tilde{U}\tilde{U}^\dagger)(3\tilde{U} - \tilde{U}\tilde{U}^\dagger\tilde{U})\| \quad (36)$$

$$= \frac{1}{4} \|\{9\tilde{U}^\dagger\tilde{U} - 6(\tilde{U}^\dagger\tilde{U})^2 + (\tilde{U}^\dagger\tilde{U})^3\}\| \quad (37)$$

$$= \frac{1}{4} \|\{9(\tilde{U}^\dagger\tilde{U} - I + I) - 6(\tilde{U}^\dagger\tilde{U} - I + I)^2 + (\tilde{U}^\dagger\tilde{U} - I + I)^3\}\| \quad (38)$$

$$\leq 1 - \frac{3}{4}\epsilon^2 + \mathcal{O}(\epsilon^3) \quad (39)$$

where we used  $\|\tilde{U}^\dagger\tilde{U} - I\| \leq \epsilon$ . This is verified in Fig. 8 (c),(d) as when  $\|\tilde{U} - U\| \sim 10^{-2}$ , the probability of success  $\sim 1 - 10^{-4}$ .

### B.2 Simulation of Dynamical Map

The aim is to simulate the dynamical map

$$\Lambda(|\psi\rangle\langle\psi|) = \sum_{k=1}^m A_k |\psi\rangle\langle\psi| A_k^\dagger. \quad (40)$$

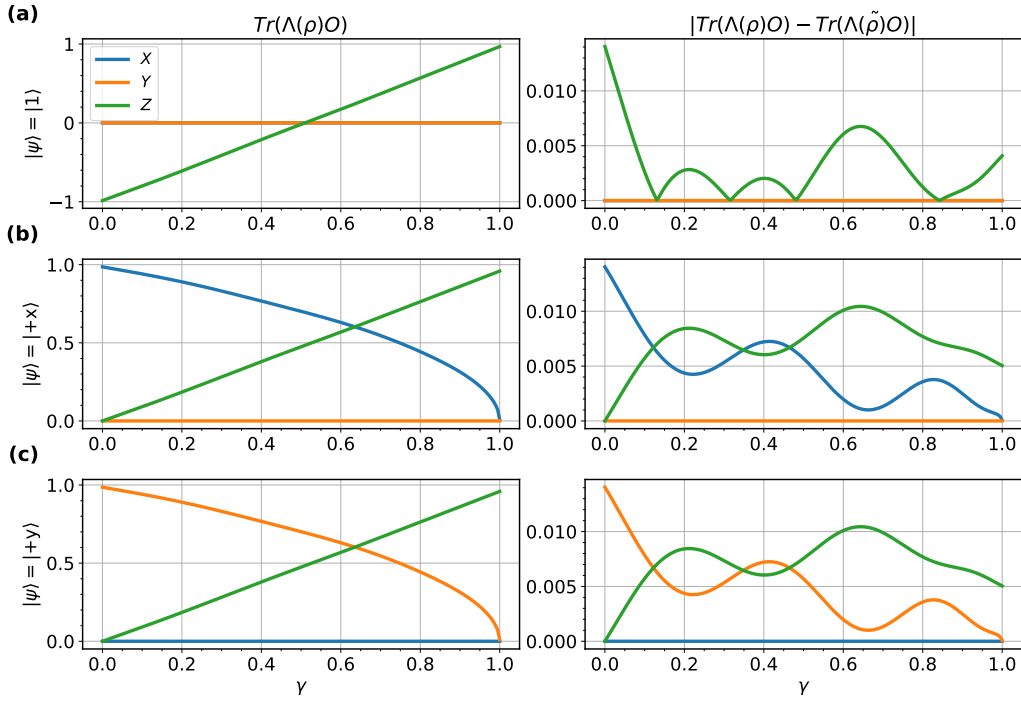


Figure 11: In row **(a)**, we plot on the left the true expectation value of a Pauli  $O \in \{X, Y, Z\}$  with respect to the evolution of the state  $|1\rangle$  under the generalized amplitude damping channel. On the right, we plot the absolute difference between the true expectation value and the approximate expectation value estimated via the four-unitary decomposition of each Kraus operator  $A_k(\gamma)$ . Rows **(b)** and **(c)** show the same for initial states  $|+x\rangle$  and  $|+y\rangle$ , respectively.

and calculate the expectation value

$$\langle O \rangle = \sum_{k=1}^m \langle A_k^\dagger O A_k \rangle. \quad (41)$$

We show the calculation below for each term  $\langle A_k^\dagger O A_k \rangle$  and thus remove  $k$  for convenience. We have a block-encoded operator  $A$  and write its ideal decomposition as  $A = \frac{1}{2}(U_1 + U_2)$ . Here  $U_1 = A + if(A)$  and  $U_2 = A - if(A)$ . We can expand the expectation as

$$\langle A^\dagger O A \rangle = \frac{1}{4} \{ \langle U_1^\dagger O U_1 \rangle + \langle U_2^\dagger O U_2 \rangle + 2 \operatorname{Re} \langle U_1^\dagger O U_2 \rangle \}. \quad (42)$$

In reality, we might have an error  $h$  in the block encoding given by  $(1, \ell, h) \tilde{A} : \|A - \tilde{A}\| \leq h$ . We use our two-unitary decomposition algorithm to implement the function  $\tilde{f}(\tilde{A}) : \|f(\tilde{A}) - \tilde{f}(\tilde{A})\| \leq \epsilon$ , where  $\tilde{f}$  is the polynomial approximation of  $f$ . Therefore we implement approximations of  $U_{1k}, U_{2k}$  which are  $\tilde{U}_{1k} = \tilde{A} + i\tilde{f}(\tilde{A}) + \Delta c$  and  $\tilde{U}_{2k} = \tilde{A} - i\tilde{f}(\tilde{A}) + \Delta c$ , where  $|\Delta c| \leq \epsilon$  comes from the error in oblivious amplitude amplification as  $\tilde{U}_{1k}, \tilde{U}_{2k}$  are not perfect unitaries. We write  $U_{1k} = \tilde{U}_{1k} + \Delta A + i\Delta f + \Delta c$  and  $U_{2k} = \tilde{U}_{2k} + \Delta A - i\Delta f + \Delta c$  where  $\Delta A = A - \tilde{A}$  and  $\Delta f = f(A) - \tilde{f}(\tilde{A})$ . We implement  $\langle \tilde{A}^\dagger O \tilde{A} \rangle = \frac{1}{4} \{ \langle \tilde{U}_1^\dagger O \tilde{U}_1 \rangle + \langle \tilde{U}_2^\dagger O \tilde{U}_2 \rangle + 2 \operatorname{Re} \langle \tilde{U}_1^\dagger O \tilde{U}_2 \rangle \}$  in the circuit. Therefore we obtain

$$\begin{aligned} \langle A^\dagger O A \rangle - \langle \tilde{A}^\dagger O \tilde{A} \rangle &= \frac{1}{4} \{ 2 \operatorname{Re} \langle \tilde{U}_1^\dagger O (\Delta A + \Delta c + i\Delta f) \rangle + 2 \operatorname{Re} \langle \tilde{U}_2^\dagger O (\Delta A + \Delta c - i\Delta f) \rangle \\ &\quad + 2 \operatorname{Re} \langle \tilde{U}_1^\dagger O (\Delta A + \Delta c - i\Delta f) \rangle + 2 \operatorname{Re} \langle \tilde{U}_2^\dagger O (\Delta A + \Delta c + i\Delta f) \rangle \}, \quad (43) \end{aligned}$$

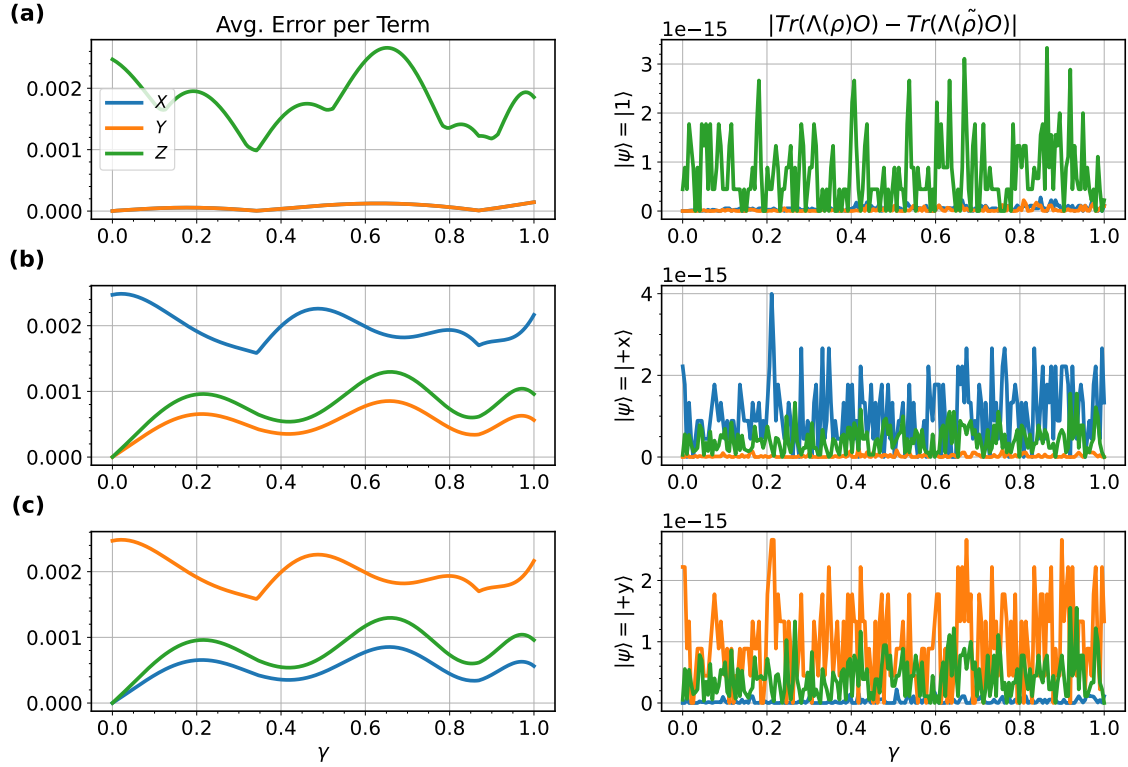


Figure 12: In row **(a)**, we plot on the right the absolute difference between the true expectation value and the approximate expectation value estimated via the four-unitary decomposition of each Kraus operator  $A_k(\gamma)$  with initial state  $|1\rangle$ , but *before OAA has been performed*. Note the small scale of the  $y$ -axis. On the left, we plot the *average* error over the 10 terms in the four-unitary decomposition of  $\langle A^\dagger O A \rangle$  (see Eq. (26)). We also average this error over the terms from the other three operators of the channel which contribute to  $\text{Tr}(\Lambda(\rho)O)$ . The scale on the right is 12 orders of magnitude larger, a clear indication that the error must cancel to obtain the scale on the left. Rows **(b)** and **(c)** show the same for initial states  $|+x\rangle$  and  $|+y\rangle$ , respectively.

where we kept the leading order terms in error.

$$\langle A^\dagger O A \rangle - \langle \widetilde{A^\dagger O A} \rangle = \text{Re}(\langle (\tilde{U}_1^\dagger + \tilde{U}_2^\dagger) O (\Delta A + \Delta c) \rangle) = 2 \text{Re}(\langle \tilde{A}^\dagger O (\Delta A + \Delta c) \rangle) \quad (44)$$

For each  $k$ , we have  $\langle A_k^\dagger O A_k \rangle - \langle \widetilde{A_k^\dagger O A_k} \rangle = 2 \text{Re}(\langle \tilde{A}_k^\dagger O (\Delta A_k + \Delta c) \rangle)$ . Taking the norm,

$$|\langle A_k^\dagger O A_k \rangle - \langle \widetilde{A_k^\dagger O A_k} \rangle| = |2 \text{Re}(\langle \tilde{A}_k^\dagger O (\Delta A_k + \Delta c) \rangle)| \leq 2(|\langle \tilde{A}_k^\dagger O \Delta A_k \rangle| + |\langle \tilde{A}_k^\dagger O \Delta c \rangle|), \quad (45)$$

and using the Cauchy-Schwarz inequality,

$$\langle \tilde{A}_k^\dagger O \Delta A_k \rangle = \langle \psi | \tilde{A}_k^\dagger O \Delta A_k | \psi \rangle \leq \sqrt{\langle \psi | \tilde{A}_k^\dagger O^2 \tilde{A}_k | \psi \rangle} \sqrt{\langle \psi | \Delta A_k^\dagger \Delta A_k | \psi \rangle} \leq h \quad (46)$$

$$\langle \tilde{A}_k^\dagger O \Delta c \rangle = \langle \psi | \tilde{A}_k^\dagger O \Delta c | \psi \rangle \leq \sqrt{\langle \psi | \tilde{A}_k^\dagger O^2 \tilde{A}_k | \psi \rangle} \sqrt{\langle \psi | \Delta c^\dagger \Delta c | \psi \rangle} \leq \epsilon \quad (47)$$

since  $\langle \psi | \tilde{A}_k^\dagger O^2 \tilde{A}_k | \psi \rangle \leq 1$  as all operators are contractions. Therefore, we obtain that  $|\langle A_k^\dagger O A_k \rangle - \langle \widetilde{A_k^\dagger O A_k} \rangle| \leq 2(\epsilon + h)$ . We can see that when block encoding error  $h = 0$ , the only error  $\epsilon$  is due to the oblivious amplitude amplification. To be clear, the error  $\Delta f$  induced by QSVT itself cancels out leaving only the error from oblivious amplitude amplification (which also originates from the QSVT error).



A similar calculation applies when we have the four-unitary decomposition instead of the two-unitary decomposition shown above. We calculate the expectation value and error for the example of generalized amplitude damping channel shown in the main text. Each term can be evaluated in parallel using the four unitary decomposition as follows

$$\begin{aligned} \widetilde{\langle A_k^\dagger O A_k \rangle} &= \frac{1}{4} \{ \langle \tilde{U}_{1k}^\dagger O \tilde{U}_{1k} \rangle + \langle \tilde{U}_{1k} O \tilde{U}_{1k}^\dagger \rangle + \langle \tilde{U}_{2k}^\dagger O \tilde{U}_{2k} \rangle + \langle \tilde{U}_{2k} O \tilde{U}_{2k}^\dagger \rangle \} + \frac{1}{2} \text{Re} \langle \tilde{U}_{1k} O \tilde{U}_{1k} \rangle \\ &+ \frac{1}{2} \text{Re} \langle \tilde{U}_{2k} O \tilde{U}_{2k} \rangle - \frac{1}{2} \text{Im} \{ \langle \tilde{U}_{1k} O \tilde{U}_{2k} \rangle + \langle \tilde{U}_{1k} O \tilde{U}_{2k}^\dagger \rangle + \langle \tilde{U}_{1k}^\dagger O \tilde{U}_{2k} \rangle + \langle \tilde{U}_{1k}^\dagger O \tilde{U}_{2k}^\dagger \rangle \}. \end{aligned} \quad (48)$$

When OAA is used, the error  $|\langle A_k^\dagger O A_k \rangle - \alpha^2 \widetilde{\langle A_k^\dagger O A_k \rangle}| \leq \alpha^2 \epsilon$ , which vanishes without OAA. Figure 11 (a) shows the total expectation value  $\langle O \rangle$  plotted on the left calculated using the four-unitary decomposition with the initial state  $|\psi\rangle = |1\rangle$ . The observable  $O$  is taken as  $X, Y, Z$  denoted by blue, orange and green lines respectively. The error in estimation is shown on the right for an infinite number of measurements which remains below  $\approx 10^{-2}$ . Similar calculation is shown for different initial states  $|+x\rangle, |+y\rangle$  in Fig. 11 (b), (c). We plot the error in Fig. 12 and show the calculation of expectation values without performing oblivious amplitude amplification to show that it cancels for each term  $\langle A_k^\dagger O A_k \rangle$ .

## C Measuring Expectation Value: Hadamard Test and Variance

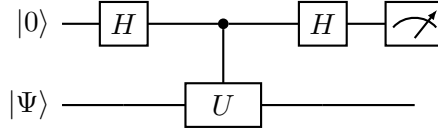


Figure 13: Hadamard test

To calculate the expectation value  $\text{Re} \langle \Psi | U | \Psi \rangle$ , we perform the Hadamard test [80] shown in Fig. 13, denoted by the circuit  $HC$ . The output of the Hadamard test circuit  $HC$  is

$$HC |0\rangle |\Psi\rangle = \frac{1}{2} \{ |0\rangle \otimes (I + U) |\Psi\rangle \} + \frac{1}{2} \{ |1\rangle \otimes (I - U) |\Psi\rangle \} \quad (49)$$

Let the random variable  $X$  take the values  $X = \pm 1$  when the auxiliary qubit is measured in state  $|0\rangle, |1\rangle$  respectively. The expected value of  $X$  is then

$$\mathbb{E}(X) = (+1)(Pr(X = +1)) + (-1)(Pr(X = -1)) \quad (50)$$

$$= \frac{1}{4} \langle \Psi | (I + U^\dagger)(I + U) - (I - U^\dagger)(I - U) | \Psi \rangle \quad (51)$$

$$= \text{Re} \langle \Psi | U | \Psi \rangle \quad (52)$$

and the variance of  $X$  is

$$\text{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2 \quad (53)$$

$$= [(+1)^2 Pr(X = +1) + (-1)^2 Pr(X = -1)] - [\text{Re} \langle \Psi | U | \Psi \rangle]^2 \quad (54)$$

$$= 1 - [\text{Re} \langle \Psi | U | \Psi \rangle]^2 \quad (55)$$

### C.1 Measurement with Block Encoding Method

We discuss the measurement of the expectation  $\langle A^\dagger O A \rangle$  and estimation of the variance. We are given  $U_A$ , a  $(1, \ell, 0)$  block encoding of  $A$ , and  $U_O$ , a  $(1, q, 0)$  block encoding of the observable  $O$  such that  $\|O\| \leq 1$ . We discuss in the main text that  $A|\psi\rangle/\sqrt{p}$  is implemented with probability  $p$ . Let this process be denoted by the Bernoulli random variable  $Y = 1, 0$ , such that  $Pr(Y = 1) = p$  for successfully implementing the state. Once we know the state is successfully implemented by measuring the auxiliary qubit  $|0\rangle^{\otimes \ell}$ , we can take  $|\Psi\rangle = |0\rangle^{\otimes n} A|\psi\rangle/\sqrt{p}$  and perform a Hadamard test with the unitary  $U = U_O$  to obtain  $\langle A^\dagger O A \rangle/p$ . Let  $Z$  denote the random variable taking values  $Z = \pm 1$  when the auxiliary qubit is measured in state  $|0\rangle, |1\rangle$  respectively in the Hadamard test. Therefore,  $Pr(Z = 1|Y = 1) = \|(I + U)|\Psi\rangle\|^2/4$  and  $Pr(Z = -1|Y = 1) = \|(I - U)|\Psi\rangle\|^2/4$ . We define the random variable  $X = YZ$ , where it can be shown that  $\mathbb{E}(X) = \langle A^\dagger O A \rangle$  and  $Var(X) = p - \langle A^\dagger O A \rangle^2$ . Given  $N$  shots of the input state  $|\psi\rangle$ , we can build an estimator of the expectation value of  $X$ :  $\widehat{\langle X \rangle} = \frac{1}{N} \sum_{i=1}^N Y_i Z_i$ , where  $Y_i, Z_i$  are the random variables associated with the  $i^{\text{th}}$  measurement of the auxiliary qubit  $|0\rangle^{\otimes \ell}$  to implement the state successfully and the auxiliary qubit in the Hadamard test respectively. Therefore,

$$\mathbb{E}(\widehat{\langle X \rangle}) = \langle \psi | A^\dagger O A | \psi \rangle, \quad Var_D(\widehat{\langle X \rangle}) = \frac{1}{N} \left( p - \langle \psi | A^\dagger O A | \psi \rangle^2 \right). \quad (56)$$

For each measurement in the worst cast we implement both block encodings  $U_A$  requiring at most  $\mathcal{O}(L^2 d^2)$  and a controlled  $U_O$  needing at most  $\mathcal{O}(Q^2 d^2)$ , where  $L = 2^\ell$  and  $Q = 2^q$ . When  $\ell = 1, q = 1$  such as in the Sz.-Nagy encoding, the measurement requires  $\mathcal{O}(d^2)$  one and two-qubit gates in the worst case.

### C.2 Measurement using the Two-Unitary Decomposition Algorithm

Alternatively, we can estimate  $\langle \psi | A^\dagger O A | \psi \rangle$  via the TUD algorithm by measuring the three expectation values below,

$$\langle A^\dagger O A \rangle = \frac{1}{4} \{ \langle U_1^\dagger O U_1 \rangle + \langle U_2^\dagger O U_2 \rangle + 2 \text{Re} \langle U_1^\dagger O U_2 \rangle \}. \quad (57)$$

Since we are interested in comparing the variances of both methods, we removed the  $\epsilon$ -approximations of the unitaries which has a negligible effect on this calculation. Each of the terms above are measured individually using three Hadamard tests and are added together to obtain  $\langle A^\dagger O A \rangle$ . Defining three random variables  $X_1, X_2, X_3 \in \{+1, -1\}$  that denote the auxiliary qubit being measured as  $|0\rangle$  or  $|1\rangle$  for each Hadamard test such that

$$\mathbb{E}(X_1) = \langle U_1^\dagger O U_1 \rangle, \quad \mathbb{E}(X_2) = \langle U_2^\dagger O U_2 \rangle, \quad \mathbb{E}(X_3) = \text{Re} \langle U_1^\dagger O U_2 \rangle. \quad (58)$$

Let the random variable  $X$  be defined such that

$$X = \frac{1}{4} \{ X_1 + X_2 + 2X_3 \}, \quad (59)$$

where we see that  $\mathbb{E}(X) = \langle A^\dagger O A \rangle$ . Given  $N'$  shots of initial state  $|\psi\rangle$ , we define the estimators

$$\widehat{\langle X_1 \rangle} = \frac{4}{N'} \sum_{i=1}^{N'/4} X_{1i}, \quad \widehat{\langle X_2 \rangle} = \frac{4}{N'} \sum_{i=1}^{N'/4} X_{2i}, \quad \widehat{\langle X_3 \rangle} = \frac{2}{N'} \sum_{i=1}^{N'/2} X_{3i}, \quad (60)$$

where  $i$  denotes the  $i^{\text{th}}$  measurement. Here  $X_1, X_2$  are measured  $N'/4$  times and  $X_3$  is measured  $N'/2$  times which is in proportion to their respective weights in Eq. (59). The variance of the estimator  $\hat{X}$  is

$$\text{Var}_{TUD}(\langle \widehat{X} \rangle) = \frac{1}{N'} \left( 1 - \frac{1}{16} (4\langle U_1^\dagger O U_1 \rangle^2 + 4\langle U_2^\dagger O U_2 \rangle^2 + 8 \text{Re}\langle U_1^\dagger O U_2 \rangle^2) \right) \quad (61)$$

## D Quantum Singular Value Transformation

In this section, we briefly summarize the quantum singular value transform and provide details regarding our specific usage of it. Readers seeking more information should consult the references cited herein.

The quantum singular value transform (QSVT) [58] is an algorithm that allows one to apply a polynomial transformation to the singular values of an arbitrary matrix which is block-encoded inside of a larger unitary. The intuition for the structure of this algorithm lies with earlier work on quantum signal processing (QSP) [81] which studied what kinds of unitary transformations are achievable by alternating between a “signal” unitary (which encodes some kind of accessible quantum information) and a single-qubit phase gate with a varying rotation angle (which provides a second control axis). A rich set of polynomial transformations of the information encoded by the signal unitary was obtained, and this behavior was later extended beyond the single-qubit case, where it was shown that the entire spectrum of a block-encoded Hermitian matrix  $H$  could be processed similarly [59]. To achieve this, one first doubles the dimension of the Hilbert space by appending an auxiliary qubit, and then transforms the block-encoding into an iterate that takes the form of a product of two reflection operators. By Jordan’s Lemma, each 1-dimensional eigenspace of  $H$  then defines an  $SU(2)$  subspace (hence the name “qubitization”) in which one can generate coherent rotations without mixing into any of the other eigen(sub)spaces. With additional access to controlled-phase gates, one can achieve the same kind of “two-axis” control within each eigen(sub)space, and thus effect the same polynomial transformations of quantum signal processing on each eigenvalue, and therefore the matrix itself, in parallel. Lastly, these results were further generalized [58] to arbitrary (even rectangular) matrices, where by carefully considering the left and right singular vector spaces, the same behavior can be derived with even fewer assumptions. The importance of these results is well-motivated by recent observations that many quantum algorithms with no overt similarities and widely differing use-cases can be generalized as instances of QSVT [58, 5].

Let us begin by assuming that we have a matrix with SVD form  $A = W\Sigma V = \sum_i \sigma_i |w_i\rangle\langle v_i|$  which is block-encoded within a larger unitary  $U$ , given by  $A = \tilde{\Pi}U\Pi$ . The projector  $\Pi$  projects into the right singular vector space, i.e.  $\text{img}(\Pi) = \sum_i |v_i\rangle\langle v_i|$ , and  $\text{img}(\tilde{\Pi}) = \sum_i |w_i\rangle\langle w_i|$  thus projects into the left singular vector space. These projectors then locate  $A$  within  $U$ , and with access to controlled versions of  $\Pi$  and  $\tilde{\Pi}$  (i.e. rotations conditioned on lying in the span of  $\Pi$  or  $\tilde{\Pi}$ ), one can interleave them with  $U$  to produce the desired polynomial transformation:  $P(A) = \sum_i P(\sigma_i) |w_i\rangle\langle v_i|$ . When restricted to the appropriate invariant  $SU(2)$  subspace, the actions of  $U$  and the controlled rotations have the form

$$[U]_{\mathcal{B}_i} = \begin{bmatrix} \sigma_i & \sqrt{1 - \sigma_i^2} \\ \sqrt{1 - \sigma_i^2} & -\sigma_i \end{bmatrix} \quad [e^{i\phi(2\Pi - I)}]_{\mathcal{B}_i} = \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix} \quad (62)$$

where  $\mathcal{B}_i$  denotes the  $SU(2)$  subspace of the  $i$ -th singular value. The operator  $[U]_{\mathcal{B}_i}$  is often referred to as the “iterate” or the signal operator, and the operator  $[e^{i\phi(2\Pi - I)}]_{\mathcal{B}_i}$  is

sometimes called the signal processing operator or  $S(\phi)$  [5]. The actions of  $U^\dagger$  and  $e^{i\phi(2\tilde{\Pi}-I)}$  are identical but with swapped input/output spaces, with respect to the above operators. Referring to Definition 8 of [58], we then denote an alternating phase modulation sequence  $U_\Phi^{(APM)}$  by

$$U_\Phi^{(APM)} = \begin{cases} U e^{i\phi_n(2\Pi-I)} \dots U^\dagger e^{i\phi_2(2\tilde{\Pi}-I)} U e^{i\phi_1(2\Pi-I)} & \text{if } n \text{ is odd} \\ U^\dagger e^{i\phi_n(2\tilde{\Pi}-I)} \dots U^\dagger e^{i\phi_2(2\tilde{\Pi}-I)} U e^{i\phi_1(2\Pi-I)} & \text{if } n \text{ is even} \end{cases} \quad (63)$$

The set of angles  $\{\phi_n\}$  determines the resulting degree- $n$  polynomial, and can be efficiently computed [75, 76, 77]. We see that the degree of the polynomial determines the number of alternations between  $U$  and  $U^\dagger$ , and thus also determines the final output space (left if odd, right if even). Once this sequence is constructed, we need only to project into the required block to obtain the transformed operator  $P(A)$ , where  $P$  denotes the desired polynomial (Theorem 10 [58]):

$$P(\tilde{\Pi}U\Pi) = \begin{cases} \tilde{\Pi}U_\Phi^{(APM)}\Pi & \text{if } n \text{ is odd} \\ \Pi U_\Phi^{(APM)}\Pi & \text{if } n \text{ is even} \end{cases} \quad (64)$$

The action of the polynomial needs only be correct over the domain  $x \in [-1, 1]$  due to the fact that  $\|A\| < 1$  for  $A$  to be properly embedded in  $U$ , and moreover that singular values are always bounded between 0 and 1. Because the polynomial produced by QSVT is in general complex ( $P \in \mathbb{C}[x]$ ), it is important to know how to implement real polynomials ( $P \in \mathbb{R}[x]$ ). This can be achieved by noting that the angle sequence  $\{-\phi_n\}$  produces the conjugate polynomial  $P^*$ . One then only needs to perform an LCU-style addition of  $U_\Phi^{(APM)}$  and  $U_{-\Phi}^{(APM)}$  to produce a unitary that block-encodes  $\text{Re}(P)$ , because the imaginary components will cancel out [58]:

$$P_{\text{real}}(\tilde{\Pi}U\Pi) = \begin{cases} (\langle + | \otimes \tilde{\Pi} \langle 0 | \langle 0 | \otimes U_\Phi^{(APM)} + |1\rangle\langle 1| \otimes U_{-\Phi}^{(APM)}) (|+\rangle \otimes \Pi) & \text{if } n \text{ is odd} \\ (\langle + | \otimes \Pi \langle 0 | \langle 0 | \otimes U_\Phi^{(APM)} + |1\rangle\langle 1| \otimes U_{-\Phi}^{(APM)}) (|+\rangle \otimes \Pi) & \text{if } n \text{ is even} \end{cases} \quad (65)$$

The same result can be achieved without the extra LCU qubit by the observation in Appendix B of [77], where one can simply use a  $|+\rangle$  state to control the QSVT auxiliary qubit, and then measure in the  $x$ -basis to add the two cases.

In general, there exist several conventions in the literature for the QSVT formalism, due to flexibility in how one chooses to define the unitaries used to generate the polynomials. In [5], the authors define four main ingredients, already alluded to above: a signal operator  $W$  (the block-encoding), a set of phase angles  $\{\phi_n\}$ , a signal *processing* operator  $S(\phi)$  (the projector-controlled phase gates), and a basis  $M$  in which we measure the QSVT auxiliary qubit to obtain the encoded output. We can then use  $(W, S, M)$  to denote a particular QSP convention. The form of  $W$  used in Eq. (62) is known as the  $R$ -iterate (for “reflection”), and results from the particular relations between the left and right spaces proved by [58]. The  $R$ -iterate can be converted to other conventions, such as  $W_x$  or  $W_z$ , by appropriate single-qubit rotations in the invariant subspace(s). The equivalence between different QSP conventions, such as  $(W_z, S_x, \langle 0 | \cdot | 0 \rangle)$  and  $(W_x, S_z, \langle + | \cdot | + \rangle)$ , stems from the choice of using either the Laurent or Chebyshev bases for representing polynomials.

## D.1 QSVT for $f(A)$

In this section, we provide details regarding our specific implementation of the function  $\sqrt{1-x^2}$  using open-source Python repository `pyqsp` [78]. We choose the  $(R, S_z, \langle 0 | \cdot | 0 \rangle)$

convention, due to the fact that we presume access to block encodings of the more general format

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad (66)$$

where the  $R$ -iterate form implicitly holds by the results of [58]. If one assumes direct access to block-encodings of the Sz.-Nagy form, one immediately recovers the  $R$ -iterate in the computational basis and can take advantage of measuring in the  $\langle + | \cdot | + \rangle$  convention, which is more convenient for representing real polynomials. In fact, such an assumption immediately gives us our results without needing QSVT at all (Sec. 4.2.2). Other oracles for the block-encoding, such as LCU, require the  $\langle 0 | \cdot | 0 \rangle$  convention, which results in complex polynomial outputs that must have their imaginary components removed in the manner of Eq. (65) if one desires a real output. In particular, this convention is much less convenient because the polynomial outputs must obey two additional constraints [5, 58]

$$|P(\pm 1)| = 1 \quad (67)$$

$$P(ix)P^*(ix) \geq 1 \quad (68)$$

where  $P(\cdot)$  is the implemented polynomial. Although these constraints result in higher approximation error for a given degree, we choose to use this convention because it is compatible with more realistic oracle models such as LCU.

The angles output by `pyqsp` are in the  $W_x$  form, and must be converted to the  $R$ -form by the following map [5]:

$$\phi_i \rightarrow \begin{cases} \phi_i + \frac{\pi}{4}(2L - 1) & \text{if } i = 0 \\ \phi_i - \frac{\pi}{4} & \text{if } i = L \\ \phi_i - \frac{\pi}{2} & \text{else} \end{cases} \quad (69)$$

The final circuit for the QSVT is shown in Fig. 14. As described in the main text, we must start in the right space and end in the left space to obtain the desired map  $\sum_i \sqrt{1 - \sigma_i^2} |w_i\rangle \langle v_i|$ . This necessitates either an odd polynomial approximation to the even function  $\sqrt{1 - x^2}$ , or a decomposition of  $A$  into Hermitian and anti-Hermitian components so that one resolves the input/output space issue. For reproducibility, the 51 angles obtained via `pyqsp` for the QSVT of Fig. 7 (*before* using the above conversion) are

0.5945105	0.60960954	0.7298763	2.8723814	1.550056	1.2478894
2.861402	0.76238096	2.014291	1.1355253	2.2245793	3.4903798
3.044659	0.3553313	3.0701022	0.43656695	0.7831127	2.5623524
2.491012	0.32272875	3.107868	1.2339981	1.6407303	0.9969318
1.0959318	2.647887	2.8997867	2.795346	0.04209794	2.7231102
1.8544827	1.3496944	2.506372	1.5447122	1.5370306	-0.6786906
0.74581164	0.21720853	2.302732	0.7340103	0.32469824	3.3565025
2.1487994	2.6228967	0.3943159	0.22044371	-0.06635685	0.17362866
3.080445	0.7867449	1.2483609	0.24187739]		

and the 30 angles for the QSVT of Fig. 8 are

2.879597	3.2655551	3.1311133	0.20452265	3.5637999	3.1725397
2.182688	0.55169135	2.6656814	3.470747	3.7887444	0.34566507

-0.42822552	3.759092	0.41655388	-0.68946004	2.8910365	0.09444891
2.773645	0.6924566	2.0532732	2.8459213	3.2667234	0.01168075
3.6077757	3.4687052	-0.6682357	-0.05417808	-0.09065018	3.120173
1.693089]					

## D.2 Complexity for Scaling in Two-Unitary Decomposition

The complexity for step 1 in the two-unitary decomposition algorithm that involves the use of QSVT to create the polynomial approximation to the matrix-function  $f(A) = \sum_i \text{sign}(\sigma_i) \sqrt{1 - \sigma_i^2} |w_i\rangle \langle v_i|$  depends on the number of oracle calls to the block encoding oracle  $U_A$ . Since each call to the  $U_A$  increases the degree of the polynomial by one, the complexity by finding the degree  $n$  of the  $\epsilon$  approximating scalar polynomial  $\tilde{f}(x)$  for the scalar function  $f(x) = \text{sign}(x) \sqrt{1 - x^2}$ .

Let  $f_1(x) = \text{sign}(x)$  and  $f_2(x) = \sqrt{1 - x^2}$ , such that  $f(x) = f_1(x)f_2(x)$ . Then, Corollary 6 of the Ref. [82] or Ref. [58, 5] show that

$$\max_{x \in [-1, -\delta_1/2] \cup [\delta_1/2, 1]} |f_1(x) - \tilde{f}_1(x)| \leq \epsilon, \quad (70)$$

where  $\tilde{f}_1(x)$  is a polynomial of degree  $n_1 = \mathcal{O}(1/\delta_1 \log(1/\epsilon))$ . We can obtain the condition for the approximation of the function  $f_2(x)$  by a  $n_2$ -degree polynomial  $\tilde{f}_2(x)$  over the domain  $x \in [-1 + \delta_2, 1 - \delta_2]$  such that

$$|f_2(x) - \tilde{f}_2(x)| \leq \sum_{i=n_2}^{\infty} |\gamma_i x^i| \leq (1 - \delta_2)^{n_2} B \leq e^{-n_2 \delta} B \leq \epsilon. \quad (71)$$

where  $\gamma_i$  are the Taylor expansion coefficients and  $B = \sum_{i=0}^{\infty} |\gamma_i|$  [83]. Then  $n_2 \geq (1/\delta_2 \log(B/\epsilon))$ <sup>7</sup>, since we are concerned with the order of magnitude estimate in terms of  $\delta_2$  and  $\epsilon$ ,  $n_2 = \mathcal{O}(1/\delta_2) \log(1/\epsilon)$ . Let the approximating polynomial of the function  $f(x) = f_1(x)f_2(x)$  be given by  $\tilde{f}(x) = \tilde{f}_1(x)\tilde{f}_2(x)$ , then

$$|f(x) - \tilde{f}(x)| = |f_1(x)f_2(x) - \tilde{f}_1(x)\tilde{f}_2(x)| \quad (72)$$

$$= |\tilde{f}_1(x)(f_2(x) - \tilde{f}_2(x)) + \tilde{f}_2(x)(f_1(x) - \tilde{f}_1(x)) + (f_1(x) - \tilde{f}_1(x))(f_2(x) - \tilde{f}_2(x))| \quad (73)$$

$$\leq 2\epsilon, \quad (74)$$

where we used  $|\tilde{f}_1(x)|, |\tilde{f}_2(x)| \leq 1$  and neglected the  $\mathcal{O}(\epsilon^2)$  error. The degree of approximating polynomial  $\tilde{f}(x)$  is therefore  $n = n_1 + n_2 = \mathcal{O}(1/\delta \log(1/\epsilon))$ , where we chose  $\delta = \min\{\delta_1, \delta_2\}$  such that  $x \in [-1 + \delta, -\delta] \cup [\delta, 1 - \delta]$ .

## D.3 Complexity for Scaling in Four Unitary Decomposition

We chose the even function  $f(x) = \sqrt{1 - x^2}$  to be approximated by an even polynomial to avoid a discontinuity at 0 in the error profile. The degree  $n$  of approximating polynomial can be found as

$$|f(x) - \tilde{f}(x)| \leq \sum_{i=n}^{\infty} |\gamma_i \alpha^i x^i / \alpha^i| \leq e^{-n\delta} / \alpha^n B_\alpha, \quad (75)$$

<sup>7</sup>A better bound may be obtained by expanding in Chebyshev polynomials.

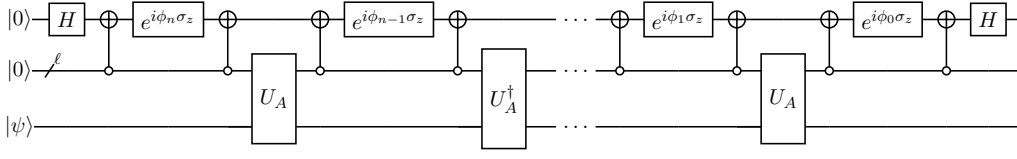


Figure 14: QSVT circuit used for applying real and odd polynomial transformation to the singular values of an arbitrary matrix  $A$ , in the convention used by pyqsp. If one desires an even polynomial, the final application of the block-encoding will be  $U^\dagger$  instead of  $U$ .

where we used that  $x/\alpha \in [(-1+\delta)/\alpha, (1-\delta)/\alpha]$  and  $B_\alpha \leq \sum_{i=n}^{\infty} |\gamma_i \alpha^i|$  is an upper bound on the weighted Taylor series coefficients. We find that  $n \geq 1/(\log \alpha + \delta) \log(B_\alpha/\epsilon)$  for the  $\epsilon$  approximation to hold within the domain of  $x/\alpha$ . We see that when we do not know  $\delta$  or  $\delta = 0$ , we can choose  $n \geq 1/\log(\alpha) \log(B_\alpha/\epsilon) \geq 1/(\log \alpha + \delta) \log(B_\alpha/\epsilon)$  removing the dependence on  $\delta$ .

However, if we have a matrix that is sufficiently sparse implying only few non-zero eigenvalues, the compression of the eigenvalues will not affect the distribution of eigenvalues much as it is not dense. For this special case,  $H/\alpha = H'$  where  $H'$  is another sparse matrix such that its eigenvalues  $\lambda'_i \in [-1+\delta', 1-\delta']$  where  $\delta' = 1-1/\alpha$ . The complexity in this case is then  $\mathcal{O}(1/\delta' \log(1/\epsilon))$ . The number of oracle calls  $n$  decreases with increasing  $\alpha$  for the  $\epsilon$  approximation valid over a smaller domain. However this comes at the cost of the approximation error, as when we scale back we will have to use  $\alpha \tilde{U}'$ ,  $\alpha \tilde{U}'^\dagger$  in calculation involving  $H$  and therefore the error will scale with  $\alpha$ .

#### D.4 Intuition for Two-Unitary Decomposition

Rather than appeal directly to SVD to derive the two-unitary decomposition, one can motivate the approach given that we have QSVT in mind. Since we prefer to work with unitaries, one might ask what kind of processing is needed to produce a unitary from  $A$ , using only basic block-matrix arithmetic (such as what is defined in Section 5 of [58]). Multiplication of  $A$  by some matrix can indeed produce a unitary via the polar decomposition of  $A$ :

$$A = UP \rightarrow A^\dagger A = P^\dagger U^\dagger U P \rightarrow P = \sqrt{A^\dagger A} \quad (76)$$

If  $A$  is invertible and if  $P$  is the positive-definite square root of  $A$ , then we can obtain a unitary  $U = AP^{-1} = A(\sqrt{A^\dagger A})^{-1}$ . Performing matrix inversion is costly through QSVT, so we may consider trying addition rather than multiplication:

$$U = A + iM \quad (77)$$

$$U^\dagger U = (A^\dagger - iM^\dagger)(A + iM) \quad (78)$$

$$= A^\dagger A + iA^\dagger M - iM^\dagger A + M^\dagger M \quad (79)$$

Given that we must always process a block-encoding of  $A$ , we may assume that  $M$  is some function of  $A$ , i.e. that  $M = f(A) = Wf(\Sigma)V^\dagger$ . Thus we may write

$$A^\dagger M - M^\dagger A = V\Sigma W^\dagger Wf(\Sigma)V^\dagger - Vf(\Sigma)W^\dagger W\Sigma V^\dagger \quad (80)$$

$$= V(\Sigma f(\Sigma) - f(\Sigma)\Sigma)V^\dagger = 0 \quad (81)$$

which follows from the fact that  $\Sigma$  is diagonal and thus commutes with  $f(\Sigma)$ . We are left

with

$$I = A^\dagger A + M^\dagger M \quad (82)$$

$$= V \Sigma^2 V^\dagger + V f(\Sigma)^2 V^\dagger \quad (83)$$

$$f(\Sigma) = \sqrt{I - \Sigma^2} \quad (84)$$

which proves that  $M = \sqrt{I - A^2}$  is sufficient to obtain a unitary, and avoids the use of matrix inversion.

## References

- [1] Yuri Manin. Computable and uncomputable. *Sovetskoye Radio, Moscow*, 128, 1980.
- [2] Richard P. Feynman. Simulating physics with computers. *Int. j. Theor. phys.*, 21(6/7), 1982. DOI: [10.1007/BF02650179](https://doi.org/10.1007/BF02650179). URL <https://doi.org/10.1007/BF02650179>.
- [3] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [4] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996. DOI: [10.1126/science.273.5278.1073](https://doi.org/10.1126/science.273.5278.1073). URL <https://www.science.org/doi/abs/10.1126/science.273.5278.1073>.
- [5] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, Dec 2021. DOI: [10.1103/PRXQuantum.2.040203](https://doi.org/10.1103/PRXQuantum.2.040203). URL <https://link.aps.org/doi/10.1103/PRXQuantum.2.040203>.
- [6] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Rev. Mod. Phys.*, 86:153–185, Mar 2014. DOI: [10.1103/RevModPhys.86.153](https://doi.org/10.1103/RevModPhys.86.153). URL <https://link.aps.org/doi/10.1103/RevModPhys.86.153>.
- [7] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 2016. DOI: [10.1038/npjqi.2015.23](https://doi.org/10.1038/npjqi.2015.23). URL <https://www.nature.com/articles/npjqi201523>.
- [8] John Preskill. Quantum computing 40 years later. *arXiv:2106.10522*, 2021. URL <https://arxiv.org/abs/2106.10522>.
- [9] Heinz-Peter Breuer, Francesco Petruccione, et al. *The theory of open quantum systems*. Oxford University Press on Demand, 2002.
- [10] Goran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, 1976. DOI: [10.1007/BF01608499](https://doi.org/10.1007/BF01608499).
- [11] Vittorio Gorini, Andrzej Kossakowski, and Ennackal Chandy George Sudarshan. Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5):821–825, 1976. DOI: [10.1063/1.522979](https://doi.org/10.1063/1.522979). URL <https://aip.scitation.org/doi/10.1063/1.522979>.
- [12] Laszlo Gyongyosi, Sandor Imre, and Hung Viet Nguyen. A survey on quantum channel capacities. *IEEE Communications Surveys & Tutorials*, 20(2):1149–1205, 2018. DOI: [10.1109/COMST.2017.2786748](https://doi.org/10.1109/COMST.2017.2786748).
- [13] Filippo Caruso, Vittorio Giovannetti, Cosmo Lupo, and Stefano Mancini. Quantum channels and memory effects. *Rev. Mod. Phys.*, 86:1203–1259, Dec 2014. DOI: [10.1103/RevModPhys.86.1203](https://doi.org/10.1103/RevModPhys.86.1203). URL <https://link.aps.org/doi/10.1103/RevModPhys.86.1203>.
- [14] Lorenza Viola, Emanuel Knill, and Seth Lloyd. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.*, 82:2417–2421, Mar 1999. DOI: [10.1103/PhysRevLett.82.2417](https://doi.org/10.1103/PhysRevLett.82.2417). URL <https://link.aps.org/doi/10.1103/PhysRevLett.82.2417>.



- [15] Dieter Suter and Gonzalo A. Álvarez. Colloquium: Protecting quantum information against environmental noise. *Rev. Mod. Phys.*, 88:041001, Oct 2016. DOI: [10.1103/RevModPhys.88.041001](https://doi.org/10.1103/RevModPhys.88.041001). URL <https://link.aps.org/doi/10.1103/RevModPhys.88.041001>.
- [16] Easwar Magesan, Daniel Puzzioli, Christopher E. Granade, and David G. Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Phys. Rev. A*, 87:012324, Jan 2013. DOI: [10.1103/PhysRevA.87.012324](https://doi.org/10.1103/PhysRevA.87.012324). URL <https://link.aps.org/doi/10.1103/PhysRevA.87.012324>.
- [17] Paolo Zanardi, Jeffrey Marshall, and Lorenzo Campos Venuti. Dissipative universal lindbladian simulation. *Phys. Rev. A*, 93:022312, Feb 2016. DOI: [10.1103/PhysRevA.93.022312](https://doi.org/10.1103/PhysRevA.93.022312). URL <https://link.aps.org/doi/10.1103/PhysRevA.93.022312>.
- [18] Marko Žnidarič, Tomaž Prosen, Giuliano Benenti, Giulio Casati, and Davide Rossini. Thermalization and ergodicity in one-dimensional many-body open quantum systems. *Phys. Rev. E*, 81:051135, May 2010. DOI: [10.1103/PhysRevE.81.051135](https://doi.org/10.1103/PhysRevE.81.051135). URL <https://link.aps.org/doi/10.1103/PhysRevE.81.051135>.
- [19] Michael J Kastoryano and Fernando GSL Brandao. Quantum Gibbs samplers: The commuting case. *Communications in Mathematical Physics*, 344(3):915–957, 2016. DOI: [10.1007/s00220-016-2641-8](https://doi.org/10.1007/s00220-016-2641-8).
- [20] Iztok Pižorn. One-dimensional Bose-Hubbard model far from equilibrium. *Phys. Rev. A*, 88:043635, Oct 2013. DOI: [10.1103/PhysRevA.88.043635](https://doi.org/10.1103/PhysRevA.88.043635). URL <https://link.aps.org/doi/10.1103/PhysRevA.88.043635>.
- [21] Tomaž Prosen and Marko Žnidarič. Matrix product simulations of non-equilibrium steady states of quantum spin chains. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(02):P02035, 2009. DOI: [10.1088/1742-5468/2009/02/p02035](https://doi.org/10.1088/1742-5468/2009/02/p02035). URL <https://doi.org/10.1088/1742-5468/2009/02/p02035>.
- [22] Tomaž Prosen. Open xxz spin chain: Nonequilibrium steady state and a strict bound on ballistic transport. *Phys. Rev. Lett.*, 106:217206, May 2011. DOI: [10.1103/PhysRevLett.106.217206](https://doi.org/10.1103/PhysRevLett.106.217206). URL <https://link.aps.org/doi/10.1103/PhysRevLett.106.217206>.
- [23] Giuliano Benenti, Giulio Casati, Tomaž Prosen, Davide Rossini, and Marko Žnidarič. Charge and spin transport in strongly correlated one-dimensional quantum systems driven far from equilibrium. *Phys. Rev. B*, 80:035110, Jul 2009. DOI: [10.1103/PhysRevB.80.035110](https://doi.org/10.1103/PhysRevB.80.035110). URL <https://link.aps.org/doi/10.1103/PhysRevB.80.035110>.
- [24] Tomaž Prosen and Marko Žnidarič. Diffusive high-temperature transport in the one-dimensional hubbard model. *Phys. Rev. B*, 86:125118, Sep 2012. DOI: [10.1103/PhysRevB.86.125118](https://doi.org/10.1103/PhysRevB.86.125118). URL <https://link.aps.org/doi/10.1103/PhysRevB.86.125118>.
- [25] Susana F Huelga and Martin B Plenio. Vibrations, quanta and biology. *Contemporary Physics*, 54(4):181–207, 2013. DOI: [10.1080/00405000.2013.829687](https://doi.org/10.1080/00405000.2013.829687).
- [26] Zixuan Hu, Kade Head-Marsden, David A. Mazziotti, Prineha Narang, and Sabre Kais. A general quantum algorithm for open quantum dynamics demonstrated with the Fenna-Matthews-Olson complex. *Quantum*, 6:726, May 2022. ISSN 2521-327X. DOI: [10.22331/q-2022-05-30-726](https://doi.org/10.22331/q-2022-05-30-726). URL <https://doi.org/10.22331/q-2022-05-30-726>.
- [27] Sarah Mostame, Patrick Rebentrost, Alexander Eisfeld, Andrew J Kerman, Dimitris I Tsomokos, and Alán Aspuru-Guzik. Quantum simulator of an open quantum system using superconducting qubits: exciton transport in photosynthetic complexes. *New Journal of Physics*, 14(10):105013, 2012. DOI: [10.1088/1367-2630/14/10/105013](https://doi.org/10.1088/1367-2630/14/10/105013). URL <https://doi.org/10.1088/1367-2630/14/10/105013>.

- [28] I. Sinayskiy, A. Marais, F. Petruccione, and A. Ekert. Decoherence-assisted transport in a dimer system. *Phys. Rev. Lett.*, 108:020602, Jan 2012. DOI: [10.1103/PhysRevLett.108.020602](https://doi.org/10.1103/PhysRevLett.108.020602). URL <https://link.aps.org/doi/10.1103/PhysRevLett.108.020602>.
- [29] Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature physics*, 5(9):633–636, 2009. DOI: [10.1038/nphys1342](https://doi.org/10.1038/nphys1342). URL <https://www.nature.com/articles/nphys1342>.
- [30] Paolo Zanardi and Lorenzo Campos Venuti. Coherent quantum dynamics in steady-state manifolds of strongly dissipative systems. *Phys. Rev. Lett.*, 113:240406, Dec 2014. DOI: [10.1103/PhysRevLett.113.240406](https://doi.org/10.1103/PhysRevLett.113.240406). URL <https://link.aps.org/doi/10.1103/PhysRevLett.113.240406>.
- [31] Jan Carl Budich, Peter Zoller, and Sebastian Diehl. Dissipative preparation of chern insulators. *Phys. Rev. A*, 91:042117, Apr 2015. DOI: [10.1103/PhysRevA.91.042117](https://doi.org/10.1103/PhysRevA.91.042117). URL <https://link.aps.org/doi/10.1103/PhysRevA.91.042117>.
- [32] Sebastian Diehl, Enrique Rico, Mikhail A Baranov, and Peter Zoller. Topology by dissipation in atomic quantum wires. *Nature Physics*, 7(12):971–977, 2011. DOI: [10.1038/nphys2106](https://doi.org/10.1038/nphys2106). URL <https://www.nature.com/articles/nphys2106>.
- [33] Charles-Edouard Bardyn, Mikhail A Baranov, Christina V Kraus, Enrique Rico, A İmamoğlu, Peter Zoller, and Sebastian Diehl. Topology by dissipation. *New Journal of Physics*, 15(8):085001, 2013. DOI: [10.1088/1367-2630/15/8/085001](https://doi.org/10.1088/1367-2630/15/8/085001). URL <https://doi.org/10.1088/1367-2630/15/8/085001>.
- [34] B. Kraus, H. P. Büchler, S. Diehl, A. Kantian, A. Micheli, and P. Zoller. Preparation of entangled states by quantum markov processes. *Phys. Rev. A*, 78:042307, Oct 2008. DOI: [10.1103/PhysRevA.78.042307](https://doi.org/10.1103/PhysRevA.78.042307). URL <https://link.aps.org/doi/10.1103/PhysRevA.78.042307>.
- [35] Florentin Reiter, David Reeb, and Anders S Sørensen. Scalable dissipative preparation of many-body entanglement. *Physical review letters*, 117(4):040501, 2016. DOI: [10.1103/PhysRevLett.117.040501](https://doi.org/10.1103/PhysRevLett.117.040501). URL <https://link.aps.org/doi/10.1103/PhysRevLett.117.040501>.
- [36] Michael James Kastoryano, Florentin Reiter, and Anders Søndberg Sørensen. Dissipative preparation of entanglement in optical cavities. *Physical review letters*, 106(9):090502, 2011. DOI: [10.1103/PhysRevLett.106.090502](https://doi.org/10.1103/PhysRevLett.106.090502). URL <https://link.aps.org/doi/10.1103/PhysRevLett.106.090502>.
- [37] Jeffrey Marshall, Lorenzo Campos Venuti, and Paolo Zanardi. Classifying quantum data by dissipation. *Phys. Rev. A*, 99:032330, Mar 2019. DOI: [10.1103/PhysRevA.99.032330](https://doi.org/10.1103/PhysRevA.99.032330). URL <https://link.aps.org/doi/10.1103/PhysRevA.99.032330>.
- [38] Martin Kliesch, Thomas Barthel, Christian Gogolin, Michael Kastoryano, and Jens Eisert. Dissipative quantum church-turing theorem. *Physical review letters*, 107(12):120501, 2011. DOI: [10.1103/PhysRevLett.107.120501](https://doi.org/10.1103/PhysRevLett.107.120501). URL <https://link.aps.org/doi/10.1103/PhysRevLett.107.120501>.
- [39] Hefeng Wang, Sahel Ashhab, and Franco Nori. Quantum algorithm for simulating the dynamics of an open quantum system. *Physical Review A*, 83(6):062317, 2011. DOI: [10.1103/PhysRevA.83.062317](https://doi.org/10.1103/PhysRevA.83.062317). URL <https://link.aps.org/doi/10.1103/PhysRevA.83.062317>.
- [40] Thomas Barthel and Martin Kliesch. Quasilocalty and efficient simulation of markovian quantum dynamics. *Physical review letters*, 108(23):230504, 2012. DOI: [10.1103/PhysRevLett.108.230504](https://doi.org/10.1103/PhysRevLett.108.230504). URL <https://link.aps.org/doi/10.1103/PhysRevLett.108.230504>.

- [41] J. Han, W. Cai, L. Hu, X. Mu, Y. Ma, Y. Xu, W. Wang, H. Wang, Y. P. Song, C.-L. Zou, and L. Sun. Experimental simulation of open quantum system dynamics via trotterization. *Phys. Rev. Lett.*, 127:020504, Jul 2021. DOI: [10.1103/PhysRevLett.127.020504](https://doi.org/10.1103/PhysRevLett.127.020504). URL <https://link.aps.org/doi/10.1103/PhysRevLett.127.020504>.
- [42] Dave Bacon, Andrew M Childs, Isaac L Chuang, Julia Kempe, Debbie W Leung, and Xinlan Zhou. Universal simulation of markovian quantum dynamics. *Physical Review A*, 64(6):062302, 2001. DOI: [10.1103/PhysRevA.64.062302](https://doi.org/10.1103/PhysRevA.64.062302). URL <https://link.aps.org/doi/10.1103/PhysRevA.64.062302>.
- [43] Ryan Sweke, Ilya Sinayskiy, Denis Bernard, and Francesco Petruccione. Universal simulation of markovian open quantum systems. *Physical Review A*, 91(6):062308, 2015. DOI: [10.1103/PhysRevA.91.062308](https://doi.org/10.1103/PhysRevA.91.062308). URL <https://link.aps.org/doi/10.1103/PhysRevA.91.062308>.
- [44] Zixuan Hu, Rongxin Xia, and Sabre Kais. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. *Scientific reports*, 10(1):1–9, 2020. DOI: [10.1038/s41598-020-60321-x](https://doi.org/10.1038/s41598-020-60321-x). URL <https://www.nature.com/articles/s41598-020-60321-x>.
- [45] Akshay Gaikwad, Arvind, and Kavita Dorai. Simulating open quantum dynamics on an NMR quantum processor using the Sz.-Nagy dilation algorithm. *arXiv:2201.07687*, 2022. URL <https://arxiv.org/abs/2201.07687>.
- [46] Kade Head-Marsden, Stefan Krastanov, David A Mazziotti, and Prineha Narang. Capturing non-Markovian dynamics on near-term quantum computers. *Physical Review Research*, 3(1):013182, 2021. DOI: [10.1103/PhysRevResearch.3.013182](https://doi.org/10.1103/PhysRevResearch.3.013182). URL <https://link.aps.org/doi/10.1103/PhysRevResearch.3.013182>.
- [47] Andrew M. Childs and Nathan Wiebe. Hamiltonian Simulation Using Linear Combinations of Unitary Operations. *Quantum Info. Comput.*, 12(11–12):901–924, nov 2012. ISSN 1533-7146. DOI: [10.26421/QIC12.11-12](https://doi.org/10.26421/QIC12.11-12).
- [48] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015. DOI: [10.1103/PhysRevLett.114.090502](https://doi.org/10.1103/PhysRevLett.114.090502). URL <https://link.aps.org/doi/10.1103/PhysRevLett.114.090502>.
- [49] Richard Cleve and Chunhao Wang. Efficient Quantum Algorithms for Simulating Lindblad Evolution. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-041-5. DOI: [10.4230/LIPIcs.ICALP.2017.17](https://doi.org/10.4230/LIPIcs.ICALP.2017.17). URL <http://drops.dagstuhl.de/opus/volltexte/2017/7477>.
- [50] Anthony W. Schlimgen, Kade Head-Marsden, LeeAnn M. Sager, Prineha Narang, and David A. Mazziotti. Quantum simulation of open quantum systems using a unitary decomposition of operators. *Phys. Rev. Lett.*, 127:270503, Dec 2021. DOI: [10.1103/PhysRevLett.127.270503](https://doi.org/10.1103/PhysRevLett.127.270503). URL <https://link.aps.org/doi/10.1103/PhysRevLett.127.270503>.
- [51] Anthony W Schlimgen, Kade Head-Marsden, LeeAnn M Sager-Smith, Prineha Narang, and David A Mazziotti. Quantum state preparation and non-unitary evolution with diagonal operators. *arXiv preprint arXiv:2205.02826*, 2022.
- [52] Seth Lloyd and Lorenza Viola. Engineering quantum dynamics. *Phys. Rev. A*, 65:

- 010101, Dec 2001. DOI: [10.1103/PhysRevA.65.010101](https://doi.org/10.1103/PhysRevA.65.010101). URL <https://link.aps.org/doi/10.1103/PhysRevA.65.010101>.
- [53] Chao Shen, Kyungjoo Noh, Victor V. Albert, Stefan Krastanov, M. H. Devoret, R. J. Schoelkopf, S. M. Girvin, and Liang Jiang. Quantum channel construction with circuit quantum electrodynamics. *Phys. Rev. B*, 95:134501, Apr 2017. DOI: [10.1103/PhysRevB.95.134501](https://doi.org/10.1103/PhysRevB.95.134501). URL <https://link.aps.org/doi/10.1103/PhysRevB.95.134501>.
- [54] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, 2020. DOI: [10.1038/s41567-019-0704-4](https://doi.org/10.1038/s41567-019-0704-4). URL <https://www.nature.com/articles/s41567-019-0704-4>.
- [55] Hirofumi Nishi, Taichi Kosugi, and Yu-ichiro Matsushita. Implementation of quantum imaginary-time evolution method on NISQ devices by introducing nonlocal approximation. *npj Quantum Information*, 7(1):1–7, 2021. DOI: [10.1038/s41534-021-00409-y](https://doi.org/10.1038/s41534-021-00409-y). URL <https://www.nature.com/articles/s41534-021-00409-y>.
- [56] Shi-Ning Sun, Mario Motta, Ruslan N. Tazhigulov, Adrian T.K. Tan, Garnet Kin-Lic Chan, and Austin J. Minnich. Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution. *PRX Quantum*, 2:010317, Feb 2021. DOI: [10.1103/PRXQuantum.2.010317](https://doi.org/10.1103/PRXQuantum.2.010317). URL <https://link.aps.org/doi/10.1103/PRXQuantum.2.010317>.
- [57] Shin Sun, Li-Chai Shih, and Yuan-Chung Cheng. Efficient quantum simulation of open quantum system dynamics on noisy quantum computers. *arXiv preprint arXiv:2106.12882*, 2021. URL <https://arxiv.org/abs/2106.12882>.
- [58] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. DOI: [10.1145/3313276.3316366](https://doi.org/10.1145/3313276.3316366). URL <https://dl.acm.org/doi/10.1145/3313276.3316366>.
- [59] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019. DOI: [10.22331/q-2019-07-12-163](https://doi.org/10.22331/q-2019-07-12-163). URL <https://quantum-journal.org/papers/q-2019-07-12-163/>.
- [60] Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017. DOI: [10.1103/PhysRevLett.118.010501](https://doi.org/10.1103/PhysRevLett.118.010501). URL <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.118.010501>.
- [61] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), Oct 2009. ISSN 1079-7114. DOI: [10.1103/physrevlett.103.150502](https://doi.org/10.1103/physrevlett.103.150502). URL <http://dx.doi.org/10.1103/PhysRevLett.103.150502>.
- [62] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. DOI: [10.1080/00107514.2014.964942](https://doi.org/10.1080/00107514.2014.964942). URL <https://www.tandfonline.com/doi/full/10.1080/00107514.2014.964942>.
- [63] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. DOI: [10.1038/nature23474](https://doi.org/10.1038/nature23474). URL <https://www.nature.com/articles/nature23474>.
- [64] Chahan M Kropf, Clemens Gneiting, and Andreas Buchleitner. Effective dynamics of disordered quantum systems. *Physical Review X*, 6(3):031023, 2016. DOI:

- 10.1103/PhysRevX.6.031023. URL <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.6.031023>.
- [65] Trevor McCourt, Charles Neill, Kenny Lee, Chris Quintana, Yu Chen, Julian Kelly, V. N. Smelyanskiy, M. I. Dykman, Alexander Korotkov, Isaac L. Chuang, and A. G. Petukhov. Learning noise via dynamical decoupling of entangled qubits. *arXiv:2201.11173*, 2022. DOI: 10.48550/ARXIV.2201.11173. URL <https://arxiv.org/abs/2201.11173>.
- [66] Koenraad M. R. Audenaert and S. Scheel. On random unitary channels. *New Journal of Physics*, 10:023011, 2008. DOI: 10.1088/1367-2630/10/2/023011. URL <https://iopscience.iop.org/article/10.1088/1367-2630/10/2/023011>.
- [67] Robert Alicki and Karl Lendi. *Quantum dynamical semigroups and applications*, volume 717. Springer, 2007. DOI: 10.1007/3-540-70861-8. URL <https://link.springer.com/book/10.1007/3-540-70861-8>.
- [68] Gilles Brassard and Peter Hoyer. An exact quantum polynomial-time algorithm for simon’s problem. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, pages 12–23. IEEE, 1997.
- [69] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- [70] Eliahu Levy and Orr Moshe Shalit. Dilation theory in finite dimensions: the possible, the impossible and the unknown. *Rocky Mountain Journal of Mathematics*, 44(1): 203–221, 2014.
- [71] Béla Sz Nagy, Ciprian Foias, Hari Bercovici, and László Kérchy. *Harmonic analysis of operators on Hilbert space*. Springer Science & Business Media, 2010.
- [72] Robin Kothari. *Efficient algorithms in quantum query complexity*. PhD thesis, University of Waterloo, August 2014. URL <http://hdl.handle.net/10012/8625>.
- [73] Jing Xin Cui, Tao Zhou, and Gui Lu Long. An optimal expression of a kraus operator as a linear combination of unitary matrices. *Journal of Physics A: Mathematical and Theoretical*, 45(44):444011, 2012. DOI: 10.1088/1751-8113/45/44/444011. URL <https://iopscience.iop.org/article/10.1088/1751-8113/45/44/444011>.
- [74] Pei Wu. Additive combinations of special operators. *Banach Center Publications*, 30(1):337–361, 1994. URL <http://eudml.org/doc/262750>.
- [75] Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, 2019. DOI: 10.22331/q-2019-10-07-190. URL <https://quantum-journal.org/papers/q-2019-10-07-190/>.
- [76] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, and Mario Szegedy. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*, 2020. URL <https://arxiv.org/abs/2003.02831>.
- [77] Yulong Dong, Xiang Meng, K. Birgitta Whaley, and Lin Lin. Efficient phase-factor evaluation in quantum signal processing. *Phys. Rev. A*, 103:042419, Apr 2021. DOI: 10.1103/PhysRevA.103.042419. URL <https://link.aps.org/doi/10.1103/PhysRevA.103.042419>.
- [78] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Quantum signal processing. <https://github.com/ichuang/pyqsp>.
- [79] J.R. Johansson, P. D. Nation, and F. Nori. Qutip 2: A python framework for the dynamics of open quantum systems. *Comp. Phys. Comm.*, 184(1234), 2013. DOI: 10.1016/j.cpc.2012.11.019.
- [80] Yu Tong, Dong An, Nathan Wiebe, and Lin Lin. Fast inversion, preconditioned quantum linear system solvers, fast green’s-function computation, and fast evaluation of matrix functions. *Physical Review A*, 104(3):032422, 2021. DOI:

- 10.1103/PhysRevA.104.032422. URL <https://journals.aps.org/pr/abstract/10.1103/PhysRevA.104.032422>.
- [81] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of resonant equiangular composite quantum gates. *Physical Review X*, 6(4), Dec 2016. ISSN 2160-3308. DOI: 10.1103/physrevx.6.041067. URL <http://dx.doi.org/10.1103/PhysRevX.6.041067>.
- [82] Guang Hao Low and Isaac L Chuang. Hamiltonian simulation by uniform spectral amplification. *arXiv preprint arXiv:1707.05391*, 2017. URL <https://arxiv.org/abs/1707.05391>.
- [83] Sathyawageeswar Subramanian, Stephen Brierley, and Richard Jozsa. Implementing smooth functions of a hermitian matrix on a quantum computer. *Journal of Physics Communications*, 3(6):065002, 2019. DOI: 10.1088/2399-6528/ab25a2. URL <https://iopscience.iop.org/article/10.1088/2399-6528/ab25a2>.