

# Lawrence Berkeley National Laboratory

## LBL Publications

### Title

Numerical Circuit Synthesis and Compilation for Multi-State Preparation

### Permalink

<https://escholarship.org/uc/item/2pr9231c>

### Authors

Szasz, Aaron

Younis, Ed

De Jong, Wibe

### Publication Date

2023-01-22

### DOI

10.1109/qce57702.2023.00092

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

# Numerical circuit synthesis and compilation for multi-state preparation

Aaron Szasz

Computational Research Division  
Lawrence Berkeley National Laboratory  
Berkeley, USA  
aszasz@lbl.gov, [ID](#)

Ed Younis

Computational Research Division  
Lawrence Berkeley National Laboratory  
Berkeley, USA  
edyounis@lbl.gov, [ID](#)

Wibe de Jong

Computational Research Division  
Lawrence Berkeley National Laboratory  
Berkeley, USA  
wadejong@lbl.gov, [ID](#)

**Abstract**—Near-term quantum computers have significant error rates and short coherence times, so compilation of circuits to be as short as possible is essential. Two types of compilation problems are typically considered: circuits to prepare a given state from a fixed input state, called “state preparation”; and circuits to implement a given unitary operation, for example by “unitary synthesis”. In this paper we solve a more general problem: the transformation of a set of  $m$  states to another set of  $m$  states, which we call “multi-state preparation”. State preparation and unitary synthesis are special cases; for state preparation,  $m=1$ , while for unitary synthesis,  $m$  is the dimension of the full Hilbert space. We generate and optimize circuits for multi-state preparation numerically. In cases where a top-down approach based on matrix decompositions is also possible, our method finds circuits with substantially (up to 40%) fewer two-qubit gates. We discuss possible applications, including efficient preparation of macroscopic superposition (“cat”) states and synthesis of quantum channels.

**Index Terms**—quantum computing, state preparation, compilation, synthesis

## I. INTRODUCTION

Just as classical computations are carried out by a series of gates each acting on one or two bits, digital quantum computations are described by quantum circuits in which quantum gates act on quantum bits (“qubits”) [1]. Unlike in the classical case, in noisy intermediate-scale quantum (NISQ) computing [2], significant error rates impose strong limits on the useful runtime of quantum computers. Single qubits lose coherence over a short time, meaning that the state of a qubit becomes random and uncorrelated with neighboring qubits.

Even more problematic are the errors due to gate operations, especially from gates acting on two (or more) qubits at a time. Typical error rates for two-qubit gates, measured by infidelity,  $\epsilon$ , are in the range of 0.1% to 1% for quantum computing architectures including trapped ions [3], [4], neutral atoms [5]–[7], silicon quantum dots [8], and superconducting qubits [9]–[12]. Since the overall fidelity of a circuit shrinks exponentially in the number of gates  $d$ , scaling as  $(1 - \epsilon)^d$ , reducing the number of two-qubit gates in a circuit is essential for making near-term quantum computing feasible and useful.

To reduce the number of gates in a circuit as much as possible, we can consider two related tasks: synthesis and compilation. In the synthesis task, we start with a given unitary operator (or quantum state) and search for the circuit with the

fewest two-qubit gates that reproduces the unitary (or state). In the compilation task, we start with a circuit that implements the unitary or state, and we try to find a new, shorter circuit that performs the same operation as the original one.

Unitary synthesis is a widely-studied problem [13]–[23], and algorithms to carry out this task are implemented in a variety of publicly available software packages, including Qiskit [24], Cirq [25], Tket [26], and Lawrence Berkeley National Laboratory’s Berkeley Quantum Synthesis Toolkit (BQSKit) [27], [28]. State preparation is also widely studied [22], [23], [29]–[33], with algorithms again implemented in the software packages listed above.

These two synthesis/compilation tasks, state preparation and unitary synthesis, can be generalized by considering isometries. An isometry is a norm-preserving linear map from  $n_{\text{in}}$  to  $n_{\text{out}}$  qubits with  $n_{\text{in}} \leq n_{\text{out}}$ , and it can always be implemented (in an under-determined manner for  $n_{\text{in}} < n_{\text{out}}$ ) by an  $n_{\text{out}}$ -qubit unitary acting on  $n_{\text{in}}$  input qubits and  $(n_{\text{out}} - n_{\text{in}})$  ancillas each initialized to  $|0\rangle$ . From this perspective, the state preparation problem, in which we prepare an  $n$ -qubit state from the all- $|0\rangle$  state, is isometry synthesis with  $n_{\text{in}} = 0$ , while unitary synthesis is the case of  $n_{\text{in}} = n_{\text{out}}$ . Then intermediate problems are those with  $0 < n_{\text{in}} < n_{\text{out}}$ . Prescriptive (matrix decomposition-based) numerical methods for the general isometry synthesis problem are known [34], [35] and are also available in public software packages [24], [25], [36].

However, a further generalization is possible. We can view state preparation as the problem of finding a way to prepare one state from a given (simple) input state, while unitary synthesis is the problem of simultaneously preparing  $2^n$  different states of  $n$  qubits from the  $2^n$  standard computational basis states. From this perspective, the more general problem is the simultaneous preparation of  $m$  states efficiently for some  $1 \leq m \leq 2^n$ . We call this problem “multi-state preparation”.

Multi-state preparation not only generalizes state preparation and unitary synthesis, but also encompasses isometry synthesis (with  $m = 2^{n_{\text{in}}}$ ) and controlled quantum state preparation (CQSP) [37]. Furthermore, as we discuss in Sec. V below, multi-state preparation allows for reductions in circuit depth for a variety of practical applications, in some cases beyond what is possible via any of these four more specialized methods.

In this paper, we first define the multi-state preparation problem formally in Section II. In Section III we provide (with rigorous proof) the conditions to determine whether a multi-state preparation problem has a solution. In Section IV we show how the problem can be solved numerically in practice; in subsection A. we review our general approach to numerical optimization of circuits, namely circuit templating and instantiation, then in B. we present a good cost function for the multi-state preparation problem in particular. In C., we provide a detailed mathematical motivation for the cost function; readers primarily interested in using multi-state preparation as a practical tool could skip this section on the first read. In Section V, we discuss several applications of multi-state preparation. In Section VI we demonstrate the implementation of the numerical optimization procedure in BQSKit. We provide code for a simple example, and we show numerical results on random 3- and 4-qubit mappings, where we outperform other available software packages. We conclude in Section VII.

## II. THE MULTI-STATE PREPARATION PROBLEM

We begin with a formal definition.

*Definition of multi-state preparation:* Given two sets of  $m$  pure states of  $n \geq \log_2(m)$  qubits,  $\{|v_1\rangle, \dots, |v_m\rangle\}$  and  $\{|w_1\rangle, \dots, |w_m\rangle\}$ , find a quantum circuit that maps  $|v_i\rangle$  to  $|w_i\rangle$  for all  $i$ . (The action of the circuit on states not in the span of  $\{|v_1\rangle, \dots, |v_m\rangle\}$  is not constrained.)

The multi-state preparation problem includes both state preparation and unitary synthesis as special cases.

*State preparation* is the problem of finding a circuit that prepares an  $n$ -qubit pure state,  $|\psi\rangle$ , from the input product state  $|0\rangle^{\otimes n}$ . Thus state preparation is a special case of multi-state preparation with  $m = 1$ ,  $|v_1\rangle = |0\rangle^{\otimes n}$ , and  $|w_1\rangle = |\psi\rangle$ .

*Unitary synthesis* is the problem of finding a circuit that implements an  $n$ -qubit unitary operator,  $U$ . However, any unitary is precisely specified by its actions on the  $2^n$  computational basis states,  $|0\rangle^{\otimes n}$ ,  $|1\rangle \otimes |0\rangle^{\otimes n-1}$ ,  $\dots$ ,  $|1\rangle^{\otimes n}$ ; when viewed as a matrix, the images of these states written in the computation basis are precisely the columns of  $U$ . Thus letting  $\{|v_i\rangle\}$  be the computational basis states and  $\{|w_i\rangle\}$  their images,  $|w_i\rangle = U|v_i\rangle$ , we see that unitary synthesis is multi-state preparation with  $m = 2^n$ .

Synthesis of isometries is also a special case of multi-state preparation. An isometry is defined by the mapping of the computational basis states on  $n_{\text{in}}$  qubits to some set of  $m = 2^{m_{\text{in}}}$  states on  $n \geq n_{\text{in}}$  qubits. We can then append  $n - n_{\text{in}}$  ancillas, each in the state  $|0\rangle$ , to each of the input states. The result is a multi-state preparation problem with  $m$  input states and  $m$  output states, each of  $n$  qubits.

One important note is that in each of these three cases, state preparation, unitary synthesis, and isometry synthesis, the input states are all orthogonal to one another. As we will see in Sec. III below, orthogonality guarantees the existence of a solution to a multi-state preparation problem. However, it

is not a necessary condition. Thus our statement of the multi-state preparation problem does not make any assumption on the inner products  $\langle v_i | v_j \rangle$ .

### A. Two-qubit examples

*Example 1:* Let

$$\begin{aligned} |v_1\rangle &= |00\rangle, & |v_2\rangle &= |01\rangle \\ |w_1\rangle &= |00\rangle, & |w_2\rangle &= |01\rangle. \end{aligned} \quad (1)$$

This apparently trivial example actually illustrates precisely why a dedicated approach to multi-state preparation is useful. In the standard basis, this example problem amounts to specifying the first two columns of the  $4 \times 4$  unitary matrix that a quantum circuit should implement. We could imagine solving this problem by first appending two additional orthonormal columns, then performing a full unitary synthesis.

If we were lucky and picked  $|10\rangle \mapsto |10\rangle$  and  $|11\rangle \mapsto |11\rangle$ , giving the identity transformation as the full unitary, we would find a circuit with no gates whatsoever, evidently the shortest possible circuit solving the original problem. On the other hand, if we happened to fill out the rest of the unitary transformation as  $|10\rangle \mapsto |11\rangle$  and  $|11\rangle \mapsto |10\rangle$ , the full unitary would be given by a single CNOT with the first qubit as the control.

In other words, each multi-state preparation problem (with  $m < 2^n$ ) corresponds to many unitary synthesis problems, and some will have shorter circuits as solutions than others. From this perspective, in solving a multi-state preparation problem, we aim to find the optimal unitary transformation for the given state mapping as well as the shortest implementation of that unitary. Using the numerical approach described in Section IV, we perform both optimizations simultaneously.

*Example 2:* Let

$$\begin{aligned} |v_1\rangle &= |00\rangle, & |v_2\rangle &= |11\rangle \\ |w_1\rangle &= |00\rangle, & |w_2\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \end{aligned} \quad (2)$$

An optimal circuit (in terms of two-qubit gate count) for this example looks like



$$= \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \bullet \\ \bullet \\ \oplus \end{array} \begin{array}{c} \oplus \\ \bullet \\ \oplus \end{array} \begin{array}{c} R_y\left(\frac{3\pi}{2}\right) \\ R_y\left(\frac{7\pi}{4}\right) \\ R_y\left(\frac{\pi}{2}\right) \end{array} \begin{array}{c} \bullet \\ \bullet \\ \oplus \end{array} \begin{array}{c} R_y\left(\frac{3\pi}{2}\right) \\ R_y\left(\frac{7\pi}{4}\right) \\ R_y\left(\frac{\pi}{2}\right) \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (4)$$

Notably, two 2-qubit gates are required. In comparison, one could imagine solving this problem using a unitary synthesis method, which would require first adding two more orthonormal output states for inputs  $|01\rangle$  and  $|10\rangle$ ; however, generic two-qubit unitaries require three CNOTs [38], [39],

so in general we get a shorter circuit by doing the multi-state preparation problem directly. On the other side, single-state preparation for two qubits uses at most one CNOT gate [40], so all but a set of measure 0 of multi-state preparation problems cannot be solved by single-state preparation. Thus there are situations in which multi-state preparation will be beneficial relative to both of the more common synthesis/compilation problems.

*Example 3:* As noted above, the states can also be non-orthogonal. For example, with  $|v_i\rangle, |w_i\rangle$  as in Eq. 2, let

$$\begin{aligned} |\tilde{v}_2\rangle &= \frac{|v_1\rangle + |v_2\rangle}{\sqrt{2}} \\ |\tilde{w}_2\rangle &= \frac{|w_1\rangle + |w_2\rangle}{\sqrt{2}}. \end{aligned} \quad (5)$$

Then the exact same circuits from (3) and (4) will also solve the problem  $|v_1\rangle \mapsto |w_1\rangle, |\tilde{v}_2\rangle \mapsto |\tilde{w}_2\rangle$ .

*Example 4:* As discussed in the next section, some problems with non-orthogonal states have no solution. For example, let

$$\begin{aligned} |v_1\rangle &= |00\rangle, \quad |v_2\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ |w_1\rangle &= \frac{|00\rangle + |01\rangle}{\sqrt{2}}, \quad |w_2\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}. \end{aligned} \quad (6)$$

Note that  $\langle v_1|v_2\rangle = 1/\sqrt{2}$  while  $\langle w_1|w_2\rangle = 1/2$ . As we show below, this implies that there is no solution to this particular multi-state mapping problem.

### III. WHEN DOES A MULTI-STATE PREPARATION PROBLEM HAVE A SOLUTION IN PRINCIPLE?

Given two sets of  $m$   $n$ -qubit states each,  $\{|v_i\rangle\}$  and  $\{|w_i\rangle\}$ , we want to find a quantum circuit that maps  $|v_i\rangle$  to  $|w_i\rangle$  for all  $i \in \{1, \dots, m\}$ . The first question to ask is whether a solution exists at all. Fortunately, there is a simple answer.<sup>1</sup>

*Solution:* Let  $V$  be the matrix whose columns contain the states  $|v_i\rangle$ , represented as vectors in the computational basis, and likewise let  $W$  be a matrix with columns  $|w_i\rangle$ . Both matrices are  $2^n \times m$ . A circuit that maps  $|v_i\rangle$  to  $|w_i\rangle \forall i$  exists if and only if  $V^\dagger V = W^\dagger W$ , i.e. if and only if  $\langle v_i|v_j\rangle = \langle w_i|w_j\rangle$  for all  $i, j$ .

**Intuition:** By analogy, we consider four vectors on the unit circle,  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{w}_1,$  and  $\mathbf{w}_2$ , and we ask whether there is some rotation plus reflection that takes  $\mathbf{v}_1$  to  $\mathbf{w}_1$  and  $\mathbf{v}_2$  to  $\mathbf{w}_2$ . The situation is illustrated in Fig. 1.

We can of course always find a rotation that maps  $\mathbf{v}_1$  to  $\mathbf{w}_1$ . Once we carry out this rotation, it becomes visually clear whether or not we can simultaneously map  $\mathbf{v}_2$  to  $\mathbf{w}_2$ : as shown in the figure, this is possible only when the angle between  $\mathbf{v}_1$  and  $\mathbf{v}_2$  is the same (up to sign) as the angle between  $\mathbf{w}_1$  to  $\mathbf{w}_2$ . Intuitively, this is because rotations and reflections preserve

<sup>1</sup>Note that this result can also be found on the Mathematics Stack Exchange forum [41].

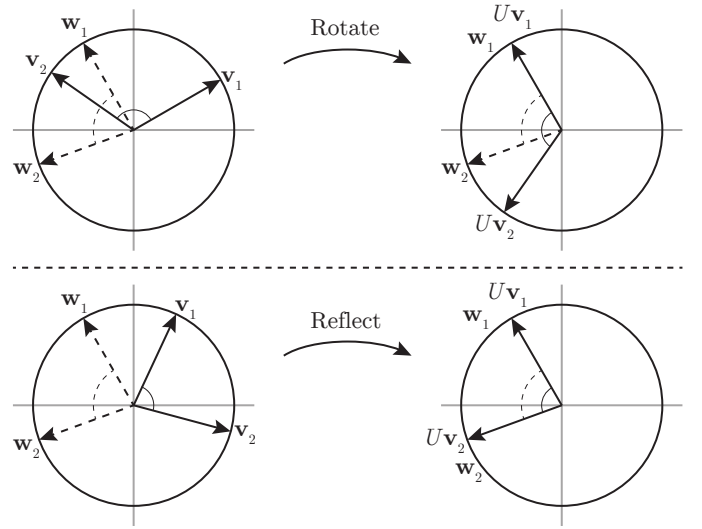


Fig. 1. There exists a unitary that maps vectors  $(\mathbf{v}_1, \mathbf{v}_2)$  to  $(\mathbf{w}_1, \mathbf{w}_2)$  if and only if the angles between the vectors are the same,  $\mathbf{v}_1 \cdot \mathbf{v}_2 = \mathbf{w}_1 \cdot \mathbf{w}_2$ . In the top row, we see that when the angles are different, a unitary  $U$  that takes  $\mathbf{v}_1$  to  $\mathbf{w}_1$  cannot also take  $\mathbf{v}_2$  to  $\mathbf{w}_2$ . Conversely, on the bottom is an example where the angles match, and thus there exists a unitary that maps both vectors simultaneously. This is the intuition behind the condition  $V^\dagger V = W^\dagger W$  for solvability of a multi-state preparation problem.

angles, so one set of vectors can be transformed to another if and only if the angles between the initial and between the final vectors match.

Returning to our multi-state preparation problem, the rotations and reflections become the unitary action of the desired quantum circuit, while the angles between vectors correspond to the overlaps  $\langle v_i|v_j\rangle$  and  $\langle w_i|w_j\rangle$ . These are precisely contained in the matrices  $V^\dagger V$  and  $W^\dagger W$ :

$$V^\dagger V = \begin{pmatrix} \langle v_1|v_1\rangle & \langle v_1|v_2\rangle & \cdots & \langle v_1|v_m\rangle \\ \langle v_2|v_1\rangle & \langle v_2|v_2\rangle & \cdots & \langle v_2|v_m\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_m|v_1\rangle & \langle v_m|v_2\rangle & \cdots & \langle v_m|v_m\rangle \end{pmatrix}. \quad (7)$$

**Proof:** We now prove the result.

*If a circuit exists,  $V^\dagger V = W^\dagger W$ :* We first suppose that a circuit for the multi-state preparation indeed exists. This circuit implements a unitary operator  $U$ , which by assumption realizes the mapping  $|w_i\rangle = U|v_i\rangle$ . Then

$$\langle w_i|w_j\rangle = \langle Uv_i|Uv_j\rangle = \langle v_i|U^\dagger Uv_j\rangle = \langle v_i|v_j\rangle \quad (8)$$

and hence each element of the two matrices is equal and  $W^\dagger W = V^\dagger V$ . ■

*If  $V^\dagger V = W^\dagger W$ , a circuit exists:* We now suppose that  $V^\dagger V = W^\dagger W$ . Let's call this matrix  $O$ , for "overlap".  $O$  is Hermitian by construction, hence diagonalizable. Furthermore,  $O$  is positive semi-definite:  $\langle \psi|V^\dagger V|\psi\rangle = \langle V\psi|V\psi\rangle = \|V\psi\|^2 \geq 0$ . We thus let  $O = SDS^\dagger$  where  $S$  is unitary and  $D$  is diagonal with only positive real entries. ( $D$  is only guaranteed to be nonnegative, but if  $D$  has zero entries, we can just remove them and remove the corresponding columns of  $S$

to get smaller matrices, making  $D$  strictly positive. Note that  $D$  will have zero entries when the columns of  $V$  are linearly dependent.)

Then let  $\tilde{V} = VSD^{-1/2}$ ,  $\tilde{W} = WSD^{-1/2}$ . Intuitively,  $VS$  is the matrix whose columns are derived by orthogonalizing the columns of  $V$ , and the factor of  $D^{-1/2}$  normalizes the vectors to be orthonormal. Then both  $\tilde{V}$  and  $\tilde{W}$  are isometries:

$$\begin{aligned}\tilde{V}^\dagger \tilde{V} &= D^{-1/2} S^\dagger V^\dagger V S D^{-1/2} = D^{-1/2} S^\dagger O S D^{-1/2} \\ &= D^{-1/2} D D^{-1/2} = \text{Id}\end{aligned}\quad (9)$$

and likewise for  $\tilde{W}$ . In general,  $\tilde{V}$  and  $\tilde{W}$  have more rows than columns: they are isometries from a smaller space into a larger one. In that case, we append further orthogonal columns to make them into square matrices while maintaining the isometric conditions  $\tilde{V}^\dagger \tilde{V} = \text{Id}$  and  $\tilde{W}^\dagger \tilde{W} = \text{Id}$ . Any square isometric condition is unitary, so both  $\tilde{V}$  and  $\tilde{W}$  are then unitary.

Finally, let  $U = \tilde{W} \tilde{V}^\dagger$ . As a product of two unitaries,  $U$  is also unitary.

Furthermore, if we didn't need to add extra columns to  $\tilde{V}$  and  $\tilde{W}$ ,

$$\begin{aligned}UV &= WSD^{-1} S^\dagger V^\dagger V = WSD^{-1} S^\dagger O \\ &= WSD^{-1} S^\dagger S D S^\dagger = W.\end{aligned}\quad (10)$$

If we did add extra columns, then letting the matrices containing just the extra columns be  $C_V$  and  $C_W$  respectively, we have

$$UV = W + C_W C_V^\dagger V, \quad (11)$$

but all the columns of  $V$  are in the span of the columns of  $VSD^{-1/2}$  and the columns in  $C_V$  were chosen to be orthogonal, so  $C_V^\dagger V = 0$ .

Thus given  $V^\dagger V = W^\dagger W$ , we have explicitly constructed a unitary that maps  $V$  to  $W$  and hence  $|v_i\rangle$  to  $|w_i\rangle$  for all  $i$ . ■

#### IV. NUMERICAL IMPLEMENTATION

Now that we know how to tell whether a solution to a multi-state preparation problem exists, we turn to the question of finding that solution in practice. We consider the framework of synthesis and compilation using instantiation of parameterized circuit templates with numerical optimization [21]. In this framework, we find an optimal circuit using a cost function that is minimized when the circuit solves the problem of interest.

Here we first briefly review parameterized circuit instantiation, then we present and motivate a cost function for the multi-state preparation problem.

##### A. Background: numerical optimization via instantiation

Suppose we want to find a quantum circuit that solves some task, for example preparing a given state from  $|0\rangle^{\otimes n}$  or synthesizing a unitary. We perform synthesis by repeating two steps:

- 1) Generate a circuit “template” containing a variety of parameterized gates such as  $\exp(i\theta Z)$ , with unspecified parameters.

- 2) Given this circuit template, find the parameter values that bring the circuit as close as possible to a solution to the problem of interest. If the best possible parameterization fails to produce a circuit that solves the problem, go back to step 1.

Step 2 requires evaluating, for each set of parameters, how close the parameterized circuit comes to solving the problem (e.g. preparing the desired state). We perform this evaluation by means of a problem-specific cost function.

For example, for unitary synthesis, if  $U_C$  is the unitary operator implemented by the circuit and  $U$  is the desired unitary, a good cost function is  $1 - |\text{Tr}[U_C U^\dagger]|/2^n$  [21], which takes values in  $[0, 1]$  and is minimized at 0 if and only if  $U_C = U$ . For preparation of a state  $|\psi\rangle$ , a good cost function is  $1 - |\langle \psi | U_C | \mathbf{0} \rangle|$  or equivalently  $1 - |\text{Tr}[U_C | \mathbf{0} \rangle \langle \psi |]|$ ,<sup>2</sup> which is again minimized at 0 for a correct solution.

For a given cost function, there remain many possible approaches to generating circuit templates and to carrying out the parameter optimization. QBSKit includes a variety of strategies, including QSearch [16], QFAST [17], and LEAP [18]. The circuit templates can also be generated in a way that respects the connectivity of some particular quantum hardware [42].

##### B. Cost function for multi-state preparation

Consider the matrices  $V$  and  $W$  introduced in Sec. III, each of whose  $m$  columns are  $n$ -qubit states. Then a natural choice of cost function for multi-state preparation is:

$$1 - \frac{1}{m} |\text{Tr}[U_C V W^\dagger]|. \quad (12)$$

This cost function includes the ones for state preparation and for unitary synthesis as special cases:

- For state preparation,  $V$  is the column vector  $(1, 0, \dots, 0)^\top$ , while  $W$  is the column vector containing the elements of  $|\psi\rangle$  in the computational basis. Then the cyclic property of the trace gives  $\text{Tr}[U_C V W^\dagger] = \langle \psi | U_C | \mathbf{0} \rangle$ .
- For unitary synthesis,  $V$  is just the identity matrix, so that its columns are the successive computational basis vectors, and  $W = U$  since its columns are the states mapped to by said basis vectors. Also noting that  $m = 2^n$ , we exactly recover the cost function for synthesis.

For any multi-state preparation problem, the cost function (12) lies in the range  $[0, 1]$  and has a minimum value at 0 for a correct solution: in that case,  $U_C V = W$ , and

$$\text{Tr}[W W^\dagger] = \text{Tr}[W^\dagger W] = \sum_i \langle w_i | w_i \rangle = m. \quad (13)$$

Note that this proof does not require orthogonality of the  $|w_i\rangle$ , or even linear independence.

<sup>2</sup>We use  $|0\rangle$  as shorthand for  $|0\rangle^{\otimes n}$

### C. Motivation for cost function

Suppose that the state system in question has a solution,  $U_C$ , which precisely maps  $|v_i\rangle$  to  $|w_i\rangle$ . Then for all  $i$ ,  $\langle w_i|U_C|v_i\rangle = 1$ . For arbitrary  $U$ , we have  $|\langle w_i|U|v_i\rangle| \leq 1$ , so any cost function of the form

$$1 - \left| \sum_i a_i \langle w_i|U|v_i\rangle \right| \quad (14)$$

for  $a_i > 0$ ,  $\sum_i a_i = 1$  will satisfy

$$1 - \left| \sum_i a_i \langle w_i|U|v_i\rangle \right| \geq 1 - \sum_i a_i |\langle w_i|U|v_i\rangle| \geq 1 - \sum_i a_i = 0 \quad (15)$$

and will achieve the minimum value of 0 precisely for correct solutions  $U_C$ .

The cost function given in Eq. (12) is of the form (14):

$$\frac{1}{m} \text{Tr}[UVW^\dagger] = \frac{1}{m} \sum_i \text{Tr}[U|v_i\rangle\langle w_i|] = \sum_i \frac{1}{m} \langle w_i|U|v_i\rangle. \quad (16)$$

However, the choice of  $a_i = 1/m$  is not obvious and deserves some justification.

First, suppose that all the input (equivalently, output) vectors form an orthonormal set,  $V^\dagger V = \text{Id}$ . Then all pairs of vectors ( $|v_i\rangle, |w_i\rangle$ ) are equivalent under some high-dimensional reflection and rotation, so it is natural to weight the mapping of each pair equally.  $1/m$  is thus a natural weight to assign.

What about the more general case where the provided input and output vectors are not orthogonal? Consider the following example, which we will return to several times:

*Example:* Let  $|\psi_1\rangle$  and  $|\psi_2\rangle$  be orthogonal states, and define  $|v_\pm\rangle = \sqrt{1-\epsilon^2}|\psi_1\rangle \pm \epsilon|\psi_2\rangle$  for some small  $\epsilon > 0$ . So the two specified input states (and, if the problem is solvable, the corresponding output states) are nearly parallel, with

$$V^\dagger V = \begin{pmatrix} 1 & 1-2\epsilon^2 \\ 1-2\epsilon^2 & 1 \end{pmatrix}. \quad (17)$$

The desired images of the input states are  $|v_\pm\rangle \mapsto |w_\pm\rangle$  and  $|\psi_i\rangle \mapsto |\phi_i\rangle$ .

We will keep this example in mind as we proceed to derive two possible cost functions and then compare them.

An exact solution to the multi-state preparation problem would have each state in the span of  $\{|v_i\rangle\}$  mapped to the same linear combination of the  $\{|w_i\rangle\}$ , so a natural way to define the total cost function for the system is to average over the error in mapping each individual state in  $\text{Span}(\{|v_i\rangle\})$ . Consider some such state  $|v\rangle$ , which in the computational basis we write as  $V\mathbf{c}$ , for some coefficient vector  $\mathbf{c}$ ; this state should map to  $W\mathbf{c}$ , so for a given circuit implementing the unitary  $U_C$ , the error for this state is  $\|V\mathbf{c}\|^2 - \mathbf{c}^\dagger W^\dagger U_C V \mathbf{c}$ . (If the columns of  $V$  are orthonormal,  $\|V\mathbf{c}\|^2 = 1$ , but that is not true in general.) Thus we could write the overall cost for the circuit as an average over different coefficient vectors  $\mathbf{c}$ :

$$\Delta = \frac{1}{\mathcal{V}} \int_{\mathcal{C}} \mathbf{c}^\dagger O \mathbf{c} \, d\mathbf{c} - \frac{1}{\mathcal{V}} \int_{\mathcal{C}} \mathbf{c}^\dagger W^\dagger U_C V \mathbf{c} \, d\mathbf{c} \quad (18)$$

where  $\mathcal{V}$  measures the volume of  $\text{Span}(\{|v_i\rangle\})$  and  $O$  is the overlap matrix  $V^\dagger V$ . Of course, this is not yet well-defined, since we need an integration measure on the coefficient vectors. This choice of integration measure effectively assigns different weights to the states in  $\text{Span}(\{|v_i\rangle\})$  and is what determines the weights  $a_i$  in Eq. (14). We take two different approaches.

*Approach 1:* First, we could sample  $\mathbf{c}$  as random  $m$ -dimensional vectors, i.e. from the first column of  $m \times m$  Haar-random unitaries; for unitary  $U_m$ , the first column is  $U_m \mathbf{e}_1$  with  $\mathbf{e}_1 = (1, 0, \dots, 0)^\top$ . Then Eq. (18) becomes

$$\mathbf{e}_1^T \left[ \int_{\text{Haar}} U_m^\dagger O U_m \, d\mu(U_m) - \int_{\text{Haar}} U_m^\dagger W^\dagger U_C V U_m \, d\mu(U_m) \right] \mathbf{e}_1 \quad (19)$$

where  $d\mu$  is the Haar measure. The Haar average can be computed exactly, by the ‘‘twirling’’ integral [43]

$$\int_{\text{Haar}} U^\dagger A U \, d\mu(U) = \frac{\text{Tr}[A]}{m} \text{Id}. \quad (20)$$

Using  $\text{Tr}[O] = m$ , we thus get a cost function of

$$\left[ 1 - \frac{1}{m} \text{Tr}[W^\dagger U_C V] \right] (\mathbf{e}_1^T \cdot \text{Id} \cdot \mathbf{e}_1) = 1 - \frac{1}{m} \text{Tr}[W^\dagger U_C V], \quad (21)$$

which is precisely Eq. (12) up to taking the magnitude, which we do in (12) only for the convenience of working with real numbers in numerical optimization.

We claimed that our choice of distribution for sampling the coefficients  $\mathbf{c}$  corresponds to a choice of weights for the average over states in  $\text{Span}(\{|v_i\rangle\})$ . To understand the weights that arise from unitary sampling of  $\mathbf{c}$  as considered here, we turn back to the example above. The cost function becomes

$$1 - (\langle w_+|U_C|v_+\rangle + \langle w_-|U_C|v_-\rangle) / 2 \quad (22)$$

which in terms of the orthogonal states  $|\psi_1\rangle$  and  $|\psi_2\rangle$  is

$$1 - ((1-\epsilon^2)\langle \phi_1|U_C|\psi_1\rangle + \epsilon^2\langle \phi_2|U_C|\psi_2\rangle). \quad (23)$$

So in the cost function, the orthogonal basis states for  $\text{Span}(\{|v_i\rangle\})$  are effectively weighted by their prevalence in the specified input states. In this case, most of the weight is given to  $|\psi_1\rangle$  since that is also the case for  $|v_\pm\rangle$ .

To get the corresponding general result, we recall that the overlap matrices can be diagonalized as  $V^\dagger V = W^\dagger W = SDS^\dagger$ . Then the cost function can be rewritten with

$$\text{Tr}[W^\dagger U_C V] = \text{Tr}[D (WSD^{-1/2})^\dagger U (VSD^{-1/2})]. \quad (24)$$

Letting  $|\tilde{v}_i\rangle$  and  $|\tilde{w}_i\rangle$  be the (orthonormal) columns of  $VSD^{-1/2}$  and  $WSD^{-1/2}$ , we can then rewrite the cost function (12) as

$$1 - \frac{1}{m} \left| \sum_i d_i \langle \tilde{w}_i|U_C|\tilde{v}_i\rangle \right|. \quad (25)$$

Thus in terms of this orthonormal basis for  $\text{Span}(\{|v_i\rangle\})$ , the weights are given by the eigenvalues of  $V^\dagger V$ , which are the weights of the  $|\tilde{v}\rangle$  basis states in the original input vectors:

$$d_i = \sum_j |\langle v_j|\tilde{v}_i\rangle|^2. \quad (26)$$

*Approach 2:* Alternatively, we could sample  $Vc$  as random  $m$ -dimensional vectors from the first column of  $m \times m$  Haar-random unitaries. Conceptually, this corresponds to first finding  $\text{Span}(\{|v_i\rangle\})$ , then forgetting about the original input states and just averaging over the space in a uniform way; in practice, we can take linear combinations given by  $U_m e_1$  over some orthonormal basis of the span, for example the columns of  $VSD^{-1/2}$ . Then Eq. (18) becomes

$$1 - e_1^T \left[ \int_{\text{Haar}} U_m^\dagger (WSD^{-1/2})^\dagger U_C (VSD^{-1/2}) U_m d\mu(U_m) \right] e_1. \quad (27)$$

This gives

$$1 - \frac{1}{m} \text{Tr}[(WSD^{-1/2})^\dagger U_C (VSD^{-1/2})]. \quad (28)$$

In the style of Eq. (14), we have

$$\Delta = 1 - \frac{1}{m} \left| \sum_i \langle \tilde{w}_i | U_C | \tilde{v}_i \rangle \right|. \quad (29)$$

*Which approach is best?* Both approaches seem reasonable, a priori, but Approach 1 is preferred for several reasons:

- Approach 1 is more flexible as a software tool. If we implemented Approach 2, an end-user would have no way to access Approach 1, since the code would automatically perform the orthogonalization and re-weighting, so that the user could not choose to specify other weights if desired. Conversely, a user could separately perform diagonalization of  $V^\dagger V$  and input  $VSD^{-1/2}$  and  $WSD^{-1/2}$  in place of  $V$  and  $W$ , thus recreating Approach 2 from Approach 1.
- Computing Eq. (28) requires inverting the overlap matrix, so if the input states are nearly linearly dependent as in the example above with a small  $\epsilon$ , the cost function involves a numerically unstable operation.
- One might want to find a circuit that gives a good approximate mapping in the case that  $V^\dagger V \neq W^\dagger W$ . The cost function of Eq. (12) remains well-defined and corresponds to an intuitive picture of how an approximate mapping should behave.

Consider again our example problem, but now suppose that the requested mapping is

$$|v_\pm\rangle = \sqrt{1 - \epsilon^2} |\psi_1\rangle \pm \epsilon |\psi_2\rangle \quad (30)$$

$$\mapsto |w_\pm\rangle = \sqrt{1 - 4\epsilon^2} |\psi_1\rangle \pm 2\epsilon |\psi_2\rangle. \quad (31)$$

Evidently, there is no exact solution, but for small  $\epsilon$  the identity transformation is close. Specifically, using Eq. (12), we find an appropriately small cost of  $\approx \epsilon^2/2$ .

On the other hand, Eq. (28) is no longer well-defined. We could define  $VSD^{-1/2}$  using  $S$  and  $D$  from  $V^\dagger V$  and  $WSD^{-1/2}$  using  $S$  and  $D$  from  $W^\dagger W$ , but then in the example we would find a cost of exactly 0, a clearly nonsensical result.

Multi-state preparation has a variety of practical applications that go beyond single-state preparation and unitary synthesis. We address three such applications in some detail below. Other applications (for which we would use multi-state preparation as a subroutine in CQSP [37]) include solving linear systems [44] and quantum clustering algorithms [45].

#### A. Preparation of macroscopic superposition states

Multi-state preparation on a few qubits is helpful as a subroutine for preparing a single highly-entangled state of many qubits. Imagine we want to prepare a macroscopic superposition state (often called a ‘‘cat’’ state) of the form

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|\phi_1\rangle + |\phi_2\rangle) \quad (32)$$

where  $|\phi_1\rangle$  and  $|\phi_2\rangle$  are orthogonal  $n$ -qubit states.

An important special case is where the system can be divided into few-qubit pieces, and between each piece there is no entanglement in either  $|\phi_1\rangle$  or  $|\phi_2\rangle$ , and furthermore where for each few-qubit piece, the local states of  $|\phi_1\rangle$  and  $|\phi_2\rangle$  are orthogonal. This class of states sounds contrived, but it includes some very important examples both for theory and for practical simulation of physical systems:

- The Greenberger-Horne-Zeilinger (GHZ) state:  $|\psi\rangle = (|0\rangle^{\otimes n} + |1\rangle^{\otimes n})/\sqrt{2}$ . This is the simplest example, where each of the two pieces is a product over single-qubit states.
- Any state of the form

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left( (a|00\rangle + b|11\rangle)^{\otimes n/2} + (c|01\rangle + d|10\rangle)^{\otimes n/2} \right) \quad (33)$$

Here the system is divided into two-qubit pieces, and on each piece  $|\phi_1\rangle$  has a factor  $(a|00\rangle + b|11\rangle)$  while  $|\phi_2\rangle$  has a factor  $(c|01\rangle + d|10\rangle)$ .

- More general states similar to (33). For example, for the first two qubits,  $|\phi_1\rangle$  could have a factor  $|01\rangle$  and  $|\phi_2\rangle$  a factor  $|10\rangle$ , while for the next three,  $|\phi_1\rangle$  could have a factor  $(|000\rangle + |111\rangle)/\sqrt{2}$  and  $|\phi_2\rangle$  a factor  $(|001\rangle + |010\rangle + |100\rangle)/\sqrt{3}$ . The key requirement is that the state is divided into chunks of qubits, where within each of  $|\phi_1\rangle$  and  $|\phi_2\rangle$  there is no entanglement between chunks, and where for each chunk the states on  $|\phi_1\rangle$  and  $|\phi_2\rangle$  are orthogonal.

The GHZ state can be prepared in a constant-depth circuit (not scaling with system size) by using measurement and reset in addition to unitary operations [46]. We can then combine the GHZ state with multi-state preparation to immediately prepare a state like (33). We solve the multi-state preparation problem for the system

$$|00\rangle \mapsto a|00\rangle + b|11\rangle \quad (34)$$

$$|11\rangle \mapsto c|01\rangle + d|10\rangle \quad (35)$$

and then just append that circuit to the one for GHZ preparation on each of the  $n/2$  two-qubit blocks.

Precisely this kind of superposition is useful for measuring expectation values of operators on a quantum computer, with applications in physics and chemistry. Suppose, for example, that one has an initial state  $|\psi_0\rangle$  and wants to find the expectation value of the time-evolution operator for a Hamiltonian  $H$ ,  $U = e^{-iHt}$ . This expectation value is important in algorithms such as variational quantum phase estimation (VQPE) for finding ground states [47]. A common approach to finding both the magnitude and phase of  $\langle\psi_0|U|\psi_0\rangle$  is the Hadamard test, but since all operations in the state preparation and application of  $U$  become controlled operations with an ancilla, the circuit depth becomes much longer than for running  $U$  on its own, a major problem for near-term devices.

A different approach, proposed in [48], makes use of a reference state  $|\text{Ref}\rangle$ , one which is easy to prepare, is orthogonal to  $|\psi_0\rangle$ , and is an eigenstate of  $H$  with known eigenvalue  $E$ . In that case, the expectation value  $\langle\psi_0|U|\psi_0\rangle = re^{i\theta}$ , including phase, can be computed using  $|\langle\psi_0|U|\psi_0\rangle|$ ,  $|\langle\psi_0| + \langle\text{Ref}|U(|\psi_0\rangle + |\text{Ref}\rangle)|/2$ , and  $|\langle\psi_0| + \langle\text{Ref}|U(|\psi_0\rangle + i|\text{Ref}\rangle)|/2$ ; respectively, these give access to  $r$ ,  $\cos(\theta)$ , and  $\sin(\theta)$ . All three can be measured without adding additional controlled gates and thus without significantly increasing circuit depth.

One example of a Hamiltonian where this reference state approach is useful is the spin Heisenberg model on a lattice, used for simulating magnetism. Typical initial states  $|\psi_0\rangle$  of interest have total spin 0 (Marshall's theorem, [49]§5.1), meaning that they are linear combinations of basis states that have an equal number of  $|0\rangle$ s and  $|1\rangle$ s. On the other hand, the fully polarized states  $|0\rangle^{\otimes n}$  and  $|1\rangle^{\otimes n}$  are exact eigenstates of  $H$  with total spin  $n/2$  and make for good reference states. Thus when, e.g., running VQPE to find the ground state of a spin system, we might want to prepare the superposition of the spin-0 initial state  $(|01\rangle - |10\rangle)/\sqrt{2}$  with the reference state  $|0\rangle^{\otimes n}$ , which is precisely of the form (33).

### B. Hamiltonian evolution for certain symmetry sectors

Related to the previous problem, if we want to apply the operator  $e^{-iHt}$  to a given state  $|\psi_0\rangle$ , we may be able to use multi-state preparation to construct a different operator from  $e^{-iHt}$  with a shorter circuit but which still acts correctly on  $|\psi_0\rangle$  in particular.

For example, consider a two-qubit system with  $H$  given by

$$H = \begin{pmatrix} 1 & & & \\ & -1 & 2 & \\ & 2 & -1 & \\ & & & 1 \end{pmatrix}. \quad (36)$$

(This is, in fact, the aforementioned spin Heisenberg model, specialized to two spins.) The time evolution operator  $e^{-iHt}$  maintains the same block structure:

$$e^{-iHt} = \begin{pmatrix} e^{-it} & & & \\ & e^{it} \cos(2t) & -ie^{it} \sin(2t) & \\ & -ie^{it} \sin(2t) & e^{it} \cos(2t) & \\ & & & e^{-it} \end{pmatrix}. \quad (37)$$

If the initial state  $|\psi_0\rangle$  is of the form  $a|01\rangle + b|10\rangle$ , the block structure of  $e^{-iHt}$  ensures that the time-evolved state will remain a superposition of  $|01\rangle$  and  $|10\rangle$ . Consequently, we can replace  $e^{-iHt}$  by any unitary of the form

$$U = \begin{pmatrix} * & & & * \\ & e^{it} \cos(2t) & -ie^{it} \sin(2t) & \\ & -ie^{it} \sin(2t) & e^{it} \cos(2t) & \\ * & & & * \end{pmatrix} \quad (38)$$

without changing the action on  $|\psi_0\rangle$ . Thus instead of finding a circuit for  $e^{-iHt}$  using unitary synthesis, we can solve the multi-state preparation problem

$$|01\rangle \mapsto e^{it} \cos(2t)|01\rangle - ie^{it} \sin(2t)|10\rangle \quad (39a)$$

$$|10\rangle \mapsto e^{it} \cos(2t)|10\rangle - ie^{it} \sin(2t)|01\rangle, \quad (39b)$$

which in general will produce a shorter circuit.

### C. Preparation of isometry circuits and quantum channels

General operations on quantum states are described by quantum channels, i.e. completely positive trace-preserving (CPTP) maps [1], so we would like to be able to find a short and efficient circuit that carries out a given channel. Although channels are non-unitary, generically involving measurement, reset, and feed-forward, Stinespring's dilation theorem [50] guarantees that any channel can be implemented by an isometry followed by partial trace, i.e. measurement. More precisely, a channel from  $n_{\text{in}}$  to  $n_{\text{out}}$  qubits can be implemented by an isometry from  $n_{\text{in}}$  qubits to  $n_{\text{in}} + 2n_{\text{out}}$  qubits, followed by discarding all but  $n_{\text{out}}$  of the output qubits [35]. This is the approach to channel compilation taken in, for example, the Mathematica library UniversalQCompiler [36].

As we demonstrate in Sec. VI below, our numerical solution to the multi-state preparation problem finds shorter circuits than do existing software implementations for compilation of isometries [24], [25], [36]. We will therefore be able to reduce gate depths and thus improve fidelity for simulations of quantum channels. One key application where this compression will be essential is simulation of open quantum systems, where recent papers have already made use of the representation of a channel as an isometry followed by measurement [51], [52].

## VI. SOFTWARE IMPLEMENTATION AND NUMERICAL DEMONSTRATION

### A. BQSKit code

Multi-state preparation is included in the Berkeley Quantum Synthesis Toolkit (BQSKit), an open-source quantum compiler library written in Python and Rust and maintained by Lawrence Berkeley National Laboratory [27], [28], beginning with version 1.1. The problem is specified as a `StateSystem` object, defined using a dictionary where the input states are the keys and the corresponding output states the values, `{psi_in : psi_out}`. Here we include python code for a small example, the two-qubit problem from Eq. 2.



```

from bqskit import compile
from bqskit.qis import StateVector
from bqskit.qis import StateSystem

in1 = StateVector(np.array([1,0,0,0]))
in2 = StateVector(np.array([0,0,0,1]))

out1 = StateVector(in1)
out2 = np.array([0,1,-1,0])/np.sqrt(2)
out2 = StateVector(out2)

system = StateSystem({in1:out1, in2:out2})

c = compile(system, optimization_level=2)

out1_c = c.get_statevector(in1)
out2_c = c.get_statevector(in2)

print(c.gate_counts)
print(out1_c)
print(out2_c)
print(out1_c.get_distance_from(out1))
print(out2_c.get_distance_from(out2))

>>> {U3Gate: 1, RYGate: 4, CNOTGate: 2}
>>> [ 1.00e+00+3.06e-07j  1.63e-05+8.47e-12j
      5.56e-06+3.77e-12j -1.56e-06-1.27e-07j]
>>> [-7.61e-06-1.54e-12j  7.07e-01+5.03e-08j
      -7.07e-01+1.29e-07j  7.86e-06+3.48e-12j]
>>> 2.999380743773372e-10
>>> 2.644338081836395e-10

```

The “off-the-shelf” `compile` function allows some error in the output vectors as seen here, but the user can request higher precision.

While one solution to this particular problem, the circuit in Eq. (3), could be found relatively easily with pencil and paper, a solution with CNOT as the only two-qubit gate is much more challenging to find by hand; the circuit in Eq. (4) is the solution generated by the code just above. Furthermore, the BQSKit implementation of multi-state preparation provides optimal or near-optimal solutions for three- and four-qubit problems that are infeasible to solve by hand.

### B. Numerical simulation examples

In Sec. II A, we considered the number of entangling gates such as CNOT and CH required to solve some simple multi-state preparation problems on two qubits. These sample problems exemplify the theoretically known number of CNOTs required, namely at most one for single-state preparation [40], at most two for a 1 qubit  $\rightarrow$  2 qubit isometry [35], and at most three for a full unitary operation [38], [39].

Here we demonstrate our numerical approach on a more difficult case, multi-state preparation for three-qubit and four-qubit states. We find how the required number of CNOT gates scales with the number of states to simultaneously prepare, and we also show that when the number of states is a power of 2, so that the problem is equivalent to isometry synthesis, we find substantially lower CNOT counts than does the “top-down” matrix decomposition approach of [35] (used in UniversalQCompiler [36], Qiskit [24], and Cirq [25]); for large  $m$ , the reduction is around 30-40%.

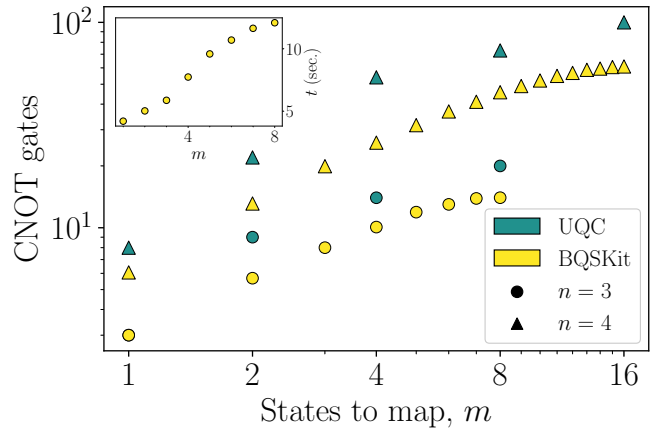


Fig. 2. Number of CNOT gates for the shortest circuit found to solve the multi-state mapping problem with 3 (circle markers) and 4 qubits (triangles), and with varying numbers of states to map,  $m$ . Yellow points show the CNOT counts for numerical optimization via instantiation with the cost function from Eq. (12) as implemented in BQSKit [28], while teal markers show the CNOT counts from the software package UniversalQCompiler [36]. Evidently, the numerical approach finds significantly shorter circuits. The inset shows the mean computation time to numerically solve the multi-state problem for  $n = 3$  qubits as a function of the number of states  $m$ , showing a monotonic and approximately linear dependence.

To be precise, we carry out the following numerical experiment  $N_{\text{shots}}$  times, with  $n = 3, 4$ :

- 1) Generate a  $2^n \times 2^n$  unitary operator,  $U$ .
- 2) For  $m \in \{1, \dots, 2^n\}$ , perform multi-state preparation where the  $|v_i\rangle$  are the first  $m$  standard basis vectors and  $|w_i\rangle$  the first  $m$  columns of  $U$ . Find the minimum number of CNOT gates needed for each  $m$ .<sup>3</sup>

We show the results of the experiment, both for  $n = 3$  and  $n = 4$ , in Figure 2. The inset shows how the time required for the compilation scales with the number of states to be mapped,  $m$ . Evidently, if one needs a few states to be mapped correctly but does not care about the actions of the circuit on the rest of the Hilbert space, there is substantial freedom to shorten the required circuit by performing multi-state preparation rather than unitary synthesis, with the side benefit that the classical computation time to find a good circuit will also be reduced.

Recalling Example 1 from Section II A., there are two possible reasons for why shorter circuits are returned by our numerical instantiation approach than by top-down matrix decomposition: (a) among all unitary transformations consistent with the multi-state problem, numerical instantiation may be finding one that has a shorter optimal circuit implementation; and/or (b) the top-down approach may be finding a longer than necessary circuit to implement its unitary transformation. If only (b) were the case, but not (a), we could feed the output circuits from the top-down approach into a numerical compiler, such as the one included with BQSKit, and find equivalent circuits of the same depth as those from the numerical instantiation approach. When we perform this additional

<sup>3</sup>In this numerical experiment, we assume all-to-all connectivity. BQSKit also supports multi-state preparation with any connectivity graph.

experiment, we find that, while some circuit depths are indeed reduced, for all cases with  $m < 2^n$  we still find longer circuits for top-down plus recompiling compared with direct numerical instantiation. (For example, with  $n = 4$ ,  $m = 2^3$ , UniversalQCompiler outputs a circuit with 73 CNOTs, which BQSKit compiles down to 57. The direct numerical approach gives 46 CNOTs.) We conclude that the numerical instantiation approach produces fundamentally better solutions that cannot be matched by post-processing the output circuits of the top-down approach.

Finally, we comment on the level of precision in the two approaches. The matrix decomposition approach returns circuits that produce the desired isometry exactly up to machine precision. On the other hand, the numerical approach produces circuits that match the desired isometry only up to a specified threshold of the cost function; we currently target a cost threshold that returns isometries accurate to about  $10^{-4}$  in each matrix element. This reduction in precision should not be viewed as a drawback of the method. Rather, as long as these errors are significantly smaller than those that arise in running the circuit on hardware, the imprecision will have no meaningful impact on the results of real simulations; currently, for a circuit with dozens of two-qubit gates, the hardware errors are orders of magnitude larger. Furthermore, by allowing some imprecision, it may be possible to produce shorter circuits than if full precision is required, and thus we can choose to slightly increase the error in the theoretical circuit and in exchange substantially lower the noise and infidelity from hardware, a worthwhile tradeoff.

## VII. DISCUSSION

We have studied a very general quantum circuit synthesis problem that falls between the commonly considered cases of single-state preparation and full unitary synthesis, and which includes each of those problems as a special case. We prove that a set of input states  $\{|v_i\rangle\}$  can be exactly mapped to a set of output states  $\{|w_i\rangle\}$  if and only if the overlaps for the two sets of states are equal,  $\langle v_i|v_j\rangle = \langle w_i|w_j\rangle$  for all pairs  $i, j$ . Furthermore, we show that to numerically optimize a quantum circuit to perform the state mapping, a good cost function is given by equation (12). We have implemented both the solvability check and the circuit optimization in Berkeley Lab’s open-source quantum compiler, BQSKit, and we demonstrated the strong performance of the numerical optimizer on three- and four-qubit problems.

Multi-state preparation is a useful tool for a variety of applications, including preparation of efficient circuits to implement isometries and hence quantum channels. Other applications include preparation of macroscopic superposition states and reduction of circuits for time evolution in physics and chemistry simulations by preparing a circuit only for the part of the time evolution operator that acts within a specific block of a block-diagonal Hamiltonian.

With the addition of multi-state preparation to BQSKit in version 1.1, we now support the full range of pure-state preparation and unitary synthesis. Planned future updates to BQSKit

will add further capabilities, including direct compilation of non-unitary channels as well as state preparation making use of measurement and reset.

## ACKNOWLEDGMENT

We thank Huo Chen for helpful conversations and for pointing out several useful references. This work was supported Office of Science, Office of Advanced Scientific Computing Research Accelerated Research for Quantum Computing Program of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory.

## REFERENCES

- [1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [2] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: <https://doi.org/10.22331/q-2018-08-06-79>
- [3] “Quantinuum Sets New Record with Highest Ever Quantum Volume”. [Online]. Available: <https://www.quantinuum.com/news/quantinuum-sets-new-record-with-highest-ever-quantum-volume>
- [4] “IONQ Aria: Practical Performance”. [Online]. Available: <https://ionq.com/resources/ionq-aria-practical-performance>
- [5] M. Saffman, I. I. Beterov, A. Dalal, E. J. Páez, and B. C. Sanders, “Symmetric rydberg controlled-z gates with adiabatic pulses,” *Phys. Rev. A*, vol. 101, p. 062309, Jun 2020. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.101.062309>
- [6] I. S. Madjarov, J. P. Covey, A. L. Shaw, J. Choi, A. Kale, A. Cooper, H. Pichler, V. Schkolnik, J. R. Williams, and M. Endres, “High-fidelity entanglement and detection of alkaline-earth rydberg atoms,” *Nature Physics*, vol. 16, no. 8, pp. 857–861, Aug 2020. [Online]. Available: <https://doi.org/10.1038/s41567-020-0903-z>
- [7] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletic, and M. D. Lukin, “High-fidelity parallel entangling gates on a neutral atom quantum computer,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.05420>
- [8] A. R. Mills, C. R. Guinn, M. J. Gullans, A. J. Sigillito, M. M. Feldman, E. Nielsen, and J. R. Petta, “Two-qubit silicon quantum processor with operation fidelity exceeding 99%,” *Science Advances*, vol. 8, no. 14, p. eabn5130, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/sciadv.abn5130>
- [9] A. Kandala, K. X. Wei, S. Srinivasan, E. Magesan, S. Carnevale, G. A. Keefe, D. Klaus, O. Dial, and D. C. McKay, “Demonstration of a high-fidelity cnot gate for fixed-frequency transmons with engineered zz suppression,” *Phys. Rev. Lett.*, vol. 127, p. 130501, Sep 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.127.130501>
- [10] L. B. Nguyen, G. Koolstra, Y. Kim, A. Morvan, T. Chistolini, S. Singh, K. N. Nesterov, C. Jünger, L. Chen, Z. Pedramrazi, B. K. Mitchell, J. M. Kreikebaum, S. Puri, D. I. Santiago, and I. Siddiqi, “Blueprint for a high-performance fluxonium quantum processor,” *PRX Quantum*, vol. 3, p. 037001, Aug 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PRXQuantum.3.037001>
- [11] I. N. Moskalenko, I. A. Simakov, N. N. Abramov, A. A. Grigorev, D. O. Moskalev, A. A. Pishchimova, N. S. Smirnov, E. V. Zikiy, I. A. Rodionov, and I. S. Besedin, “High fidelity two-qubit gates on fluxoniums using a tunable coupler,” *npj Quantum Information*, p. 130, Nov. 2022. [Online]. Available: <https://doi.org/10.1038/s41534-022-00644-x>
- [12] Google Quantum AI, “Suppressing quantum errors by scaling a surface code logical qubit,” *Nature*, vol. 614, no. 7949, pp. 676–681, Feb 2023. [Online]. Available: <https://doi.org/10.1038/s41586-022-05434-1>
- [13] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, “Optimal quantum circuit synthesis from controlled-unitary gates,” *Phys. Rev. A*, vol. 69, p. 042309, Apr 2004. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.69.042309>

- [14] V. Shende, S. Bullock, and I. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, 2006. [Online]. Available: <https://ieeexplore.ieee.org/document/1629135>
- [15] D. Camps and R. Van Beeumen, "Approximate quantum circuit synthesis using block encodings," *Phys. Rev. A*, vol. 102, p. 052411, Nov 2020. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.102.052411>
- [16] M. G. Davis, E. Smith, A. Tudor, K. Sen, I. Siddiqi, and C. Iancu, "Towards optimal topology aware quantum circuit synthesis," *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 223–234, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9259942>
- [17] E. Younis, K. Sen, K. A. Yelick, and C. Iancu, "Qfast: Conflating search and numerical optimization for scalable quantum circuit synthesis," *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 232–243, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9605287>
- [18] E. Smith, M. G. Davis, J. Larson, E. Younis, L. B. Ofstie, W. Lavrijsen, and C. Iancu, "Leap: Scaling numerical optimization based synthesis using an incremental approach," *ACM Transactions on Quantum Computing*, vol. 4, no. 1, feb 2023. [Online]. Available: <https://doi.org/10.1145/3548693>
- [19] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles, "Machine learning of noise-resilient quantum circuits," *PRX Quantum*, vol. 2, p. 010324, Feb 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PRXQuantum.2.010324>
- [20] P. Rakyta and Z. Zimborás, "Efficient quantum gate decomposition via adaptive circuit compression," 2022. [Online]. Available: <https://arxiv.org/abs/2203.04426>
- [21] E. Younis and C. Iancu, "Quantum circuit optimization and transpilation via parameterized circuit instantiation," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 465–475. [Online]. Available: <https://ieeexplore.ieee.org/document/9951320>
- [22] S. Ashhab, N. Yamamoto, F. Yoshihara, and K. Semba, "Numerical analysis of quantum circuits for state preparation and unitary operator synthesis," *Phys. Rev. A*, vol. 106, p. 022426, Aug 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.106.022426>
- [23] X. Sun, G. Tian, S. Yang, P. Yuan, and S. Zhang, "Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10044235>
- [24] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023. [Online]. Available: <https://doi.org/10.5281/zenodo.2573505>
- [25] Cirq Developers, "Cirq," Dec. 2022, See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>. [Online]. Available: <https://doi.org/10.5281/zenodo.7465577>
- [26] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, "t—ket: a retargetable compiler for nisq devices," *Quantum Science and Technology*, vol. 6, no. 1, p. 014003, nov 2020. [Online]. Available: <https://dx.doi.org/10.1088/2058-9565/ab8e92>
- [27] E. Younis, C. C. Iancu, W. Lavrijsen, M. Davis, and E. Smith, "Berkeley quantum synthesis toolkit (bqskit) v1," 4 2021. [Online]. Available: <https://www.osti.gov/biblio/1785933>
- [28] BQSKit code repository. [Online]. Available: <https://github.com/BQSKit/bqskit>
- [29] W. P. Schleich and M. G. Raymer, "Special issue on quantum state preparation and measurement," *Journal of Modern Optics*, vol. 44, no. 11–12, pp. 2021–2022, 1997. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/09500349708231863>
- [30] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, "Quantum circuits with uniformly controlled one-qubit gates," *Phys. Rev. A*, vol. 71, p. 052330, May 2005. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.71.052330>
- [31] M. Plesch and C. Brukner, "Quantum-state preparation with universal gate decompositions," *Phys. Rev. A*, vol. 83, p. 032302, Mar 2011. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.83.032302>
- [32] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, "A divide-and-conquer algorithm for quantum state preparation," *Scientific Reports*, vol. 11, no. 1, p. 6329, Mar 2021. [Online]. Available: <https://doi.org/10.1038/s41598-021-85474-1>
- [33] X.-M. Zhang, T. Li, and X. Yuan, "Quantum state preparation with optimal circuit depth: Implementations and applications," *Phys. Rev. Lett.*, vol. 129, p. 230504, Nov 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.129.230504>
- [34] E. Knill, "Approximation by quantum circuits," 1995. [Online]. Available: <https://arxiv.org/abs/quant-ph/9508006>
- [35] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, "Quantum circuits for isometries," *Phys. Rev. A*, vol. 93, p. 032318, Mar 2016. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.93.032318>
- [36] R. Iten, O. Reardon-Smith, E. Malvetti, L. Mondada, G. Pauvert, E. Redmond, R. S. Kohli, and R. Colbeck, "Introduction to UniversalQCompiler," 2021. [Online]. Available: <https://arxiv.org/abs/1904.01072>
- [37] P. Yuan and S. Zhang, "Optimal (controlled) quantum state preparation and improved unitary synthesis by quantum circuits with any number of ancillary qubits," *Quantum*, vol. 7, p. 956, Mar. 2023. [Online]. Available: <https://doi.org/10.22331/q-2023-03-20-956>
- [38] G. Vidal and C. M. Dawson, "Universal quantum circuit for two-qubit transformations with three controlled-not gates," *Phys. Rev. A*, vol. 69, p. 010301, Jan 2004. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.69.010301>
- [39] F. Vatan and C. Williams, "Optimal quantum circuits for general two-qubit gates," *Phys. Rev. A*, vol. 69, p. 032315, Mar 2004. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.69.032315>
- [40] M. Žnidarič, O. Giraud, and B. Georgeot, "Optimal number of controlled-not gates to generate a three-qubit state," *Phys. Rev. A*, vol. 77, p. 032320, Mar 2008. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.77.032320>
- [41] copper.hat ([https://math.stackexchange.com/users/27978/copper\\_hat](https://math.stackexchange.com/users/27978/copper_hat)), "Unitary map between sets of vectors," Mathematics Stack Exchange, uRL:<https://math.stackexchange.com/q/583653> (version: 2013-11-27). [Online]. Available: <https://math.stackexchange.com/q/583653>
- [42] M. Weiden, J. Kallor, J. Kubiatiowicz, E. Younis, and C. Iancu, "Wide quantum circuit optimization with topology aware synthesis," in *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*, 2022, pp. 1–11. [Online]. Available: <https://ieeexplore.ieee.org/document/10025533>
- [43] D. A. Roberts and B. Yoshida, "Chaos and complexity by design," *Journal of High Energy Physics*, vol. 2017, p. 121, Apr 2017. [Online]. Available: [https://doi.org/10.1007/JHEP04\(2017\)121](https://doi.org/10.1007/JHEP04(2017)121)
- [44] L. Wossnig, Z. Zhao, and A. Prakash, "Quantum linear system algorithm for dense matrices," *Phys. Rev. Lett.*, vol. 120, p. 050502, Jan 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.120.050502>
- [45] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "q-means: A quantum algorithm for unsupervised machine learning," in *Advances in Neural Information Processing Systems*, H. Wallach, L. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/16026d60ff9b54410b3345b403afd226-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/16026d60ff9b54410b3345b403afd226-Paper.pdf)
- [46] H. J. Briegel and R. Raussendorf, "Persistent entanglement in arrays of interacting particles," *Phys. Rev. Lett.*, vol. 86, pp. 910–913, Jan 2001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.86.910>
- [47] K. Klymko, C. Mejuto-Zaera, S. J. Cotton, F. Wudarski, M. Urbanek, D. Hait, M. Head-Gordon, K. B. Whaley, J. Moussa, N. Wiebe, W. A. de Jong, and N. M. Tubman, "Real-time evolution for ultracompact hamiltonian eigenstates on quantum hardware," *PRX Quantum*, vol. 3, p. 020323, May 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PRXQuantum.3.020323>
- [48] C. L. Cortes and S. K. Gray, "Quantum krylov subspace algorithms for ground- and excited-state energy estimation," *Phys. Rev. A*, vol. 105, p. 022417, Feb 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.105.022417>
- [49] A. Auerbach, *Ground States of the Heisenberg Model*. New York, NY: Springer New York, 1994. [Online]. Available: <https://doi.org/10.1007/978-1-4612-0869-3>
- [50] W. F. Stinespring, "Positive functions on  $c^*$ -algebras," *Proceedings of the American Mathematical Society*, vol. 6, no. 2, pp. 211–216, 1955. [Online]. Available: <http://www.jstor.org/stable/2032342>

- [51] L. Del Re, B. Rost, A. F. Kemper, and J. K. Freericks, "Driven-dissipative quantum mechanics on a lattice: Simulating a fermionic reservoir on a quantum computer," *Phys. Rev. B*, vol. 102, p. 125112, Sep 2020. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.102.125112>
- [52] Z. Hu, R. Xia, and S. Kais, "A quantum algorithm for evolving open quantum dynamics on quantum computing devices," *Scientific Reports*, vol. 10, no. 1, p. 3301, Feb 2020. [Online]. Available: <https://doi.org/10.1038/s41598-020-60321-x>