Quantum compiling with diffusive sets of gates

Y. Zhiyenbayev,[1] V. M. Akulin,[2,3] and A. Mandilara[1]

[1]*Department of Physics, School of Science and Technology, Nazarbayev University, 53 Kabanbay Batyr Avenue, Astana, 010000, Republic of Kazakhstan*

[2]*Laboratoire Aimé-Cotton CNRS UMR 9188, L'Université Paris-Sud et L'École Normale Supérieure de Cachan, Bât. 505, Campus d'Orsay, 91405 Orsay Cedex, France*

[3]*Institute for Information Transmission Problems of the Russian Academy of Science, Bolshoy Karetny per. 19, Moscow, 127994, Russia*

(Received 15 January 2018; published 24 July 2018)

Given a set of quantum gates and a target unitary operation, the most elementary task of quantum compiling is the identification of a sequence of the gates that approximates the target unitary to a determined precision ε. The Solovay-Kitaev theorem provides an elegant solution which is based on the construction of successively tighter "nets" around unity comprised of successively longer sequences of gates. The procedure for the construction of the nets, according to this theorem, requires accessibility to the inverse of the gates as well. In this work, we propose a method for constructing nets around unity without this requirement. The algorithmic procedure is applicable to sets of gates which are diffusive enough, in the sense that sequences of moderate length cover the space of unitary matrices in a uniform way. We prove that the number of gates sufficient for reaching a precision ε scales as $\log(1/\varepsilon)^{\log 3/\log 2}$ while the precompilation time is increased as compared to that of the Solovay-Kitaev algorithm by the exponential factor $3/2$.

DOI: 10.1103/PhysRevA.98.012325

Approximation up to a given accuracy of an arbitrary unitary transformation by a series of standard transformations is an important ingredient of programming of quantum computers, which was formulated and solved [1,2] in the case where the set of $\mathcal{M}$ predetermined standard transformations contains both direct operations and their inverses. The so-called Solovay-Kitaev (SK) theorem provides the proof of existence together with the method for constructing the solution. Based on the elements in the proof of the SK theorem, the Dawson-Nielsen (DNSK) algorithm [3] provides the exact steps for identifying a series of length L, which scales with the required accuracy ε as $O[\log(1/\varepsilon)^{3.97}]$, and with running time as $O[\log(1/\varepsilon)^{2.71}]$. For the special case of qubits, different techniques have been suggested [4,5] improving the running time of this algorithm, while in the general case it has been proved [2,6] that the use of extra ancilla qubits further improves the relations of both the length and the running time, with the accuracy ε.

Here we address the question [3] whether it is possible to generalize the results of the SK theorem onto the case where the set of predetermined operations does not contain the inverses. In view of the fast development of quantum technologies, this problem has theoretical but also practical interest since experimentalists sometimes do not have access to inverse operations. For instance, time is the main quantum control (positive) parameter and one may employ it to construct both a gate and its inverse. On the other hand decoherence effect inducing constraints in time might prevent one from doing so in practice.

Progress on the possibility of extending the SK theorem has been reported in [7,8] and also in a very recent related work [9]. Our answer is also positive and conditional on a specific property of the given set. We require that sequences of gates of moderate length (composed of 15–20 gates) cover the space of unitary matrices in a uniform way. This property of sets was initially investigated in [10] and criteria have been formally developed in [11] in the case where the inverse operators are included in the set. More recently the powerfulness of such sets, so-called efficiently universal, over just computationally universal ones, has been demonstrated in [12] for the problem of quantum compiling. In view of lack of formal criteria for the case where the inverses are not available, we avoid the use of the term of "efficiently universal sets" and we employ instead the loosely defined term of "diffusive sets." As in [12], we require that such sets are composed of noncommuting computationally universal gates and in addition that sequences of *moderate* length composed of the gates of these sets cover densely and uniformly the space of unitary matrices. For the special case of qubits where the property of diffusivity can be visually over viewed (see Fig. 1), we have found out that a considerable number of computationally universal sets are also diffusive (our estimation $\approx 30\%$). In addition, we have noticed that by multiplying a set of computationally universal gates with a random unitary matrix [13], one transforms with high probability the former into a diffusive set. Physically this random matrix may stand for the free evolution of the quantum system that interpolates the control actions which generate the gate operations.

Let us now briefly present the idea of our approach and the structure of this manuscript, assuming that the reader is a little familiar with the proof of the SK theorem. As in the latter, our aim is to develop a universal algorithmic method for constructing a series of successive $(\varepsilon_i, \varepsilon_i^2)$ nets around the identity and with the requirement $\varepsilon_{i+1} = \varepsilon_i^2$ [14]. A $(\varepsilon_i, \varepsilon_i^2)$ net signifies that in the ε_i neighborhood of the identity operator there are sequences of gates of length L_i and for each of these sequences there is another point at distance less than ε_i^2. After

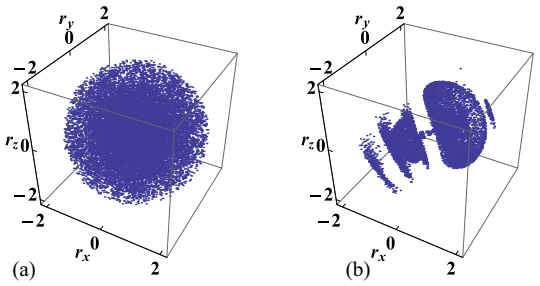

FIG. 1. Geometric representation of all sequences $\widehat{T}$ of length 17 generated by two different computationally universal sets of (two) single-qubit gates. (a) A diffusive set and (b) a nondiffusive one.

we introduce our notation and a geometric picture that permits us to interpreter the nets and our methods in a geometric fashion, we explain how the nets can be used to improve the approximation for given $\widehat{U}$ in the standard recursive way. Then we present our main result, a method for successively producing the nets via "shrinking." We justify the limits of this method using the theory of random walks and we confirm our theoretical predictions with a compiling example for phase rotation gates.

Throughout this work we consider that $\mathcal{M}$, the set of given gates, contains just two unitary operations, $\widehat{A} = \exp[-i\widehat{O}_0]$ and $\widehat{B} = \exp[-i\widehat{O}_1]$, determined by two Hermitian operators $\widehat{O}_{i=0,1}$ in a Hilbert space of $\mathcal{N}$ dimensions. Each sequence of k transformations, i.e., of length $L = k$, picked from the set $\mathcal{M}$ can be encoded by a binary number $\mu = \{f, j, \ldots p, q\}$ in k registers

$$\widehat{T}_{f,j,\ldots,p,q} = e^{-i\widehat{O}_f} e^{-i\widehat{O}_j} l \cdots e^{-i\widehat{O}_q} e^{-i\widehat{O}_p}, \tag{1}$$

where $f, j, \ldots p, q = 0, 1$. One may attribute a $d = \mathcal{N}^2 - 1$ dimensional real vector $\overrightarrow{r} = \{r_n\}$ to such a sequence $\widehat{T}$, and in general to any unitary transformation $\widehat{U}$, and this geometric representation is particularly useful for our analysis. The "mapping" can be achieved by casting $\log \widehat{U}$ in the sum of the $su(\mathcal{N})$ traceless generators $\widehat{g}_n$:

$$-i \log \widehat{U} = \sum_{n=1}^{d} r_n \widehat{g}_n = \overrightarrow{r} \cdot \overrightarrow{g}. \tag{2}$$

If the generators are normalized, $\mathrm{Tr}(\widehat{g}_i \cdot \widehat{g}_j) = \delta_{ij}$, then $r_n = \mathrm{Tr}(-i \log \widehat{U} \cdot \widehat{g}_n)$. The unity operator corresponds to $\overrightarrow{r} = 0$, while all other unitary operations corresponds to points in the hypersphere around it. Some additional information about this mapping is provided in the Appendix and hereafter we sometimes refer to a unitary $\widehat{U}$ as to a *point*, implying the edge of the corresponding vector $\overrightarrow{r}_{\widehat{U}}$ in the d-dimensional space. For single-qubit operations the a vector $\overrightarrow{r} = (r_x, r_y, r_z)$ is three dimensional and can be visualized, and in Fig. 1 we use this possibility to observe the difference between a diffusive and a nondiffusive set of computationally universal pairs of gates.

The representation of unitary operators as vectors leads naturally to the following definition of *distance* $\mathcal{D}$ between two unitary operators:

$$\mathcal{D}(\widehat{U}_1, \widehat{U}_2) = |\overrightarrow{r}_{(\widehat{U}_1^{-1} \cdot \widehat{U}_2)}|, \tag{3}$$

where the right-hand side of the equation describes the length of the vector for the unitary matrix $\widehat{U}_1^{-1} \cdot \widehat{U}_2$. By definition the following property holds: $\mathcal{D}(\widehat{U}_1, \widehat{U}_2) = \mathcal{D}(\widehat{U}_1 \cdot \widehat{U}_2^{-1}, \widehat{I})$. In our proofs where we are mostly interested in the regime of *small* distances, we use $\mathcal{D}$ as a measure of distance between unitary operators; in fact we have tested that this perfectly correlates with other measures in use [1,15].

Using the introduced notation we can more clearly state now the objective of the quantum compiling task and the utility of constructing successive nets.

Quantum Compiling: Given an arbitrary unitary transformation $\widehat{U}$, identify a sequence $\widehat{T}_{f,j,\ldots,p,q}$ of gates from the set $\mathcal{M}$, of a total length L, which approximates $\widehat{U}$ to a given accuracy ε, or else,

$$\mathcal{D}(\widehat{T}_{\underbrace{f, j, \ldots, p, q}_{L}}, \widehat{U}) < \varepsilon. \tag{4}$$

Different strategies can be in principle designed to solve this problem and each of them is characterized by three relations: the relation between the total length L of the sequence and the achieved precision ε, the relation between the running time of the algorithm and ε, and the precompilation time. All these relations cannot simultaneously scale in an optimal way and there is an apparent interplay among them. For instance, the simplest strategy is the exploration of all possible sequences of a given length and identification of a sequence closest to the required transformation $\widehat{U}$. This is the well-known coverage problem, typical of coding theory, which yields indeed the shortest sequence $L \propto \log(1/\varepsilon)$ for a given accuracy. However, the identification of such a sequence requires a time-consuming work of exploration of all possibilities, whereas the running time scales exponentially with L thus making the approach intractable in the high accuracy limit (however, see [15] for an enhanced protocol). This strategy does not require precompilation time since every new $\widehat{U}$ requires a new search; but on the other hand, and for the same reason, this is not a universal strategy.

The SK theorem for sets including inverses offers [3] a balance between the three relations, which could be possibly optimal; both the length of sequence and running time scale poly-logarithmically with ε and notably these are independent of the dimension of the Hilbert space $\mathcal{N}$, while the precompilation time scales polynomially with ε and exponentially with $d = \mathcal{N}^2 - 1$. The SK theorem, and in consequence the DNSK algorithm, are heavily based on a "succesful" construction of a sequence of $(\varepsilon_i, \varepsilon_i^2)$nets around the identity. Once these nets are constructed and stored, one can perform a standard procedure (ignoring always the telescoping step and assuming $\varepsilon_{i+1} = \varepsilon_i^2$) to approximate a given unitary $\widehat{U}$:

(i) One first performs a number of relatively short sequences of transformations $\widehat{T}_{\underbrace{f, j, \ldots p, q}_{r}}$ of length $\approx$ 16–20 that serve as initial reference points. Let us call this net *the sampling net*.

(ii) For the given $\widehat{U}$ one has to identify by exhaustive search in the sampling net the closest reference point $\widehat{T}^{(0)}$ such that distance $\mathcal{D}(\widehat{U}^{-1}\widehat{T}^{(0)}, \widehat{I}) < \varepsilon_0$. If it is not the case, one should restart augmenting the length r.

(iii) One can then use the net $(\varepsilon_0, \varepsilon_0^2)$ in order to identify the sequence $\widehat{T}^{(\varepsilon_0)}$ such as $\mathcal{D}(\widehat{U}^{-1}\widehat{T}^{(0)}\widehat{T}^{(\varepsilon_0)}, \widehat{I}) < \varepsilon_0^2$.

(iv) The procedure is repeated n times and at the last step one arrives at the desired result: a sequence of gates $\widehat{A}$ and $\widehat{B}$

$$\widehat{T}^{(0)}\widehat{T}^{(\varepsilon_0)} \ldots \widehat{T}^{(\varepsilon_n)}$$

of length $r + \sum_{i=0}^{n} L_i$ that reproduces the given unitary in $\varepsilon = \varepsilon_n^2$ approximation: $\mathcal{D}(\widehat{U}^{-1}\widehat{T}^{(0)}\widehat{T}^{(\varepsilon_0)} \ldots \widehat{T}^{(\varepsilon_n)}, \widehat{I}) < \varepsilon$.

The dependence of the final length L with ε is determined by the relation between L_{i+1} and L_i. If $L_{i+1} = ML_i$ where $M \in \mathbb{N}$, it is straightforward to prove the dependence is the desired, poly-logarithmic one:

$$L_n = r\left[\log(1/\varepsilon_n)/\log(1/\varepsilon_0)\right]^{\log M/\log 2}. \tag{5}$$

Now, let us turn to the main question of how to construct the sequence of nets and let us assume hereafter that the given set of gates (including the inverses or not) is a diffusive one. The latter condition permits us to consider the points on these nets—even on the sampling one—as uniformly distributed according to the Haar measure. Using then the formula for the volume of a sphere in the d-dimensional space, one can calculate that the number of required points is for a sufficient density up to a constant factor $(\varepsilon_i)^{-d}$. We desire to employ the $K_i \propto (\varepsilon_i)^{-d}$ points (sequences) of the ε_i net to identify the $K_{i+1} \propto (\varepsilon_i)^{-2d}$ points of the consecutive ε_{i+1} net.

We first consider the case where the inverses are available in the set, we follow the main idea introduced in [1,2] in order to arrive at a simplified version of the DNSK algorithm. The key idea in [1,2] is to apply the *normal commutator* to a pair of sequences $\widehat{T}_{(1)}^{(\varepsilon_i)}$ and $\widehat{T}_{(2)}^{(\varepsilon_i)}$ of the ε_i net, employing also the inverses of these, $\widehat{T}_{(1)}^{(\varepsilon_i)^{-1}}$ and $\widehat{T}_{(2)}^{(\varepsilon_i)^{-1}}$, which are naturally included in the same net. By definition a normal commutator is $\widehat{T}_{(1)}^{(\varepsilon_i)}\widehat{T}_{(2)}^{(\varepsilon_i)}\widehat{T}_{(1)}^{(\varepsilon_i)^{-1}}\widehat{T}_{(2)}^{(\varepsilon_i)^{-1}}$ and the result of this product is a new sequence at distance less than ε_i^2 from unity. This new point or sequence can be included in the consequent ε_{i+1} net. The normal commutator thus naturally leads to the so-called shrinking of the initial net. The number of distinct normal commutators that can be formed by K_i points of the ε_i net matches the number of points required in the ε_{i+1} net, independently of the dimension of the Hilbert space, and there is no need for additional "search" steps during the precompilation stage. Now concerning the scaling of the length of the sequences with the ith order of the net, $L_{\varepsilon_{i+1}} = 4L_{\varepsilon_i}$, and by inserting $M = 4$ into Eq. (5) we arrive at a quadratic dependence between length and accuracy: $L_n \sim [(\log(1/\varepsilon_n)]^2$. In what has been described we have ignored the extra step of "telescoping" [14] and we name this simple and faster version *fast DNSK*. This simplified version can be compared with the algorithm that we suggest on the same grounds. In addition the requirement of diffusive characters of gates seems to partially compensate for the telescoping procedure (see standard deviation of the approximation in Fig. 2).

Now, let us consider the case where the inverses are not accessible and therefore the idea of normal commutators is not applicable. Let us start as before with the ε_i net and select at random M sequences from this net. Then construct a new sequence by taking one of the $(M!)$ products of these sequences. If ε_i is small enough one may interpret, in

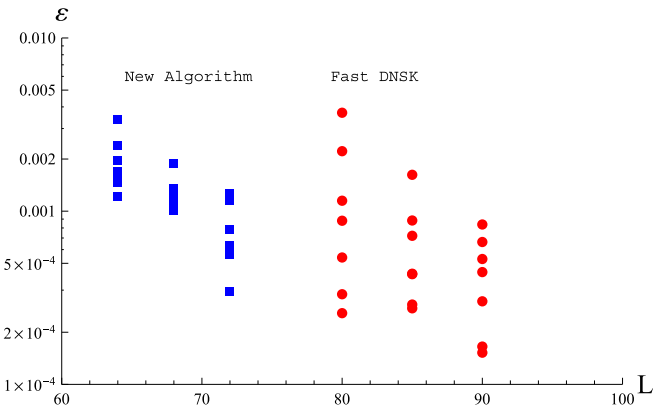

FIG. 2. Accuracy of approximation ε of phase rotation gates R_{2^d} for $d = 1, \ldots, 7$ by sequences of two diffusive gates, plotted vs length of the sequence. Blue squares: results obtained with the introduced algorithm. Red circular dots: results of the fast DNSK. Different columns correspond to different initial lengths r of the sequences in the sampling net. From left to right: $r = 16, 17, 18, 16, 17, 18$.

approximation $O(\varepsilon_i^2)$, this new point as a result of an M-steps random *walk* in the d -dimensional space (see Lemma in the Appendix). More precisely the steps of this random walk are the vectors (sequences) of the ε_i net, which are isotropic in the d-dimensional space and their size is in the interval $[0, \varepsilon]$ with a standard deviation of the order ε as well. If now all M products are produced from points (sequences) of the ε_i net, the resulting $(\varepsilon_i)^{-Md}$ new points are going to follow the distribution of a random walk and diffuse out of the unity. Such random walks have been well studied (see, for instance, [16]) and it is straightforward to derive the probability of finding a new point or sequence at distance $r = |\vec{r}|$ from the origin of the hypersphere after M steps,

$$P_M(r) = 2\left(\frac{d}{2M\varepsilon^2}\right)^{d/2} \frac{r^{d-1}}{\Gamma\left(\frac{d}{2}\right)} e^{-\frac{dr^2}{2M\varepsilon^2}}. \tag{6}$$

To build the ε_{i+1} net, one needs to post-select from the new diffused distribution of points the ones at distance $\mathcal{D} < \varepsilon_i^2$ from the origin.

To claim that the suggested method for shrinking is applicable we need though to answer three questions: (1) What is the minimum number of steps M that provides the required density of points for the consequent ε_{i+1}-net? (2) How does the time for constructing the nets compare to the precompilation time of fast DNSK? (3) Is the quality of the produced ε_{i+1} net good enough to ensure the successful construction of the ε_{i+2} net?

It turns out that $M = 3$ gives sufficient density of points inside the radius ε_i^2 for any dimension d. To prove this statement we first calculate the cumulative distribution function for probability distribution Eq. (6), plug in ε_i^2, and arrive at

$$P_M(r < \varepsilon_i^2) = 1 - \frac{\Gamma\left(\frac{d}{2}, \frac{d\varepsilon_i^2}{2M}\right)}{\Gamma\left(\frac{d}{2}\right)}. \tag{7}$$

However, an approximate formula for the distribution in Eq. (7) is more easy to overview as

$$P_M(r < \varepsilon_i^2) \approx 2\left(\frac{d}{2M}\right)^{d/2} \frac{\varepsilon^d}{\Gamma\left(\frac{d}{2}\right)}. \tag{8}$$

The latter can be derived by noting that the maximum of the Rayleigh type of distribution in Eq. (6) is outside the region of $r < \varepsilon_i^2$ and thus its contribution can be ignored. If Eq. (8) is multiplied with the total number of points $(\varepsilon_i)^{-Md}$ resulting from the diffusion process, one arrives at a formula that provides the number of points at distance less than ε_i^2 from the origin (unity). For $M = 3$, the required (for sufficient density) order $(\varepsilon_i)^{-2d}$ is reached for any dimension d.

Since $M = 3$ is sufficient one needs to perform all triplet products of the sequences in the ε_i net and then post-select the points (sequences) for which $|\overrightarrow{r}| < \varepsilon_i^2$. The number of sequences of the ε_i net is approximately $K_i \propto (\varepsilon_i)^{-d}$ and therefore the number of necessary operations is $\propto (\varepsilon_i)^{-3d}$. In the fast DNSK the number of operations for constructing all normal commutators is $\propto (\varepsilon_i)^{-2d}$. We may thus conclude that the time in our method is increased exponentially by a factor of 1.5. We believe that this is a natural consequence of the fact that the two methods have the same running time, but our suggested method achieves better scaling of length with approximation [see Eq. (5) with $M = 3$ while for fast DNSK $M = 4$]. In the Appendix we additionally prove that it is very unlikely that the long precompilation time of the algorithm that we suggest here can be shortened. There we show that if the post-selection process on the points is replaced by a pre-selection process, then this problem maps into an NP hard problem, namely, to the 0–1 knapsack problem in d dimensions.

Addressing now the last question. The number of points inside the radius ε_i^2 is increasing as r^{d-1} [see Eq. (6)] and therefore has approximately the desired dependence of a uniform distribution. In addition, under our assumption of diffusive set, the new points should be distributed in an isotropic way. Here though we suggest an additional step to ensure isotropicity which we have found very useful in practice: for each point or sequence identified to belong in the ε_{i+1} net, construct *all cyclic permutations* of the gates in the sequence. Cyclic permutations leave the spectrum of the sequence intact, and the length of the vector $|\overrightarrow{r}_{\widehat{U}}|$ depends only on the spectrum of the corresponding unitary $\widehat{U}$. Therefore, cyclic permutations of sequence leave the distance $\mathcal{D}$ from unity invariant and the corresponding new points are distributed over the hyperspherical surface of the original point.

In the Fig. 2 we present quantum compiling results obtained with the proposed algorithm versus the fast DNSK. More precisely, we approximate the phase rotation gates,

$$R_{2^d} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^d} \end{pmatrix}, \text{ with } d = 1, \ldots, 7, \tag{9}$$

using the introduced algorithm and then the fast DNSK, keeping the parameters of produced nets very similar in both cases. For both methods we have used the same pair of diffusive gates (see Appendix), but naturally for the latter we have included the inverses. On each column the seven points describe the approximation of the seven phase rotations of Eq. (9). There is no correlation between the precision achieved and the order d of the phase gates and for this reason we have not marked with d the points on the plot. For each method we present three numerical results (three columns) that correspond to three different lengths of the initial sampling net $r = 16, 17, 18$, giving different lengths L to the final sequence that approximate the gate (horizontal axis on Fig. 2). For both methods we have used the sampling and the ε_0 nets. To quantify the accuracy ε we use as measure of distance $d_F(\widehat{U}_1, \widehat{U}_2) = \sqrt{\frac{2-|\mathrm{Tr}(\widehat{U}_1\widehat{U}_2^{-1})|}{2}}$, introduced in [15]. More technical details on this example can be found in the Appendix while the related programs can be downloaded from the site www.qubit.kz.

The numerical results in Fig. 2 confirm our theoretical prediction that the suggested algorithm provides better scaling of length with accuracy than the DNSK. On the contrary from the graph one cannot extract the scaling of the length with accuracy, Eq. (5). This would require results where different orders of nets are used; here we only change the parameters of the sampling net and we use the first net around unity for all the results. The accuracy achieved for different gates is not uniform because we do not employ a procedure for extracting extra points which would improve the quality of nets in terms of homogeneity. Therefore we think that our suggested algorithm can be further upgraded by adding this additional procedure and possibly other procedures which would extend its applicability to sets of gates which are not completely but close to being diffusive.

In conclusion, we have suggested an algorithmic procedure for generating nets of seq'uences of gates around unity under the condition that the given sets of gates are diffusive. This algorithm results in better scaling of length with the approximation than a simplified fast version of the DNSK algorithm does, and works in both the presence and absence of inverses. The improvement in scaling can be justified by the fact that there is an exponential counter-increase in precompilation time, as compared to the DNSK algorithm. This confirms an expected interplay between the relations characterizing algorithmic procedures solving the same problem. When the inverses are included in the set, the notion of diffusive sets converges to the notion of "efficiently computational sets" introduced in [12] and our results partially fulfill the predictions of that work concerning the considerable improvement of the scaling of length with accuracy. Finally, the accurate characterization of the diffusive property of a set of gates remains an interesting open question, deserving further investigation.

A.M. is thankful to M. Lukac for bringing this problem to her attention. A.M. and V.A. are thankful to G. Kabatyanski for pointing out the connection to the knapsack problem and to M. S. Byrd for many useful initial discussions. The authors are grateful to A. Harrow for his comments which have helped them to draw conclusions from preliminary results and to the unknown referee for much valuable feedback. A.M. and Y.Z. acknowledge financial support during this work from the Ministry of Education and Science (MES) of the Republic of Kazakhstan via the Contract No. 339/76-2015. A.M. also acknowledges financial support from Nazarbayev University ORAU grant "Dissecting the Collective Dynamics of Arrays of Superconducting Circuits and Quantum Metamaterials" and MES RK state-targeted program BR05236454.

APPENDIX

1. Additional information on the geometric representation of unitary matrices

The representation of a unitary $\widehat{U}$ via $\overrightarrow{r}$ in general ignores the global phase since $\widehat{g}_0 = \widehat{1}$ is excluded from the set of generators. On the other hand one obtains different vectors for $\widehat{U}$ and $-\widehat{U}$ and in this work we want to totally ignore the global phase. This problem can be resolved when the $\mathcal{N}$-dimensional quantum system stands for an assembly of n qubits, ($\mathcal{N} = 2^n$). In this case, the introduced geometric space of unitaries is a d-dimensional hypersphere of radius $|\overrightarrow{r}_{\max}| = 2^{\frac{n}{2}}\pi$ centered at $|\overrightarrow{r}_{\min}| = 0$. The unitary $-\widehat{I}$ is located on the outer surface while $\widehat{I}$ in the center of the hypersphere. This unwanted discrepancy can be corrected by the following mapping: if $|\overrightarrow{r}| > 2^{\frac{n}{2}-1}\pi$ then $\overrightarrow{r} \longrightarrow -\overrightarrow{r}(2^{\frac{n}{2}}\pi - |\overrightarrow{r}|)/|\overrightarrow{r}|$.

We add here a Lemma which can be proved using the Baker-Campbell-Hausdorff formula:

Lemma. If $|\overrightarrow{r}_{\widehat{U}_1}| < \varepsilon$ and $|\overrightarrow{r}_{\widehat{U}_2}| < \varepsilon$ where $\varepsilon << 1$, then

$$\overrightarrow{r}_{(\widehat{U}_1\cdot\widehat{U}_2)} = \overrightarrow{r}_{\widehat{U}_1} + \overrightarrow{r}_{\widehat{U}_2} + O(\varepsilon^2). \tag{A1}$$

2. Details of the example

To generate Fig. 2 we used a diffusive set $\mathcal{M}$ composed of the gates $\{\widehat{A}, \widehat{B}\}$. More precisely, $\widehat{A} = \widehat{H}\cdot\widehat{F}$ and $\widehat{B} = \widehat{T}\cdot\widehat{F}$ where $\widehat{H}$ is the Hadamard gate, $\widehat{T}$ the T gate, and $\widehat{F}$ a randomly generated unitary matrix

$$\widehat{F} = \begin{pmatrix} -0.40194 - i0.43507 & -0.36803 - i0.71674 \\ 0.36803 - i0.71674 & -0.40194 + i0.43507 \end{pmatrix}. \tag{A2}$$

For the fast SK we used the analogous set $\mathcal{M}'$ composed of the gates $\{\widehat{A}, \widehat{B}, \widehat{A}^{-1}, \widehat{B}^{-1}\}$.

To achieve the approximations to the phase rotation gates R_{2^d} for the case of the algorithm based on diffusion, we perform the following steps:

(i) We create all the sequences of length $L = 16$. This is the sampling net composed by $k = 2^{16}$ points.

(ii) From this net covering all space we select the points inside the radius $\varepsilon_s = 0.3$. The ε_s is calculated according to the formula

$$\varepsilon_s = \frac{2^{1/4}\sqrt{\pi}}{k^{1/3}}. \tag{A3}$$

(iii) We perform the diffusion process creating all triplets, and then we post-select the ones which are inside the radius $\varepsilon_0 = (\varepsilon_s)^2$. We add 45 permutations for each successful sequence. This way we create more points than the ones needed for ε_0 so we randomly select from these the sufficient number, $8/\varepsilon_0^3$.

(iv) We use the sampling and the ε_0 net to identify the sequences of total length 65 that approximate each of the seven gates.

(v) We repeat the procedure for initial lengths $L = 17$ and $L = 18$, to obtain better approximation with sequences of lengths 68 and 72 respectively. We note here that the whole procedure is very fast since of course the aimed precision is low.

For the fast DNSK algorithm the steps are identical apart from the fact that we construct the normal commutators instead of triple products. For consistency, we include the permutation step.

3. Pre-selecting instead of post-selecting

In the main text we have studied the straightforward method for achieving shrinking by performing a diffusion process followed up by post-selection. More precisely, our suggestion is to construct all possible triplets from the $K_i \propto (\varepsilon_i)^{-d}$ points or sequences of the ε_i net, calculate for each of the resulting sequences $|\overrightarrow{r}|$, and then post-select those with $|\overrightarrow{r}| < \varepsilon_i^2$. Is there a more efficient way for doing this? Let us first replace the post-selection by pre-selection noting the following:

Given the sequences $\widehat{T}_1^{(\varepsilon_i)}$, $\widehat{T}_2^{(\varepsilon_i)}$, and $\widehat{T}_3^{(\varepsilon_i)}$ with corresponding vectors $\overrightarrow{r}_1$, $\overrightarrow{r}_2$, and $\overrightarrow{r}_3$ which satisfy the condition $|\sum_{j=1}^3 \overrightarrow{r}_j| < \varepsilon_i^2$ then $\mathcal{D}(\widehat{T}_1^{\varepsilon_i}.\widehat{T}_2^{\varepsilon_i}.\widehat{T}_3^{\varepsilon_i}, \widehat{I}) < O(\varepsilon_i^2)$.

This pre-selection process on the points of the initial ε_i net closely resembles a known computational problem: the 0–1 knapsack problem in d dimensions. Let us briefly state this problem:

Given n d-dimensional vectors $\vec{v}_i$ with positive entries and $p_i > 0$ profit for each of them, and a d-dimensional bin $\vec{B}$ find the n-dimensional vector $\vec{x}$ with 0–1 entries such that (i) $\sum_{i=1}^n x_i p_i$ is maximized and (ii) it is subject to $\sum_{i=1}^n x_i \vec{v}_i \leqslant \vec{B}$.

The mapping of the pre-selection problem to the knapsack problem is almost straightforward: one needs to (a) make the entries of input points and vectors $\overrightarrow{r}_i$ from the ε_i net strictly positive (so that they can represent $\vec{v}_i$), (b) attribute a profit p_i to these vectors, and (c) adjust the entries of the bins $\vec{B}$ to the requirements of the ε_{i+1} net. The first task can be done by adding a fixed vector $\vec{v}_0$ with $|\vec{v}_0| > \varepsilon_i$ to all $\overrightarrow{r}_i$. Concerning the profit one can attribute the same profit to all input vectors, but instead of maximizing the total cost, just minimize it. Finally, the entries of the bin should be adjusted to $B_k = 3\vec{v}_0 + \varepsilon_{i+1}/\sqrt{d}$ for $k = 1, \ldots, d$.

It has been proven [17] that there is no fully polynomial time approximation scheme for d-dimensional knapsack and that this is an NP-hard problem. In addition there is no efficient polynomial time approximation scheme (EPTAS) [18] even for low dimensions as $d = 2$. We may conclude that the pre-selection process for our suggested method is not computationally tractable in d dimensions given also the fact that n (= K_i for our case) increases exponentially with d.

[1] A. Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997).

[2] A. Yu. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and Quantum Computation*, Vol. 47 of Graduate Studies in Mathematics, 1st ed. (American Mathematical Society, 2002).

[3] C. M. Dawson and M. A. Nielsen, Quant. Inf. Comput. **6**, 81 (2006).

[4] T. T. Pham, R. Van Meter, and C. Horsman, Phys. Rev. A **87**, 052332 (2013).

[5] A. Bocharov and K. M. Svore, Phys. Rev. Lett. **109**, 190501 (2012).
[6] V. Kliuchnikov, D. Maslov, and M. Mosca, Phys. Rev. Lett. **110**, 190502 (2013).
[7] I. S. B. Sardharwalla, T. S. Cubitt, A. W. Harrow, and N. Linden, arXiv:1602.07963
[8] In [7] the set of given operations $\mathcal{M}$ is complemented by the so-called Weyl operations, which are mutually orthogonal roots of unity. Since the inverse of a root of unity is just the same operator to some integer power, it seems that this assumption relaxes the condition of no access to inverse operations.
[9] A. Bouland and M. Ozols, arXiv:1712.09798
[10] V. I. Arnold and A. L. Krylov, Soviet Math. Dokl. **4**, 1 (1962).
[11] A. Lubotsky, R. Phillips, and P. Sarnak, I. Commun. Pure Appl. Math. **39**, S149 (1986).
[12] A. W. Harrow, B. Recht, and I. L. Chuang, J. Math. Phys. **43**, 4445 (2002).
[13] K. Zyczkowski and M. Kus, J. Phys. A Math. Gen. **27**, 4235 (1994).
[14] In the SK theorem [1] the sequence of nets follows a "weaker" relation, $\varepsilon_{i+1} = \varepsilon_i^{3/2}$, than the one targeted in this work. The reason is that for preserving a good quality of nets one needs to complement the shrinking process with a telescoping one, at some cost.
[15] A. G. Fowler, Quantum Inf. Comput. **11**, 867 (2011).
[16] C. H. Rycroft and M. Z. Bazant, online lecture notes: https://ocw.mit.edu/courses/mathematics/18-366-random-walks-and-diffusion-fall-2006/lecture-notes/lec01.pdf
[17] A. M. Frieze and M. Clarke, Eur. J. Oper. Res. **15**, 100 (1984).
[18] A. Kulik and H. Shachnai, Inf. Process. Lett. **110**, 707 (2010).