No-go theorem and a universal decomposition strategy for quantum channel compilation

Weiyuan Gong[1,*] Si Jiang[1,*] and Dong-Ling Deng[1,2,†]

[1]*Center for Quantum Information, IIIS, Tsinghua University, Beijing 100084, China*
[2]*Shanghai Qi Zhi Institute, 41st Floor, AI Tower, No. 701 Yunjin Road, Xuhui District, Shanghai 200232, China*

(Received 13 November 2021; accepted 21 December 2022; published 30 January 2023)

We rigorously prove a no-go theorem that, in sharp contrast to the case of compiling unitary gates, it is impossible to compile an arbitrary channel to arbitrary precision with any given finite elementary channel set, regardless of the length of the decomposition sequence. However, for a fixed error bound ϵ, we find a general and systematic strategy to compile arbitrary quantum channels. We construct a universal set with a constant number of ϵ-dependent elementary channels, such that an arbitrary quantum channel can be decomposed into a sequence of these elementary channels followed by a unitary gate, with the sequence length bounded by $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ in the worst case. We further optimize this approach by exploiting proximal policy optimization—a powerful deep reinforcement learning algorithm for the gate compilation. We numerically evaluate the performance of our algorithm concerning topological compiling of Majorana fermions, and we show that our algorithm can conveniently and effectively reduce the use of expensive elementary operations.

DOI: 10.1103/PhysRevResearch.5.013060

I. INTRODUCTION

Quantum compilers, which decompose quantum operations into hardware-compatible elementary operations, play an important role in quantum computation [1] and digital simulation [2]. This technique is especially crucial for the applications of noisy intermediate-scale quantum devices [3], where the performance of deep quantum circuits might be limited by noises and quantum decoherences. A number of notable approaches have been proposed to compile unitary gates and the dynamics of isolated quantum systems [4–16]. However, in reality quantum systems cannot be perfectly isolated and will inevitably interact with the external environment, making the more general quantum channel compiling indispensable for a wide range of applications [17,18]. Yet, quantum channel compilation has been barely explored [19], with major previous attention paid to exploiting the Stinespring dilation theorem [20] and compiling arbitrary quantum channels through elementary gates acting on an expanded Hilbert space with ancillary qubits playing a prerequisite role [21–30]. Hitherto, a general and systematic theorem for the universal compilation of arbitrary quantum channels has not been established. Here, we prove a generic no-go theorem regarding universal channel compiling to arbitrary accuracy. We introduce a universal decomposition strategy to approximate any channels within bounded error, and we optimize the strategy based on deep reinforcement learning (see Fig. 1 for an illustration). Machine learning, or more broadly artificial intelligence, has recently cracked a number of notoriously challenging problems, such as playing the game of Go [31,32], predicting protein spatial structures [33], and weather forecasting [34]. Its tools and techniques have been broadly exploited in various quantum physics tasks, including representing quantum many-body states [35,36], quantum state tomography [37,38], learning topological phases of matter [39–49], and nonlocality detection [50]. For quantum compiling on unitary gates, machine learning approaches have also been introduced to provide a near-optimal sequence [51,52]. In this paper, we first rigorously prove that it is *impossible* to compile any quantum channel to arbitrary accuracy using unitary gates and a finite set of elementary channels, which is in sharp contrast to the case of compiling a unitary gate. As illustrated in Fig. 1, we propose a quantum channel compiler that, given a bounded error demand ϵ, decomposes any quantum channel into a sequence of finite types of elementary quantum channels followed by a unitary gate. We provide a constructive method to obtain the elementary channel set, and we show that the size of the set scales as $O(d^2)$ with the dimension d of the Hilbert space and is independent of ϵ. We additionally prove that the length of the elementary channel sequence in the decomposition is bounded above by $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ in the worst case and $O(\min\{n, \log(\frac{1}{\epsilon})\}/\epsilon)$ on average, with n the number of qubits. For the unitary gate at the end of the decomposition, we optimize our strategy by training a deep-reinforcement-learning (DRL) agent to compile it using elementary gates. To reduce the resource requirement, we exploit the proximal policy optimization (PPO) algorithm [53–55] to train our agent with weighted cost in the reward function to reduce the use of experimentally expensive elementary gates. We further prove a $\Omega(\log(1/\epsilon))$ lower bound for any indispensable elementary gate count to compile an arbitrary unitary gate within error ϵ. As a benchmark,

[*] These authors contributed equally to this work.
[†] dldeng@tsinghua.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

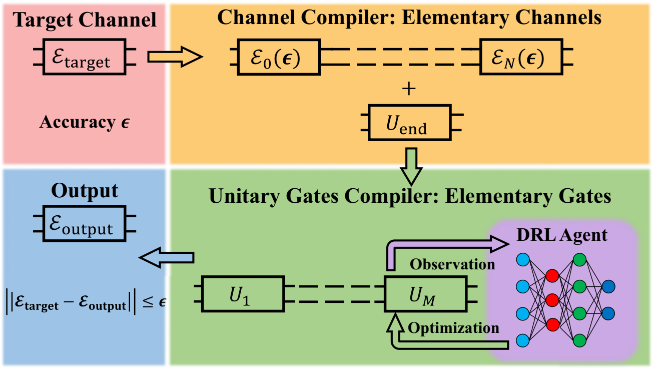
Target Channel
$\mathcal{E}_{\text{target}}$
Accuracy ϵ
Channel Compiler: Elementary Channels
$\mathcal{E}_0(\epsilon)$
$\mathcal{E}_N(\epsilon)$
+
U_{end}
Output
$\mathcal{E}_{\text{output}}$
$||\mathcal{E}_{\text{target}} - \mathcal{E}_{\text{output}}|| \leq \epsilon$
Unitary Gates Compiler: Elementary Gates
DRL Agent
Observation
U_1
U_M
Optimization

FIG. 1. A schematic illustration of the quantum channel compilation within bounded error. The compiler constructs a constant size elementary channel set using ϵ as parameters for each channel. It first decomposes the target channel into a sequence of elementary channels $\mathcal{E}_0(\epsilon), \ldots, \mathcal{E}_N(\epsilon)$ followed by a unitary gate. Then we use a unitary gate compiler to decompose the final unitary gate into elementary gates. We further optimize this strategy by employing a deep-reinforcement-learning (DRL) agent to produce approximation sequence $U_1, \ldots, U_M$. The DRL agent receives an observation from the current sequence and makes an action accordingly.

we apply our algorithm to randomly generated single-qubit channels realized by the topological quantum compiling of Majorana fermions [56–60], whose braidings together with a nontopological T gate form a universal set. We numerically show that our algorithm can reduce the use of a T gate by a factor of 2 compared to the traditional Solovay-Kitaev algorithm.

II. A NO-GO THEOREM

A quantum state can be represented by a positive-semidefinite operator $\rho \in \mathcal{O}(\mathcal{H}_S)$ with $\text{Tr}(\rho) = 1$, where $\mathcal{H}_S$ is the Hilbert space and $\mathcal{O}(\mathcal{H}_S)$ is the set of operators on $\mathcal{H}_S$. In general, a quantum channel $\mathcal{E}$ can be characterized by a completely positive, trace-preserving (CPTP) map which maps a quantum state ρ into another state $\mathcal{E}(\rho) \in \mathcal{O}(\mathcal{H}_S)$. Any single-qubit state can be represented as $\rho = \frac{1}{2}(I + \boldsymbol{a} \cdot \boldsymbol{\sigma})$, where $\boldsymbol{a}$ is a three-dimensional vector within the Bloch sphere, and $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ are Pauli matrices. Any linear CPTP map for a single-qubit system can be represented by a four-by-four matrix $\mathcal{T}$ [21,61–63]:

$$\mathcal{E} \rightarrow \mathcal{T} = \begin{pmatrix} 1 & 0 \\ t & T \end{pmatrix}, \quad \mathcal{T}_{ij} = \frac{1}{2}\text{Tr}[\sigma_i \mathcal{E}(\sigma_j)], \tag{1}$$

where $i, j \in \{0, 1, 2, 3\}$, $\sigma_0, \sigma_1, \sigma_2, \sigma_3$ represent the identity I and Pauli matrices $\sigma_x, \sigma_y, \sigma_z$, and $T \in \mathbb{R}^{3\times3}, t \in \mathbb{R}^3$. Under this representation, a channel is an affine map [64] $\mathcal{E} : \frac{1}{2}(1 + \boldsymbol{a} \cdot \boldsymbol{\sigma}) \rightarrow \frac{1}{2}(1 + \boldsymbol{a}' \cdot \boldsymbol{\sigma}), \boldsymbol{a}' = T\boldsymbol{a} + t$. Geometrically, $\mathcal{E}$ maps the states within the Bloch sphere into states enveloped by an ellipsoid, with t the center shift from the original center and T the distortion matrix for the ellipsoid. To make the quantum channel $\mathcal{E}(T, t)$ a CPTP map, T must be a semidefinite Hermitian matrix, and the ellipsoid should be completely within the Bloch sphere. When $\det(T) = 1$, the CPTP map reduces to a unitary gate. In this sense, unitary gates can be regarded as special channels. Throughout this paper, we differentiate unitary gates from channels for clarity.

Suppose we have a set of elementary channels and want to approximate the target channel with a sequence of unitary gates and elementary channels chosen from the set. For technical convenience and simplicity, we consider the superoperator trace norm [65] $||\mathcal{E}_{\text{target}} - \mathcal{E}_{\text{approx}}||_1 = \max_{\rho\in\mathcal{O}(\mathcal{H}_S)} ||\mathcal{E}_{\text{target}}(\rho) - \mathcal{E}_{\text{approx}}(\rho)||_1$ as the distance measure. Our results can be extended directly to other distance measures such as the Bures and the Hilbert-Schmidt metric.

Theorem 1. Given a finite set of elementary channels with arbitrary unitary gates, it is impossible to compile an arbitrary single-qubit channel to arbitrary accuracy.

Proof. We only sketch the major steps here and leave the details of the full proof to Appendix A. Suppose that we are provided with a finite set of elementary channels $\mathcal{C} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ with corresponding distortion matrices $\{T_1, \ldots, T_n\}$ and center shifts $\{t_1, \ldots, t_n\}$. Without loss of generality, we assume that $\det(T_n) \leqslant \cdots \leqslant \det(T_1) < 1$. Noting that the composition of channels cannot increase the determinant of the distortion matrix, thus a target channel with $\det(T_1) < \det(T)$ cannot be compiled by the channels chosen from $\mathcal{C}$ to arbitrary accuracy, independent of how long the decomposition sequence is.

The above theorem carries over to multiqubit channels. For a d-dimensional quantum state $\rho \in \mathcal{O}(\mathcal{H}_S)$, there exists a canonical and orthonormal basis $\{O_\alpha, \alpha = 1, \ldots, d^2 - 1\}$ [63,66]. A density operator ρ under such a basis reads $\rho = \frac{1}{d}(I + \sum_{\alpha=1}^{d^2-1} p_\alpha Q_\alpha)$, where $Q_\alpha = \sqrt{d(d-1)}O_\alpha$. The parameters in $\{p_\alpha\}$ form the polarization vector $\boldsymbol{p} = (p_1, \ldots, p_{d^2-1})$ of a $(d^2 - 1)$-dimensional ball with the Euclidean norm $||p||_2 = 1$ representing pure states and $||p||_2 < 1$ representing mixed states. As a quantum state ρ can be represented by a vector within a ball, a quantum channel $\mathcal{E} : \mathcal{O}(\mathcal{H}_S) \rightarrow \mathcal{O}(\mathcal{H}_S)$ can be written as an affine map represented by a distortion matrix $T \in \mathbb{R}^{(d^2-1)\times(d^2-1)}$ and a center shift $t \in \mathbb{R}^{d^2-1}$ similar to Eq. (1). With this representation, we can extend Theorem 1 to the multiqubit case.

The above results imply that a finite number of elementary channels cannot approximate an arbitrary target channel to arbitrary accuracy, regardless of the specific structure of each elementary channel and the length of the compiling sequence. This is in sharp contrast with the case of unitary gate compiling, where we can use a small number of elementary gates to compile an arbitrary unitary gate within any accuracy demand. We remark that any quantum channel can be implemented by a sequence of elementary unitary gates acting on an enlarged Hilbert space, and this seems to contradict the claim of Theorem 1. However, this spurious contradiction dissolves after noting the fact that tracing out the ancillary qubits at different sequence locations will effectively result in different channels even for the same elementary unitary gates. In other words, although a small number of different unitary gates suffice to implement any quantum channel with ancillary qubits, when restricted to the targeted system *no* finite set of elementary channels is universal.

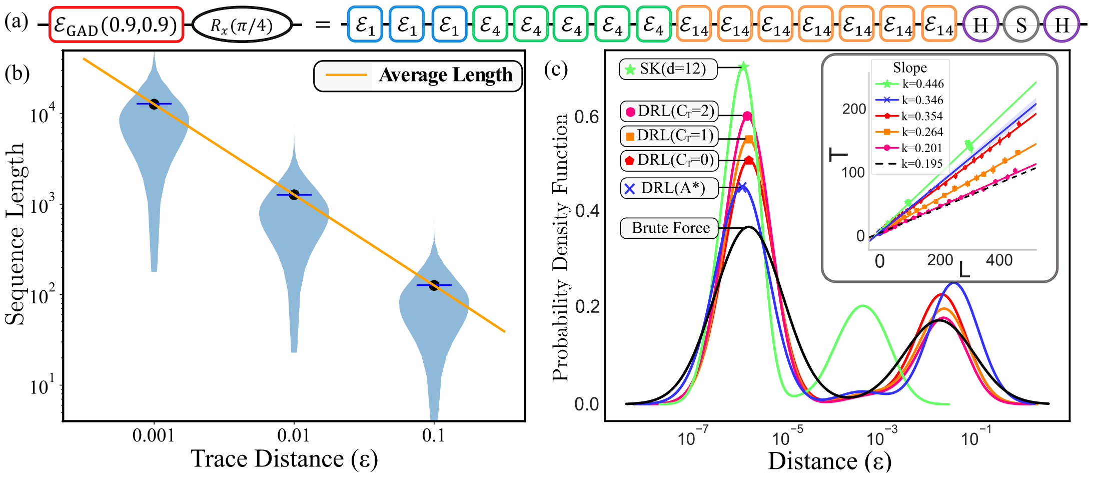

FIG. 2. (a) A decomposition of the channel $R_x(\frac{\pi}{4}) \circ \mathcal{E}_{\text{GAD}}(0.9, 0.9)$ within the error threshold $\epsilon = 0.05$, where $\mathcal{E}_{\text{GAD}}(\gamma, p)$ is the generalized amplitude damping (GAD) channel [1]. (b) The distribution of the length of output elementary channel sequences and the average value for randomly generated single-qubit channels. (c) Comparison of the distribution of the distance $\epsilon = d(U_t, U_{\text{approx}})$ between the target gate U_t and the approximation gate U_{approx} for our algorithm with different T cost $C_T = 0, 1, 2$, the Solovay-Kitaev (SK) algorithm with a net of depth 12, and the brute force algorithm with the sequence length not larger than 10. The inset shows the proportion of T gates in the compilation sequences. The black dashed line shows the proportion of T gates used in the brute force algorithm, which is considered as a numerical lower bound for the T gate ratio. The slope k and the error bars are calculated by fitting the scatter points over 1500 test samples.

III. A UNIVERSAL CHANNEL DECOMPOSITION STRATEGY

Although we have proven that compiling an arbitrary quantum channel within unbounded error using a finite set of elementary channels is impossible, it is still crucial to explore whether a finite channel set can realize universal compilation within a given error demand ϵ under experimental settings. To this end, we introduce and prove the following theorem:

Theorem 2. Consider compiling single-qubit channels with a given accuracy demand $\epsilon \ll 1$. We can construct a finite set of elementary quantum channels using ϵ as a parameter such that any single-qubit channel can be compiled by the elementary channels from this set and a unitary gate within error ϵ. The length of the sequence is bounded above by $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ in the worst case and $O(\frac{1}{\epsilon})$ on average.

Proof. We provide the main idea here. The full proof is technically involved and thus left to Appendix B. We decompose the compiling process into several steps and provide a constructive proof. For a target channel $\mathcal{E}$ with distortion matrix T with eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and center shift t, we first implement an intermediate channel $\mathcal{E}^*$ with parameters $T^* = \text{diag}(|\lambda_1|, |\lambda_2|, |\lambda_3|)$ using an elementary channel set containing 14 channels $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_{14}(T_{14}, t_{14})$ parametrized by $\delta = \frac{\epsilon}{7}$. The first eight channels have the distortion matrix $T_1 = \cdots = T_8 = \text{diag}\{1-\delta, 1-\delta, 1-\delta\}$, and each element of their center shift can be 0 or δ. The next four elementary channels have the distortion matrix $\text{diag}\{1, 1-\delta, 1-\delta\}$, and their center shifts have the first element 0 and other elements either 0 or δ. The last two channels have the distortion matrix $T_{13} = T_{14} = \text{diag}\{1, 1, 1-\delta\}$ and center shifts (0,0,0) and $(0, 0, \delta)$. We then use a unitary gate to realize the basis transformations and transfer $\mathcal{E}^*$ into the target $\mathcal{E}$.

One can extend the conclusion of Theorem 2 to the multiqubit case, where the size of the constructed elementary quantum channel set scales as $O(d^2)$. We provide two explicit constructions to decompose an arbitrary multiqubit quantum channel into a sequence of elementary channels followed by a unitary gate. The first construction has an elementary channel set of $O(d^2)$ size with a sequence length $O(d^2\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$, and the other uses a much larger elementary channel set [of size $O(2^{d^2})$] but a much shorter decomposition sequence [of length $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$]. We can also bound the average sequence length for the two constructions by $O(d^2\frac{1}{\epsilon})$ and $O(\min\{n, \log(\frac{1}{\epsilon})\}/\epsilon)$, respectively. In other words, we can decompose any target quantum channel into a *fixed sequence* of elementary channels followed by an n-qubit unitary gate. The channel compilation task has thus been reduced to unitary compiling with elementary unitary gates. We provide an example of decomposing a quantum channel within error $\epsilon = 0.05$ in Fig. 2(a). We remark that the elementary channels mentioned above can be physically implemented by exploiting an ancillary qubit. Compared to the traditional approaches that use ancillary qubits and elementary gates, the circuits to implement each elementary channel in our protocol are fixed for a given ϵ and hence might be more convenient for experimental implementation.

IV. NUMERICAL BENCHMARKS

Though the sequence of elementary channels for a given target channel is deterministic, we can optimize our strategy by promote the compiler for the desired unitary gate. We introduce a general approach (in the sense that it does not rely on the special properties of the elementary gate set and thus bears universal applicability). We exploit a reinforcement learning technique, the proximal policy optimization [53] in particular, to reduce the count of experimentally expensive gates. Compared with the DRL algorithm designed in [52], which applies the neural network to estimate the value function for each state and uses it as heuristics in the A^* search, PPO directly represents a policy explicitly as $\pi_\theta(a|s)$ by a neural network and searches the approximation sequence in a depth-first search scheme. This makes the PPO algorithm run with significantly less time and memory in both the training and searching stages, which is essential for compiling tasks for larger systems with more complicated actions and longer sequence lengths. The updating rules used by PPO explore the biggest possible improvement step without causing a performance collapse [53]. This property makes PPO particularly suitable for quantum compilation tasks since applying an inappropriate gate in the sequence will dramatically destruct the approximation gate.

Moreover, we modify the reward function used in PPO, which efficiently reduces the count of a specific gate that is experimentally costly, by setting $r_n = r(U_n, U_t) - C_{g^*}$, where r_n is the reward that the agent receives for the nth step, $r(U_n, U_t)$ is the term comparing the approximation gate U_n and the target gate U_t, and C_{g^*} is the additional weighted punishment for the employment of g^* gates. By increasing C_{g^*}, the agent tends to avoid using g^* gates and thus the g^* count is reduced in the decomposition. The setting and detailed training process of the PPO agent is provided in Appendixes D and E. In addition, we prove a lower bound $\Omega(d^2 \log(\frac{1}{\epsilon})/|G|)$ for the number of g^* in Appendix C, where G is the group generated by elementary gates except g^*. To benchmark the performance of our strategy, we carry out numerical experiments concerning decomposition of arbitrary single-qubit quantum channels including the analytical sequence of elementary channels and the DRL-based compilation of the desired unitary. For the channel sequence, we randomly generate channels of different distortion matrix T and center shift t and calculate the average sequence length. For the unitary part, we consider topological compiling with the four-quasiparticle encoding scheme for Majorana fermions. The details are left to Appendix F. It is well known that braidings of Majorana fermions only lead to an elementary gate set $S_E = \{\text{CNOT, H, S}\}$, which is not sufficient for universal quantum computation [57,67]. To achieve universality, a nontopological T gate with a relatively high experimental cost is necessary. A lot of effort has been put into reducing the use of T gates in unitary compiling [13–16,68–70], which either rely on the specific structure of the Clifford + T gate set or utilize ancillary qubits. In contrast, our PPO algorithm is generally applicable to any universal gate set without using ancillas.

Here, we exploit the average gate fidelity $F(U, V) = \int d|\psi\rangle |\langle\psi|UV^\dagger|\psi\rangle|^2$ provided in the open-source QuTiP package [71] to measure the distance $d(U, V) = 1 - F(U, V)$ between an approximated unitary gate V and a target unitary gate U. To exploit our DRL algorithm as a single-qubit quantum compiler, the action space is a set $A = \{B_{12}, B_{12}^{-1}, B_{23}, B_{23}^{-1}, T, T^{-1}\}$ containing six elementary gates. To train the agent, we employ a deep neural network (DNN) with five layers of fully connected neurons and train it with a PPO algorithm encapsulated in the OpenAI gym and baseline package [72–75]. We generate the training dataset of length L sequences consisting of the elementary gates to be the target gate and the test dataset of 1500 random sequences of gates from A of length between 10 and 80. We compare the performance of our algorithm, the traditional Solovay-Kitaev algorithm, and the DRL algorithm with A^* search [52] on such a dataset. As a baseline, we also use brute force to generate all sequences with length up to 10 and find the sequence with minimal approximation errors. We plot the sequence length distribution of elementary channels and the average value in Fig. 2(b). This result agrees very well with the $O(\frac{1}{\epsilon})$ analytical performance for our strategy.

In Fig. 2(c), we plot the error distribution over the test dataset and the proportion of the T gate. On average, the Solovay-Kitaev algorithm with net depth 12 provides a more accurate approximation than our algorithm at the price of a longer average length, while the DRL algorithm with A^* search performs a little worse than our algorithm because of limited memory and time used in searching. In the inset of Fig. 2(c), we present the ratio of T gates used in each algorithm. The dashed line is the proportion of T gates used in the brute force algorithm, which is considered as a numerical lower bound for the T gate ratio. We find that by increasing the cost of the T gate from $C_T = 0$ to 2, the average T gate rate over the test set decreases from 0.354 to 0.201, which is very close to the lower bound (0.195). This shows that adding the cost of the T gate as a punishment in the DRL algorithm can effectively reduce the T count, compared with the Solovay-Kitaev algorithm, which gives a T gate rate of 0.445, and the DRL algorithm with A^* search, which gives 0.346. We remark that the depth-first search scheme makes the time complexity scale linearly with the maximal search depth $L_{\max}$. This is distinct from the A^* search (which is breadth-first and hence exhibits an exponential scaling with $L_{\max}$) used in Ref. [52]. As a result, the PPO algorithm is significantly less time- and memory-consuming in both the training and searching stages. This advantage makes our algorithm feasible for compiling tasks for larger systems with more complicated actions and rewards. As a tradeoff, the PPO algorithm will not output the near-optimal sequence for a given target unitary and accuracy demand, and may even fail to find a decomposition if the accuracy threshold is too small. We mention that one can increase the successful rate of the compilation by increasing $L_{\max}$ in Appendix E.

V. DISCUSSION AND CONCLUSION

Theorem 1 implies that for channel compiling, there exists *no* finite universal elementary channel set. However, for a given target channel, this theorem does not tell whether it can be decomposed into predetermined elementary channels to a desired accuracy. Finding a general and efficiently computable criterion for determining whether a given channel can be compiled with a fixed elementary channel set or not is of both theoretical and experimental importance, and worth future investigation. Another interesting and important future direction is to incorporate a partial breadth-first search mechanism into the current PPO algorithm to increase the success rate and reduce the total length of the output sequences, at the cost of a slightly more time- and memory-consuming training process. In addition, our proposed PPO algorithm may carry over straightforwardly to other scenarios, including quantum control problems [76] and digital quantum simulations [77–81] for both closed and open systems.

In summary, we have found and rigorously proved a no-go theorem concerning compiling an arbitrary channel to arbitrary accuracy with any given finite elementary channel set. We provided a universal decomposition scheme to decompose arbitrary channels within a given error bound. We further introduced a DRL gate-compilation algorithm that is generally applicable and uses no ancillary qubit as an optimization for our strategy. Our results shed new light on the general problem of quantum compiling, which provides a valuable guide for future studies in both theory and experiment.

ACKNOWLEDGMENTS

We thank Zhengwei Liu, Weikang Li, Yuanhang Zhang, Qi Ye, Xun Gao, and Dong Yuan for helpful discussions. This work was supported by the start-up fund from Tsinghua University (Grant No. 53330300320), the National Natural Science Foundation of China (Grant No. 12075128), and the Shanghai Qi Zhi Institute.

APPENDIX A: PROOF FOR THEOREM 1

To formulate the problem, we consider a set S with metric $d(\cdot)$. A set $\Gamma \subset S$ is called a δ-net if for any $x \in S$ there exists $y \in \Gamma$ such that $d(x, y) \leqslant \delta$ [5]. The subset $\Gamma \subset S$ is called a dense subset under metric $d(\cdot)$ if it is a δ-net of S for arbitrary δ.

We recall Eq. (1) in the main text, which provides a matrix-form mapping representation for a linear CPTP quantum channel $\mathcal{E}$ as

$$\mathcal{E} \to \mathcal{T} = \begin{pmatrix} 1 & 0 \\ t & T \end{pmatrix} : \frac{1 + \boldsymbol{a} \cdot \boldsymbol{\sigma}}{2} \to \frac{1 + \boldsymbol{a}' \cdot \boldsymbol{\sigma}}{2}, \qquad \text{(A1)}$$

where T is called a distortion matrix, t is called a center shift, and the state vectors $\boldsymbol{a}'$ and $\boldsymbol{a}$ are chosen within the Bloch sphere satisfying the relation $\boldsymbol{a}' = T\boldsymbol{a} + t$. This representation indicates that the channel $\mathcal{E}$ maps the Bloch sphere into an ellipsoid. To guarantee the physical feasibility, the ellipsoid must be enveloped within the original Bloch sphere and $|a|^2, |a'|^2 \leqslant 1$. Therefore, T cannot have an eigenvalue that has magnitude larger than 1. Moreover, if all eigenvalues of T have magnitude 1, then $t = 0$ and $\mathcal{E}$ is a unitary gate. For simplicity, we represent a quantum channel $\mathcal{E}$ with distortion matrix T and center shift t as $\mathcal{E}(T, t)$.

As mentioned in the main text, the distance measure between channels used in this paper is the superoperator trace norm [65], which measures the maximal 1-norm distance between the output states of different quantum channels with the same quantum state ρ chosen from $\mathcal{H}_S$ as the input state. For single-qubit states $\rho_1 = (1 + \boldsymbol{a}_1 \cdot \boldsymbol{\sigma})/2$ and $\rho_2 = (1 + \boldsymbol{a}_2 \cdot \boldsymbol{\sigma})/2$, the trace distance between them reads $||\rho_1 - \rho_2||_1 = |\boldsymbol{a}_1 - \boldsymbol{a}_2|/2$. This indicates that the distance between channels $\mathcal{E}_1(T_1, t_1)$ and $\mathcal{E}_2(T_2, t_2)$ is D if $\max_{\rho=(1+\boldsymbol{a}\cdot\boldsymbol{\sigma})/2\in\mathcal{H}_S} |(T_1 - T_2)\boldsymbol{a} + t_1 - t_2| = 2D$.

Now we start the proof for Theorem 1. We first introduce the following lemma:

Lemma 1. Suppose we have two single-qubit quantum channels $\mathcal{E}_1(T_1, t_1)$ and $\mathcal{E}_2(T_2, t_2)$, and $|\det(T_1)| > |\det(T_2)|$. If $|\det(T_1)| - |\det(T_2)| > 6\epsilon$, where ϵ is a constant strictly smaller than the magnitude of any eigenvalue of T_1, then $||\mathcal{E}_1 - \mathcal{E}_2||_{1\to1} > \epsilon$.

Proof. $\mathcal{E}_1(T_1, t_1)$ and $\mathcal{E}_2(T_2, t_2)$ in Eq. (A1) map the Bloch sphere into two ellipsoids. Consider quantum channels $\mathcal{E}_1(T_1, t_1)$, $\mathcal{E}_2'(T_2, t_1)$, and $\mathcal{E}_2(T_2, t_2)$, and suppose $\rho^* = \frac{1+\boldsymbol{a}\cdot\boldsymbol{\sigma}}{2} = \arg\max_{\rho\in\mathcal{H}_S} ||\mathcal{E}_1(\rho) - \mathcal{E}_2'(\rho)||_1$. Notice that $\mathcal{E}_1$ and $\mathcal{E}_2'$ have the same center shift t_1. We can find another quantum state that also yields the maximal output state distance as $\rho^*_{\text{opp}} = \frac{1-\boldsymbol{a}\cdot\boldsymbol{\sigma}}{2} = \arg\max_{\rho\in\mathcal{H}_S} ||\mathcal{E}_1(\rho) - \mathcal{E}_2'(\rho)||_1$ according to the symmetry property of the ellipsoid. Since the ellipsoid derived by $\mathcal{E}_2$ can be regarded as the ellipsoid derived by $\mathcal{E}_2'$ with an additional center shift $t_2 - t_1$, at least one of the distances $||\mathcal{E}_1(\rho^*) - \mathcal{E}_2(\rho^*)||_1$ and $||\mathcal{E}_1(\rho^*_{\text{opp}}) - \mathcal{E}_2(\rho^*_{\text{opp}})||_1$ is not smaller than $||\mathcal{E}_1(\rho^*) - \mathcal{E}_2'(\rho^*)||_1$. Therefore, we conclude that $||\mathcal{E}_1 - \mathcal{E}_2||_{1\to1} \geqslant ||\mathcal{E}_1 - \mathcal{E}_2'||_{1\to1}$.

Now we prove that if $|\det(T_1)| - |\det(T_2)| > 6\epsilon$, then $||\mathcal{E}_1 - \mathcal{E}_2||_{1\to1} \geqslant ||\mathcal{E}_1 - \mathcal{E}_2'||_{1\to1} > \epsilon$. We denote the eigenvalues of T_1 as λ_1, λ_2, and λ_3 with $\lambda_1\lambda_2\lambda_3 = \det(T_1)$, and the eigenvalues of T_2 as λ_1', λ_2', and λ_3' with $\lambda_1'\lambda_2'\lambda_3' = \det(T_2)$. If $||\mathcal{E}_1 - \mathcal{E}_2'|| \leqslant \epsilon$, the ellipsoid produced by $\mathcal{E}_2'$ will envelope all quantum states that are at least 2ϵ within the surface of the ellipsoid produced by $\mathcal{E}_1$. Therefore, the ellipsoid with a semimajor axis of length $|\lambda_1| - 2\epsilon$, $|\lambda_2| - 2\epsilon$, $|\lambda_3| - 2\epsilon$ and the same direction with the ellipsoid produced by map $\mathcal{E}_1$ should fall completely within the ellipsoid produced by $\mathcal{E}_2'$. That is to say, $|\det(T_2)| = |\lambda_1'||\lambda_2'||\lambda_3'| \geqslant (|\lambda_1| - 2\epsilon)(|\lambda_2| - 2\epsilon)(|\lambda_3| - 2\epsilon)$. As we mentioned before, T_1 and T_2 cannot contain eigenvalues with absolute values larger than 1, thus $|\det(T_2)| \geqslant |\det(T_1)| - 6\epsilon$. This completes the proof for the lemma.

Then we prove Theorem 1 based on the above lemma. Suppose we have a finite number N of elementary channels $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_N(T_N, t_N)$ and arbitrary unitary gates in the channel compiler, and the distortion matrix T_i has a determinant of absolute value $d_i = |\det(T_i)|$. Without loss of generality, we further assume that $1 > d_1 \geqslant d_2 \geqslant \cdots \geqslant d_N \geqslant 0$. Therefore, any channel $\mathcal{E}(T, t)$ that can be represented by a sequence of elementary channels and unitary gates satisfies $\det(T) \leqslant d_1$, or $\det(T) = 1$, when the sequence only consists of unitary gates.

Considering the compilation of a target channel $\mathcal{E}^*(T^*, t^*)$ with $\det(T^*) = \frac{1+d_1}{2}$ and accuracy demand $\epsilon < \frac{1-d_1}{12}$, any decomposition sequence, whose corresponding generated channel is denoted as $\mathcal{E}(T, t)$, satisfies either $\det(T) \leqslant d_1 < \det(T^*) - 6\epsilon$ or $\det(T) = 1 > \det(T^*) + 6\epsilon$. Therefore, according to Lemma 1 this target channel cannot be decomposed into elementary channels and unitary gates under such accuracy demand ϵ. This completes the proof for Theorem 1 in the main text.

APPENDIX B: PROOF FOR THEOREM 2

In this Appendix, we provide the detailed proof for Theorem 2 in the main text. We assume that the target channel is $\mathcal{E}^*(T^*, t^*)$, and T^* has eigenvalues λ_1^*, λ_2^*, and λ_3^* with $|\lambda_1^*| \geqslant |\lambda_2^*| \geqslant |\lambda_3^*|$ and orthonormal eigenvectors v_1^*, v_2^*, and v_3^*. We denote $t^* = (t_1^*, t_2^*, t_3^*)$ under basis $\{v_1^*, v_2^*, v_3^*\}$. Without loss of generality, we assume $t_i^* > 0$ and suppose the accuracy demand is ϵ. Now, we propose a two-step procedure

TABLE I. A table illustration for our sequence of length k_3 to approximate the channel $\mathcal{E}_{\text{step 1}}$ within distance 6δ consisting of elementary channels in Eqs. (B1a)–(B1f). The first row shows the position of the elementary channel in the decomposition sequence (abbreviated as Seq. Pos.). For the jth elementary channel in this sequence, it has a distortion matrix $T_j = \text{diag}\{T_j^{11}, T_j^{22}, T_j^{33}\}$ and a center shift $t_j = (s_{x,j}\delta, s_{y,j}\delta, s_{z,j}\delta)$ for $j = 1, \ldots, k_1$, $t_j = (0, s_{y,j}\delta, s_{z,j}\delta)$ for $j = k_1 + 1, \ldots, k_2$, and $t_j = (0, 0, s_{z,j}\delta)$ for $j = k_2 + 1, \ldots, k_3$.

Seq. Pos.	1	$\cdots$	k_1	k_1+1	$\cdots$	k_2	k_2+1	$\cdots$	k_3
T_j^{11}	$1-\delta$	$\cdots$	$1-\delta$	1	$\cdots$	1	1	$\cdots$	1
s_x	$s_{x,1}$	$\cdots$	s_{x,k_1}						
T_j^{22}	$1-\delta$	$\cdots$	$1-\delta$	$1-\delta$	$\cdots$	$1-\delta$	1	$\cdots$	1
s_y	$s_{y,1}$	$\cdots$	s_{y,k_1}	s_{y,k_1+1}	$\cdots$	s_{y,k_2}			
T_j^{33}	$1-\delta$	$\cdots$	$1-\delta$	$1-\delta$	$\cdots$	$1-\delta$	$1-\delta$	$\cdots$	$1-\delta$
s_z	$s_{z,1}$	$\cdots$	s_{z,k_1}	s_{z,k_1+1}	$\cdots$	s_{z,k_2}	s_{z,k_2+1}	$\cdots$	s_{z,k_3}

to decompose the target channel into a sequence of unitary gates and channels chosen from 14 elementary channels.

Step 1. We consider realizing a channel $\mathcal{E}_{\text{step 1}}(T_{\text{step 1}}, t_{\text{step 1}})$, where $T_{\text{step 1}} = \text{diag}\{|\lambda_1^*|, |\lambda_2^*|, |\lambda_3^*|\}$ and $t_{\text{step 1}} = (t_1^*, t_2^*, t_3^*)$. We construct the following 14 elementary channels $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_{14}(T_{14}, t_{14})$ using a parameter $\delta(\epsilon)$ to be fixed later:

$$T_1 = \cdots = T_8 = \text{diag}\{1-\delta, 1-\delta, 1-\delta\}, \tag{B1a}$$

$$T_9 = \cdots = T_{12} = \text{diag}\{1, 1-\delta, 1-\delta\}, \tag{B1b}$$

$$T_{13} = T_{14} = \text{diag}\{1, 1, 1-\delta\}, \tag{B1c}$$

$$\begin{aligned}
t_1 &= t_9 = t_{13} = \mathbf{0}, \\
t_8 &= (\delta, \delta, \delta)^T, \\
t_2 &= (\delta, 0, 0)^T, &\text{(B1d)} \\
t_3 &= t_{10} = (0, \delta, 0)^T, \\
t_4 &= t_{11} = t_{14} = (0, 0, \delta)^T, &\text{(B1e)} \\
t_5 &= (\delta, \delta, 0)^T, \\
t_6 &= (\delta, 0, \delta)^T, \\
t_7 &= t_{12} = (0, \delta, \delta)^T. &\text{(B1f)}
\end{aligned}$$

Denoting $k_i = \lceil \min\{\log_{(1-\delta)} |\lambda_i^*|, \log_{(1-\delta)} \delta\} \rceil$, $i = 1, 2, 3$, we have $||\lambda_i^*| - (1-\delta)^{k_i}| < \delta$, $i = 1, 2, 3$. We introduce a procedure to use the above elementary channels to compile $T_{\text{step 1}}$ within distance 6δ using a sequence of k_3 elementary channels. Table I contains a record of each elementary channel in this sequence. In this sequence, we introduce three $\{0, 1\}$ strings $s_x = s_{x,1} \cdots s_{x,k_1}$, $s_y = s_{y,1} \cdots s_{y,k_2}$, and $s_z = s_{z,1} \cdots s_{z,k_3}$. For $s_{i,j}$ $(i = x, y, z)$, it is 0 or 1 when the center shift of the jth channel in the sequence on the i-axis is 0 or δ. We use T_j^{ii} $(i = 1, 2, 3, j = 1, \ldots, k_3)$ to represent the (i, i)th entry of the distortion matrix T_j for the jth channel in the sequence.

The sequence in the above table composes a channel $\mathcal{E}'(T', t')$, where $T' = \text{diag}\{(1-\delta)^{k_1}, (1-\delta)^{k_2}, (1-\delta)^{k_3}\}$ and $t' = (t_1', t_2', t_3')$ with each element

$$t_1' = \delta \sum_{j=1}^{k_1} s_{x,j}(1-\delta)^{j-1}, \tag{B2a}$$

$$t_2' = \delta \sum_{j=1}^{k_2} s_{y,j}(1-\delta)^{j-1}, \tag{B2b}$$

$$t_3' = \delta \sum_{j=1}^{k_3} s_{z,j}(1-\delta)^{j-1}. \tag{B2c}$$

From Eqs. (B2a)–(B2c), as $||\lambda_i^*| - (1-\delta)^{k_i}| < \delta, i = 1, 2, 3,$ we can observe that by changing $\{0, 1\}$ strings s_x, s_y, s_z, t_1', t_2', t_3' forms a δ-net of interval $[0, 1 - |\lambda_1^*|]$, $[0, 1 - |\lambda_2^*|]$, and $[0, 1 - |\lambda_3^*|]$, which include all possible value of t_1^*, t_2^*, t_3^* because the output ellipsoid of the quantum channel should be within the original Bloch sphere. Given a center shift element $t_i^* \in [0, 1 - |\lambda_i^*|]$, we can calculate s_i by extending t_i^* into the summation over a series $\delta \sum_{j=1}^{k_1} s_{i,j}(1-\delta)^{j-1}$ on $(1-\delta)$. Therefore, for an arbitrary $\mathcal{E}_{\text{step 1}}$, we can find strings s_x, s_y, and s_z such that each entry of T' and t' differs from $T_{\text{step 1}}$ and $t_{\text{step 1}}$ by δ at most. Hence, the total distance is strictly smaller than the sum of the distance, which is 6δ.

Step 2. In this step we apply a unitary transformation to transfer the current basis to the $\{v_1^*, v_2^*, v_3^*\}$ basis.

In the above two steps, the unitary gates in steps 2–4 can be combined as one unitary gate, and according to Ref. [1] this unitary gate can be approximated within error δ by a sequence of elementary gates chosen from a universal gate set. Therefore, the total error cannot exceed the sum of error in all steps, which is 7δ. By fixing $\delta = \frac{\epsilon}{7}$, we can decompose an arbitrary quantum channel into a sequence of unitary gates and elementary channels chosen from the 14 elementary channels $\mathcal{E}_1, \ldots, \mathcal{E}_{14}$ constructed in Eqs. (B1a)–(B1f).

Following the two steps above, we can also bound the length of the sequence for the compilation. In steps 2–4 we need one unitary gate, while in step 1 the length of the table does not exceed $\log_{(1-\delta)} \delta + 1$. In practice, $\delta = \frac{\epsilon}{7}$ is usually a small number close to 0. Therefore, we can do the approximation $\log_{(1-\delta)} \delta = \frac{\ln(\delta)}{\ln(1-\delta)} \approx \frac{1}{\delta} \ln(\frac{1}{\delta}) = O(\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))$. This indicates that the length of the entire sequence is $O(\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))$. This completes the proof of Theorem 2 in the main text.

We can extend Theorem 2 to the multiqubit case. As mentioned in the main text, for a d-dimensional quantum state $\rho \in \mathcal{O}(\mathcal{H}_S)$, a canonical and orthonormal basis [63,66] $\{O_\alpha\}$, $O_\alpha \in \mathcal{O}(\mathcal{H}_S)$ satisfies (i) $O_0 = I$, (ii) $\text{tr}(O_\alpha) = 0, \forall \alpha \neq 0$, and (iii) $\text{tr}(O_\alpha^\dagger O_\beta) = \delta_{\alpha\beta}$. An arbitrary density operator ρ can be written as $\rho = \frac{1}{d}(I + \sum_{\alpha=1}^{d^2-1} p_\alpha Q_\alpha)$, where $Q_\alpha = \sqrt{d(d-1)} O_\alpha$. The parameters in $\{p_\alpha\}$ form the polarization vector $\boldsymbol{p} = (p_1, \ldots, p_{d^2-1})$ of a $(d^2 - 1)$-dimensional ball with the Euclidean norm $||p||_2 = 1$ representing pure states and the Euclidean norm $||p||_2 < 1$ representing mixed states. Since a quantum state ρ can be represented as a vector within a ball, a quantum channel $\mathcal{E} : \mathcal{O}(\mathcal{H}_S) \to \mathcal{O}(\mathcal{H}_S)$ can be written as an affine map represented by a distortion matrix $T \in \mathbb{R}^{(d^2-1)\times(d^2-1)}$ and a center shift $t \in \mathbb{R}^{d^2-1}$,

$$\mathcal{E} \to \mathcal{T} = \begin{pmatrix} 1 & 0 \\ t & T \end{pmatrix},\ \mathcal{T}_{ij} = \frac{1}{d}\mathrm{tr}[Q_\alpha \mathcal{E}(Q_\beta)], \tag{B3a}$$

$$\boldsymbol{p} \to T\boldsymbol{p} + t. \tag{B3b}$$

Notice that the above affine map is similar to the single-qubit case mentioned in the main text, and we can extend Theorem 2 straightforwardly to the multiqubit scenario.

We assume the target channel to be $\mathcal{E}^*(T^*, t^*)$, and T^* has eigenvalues $\lambda_1^*, \ldots, \lambda_{d^2-1^*}$. Without loss of generality, we can rank λ_i^* in decreasing order of magnitude such that $|\lambda_1^*| \geqslant \cdots \geqslant |\lambda_{d^2-1}^*|$. Inherited from the proof for the single-qubit channel, k_i is defined to be $k_i = \lceil \min\{\log_{(1-\delta)} |\lambda_i^*|, \log_{(1-\delta)} \delta\} \rceil, \qquad i = 1, \ldots, d^2-1$. If we directly use the constructive approach in the proof for Theorem 1, we will create $2^{d^2} - 2$ elementary channels $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_{2^{d^2}-2}(T_{2^{d^2}-2}, t_{2^{d^2}-2})$. The first 2^{d^2-1} channels $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_{2^{d^2-1}}(T_{2^{d^2-1}}, t_{2^{d^2-1}})$ have the same distortion matrix $T_1 = \cdots = T_{2^{d^2-1}} = \mathrm{diag}\{1-\delta, \ldots, 1-\delta\}$, and their center shifts go through 2^{d^2-1} cases in which each element of the center shift can be 0 or δ. The next 2^{d^2-2} elementary channels that follow are $\mathcal{E}_{2^{d^2-1}+1}(T_{2^{d^2-1}+1}, t_{2^{d^2-1}+1}), \ldots, \mathcal{E}_{2^{d^2-1}+2^{d^2-2}}(T_{2^{d^2-1}+2^{d^2-2}}, t_{2^{d^2-1}+2^{d^2-2}})$. They share the same distortion matrix $\mathrm{diag}\{1, 1-\delta, \ldots, 1-\delta\}$, and their center shifts go through all cases that have the first element as 0 and other elements as either 0 or δ. The remaining channels are constructed similarly until the last two channels $\mathcal{E}_{2^{d^2}-3}(T_{2^{d^2}-3}, t_{2^{d^2}-3})$, $\mathcal{E}_{2^{d^2}-2}(T_{2^{d^2}-2}, t_{2^{d^2}-2})$, which have the distortion matrix $\mathrm{diag}\{1, \ldots, 1, 1-\delta\}$ and center shifts $(0, \ldots, 0)$ and $(0, 0, \ldots, 0, \delta)$. Similar to the previous proof, we hold strings s_i, $i = 1, \ldots, d^2-1$ with the jth element $s_{i,j} = 0, 1$ representing whether the ith element of the center shift for the jth channel of the sequence is 0 or δ.

Under this construction, the error in total can be bounded above by $[2(d^2-1)+1]\delta$. Therefore, we fix $\delta = \frac{\epsilon}{2(d^2)-1}$ to guarantee that the distance between our approximation and target channel is no more than ϵ. The length of the sequence is still bounded by $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$.

However, notice that the above construction requires $O(2^{d^2})$ elementary channels, which is double exponential to the number of qubits n. Here, we propose another construction that only requires $O(d^2)$ elementary channels. We still exploit the four-step compiling process in the previous section. Steps 2–4 remain the same as the previous construction to implement complex, negative eigenvalues in the distortion matrix and perform orthonormal basis transformation. In step 1, we simply use $\mathcal{E}_1(T_1, t_1), \ldots, \mathcal{E}_{2(d^2-1)}(T_{2(d^2-1)}, t_{2(d^2-1)})$ with $T_{2i-1} = T_{2i} = \mathrm{diag}\{1, \ldots, 1, 1-\delta, 1, \ldots, 1\}$, where the ith diagonal element is $1-\delta$, $t_{2i-1} = (0, \ldots, 0)$, and $t_{2i} = (0, \ldots, 0, \delta, 0, \ldots, 0)$, with the ith element being δ.

Under this construction, step 1 can be decomposed into substeps compiling $\mathcal{E}_{\text{step i}}(T_{\text{step i}}, t_{\text{step i}})$ with $T_{\text{step i}} = \mathrm{diag}\{1, \ldots, 1, |\lambda_i^*|, 1, \ldots, 1\}$, $t_{\text{step i}} = (0, \ldots, 0, t_i^*, 0, \ldots, 0)$ using $\mathcal{E}_{2i-1}, \mathcal{E}_{2i}$ separately, where $|\lambda_i^*|$ and t_i are the ith diagonal element of the distortion matrix and the ith element of the center shift. In the ith substep, we keep a k_i length 0-1 string s_i and decompose $\mathcal{E}_{\text{step i}}$ into a sequence containing k_i elementary channels. If the jth element $s_{i,j}$ of s_i is 0, we add $\mathcal{E}_{2i-1}$ to the sequence and otherwise we add $\mathcal{E}_{2i}$. Therefore, the ith element of approximation for the center shift is $\sum_{j=1}^{k_i} s_{i,j}\delta(1-\delta)^{j-1}$, which forms a δ-net in the range $[0, 1-|\lambda_i^*|]$ as $||\lambda_i^*| - (1-\delta)^{k_i}| < \delta, \quad i = 1, \ldots, d^2-1$. Therefore, in each substep we can approximate $|\lambda_i^*|$ and t_i^* within distance δ.

The total distance in this step cannot exceed $2(d^2-1)\delta$, which is the same as the previous construction. We can still fix $\delta = \frac{\epsilon}{2(d^2)-1}$. It is worthwhile to mention that there exists a tradeoff between the above two constructions. Though the second construction only requires $O(d^2)$ elementary channels, the output sequence will have a total length $O(d^2\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$, which is exponential in the number of qubits n.

To bound the average performance, we assume that the eigenvalues of the distortion matrix are i.i.d random variables distributed in [0,1]. The minimal eigenvalue has the expectation value $\mathbb{E}[\min_i |\lambda_i|] = \frac{1}{d^2}$ and $\mathbb{E}[\prod_i |\lambda_i|] = \frac{1}{2^{d^2-1}}$. Notice that the length of the sequence is $\log_{(1-\delta)} \min_i |\lambda_i|$ in the first construction and $\sum_i \log_{(1-\delta)} |\lambda_i| = \log_{(1-\delta)}\{\prod_i |\lambda_i|\}$ in the second construction. By the concavity of $\log_a(x)$, $0 < a < 1$, the expectation sequence length is $O(\frac{\min\{n, \log(\frac{1}{\epsilon})\}}{\epsilon})$ in the first construction and $O(\frac{d^2}{\epsilon})$ for the second universal set.

APPENDIX C: LOWER BOUND FOR AN INDISPENSABLE GATE IN A UNITARY COMPILATION

We now consider quantum compilation for unitary gates $U \in SU(d)$ with the elementary gate set $S_E = \{g_1, \ldots, g_n | g_i \in SU(d)\}$. A gate set is universal if it can compile arbitrary unitary gates to any given accuracy demand under the distance measure. In other words, a gate set is universal if and only if it generates a dense subgroup in $SU(d)$ [5]. We present the following theorem concerning the lower bound of any indispensable gate in compiling an arbitrary unitary.

Theorem S1. For a nondense subgroup $G \subset SU(d)$ generated by S_E, suppose we can find $g^* \in SU(d)$ such that G' generated by $\{g_1, \ldots, g_n, g^*\}$ is dense in $SU(d)$. When employing G' as an elementary gate set for a quantum compilation task on $SU(d)$, the number of gates g^* to compile an arbitrary gate within distance ϵ is bounded below by

$$N^* = \Omega\left(\frac{d^2-1}{\log(|G|)}\log\left(\frac{1}{\epsilon}\right)\right). \tag{C1}$$

To derive a lower bound, we exploit the volume method [5,82] based on the constraint that the whole space of $SU(d)$ should be covered by the ϵ-balls centered by the gates that can be implemented by an elementary gate sequence. The detailed proof is given below:

Proof. We start with the case when the subgroup $G = \langle g_1, \ldots, g_n \rangle$ generated by $\{g_1, \ldots, g_n\}$ is finite. We denote the size of G as $|G| = K$. Consider a compilation with fewer than t g^* gates and an unlimited number of gates chosen from G. If no g^* gate is used, we can only compile the K gates in G. When we use at least one g^* gate, any sequence containing $0 < k \leqslant t$ g^* gates can always be written as $g = (g_{i_{11}} g^* g_{i_{12}}) \cdots (g_{i_{k1}} g^* g_{i_{k2}})$, where $g_{i_{11}}, g_{i_{12}}, \ldots, g_{i_{k1}}, g_{i_{k2}}$ are chosen from G. Consider the subset $G^* = \{g_s g^* g_t | g_s, g_t \in$

$G\}$ that can generate a dense subgroup of $SU(d)$. Any sequence that contains k g^* gates can be regarded as k gates in G^*. Therefore, the number of g^* gates can be regard as the number of gates from G^*. As the size of G^* is $|G^*| = K^2$, the possible number of gate sequences containing no more than t g^* can be bounded above by $|G^*| + |G^*|^2 + \cdots + |G^*|^t \leqslant O(K^{2t+2})$. Therefore, the total number of gates we can accurately compile with a gate sequence with fewer than t g^* gates is bounded above by $O(K^{2t+2})$.

We exploit the normalized Haar measure on $SU(d)$ space such that the volume of $SU(d)$ is 1 [82]. Under this measure, the volume of ϵ-ball in the $SU(d)$ group scales as $\Theta(\epsilon^{d^2-1})$. If any gate in $SU(d)$ can be approximated within distance ϵ, all ϵ-balls centered by possible gate sequences generated by no more that t g^* gates should cover $SU(d)$. Hence, we can deduce that $k \geqslant \frac{d^2-1}{\log|G|}\log(\frac{1}{\epsilon})$. This completes the proof of Theorem 2 in the main text for a finite G.

When the subgroup G is infinite, the lower bound given by the volume method will reduce to $\Omega(1)$. This lower bound, however, is not as trivial as it seems. Indeed, we can even find a simple extreme example, in which one can compile arbitrary gates with only two g^* gates and an unlimited number of gates chosen from an infinite G. Consider a single-qubit gate compilation with $g^* = H$ and $G = \langle R_x(\alpha\pi)\rangle$ with a generator $R_x(\alpha\pi)$ of an irrational number α. In this example, G is an infinite group that can approximate all the rotations along the x-axis within arbitrary accuracy demands. Notice that an arbitrary single-qubit gate can be written as $R_x(\theta_1)R_z(\theta_2)R_x(\theta_3)$ [1] and $R_z(\theta_2) = HR_x(\theta_2)H$, and any single-qubit gate can be compiled by gates chosen from G and at most two H gates.

This theorem gives a lower bound for the count of any indispensable gate g^* in compiling an arbitrary unitary, which scales linearly in $\log(\frac{1}{\epsilon})$ but is quadratic in the Hilbert space dimension d that is exponentially large as the system size increases. In practical applications, g^* may represent some experimentally expensive or flawed gate, thus reducing its count in compiling can be crucial. For the case of quantum compiling with a Clifford+T gate set, a number of striking algorithms [12–16], which either exploit its specific structures or utilize ancillary qubits, have been proposed to reduce the T count. In the main text, we focus on the quantum compiling on Majorana fermions and nontopological T gates. We note that the topological gate set for n qubits generates a Clifford group $\mathrm{Cl}_n \subset SU(d)$ with $|\mathrm{Cl}_n| = O(2^{2n^2+3n})$, $d = 2^n$ [83–85]. The worst-case T count for compiling an arbitrary gate is $\Omega((d^2-1)/n^2\log(1/\epsilon))$. We mention that topological quantum compiling has been broadly explored with various algorithms proposed [86–92]. Most of the algorithms run in $O(\mathrm{poly}\log(1/\epsilon))$ and output sequences of braidings with length $O(\mathrm{poly}\log(1/\epsilon))$ to obtain an approximation within distance ϵ from a given target evolution. Based on this compilation scheme, various algorithms have been proposed to reduce the T count for both multiqubit unitary compilation [16,68] and single-qubit unitary compilation [69,70]. However, these algorithms focus on providing an optimal number of T gates for a given unitary. Compared with these approaches, our result provide a lower bound for the worst-case overall unitaries chosen from $SU(d)$.

APPENDIX D: DEEP REINFORCEMENT LEARNING AND PPO ALGORITHM

In this Appendix, we give a brief introduction to the deep reinforcement learning (DRL) and the PPO algorithm we exploit to compile unitary gates.

1. Introduction to reinforcement learning and policy gradient

To formalize the learning process, we introduce some basic concepts and notations. A reinforcement learning (RL) aims to train a decision-making agent in a Markovian decision process. This process involves a state set S and an action set A. In step n, the agent chooses an action $a_n \in A$ while the environment shifts from $s_n \in S$ to $s_{n+1} \in S$, providing the agent a scalar value reward r_n as feedback. In a Markovian decision process, the new state s_{n+1} and reward r_n only depend on the former state s_n and action a_n. Therefore, the RL process can be written as a trajectory $\tau : s_0 \to a_0 \to r_0 \to s_1 \to \cdots \to a_{N-1} \to r_{N-1} \to s_N$, where N is the maximal number of steps in an iteration.

The core problem in RL is to learn a policy to choose the optimal action a_n given the environment state s_n. Therefore, a policy function π is introduced to map the states to the action probability distributions. The agent uses this function π to perform decision-making tasks and choose the action $a \sim \pi(s)$ according to the policy for the state s. In deep reinforcement learning, a policy π_θ is represented by a policy network with parameters θ. To evaluate the performance of π_θ, an objective function is defined to be the expected return over all complete trajectories:

$$J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{n=1}^{N}\gamma^n r_n\right], \tag{D1}$$

where γ is the discount factor, and the expectation is obtained over all trajectories sampled from π_θ.

To find an optimal policy, we have to find the policy that maximizes the objective function. A straightforward approach is to perform gradient descent on the policy to solve the optimization,

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta), \tag{D2}$$

where α is a scalar factor known as the learning rate, and $\nabla_\theta J(\pi_\theta)$ is known as the *policy gradient*. It is proved that the policy gradient can be calculated by [54]

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{n=0}^{N} R_n(\tau)\nabla_\theta \log\pi_\theta(a_n|s_n)\right], \tag{D3}$$

where the action $a_n \sim \pi_\theta$ is sampled from probability distribution $\pi_\theta(a_n|s_n)$ given by the policy at step n, and $R_n(\tau) = \sum_{n'=n}^{N}\gamma^{n'-n}r'_n$ is the discounted sum of reward from the current step n to the end of the trajectory. The algorithm for this simple policy gradient method is summarized in Algorithm 1.

However, in practical policy gradient the parameter space and the policy space do not always map congruently. This fact makes it challenging to find a step size α. If α is chosen as a small constant, more iterations are potentially required in the training process. If α is chosen bigger, the agent will be vulnerable to a *performance collapse* in which the agent chooses

Algorithm 1. The Policy Gradient Algorithm.

```
Input The environment $\mathcal{E}$, the initial policy parameters $\theta_0$, the number of epochs $T$, the step size $\alpha$.
Output The optimal policy $\pi_{\theta^*}$
for $i = 0, 1, 2, \ldots, T$ do
    Running current policy $\pi_{\theta_i}$ in $\mathcal{E}$ and obtain a set of trajectories $\mathcal{D}_i = \{\tau_i\}$.
    Compute the policy gradient:
        $(\nabla_\theta J)_i = \frac{1}{|\mathcal{D}_i|} \sum_{\tau \in \mathcal{D}_i} R(\tau) \sum_{n=0}^{N(\tau)} \nabla_\theta \log \pi_{\theta_i}(a_n|s_n),$
    Update the policy by gradient ascent:
        $\theta_{i+1} = \theta_i + \alpha(\nabla_\theta J)_i$
end for
return $\pi_{\theta^*} = \pi_{\theta_{T+1}}$.
```

a bad action, resulting in a sudden drop in its performance. In addition, another issue of policy gradient descent is that it is *sample-inefficient* because it does not reuse data. To address these two issues, the proximal policy optimization algorithms [53] are proposed.

2. Proximal policy optimization

Our algorithm employs proximal policy optimization (PPO) [53], which is a policy gradient algorithm developed by OpenAI. The PPO is motivated by an algorithm called Trust Region Policy Optimization (TRPO) [55], which aims to find an optimal policy iteratively to maximize the $J(\pi)$ without causing a performance collapse. While TRPO applies second-order methods that are complex to compute, PPO uses first-order methods and tricks to keep the updated policy from changing too fast. PPO is much simpler to implement than TRPO while performing well in practice.

To introduce the PPO algorithm, we first define a value function $V_\pi(s) = \mathbb{E}_{s_0=s,\tau\sim\pi}[\sum_{n=1}^{N} \gamma^n r_n]$ and a value-action function $Q_\pi(s, a) = \mathbb{E}_{s_0=s,a_0=a,\tau\sim\pi}[\sum_{n=1}^{N} \gamma^n r_n]$ given state s, action a, discount factor γ, and policy π. These two functions are used to evaluate a state and a given state-action pair. The objective function used in the learning procedure is defined to be the expected reward of policy π as $J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{n=1}^{N} \gamma^n r_n]$. In the policy descent procedure, we will get another policy π' in the next iteration given a current policy π. The objective function changes as

$$J(\pi') - J(\pi) = \mathbb{E}_{\tau\sim\pi'}\left[\sum_{n=1}^{N} \gamma^n A_\pi(s_n, a_n)\right], \tag{D4}$$

where $A_\pi(s_n, a_n) = Q_\pi(s_n, a_n) - V_\pi(s_n)$ is defined as the advantage function. The relative policy performance $J(\pi') - J(\pi)$ provides a metric to measure the improvement of performance after a policy shift. Therefore, maximizing $J(\pi')$ is equivalent to maximizing $J(\pi') - J(\pi)$.

To approximate Eq. (D4), we use the trajectories from the old policy $\tau \sim \pi$ and adjust with importance sampling weights $R_n(\pi) = \frac{\pi'(a_n|s_n)}{\pi(a_n,s_n)}$ [54]. This approximation is given as

$$J(\pi') - J(\pi) \approx J_\pi^{\mathrm{CPI}}(\pi') = \mathbb{E}_{\tau\sim\pi'}\left[\sum_{n=1}^{N} A_\pi(s_n, a_n) R_n(\pi)\right], \tag{D5}$$

where $J_\pi^{\mathrm{CPI}}(\pi')$ is known as a surrogate objective. The surrogate objective function can be additionally written as an average over both $\tau \sim \pi'$ and n as $\mathbb{E}_{\tau\sim\pi',n}[A_\pi(s_n, a_n)R_n(\pi)]$. In our algorithm, we use an alternative version of PPO with a clipped surrogate objective function

$$J^{\mathrm{CLIP}} = \mathbb{E}_{\tau\sim\pi',n}[A_\pi(s_n, a_n) \times \min\{R_n(\pi), \mathrm{clip}(R_n(\pi), 1-\epsilon, 1+\epsilon)\}]. \tag{D6}$$

The above equation is known as the clipped surrogate objective function, and ϵ is a hyperparameter that defines the clipping bound $|R_n(\pi) - 1| \leqslant \epsilon$. This parameter will decay during the training procedure. As the term $\mathrm{clip}(R_n(\pi), 1 - \epsilon, 1 + \epsilon)A_\pi(s_n, a_n)$ bounds the value J^{CPI}, this objective function prevents the updates that create large and risky policy changes.

In objective J^{CLIP}, the most computationally costly parts are the weight $R_n(\pi)$ and advantage $A_\pi(s_n, a_n)$. However, these parts are required in any algorithm that optimizes the surrogate objective function. The remaining calculations are essentially constant-time clippings and minimizings. Therefore, the clipped objective is relatively easy to compute and understand. The whole PPO algorithm is summarized in Algorithm 2.

APPENDIX E: SUPPLEMENTARY NOTE ON THE ALGORITHM AND NUMERICAL EXPERIMENTS

In this Appendix, we provide the design details of the agent and algorithm we used. We additionally provide more numerical data about applying this algorithm to compilation based on Majorana fermion systems. We provide the illustration of our unitary gate compiler based on the PPO algorithm in Fig. 3.

Algorithm 2. The Proximal Policy Optimization Algorithm.

```
Input The environment $\mathcal{E}$, the initial policy parameters $\theta_0$, the initial value function parameters $\phi_0$, the number of epochs $T$, the step size $\alpha$.
Output The optimal policy $\pi_{\theta^*}$
for $i = 0, 1, 2, \dots, T$ do
    Running current policy $\pi_{\theta_i}$ in $\mathcal{E}$ and obtain a set of trajectories $\mathcal{D}_i = \{\tau_i\}$.
    Compute the advantage function $A_{\pi_i}(s, a)$ by using the estimation of value function $V_{\phi_i}$.
    Compute the clipped surrogate objective:
        $J_i^{\text{CLIP}} = \sum_{\tau \in \mathcal{D}_i} \frac{\sum_{n=0}^{N(\tau)} \frac{A_\pi(s_n, a_n)}{N(\tau)} \min\{R_n(\pi), \text{clip}(R_n(\pi), \epsilon)\}}{|\mathcal{D}_i|}$
    Update the policy by optimizing the objective; this step is done by gradient ascent:
        $\theta_{i+1} = \text{argmax}_\theta J_i^{\text{CLIP}}$
    Update the estimation of value function by minimizing the MSE; this step is done by gradient descent:
        $\phi_{i+1} = \text{argmax}_\phi \frac{1}{|\mathcal{D}_i|} \sum_{\tau \in \mathcal{D}_i} \sum_{n=0}^{N(\tau)} \frac{\left[V_{\phi_i}(s_n) - \sum_{t=n}^{N(\tau)} r_t\right]^2}{N(\tau)}$
end for
return $\pi_{\theta^*} = \pi_{\theta_{T+1}}$.
```

1. Training the PPO agent

Our DNN provided by OPENAI baseline package [72] consists of five full connected layers each containing 256 neurons. The activated function is the leaky ReLU function [73] throughout the neural network. We exploit the Adam algorithm [74,75] as our optimizer, and batch normalization is applied.

As mentioned in the main text, the DNN is trained to evaluate the objective function $J(\pi)$ with the reward function:

$$r_n = r_s(U_n, U_t) - C_{g^*}, \tag{E1}$$

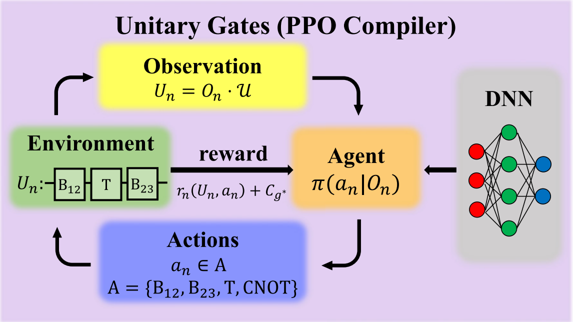
Unitary Gates (PPO Compiler)
Observation
$U_n = O_n \cdot \mathcal{U}$
DNN
Environment
U_n:
B_{12}
T
B_{23}
reward
$r_n(U_n, a_n) + C_{g^*}$
Agent
$\pi(a_n|O_n)$
Actions
$a_n \in A$
$A = \{B_{12}, B_{23}, T, CNOT\}$

FIG. 3. Our DRL environment to produce approximation sequence U_n for the target gate $\mathcal{U}$ required in the previous step using elementary gates. At each step n, the agent gets an observation O_n and feeds O_n as the input vector to the deep neural network (DNN). DNN outputs a probability distribution $\pi(a_n|O_n)$, according to which the agent chooses the next gate to apply in the decomposition sequence. The environment returns a reward r_n to the agent afterward.

$$r_s = \begin{cases} c\left(1 + \max\left\{0, 1 - \frac{n}{L_{U_t}+10}\right\}\right), & d(U_n, U_t) < \epsilon_t, \\ \frac{-d(U_n, U_t)}{L_{\max}} & \text{otherwise,} \end{cases} \tag{E2}$$

where $r_s(U_n, U_t)$ is the state reward obtained by comparing the distance $d(U_n, U_t)$ between the approximation gate U_n and the target gate U_t, C_{g^*} is the additional punishment for the employment of g^* gate, L_{U_t} is the number of gates used for generating U_t, $L_{\max}$ is the maximal number of steps allowed to compile the target gate for the agent, c is a constant to balance rewards and punishments, and ϵ_t is the distance tolerance.

Starting from the identity at each iteration, the agent chooses a gate from $A = \{B_{12}, B_{12}^{-1}, B_{23}, B_{23}^{-1}, T, T^{-1}\}$ in each step and obtains a reward value calculated by Eq. (E1). When the distance between the approximation sequence and the target unitary gate falls within the threshold ϵ_t, the agent obtains a reward and starts a new iteration with a new target gate. When the number of steps exceeds the maximal length $L_{\max}$, the iteration also terminates.

Before the training process, the DNN is initialized with random parameters. From the beginning of the training process, we feed random sequences consisting of gates from $A = \{B_{12}, B_{12}^{-1}, B_{23}, B_{23}^{-1}, T, T^{-1}\}$ of length 10. We choose the accuracy threshold $\epsilon_t = 10^{-3}$ and train the agent and the DNN to search for π with higher reward and generate approximations below the maximal length $L_{\max}$. During the training process, we hold a reward threshold as a function of the length of the random sequence in the training data. If the reward obtained by the agent when compiling the training data reaches a threshold, we increase the length of the random sequence generated as training data until the length reaches 80. In Fig. 4(a), we plot the average reward as a function of the number of steps of training. We can observe that the average reward first increases to the reward threshold and keeps dropping when

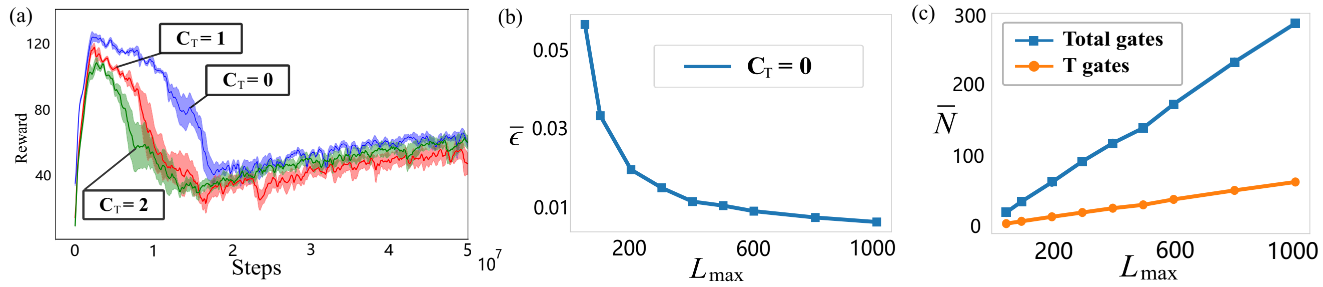

FIG. 4. (a) The training process of the DRL agents for different T cost $C_T = 0, 1, 2$. We plot the average reward value as a function of the number of steps in the training process. (b) The relation of the average compiling error $\overline{\epsilon}$ over the test dataset with the maximal length $L_{\max}$ of generated sequences from the agent. The cost of T gate is set to $C_T = 0$. (c) The average length of the gate sequences and the average number of T gates produced by the agent when we increase the maximal length $L_{\max}$ of generated sequences. The cost of T gate is set to $C_T = 0$.

we increase the length of the sequence among training data. When the length of the randomly generated training sequence reaches the upper bound 80, the average reward will increase as we no longer increase the length. Moreover, we can obtain that the average reward will decrease if we increase the cost for the T gate. We trained this model on a single NVIDIA TITAN V GPU for about one day.

2. More results on applying PPO algorithms to topological quantum compiling on Majorana fermions

To further explore our DRL algorithm in compiling topological quantum compiling on Majorana fermions, we construct another test dataset consisting of 1000 random generated sequences of length 80 with gates chosen from $A = \{B_{12}, B_{12}^{-1}, B_{23}, B_{23}^{-1}, T, T^{-1}\}$. We feed this dataset into the trained agent and increase the maximal length $L_{\max}$ of the generated sequences to observe the changes of average distance $\overline{\epsilon}$, number of T gates, and approximation sequence length.

As shown in Fig. 4(b), we input the test dataset into the agent trained with $C_T = 0$. We observe that when the maximal length $L_{\max}$ increases, the average compilation error first decreases quickly and then converges to a stable value of about 0.005. This indicates that when $L_{\max}$ exceeds a threshold, the major obstacle for improving the accuracy of the compiler will be the sparsity of the net with approximated policy reward of the agent. It is worthwhile to mention that compared with Ref. [52], the search complexity in our algorithm increases linearly rather than exponentially with $L_{\max}$. In Fig. 4(c), we plot the average length of the approximation sequences and the number of T gates in the sequences as a function of $L_{\max}$. It is shown that these two functions increase approximately linearly with $L_{\max}$ while the proportion of the T gate in the approximation sequence remains rarely changed. This result indicates that our DRL agent can stably reduce the usage of the T gate under different $L_{\max}$.

APPENDIX F: ENCODING AND OPERATION ON MAJORANA FERMIONS

In this Appendix, we briefly introduce the encoding methods on Majorana fermion systems. The fusion principle for Majorana fermions is the Ising-type $\tau \times \tau \sim \mathbf{I} + \psi$, with $\mathbf{I}$, τ, ψ representing a vacuum state, a Majorana fermion, and a normal fermion, respectively. In the main text, we consider the four-quasiparticle encoding scheme where each qubit is encoded by four Majorana fermions with the total topological charge as 0. The logical basis states for the qubit are $|0\rangle_L = |[(\bullet, \bullet)_{\mathbf{I}}, (\bullet, \bullet)_{\mathbf{I}}]_{\mathbf{I}}\rangle$ and $|1\rangle_L = |[(\bullet, \bullet)_{\psi}, (\bullet, \bullet)_{\psi}]_{\mathbf{I}}\rangle$. Here, each $\bullet$ is a Majorana fermion, and $\mathbf{I}$, ψ are the two possible fusion channels of a pair of Majorana fermions.

As shown in Fig. 5(a), the gates {H,S} can be realized by braidings on the four-quasiparticle scheme. We denote the four Majorana braiding operators on each quasiparticle as b_i, $i = 1, 2, 3, 4$ in one logic qubit, and these operators satisfy $b_i^\dagger = b_i$, $b_i^2 = I$, and anticommutation relation $\{b_i, b_j\} = 2\delta_{ij}$. As shown in Ref. [58], Pauli operators in a computational basis can be expressed as

$$\sigma^x = -ib_2b_3, \quad \sigma^y = -ib_1b_3, \quad \sigma^z = -ib_1b_2. \tag{F1}$$

Unitary operations can be realized by counterclockwise exchanges of two Majorana fermions as below:

$$B_{jj'} = e^{i\pi/4(ib_jb_{j'})}, \tag{F2}$$

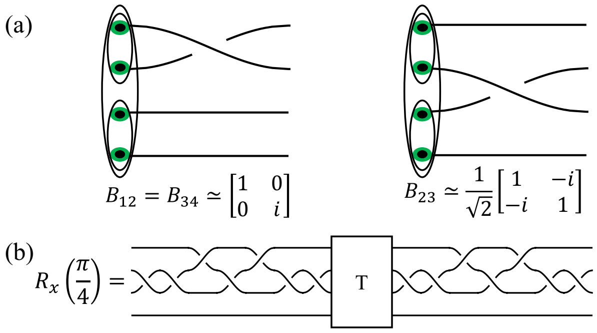
(a)
$B_{12} = B_{34} \simeq \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$B_{23} \simeq \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$
(b)
$R_x\left(\frac{\pi}{4}\right) =$
T

FIG. 5. (a) The two elementary gates that can be implemented through one braiding of Majorana fermions. A logical qubit is encoded into four Majorana fermions (enclosed in ovals). (b) An exact decomposition sequence of using Majorana braidings and a T gate for compiling the $R_x(\frac{\pi}{4})$ gate. $R_x(\theta) = \exp(-iX\theta/2)$ here represents the single qubit rotation along the x-axis in the Bloch sphere.

where j, j' are chosen as two neighboring quasiparticles. Specifically, we give three basic braidings as

$$B_{12} = B_{34} \cong \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = S, \quad B_{23} \cong \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}. \tag{F3}$$

We can implement the H gate with the braiding sequence $H = B_{23}^2 B_{12}^{-1} B_{23} B_{12}^{-1} B_{23}^2$. Hence, we have shown that a single-qubit Clifford group generated by the H gate and the S gate can be realized by Majorana braidings. To compile an arbitrary single-qubit gate, we still have to introduce a T gate to form a universal gate set. A simple example for decomposing an x-rotation is shown in Fig. 5(b).

However, an entanglement gate on a two-qubit cannot be obtained through braiding due to the no-entanglement rule [59]. To implement a two-qubit control gate, we introduce an accessory topological manipulation called nondestructive measurement of the anyon fusion [60,67], which can be implemented through the anyon interferometry. We denote the eight Majorana modes on the two logical qubits as $b_1, \ldots, b_8$, where the control (target) qubits are encoded by the first (last) four modes, respectively. The two-qubit controlled phase-flip gate $\Pi(\sigma^z)$ can be represented by

$$\Pi(\sigma^z) = e^{-(\pi/4)b_3b_4} e^{-(\pi/4)b_5b_6} e^{(i\pi/4)b_3b_4b_5b_6} e^{i\pi/4}. \tag{F4}$$

In the representation above, an ancillary pair b_9b_{10} is added. We measure the fusion of the four Majorana modes $b_4b_3b_6b_9$ and get an outcome ± 1, which corresponds to the vacuum state and the normal fermion with projectors $\Lambda_\pm^{(4)} = \frac{1}{2}(1 \pm b_4b_3b_6b_9)$. Then, we can measure fusion of the Majorana modes (operator) $-ib_5b_9$ with a similar method and get projectors $\Lambda_\pm^{(2)} = \frac{1}{2}(1 \mp ib_5b_9)$. We have the following relation:

$$e^{(i\pi/4)b_3b_4b_5b_6} = 2 \sum_{\kappa,\xi=\pm} V_{\kappa\xi} \Lambda_\kappa^{(2)} \Lambda_\xi^{(4)}, \tag{F5}$$

where $V_{++} = e^{(\pi/4)b_5b_{10}}$, $V_{--} = e^{-(\pi/4)b_5b_{10}}$, $V_{+-} = ie^{(\pi/2)b_4b_3} e^{(\pi/2)b_5b_6} e^{(\pi/4)b_5b_{10}}$, and $V_{-+} = ie^{(\pi/2)b_4b_3} e^{(\pi/2)b_5b_6} e^{-(\pi/4)b_5b_{10}}$ can be implemented by one or several braiding operations of Majorana fermions.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).

[2] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

[3] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[4] C. M. Dawson and M. A. Nielsen, The Solovay-Kitaev Algorithm, Quantum Inf. Comput. **6**, 81 (2006).

[5] A. Y. Kitaev, A. Shen, M. N. Vyalyi, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Boston, 2002), p. 47.

[6] N. C. Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. Van Meter, A. Aspuru-Guzik, and Y. Yamamoto, Faster quantum chemistry simulation on fault-tolerant quantum computers, New J. Phys. **14**, 115023 (2012).

[7] A. G. Fowler, Constructing arbitrary steane code single logical qubit fault-tolerant gates, Quantum Inf. Comput. **11**, 867 (2011).

[8] A. Bocharov and K. M. Svore, Resource-Optimal Single-Qubit Quantum Circuits, Phys. Rev. Lett. **109**, 190501 (2012).

[9] A. Bocharov, Y. Gurevich, and K. M. Svore, Efficient decomposition of single-qubit gates into v basis circuits, Phys. Rev. A **88**, 012313 (2013).

[10] T. T. Pham, R. Van Meter, and C. Horsman, Optimization of the solovay-kitaev algorithm, Phys. Rev. A **87**, 052332 (2013).

[11] Y. Zhiyenbayev, V. M. Akulin, and A. Mandilara, Quantum compiling with diffusive sets of gates, Phys. Rev. A **98**, 012325 (2018).

[12] V. Kliuchnikov, D. Maslov, and M. Mosca, Asymptotically Optimal Approximation of Single Qubit Unitaries by Clifford and T Circuits using a Constant Number of Ancillary Qubits, Phys. Rev. Lett. **110**, 190502 (2013).

[13] P. Selinger, Quantum circuits of t-depth one, Phys. Rev. A **87**, 042302 (2013).

[14] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, An algorithm for the t-count, Quantum Inf. Comput. **14**, 1261 (2014).

[15] N. J. Ross and P. Selinger, Optimal ancilla-free clifford+ t approximation of z-rotations, Quantum Inf. Comput. **16**, 901 (2016).

[16] L. E. Heyfron and E. T. Campbell, An efficient quantum compiler that reduces t count, Quantum Sci. Technol. **4**, 015004 (2018).

[17] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2002).

[18] D. A. Lidar, Lecture notes on the theory of open quantum systems, arXiv:1902.00967.

[19] D. Braun, O. Giraud, I. Nechita, C. Pellegrini, and M. Žnidarič, A universal set of qubit quantum channels, J. Phys. A **47**, 135302 (2014).

[20] W. F. Stinespring, Positive functions on c*-algebras, Proc. Am. Math. Soc. **6**, 211 (1955).

[21] D.-S. Wang, D. W. Berry, M. C. de Oliveira, and B. C. Sanders, Solovay-Kitaev Decomposition Strategy for Single-Qubit Channels, Phys. Rev. Lett. **111**, 130504 (2013).

[22] R. Iten, R. Colbeck, and M. Christandl, Quantum circuits for quantum channels, Phys. Rev. A **95**, 052316 (2017).

[23] S.-J. Wei, T. Xin, and G.-L. Long, Efficient universal quantum channel simulation in ibm's cloud quantum computer, Sci. China Phys. Mech. **61**, 70311 (2018).

[24] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, Quantum circuits for isometries, Phys. Rev. A **93**, 032318 (2016).

[25] C. Shen, K. Noh, V. V. Albert, S. Krastanov, M. H. Devoret, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, Quantum channel construction with circuit quantum electrodynamics, Phys. Rev. B **95**, 134501 (2017).

[26] R. Sweke, I. Sinayskiy, and F. Petruccione, Simulation of single-qubit open quantum systems, Phys. Rev. A **90**, 022331 (2014).

[27] M. H. M. Passos, A. de Oliveira Junior, M. de Oliveira, A. Z. Khoury, and J. A. O. Huguenin, Spin-orbit implementation of the solovay-kitaev decomposition of single-qubit channels, Phys. Rev. A **102**, 062601 (2020).

[28] D.-S. Wang and B. C. Sanders, Quantum circuit design for accurate simulation of qudit channels, New J. Phys. **17**, 043004 (2015).

[29] H. Lu, C. Liu, D.-S. Wang, L.-K. Chen, Z.-D. Li, X.-C. Yao, L. Li, N.-L. Liu, C.-Z. Peng, B. C. Sanders *et al.*, Experimental quantum channel simulation, Phys. Rev. A **95**, 042310 (2017).

[30] T. Xin, S.-J. Wei, J. S. Pedernales, E. Solano, and G.-L. Long, Quantum simulation of quantum channels in nuclear magnetic resonance, Phys. Rev. A **96**, 062303 (2017).

[31] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Mastering the game of go with deep neural networks and tree search, Nature (London) **529**, 484 (2016).

[32] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, Mastering the game of go without human knowledge, Nature (London) **550**, 354 (2017).

[33] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, Improved protein structure prediction using potentials from deep learning, Nature (London) **577**, 706 (2020).

[34] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge *et al.*, Skilful precipitation nowcasting using deep generative models of radar, Nature (London) **597**, 672 (2021).

[35] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[36] X. Gao, Z. Zhang, and L.-M. Duan, An efficient quantum algorithm for generative machine learning, arXiv:1711.02038.

[37] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[38] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, Nat. Mach. Intell. **1**, 155 (2019).

[39] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. **118**, 216401 (2017).

[40] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[41] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[42] L. Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[43] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).

[44] K. Ch’ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[45] Y. Zhang, R. G. Melko, and E.-A. Kim, Machine learning z_2 quantum spin liquids with quasiparticle statistics, Phys. Rev. B **96**, 245119 (2017).

[46] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).

[47] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).

[48] Y. Zhang, A. Mesaros, K. Fujita, S. Edkins, M. Hamidian, K. Ch’ng, H. Eisaki, S. Uchida, J. S. Davis, E. Khatami *et al.*, Machine learning in electronic-quantum-matter imaging experiments, Nature (London) **570**, 484 (2019).

[49] W. Lian, S.-T. Wang, S. Lu, Y. Huang, F. Wang, X. Yuan, W. Zhang, X. Ouyang, X. Wang, X. Huang, L. He, X. Chang, D.-L. Deng, and L. Duan, Machine Learning Topological Phases with a Solid-State Quantum Simulator, Phys. Rev. Lett. **122**, 210503 (2019).

[50] D.-L. Deng, Machine Learning Detection of Bell Nonlocality in Quantum Many-Body Systems, Phys. Rev. Lett. **120**, 240402 (2018).

[51] M. S. Alam, N. F. Berthusen, and P. P. Orth, Quantum logic gate synthesis as a markov decision process, arXiv:1912.12002.

[52] Y.-H. Zhang, P.-L. Zheng, Y. Zhang, and D.-L. Deng, Topological Quantum Compiling with Reinforcement Learning, Phys. Rev. Lett. **125**, 170501 (2020).

[53] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, arXiv:1707.06347.

[54] L. Graesser and W. L. Keng, *Foundations of Deep Reinforcement Learning: Theory and Practice in Python* (Addison-Wesley Professional, Boston, 2019).

[55] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, Trust region policy optimization, in *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (2015), http://proceedings.mlr.press/v37/schulman15.html.

[56] A. Kitaev, Anyons in an exactly solved model and beyond, Ann. Phys. **321**, 2 (2006).

[57] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, Non-abelian anyons and topological quantum computation, Rev. Mod. Phys. **80**, 1083 (2008).

[58] F. Hassler, A. Akhmerov, C. Hou, and C. Beenakker, Anyonic interferometry without anyons: How a flux qubit can read out a topological qubit, New J. Phys. **12**, 125002 (2010).

[59] S. Bravyi, Universal quantum computation with the $\nu = 5/2$ fractional quantum hall state, Phys. Rev. A **73**, 042313 (2006).

[60] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Ann. Phys. **298**, 210 (2002).

[61] M. B. Ruskai, S. Szarek, and E. Werner, An analysis of completely-positive trace-preserving maps on M2, Linear Algebra Its Appl. **347**, 159 (2002).

[62] M. M. Wolf and J. I. Cirac, Dividing quantum channels, Commun. Math. Phys. **279**, 147 (2008).

[63] D. Wang, Algorithmic quantum channel simulation, Graduate thesis, University of Calgary, 2015.

[64] C. King and M. B. Ruskai, Minimal entropy of states emerging from noisy quantum channels, IEEE Trans. Inf. Theory **47**, 192 (2001).

[65] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert, Dissipative Quantum Church-Turing Theorem, Phys. Rev. Lett. **107**, 120501 (2011).

[66] E. Brüning, H. Mäkelä, A. Messina, and F. Petruccione, Parametrizations of density matrices, J. Mod. Opt. **59**, 1 (2012).

[67] S. Bravyi and A. Kitaev, Universal quantum computation with ideal clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[68] V. Gheorghiu, M. Mosca, and P. Mukhopadhyay, T-count and T-depth of any multi-qubit unitary, npj Quantum Inf. **8**, 141 (2022).

[69] P. Selinger, Efficient clifford+ t approximation of single-qubit operators, Quantum Inf. Comput. **15**, 159 (2015).

[70] K. Matsumoto and K. Amano, Representation of quantum circuits with clifford and $\pi/8$ gates, arXiv:0806.3834.

[71] J. R. Johansson, P. D. Nation, and F. Nori, Qutip: An open-source python framework for the dynamics of open quantum systems, Comput. Phys. Commun. **183**, 1760 (2012).

[72] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, Openai gym, arXiv:1606.01540.

[73] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, Rectifier nonlinearities improve neural network acoustic models, in *Proc. icml* (Atlanta, Georgia, USA, 2013), Vol. 30, p. 3, http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[74] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[75] J. R. Sashank, K. Satyen, and K. Sanjiv, On the convergence of adam and beyond, in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 2018, Conference Track Proceedings* (2018), https://openreview.net/forum?id=ryQu7f-RZ.

[76] D. Dong and I. R. Petersen, Quantum control theory and applications: a survey, IET Control. Theor. Appl. **4**, 2651 (2010).

[77] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, J. Math. Phys. **32**, 400 (1991).

[78] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter Error with Commutator Scaling, Phys. Rev. X **11**, 011020 (2021).

[79] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, Phys. Rev. Lett. **114**, 090502 (2015).

[80] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[81] A. Bolens and M. Heyl, Reinforcement Learning for Digital Quantum Simulation, Phys. Rev. Lett. **127**, 110502 (2021).

[82] A. W. Harrow, B. Recht, and I. L. Chuang, Efficient discrete approximations of quantum gates, J. Math. Phys. **43**, 4445 (2002).

[83] A. R. Calderbank, E. M. Rains, P. Shor, and N. J. Sloane, Quantum error correction via codes over gf (4), IEEE Trans. Inf. Theor. **44**, 1369 (1998).

[84] G. Nebe, E. M. Rains, and N. J. Sloane, The invariants of the clifford groups, Des. Codes Cryptogr. **24**, 99 (2001).

[85] M. Planat and P. Jorrand, Group theory for quantum gates and quantum coherence, J. Phys. A **41**, 182001 (2008).

[86] D.-L. Deng, C. Wu, J.-L. Chen, and C. H. Oh, Fault-Tolerant Greenberger-Horne-Zeilinger Paradox Based on Non-Abelian Anyons, Phys. Rev. Lett. **105**, 060402 (2010).

[87] N. E. Bonesteel, L. Hormozi, G. Zikos, and S. H. Simon, Braid Topologies for Quantum Computation, Phys. Rev. Lett. **95**, 140503 (2005).

[88] H. Xu and X. Wan, Constructing functional braids for low-leakage topological quantum computing, Phys. Rev. A **78**, 042325 (2008).

[89] L. Hormozi, G. Zikos, N. E. Bonesteel, and S. H. Simon, Topological quantum compiling, Phys. Rev. B **75**, 165310 (2007).

[90] M. Burrello, H. Xu, G. Mussardo, and X. Wan, Topological Quantum Hashing with the Icosahedral Group, Phys. Rev. Lett. **104**, 160502 (2010).

[91] V. Kliuchnikov, A. Bocharov, and K. M. Svore, Asymptotically Optimal Topological Quantum Compiling, Phys. Rev. Lett. **112**, 140504 (2014).

[92] C. Carnahan, D. Zeuch, and N. E. Bonesteel, Systematically generated two-qubit anyon braids, Phys. Rev. A **93**, 052328 (2016).