PAPER

Efficient and practical quantum compiler towards multi-qubit systems with deep reinforcement learning*

IOP Publishing Quantum Sci. Technol. 9 (2024) 045002 https://doi.org/10.1088/2058-9565/ad420a

Quantum Science and Technology

RECEIVED
7 September 2023

REVISED
27 February 2024

ACCEPTED FOR PUBLICATION
23 April 2024

PUBLISHED
3 July 2024

PAPER

Efficient and practical quantum compiler towards multi-qubit systems with deep reinforcement learning*

Qiuhao Chen[1,2], Yuxuan Du[2,**], Yuliang Jiao[1], Xiliang Lu[1], Xingyao Wu[2,**] and Qi Zhao[3]

[1] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, People's Republic of China
[2] JD Explore Academy, Beijing 101111, People's Republic of China
[3] Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, United States of America
**Authors to whom any correspondence should be addressed.

E-mail: duyuxuan123@gmail.com and wu.x.yao@gmail.com

Keywords: quantum compiling, reinforcement learning, Solovay–Kitaev theorem

Abstract

Efficient quantum compiling is essential for complex quantum algorithms realization. The Solovay–Kitaev (S–K) theorem offers a theoretical lower bound on the required operations for approaching any unitary operator. However, it is still an open question that this lower bound can be actually reached in practice. Here, we present an efficient quantum compiler which, for the first time, approaches the S–K lower bound in practical implementations, both for single-qubit and two-qubit scenarios, marking a significant milestone. Our compiler leverages deep reinforcement learning (RL) techniques to address current limitations in terms of optimality and inference time. Furthermore, we show that our compiler is versatile by demonstrating comparable performance between inverse-free basis sets, which is always the case in real quantum devices, and inverse-closed sets. Our findings also emphasize the often-neglected constant term in scaling laws, bridging the gap between theory and practice in quantum compiling. These results highlight the potential of RL-based quantum compilers, offering efficiency and practicality while contributing novel insights to quantum compiling theory.

1. Introduction

The first-generation quantum computers [1–6] have shown their potential across many scientific domains such as quantum machine learning [7–15], quantum information processing [16–21], and quantum simulation [22–29]. In general, the power of a quantum chip heavily depends on the efficiency of its quantum compiler. That is, an optimal quantum compiler can translate a high-level quantum algorithm into the hardware-level operations (i.e. assembly language) using the fewest number of instructions on a universal basis set to achieve the highest accuracy [30]. Owing to its crucial role, huge efforts have been dedicated to devising efficient quantum compilers and understanding their capabilities. On the theoretical side, the Solovay–Kitaev (S–K) theorem [31] has inspired studies [32, 33] demonstrating that when considering an inverse-closed universal basis set, defined as a set of gates and their inverses, any unitary transformation can be approximated via sequential elemental operators. This sequence exhibits a scaling length of $O(\log^c(1/\varepsilon))$ where $c \geqslant 1$, under the constraint of an arbitrary tolerance ε. However, it is normally the case that the real quantum hardware only permits an inverse-free universal basis set, meaning that not every inverse of the basis gates is included in this set, which makes these results not applicable. To this end, an important line of research is deriving the optimal sequence length in the inverse-free setting. A recent study [34] has proved an inverse-free version of the S–K theorem, which states that the sequence length scales with $O(\log^c(1/\varepsilon))$ but has a larger $c = 8.62$ compared to the inverse-closed setting. Nevertheless, it is still an open question whether the value of c could be further reduced.

* The authors list is in alphabetical order.

The formidable computational challenge stemming from the exponential search space has posed practical obstacles for numerous theoretical quantum compilers [35–46]. To design quantum compilers towards noisy intermediate-scale quantum (NISQ) devices [47–49] mapped the quantum circuit compiling problem into a temporal planning problem. This paradigm shift has facilitated the deployment of classical optimization algorithms, thereby enabling the transformation of a wide spectrum of quantum circuits onto near-term quantum processors. Another line of research [50–57] has involved the application of hybrid quantum–classical algorithms to the field of quantum compiling, often referred to as the variational quantum compiling algorithm. This approach involves the training of parameterized quantum circuits to approximate the desired unitary operation, a task necessitating intricate optimization of circuit architecture and rotation gate parameters throughout the training procedure. However, the aforementioned optimization-based compilers exhibit suboptimal performance in terms of inference time, necessitating the re-execution of their optimization algorithms when confronted with a new target unitary.

Among recent studies on quantum compiling, a novel avenue of exploration [58, 59] has emerged: harnessing reinforcement learning (RL) [60, 61] methodologies to yield the advancement of quantum compilers. Two compelling reasons underpin this trajectory. The first stems from the empirical evidence witnessed in contexts such as AlphaGo [62], AlphaZero [63], and AlphaTensor [64], where RL has demonstrated remarkable efficacy in navigating vast, exponentially scaled search spaces to efficiently identify promising solutions. The second reason derives from the shared inherent mechanics of *sequential decision making* exhibited by both quantum compiling and RL, which invites the prospect of synergizing these two domains. The central concept behind this track is reformulating quantum compiling as an RL-solvable task, which is finding the shortest path (i.e. minimum sequence length) to reach the location closest to the destination (i.e. target unitary). To accomplish this task, the RL-based compiling algorithm can be divided into two distinct phases: the training phase and the inference phase. In the training phase, the initial step involves constructing a training dataset that comprises a set of target unitaries with their optimal compiling sequences derived from the chosen basis set. These unitary-sequence pairs serve as guidelines for the RL-based compiler, enabling the trained compiler to decompose unknown target unitaries in similar performance. The optional utilization of the universal basis set underscores the adaptability and versatility of the RL-based compiler. In the ensuing inference process, these trained RL compilers are capable of directly choosing elements from the universal basis set sequentially thus constructing the compiling sequence for any given target unitary without additional training overhead. Consequently, RL-based quantum compilers embrace two favorable merits over conventional strategies, i.e. the compatibility of different quantum systems and an efficient inference process. However, there are several flaws in current RL-based compilers impeding their practical deployment. First, current RL-based compilers can only be applied to the single-qubit case, or multi-qubit scenario under restricted circumstances. The main reason behind this is that the search space scales exponentially with the increase of the qubit number, which brings the challenge to the RL model. Second, current RL-based compilers operate in relative isolation from established traditional quantum compiling theories. On one hand, the ongoing theoretical exploration of RL remains in a state of flux, frequently characterized as a 'black box' model, thereby rendering the theoretical assessment of RL compilers intricate. This, in turn, impedes the comparative analysis of their performance against conventional quantum compiling algorithms, such as the S–K algorithm. Conversely, the complex structure of composite circuits produced by RL-based compilers presents challenges in their applicability as sources of inspiration for the advancement of compiling theories.

In this work, we devise a novel deep RL-based quantum compiling algorithm with AQ* search. Our proposal serves to enhance both the effectiveness and efficiency of existing RL-based quantum compilers, concurrently advancing the comprehension of quantum compiling principles. Two key technical components of our proposal are the deep Q-network (DQN) [65–67] and the AQ* search strategy [68–70]. In essence, the DQN enriches the resilience of the RL model during the training phase, enhancing its capacity for unitary decomposition by retaining more pertinent information compared to the cost-to-go model utilized in [58]. This augmentation alleviates the challenges associated with multi-qubit compiling tasks. Diverging from conventional inference strategies that generate every compiling sequence element by traversing the universal basis set, the AQ* search leverages the trained DQN to instantaneously output the optimal element in a parallel manner. This results in an efficient inference phase, particularly valuable in the domain of multi-qubit compiling tasks.

We conduct systematic numerical simulations to exhibit the superiority of our proposal on both single-qubit and two-qubit operator compiling. It is noteworthy that previous RL-based compilers [58, 59] for multi-qubit compilation frequently enforce specific structural constraints on the compiling sequence. Our present study operates without such limitations. In the task of single-qubit operator compiling, our algorithm can generate logic quantum operators within a tolerance of 0.999 99 average fidelity under the *inverse-closed* universal basis sets, i.e. the *Clifford*+T universal basis set, Fibonacci anyons basis, and the HRC

efficient universal basis set [33], 0.999 average fidelity under the *inverse-free* universal basis set [71], and 0.9996 average fidelity under the two-qubit universal basis set. For clarity, the comparison among different quantum compilers with respect to the achieved length complexities is summarized in table 1. Notably, the numerical findings underscore the near-optimality of our proposal. Specifically, with the improvement of the compiling accuracy ε, the output sequence length complexity scales as $O\left(\log^{1.025}(1/\varepsilon)\right)$ and $O\left(\log^{1.014}(1/\varepsilon)\right)$ using HRC basis set under single- and two-qubit situations, which matches the lower bound of the length complexity from counting volume up to a constant factor. It's worth noting that while there exist quantum compiling algorithms theoretically approaching optimality, their practical implementation falls short due to computational constraints, such as iteration steps in the S–K algorithm. To the best of our knowledge, this is the first numerical evidence showing the practical saturation of the S–K lower bound. Besides, using an inverse-free basis set, our proposal generates compiling sequence scaling $O\left(\log^{0.9735}(1/\varepsilon)\right)$, which provides empirical evidence for pursuing a more advanced inverse-free S–K theorem. In addition to this, we empirically ascertain that the omitted constant term within the aforementioned sequence length complexity significantly impacts the performance assessment of universal basis sets. For instance, our approach yields compiling sequences that scale with $2.2\log^{1.25}(1/\varepsilon)$ for the *Clifford*+T set and $0.7\log^{1.52}(1/\varepsilon)$ for the Fibonacci anyons basis. While a comparison of the polynomial coefficients of the logarithmic term suggests greater efficiency for the *Clifford*+T set, numerical findings consistently reveal that our proposal attains higher accuracy with fewer operations utilizing the Fibonacci anyons set.

In conclusion, our findings demonstrate the remarkable proximity of our proposed solution to optimality, closely aligning with the lower threshold established by the S–K theorem. This achievement offers a robust benchmark for evaluating traditional compiling strategies within both single-qubit and multi-qubit contexts, across experimental and theoretical realms. Through comprehensive numerical analyses, we have not only advanced quantum compiling algorithms but also contributed to bridging the gap between theoretical results and practical implementations. These results highlight the potential of RL-based quantum compilers, offering efficiency and practicality while contributing novel insights to quantum compiling theory.

2. Quantum compiling in the framework of Markov decision process

Before moving on to present our proposal, we first recap quantum compiling and its reformulation in the language of RL. Suppose that the target unitary is U and a discrete universal basis set is $\mathcal{A}_{\mathcal{U}}$. Mathematically, $\mathcal{A}_{\mathcal{U}}$ is *inverse-closed* if, for every element in this gate set, its exact inverse is also contained; otherwise, $\mathcal{A}_{\mathcal{U}}$ is *inverse-free*. The purpose of quantum compiling is to find a minimum sequence of basis $\{A_0, A_1, \ldots, A_{L-1}\} \in \mathcal{A}_{\mathcal{U}}$ such that the distance between U and $\prod_{j=0}^{L-1} A_j$ is bounded within a pre-defined error ε, i.e.

$$d\left(\prod_{j=0}^{L-1} A_j, U\right) < \varepsilon. \tag{1}$$

The composition of the basis gates $\prod_{j=0}^{L-1} A_j$ refers to the final circuit generated by the specific compiling algorithm. Throughout the whole study, the distance $d(X, Y) = ||X - Y||$ refers to Frobenius Norm (F-norm) and $\mathcal{A}_{\mathcal{U}}$ specifies the universal basis set shown in table 1. Note that our proposal can be easily extended to other universal basis sets and distance measures, e.g. quaternion distance [72], diamond norm [73], Hilbert-Schmidt distance [74]. Besides, the basis set $\mathcal{A}_{\mathcal{U}}$ can be initialized to match the connectivity constraints on real quantum devices.

2.1. Markov decision process

Within the paradigm of standard RL systems [75], an agent engages in an iterative process characterized by trial and error interactions with an environment. The primary aim of the agent is to acquire the capability to make decisions that maximize its cumulative rewards over an extended period. In this iterative process, each discrete reward is acquired from the environment subsequent to the agent's execution of particular actions. These individual rewards merge into cumulative rewards that encapsulate the entirety of the agent's acquired gains throughout the learning trajectory. A succinct representation of this trajectory is visually depicted in figure 1, which provides a detailed insight into a single iteration of interaction while intentionally excluding additional iterations for clarity and focus. To facilitate this learning process, RL researchers commonly employ a mathematical framework known as the discounted Markov Decision Process (MDP) [76]. This framework provides a structured way to describe how the agent's actions influence the environment and how

Table 1. Comparison of our RL-based quantum compiler with other compilers based on RL or conventional strategies along the basis set, scaling (number of qubits), and the length complexity of the compiled sequence.

Compiling method	Basis set	Scaling	Complexity
Inverse-free algorithm [71]	Inverse-free diffusive set	Single-qubit	$O\left(\log^{1.585}(1/\varepsilon)\right)$
Our RL-based compiler	**Inverse-free diffusive set**	**Single-qubit**	$\boldsymbol{O}\left(\log^{0.9735}(1/\varepsilon)\right)$
Other RL-based compiler [58]	Fibonacci anyons	Single-qubit	$O\left(\log^{1.6}(1/\varepsilon)\right)$
Our RL-based compiler	**Fibonacci anyons**	**Single-qubit**	$\boldsymbol{O}\left(\log^{\mathbf{1.52}}\boldsymbol{(1/\varepsilon)}\right)$
Other RL-based compiler [59]	HRC efficient universal set	Single-qubit	$O\left(\log^{1.25}(1/\varepsilon)\right)$
Our RL-based compiler	**HRC efficient universal set**	**Single-qubit**	$\boldsymbol{O}\left(\log^{1.025}(1/\varepsilon)\right)$
Our RL-based compiler	**HRC efficient universal set**	**Two-qubit**	$\boldsymbol{O}\left(\log^{1.014}(1/\varepsilon)\right)$

it can make optimal decisions to achieve its goals. It's worth noting that the RL is a field of study focused on training agents to make intelligent decisions in dynamic and uncertain environments, making MDPs a crucial tool in this task.

Formally, a discounted MDP is termed as a quintuple $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma)$. In this formulation, $\mathcal{S}$ represents a measurable state space, while $\mathcal{A}$ denotes a measurable action space. The mapping $\mathcal{P}:\mathcal{S}\times\mathcal{A}\to\mathcal{M}(\mathcal{S})$ characterizes the Markovian transition probability distribution, indicating how the system moves from one state to another when taking a particular action $a\in\mathcal{A}$. Similarly, $\mathcal{R}:\mathcal{S}\times\mathcal{A}\to\mathcal{M}(\mathbb{R})$ is the immediate reward distribution associated with these state transitions. Additionally, the discount factor $0\leqslant\gamma\leqslant 1$ is introduced to ensure the convergence of the accumulated rewards. In this framework, the transition probability distribution $\mathcal{P}$ and reward distribution $\mathcal{R}$ control the mechanics of state transitions and the distribution of rewards, respectively. Both of these aspects are governed by the characteristics of the environment. The state space $\mathcal{S}$ and action space $\mathcal{A}$ often encompass sets with either a finite or infinite number of components. The agent's goal is to establish a mapping from $\mathcal{S}$ to $\mathcal{A}$ with the purpose of maximizing the cumulative rewards. This mapping can exhibit a probabilistic nature, commonly referred to as a policy.

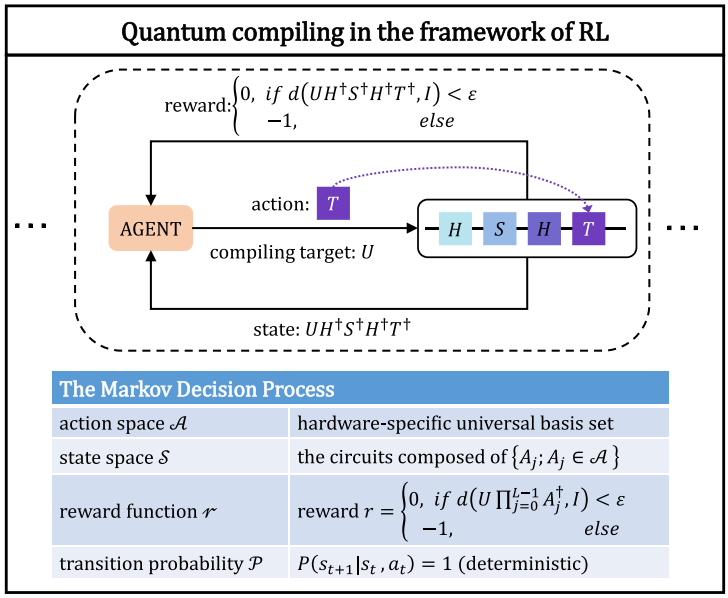
Let us illustrate the concept of an MDP through a concrete example. In this scenario, an agent initiates from an initial state, denoted as s_0. Subsequently, at each iteration t, the agent selects an action $a_t\in\mathcal{A}$ in accordance with a policy $\pi(\cdot\mid s_t)$. Following this action, the agent proceeds to observe the subsequent state, denoted as s_{t+1}, which obeys the transition probability distribution $P(\cdot\mid s_t,a_t)$. Simultaneously, the agent receives an immediate reward, represented as $r_t=r(\cdot\mid s_t,a_t)$. This sequence of events constitutes a single iteration. Upon repeating this process for a total of T iterations, the cumulative sequence of interactions, denoted as $\tau=(s_0,a_0,r_1,s_1,a_1,r_2,\cdots,s_{T-1},a_{T-1},r_{T-1})$, is commonly referred to as a *trajectory*. To optimize its performance and maximize the cumulative rewards, expressed as $\sum_{j=1}^{\infty}\gamma^j r_j$, the agent embarks on a trial-and-error learning journey. It learns the optimal policy, denoted as π^*, by drawing insights from the experiences gathered through various trajectories $\{\tau\}$. These experiences guide the agent's decision-making process, enabling it to establish more rewardful policies.

2.2. Quantum compiling in the framework of MDP

To compile any target U within the tolerance ε, the agent aims to construct a unitary $U_t=\prod_{j=t-1}^{0}A_j$ from a discrete universal basis set $\mathcal{A}_{\mathcal{U}}$ in the sense that the composition of U_t uses the minimum number of basis gates in $\mathcal{A}_{\mathcal{U}}$ and satisfies $d(UU_t^{\dagger},I)<\varepsilon$. This problem of discrete optimization can be cast to an MDP [77]. In particular, as shown in figure 1, the action space $\mathcal{A}$ can be defined as the hardware-specific universal operators $\mathcal{A}_{\mathcal{U}}$, and for simplicity, we refer to both as $\mathcal{A}$ subsequently. The state space $\mathcal{S}$ is referred to as the unitary space $\{S_t=U\cdot U_t^{\dagger}\}$, which encodes all of the information needed by the agent to execute compiling. The transition probability distribution $\mathcal{P}$ is referred to as a deterministic distribution $P(S_{t+1}=UU_{t+1}^{\dagger}\mid S_t,A_t)=1$, where $U_{t+1}=A_t\cdot U_t=\prod_{j=t}^{0}A_j$. Furthermore, the Markov property of this transition can be formulated as $P[S_{t+1}\mid S_t,A_t]=P[S_{t+1}\mid S_1,A_1,S_2,A_2,\ldots,S_t,A_t]$, which means that the distribution of the next state only depends on the present state-action pair and is independent of the previous state-action pairs. Without loss of generality, we treat the cost of different operators A_j chosen from the action space $\mathcal{A}$ to be equal and thus obtain a general reward function

$$R(S_{t-1},A_{t-1})=\begin{cases}0 & \text{if } d(S_t,I)<\varepsilon\\ -1 & \text{otherwise,}\end{cases}\tag{2}$$

where S_t is the next state following the deterministic transition probability distribution $\mathcal{P}$. Specifically, equation (2) denotes the reward acquired by the agent in the tth iteration, which is also referred to as R_t. The



The Markov Decision Process	
action space $\mathcal{A}$	hardware-specific universal basis set
state space $\mathcal{S}$	the circuits composed of $\{A_j; A_j \in \mathcal{A}\}$
reward function r	reward $r = \begin{cases} 0, & if\ d(U\prod_{j=0}^{t-1} A_j^\dagger, I) < \varepsilon \\ -1, & else \end{cases}$
transition probability $\mathcal{P}$	$P(s_{t+1}\|s_t, a_t) = 1$ (deterministic)

Figure 1. Reformulation of quantum compiling into an RL problem modeled by Markov Decision Process (MDP). During an interaction between the agent and environment, the agent selects a single gate T from the universal basis set $\mathcal{A}$ to approach the compiling target U; the constructed unitary feeds the agent with a new state $HSHT$ as the consequence of the selected gate T and a reward, 0 or -1, measuring the quality of the selected gate sequence $\{H, S, H, T\}$. The MDP ends with a non-negative reward signal 0, ensuring that the compilation is accomplished. The compiled unitary is formulated by the sequence of the selected gates during this MDP .

interaction between the agent and the environment terminates at $r(S_{t-1}, A_{t-1}) = 0$, indicating that $d(S_t, I) < \varepsilon$ has been achieved.

Following the above explanations, the aim of the optimal quantum compiler is finding the optimal trajectory $\tau^* = (S_0, A_0, R_1, S_1, \cdots, S_{t-1}, A_{t-1}, R_t, S_t)$ with the maximum cumulative rewards $\sum_{j=1}^{t} \gamma^j R_j$ where $\gamma = 1$. The S–K theorem guarantees the existence of the composite unitary $U_t = \prod_{j=t-1}^{0} A_j$ satisfying $d(S_t, I) < \varepsilon$ for any tolerance ε. For the optimal trajectory τ^*, the state starts from the target unitary $S_0 = U$ and ends at $S_t = UU_t^\dagger$ satisfying $d(S_t, I) < \varepsilon$. This implies $S_t \approx I$ and $U \approx U_t$. The maximum cumulative rewards amount that the agent accomplishes the quantum compiling with the minimum number of operators below the precision threshold ε and the sequence length complexity arrives at the lower bound $O(\log(1/\varepsilon))$.

We note that finding the optimal trajectory of the above MDP is extremely challenging, due to the exponential state space and sparse reward [78]. Concretely, for a single-qubit system whose universal basis set contains 6 basis gates, when $t = 30$, the state space scales with $6^{30} \approx 10^{23}$; for a two-qubit system whose universal basis set contains 14 basis gates, when $t = 30$, the state space scales with $14^{30} \approx 10^{34}$. Due to this exponentially large space, the condition $d(S_{t+1}, I) < \varepsilon$ in equation (2) is hard to satisfy, which incurs $r = -1$ along the trajectories without useful information. Therefore, an untrained agent would may fail to see a non-negative reward signal and unlikely outputs a demanded gate sequence through the random trial and error strategy. To avoid the issue of the sparse reward, our proposal first generates the trajectories with non-negative reward signals and then adopts the value-based RL methods to learn the optimal policy π^*. Namely, by approximating the optimal action-value function

$$Q^*(s, a) = \max_{\pi} \mathbb{E}\left[\sum_{j=0}^{\infty} \gamma^j R_j \mid s_0 = s, a_0 = a, \pi\right],$$

which estimates the maximum cumulative rewards starting from state $s \in \mathcal{S}$ and taking action $a \in \mathcal{A}$. With this function, we can determine the optimal trajectory for compiling. Concretely, given the compiling target U, we can identify the optimal action $A_0 = \arg\max_{a \in \mathcal{A}} Q^*(U, a)$ and apply it to the state U. Upon obtaining a new state $UA_0^\dagger$, we can then determine the next optimal action $A_1 = \arg\max_{a \in \mathcal{A}} Q^*(UA_0^\dagger, a)$. This process can be repeated recursively to identify the compiling sequence $\{A_0, A_1, \ldots\}$ to approximate U. The optimality of $Q^*(s, a)$ guarantees the shortest compiling sequence.

It's worth noting that this reformulation process can also be applied to the noise setting by updating the state space accordingly. For example, as shown in figure 1, to obtain the matrix representation of a noisy

circuit *HSHT*, we can utilize the quantum process tomography result as a new state. This results in states constituting a higher-dimensional Hilbert space, posing additional challenges in finding the optimal trajectory.

2.3. Value-based RL methods

The value-based RL algorithms focus on detecting optimal policy π^* by approximating the optimal value function. In the context of RL, there are two kinds of value function, i.e. the state-value function $V^{\pi}(s)=\mathbb{E}\left[\sum_{j=0}^{\infty}\gamma^j R_j \mid s_0=s,\pi\right]$ that outputs the cumulative rewards starting from state s under the policy π, and action-value function $Q^{\pi}(s,a)=\mathbb{E}\left[\sum_{j=0}^{\infty}\gamma^j R_j \mid s_0=s,a_0=a,\pi\right]$ that outputs the cumulative rewards starting from the state s and action a under the policy π. Bellman optimality equation for the state-value function $V(s)$ indicates

$$V^{*}(s)=\max_{a}\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[r+\gamma V^{*}(s')\,|s\right], \tag{3}$$

where (s,a,P,r,γ) corresponds to the state, action, transition probability distribution, reward, and discount factor, respectively, and s' is the next state obeying the deterministic distribution $P(\cdot\mid s,a)$. Using this Bellman equation as an iterative update, i.e. $V^{(t)}(s)=\max_a\mathbb{E}\left[r+\gamma V^{(t-1)}(s')\,|s\right]$, the state-value function $V^{(t)}$ also converges to the optimal state-value function obeying equation (3), i.e. $V^{(t)}\to V^{*}$ as $t\to\infty$. Similarly, the bellman iteration for the action-value function $Q(s,a)$ will be elaborated in the next section. After the bellman iteration, the optimal value function can guide the agent towards the optimal trajectory τ^* equipped with maximum cumulative rewards.

3. RL-enhanced quantum compiler

The quantum compiling problem in equation (1) can be reformulated into a *value-based RL problem*, which in turn can be addressed by the deep Q-network [65] (detailed in the subsequent context). As shown in figure 1, the agent proceeds compilation by adding gates sequentially rather than searching a complete gate sequences directly. To avoid the sparse reward issue, our proposal adopts the distance measure $d(U\prod_{j=0}^{L-1}A_j^{\dagger},I)$ instead of $d(\prod_{j=0}^{L-1}A_j^{\dagger},U)$, where U is the target unitary and I is the identity operator. This tactic was first proposed by [58]. Using the language of value-based RL, quantum compiling is equivalent to finding the action-value function $Q(s,a)$ obeying the following *Bellman optimality equation*,

$$Q^{*}(s,a)=\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[r+\gamma\max_{a'}Q^{*}(s',a')\,|s,a\right], \tag{4}$$

where the physical meaning of s, a, P, r, and γ is state (i.e. $U\prod_{j=0}^{L-1}A_j^{\dagger}$), action (i.e. the adopted quantum gate $A_j\in\mathcal{A}$), transition probability distribution, reward, and the discount factor, respectively. The notation s' refers to the next state of the composed unitary obeying the distribution $P(\cdot\mid s,a)$. The optimal action-value function $Q^{*}(s,a)$ is defined as the maximum expected cumulative rewards achievable by taking action a when some states $\{s\}$ are observed. In the task of quantum compiling, $Q^{*}(s=U\prod_{j=1}^{L}A_j^{\dagger},a=A_{L+1})$ refers to the negative shortest distance (the minimum sequence length) between the identity state I and the next state $s'=U\prod_{j=1}^{L+1}A_j^{\dagger}$, which is obtained by interacting the state s with the action gate a. Using the Bellman equation to attain the optimal policy, the connection of the action-value function between the tth iteration and the $(t-1)$th iteration can be established by the Bellman operator, i.e.

$$Q^{(t)}(s,a)=\mathbb{E}\left[r+\gamma\max_{a'}Q^{(t-1)}(s',a')\mid s,a\right]. \tag{5}$$

It has been proved that such action-value function converges to the optimal action-value function, $Q^{(t)}\to Q^{*}$ as $t\to\infty$ [76]. Compared with prior RL-based strategies, this reformulation can not only alleviate the computational bottleneck but also enables the theoretical guarantee for convergence. Concretely, different from [58] that approximates the optimal state-value function $V^{*}(s)$, we employ the action-value function $Q^{*}(s,a)$, which has advantages in both the training phase and the inference phase. The state-value function $V(s)$ can be represented as a deep neural network, which outputs a scalar approximating the value of the input state s. Meanwhile, the action-value function $Q(s,a)$ can also represented as the deep Q-network, which outputs a vector approximating the value of every next state s'. While the number of parameters of the deep Q-network grows linearly with the size of the action space, the number of forward passes needed to compute the loss function stays constant for each update. Moreover, [68] has indicated that in a large action space $\mathcal{A}$, the training time for deep Q-network can be up to 100 more times faster than the state-value

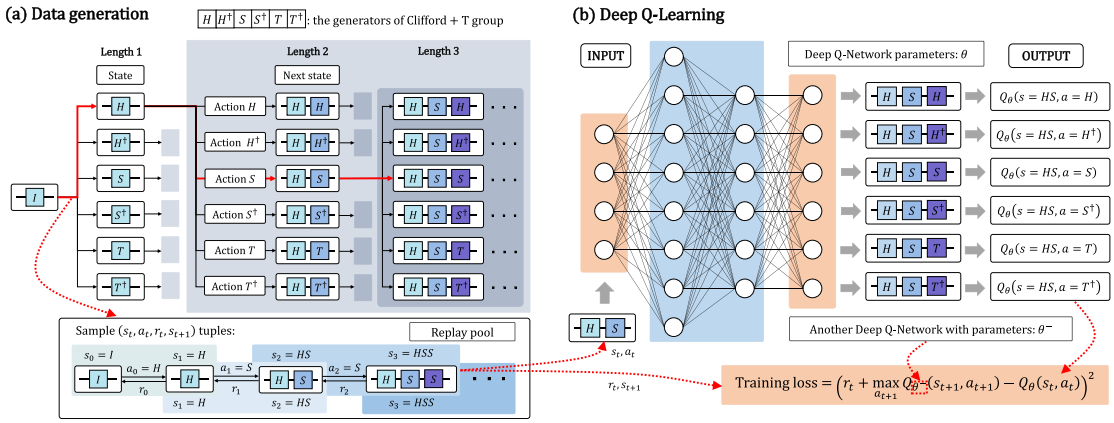


Figure 2. The process of data generation and deep Q-learning. (a) The process of data generation for single-qubit cases under the *Clifford+T* universal basis set. The training experience is composed of the sampled trajectories across the whole circuit space, e.g. the trajectory (highlighted by the red line) is randomly sampled from the whole circuit space expanded by a universal basis set. The selected trajectories are put into the replay pool to decouple the correlation between the sample tuples. Namely, a trajectory is decomposed into sequences of sample tuples, and then fed into the learning process, e.g. taking action $a_t = S$ from state $s_t = UH^\dagger S^\dagger$ leads to the next state $s_{t+1} = UH^\dagger S^\dagger S^\dagger$ with reward -1. (b) The process of deep Q-learning amounts to minimizing the training loss at the bottom right plot. The blue (orange) boxes are hidden layers (input and output layers) of deep Q-networks (DQNs). A key feature of DQNs is that their output layers describe the action-value functions. The input of the DQN is a vector transformed from the unitary matrix corresponding to the state unitary, and the output is a vector corresponding to the value function of all next state unitaries.

function with a similar layout. Besides, in the inference phase, each query of the action-value function is equivalent to querying the state-value function with $|\mathcal{A}|$ times. These reductions are beneficial to generalize our algorithm to the multi-qubit scenario.

3.1. Data generation and deep Q-learning

We now elucidate how our proposal accomplishes the quantum compiling task described in equation (5). Our proposal consists of three components. At the initialization stage, our protocol builds an efficient sample distribution that approximates a uniform distribution over each state-action pair. These sample pairs are obtained through the decomposition of trajectories characterized by non-positive rewards. In the training procedure, our protocol employs the *deep Q-network* (DQN) [65, 68] to minimize the cost function derived from equation (5), which efficiently approximates the optimal action-value function $Q^*(s,a)$ taking advantage of the generated sample experience. At the inference stage, our protocol exploits the AQ* search guided by the trained DQN to seek the optimal gate sequence for a given quantum operation.

The optimization of DQN is as follows. Recall that in the context of quantum compiling [79], the state space is exponentially scaled with the sequence length and number of qubits. As such, the updating rule in equation (5) is impractical for multi-qubit systems. To this end, an alternative is employing a DQN [65, 68] $Q(s,a;\theta^{(t)})$ as shown in figure 2(b) to estimate the action-value function $Q^{(t)}(s,a)$ in equation (5). This approximation is achieved by tuning trainable parameters $\theta^{(t)}$ to minimize a sequence of loss function $\{L_t(\theta^{(t)}); t = 0, 1, 2, \cdots\}$ with

$$L_t\left(\theta^{(t)}\right) = \mathbb{E}_{s,a\sim\phi(\cdot)}\left[\left(y^{(t)} - Q\left(s,a;\theta^{(t)}\right)\right)^2\right], \tag{6}$$

where $y^{(t)} = \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[r + \gamma \max_{a'} Q\left(s',a';\theta^{(t-1)}\right)|s,a\right]$ is the target for iteration t and $\phi(s,a)$, a.k.a., *behavior distribution*, can be any probability distribution over states and actions [65, 76]. To attain a good learning performance, here we construct the empirical behavior distribution over the sample pairs $S_{\text{pair}} := \{(s_t, a_t) : t = 0, 1, \ldots\}$ by randomly and sequentially taking actions from the goal state I. An intuition of generating trajectories is shown in figure 2(a). In this way, the empirical loss function can be calculated by $\hat{L}_t\left(\theta^{(t)}\right) = 1/|S_{\text{pair}}| \sum_{(s,a)\in S_{\text{pair}}} [(y^{(t)} - Q(s,a;\theta^{(t)}))^2]$. Here $y^{(t)}$ can be readily computed since we know the distribution $s' \sim P(\cdot \mid s,a)$; $Q\left(s,a;\theta^{(t)}\right)$ can be readily computed given s, a, and $\theta^{(t)}$. Besides, the implementation of DQN allows a stable learning performance. Specifically, we start with a DQN with random initialized parameters, and the training experience below a predefined gate sequence length d is collected to optimize the DQN by minimizing the cost function in equation (6). The optimization of DQN with gate sequence length d is continuously proceed until the training loss in equation (6) is below a pre-fixed threshold δ. Subsequently, the training experience below a predefined gate sequence length $d+1$ is fed into DQN to minimize the cost function in equation (6). Similarly, the optimization of DQN is

continuously proceed until the training loss reaches a threshold δ. The learning process stops when d reaches a pre-set threshold.

We remark that both behavior distribution $\phi(s,a)$ and the adopted DQN contribute to the superior performance of our protocol. Specifically, $\phi(s,a)$ determines the efficiency of the learning process. It ensures that every generated trajectory is an experience of a successful compilation, which avoids the dilemma of sparse reward [78] and allows the production of high-quality sample information. In addition, DQN fully exploits the sample trajectories, which in turn efficiently approximates the optimal strategy and ensures the achievement of multi-qubit compilation. The process of successively increasing d in optimization ensures learning stability.

3.2. AQ* search

Once the training is completed, the proposed quantum compiler becomes capable of inferring the gate sequence for previously unseen quantum operators via the AQ* search algorithm [68]. The inference process is iterative in nature. In the initial iteration, the AQ* search takes the target unitary U as input and generates a vector that assesses the quality of each action within the action space $\mathcal{A}$, guided by the trained DQN. Specifically, this action quality of an action $A' \in \mathcal{A}$, is quantified by the number of required basis gates to transform the state $A'U$ into a state close to the identity matrix. This quality metric aligns with the negative cumulative rewards for the state $A'U$. Thus, the accuracy of the inference process is contingent on the quality of the trained DQN. Subsequently, in each subsequent iteration, the best action identified in the preceding iteration is applied to the input state from the prior iteration (e.g. $A'U$ in the second iteration). The inference process continues until the AQ* search method outputs a state that closely approximates the identity matrix, satisfying equation (1). It's important to note that this approach diverges from conventional quantum compiling algorithms and RL-based compilers based on state value functions [58]. In these alternatives, each iteration within the inference process requires traversing the entire universal basis set $\mathcal{A}$ and executing a heuristic function for $|\mathcal{A}|$ times to select a promising action. In contrast, our proposal only necessitates a single execution of the DQN for this task. Celebrated by the parallel testing of $|\mathcal{A}|$ actions, the inference time of our proposal is ignorable, which ensures its scalability. To attain the optimal trajectory τ^* starting from any target U, we define an evaluation function

$$f(s,a;U) = G(s;U) + Q(s,a;\theta) \tag{7}$$

where $G(s;U)$ represents the realized cumulative rewards from the target unitary U to the state s, and $Q(s,a;\theta)$ refers to DQN approximating the maximum cumulative rewards from the state s to I taking action a. It's important to note that constructing both of these functions involves minimal computational overhead. For the construction of $G(s;U)$, it is only necessary to maintain a record of the number of basis gates required to transform the target unitary U into the state s. On the other hand, $Q(s,a;\theta)$ essentially represents the trained RL model and does not require additional computational burden for its construction. The essence of $f(s,a;U)$ is to provide effective guidance for finding the shortest gate sequence approximating the target unitary U.

In practical terms, we initiate with a set of intermediate states and the target for compilation, $\{U\prod_j A_j^\dagger, U\}$. Subsequently, we select the action yielding the highest reward $A = \arg\max_a f(s,a;U)$ for each s and update s with its subsequent state s' as determined by the action A, following the transition probability distribution P. When the distance between a state in $\{s\}$ and the identity state I falls below a predefined termination accuracy threshold ε, i.e. satisfying equation (1), the desired sequence between U and I, within the specified tolerance, is achieved. The AQ* Search harnesses the advantages of DQNs, resulting in superior performance compared to other methodologies, as outlined in the next section.

4. Results

We conducted extensive experiments to exhibit the effectiveness of our proposal. To assess the universality and the efficiency of our proposal, we separately apply our proposal to compile inverse-closed single-qubit and two-qubit operators with different universal basis sets. In addition, we comprehensively evaluated our proposal's performance by employing different distance metrics, including the F-norm, fidelity, and spectral norm. Furthermore, to address a long-standing problem in the inverse-free S–K theorem, we apply our proposal to compile single-qubit operators under an inverse-free basis to seek a lower exponent c. In addition, we explored the influence of heuristic search algorithms on the quality of the compiled sequences and demonstrated the advantages of the AQ* search algorithm.

The experiments were conducted using a Nvidia Tesla V100 GPU. Table 2 presents the computational costs associated with the training phase, including training time, sample size, and parameter size. It is evident

Table 2. The computational cost during the training phase.

Compiling task	Training time (days)	Sample size (billions)	Parameter size (thousands)
Single-qubit	2–4	1.898	72
Two-qubit	7–8	12.665	128

that the computational cost in the two-qubit case is nearly twice that of the single-qubit case, with a notable difference in sample count, exceeding six times.

4.1. Universal basis sets

Here we present all the universal basis sets utilized in this study, encompassing both the inverse-closed and inverse-free sets.

Fibonacci anyons basis sets. Fibonacci anyons are quasiparticle excitations of topological states that obey non-Abelian braiding statistics [80] and the simplest non-Abelian quasiparticles that enable universal topological quantum computation [81] by braiding alone [82]. Their mathematical expression is

$$A_1 = \begin{pmatrix} \eta^{-4} & 0 \\ 0 & \eta^{3} \end{pmatrix}, \quad A_2 = \begin{pmatrix} -\phi^{-1}\eta^{-1} & \phi^{-\frac{1}{2}}\eta^{-3} \\ \phi^{-\frac{1}{2}}\eta^{-3} & -\phi^{-1} \end{pmatrix},$$

where $\eta = e^{i\pi/5}$ and $\phi = \frac{\sqrt{5}+1}{2}$.

HRC basis set. The HRC universal basis set proposed in [33] takes the form

$$B_1 = \tfrac{1}{\sqrt{5}}\begin{pmatrix} 1 & 2i \\ 2i & 1 \end{pmatrix}, \quad B_2 = \tfrac{1}{\sqrt{5}}\begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, \quad B_3 = \tfrac{1}{\sqrt{5}}\begin{pmatrix} 1+2i & 0 \\ 0 & 1-2i \end{pmatrix}.$$

***Clifford+T* basis set.** The n-qubit *Clifford* group is generated by the Hadamard gate H, the phase gate S, the controlled-not gate [83]. One can obtain a universal basis set by adding the *non-Clifford* operator T into *Clifford* group. The mathematical expression of the basis gates is

$$H = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

In the numerical simulations, considering that the training difficulty is exacerbated by the sparsity of T and S, we replace S by $H \cdot S$ in training DQN.

Inverse-free basis set. A limitation of the standard S–K theorem is that it requires the gate set to be inverse-closed. An alternative would be inverse-free gate sets, which are diffusive enough such that the sequences of moderate length cover the space of unitary matrices in a uniform way [71]. To verify our proposal in the inverse-free setting, we exploit a diffusive set $\mathcal{M}$ composed of two 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

$$\widehat{F} = \begin{pmatrix} -0.40194 - \mathrm{i}0.43507 & -0.36803 - \mathrm{i}0.71674 \\ 0.36803 - \mathrm{i}0.71674 & -0.40194 + \mathrm{i}0.43507 \end{pmatrix}.$$

4.2. Inverse-closed single-qubit universal gates

To demonstrate the versatility of our proposal, we focus on three different inverse-closed universal basis sets to make single-qubit compiling, i.e. *Clifford+T* group, Fibonacci anyons, and HRC universal basis set. In the following, we evaluate the performance of our RL-based quantum compiler by separately compiling single-qubit and multi-qubit operators using these basis sets.

The protocol setting is as follows. The employed DQN $Q(s, a; \theta)$ consists of two hidden layers, six residual blocks, and $|\mathcal{A}|$ output neurons, where $|\mathcal{A}| = 5, 6, 4$ for *Clifford+T* group, HRC gates, and Fibonacci anyons, respectively. A visual description of the DQN can be seen in figure 3. The first two hidden layers are of sizes 5000 and 1000 for single-qubit cases, 6000 and 2000 for multi-qubit cases, respectively, and each residual block consists of two hidden layers with 1000 hidden neurons each. We exploit the Adam [84] optimizer to optimize DQN, and set the learning rate as $\eta = 10^{-3}$ without weight decay. We set the accuracy threshold in equation (1) as $\varepsilon = 10^{-3}$ and the threshold of the mean square error loss in equation (6) as $\delta = 10^{-2}$. The gate sequence length d in figure 2(a) varies from 3 to 40. In the inference stage, the number of test samples is set as 10^3. During the training process, as a demonstration of efficiency and practicality, we keep the time

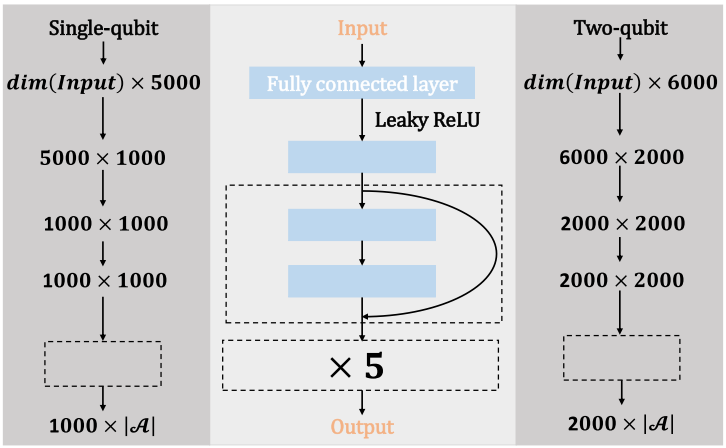


Figure 3. The architecture of the deep Q-network for single-qubit and multi-qubit situations.

consumption of the training phase within an acceptable range: no more than one week. This thrift generates a separation between the actual DQN we learned within a fixed training time and the best DQN our algorithm can learn, and this separation gets smaller as training time increases.

To quantitively measure the efficiency of our protocol under different basis sets, we exploit three common metrics in quantum compiling, i.e. Frobenius norm

$$\begin{aligned} M_1(U_n, U) &= ||{U_n}^\dagger U - I||_F \\ &= \sqrt{\mathrm{Tr}\left(\left(U_n^\dagger U - I\right)\left(U_n^\dagger U - I\right)^\dagger\right)} \\ &= \sqrt{\mathrm{Tr}\left(2I - U^\dagger U_n - U_n^\dagger U\right)}, \end{aligned}$$

fidelity

$$M_2(U_n, U) = \int \langle\psi|{U_n}^\dagger U|\psi\rangle\langle\psi|U^\dagger U_n|\psi\rangle \mathrm{d}\psi,$$

with Haar measure $\int \mathrm{d}\psi = 1$, and spectral norm

$$M_3(U_n, U) = \sup_{\langle\psi|\psi\rangle=1} \|(U_n - U)|\psi\rangle\|_2,$$

where $\|\cdot\|_2$ represents the vector 2-norm. The metric M_1 measures the distance between two matrices and its derivatives have been used in many quantum subfields [58, 85]. The metric M_2 focuses on the fidelity between the complied and target unitary, as discussed in [59]. Additionally, the spectral norm M_3 has been utilized in [34].

The simulation results of compiling single-qubit operators under the fidelity metric M_2 are shown in figure 4. In particular, figures 4(a)–(c) exhibit the simulation results under different basis sets. Particularly, the average fidelity of the HRC set, Fibonacci anyons, and the *Clifford*+T group is $\overline{M_2} = 99.999\%$ when $d = 15$, $\overline{M_2} = 99.996\%$ when $d = 40$, and $\overline{M_2} = 99.88\%$ when $d = 40$, respectively. According to the two metrics and the learning-time consumption, the HRC set outperforms the rest two universal basis sets. The inferior performance of the *Clifford*+T group is prohibited by its sparsity. Figure 4(b) depicts the compiled gate sequence during the training process. That is, with increasing the gate sequence length d, the compiling accuracy is constantly enhanced.

We further utilize the simulation results to infer the sequence length complexity of our RL-based quantum compiler via extrapolation. With setting $\varepsilon = 1 - \overline{M_2}$, the sequence length complexity of our RL-based quantum compiler scales with $0.89\log^{1.025}(1/\varepsilon)$ for the HRC gates, $0.737\log^{1.52}(1/\varepsilon)$ for the Fibonacci anyons, and $2.196\log^{1.25}(1/\varepsilon)$ for the *Clifford*+T group, respectively. An illustration is demonstrated in figure 4(d) and a comparison with other quantum compilers is summarized in table 1. Specifically, our RL-based compiler outperforms prior RL-based compilers and is near-optimal, guaranteed by the inverse-closed S–K Theorem. In addition to this, we empirically ascertain that the omitted constant

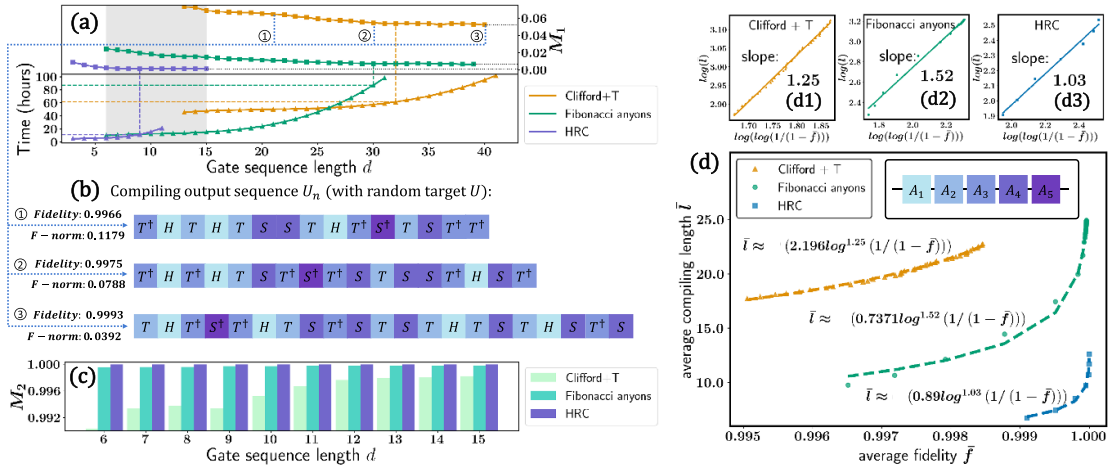


Figure 4. Comparison between different inverse-closed universal basis sets. (a) Comparison of accuracy among different universal basis sets. The accuracy is measured by the F-norm metric M_1. Every square point is obtained by averaging the distance $M_1(U_n, U)$ of 10^3 samples, where U_n is generated by the AQ* search guided by the trained DQN. Every triangle point denotes the time consumption during the training phase for the DQN expanded by different basis sets with the gate sequence length d. (b) The demonstration of the training process indexed by the gate sequence length d using *Clifford+T* universal basis set. (c) The compiling accuracy of the shadow region in (a) using the fidelity metric M_2. (d) The top three subplots examine the linearity between $\log\left(\log\left(1/(1-\bar{f})\right)\right)$ and $\log\left(\bar{l}\right)$ under three basis sets. The lower plot depicts the sequence length complexity versus the average fidelity $\bar{f}$ under different inverse-closed universal basis sets in the single-qubit scenario.

term within the compiling sequence length complexity significantly impacts the performance assessment of universal basis sets. For instance, our approach yields compiling sequences that scale with $2.2\log^{1.25}(1/\varepsilon)$ for the *Clifford+T* set and $0.7\log^{1.52}(1/\varepsilon)$ for the Fibonacci anyons basis. While a comparison of the polynomial coefficients of the logarithmic term suggests greater efficiency for the *Clifford+T* set, numerical findings consistently reveal that our proposal attains higher accuracy with fewer operations utilizing the Fibonacci anyons set. These observations collectively affirm the efficacy of our proposal, concurrently enhancing our comprehension of quantum compiling challenges.

4.3. Inverse-closed two-qubit gates

Most RL-based quantum compilers suffer from the exponentially large state space with respect to the number of qubits. As a result, they have to restrict the state space for multi-qubit compiling. In [58], they leveraged insights from KAK decomposition [86] to fix the number of the employed CNOT gates and their position in the complied unitary. As a result of this approach, the RL-based compiler only optimizes the single-qubit components in this template. This external intervention significantly simplifies the challenge of two-qubit quantum compiling, which reduces the two-qubit operator compiling task into a single-qubit operator compiling task. However, it is important to note that this approach cannot provide convincing evidence for RL compilers to finish multi-qubit compiling tasks. Alternatively, [59] fixed the compiling targets as the multiplication of the universal gates. In contrast, our simulation selects compiling targets exclusively from the rotation $X\otimes Z$ gate category, without imposing any constraints on the angles involved in these rotations.

We now demonstrate more simulation results. In particular, the adopted universal basis set is formed by HRC universal gates and the CNOT gates. Mathematically, the action space yields

$$\begin{aligned}\mathcal{A}' = \Big\{ & B_1\otimes I, B_1^\dagger\otimes I, I\otimes B_1, I\otimes B_1^\dagger,\\ & B_2\otimes I, B_2^\dagger\otimes I, I\otimes B_2, I\otimes B_2^\dagger,\\ & B_3\otimes I, B_3^\dagger\otimes I, I\otimes B_3, I\otimes B_3^\dagger,\\ & |0\rangle\langle 0|\otimes I + |1\rangle\langle 1|\otimes X,\\ & I\otimes|0\rangle\langle 0| + X\otimes|1\rangle\langle 1|\Big\},\end{aligned}$$

where the dimension of the action space is $|\mathcal{A}| = 14$. The gate sequence length d ranges from 3 to 20. For ease of illustration, we set the target operator as $U = e^{-i\frac{\alpha}{2}(X\otimes Z)}$.

The simulation results of compiling two-qubit operators are shown in figure 5. Here we only exploit the HRC universal basis set and the CNOT gate. Compared to the single-qubit case, most of the hyperparameters settings remain unchanged during training, except for the structure of DQN, which is detailed in figure 3.

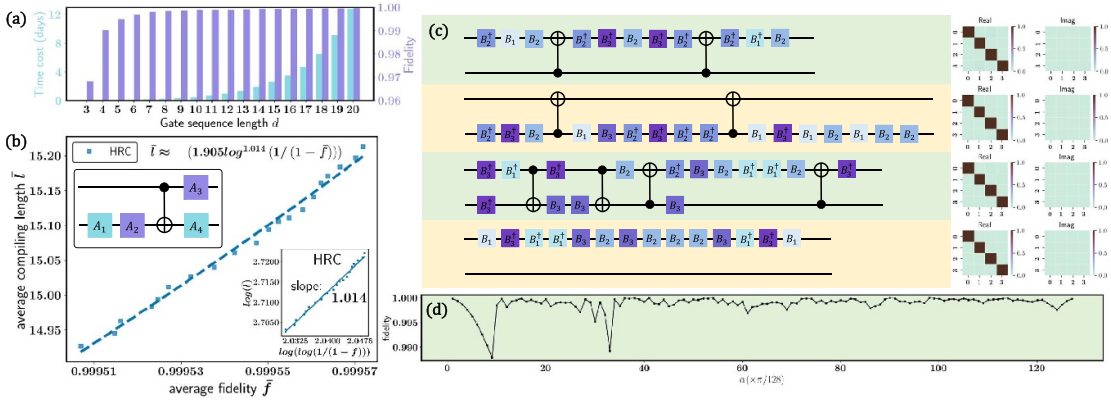


Figure 5. The performance of our RL-based quantum compiler on a two-qubit universal basis set. (a) The light-colored portion of the bar chart records the time cost of the DQN expanded by a two-qubit universal basis set with gate sequence length d; and the dark part records the compiling fidelity M_2 induced by the DQN with gate sequence length d. (b) Every data point is obtained by averaging the length of the approximating sequence generated by AQ* search using a validation set of 10^3 unitary targets. We check the linearity between $\log(\log(1/\varepsilon))$ and $\log(\bar{l})$ in the subplot at the bottom right. We plot the output length complexity of the two-qubit universal basis set as a function of average fidelity $\bar{f}$. (c) The compiling target operator is $U = e^{-i\frac{\alpha}{2}(X\otimes Z)}$. Every unitary U_n is generated by AQ* search guided by the learned DQN. Here we examplify 4 target unitary matrics $\alpha = \{1, 13, 26, 44\}$. The heatmaps on the right panel depict the real and imaginary parts of the unitary matrix $U_n^\dagger U$, where U_n is generated from our proposal. The horizontal and vertical coordinates in the heatmaps correspond to the corresponding positions of the unitary matrix. (d) We generalize our compiling target as $\{e^{-i\frac{\alpha}{2}(X\otimes Z)} | \alpha = k \times \frac{\pi}{128}\}$, where $k \in \{1, 2, \cdots, 127\}$.

After training, the sequence length complexity of our RL-based compiler scales with $1.905\log^{1.014}\left(1/\left(1-\bar{f}\right)\right)$ with fidelity above 0.9995, as demonstrated in figure 5(b). Here we note that the sequence length is measured by the total number of quantum gates, including both single- and two-qubit gates. This empirically indicates that our RL-based compiler approaches to the optimal compiler under the HRC universal basis set in the two-qubit case. Besides, each row in figure 5(c) stands for the compiling result for a certain target unitary. More specifically, the circuit diagram depicts the complied gates generated by AQ* search guided by the trained DQN. The heatmap shows the fidelity $U_n^\dagger U$. In particular, it indicates that when $\alpha = 1$, we have $M_1(U_n, U) = 0.0869$, $M_2(U_n, U) = 0.9996$, when $\alpha = 13$, we obtain $M_1(U_n, U) = 0.0549$, $M_2(U_n, U) = 0.9998$, when $\alpha = 26$, we obtain $M_1(U_n, U) = 0.0694$, $M_2(U_n, U) = 0.9997$, when $\alpha = 44$, we obtain $M_1(U_n, U) = 0.0559$, $M_2(U_n, U) = 0.9998$. In figure 5(d), we present the results of compiling general unitaries from $\{e^{-i\frac{\alpha}{2}(X\otimes Z)} | \alpha = k \times \frac{\pi}{128}\}$. These results verify the efficiency of the HRC universal basis set in compiling multi-qubit operators.

4.4. Inverse-free single-qubit universal gates

We next apply our proposal to complete single-qubit operator compilation using an inverse-free diffusive basis set. The exploited basis set [71] is composed of the gates $\mathcal{A} = \{\widehat{A}, \widehat{B}\}$. During the training stage, the setup of our protocol is identical to those introduced in the inverse-closed scenario. The simulation results are demonstrated in figure 6. The sequence length complexity of our RL-based compiler scales with $2.683\log^{0.974}\left(1/\left(1-\bar{f}\right)\right)$ and the fidelity $\overline{M_2}$ is above 0.9987.

The achieved results also address a long-standing problem in the inverse-free S–K theorem, which is designing an optimal compiling algorithm with the lowest exponent c. Recall that under the same diffusive set presented above, the most advanced inverse-free quantum compiling algorithm generates sequence with length complexity $c = \log 3/\log 2$ [71]. According to the achieved simulation results, our protocol allows a lower exponent c than this deterministic solution. A recent study proposed an inverse-free S–K theorem [34]. That is, in the single-qubit situation, the sequence length for the inverse-free universal basis sets scales with $O(\log^c(1/\varepsilon))$ with $c = 8.62$. In conjunction with the achieved results and the conclusion of [34], our proposal provides certain empirical evidence for the existence of a more efficient inverse-free S–K theorem.

4.5. Supplemented single-qubit experiments

In section 4.2, we assessed our proposal's performance using the fidelity metric M_2. However, to provide a comprehensive evaluation, we now explore its performance under different metrics. Notably, prior studies have utilized the F-norm [58] and spectral norm [34] as alternative metrics to quantify the dissimilarity between U_n and U. To facilitate a more accurate comparison with these studies, here we employ the F-norm

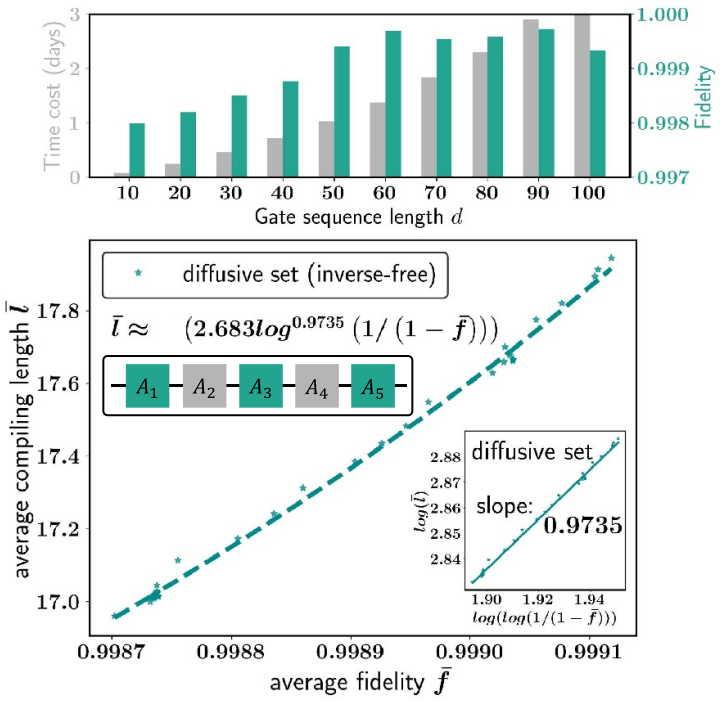


Figure 6. The performance of our RL-based quantum compiler on inverse-free universal basis set. (a) The light-colored portion of the bar chart records the time cost of the DQN expanded by the diffusive universal basis set with gate sequence length d; and the dark part records the compiling fidelity M_2 induced by the DQN with gate sequence length d. (b) Every data point is obtained by averaging the length of the approximating sequence generated by AQ* search using a validation set of 10^3 unitary targets. We check the linearity between $\log(\log(1/\varepsilon))$ and $\log(\bar{l})$ in the subplot at the bottom right. We plot the output length complexity of the diffusive universal basis set as a function of average fidelity $\bar{f}$.

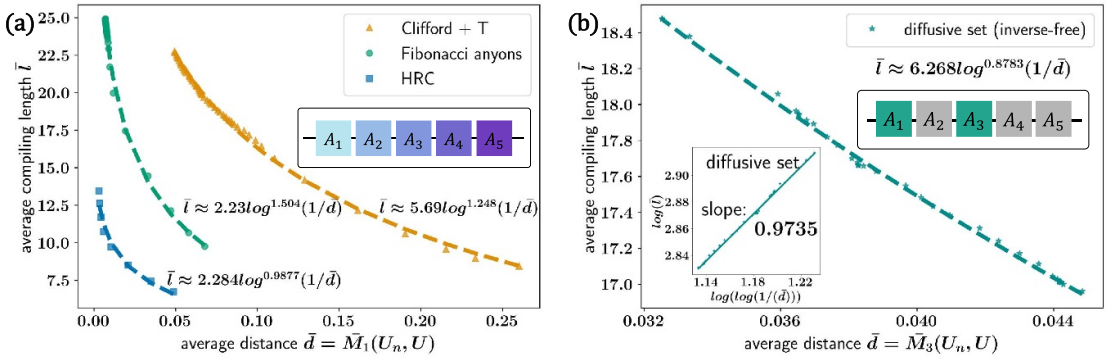


Figure 7. Comparison between different single-qubit universal basis sets under the metric of (a) F-norm and (b) spectral norm.

M_1 and the spectral norm M_3 to evaluate our proposal's performance. The structure of the DQN and the hyperparameter settings are not changed in this setting. We set the accuracy threshold in equation (1) as $\varepsilon = 10^{-3}$ and the threshold of the MSE loss in equation (6) as $\delta = 10^{-2}$. The gate sequence length d in figure 2(a) varies from 3 to 40. Every data point is obtained by averaging the length of the compiling sequence generated by AQ* search using a validation set containing 10^3 unseen unitary targets. The achieved sequence length complexity of our proposal via extrapolation under the metric of F-norm M_1 is shown in figure 7(a). With setting $\varepsilon = \overline{M_1}$, the sequence length complexity of our RL-based quantum compiler scales with $2.284\log^{0.9877}(1/\varepsilon)$ for the HRC gates, $2.23\log^{1.504}(1/\varepsilon)$ for the Fibonacci anyons, and $5.69\log^{1.248}(1/\varepsilon)$ for the *Clifford+T* group, respectively. These results validate that our proposal is robust under different distance metrics $d(U_n, U)$. For the diffusive gates set, the achieved sequence length complexity of our proposal under the metric spectral norm M_3 is plotted in figure 7(b). When $\varepsilon = \overline{M_3}$, the sequence length complexity of our RL-based quantum compiler scales with $6.268\log^{0.8783}(1/\varepsilon)$. Our protocol allows a lower exponent $c = 0.8783$ than $c = 8.62$ in [34] under same metric, providing concrete empirical evidence in pursuing a

more advanced inverse-free S–K theorem. In a summary, above results indicate that the choice of distance metric will not have a significant impact on the performance of our proposal.

4.6. More details of AQ* search

Algorithm 1. AQ* Search.

Require: starting state s_0, deep Q-network $Q(\theta)$
1. OPEN ⟵ priority queue
2. CLOSED ⟵ dictionary that maps nodes to path costs
3. $n_0 = \text{NODE}(s = s_0, g = 0)$ ▷ $n_0 \in$CLOSED
4. $a_0 = \arg\max_{a' \in \mathcal{A}} Q(s_0, a'; \theta)$
5. $v_0 = Q(s_0, a_0; \theta)$
6. Push (s_0, a_0) to OPEN with reward $0 + v_0$ ▷ $(s_0, a_0) \in$OPEN
7. **while** not IS_EMPTY(OPEN) **do**
8. $(s, a) = \text{POP(OPEN)}$
9. $s' = A(s, a)$
10. **if** IS_GOAL(s') **then**
11. **return** PATH_TO_GOAL(s, a)
12. $g' = n.g + r^a(s, s')$
13. $n' =$NODE($s = s', g = g'$)
14. **if** n' not in CLOSED or $g' <$CLOSED$[n']$ **then**
15. CLOSED$[n'] = g'$
16. **for** $a' \in \mathcal{A}$ **do**
17. $q' = V_\theta(s', a')$
18. $v' = g' + q'$
19. Push (s', a') to OPEN with reward v'
20. **return** Failure

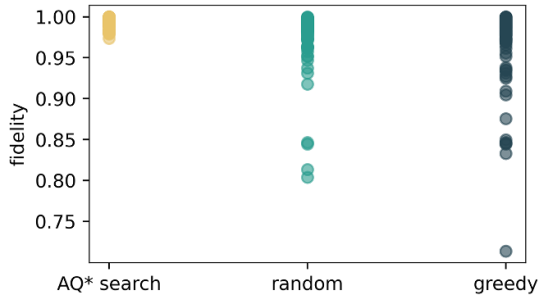


Figure 8. Comparison the performance of AQ* search with other methods

The Pseudo code of AQ* search is summarized in Algorithm 1. Unlike conventional approaches that simply normalize the learned deep Q-network to derive the policy, AQ* search facilitates the generation of the optimal trajectory by combining the DQN with search methods, which is widely used in high-dimensional complex control problems (e.g. Go [62], Rubik's cube [87]). To exhibit the power of AQ* search, in the following, we compare the performance of AQ* search with two typical conventional policies. The first one is a random policy, $\pi_g(a \mid s) = \text{argmax}_a Q(s, a; \theta)$, which constantly performs the action that is believed to yield the highest expected reward. The second one is a Boltzmann policy, i.e. $\pi_b(a \mid s) = \frac{e^{-Q(s,a;\theta)/(kT)}}{\sum_{j=1}^{|\mathcal{A}|} e^{-Q(s,a_j;\theta)/(kT)}}$, where k is the Boltzmann constant, T is the temperature.

We compare the efficiency of different inference strategies under single-qubit *Clifford*+T universal basis set. The hyper-parameters settings are as follows. All the inference algorithms are guided by the learned DQN with gate sequence length $d = 40$. The random strategy is realized by using the Boltzmann distribution with setting $T = 1/k$. We sample 100 single-qubit unseen unitaries and employ the above three inference algorithms to build the compiling strings.

The simulation results in the measure of fidelity are illustrated in figure 8. The average fidelity scales with 0.9955 for the AQ* search, 0.9784 for the Boltzmann distribution strategy, 0.9708 for the random strategy. From the perspective of robustness, AQ* search has less variance and thus is more stable. These results validate that AQ* search is a more powerful inference strategy compared with conventional inference strategies.

5. Conclusion and discussion

We have proposed an efficient and practical quantum compiler for multi-qubit operators. Attributed to the power of deep RL models and heuristic search, our proposal is compatible with various universal basis sets and quantum platforms. Under the inverse-closed setting, the achieved empirical results imply that our proposal is near-optimal in compiling single-qubit and two-qubit operators, supported by the S–K theorem. In addition, we demonstrate how to use an RL-based compiler to compile multi-qubit operators without additional constraints. Under the inverse-free setting, our proposal outperforms the current state-of-the-art method. This provides concrete empirical evidence for pursuing a more advanced inverse-free S–K theorem. Furthermore, our numerical results indicate that the omitted constant term within the compiling sequence length complexity significantly impacts the performance assessment of universal basis sets, which is ignored by most theoretical studies. Consequently, our proposal can not only accelerate the realization of large-scale quantum algorithms and quantum supremacy but also facilitate theoretical understanding of the capabilities and limitations of quantum compiling.

However, while our method can be applied to noisy settings or real quantum devices in principle, its cost and performance in such cases remain unexplored. Specifically, its ability to approach the S–K lower bound in noisy environments is an open question, both theoretically and practically. Although our main focus in this study is utilizing RL to design efficient quantum compilers, the concept behind our proposal can be generalized to other crucial quantum learning tasks from pulse and circuits to the logical level. In particular, at the pulse level, RL techniques can be used to solve quantum control problems [88–90]; at the circuit level, RL methods have potential to accelerate Hamiltonian simulation [91]; at the logical level, RL approaches contribute to execute complicated circuits on near-term quantum machines by reducing the number of gates [92–94]. For a specified quantum learning task listed above, how to leverage the prior knowledge to design a powerful RL model to attain a better performance deserves to be further explored.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. Numerical calculations were performed on the supercomputing system of the Supercomputing Center at Wuhan University. This work is supported by the National Key Research and Development Program of China (Grant No. 2023YFA1000103), the National Nature Science Foundation of China (Grants No. 12371424 and No. 12371441), the 'Fundamental Research Funds for the Central Universities', and the research fund of KLATASDSMOE of China. This work was supported by JD.com through the Research Intern Program, and conducted when Q C was a research intern at JD Explore Academy.

ORCID iDs

Qiuhao Chen https://orcid.org/0009-0001-3795-4380
Yuxuan Du https://orcid.org/0000-0002-1193-9756
Xiliang Lu https://orcid.org/0000-0002-7592-5994
Xingyao Wu https://orcid.org/0009-0008-2018-0367
Qi Zhao https://orcid.org/0000-0002-8091-0682

References

[1] Wang H *et al* 2017 High-efficiency multiphoton boson sampling *Nat. Photon.* **11** 361–5
[2] Wang H *et al* 2018 Toward scalable boson sampling with photon loss *Phys. Rev. Lett.* **120** 230502
[3] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[4] Gong M *et al* 2021 Quantum walks on a programmable two-dimensional 62-qubit superconducting processor *Science* **372** 948–52
[5] Yulin W *et al* 2021 Strong quantum computational advantage using a superconducting quantum processor *Phys. Rev. Lett.* **127** 180501
[6] Sun H, Yang B, Wang H-Y, Zhou Z-Y, Guo-Xian S, Dai H-N, Yuan Z-S and Pan J-W 2021 Realization of a bosonic antiferromagnet *Nat. Phys.* **17** 990–4
[7] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
[8] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3
[9] Yuxuan D, Zhuozhuo T, Xiao Y and Dacheng T 2022 Efficient measure for the expressivity of variational quantum algorithms *Phys. Rev. Lett.* **128** 080506

[10] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 Power of data in quantum machine learning *Nat. Commun.* **12** 2631
[11] Huang H-L *et al* 2021 Experimental quantum generative adversarial networks for image generation *Phys. Rev. Appl.* **16** 024051
[12] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503
[13] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504
[14] Wang X, Yuxuan D, Luo Y and Tao D 2021 Towards understanding the power of quantum kernels in the NISQ era *Quantum* **5** 531
[15] Situ H, Zhimin H, Wang Y, Lvzhou Li and Zheng S 2020 Quantum generative adversarial network for generating discrete distribution *Inf. Sci.* **538** 193–208
[16] Devoret M H and Schoelkopf R J 2013 Superconducting circuits for quantum information: an outlook *Science* **339** 1169–74
[17] Barends R *et al* 2014 Superconducting quantum circuits at the surface code threshold for fault tolerance *Nature* **508** 500–3
[18] Yin X-F *et al* 2022 Efficient bipartite entanglement detection scheme with a quantum adversarial solver *Phys. Rev. Lett.* **128** 110501
[19] Yuxuan D and Tao D 2021 On exploring practical potentials of quantum auto-encoder with advantages (arXiv:2106.15432)
[20] Gur T, Hsieh M-H and Subramanian S 2021 Sublinear quantum algorithms for estimating von Neumann entropy (arXiv:2111.11139)
[21] Dian W *et al* 2021 Robust self-testing of multiparticle entanglement *Phys. Rev. Lett.* **127** 230503
[22] Lloyd S 1996 Universal quantum simulators *Science* **273** 1073–8
[23] Berry D W, Childs A M, Cleve R, Kothari R and Somma R D 2015 Simulating Hamiltonian dynamics with a truncated Taylor series *Phys. Rev. Lett.* **114** 090502
[24] Georgescu I M, Ashhab S and Nori F 2014 Quantum simulation *Rev. Mod. Phys.* **86** 153–85
[25] O'Malley P J J *et al* 2016 Scalable quantum simulation of molecular energies *Phys. Rev.* X **6** 031007
[26] Kokail C *et al* 2019 Self-verifying variational quantum simulation of lattice models *Nature* **569** 355–60
[27] Yuan X, Endo S, Zhao Q, Ying Li and Benjamin S C 2019 Theory of variational quantum simulation *Quantum* **3** 191
[28] Endo S, Sun J, Ying Li, Benjamin S C and Yuan X 2020 Variational quantum simulation of general processes *Phys. Rev. Lett.* **125** 010501
[29] Hao Low G and Chuang I L 2019 Hamiltonian simulation by qubitization *Quantum* **3** 163
[30] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* 10th Anniversary edn (Cambridge University Press)
[31] Yu Kitaev A 1997 Quantum computations: algorithms and error correction *Russ. Math. Surv.* **52** 1191–249
[32] Dawson C M and Nielsen M A 2006 The Solovay-Kitaev algorithm *Quantum Inf. Comput.* **6** 81–95
[33] Harrow A W, Recht B and Chuang I L 2002 Efficient discrete approximations of quantum gates *J. Math. Phys.* **43** 4445–51
[34] Bouland A and Giurgica-Tiron T 2021 Efficient universal quantum compilation: an inverse-free Solovay-Kitaev algorithm (arXiv:2112.02040)
[35] Kliuchnikov V, Maslov D and Mosca M 2013 Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates *Quantum Inf. Comput.* **13** 607–30
[36] Kliuchnikov V, Maslov D and Mosca M 2013 Asymptotically optimal approximation of single qubit unitaries by Clifford and *T* circuits using a constant number of ancillary qubits *Phys. Rev. Lett.* **110** 190502
[37] Neil J R 2015 Optimal ancilla-free Clifford+V approximation of z-rotations *Quantum Inf. Comput.* **15** 932–50
[38] Kliuchnikov V, Maslov D and Mosca M 2016 Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits *IEEE Trans. Comput.* **65** 161–72
[39] Selinger P 2015 Efficient Clifford+T approximation of single-qubit operators *Quantum Inf. Comput.* **15** 159–80
[40] Cody Jones N, Whitfield J D, McMahon P L, Yung M-H, Van Meter R, Aspuru-Guzik A and Yamamoto Y 2012 Faster quantum chemistry simulation on fault-tolerant quantum computers *New J. Phys.* **14** 115023
[41] Wiebe N and Kliuchnikov V 2013 Floating point representations in quantum circuit synthesis *New J. Phys.* **15** 093041
[42] Jones C 2014 Distillation protocols for Fourier states in quantum computing *Quantum Inf. Comput.* **14** 560–76
[43] Duclos-Cianci G and Svore K M 2013 Distillation of nonstabilizer states for universal quantum computation *Phys. Rev.* A **88** 042325
[44] Bocharov A, Gurevich Y and Svore K M 2013 Efficient decomposition of single-qubit gates into *v* basis circuits *Phys. Rev.* A **88** 012313
[45] Bocharov A, Roetteler M and Svore K M 2015 Efficient synthesis of probabilistic quantum circuits with fallback *Phys. Rev.* A **91** 052317
[46] Bocharov A, Roetteler M and Svore K M 2015 Efficient synthesis of universal repeat-until-success quantum circuits *Phys. Rev. Lett.* **114** 080502
[47] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79
[48] Venturelli D, Minh D, Rieffel E and Frank J 2018 Compiling quantum circuits to realistic hardware architectures using temporal planners *Quantum Sci. Technol.* **3** 025004
[49] Booth K, Do M, Beck J, Rieffel E, Venturelli D and Frank J 2018 Comparing and integrating constraint programming and temporal planning for quantum circuit compilation *Proc. Int. Conf. on Automated Planning and Scheduling* vol 28 pp 366–74
[50] Khatri S, LaRose R, Poremba A, Cincio L, Sornborger A T and Coles P J 2019 Quantum-assisted quantum compiling *Quantum* **3** 140
[51] Rakyta P and Zimborás Z 2022 Efficient quantum gate decomposition via adaptive circuit compression (arXiv:2203.04426)
[52] Peres F C R and Galvão E F 2022 Quantum circuit compilation and hybrid computation using Pauli-based computation (arXiv:2203.01789)
[53] Huang H-L, Xiao-Yue X, Guo C, Tian G, Wei S-J, Sun X, Bao W-S and Long G-L 2023 Near-term quantum computing techniques: variational quantum algorithms, error mitigation, circuit compilation, benchmarking and classical simulation *Sci. China Phys. Mech. Astron.* **66** 250302
[54] Sharma K, Sumeet Khatri M C and Coles P J 2020 Noise resilience of variational quantum compiling *New J. Phys.* **22** 043006
[55] Mizuta K, Nakagawa Y O, Mitarai K and Fujii K 2022 Local variational quantum compilation of large-scale Hamiltonian dynamics *PRX Quantum* **3** 040302
[56] Zhimin H, Lvzhou Li, Zheng S, Yongyao Li and Situ H 2021 Variational quantum compiling with double Q-learning *New J. Phys.* **23** 033002
[57] Ying L, Zhou P-F, Fei S-M and Ran S-J 2023 Quantum compiling with a variational instruction set for accurate and fast quantum computing *Phys. Rev. Res.* **5** 023096
[58] Zhang Y-H, Zheng P-L, Zhang Y and Deng D-L 2020 Topological quantum compiling with reinforcement learning *Phys. Rev. Lett.* **125** 170501

[59] Moro L, Paris M G A, Restelli M and Prati E 2021 Quantum compiling by deep reinforcement learning *Commun. Phys.* **4** 178
[60] Jordan M I and Mitchell T M 2015 Machine learning: trends, perspectives and prospects *Science* **349** 255–60
[61] Bengio Y, Lecun Y and Hinton G 2021 Deep learning for AI *Commun. ACM* **64** 58–65
[62] Silver D *et al* 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
[63] Silver D *et al* 2018 A general reinforcement learning algorithm that masters chess, shogi and go through self-play *Science* **362** 1140–4
[64] Fawzi A *et al* 2022 Discovering faster matrix multiplication algorithms with reinforcement learning *Nature* **610** 47–53
[65] Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33
[66] Bellman R 1957 *Dynamic Programming* (Dover Publications)
[67] Puterman M L and Chirl Shin M 1978 Modified policy iteration algorithms for discounted Markov decision problems *Manage. Sci.* **24** 1127–37
[68] Agostinelli F, Shmakov A, McAleer S, Fox R and Baldi P 2023 A* search without expansions: learning heuristic functions with deep Q-networks (arXiv:2102.04518)
[69] Hart P E, Nilsson N J and Raphael B 1968 A formal basis for the heuristic determination of minimum cost paths *IEEE Trans. Syst. Sci. Cybern.* **4** 100–7
[70] Bonet B and Geffner H 2001 Planning as heuristic search *Artif. Intell.* **129** 5–33
[71] Zhiyenbayev Y, Akulin V M and Mandilara A 2018 Quantum compiling with diffusive sets of gates *Phys. Rev.* A **98** 012325
[72] Du Q H 2009 Metrics for 3D rotations: comparison and analysis *J. Math. Imaging Vis.* **35** 155–64
[73] Aharonov D, Kitaev A and Nisan N 1998 Quantum circuits with mixed states *Proc. 13th Annual ACM Symp. on Theory of Computing (STOC'98)* (Association for Computing Machinery) pp 20–30
[74] Patel T, Younis E, Iancu C, de Jong W and Tiwari D 2021 Robust and resource-efficient quantum circuit approximation (arXiv:2108.12714)
[75] Pack Kaelbling L, Littman M L and Moore A W 1996 Reinforcement learning: a survey *J. Artif. Intell. Res.* **4** 237–85
[76] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* (A Bradford Book)
[77] Sohaib Alam M, Berthusen N F and Orth P P 2022 Quantum logic gate synthesis as a Markov decision process *npj Quantum Inf.* **9** 108
[78] Riedmiller M, Hafner R, Lampe T, Neunert M, Degrave J, van de Wiele T, Mnih V, Heess N and Tobias Springenberg J 2018 Learning by playing solving sparse reward tasks from scratch *Proc. 35th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research* vol 80) ed J Dy and A Krause (PMLR) pp 4344–53
[79] Gradl T, Spörl A, Huckle T, Glaser S J and Schulte-Herbrüggen T 2006 Parallelising matrix operations on clusters for an optimal control-based quantum compiler *Euro-Par 2006 Parallel Processing* ed W E Nagel, W V Walter and W Lehner (Springer) pp 751–62
[80] Nayak C, Simon S H, Stern A, Freedman M and Das Sarma S 2008 Non-Abelian anyons and topological quantum computation *Rev. Mod. Phys.* **80** 1083–159
[81] Kitaev A 2006 Anyons in an exactly solved model and beyond *Ann. Phys., NY* **321** 2–111
[82] Freedman M H, Larsen M and Wang Z 2002 A modular functor which is universal for quantum computation *Commun. Math. Phys.* **227** 605–22
[83] Giles B and Selinger P 2013 Exact synthesis of multiqubit Clifford+T circuits *Phys. Rev.* A **87** 032332
[84] Kingma D and Jimmy B 2015 Adam: a method for stochastic optimization *Int. Conf. on Learning Representations (ICLR)*
[85] Watrous J 2009 Semidefinite programs for completely bounded norms *Theory Comput.* **5** 217–38
[86] Vidal G and Dawson C M 2004 Universal quantum circuit for two-qubit transformations with three controlled-not gates *Phys. Rev.* A **69** 010301
[87] Agostinelli F, McAleer S, Shmakov A and Baldi P 2019 Solving the Rubik's cube with deep reinforcement learning and search *Nat. Mach. Intell.* **1** 356–63
[88] Magann A B, Arenz C, Grace M D, Tak-San H, Kosut R L, McClean J R, Rabitz H A and Sarovar M 2021 From pulses to circuits and back again: a quantum optimal control perspective on variational quantum algorithms *PRX Quantum* **2** 010101
[89] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev.* X **8** 031086
[90] Yuezhen Niu M, Boixo S, Smelyanskiy V N and Neven H 2019 Universal quantum control through deep reinforcement learning *npj Quantum Inf.* **5** 33
[91] Bolens A and Heyl M 2021 Reinforcement learning for digital quantum simulation *Phys. Rev. Lett.* **127** 110502
[92] Fösel T, Yuezhen Niu M, Marquardt F and Li L 2021 Quantum circuit optimization with deep reinforcement learning (arXiv:2103.07585)
[93] Yuxuan D, Huang T, You S, Hsieh M-H and Tao D 2022 Quantum circuit architecture search for variational quantum algorithms *npj Quantum Inf.* **8** 62
[94] Kuo E-J, Fang Y-L L and Chen S Y C 2021 Quantum architecture search via deep reinforcement learning (arXiv:2104.07715)