Modularized and Scalable Compilation for Double Quantum Dot Quantum Computing

Run-Hong He[1,4], Xu-Sheng Xu[2*] , Mark S. Byrd[3] and Zhao-Ming Wang[1†]

1.College of Physics and Optoelectronic Engineering, Ocean University of China, Qingdao 266100, China

2.Department of Physics, State Key Laboratory of Low-Dimensional Quantum Physics, Tsinghua University, Beijing 100084, China

3.Department of Physics, Southern Illinois University, Carbondale, Illinois 62901-4401, USA

4.State Key Laboratory of Computer Science, Institute of Software Chinese Academy of Science, BeiJing 101408, China

E-mail: *thuxuxs@163.com †wangzhaoming@ouc.edu.cn

March 2023

Abstract. Any quantum program on a realistic quantum device must be compiled into an executable form while taking into account the underlying hardware constraints. Stringent restrictions on architecture and control imposed by physical platforms make this very challenging. In this paper, based on the quantum variational algorithm, we propose a novel scheme to train an Ansatz circuit and realize high-fidelity compilation of a set of universal quantum gates for singlet-triplet qubits in semiconductor double quantum dots, a fairly heavily constrained system. Furthermore, we propose a scalable architecture for a modular implementation of quantum programs in this constrained systems and validate its performance with two representative demonstrations, the Grover's algorithm for the database searching (static compilation) and a variant of variational quantum eigensolver for the Max-Cut optimization (dynamic compilation). Our methods are potentially applicable to a wide range of physical devices. This work constitutes an important stepping-stone for exploiting the potential for advanced and complicated quantum algorithms on near-term devices.

Keywords: Semiconductor double quantum dots, Quantum gate compilation, Quantum program compilation, Variational quantum algorithm

1. INTRODUCTION

Using superposition and entanglement, quantum computers could admit superpolynomial or even exponential speedup over their classical counterparts when solving certain important and otherwise intractable problems [1, 2]. In the race to construct quantum computing prototypes, a wide range of physical models have been proposed

and experimentally demonstrated over the past decades, including trapped ions [3], ultracold atoms [4], nitrogen-vacancy centers [5], superconducting circuits [6], optical system [7] and semiconductor quantum dots [8, 9], etc. In the past, using electron spins in solid-state systems, semiconductor quantum dots have been favored due to the prospective scalability and compatibility with existing mature semiconductor manufacturing techniques [10, 11, 12, 13]. By leveraging the degrees of spin and charge of electrons trapped in quantum dots, various qubit modalities have been suggested and realized in experiments, such as the single-electron spin-1/2 qubit [12], two-electron singlet-triplet (S-T_0) qubit in double quantum dots (DQDs) [14], three-electron exchange-only qubit [15] and hybrid qubit [16]. Among them, the S-T_0 qubit has attracted considerable interest because it can be manipulated all electrically and provides rapid gates in sub-nanosecond time scales, which is fast enough to enable 10^4 gates before the decoherence takes over [14].

Any high-level algorithm for quantum computation needs to be translated into low-level instructions that can be executed step by step on specific quantum hardware [17, 18]. While the S-T_0 qubit is one of the most promising modalities for constructing a scalable quantum computer from the fabrication perspective, the tight restrictions on the control limits the applications of many existing optimization methods and make it challenging to manipulate this quantum system for computational tasks [19, 20]. The primary hurdles arise in physical details that **1)** one could only precisely control the rotation rate around the $+z$-axis of a single-qubit state on the Bloch sphere, while the rotation rate around the x-axis is constant and non-zero; **2)** the undesired coupling introduced by simultaneous operations on adjacent qubits would alter these intended individual operations. To date, an enormous amount of ingenuity and effort has been devoted to compiling required operations into sequential native gates on this platform, such as the so-called SUPCODE [21, 22, 20] for generating robust quantum gates and fast geometric gates [23] for cancelling out the accompanied dynamical phase during evolution. In this paper a native gate refers to an operation that can be readily implemented in DQDs with a single (for single-qubit operation) or pair (for two-qubit operation) of pulses. Typically, to perform arbitrary rotation gates, it is necessary to iteratively solve a set of nonlinear equations to determine the appropriate composite pulse sequence [20, 24]. This is a resource-expensive and time-demanding task in practice. A significant surge of interest has recently been focused on nascent tools that are aimed at improving the efficiency of tailoring pulse sequences. For examples, by employing the machine learning [25], Ref. [24] studies how to directly predict the required pulse sequence with a well-trained neural network; Ref. [19] promises to dynamically steer a specific quantum state to another; Refs. [26, 27] could reset an arbitrary quantum state to a target one, and Ref. [28] shows how to prepare an arbitrary quantum state from $|0\rangle$. Furthermore, high fidelity universal quantum state preparation is also observed in [29] with the revised greedy algorithm. All of these exhibit excellent performance for universal gate sets or few-qubit state preparation. Whereas the lack of an architecture that can properly combine these pulse sequences of varying lengths makes

it quite difficult to perform a practical large-scale computation due to the aforementioned second limitation in DQDs - an unfinished operation on one qubit will interfere with the execution of quantum gates on its neighboring qubits, thus resulting in the absence of scalability. An open problem and important direction of investigation for this special system is therefore to develop an architecture which could layout various native gates such that the system can scalably compile impactful computational tasks. Available traditional optimization approaches that could yield a fixed-length pulse sequence are mainly based on the gradient, such as the stochastic gradient descent [30]. Partly due to the fact that each gradient evaluation requires two forward passes for loss function calculation, their resource consumption grows very quickly as the size of parameter space increases. In addition, their performance is sensitive to the initialization, and as a consequence, the optimization routine may often be stucked in a local optimum and then end with an inadequate fidelity.

In this paper, we enlist another emerging and powerful technique to compile a desired unitary into a native gate sequence without changed length - the variational quantum algorithm (VQA) [31], which has recently been used for decomposing complex unitary transformations into ordered universal gates [32, 33] or Schmidt decomposition [34], etc. Utilizing the adjoint differentiation [31], the VQA permits gradients collection of all parameters with only one forward pass and one recursive backward pass [31] - a significant calculation saving compared to traditional gradient-based methods. Additionally, we suggest training the Ansatz, a circuit of parametric native gates, with random quantum states. Compared to the widely used matrix-based approaches [32, 33, 34], this training scheme offers reduced overhead in loss calculation and smaller possibility of being stucked in local optimums during optimization. Considering single-qubit rotations and entangling two-qubit gates are crucial ingredients for universal quantum computation, a series of them are compiled and high fidelities are reached, underling the foundations for accurately implementing quantum programs in DQDs.

More importantly, inspired by the special stacking style of Mahjong cards, a traditional Chinese game (see Fig. 2(a)), we present a modularized and scalable (MS) compilation architecture that can systematically combine native gate sequences into logic operations to perform advanced and complex reference quantum circuits in DQDs. Furthermore, we demonstrate our MS compilation with two representative quantum computing tasks, i.e., the Grover's algorithm [35, 36] for database search and a variant [37] of variational quantum eigensolver (VQE) [38, 39] for graph Max-Cut optimization [40], and achieve excellent results. Finally, to simulate the performance of compiled quantum programs on actual near-term devices, we study the impact of experimental noises on them and observe promising levels of robustness. These results show that our MS compilation can be a powerful tool used to exploit the potential of DQDs in the current noisy intermediate-scale quantum (NISQ) devices era [41].

2. Model

In the present work, the model of interest is the semiconductor DQDs system, where the S-T_0 qubit is encoded in the collective spin states of two electrons confined in a double-well potential [14, 42]. The Hamiltonian of a single S-T_0 qubit, governed by external electric pulses is [43, 44, 8, 9]

$$H(t) = J(t)\sigma_z + h\sigma_x, \tag{1}$$

in the computational basis $\{|0\rangle \equiv |S\rangle = (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)/\sqrt{2}, |1\rangle \equiv |T_0\rangle = (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/\sqrt{2}\}$. σ_z and σ_x are Pauli matrices and represent rotations of the quantum state around the z- and x-axes with rates $J(t)$ and h, respectively. The coefficient h describes the Zeeman energy splitting between $|S\rangle$ and $|T_0\rangle$, commonly raised by nearby deposited permanent micromagnet [43] and its value is difficult to vary during the quantum gate time in experiment [43, 42] (although tunable splitting by Overhauser field has been observed [14], its time consumption is much longer than a typical gate time). We assume $h = 1$ here to facilitate our theoretical treatment, and take this as the energy unit. In addition, for simplicity, we let the reduced Planck constant $\hbar = 1$ which determines the time-scale throughout. The only effective tunable parameter in this system is the exchange coupling $J(t)$ between two captured electrons and can be rapidly manipulated by applying calibrated voltage pulses to the associated electrodes. Because of the nature of the exchange interaction and to avoid altering the charge configuration of the quantum dots, $J(t)$ is constrained to be non-negative and bounded, i.e., $J_{\max} \geq J(t) \geq 0$, where the maximal value $J_{\max} \sim h$ [21]. In other words, we have only precise control over the rotation rate of the quantum state around the $+z$-axis on Bloch sphere, while the rotation rate around the x-axis is constant and always present.

An experimentally available native gate (one-piece rotation) can be implemented by applying an electrical pulse with specific intensity J for a certain amount of time $\triangle t$, which produces a rotation around the axis $Jz + x$:

$$g(J, \triangle t) = \exp(-i(J\sigma_z + \sigma_x)\triangle t). \tag{2}$$

As a special case, if we set $J = 0$, then $g(0, \triangle t) = \cos(\triangle t) \cdot I + i\sin(\triangle t) \cdot \sigma_x$, where I refers to the identity operator. In this case, the identity gate g_I can be achieved, up to an irrelevant global phase, when $\triangle t = k\pi$, where k are integers. This point will prove valuable and plays the role of a major groundstone for our MS compiling architecture in DQDs as we will discuss in the following section.

In order to generate entanglement between qubits, multi-qubit entangling gates are necessary. In DQDs, the effective Hamiltonian of two adjacent qubits based on Coulomb interaction can be written as [45, 46, 9]

$$\begin{aligned} H_{2\text{–qubit}} = &\frac{\hbar}{2}(J_1(\sigma_z \otimes I) + J_2(I \otimes \sigma_z) + h_1(\sigma_x \otimes I) \\ &+ h_2(I \otimes \sigma_x) + \frac{J_{12}}{2}((\sigma_z - I) \otimes (\sigma_z - I))), \end{aligned} \tag{3}$$

in the basis of $\{|SS\rangle, |ST_0\rangle, |T_0S\rangle, |T_0T_0\rangle\}$. J_i and h_i are exchange interaction and Zeeman splitting respectively, with the subscript $i = 1, 2$ referring to the corresponding qubit. Empirically, the inter-qubit coupling $J_{12} \propto J_1 J_2$ [45] and here we assume $J_{12} = J_1 J_2/2$ for simplicity. As in the case of single S-T_0 qubit, the operation on this two-qubit system requires only pulsed electric fields, i.e., J_1 and J_2. To maintain the coupling between two qubits, $J_i > 0$ during entangling operations. It is important to notice that the coupling term in Eq. (3) also indicates that individual qubit manipulation of a given qubit requires the suspension of the control on its neighboring qubits to avoid undesired inter-qubit coupling. These restrictions should be kept in mind for the effective design of the architecture to run quantum algorithms.

3. Methods

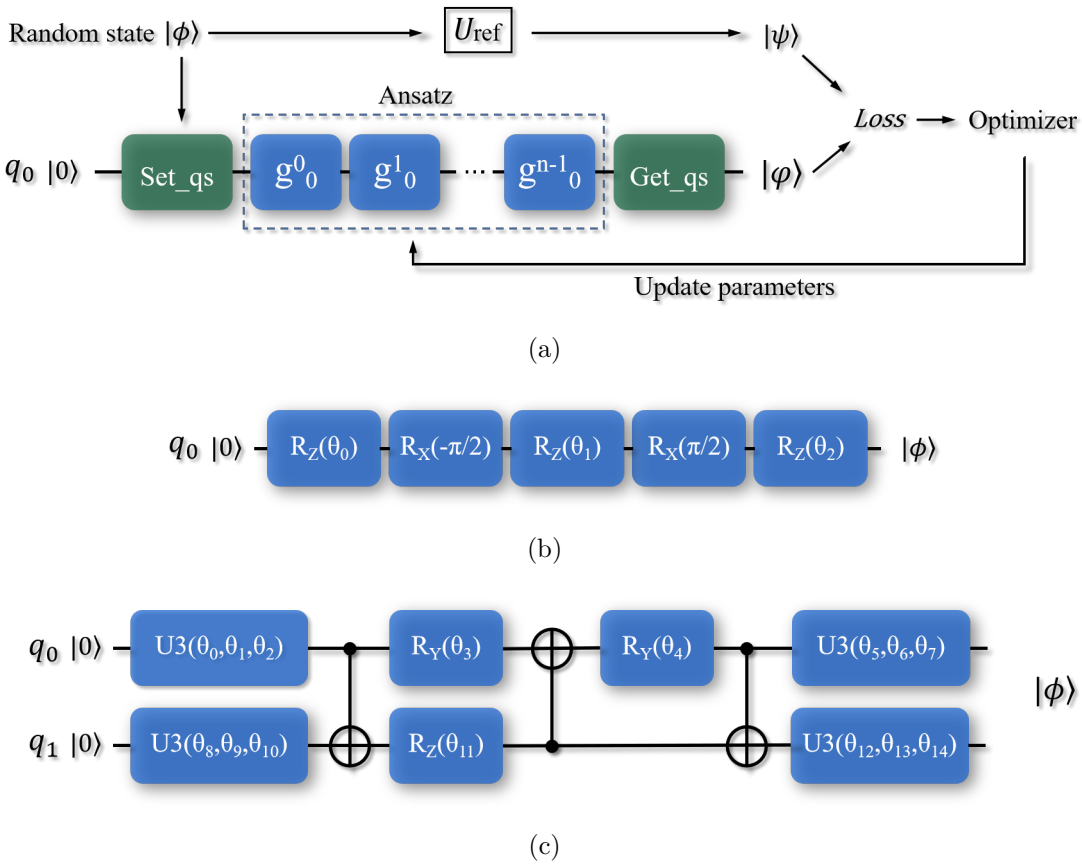


Figure 1: (a) The overall workflow of the compilation of a reference logical single-qubit unitary U_{ref} with an Ansatz consisting of sequential native gates. Each native gate, denoted as a blue block, is carried out by a single electric pulse J with a certain duration acting on the DQD qubit. The details of this workflow are described in text. (b) The so-called U3 circuit [47, 48] used to generate single-qubit random quantum states for training one-qubit Ansatz. (c) The parametric circuit [49] used to generate two-qubit random quantum states for training two-qubit Ansatz.

In the previous section, we presented a typical restricted system, S-T_0 qubits in DQDs and pointed out where the difficulty of gate compiling lies. In this section, we start by explaining how to compile a single-qubit arbitrary gate into sequential DQDs native gates by VQA and next describe an architecture that orchestrates operations on neighbor qubits to avoid unexpected interaction effects. Finally, a structure that enables two-qubit entangling gates is presented. With these elements, advanced quantum algorithms can be implemented on a system consisting of DQDs.

In our work, any reference unitary U_{ref} is compiled into DQDs native gates by leveraging the VQA. Taking the complication of an arbitrary single-qubit gate as an example, the overall workflow is schematically outlined in Fig. 1(a), where the Ansatz is composed of n native gates, denoted as blue blocks with subscripts indicating which qubit is affected. Each native gate is generated by an electric pulse with a certain strength J and duration. The process of achieving an appropriate native gates sequence that is equivalent to the U_{ref} goes as follows:

Step 1: Ansatz's variational parameters (Js) are all ones initialized before optimization and additionally restricted to the non-negative domain throughout the training process, accounting for the realistic physical limitation on non-negative exchange interaction strengths as introduced in the previous sections.

Step 2: Certain number of random quantum states is generated using quantum circuit containing random parameters, and divided into training and validation sets. The single- and two-qubit quantum circuit generating random states in this work are shown in Fig. 1(b) and (c), whose outputs can theoretically cover the entire Hilbert space as the real-valued parameters vary [47, 49].

Step 3: A random quantum state $|\phi\rangle$ sampled from the training set will: **1)** be fed to the Ansatz which then outputs the final state $|\varphi\rangle$ after evolution; and **2)** be acted by the U_{ref} and converted to the target state $|\psi\rangle$.

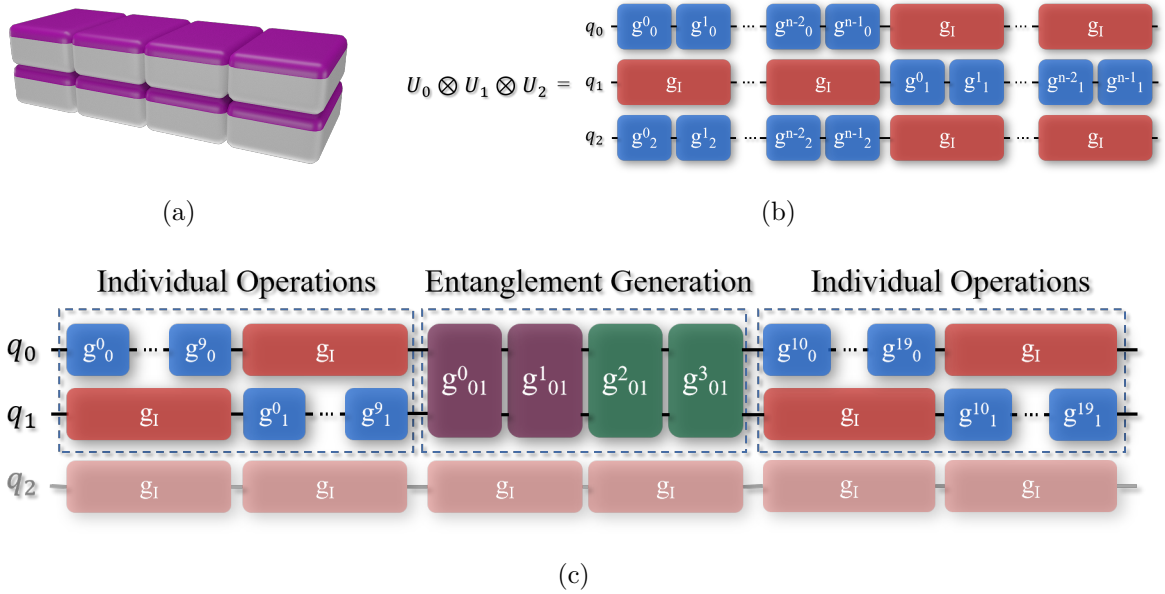
Step 4: Compute the loss function $Loss = -|\langle\psi|\varphi\rangle|^2$, which quantifies the fitness of the Ansatz to the U_{ref}. With the $Loss$ available, the gradient with respect to each Ansatz's parameter can be calculated [31] and then used to update the parameters for improving the performance by an optimizer, such as the Adam [50, 51].

Step 5: Validate all random states in the validation set using a process similar to the step 3 and then take the maximal $1 + Loss$ among them as the current error ϵ of the Ansatz relative to the reference unitary at this training step.

Repeat the sequential steps 3-5 until the error ϵ over the validation set smaller than an acceptability threshold, such as 10^{-5}, where the effect of the Ansatz will be approximately the target U_{ref}.

Compared to schemes such as [32, 33, 34] where Ansatz's parameters are updated in a closed-loop style, the application of random states reduces the susceptibility of the optimization to local optima akin to the usage of random samples in the training landscape of classical neural networks [26]. In addition, this utilization of quantum states is more resource efficient than commonly used approaches based on matrices in loss calculation, such as the Hilbert-Schmidt distance or other customized metrics

between the associated unitaries [32, 33, 34]. It is worth noting that this training approach is scalable and applicable generally to the construction or decomposition of unitary transformations with any number of qubits.



(a) (b) (c)

Figure 2: (a) The vertically stacked mahjong cards which inspired us to explore the MS compilation. (b) Schematic diagram of the compiling architecture formed according to the MS compilation that enables interference-free control between qubits. (c) An Ansatz constructed in accordance with the MS compilation which enables two-qubit entangling gates acting on q_0 and q_1, whose adjacent qubits, e.g., the q_2, remain idle during this phase to avoid undesirable interactions. The Ansatz contains three parts: two Individual Operations and one Entanglement Generation, which consist of only single-parameter native gates.

As for the implementation of individual operations on different qubits, the challenge arises from the fact that if the pulses imposed on adjacent qubits are non-zero simultaneously, these qubits will interact with each other due to capacitive coupling [9], and therefore interrupts intended operations. Inspired by the Mahjong (shown in Fig. 2(a)) whose cards are stacked upright so that no interaction is produced between adjacent columns and so can be operated independently, we propose the following architecture to circumvent this problem.

We make the execution time of all Ansatzes for the associated logical gates to be the same (constant pulse sequence length), and when a certain qubit is operated on, its neighboring qubits are set to be idle (no pulses acting on), so as to avoid interference between qubits. While the idle qubits undergo free evolution all the time due to invariable and non-zero rotation rate h around the x-axis, as long as the gate time is chosen appropriately, these idle qubits will evolve exactly to their original status when the operations on the qubit of interest is completed. In other words, this period of free evolution is equivalent to identity gates acting on the idle qubits. For any two adjacent qubits, there always exists one qubit with no pulse acting on it at any time,

leaving them be free from the undesired coupling. With this principle, operations on adjacent qubits will be made alternately. Fig. 2(b) outlines an example which allows individual arbitrary rotation operations $U_0 \otimes U_1 \otimes U_2$ on three adjacent qubits, subscript $i = 0, 1, 2$ referring to the corresponding qubit that is transformed. The red blocks "g_I" refer to identity gates, due to the zero-strength pulses with $k\pi$ duration, where k is an integer (as explained in Eq. (2)). The duration of g_I here is taken to be twice the typical pulse time (the duration of a parametric native gate, or blue block) for clarity.

To create two-qubit entangling gates in DQDs, e.g., CX and CZ, we likewise empirically design an Ansatz form schematically visualized in Fig. 2(c), where the entangling gate acts on q_0 and q_1, while other adjacent qubits, e.g., the q_2, are left idle to avoid interference. This entanglement gate consists of three parts: one Entanglement Generation sandwiched between two Individual Operations. The part of Entanglement Generation consists of 4 sequential native 2-qubit gates (cross-qubit purple and green blocks) for generating enough entanglement between two target qubits. Considering the fact that a two-qubit native gate is brought about by two simultaneous pulses, which impose a challenge in the optimization using VQA, we fix one of them to be 1 and leave the other variable. For the first two two-qubit native gates (purple blocks) we fix their parameter on q_0, while the last two (green blocks) on q_1. The two Individual Operations parts located before and after the Entanglement Generation are used to apply additional rotations to each qubit to achieve fine tuning for an overall improved performance. *This architecture is extremely important as it allows programs implemented on this platform to not only precisely execute quantum gates but also reliably implement quantum algorithms.*

4. numerical Results

4.1. Compilation for universal gates

A set of universal quantum gates is necessary for the implementation of quantum programs [1, 2]. In this subsection, we first study the applicability of the MS compilation for arbitrary single-qubit gates and then move our attention to enacting common two-qubit entangling quantum gates. The details of the gate compiling process are shown in the Section 3. We stress that, for conciseness, the control pulses are assumed to be piecewise-constant and so are turned on and off instantaneously. In real experiments, pulses possessing finite rise times will result in only a minimally alteration in the pulse parameters yielded by our work but will not alter the conclusions as clearly explained in Ref. [21].

We use an Ansatz circuit to approximate the reference gate and its structure is crucial for the final result. The performance of the Ansatzes for same 32 random reference quantum gates versus the number of employed native gates is plotted in Fig. 3, whose caption describes the details, such as pulse duration and learning rate. Fig. 3(a) describes the distribution of Ansatzes' final errors compared to the associated reference

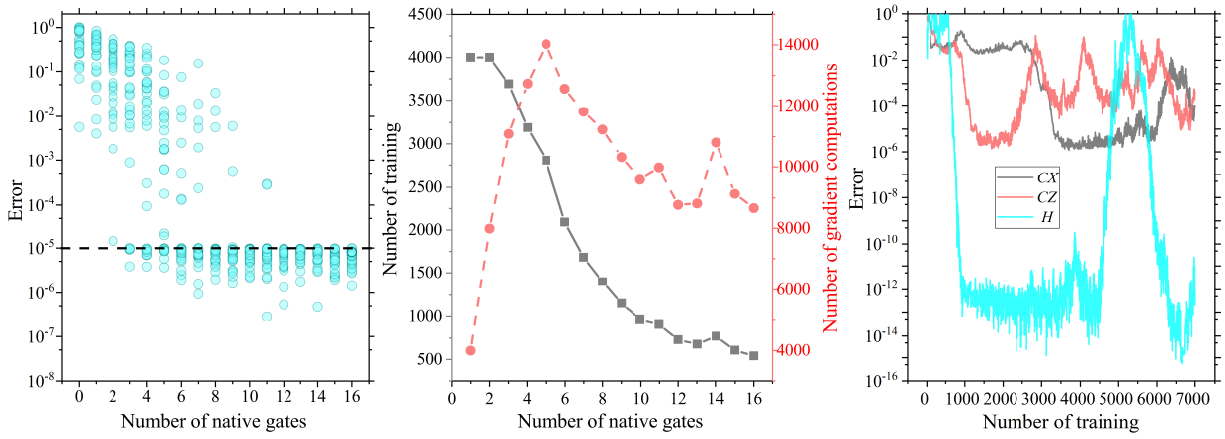


Figure 3: The performance of quantum gates compilation with the Ansatz structures depicted in Fig. 2. (a) The distributions of compiled errors for 32 single-qubit random reference unitaries vary with the number of employed native gates. The optimization ends when the error reaches the expected threshold 10^{-5} (the dashed line) or the number of training rounds exceeds 4000. (b) The average numbers of training and gradient computations to end the optimization for 32 single-qubit random reference unitaries compilation as functions of the native gate count involved in the Ansatz. (c) The compilation errors of gates CX, CZ and H vary with the number of training steps. The duration of native gates is $\pi/2$ in single-qubit quantum gates compilation, while $\pi/10$ for single-qubit operations and $\pi/2$ for two-qubit operations in two-qubit quantum gates compilation. The optimizer used here is the Adam [50] with learning rate 0.05 for single-qubit gates and 0.01 for two-qubits gates compilation, respectively. All parameters are ones initialized before training and are kept in the non-negative domain during the optimization to take into account experimental constraints.

quantum gates and varies with the number of native gates, where the black dashed line is a predefined error threshold, 10^{-5}. When the validating error ϵ reaches this expected threshold (compilation succeeded) or the number of training exceeds 4000 (compilation failed), the optimization ends. Meanwhile, we stress that despite using random quantum states as training samples can offer broad coverage in parameter space, it does not guarantee coverage of all possible quantum states with a finite-sized training set, which may lead to compilation failures or reduced accuracy for certain target gates.

From the evaluation results on sampled reference gates showed in Fig. 3(a), we see that once the native gate count drops below 10, there always exists points that cannot reach the error threshold, and this phenomenon mainly comes from the limited Ansatz space due to the fact that an insufficient number of native gates cannot produce the intended reference unitary. Although two bad points (which overlap because of similar errors) are present when the native gate number is 11, they can also achieve the error requirement after 630 and 292 rounds of training respectively by changing the learning rate from 0.05 to 0.1. When the involved native gate count reaches 12 or more, arbitrary single-qubit operations can be implemented with a fair amount of confidence.

Fig. 3(b) depicts the averaged number of training and gradient computations for reaching the termination condition of optimization in terms of the applied native gate count. The required number of training rounds gradually drops off as the native gate count increases, implying that a deeper Ansatz circuit is more adequate for approximating the reference unitary. And this trend is well maintained until the native gate count reaches 12. In addition, the number of gradient computations, which reflects the computational overhead of the training process, also reaches the minimal value for the native gate number of 12. Considering more native gates would impose additional overhead for optimization and experimental implementation, we believe that under the existing parameter setting an Ansatz composed of 12 native gates is a good trade-off between compilation effectiveness and consumed overhead for arbitrary single-qubit operations. We emphasize that the above results are obtained under the assumption that the duration of the native gate is $\pi/2$. Other settings, such as using different pulse durations, selected error thresholds, classical optimizers and initial parameters may deliver varying compilation results and performance, but the final conclusion will be similar that the ability to accurately compile arbitrary quantum gates is afforded by a sufficient number of native gates. A further study would be the more advanced setting or method to minimize the number of native gates. Other alternative methods, such as the stochastic gradient descent [30] algorithm, can also be used to perform certain quantum gate compilation tasks, although lacking scalability as mentioned in the introduction, and may yield different optimal number of native gates. Ref. [19] presents a detailed performance comparison between them in solving relevant tasks.

Quantum algorithms and error correction use a standard set of gates. So for the implementation of algorithms, we must be able to form a universal set of such gates. For example, Hadamard transformations composed of H gates are indispensable in the Grover's search algorithm, and X gates are required in VQE to generate the Hartree-Fock initial states [52]. Table 1 provides a list of common single- and two-qubit standard quantum gates realized in the context of DQDs according to the MS compilation. We note, these results can be applied in different contexts as will be demonstrated in the forthcoming subsection. Within 7000 (specifically tailored for these commonly used non-parametric gates) optimizations, all of the Ansatzes achieve a respectable accuracy compared to the associated reference gates. All parameters are initialized to 1 before training and kept in the non-negative domain during the optimization to take into account experimental constraints. The Ansatzes used for realizing single-qubit gates employ 12 single-qubit native gates with $\pi/2$ duration. The form of the Ansatz for two-qubit gate is that as depicted in Fig. 2(c) in the Section 3, with pulse duration $\pi/10$ for single-qubit native gates, and $\pi/2$ for two-qubit native gates, making a 6π entire runtime. Thus this work in synergy with Ansatzes with the same runtime for an arbitrary single-qubit gate. The corresponding learning rate and number of training rounds used in optimization is collected in the second and third lines, respectively. More detailed information, e.g., the final pulse strengths, can be found in the Section **Data and code availability**.

In general, the design of two-qubit gates is more challenging than the single-qubit operations, as revealed by the fact that the former requires a prudently selected learning rate and reduced fidelities after convergence. To show the convergence properties of two-qubit Ansatzs, Fig. 3(c) visually illustrates the change in compilation error with training iterations for the two-qubit gates CX and CZ, as well as a single-qubit H gate for comparison. It can be observed that within 7000 training rounds, both CX and CZ achieve a minimum error of approximately 10^{-6}, which is significantly larger than that of their single qubit counterpart ($\sim 10^{-16}$). However, in practice, this level of error is already sufficient to meet the requirements of shallow-depth quantum algorithms. In the following subsection, we will demonstrate that utilizing the quantum circuit constructed with these compiled gates can yield satisfactory computational outcomes.

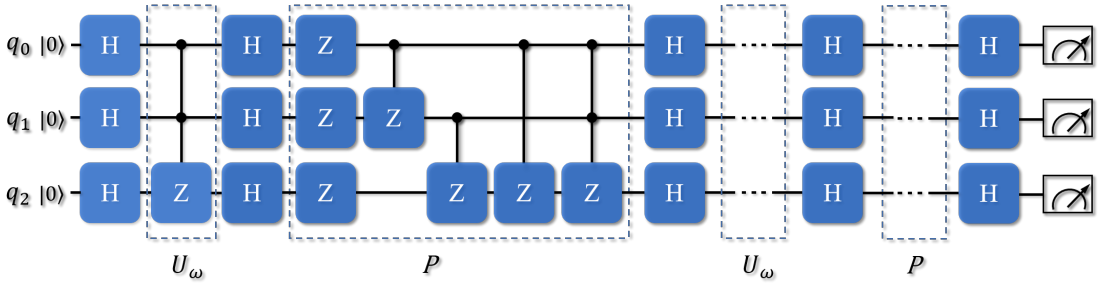
Table 1: A series of common non-parametric universal quantum gate has been realized with MS compilation in DQDs. Within 7000 training rounds, they all achieve respectable accuracies relative to the associated reference gates. Every single-qubit logical gate is obtained by 12 single-qubit native gates with duration $\pi/2$. The structure of the Ansatz for two-qubit gates is depicted in Fig. 2(c) in the Section 3, with pulse duration $\pi/10$ for single-qubit native gates, and $\pi/2$ for two-qubit native gates. All parameters are initialized to 1 before training. The learning rate for optimization is captured in the second line.

	H	T	$T^\dagger$	S	X	Y	Z	CX	CZ
Learning rate	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.01	0.01
Error	5.6×10^{-16}	3.9×10^{-15}	3.3×10^{-16}	7.8×10^{-16}	1.3×10^{-15}	1.7×10^{-15}	1.2×10^{-15}	1.7×10^{-6}	1.4×10^{-6}
Number of training	6649	1248	4424	3204	5036	5537	2512	3706	1694

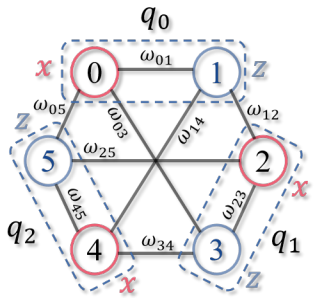
4.2. Compilation for quantum programs

In this subsection, we validate the performance of our MS compilation with two demonstrations in DQDs: the Grover's search algorithm [35, 36] and graph Max-Cut optimization [40]. Our first demonstration checks whether the compiled circuit will faithfully execute the pre-designed and non-parametric reference circuit, i.e., static compilation [17]. While the second demonstration tests whether the MS compilation allows the DQDs native circuit to dynamically update its parameters efficiently to implement variational tasks, i.e., dynamic compilation [17].

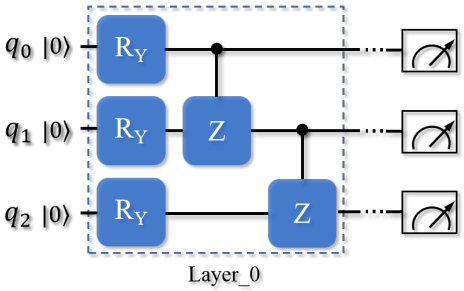
We first focus on the implementation of the Grover's search algorithm in DQDs system by leveraging the MS compilation. For the problem of searching the target items from an unordered database, the quantum Grover's search algorithm permits a quadratic speedup compared to classical algorithms [1]. Here, we consider a three-qubit Grover's search algorithm, with the target state $|\omega\rangle = |111\rangle$. The reference circuit for this task is shown in Fig. 4(a), which contains Hadamard transformations, oracle operators U_ω and flip operators P. The oracle operator U_ω is executed by the CCZ gate,



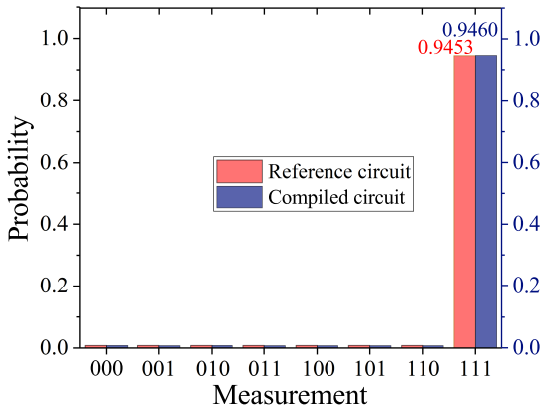
(a)



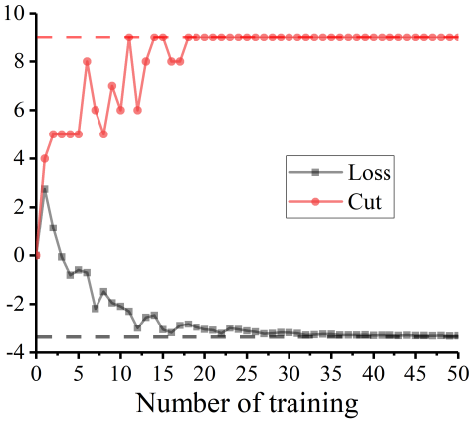
(b)



(c)



(d)



(e)

Figure 4: Two demonstrations of MS compilation in three-qubit DQDs system: the Grover's search algorithm and the MBE-VQE for graph Max-Cut optimization. (a) The reference circuit for Grover's search algorithm with target item "111". This reference circuit will be compiled, according to the MS compilation, into circuit of native gates with fixed structure and parameters before execution. (b) An undirected 3-regular graph with 6 vertices and 9 edges can be represented with a three-qubit system by MBE-VQE [37]. (c) The reference circuit with multiple basic layers used to implement a three-qubit MBE-VQE for graph Max-Cut problem [37]. (d) Numerically calculated probability distributions of different measurements, yield by the compiled version of the reference circuit (a) in DQDs system with MS compilation. (e) The predicted cut result and loss value as functions of the number of training rounds in DQDs with MBE-VQE and MS compilation for the Max-Cut optimization of graph (c). The red and black dashed lines are the final values of cut count and loss obtained by unrestricted system, respectively.

which was decomposed based on the gates set $\{H, T, T^\dagger, CX\}$ [1]. All of logical gates in this reference circuit are compiled by the MS compilation into a native gate sequence which consists of only DQDs. In addition, the forms and parameters of Ansatzes come directly from the results in the previous subsection and the Section 3. After execution, measurements are performed on the qubits, and the probability distributions for each result obtained from both the compiled and reference circuits are shown in Fig. 4(d). From Fig. 4(d), we observe that the compiled and the reference circuits exhibit only a negligible probability difference ($\sim 7.4 \times 10^{-4}$) in predicting the correct result "111". This demonstrates that the DQDs systems with MS compilation can precisely execute pre-designed static quantum programs.

The second demonstration for our MS compilation solves the graph Max-Cut optimization in the DQDs system. The Max-Cut optimization is an NP-hard problem, which divides the vertices of an undirected graph into two parts and maximizes the sum of weighted edges being cut. For a graph with few vertices, its Max-Cut solution can be found within a reasonable amount of time by exhaustive enumeration. However, the classical algorithm will fail quickly because the resource overhead rises exponentially as the graph size scales up. Various VQA are developed to provide the quantum advantage [53] with contemporary NISQ devices, such as the quantum approximate optimization algorithm [54] (QAOA). QAOA utilizes n qubits for the Max-Cut optimization of a graph with n vertices. By virtue of multi-basis graph encoding and nonlinear activation functions, based on traditional VQE [38, 52] and tensor networks [55], the multi-basis encodings VQE (MBE-VQE) improves performance using only half the quantum hardware overhead (qubit number) as well as shallower quantum circuit, and admits a quadratic reduction in measurement complexity compared to the traditional VQE [37].

For concreteness, we consider an undirected 3-regular graph with 6 vertices and 9 edges (see Fig. 4(b) for a schematic representation). In the MBE-VQE algorithm, this graph can be represented with a three-qubit system: for examples, vertices 0 and 1 are encoded into qubit q_0, vertices 2 and 3 into qubit q_1. In addition, the vertices 0 and 2 are mapped to the x-axis while the vertices 1 and 3 to the z-axis as showed in Fig. 4(c). The loss function is made up of products of single-qubit measurements $\langle\sigma_i^z\rangle$ and $\langle\sigma_i^x\rangle$, which are dressed by function $\tanh(x)$ and edge-weight ω_{ij}:

$$\begin{aligned} Loss =& \omega_{01}\tanh(\langle\sigma_0^x\rangle)\tanh(\langle\sigma_0^z\rangle) + \omega_{12}\tanh(\langle\sigma_0^z\rangle)\tanh(\langle\sigma_1^x\rangle) + \omega_{23}\tanh(\langle\sigma_1^x\rangle)\tanh(\langle\sigma_1^z\rangle) \\ &+ \omega_{34}\tanh(\langle\sigma_1^z\rangle)\tanh(\langle\sigma_2^x\rangle) + \omega_{45}\tanh(\langle\sigma_2^x\rangle)\tanh(\langle\sigma_2^z\rangle) + \omega_{05}\tanh(\langle\sigma_0^x\rangle)\tanh(\langle\sigma_2^z\rangle) \\ &+ \omega_{03}\tanh(\langle\sigma_0^x\rangle)\tanh(\langle\sigma_1^z\rangle) + \omega_{14}\tanh(\langle\sigma_0^z\rangle)\tanh(\langle\sigma_2^x\rangle) + \omega_{25}\tanh(\langle\sigma_1^x\rangle)\tanh(\langle\sigma_2^z\rangle). \end{aligned} \tag{4}$$

The Ansatz circuit in this task contains two basic layers composed of single-qubit rotation gates R_Y and entangling two-qubit gates CZ, as illustrated in Fig. 4(c). Each rotation operation R_Y is completed by 12 native gates with variable parameters and $\pi/2$ duration. While the structure and parameters for the CZ gates use the results obtained in the previous subsection. According to the MBE-VQE algorithm, the predicted cut

count is defined as

$$\begin{aligned} Cut =& \frac{\omega_{01}}{2}[1 - R(\langle\sigma_0^x\rangle)R(\langle\sigma_0^z\rangle)] + \frac{\omega_{12}}{2}[1 - R(\langle\sigma_0^z\rangle)R(\langle\sigma_1^x\rangle)] + \frac{\omega_{23}}{2}[1 - R(\langle\sigma_1^x\rangle)R(\langle\sigma_1^z\rangle)] \\ &+ \frac{\omega_{34}}{2}[1 - R(\langle\sigma_1^z\rangle)R(\langle\sigma_2^x\rangle)] + \frac{\omega_{45}}{2}[1 - R(\langle\sigma_2^x\rangle)R(\langle\sigma_2^z\rangle)] + \frac{\omega_{05}}{2}[1 - R(\langle\sigma_0^x\rangle)R(\langle\sigma_2^z\rangle)] \\ &+ \frac{\omega_{03}}{2}[1 - R(\langle\sigma_0^x\rangle)R(\langle\sigma_1^z\rangle)] + \frac{\omega_{14}}{2}[1 - R(\langle\sigma_0^z\rangle)R(\langle\sigma_2^x\rangle)] + \frac{\omega_{25}}{2}[1 - R(\langle\sigma_1^x\rangle)R(\langle\sigma_2^z\rangle)], \end{aligned} \tag{5}$$

where $R(x)$ denotes the classical rounding procedure.

For simplicity, the weights of all the edges are taken to be $\omega_{ij} = 1$. In this case, the correct solution of this Max-Cut problem is 9. Fig. 4(e) displays the loss value *Loss* and predicted cut count *Cut* as functions of number of training rounds, where the learning rate is set be 0.1 and all native gate parameters are initially ones. The red and black dashed lines refer to the final values of *Cut* and *Loss* yielded by unrestricted system, respectively. From Fig. 4(e) we see that the results of the variational native circuit compiled dynamically by the MS compilation, after about 25 rounds of training, has almost no difference in both the predicted cut count and loss value compared to ideal results obtained in unconstrained system. This shows that, *by using our MS compilation, the DQD system has the potential to reliably implement variational algorithms for non-trivial applications.*

4.3. Quantum programs compilation in noisy environment

In the previous subsection, we examin the quantum program compilation without taking into account the surrounding environment. However, in actual experiments, qubits are constantly subject to various fluctuations, making it challenging to achieve high precision control over them, especially considering that large-scale quantum programs may involve a vast number of native gates, resulting in significant error accumulation. Now we explore the effect of the two typical experimental noises: the charge noise and the nuclear noise [22, 20, 24, 21] on the two compiled quantum programs designed in the previous subsection, i.e., three-qubit Grover's and MBE-VQE quantum programs. The charge noise is mainly caused by electron hopping and lead an additional error term $\delta\sigma_z$ to the Hamiltonian (1). While the nuclear noise mainly stems from the hyperfine interactions with spinful nuclei of the host material and its impact on the system can be taken into account by an additional term $\delta\sigma_x$ to the Hamiltonian (1). Considering that their characteristic timescale ($\sim 100\mu$s) is much longer than the duration of native gates ($\sim$ 10ns), we treat them as constants during the execution of the two shallow-depth programs.

In addition, we emphasize that for non-parametric logical quantum gates in these programs, the noise is incorporated into the system's evolution after they have been compiled, such as for all quantum gates in the static Grover program and the CZ gates in the dynamic MBE-VQE program. In contrast, the updating of the parameters in the parametric logical gates RY in MBE-VQE relies on the program's previous outputs affected by noise. This is a reasonable assumption considering the typically

unpredictable nature of the environment. Furthermore, for simplicity again, we assume that the three qubits suffer from the same noise. But the control parameters dressed by charge noise term $J + \delta J$ will be limited to the non-negative interval to respect the intrinsic property of the exchange interactions.

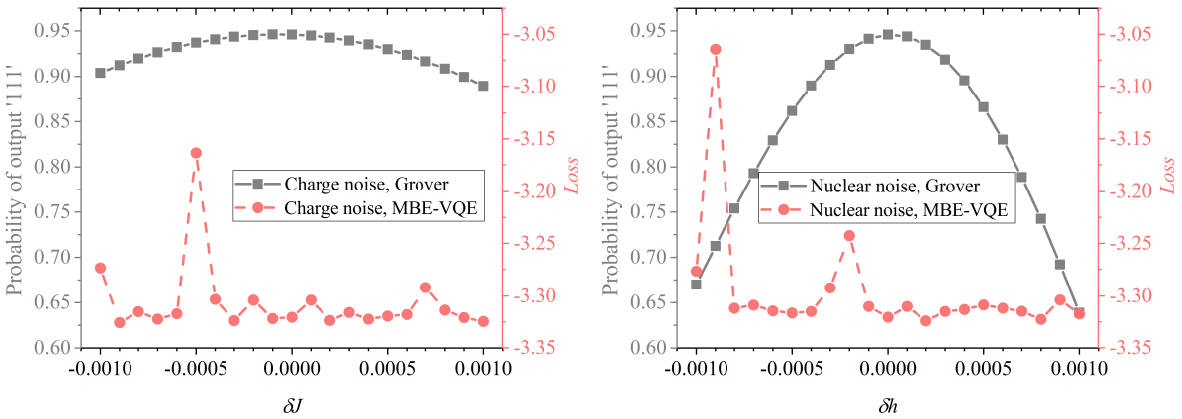


Figure 5: The probability of generating the correct result "111" with the compiled Grover's program, as well as the final loss output by the compiled MBE-VQE program after 50 training steps versus amplitudes of the charge noise (a) and the nuclear noise (b). For two types of noise, the compiled MBE-VQE program consistently predicts the correct cut count of 9 (not shown in figure for clarity).

The probability of obtaining the correct output by the compiled Grover's program, as well as the final loss values output by the compiled MBE-VQE after 50 training steps, are illustrated in Fig. 5(a) and (b), respectively, with respect to intensities of the charge and nuclear noises. For the compiled MBE-VQE program, it consistently predicts the correct cut count of 9, which is not displayed in the figure for better clarity. Fig. 5 clearly shows that for static Grover's program, the probability gradually decreases with increasing noise. However, for the dynamic MBE-VQE program, when the noise is within a reasonable range, the inherent noise resilience of variational algorithm allows the computation results to remain unchanged, thus enabling the algorithm to maintain good performance. Furthermore, we observe that the nuclear noise has a greater impact on the Grover's result compared to the charge noise. These results suggest that in the current NISQ era, shallow-depth quantum variational algorithms may have easier access to successful applications compared to deeper traditional quantum algorithms.

Note that these simulations are performed using MindQuantum [48], a nascent and quickly expanding high-performance software package for quantum computation. It allows efficient problem-solving in quantum machine learning, chemistry simulation, and optimization. All detailed code and data that support this work are available in our online repository displayed in the Section **Data and code availability** for interested readers.

5. Conclusions

For portability, quantum algorithms are generally programmed hardware-independent, which necessitates the effective compilation of gates on platform-specific hardware for practical implementation. Due to the severe and special constraints in DQDs, what should be kept in mind during design the compilation architecture is not only the compilation of high fidelity quantum gates, but also take into account hardware constraints that, for example, prevent undesirable inter-qubit coupling.

In this work, we have shown how to use random quantum states for training in VQA for compiling or decomposing a given unitary to reduce resource overhead in loss calculation and to avoid the optimization being stuck in local optima. The performance of an Ansatz consisting of various numbers of DQDs native gates to achieve single-qubit rotations was explored and the proper trade-off was found. Using these methods, examples of common universal quantum gates were presented that showed fairly high fidelities. Most importantly, we emphasized the necessity of constant-runtime quantum gates for scalability and set up a scalable architecture, the MS compilation, which is able to layout the modularized native gates to achieve logical operations that avoid unwanted couplings between the surrounding qubits, and thus enable the faithful execution of the reference quantum circuit. To obtain an estimation of the performance, we presented two representative applications, the Grover's search algorithm for static compilation and the Max-Cut optimization for dynamic compilation. Both deliver superb results, demonstrating its feasibility for realistic implementation in the experiments. Finally, to simulate the performance of compiled quantum programs on NISQ devices, we explored the effect of noises on them and robustness is observed.

We believe these advances will be immensely helpful in exploiting the potential of the DQDs system to perform complex and meaningful quantum algorithms, thereby reaching a further milestone in the path towards a practical and scalable universal quantum computing device. Furthermore, our methods could potentially be applied to a wide range of different hardware devices.

Data and code availability

The code, running environment of algorithm and all data used or presented in this paper are available from the corresponding author upon reasonable request or from Gitee in (`https://gitee.com/mindspore/mindquantum/tree/research/paper_with_code`).

Acknowledgments

This work was supported by the Natural Science Foundation of Shandong Province (Grant No. ZR2021LLZ004), the Fundamental Research Funds for the Central Universities (Grant No. 202364008) and the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 62002349). The author RHH would

also like to thank Sheng-Gang Ying, Yang-Yang Xie, Shang-Shang Shi and Guo-Long Cui personally for fruitful discussions. MSB was supported by the National Science Foundation of the US (MPS award No. PHYS-1820870).

References

[1] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] Bhupesh Bishnoi. Quantum computation, 2021.

[3] Dietrich Leibfried, Rainer Blatt, Christopher Monroe, and David Wineland. Quantum dynamics of single trapped ions. *Reviews of Modern Physics*, 75(1):281, 2003.

[4] Maciej Lewenstein, Anna Sanpera, Veronica Ahufinger, Bogdan Damski, Aditi Sen, and Ujjwal Sen. Ultracold atomic gases in optical lattices: mimicking condensed matter physics and beyond. *Advances in Physics*, 56(2):243–379, 2007.

[5] MV Gurudev Dutt, L Childress, L Jiang, E Togan, J Maze, F Jelezko, AS Zibrov, PR Hemmer, and MD Lukin. Quantum register based on individual electronic and nuclear spin qubits in diamond. *Science*, 316(5829):1312–1316, 2007.

[6] Qingling Zhu, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lianchen Han, Linyin Hong, He-Liang Huang, Yong-Heng Huo, Liping Li, Na Li, Shaowei Li, Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yulin Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jianghan Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, Youwei Zhao, Liang Zhou, Chao-Yang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin*, 67(3):240–245, 2022.

[7] Han-Sen Zhong, Yu-Hao Deng, Jian Qin, Hui Wang, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Dian Wu, Si-Qiu Gong, Hao Su, and et al. Phase-programmable gaussian boson sampling using stimulated squeezed light. *Physical Review Letters*, 127(18), Oct 2021.

[8] Xin Zhang, Hai-Ou Li, Gang Cao, Ming Xiao, Guang-Can Guo, and Guo-Ping Guo. Semiconductor quantum computation. *National Science Review*, 6(1):32–54, 2019.

[9] Lmk Vandersypen and M. A. Eriksson. Quantum computing with semiconductor spins. *Physics Today*, 72(8):38–45, 2019.

[10] J P Dodson, Nathan Holman, Brandur Thorgrimsson, Samuel F Neyens, E R MacQuarrie, Thomas McJunkin, Ryan H Foote, L F Edge, S N Coppersmith, and M A Eriksson. Fabrication process and failure analysis for robust quantum dots in silicon. *Nanotechnology*, 31(50):505001, oct 2020.

[11] I. Hansen, A. E. Seedhouse, K. W. Chan, F. E. Hudson, K. M. Itoh, A. Laucht, A. Saraiva, C. H. Yang, and A. S. Dzurak. Implementation of an advanced dressing protocol for global qubit control in silicon. *Applied Physics Reviews*, 9(3):031409, 2022.

[12] G.J. Philips Stephan, T. Mdzik Mateusz, V. Amitonov Sergey, L. de Snoo Sander, Russ Maximilian, and Kalhor Nima. Universal control of a six-qubit quantum processor in silicon. *arXiv preprint arXiv:2202.09252*, 2022.

[13] Kenta Takeda, Akito Noiri, Takashi Nakajima, Takashi Kobayashi, and Seigo Tarucha. Quantum error correction with silicon spin qubits. *Nature*, 608:682–686, 2022.

[14] Hendrik Bluhm, Sandra Foletti, Diana Mahalu, Vladimir Umansky, and Amir Yacoby. Universal quantum control of two electron spin qubits via dynamic nuclear polarization. *APS*, pages P17–008, 2009.

[15] E. A. Laird, J. M. Taylor, D. P. DiVincenzo, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Coherent spin manipulation in an exchange-only qubit. *Phys. Rev. B*, 82:075403, Aug 2010.

[16] Z. Shi, C. B. Simmons, J. R. Prance, J. K. Gamble, and S. N. Coppersmith. Fast hybrid silicon double-quantum-dot qubit. *Physical Review Letters*, 108(14):140503, 2012.

[17] Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180–187, 2017.

[18] Run-Hong He, Ren-Feng Hua, Arapat Ablimit, and Zhao-Ming Wang. Approximate quantum gates compiling with self-navigation algorithm. *arXiv preprint arXiv:2204.02555*, 2022.

[19] Xiao-Ming Zhang, Zezhu Wei, Raza Asad, Xu-Chen Yang, and Xin Wang. When does reinforcement learning stand out in quantum control? a comparative study on state preparation. *npj Quantum Information*, 5(1):1–7, 2019.

[20] Xin Wang, Lev S Bishop, Edwin Barnes, JP Kestner, and S Das Sarma. Robust quantum gates for singlet-triplet spin qubits using composite pulses. *Physical Review A*, 89(2):022310, 2014.

[21] Xin Wang, Lev S Bishop, JP Kestner, Edwin Barnes, Kai Sun, and S Das Sarma. Composite pulses for robust universal control of singlet–triplet qubits. *Nature communications*, 3(1):1–7, 2012.

[22] Jason P Kestner, Xin Wang, Lev S Bishop, Edwin Barnes, and S Das Sarma. Noise-resistant control for a spin qubit array. *Physical review letters*, 110(14):140502, 2013.

[23] Mei-Ya Chen, Chengxian Zhang, and Zheng-Yuan Xue. Fast high-fidelity geometric gates for singlet-triplet qubits. *Physical Review A*, 105(2):022620, 2022.

[24] Xu-Chen Yang, Man-Hong Yung, and Xin Wang. Neural-network-designed pulse sequences for robust control of singlet-triplet qubits. *Physical Review A*, 97(4):042324, 2018.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, USA, 2016.

[26] Run-Hong He, Rui Wang, Jing Wu, Shen-Shuang Nie, Jia-Hui Zhang, and Zhao-Ming Wang. Deep reinforcement learning for universal quantum state preparation via dynamic pulse control, 2021.

[27] Yuan-Hang Zhang, Pei-Lin Zheng, Yi Zhang, and Dong-Ling Deng. Topological quantum compiling with reinforcement learning. *Physical Review Letters*, 125(17):170501, 2020.

[28] Tobias Haug, Wai-Keong Mok, Jia-Bin You, Wenzu Zhang, Ching Eng Png, and Leong-Chuan Kwek. Classifying global state preparation via deep reinforcement learning. *Machine Learning: Science and Technology*, 2(1):01LT02, dec 2020.

[29] Run-Hong He, Hai-Da Liu, Sheng-Bin Wang, Jing Wu, Shen-Shuang Nie, and Zhao-Ming Wang. Universal quantum state preparation via revised greedy algorithm. *Quantum Science and Technology*, 6(4):045021, sep 2021.

[30] Christopher Ferrie. Self-guided quantum tomography. *Physical review letters*, 113(19):190404, 2014.

[31] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J Martinez, Jae Hyeon Yoo, Sergei V Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, et al. Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989*, 2020.

[32] Péter Rakyta and Zoltán Zimborás. Approaching the theoretical limit in quantum gate decomposition. *Quantum*, 6:710, 2022.

[33] Péter Rakyta and Zoltán Zimborás. Efficient quantum gate decomposition via adaptive circuit compression. *arXiv preprint arXiv:2203.04426*, 2022.

[34] Ranyiliu Chen, Benchi Zhao, and Xin Wang. Variational quantum algorithm for schmidt decomposition. *arXiv preprint arXiv:2109.10785*, 2021.

[35] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[36] Gui-Lu Long. Grover algorithm with zero theoretical failure rate. *Physical Review A*, 64(2):022307, 2001.

[37] Taylor L. Patti, Jean Kossaifi, Anima Anandkumar, and Susanne F. Yelin. Variational quantum optimization with multibasis encodings. *Phys. Rev. Res.*, 4:033142, Aug 2022.

[38] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love,

Alan Aspuru-Guzik, and Jeremy L Obrien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[39] Stasja Stanisic, Jan Lukas Bosse, Filippo Maria Gambetta, Raul A Santos, Wojciech Mruczkiewicz, Thomas E O'Brien, Eric Ostby, and Ashley Montanaro. Observing ground-state properties of the fermi-hubbard model using a scalable algorithm on a quantum computer. *arXiv preprint arXiv:2112.02025*, 2021.

[40] Clayton W Commander. Maximum cut problem, max-cut. *Encyclopedia of Optimization*, 2, 2009.

[41] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[42] Wonjin Jang, Min-Kyun Cho, Jehyun Kim, Hwanchul Chung, Vladimir Umansky, and Dohun Kim. Individual two-axis control of three singlet-triplet qubits in a micromagnet integrated quantum dot array. *Applied Physics Letters*, 117(23):234001, 2020.

[43] Xian Wu, Daniel R Ward, JR Prance, Dohun Kim, John King Gamble, RT Mohr, Zhan Shi, DE Savage, MG Lagally, Mark Friesen, et al. Two-axis control of a singlet–triplet qubit with an integrated micromagnet. *Proceedings of the National Academy of Sciences*, 111(33):11938–11942, 2014.

[44] J. R Petta. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science*, 309(5744):2180–2184, 2005.

[45] Michael D Shulman, Oliver E Dial, Shannon P Harvey, Hendrik Bluhm, Vladimir Umansky, and Amir Yacoby. Demonstration of entanglement of electrostatically coupled singlet-triplet qubits. *Science*, 336(6078):202–205, 2012.

[46] John M Nichol, Lucas A Orona, Shannon P Harvey, Saeed Fallahi, Geoffrey C Gardner, Michael J Manfra, and Amir Yacoby. High-fidelity entangling gate for double-quantum-dot spin qubits. *npj Quantum Information*, 3(1):1–5, 2017.

[47] David C McKay, Christopher J Wood, Sarah Sheldon, Jerry M Chow, and Jay M Gambetta. Efficient z gates for quantum computing. *Physical Review A*, 96(2):022330, 2017.

[48] MindQuantum Developer. Mindquantum, version 0.6.0, March 2021.

[49] Vivek V Shende, Igor L Markov, and Stephen S Bullock. Minimal universal two-qubit controlled-not-based circuits. *Physical Review A*, 69(6):062321, 2004.

[50] P. Kingma Diederik and Ba Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980v9*, 2017.

[51] Yang-Yang Xie, Feng-Hua Ren, Run-Hong He, Arapat Ablimit, and Zhao-Ming Wang. Stochastic learning control of adiabatic speedup in a non-markovian open qutrit system. *Physical Review A*, 106(6):062612, 2022.

[52] Sam McArdle, Suguru Endo, Alan Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, 2020.

[53] Andrew Lucas. Ising formulations of many np problems. *Frontiers in physics*, page 5, 2014.

[54] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[55] Matthew Fishman, Steven White, and Edwin Stoudenmire. The itensor software library for tensor network calculations. *SciPost Physics Codebases*, page 004, 2022.