

Jenga-Krotov algorithm: Efficient compilation of multiqubit gates for exchange-only qubits

Jiahao Wu^{1,2,3,*}, Guanjie He^{1,*}, Wenyuan Zhuo^{1,2}, Quan Fu^{1,4}, and Xin Wang^{1,2,†}

¹Department of Physics, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR, China

²City University of Hong Kong Shenzhen Research Institute, Shenzhen, Guangdong 518057, China

³Quantum Science Center of Guangdong-Hong Kong-Macao Greater Bay Area, Shenzhen, Guangdong 518045, China

⁴School of Physics and Technology, Wuhan University, Wuhan 430072, China



(Received 16 July 2025; accepted 3 October 2025; published 12 November 2025)

Exchange-only (EO) qubits, implemented in triple-quantum-dot systems, offer a compelling platform for scalable semiconductor-based quantum computing by enabling universal control through purely exchange interactions. While high-fidelity single- and two-qubit gates have been demonstrated, the synthesis of efficient multiqubit operations—such as the Toffoli gate—remains a key bottleneck. Conventional gate decompositions into elementary operations lead to prohibitively long and error-prone pulse sequences, limiting practical deployment. In this work, we introduce a gradient-based optimization algorithm, Jenga-Krotov (JK), tailored to discover compact, high-fidelity EO gate sequences. Applying JK to the Toffoli gate, we reduce the number of required exchange unitaries from 216 (in direct decomposition) to 92, and compress the time steps required from 162 to 50, all while maintaining target fidelity. Under realistic noise, the accumulated gate error from our optimized sequence is an order of magnitude lower than that of conventional approaches. We have also applied the JK algorithm to other multiqubit gates and algorithm. For the Fredkin gate, it reduces the number of time steps from 200 to 104 and the number of exchange unitaries from 276 to 172. For the quantum Fourier transform, it compresses the sequence from 180 to 80 time steps and from 237 to 202 exchange unitaries. These results demonstrate that the JK algorithm is a general and scalable strategy for multiqubit gate synthesis in EO architectures, potentially facilitating realization of multiqubit algorithms on semiconductor platforms.

DOI: [10.1103/1x1x-j1xl](https://doi.org/10.1103/1x1x-j1xl)

I. INTRODUCTION

Quantum computing promises exponential speedup for problems intractable on classical machines, ranging from cryptographic factoring to quantum simulation. Over the past two decades, significant progress in quantum system design, control, and coherent manipulation has brought this vision closer to reality, leading to the Noisy Intermediate-Scale Quantum (NISQ) era [1–4]. However, NISQ devices suffer from imperfect gate fidelities and limited error correction, posing major obstacles to scalable computation [5]. Among the various physical platforms under development, semiconductor-based spin qubits stand out for their scalability, long coherence times, and compatibility with industrial fabrication processes [6–15].

Within this class, exchange-only (EO) qubits—encoded in triple quantum dots [16–18]—enable universal quantum computation using only Heisenberg exchange interactions, without requiring electron spin resonance or magnetic field

gradients [19–23]. By operating within decoherence-free subspaces (DFS) [24], EO qubits offer fast and efficient gate operations that can potentially approach subnanosecond timescales [9,25]. These features allow more operations to be executed within the limited coherence window, making EO qubits an attractive candidate for implementing quantum algorithms in NISQ devices.

While many variants of EO qubits have been proposed, such as hybrid, resonant-exchange, and quadrupolar designs [26–36], this work focuses on the canonical implementation: a linear triple-dot array, each dot hosting a single electron. This configuration offers a clean and well-controlled platform. In particular, it provides an ideal testbed for high-fidelity multiqubit gate synthesis—a key step toward scalable EO-based quantum processors.

Efficient implementation of multiqubit gates is a central challenge in scaling up quantum computing, particularly in platforms like EO qubits, where gate operations are restricted to nearest-neighbor exchange interactions. While single- and two-qubit gates have been extensively studied and optimized—for example, the CNOT sequence proposed by Fong and Wandzura remains the most efficient known construction for EO systems [37–39]—the synthesis of multiqubit gates remains underexplored due to its inherent complexity.

To address this gap, we focus on the construction of a three-qubit gate—the Toffoli gate—as a prototypical benchmark for scalable multiqubit gate design. The Toffoli gate is a widely used universal gate for classical reversible

*These authors contributed equally to this work.

†Contact author: x.wang@cityu.edu.hk

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

computation and plays a central role in quantum algorithms and fault-tolerant architectures [40–42]. While it is theoretically possible to decompose any multiqubit gate into sequences of single- and two-qubit gates, this strategy becomes inefficient in the EO qubit setting. For instance, the direct decomposition (later called “DIR”) of a Toffoli gate into single-qubit gates and CNOTs (using the Fong-Wandzura sequence) results in 216 unitaries and 162 time steps [43] (cf. Sec. III A), which significantly amplifies decoherence and resource overhead. Furthermore, numerical efforts using existing optimal control methods like Gradient Ascent Pulse Engineering (GRAPE) or standard Krotov encounter challenges: They either demand impractical computational time and resources, or frequently become trapped in local minima, or stall on barren plateaus in the fidelity landscape. These considerations motivate the development of optimization strategies for direct synthesis of multiqubit gates, starting with the Toffoli gate as a foundational step toward more complex gate constructions in EO qubit architectures.

In this work, we address these challenges by introducing a gradient-based numerical method—the Jenga-Krotov (JK) algorithm—which enables the efficient discovery of high-fidelity, low-depth pulse sequences for multiqubit gates. While our benchmark application focuses on the Toffoli gate in a linearly coupled triple-EO-qubit system, the JK algorithm is general and broadly applicable to the compilation of arbitrary multiqubit gates and alternative geometries (e.g., triangular configurations [44,45] within the EO qubit framework) while overcoming key limitations of previous optimal control methods.

The remainder of this paper is organized as follows: In Sec. II, we introduce the EO qubit and its encoding within a DFS, and extend this framework to a system of three EO qubits. Section III presents methods to decompose a multi-qubit gate, including DIR, GRAPE, and the Krotov’s method. We further present details of our JK algorithm for efficient pulse sequence optimization. In Sec. IV, we apply the JK algorithm to construct a high-fidelity Toffoli gate and evaluate its performance under realistic noise. Finally, Sec. V summarizes our main findings and discusses the broader implications of the JK algorithm for scalable EO qubit systems.

II. MODEL

A. EO qubit

The EO qubit considered here consists of a linear array of three spin-1/2 particles (labeled 1, 2, 3—there are complications arising from the order of the labeling, which we will discuss later), with only nearest-neighbor exchange interactions. The system is governed by the Hamiltonian

$$\mathcal{H} = \sum_{(i,j)} \mathcal{H}_{i,j},$$

$$\mathcal{H}_{i,j} = J_{i,j}(t) \mathbf{S}_i \cdot \mathbf{S}_j \quad (i, j = 1, 2, 3), \quad (1)$$

where $\mathbf{S}_i = \frac{1}{2}\boldsymbol{\sigma}_i$ is the spin of particle i , $\boldsymbol{\sigma}_i$ is the corresponding Pauli operator, $J_{i,j}(t)$ is the time-dependent exchange coupling between particles i and j , used to implement exchange unitaries, and (i, j) indicates nearest neighbors.

The system has eight eigenstates, denoted $|1\rangle \dots |8\rangle$. These can be expressed both in the spin basis and in terms of the complete set of commuting observables $\{S_{\text{tot}}^{(3)}, S_{z,\text{tot}}^{(3)}, S_{1,2}\}$ (here, the labels 1, 2, 3 are in *natural order*, which will be explained later) as follows [21,46,47]:

$$\begin{aligned} |1\rangle &= \left| \frac{1}{2}, \frac{1}{2}, 0 \right\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\uparrow\rangle - |\downarrow\uparrow\uparrow\rangle), \\ |2\rangle &= \left| \frac{1}{2}, -\frac{1}{2}, 0 \right\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\downarrow\rangle - |\downarrow\uparrow\downarrow\rangle), \\ |3\rangle &= \left| \frac{1}{2}, \frac{1}{2}, 1 \right\rangle = \sqrt{\frac{2}{3}}|\uparrow\uparrow\downarrow\rangle - \frac{1}{\sqrt{6}}|\uparrow\downarrow\uparrow\rangle - \frac{1}{\sqrt{6}}|\downarrow\uparrow\uparrow\rangle, \\ |4\rangle &= \left| \frac{1}{2}, -\frac{1}{2}, 1 \right\rangle = \frac{1}{\sqrt{6}}|\uparrow\downarrow\downarrow\rangle + \frac{1}{\sqrt{6}}|\downarrow\uparrow\downarrow\rangle - \sqrt{\frac{2}{3}}|\downarrow\downarrow\uparrow\rangle, \\ |5\rangle &= \left| \frac{3}{2}, \frac{3}{2}, 1 \right\rangle = |\uparrow\uparrow\uparrow\rangle, \\ |6\rangle &= \left| \frac{3}{2}, \frac{1}{2}, 1 \right\rangle = \frac{1}{\sqrt{3}}(|\uparrow\uparrow\downarrow\rangle + |\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\rangle), \\ |7\rangle &= \left| \frac{3}{2}, -\frac{1}{2}, 1 \right\rangle = \frac{1}{\sqrt{3}}(|\uparrow\downarrow\downarrow\rangle + |\downarrow\uparrow\downarrow\rangle + |\downarrow\downarrow\uparrow\rangle), \\ |8\rangle &= \left| \frac{3}{2}, -\frac{3}{2}, 1 \right\rangle = |\downarrow\downarrow\downarrow\rangle. \end{aligned} \quad (2)$$

Here, the middle part of the equality chain displays the states in the form $|S_{\text{tot}}^{(3)}, S_{z,\text{tot}}^{(3)}, S_{1,2}\rangle$, where $S_{\text{tot}}^{(3)}$ denotes the total spin of the three particles, $S_{z,\text{tot}}^{(3)}$ is the total spin projection along the z axis, and $S_{1,2}$ represents the total spin of particles 1 and 2. Superscript “(3)” indicates the total of three spins. The right-hand side of Eq. (2) shows the corresponding spin configurations; for example, $|\uparrow\downarrow\uparrow\rangle$ represents a state in which particle 1 is spin-up, particle 2 is spin-down, and particle 3 is spin-up.

The qubit state can be encoded in the $S_{z,\text{tot}}^{(3)} = 1/2$ subspace using states $|1\rangle$ and $|3\rangle$, which may leak into $|6\rangle$ under an inhomogeneous magnetic field; alternatively, one can use the $S_{z,\text{tot}}^{(3)} = -1/2$ subspace with states $|2\rangle$ and $|4\rangle$, which may leak into $|7\rangle$ [46,48,49]. In this work, we assume a homogeneous external magnetic field and neglect the leaked states $|6\rangle$ and $|7\rangle$; therefore, the qubit is defined entirely in the DFS [37]. In this paper, we take $S_{z,\text{tot}}^{(3)} = 1/2$; therefore, the qubit states are $|\mathbb{0}\rangle = |1\rangle$ and $|\mathbb{1}\rangle = |3\rangle$. The labels of the two qubit states (“ $\mathbb{0}$ ” and “ $\mathbb{1}$ ”) are identical to the corresponding $S_{1,2}$. We further note that once this convention is chosen, the three spins are no longer interchangeable.

To facilitate later discussion, we follow a notation introduced in Refs. [38,39,50]. In this notation, each spin is represented by the a bullet $\bullet$, and the groups of spins that related to chosen quantum numbers are enclosed in brackets (ovals in figures) labeled by their total spin. Therefore, our $|\mathbb{0}\rangle = ((\bullet\bullet)_0\bullet)_{1/2}$ and $|\mathbb{1}\rangle = ((\bullet\bullet)_1\bullet)_{1/2}$, where the subscript for the inner brackets indicates $S_{1,2}$, while that for the outer brackets indicates $S_{\text{tot}}^{(3)}$.

It is of utmost importance to discuss the arrangement of the three spins, which plays a critical role in our analysis. The conventional angular momentum coupling process

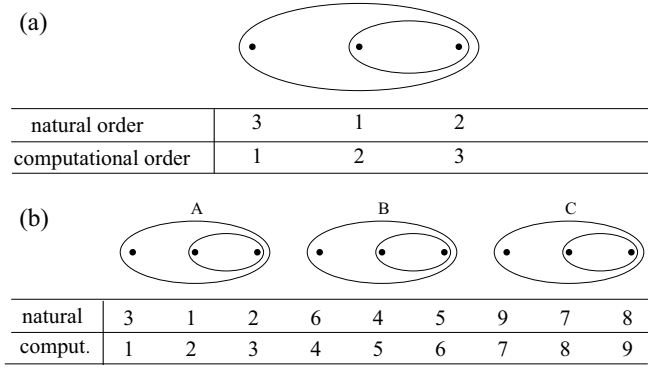


FIG. 1. The bullet notation for (a) a single EO qubit and (b) three EO qubits labeled by A, B, and C. The corresponding natural and computational orders are indicated.

inherently assumes a specific ordering: First, spin 1 and spin 2 are coupled, and then the resulting total spin is coupled to spin 3. We refer to this ordering as the *natural order*. The eight states $|1\rangle, \dots, |8\rangle$ introduced previously are constructed based on this natural order. On the other hand, when conducting quantum computation, we intuitively label spins from left to right (or top to bottom) with 1, 2, and 3. This is referred to as the *computational order*. In some cases, such as $((\bullet\bullet)_a\bullet)_{1/2}$, these two orders are the same, but in most situations that they are different. For example, in the original Fong-Wandzura sequence [37], computational order (1, 2, 3, 4, 5, 6) is actually (3, 1, 2, 4, 5, 6) in natural order, namely, $(\bullet(\bullet\bullet)_a)_{1/2}((\bullet\bullet)_b\bullet)_{1/2}$. This problem is even more pronounced when multiple EO qubits are coupled, as will be discussed later. In this work, we adopt the *natural order* when angular momentum coupling is involved, and the *computational order* when constructing Hamiltonians and evolution operators for quantum computation where the geometric ordering is more intuitive. The specific order being used will be clearly stated wherever ambiguity may arise. For clarity, Fig. 1 explicitly illustrates the correspondence between natural and computational orderings for both a single EO qubit and a three-qubit system.

To implement gates for an EO qubit, we adopt a piecewise-constant control protocol for $J_{i,j}(t)$, consistent with experimentally realizable square pulses and commonly used in optimal control simulations. The total evolution time T is

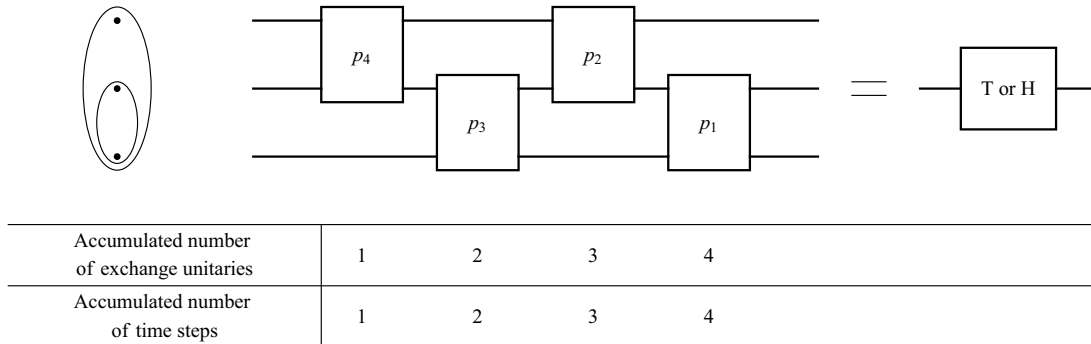


FIG. 2. Quantum circuit showing direct decomposition of T and H gates, along with the accumulated number of exchange unitaries and number of time steps indicated. Each block corresponds to an exchange unitary as defined in Eq. (4), with the p value indicated inside the block.

TABLE I. Compilation of a T gate and an Hadamard (H) gate into four exchange unitaries: $\phi_1 = \arccos(-1 + \sqrt{2} + \frac{-2+\sqrt{2}}{\sqrt{3}})/\pi$, $\phi_2 = \arccos(\frac{-1+\sqrt{2}+\sqrt{6}}{3})/\pi$, and $\theta = 1 + \arccos(\frac{2-\sqrt{2}+\sqrt{70+36\sqrt{2}}}{12})/\pi$.

Gate	$U_{23}(p_1)$	$U_{12}(p_2)$	$U_{23}(p_3)$	$U_{12}(p_4)$
T	θ	$\frac{5}{4} - \theta$	θ	$1 - \theta$
H	$\phi_1 - 1$	$1 - \phi_2$	$1 + \phi_2$	$2 - \phi_1$

divided into segments. Each segment, with a fixed duration τ and constant exchange coupling $J_{i,j} = J$, is referred to as *one* “exchange unitary,” a term we will use later for gate counting. To make the exchange unitaries explicit in our computational framework, we rewrite the Hamiltonian in *computational order* as

$$\mathcal{H}_{i,j}^{\text{ex}} = \frac{1}{4}\sigma_i \cdot \sigma_j - \frac{1}{4}(i, j = 1, 2, 3). \quad (3)$$

Defining $p = J\tau/\pi$, an exchange unitary between spins i and j is then given by

$$U_{i,j}^{\text{ex}}(p) = e^{-ip\pi\mathcal{H}_{i,j}^{\text{ex}}}. \quad (4)$$

The task of implementing quantum algorithms and control protocols essentially reduces to finding a sequence of such exchange unitaries that realizes a target quantum gate. This naturally formulates a quantum optimal control problem, which can be addressed using either analytical or numerical methods.

Here, we give two explicit examples on single-qubit gates: a T gate and a Hadamard (H) gate. In this calculation, the computational order (1, 2, 3) is (3, 1, 2) in natural order, which means $|0\rangle = (\bullet(\bullet\bullet)_0)_{1/2}$ and $|1\rangle = (\bullet(\bullet\bullet)_1)_{1/2}$. Both of them can be decomposed into four exchange unitaries in the form of

$$T, H = U_{23}(p_1)U_{12}(p_2)U_{23}(p_3)U_{12}(p_4). \quad (5)$$

The p values are given in Table I. The sequence in Eq. (5) is also shown in Fig. 2, where the accumulated number of exchange unitaries and number of time steps are indicated. While this is merely a simple example, it provides a pedagogical starting point for later, much more complicated sequences.

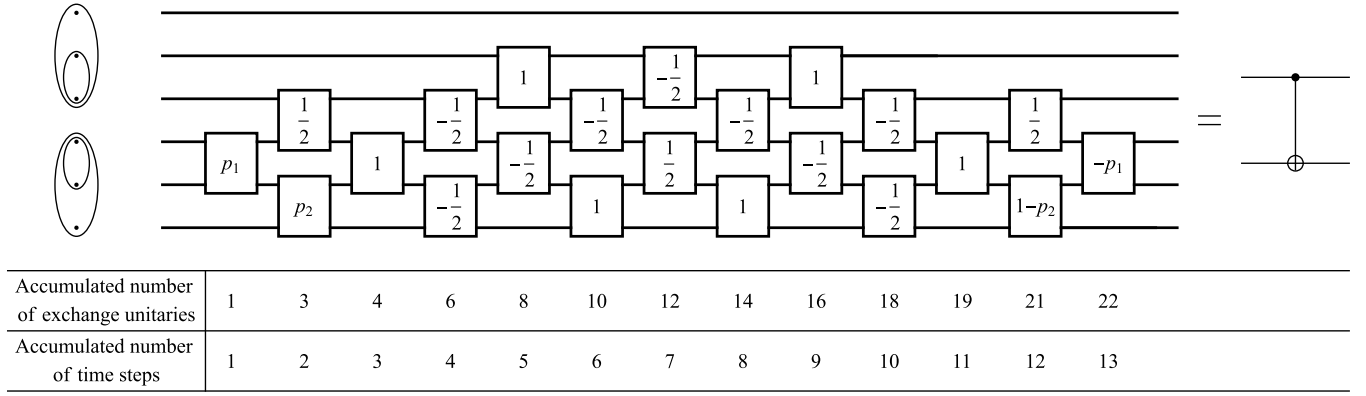


FIG. 3. Quantum circuit showing the Fong-Wandzura sequence in the natural order (3, 1, 2, 4, 5, 6) accomplishing a CNOT gate, along with the accumulated number of exchange unitaries and number of time steps indicated. Each block corresponds to an exchange unitary as defined in Eq. (4), with the p value indicated inside the block. Here, $p_1 = \arccos(-1/\sqrt{3})/\pi$ and $p_2 = \arcsin(1/3)/\pi$.

Two-qubit gates are fundamental to quantum computation, with the CNOT gate being a particularly important and extensively studied example. DiVincenzo *et al.* [19,51] first proposed explicit pulse sequences for an encoded two-qubit gate that is locally equivalent to a CNOT gate. Their approach, consisting of 19 exchange unitaries over 13 time steps, was later shown to approximate an exact analytical solution [52]. Subsequently, Zeuch *et al.* [50] derived a fully analytical CNOT gate sequence for the entire subsystem, requiring approximately 40 exchange unitaries. Fong and Wandzura [37] introduced an improved analytic pulse sequence for the encoded CNOT gate, which achieves the same operation using only 22 exchange unitaries in 13 time steps, as shown in Fig. 3. This represents the most efficient implementation of the CNOT gate in EO qubit systems to date. We emphasize again that the original Fong-Wandzura sequence is with natural order of (3, 1, 2, 4, 5, 6). The one for natural order of (3, 1, 2, 6, 4, 5), implemented in this work, has a slightly different number of exchange unitaries which is shown in Fig. 4.

B. Triple EO qubit system

We now consider a linear array of three EO qubits, labeled A, B, and C, arranged in the natural order (3, 1, 2, 6, 4, 5, 9, 7, 8), as shown in Fig. 1(b). The corresponding Hilbert space is spanned by $2^9 = 512$ basis states. However, the problem can be simplified

by classifying the states using nine quantum numbers: $\{S_{\text{tot}}^{(9)}, S_{z,\text{tot}}^{(9)}, S_{A,B}, S_A, S_B, S_C, S_{A,1,2}, S_{B,1,2}, S_{C,1,2}\}$. Here, $S_{\text{tot}}^{(9)}$ and $S_{z,\text{tot}}^{(9)}$ denote the total spin and total z component of spin of the full three-qubit system, with the superscript “(9)” indicating that these quantities are defined over all nine spins. $S_{A,B}$ is the total spin of qubits A and B; S_A (S_B , S_C) denotes the total spin of qubit A (B, C); and $S_{A,1,2}$ ($S_{B,1,2}$, $S_{C,1,2}$) represents the total spin of particles 1 and 2 in *natural order* within qubit A (B, C). The eight computational basis states can be represented as follows in bullet notation:

$$|abc\rangle = (\bullet(\bullet\bullet)_a)_{1/2}(\bullet(\bullet\bullet)_b)_{1/2}(\bullet(\bullet\bullet)_c)_{1/2}, \quad (6)$$

where a, b, c are either $\mathbb{0}$ or $\mathbb{1}$ depending on the state of its corresponding qubits A, B, C, respectively.

While the complete basis set consists of 512 states, only a subset of these is relevant for our analysis, as the focus is on implementing three-qubit gates within the DFS. Since the DFS of a single EO qubit corresponds to the spin-1/2 subspace, applying three-qubit gates to the DFS requires targeting the matrix block associated with states where S_A , S_B , and S_C are all 1/2. These states are referred to as the computational bases. In the total angular momentum basis, the full 512×512 matrix exhibits a highly block-diagonal structure. The key to reducing the basis size is to identify the smallest block matrix that encompasses all computational basis states,

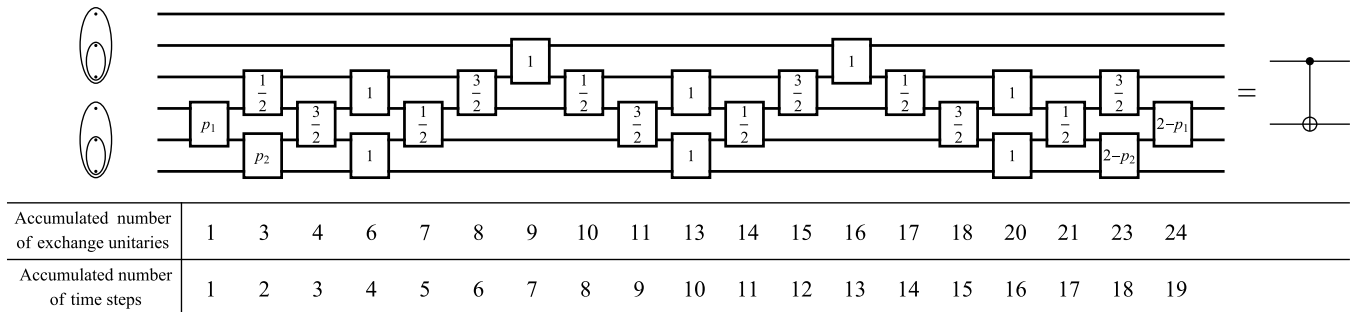


FIG. 4. Quantum circuit illustrating the equivalent Fong-Wandzura sequence in the natural order (3, 1, 2, 6, 4, 5) that implements a CNOT gate, with the accumulated number of exchange unitaries and number of time steps indicated. This sequence is the one used in this work. Each block corresponds to an exchange unitary as defined in Eq. (4), with the p value indicated inside the block. Here, $p_1 = \arccos(\frac{2\sqrt{3}-1}{3})/\pi$ and $p_2 = \arcsin(\frac{2\sqrt{3}-1}{3})/\pi$.

TABLE II. Classification of the 90 basis states relevant for the triple-EO-qubit system, selected from the $S_{\text{tot}}^{(9)} = 1/2$ and $3/2$ subspaces.

$S_{\text{tot}}^{(9)}$	$S_{A,B}$	S_C	Number of states	Computational
$\frac{1}{2}$	0	$\frac{1}{2}$	$5 \times 2 = 10$	Y
$\frac{1}{2}$	1	$\frac{1}{2}$	$9 \times 2 = 18$	Y
$\frac{1}{2}$	1	$\frac{3}{2}$	$9 \times 1 = 9$	N
$\frac{1}{2}$	2	$\frac{3}{2}$	$5 \times 1 = 5$	N
$\frac{3}{2}$	1	$\frac{1}{2}$	$9 \times 2 = 18$	Y
$\frac{3}{2}$	2	$\frac{1}{2}$	$5 \times 2 = 10$	N
$\frac{3}{2}$	0	$\frac{3}{2}$	$5 \times 1 = 5$	N
$\frac{3}{2}$	1	$\frac{3}{2}$	$9 \times 1 = 9$	N
$\frac{3}{2}$	2	$\frac{3}{2}$	$5 \times 1 = 5$	N
$\frac{3}{2}$	3	$\frac{3}{2}$	$1 \times 1 = 1$	N

which is sufficient for implementing the desired quantum gates. Since the interaction between EO qubits consists solely of exchange couplings that commute with the total angular momentum operator $S_{\text{tot}}^{(9)}$, the matrix for a given value of $S_{\text{tot}}^{(9)}$ naturally exhibits a block-diagonal structure. Among these blocks, only those corresponding to $S_{\text{tot}}^{(9)} = 1/2$ and $S_{\text{tot}}^{(9)} = 3/2$ can contain the states where S_A , S_B , and S_C are all $1/2$. Consequently, the full 512×512 matrix can be effectively reduced to the two relevant blocks associated with $S_{\text{tot}}^{(9)} = 1/2$ and $S_{\text{tot}}^{(9)} = 3/2$. Since the exchange coupling is rotationally invariant and commutes with $S_{\text{tot}}^{(9)}$, the Hamiltonian is further block diagonal with respect to $S_{z,\text{tot}}^{(9)}$. This symmetry implies that it is sufficient to consider only one specific value of $S_{z,\text{tot}}^{(9)}$ (in this work, we consistently select the maximum value). As a result, the two relevant blocks can be further reduced to a matrix constructed from only 90 basis states.

Table II provides a detailed explanation of how the 90 basis states used in this work are systematically selected from the two relevant blocks corresponding to $S_{\text{tot}}^{(9)} = 1/2$ and $S_{\text{tot}}^{(9)} = 3/2$. The second and third columns list all possible combinations of $S_{A,B}$ and S_C contributing to $S_{\text{tot}}^{(9)} = 1/2$ and $S_{\text{tot}}^{(9)} = 3/2$, respectively. The fourth column calculates the total number of bases by taking the product of the possible state counts in the second and third columns, thereby summarizing the selection process. The details of these 90 bases are listed explicitly in the Supplemental Material [53].

The last column of Table II checks whether there exists a computational subspace in the one spanned by the bases in the corresponding row. A computational subspace is composed of eight computational bases. Under these bases, an arbitrary three-qubit gate can then be expressed as the following 90×90 matrix structure:

$$\begin{pmatrix} U_{8 \times 8}^{\text{comp}} & 0 & 0 & 0 & 0 & 0 \\ 0 & U_{2 \times 2}^{\text{leak}} & 0 & U_{2 \times 24}^{\text{leak}} & 0 & 0 \\ 0 & 0 & U_{8 \times 8}^{\text{comp}} & 0 & 0 & 0 \\ 0 & U_{24 \times 2}^{\text{leak}} & 0 & U_{24 \times 24}^{\text{leak}} & 0 & 0 \\ 0 & 0 & 0 & 0 & U_{8 \times 8}^{\text{comp}} & 0 \\ 0 & 0 & 0 & 0 & 0 & U_{40 \times 40}^{\text{leak}} \end{pmatrix}. \quad (7)$$

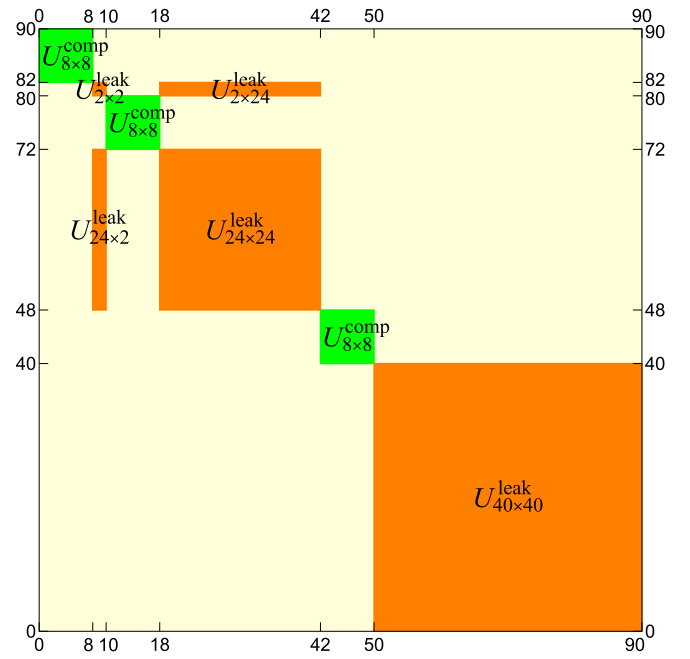


FIG. 5. Block matrix structure of the reduced 90×90 Hamiltonian in the total angular momentum basis, showing computational (green) and leakage (orange) subspaces. Yellow regions are identically zero. Three identical 8×8 computational subspaces appear along the diagonal.

Here, the superscript “comp” denotes the computational subspace, while the superscript “leak” refers to subspaces involving leaked states. Figure 5 depicts the block structure of this 90×90 matrix. The regions highlighted in yellow correspond to matrix elements that are identically zero. The green blocks represent the subspaces spanned by the states involved in the implementation of the target three-qubit gate, while the orange blocks indicate the subspaces associated with leaked states. Due to permutation symmetry and block duplication under spin projection, three identical 8×8 computational subspaces appear along the diagonal. Therefore, any three-qubit gate that acts on the DFS is encoded in the upper-left-most block $U_{8 \times 8}^{\text{comp}}$. The main goal of this work is to find an efficient compilation, using elementary exchange unitaries to realize a Toffoli gate for this $U_{8 \times 8}^{\text{comp}}$, as shown in Fig. 6.

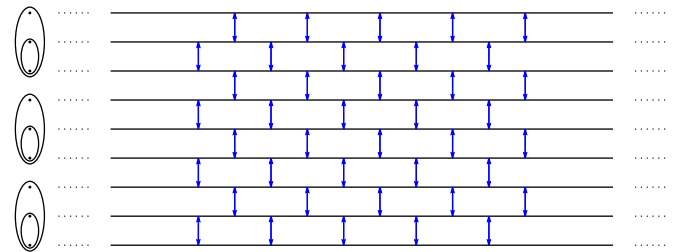


FIG. 6. Schematic illustration of the compilation of three-qubit gates in a triple-EO-qubit system into exchange unitaries. The horizontal black lines represent individual spins and vertical blue arrows indicate controllable exchange unitaries.

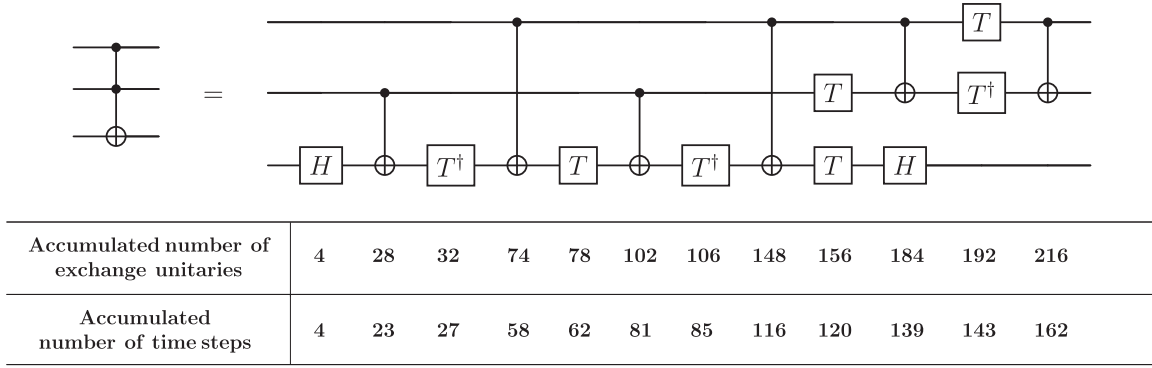


FIG. 7. Quantum circuit showing direct decomposition of Toffoli gate into T, H, and CNOT gates, namely, the DIR sequence.

III. METHODS

In this section, we discuss different approaches to decomposing a Toffoli gate into exchange unitaries. Section III A introduces the DIR method, which, while conceptually straightforward, is highly inefficient—requiring 216 exchange unitaries and 162 time steps. Numerical optimization methods, on the other hand, pose significant challenges due to the vast and complex parameter space and the intricate structure of the EO qubit Hamiltonian. Traditional algorithms often struggle to find global optima, frequently getting trapped in local minima or stalling on barren plateaus in the fidelity landscape, which leads to poor convergence or unacceptably low gate fidelities in practice. In Sec. III B, we examine the GRAPE method, followed by a discussion of Krotov’s method in Sec. III C. Neither method proves satisfactory in solving our problem. To address these limitations, in Sec. III D, we present the JK algorithm, the key contribution of this work, and demonstrate that it offers a practical and high-fidelity solution for Toffoli gate design in EO architectures, overcoming these difficulties through a hybrid update rule that balances convergence stability and global search capability.

A. DIR

A straightforward yet ultimately impractical approach to implementing a Toffoli gate is via direct gate decomposition into single- and two-qubit gates, notably T, H, and CNOT gates. However, this method typically yields excessively long pulse sequences, significantly undermining the efficiency of this fundamental operation. As illustrated in Fig. 7, a typical decomposition consists of two Hadamard gates, seven T (or $T^\dagger$) gates, six CNOT gates, and four swap gates [54], resulting in a sequence that requires at least 216 exchange unitaries and 162 time steps. Here, the SWAP gate refers to a quantum logic gate that exchanges the quantum states between two logical qubits. We refer to this as the “DIR” sequence for comparison with the JK algorithm results presented in Sec. IV. The presence of four swap gates in the decomposition arises from the connectivity constraints of the qubits. Specifically, among the six CNOT gates, four are between nearest-neighbor qubits and can be directly implemented using the equivalent FW sequence, as shown in Fig. 4. However, the remaining two CNOT gates act between non-nearest-neighbor qubits. To realize these operations, each non-nearest-neighbor CNOT gate requires two swap gates: one to bring the involved qubits

adjacent to each other before applying the CNOT and another to return them to their original positions afterward. Thus, a total of four swap gates are needed for these two non-nearest-neighbor CNOT gates. Such high overhead makes direct decomposition impractical in realistic quantum architectures, where Toffoli gates are frequently used and low-depth implementations are essential.

B. GRAPE method

GRAPE is a widely adopted numerical optimization algorithm in quantum control, particularly effective for synthesizing high-fidelity unitary gates in a wide range of quantum systems. Its primary goal is to optimize a sequence of control pulses that guides a quantum system from a given initial state to a desired target state, or equivalently, to implement a specific unitary transformation. By iteratively maximizing a fidelity function using gradient-based techniques, GRAPE offers a robust and flexible framework for addressing a broad class of quantum control tasks.

The evolution of quantum systems is governed by the time-dependent Schrödinger equation

$$\frac{d}{dt}U(t) = -i\mathcal{H}(t)U(t), \quad (8)$$

where $U(t)$ is the time-evolution operator and $\mathcal{H}(t)$ is the Hamiltonian of the system, which can typically be expressed as

$$\mathcal{H}(t) = \mathcal{H}_0 + \sum_k u_k(t)\mathcal{H}_k. \quad (9)$$

Here, $\mathcal{H}_0$ represents the drift Hamiltonian of the system, $\mathcal{H}_k$ are the control Hamiltonians, and $u_k(t)$ are the time-dependent control fields to be optimized.

The fidelity of the control process is often defined in terms of the overlap between the achieved unitary $U(T)$ at the final time T and the target unitary U_{target} . For unitary control, the fidelity function is typically expressed as

$$\Phi = \frac{1}{d} \text{Re}(\text{Tr}[U_{\text{target}}^\dagger U(T)]), \quad (10)$$

where the dimension of the Hilbert space $d = 24$.

To optimize the fidelity Φ , GRAPE calculates the gradient of Φ with respect to the control fields $u_k(t)$. The gradient can be efficiently computed using the chain rule and the propagators at intermediate time steps. Specifically, the gradient with

respect to $u_k(t_j)$ (the control amplitude of the k th field at time step t_j) is given by

$$\frac{\partial \Phi}{\partial u_k(t_j)} = \frac{2}{d} \text{Re} \left(\text{Tr} \left[U_{\text{target}}^\dagger U(T) U(t_{N_T-1}) \cdots U(t_{j+1}) \right. \right. \\ \left. \left. \times \left(-i \frac{T}{N_T} \mathcal{H}_k \right) U(t_{j-1}) \cdots U(t_1) U(0) \right] \right), \quad (11)$$

where N_T is the number of time steps, and the terms $U(0), U(t_1), \dots, U(t_{N_T-1}), U(T)$ represent the time-evolution operators at different steps.

The optimization process in GRAPE involves iteratively updating the control fields $u_k(t)$ using a gradient-based optimization algorithm, such as gradient ascent or conjugate gradient. The update rule can be written as

$$u_k^{(i+1)}(t_j) = u_k^{(i)}(t_j) + l_r \frac{\partial \Phi}{\partial u_k(t_j)}, \quad (12)$$

where l_r is the learning rate and i denotes the iteration index.

GRAPE has been successfully employed across a wide range of quantum control tasks, including quantum gate synthesis, state preparation, and dynamical decoupling, owing to its computational efficiency and flexibility. It is considered a cornerstone in quantum optimal control. However, as will be demonstrated in the next subsection, GRAPE performs poorly in our context: It fails to achieve high fidelity even when optimizing sequences that are longer than those produced by our JK algorithm. This underscores the need for tailored optimization strategies that can effectively navigate the constrained and highly structured landscape of EO qubit systems.

C. Krotov's method

Optimization algorithms can generally be categorized into gradient-free and gradient-based methods, depending on whether they rely solely on the evaluation of the cost functional or also incorporate its gradients. Among gradient-based approaches, Krotov's method has gained widespread use in quantum optimal control applications [55,56]. Its principal advantage lies in its suitability for problems involving continuous-time control sequences, a feature that aligns well with the sequential nature of our task [57,58]. Unlike GRAPE, Krotov's method ensures monotonic convergence of the fidelity under certain conditions, which makes it especially attractive for constrained control landscapes.

To minimize the total number of pulses required for implementing a gate, it is essential to design an efficient structure of control channels, thereby improving control-space utilization. Notably, the commutation relation $[\mathcal{H}_{i,j}, \mathcal{H}_{k,l}] = 0$ holds when all indices i, j, k, l are distinct. This observation enables the use of an interleaving strategy, in which the control Hamiltonian alternates between two nonoverlapping sets of interactions, defined as

$$\mathcal{H}(t) = \begin{cases} \mathcal{H}_{\text{odd}}, & \text{for odd time steps,} \\ \mathcal{H}_{\text{even}}, & \text{for even time steps,} \end{cases} \quad (13)$$

where $\mathcal{H}_{\text{even}} = J_{1,2}(t) \mathbf{S}_1 \cdot \mathbf{S}_2 + J_{3,4}(t) \mathbf{S}_3 \cdot \mathbf{S}_4 + J_{5,6}(t) \mathbf{S}_5 \cdot \mathbf{S}_6 + J_{7,8}(t) \mathbf{S}_7 \cdot \mathbf{S}_8$, and $\mathcal{H}_{\text{odd}} = J_{2,3}(t) \mathbf{S}_2 \cdot \mathbf{S}_3 + J_{4,5}(t) \mathbf{S}_4 \cdot \mathbf{S}_5 + J_{6,7}(t) \mathbf{S}_6 \cdot \mathbf{S}_7 + J_{8,9}(t) \mathbf{S}_8 \cdot \mathbf{S}_9$ (cf. Fig. 6). These

Hamiltonians are written in computational order. This interleaving exploits the commutativity of spatially disjoint exchange terms, enabling simultaneous updates in nonoverlapping qubit pairs and effectively multiplying control channel utilization.

The objective function of Krotov's method in this work is given by

$$J_{\text{obj}}(\{\phi_k^{(i)}(t)\}, \{J_{l,l+1}^{(i)}(t)\}) \\ = J_T(\{\phi_k^{(i)}(t)\}) + \sum_l \int_0^T g_a(J_{l,l+1}^{(i)}(t)) dt, \quad (14)$$

where $\{\phi_k^{(i)}(t)\}$ are the forward-propagated states initialized from $\{\phi_k\}$ under the control fields $\{J_{l,l+1}^{(i)}(t)\}$ at iteration i . For a three-qubit gate, the set $\{\phi_k\}$ corresponds to all eight logical basis states: $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$, and $|111\rangle$.

The first part of the objective function is a final time-related term J_T , which plays major role in J_{obj} . A straightforward choice of J_T for our task is the total state infidelity:

$$J_{T,\text{re}} = 1 - \frac{1}{N} \text{Re} \left[\sum_{k=1}^N \tau_k \right], \quad (15)$$

with $\tau_k = \langle \phi_k^{\text{target}} | \phi_k(T) \rangle$, $|\phi_k^{\text{target}}\rangle = U_{\text{gate}} |\phi_k\rangle$, and N being the dimension of the logical subspace.

The second part of J_{obj} is the running cost on the control field. A commonly used expression of g_a is

$$g_a(J_{l,l+1}^{(i)}(t)) = \frac{\lambda_l}{S_l(t)} (J_{l,l+1}^{(i)}(t) - J_{l,l+1}^{(i-1)}(t))^2, \quad (16)$$

where the inverse step width $\lambda_l > 0$ and the update shape function $S_l(t)$ can be any user-defined function whose values range between 0 and 1.

After setting the initial guess $J_{l,l+1}^{(0)}(t)$, the optimized field $J_{l,l+1}^{(i)}(t)$ in iteration i is updated as follows:

$$J_{l,l+1}^{(i)}(t) = J_{l,l+1}^{(i-1)}(t) + \Delta J_{l,l+1}^{(i)}(t), \quad (17)$$

where

$$\Delta J_{l,l+1}^{(i)}(t) = \frac{S_l(t)}{\lambda_l} \text{Im} \left[\sum_{k=1}^N \left\langle \chi_k^{i-1}(t) \left| \left(\frac{\partial \hat{\mathcal{H}}}{\partial J_{l,l+1}} \right)_{(i)} \right| \phi_k^{(i)}(t) \right\rangle \right], \quad (18)$$

with the forward propagation equation of motion for $|\phi_k^{(i)}(t)\rangle$ given by

$$\frac{\partial}{\partial t} |\phi_k^{(i)}(t)\rangle = -\frac{i}{\hbar} \hat{\mathcal{H}}(t) |\phi_k^{(i)}(t)\rangle. \quad (19)$$

The costates $|\chi_k^{i-1}(t)\rangle$ are propagated backward in time under the guess controls of iteration $i-1$, satisfying

$$\frac{\partial}{\partial t} |\chi_k^{i-1}(t)\rangle = -\frac{i}{\hbar} \hat{\mathcal{H}}^{(i-1)\dagger} |\chi_k^{i-1}(t)\rangle, \quad (20)$$

with the boundary condition

$$|\chi_k^{i-1}(T)\rangle = -\frac{\partial J_T}{\partial \langle \phi_k(T) \rangle} \Big|_{(i-1)}. \quad (21)$$

ALGORITHM 1. Krotov's method.

```

1: set initial  $\{J_{l,l+1}^{(0)}(t)\}, \{\phi_k\}, \{S_{l,n}\}, \{\lambda_l\}$ , forward propagator  $U$ ,
   backward propagator  $U^\dagger$ , infidelity threshold  $\epsilon = 10^{-8}$ 
2:  $i \leftarrow 0$ 
3: allocate forward storage array  $\Phi_0[1 \dots N(=24), 0 \dots N_T]$ 
4: for  $k \leftarrow 1, \dots, N$  do
5:    $\Phi_0[k, 0] \leftarrow \phi_k^{(0)}(t_0) \leftarrow \phi_k$ 
6:   for  $n \leftarrow 0, \dots, N_T$  do
7:      $U(t_n) \leftarrow H(t_n) \leftarrow J_{l,l+1}^{(0)}(t_n)$ 
8:      $\Phi_0[k, n+1] \leftarrow \phi_k^{(0)}(t_{n+1}) \leftarrow U(\phi_k^{(0)}(t_n))$ 
9:   end for
10: end for
11:  $J_{T,\text{re}} \leftarrow \Phi_0[k, N_T]$ 
12: while not converged do
13:   allocate backward storage array  $X[1 \dots N, 0 \dots N_T]$ 
14:   for  $k \leftarrow 1, \dots, N$  do
15:      $X[k, N_T] \leftarrow \chi_k^{(i)}(T) \leftarrow \phi_k^{\text{target}}/(2N)$ 
16:     for  $n \leftarrow N_T, \dots, 0$  do
17:        $U^\dagger(t_n) \leftarrow H^\dagger(t_n) \leftarrow J_{l,l+1}^{(i)}(t_n)$ 
18:        $X[k, n-1] \leftarrow \chi_k^{(i)}(t_{n-1}) \leftarrow U^\dagger(\chi_k^{(i)}(t_n))$ 
19:     end for
20:   end for
21:   allocate forward storage array  $\Phi_1[1 \dots N, 0 \dots N_T]$ 
22:    $\forall k : \Phi_1[k, 0] \leftarrow \phi_k^{(i+1)}(t_0) \leftarrow \phi_k$ 
23:   for  $n \leftarrow 0, \dots, N_T$  do
24:      $\forall k : \chi_k^{(i)}(t_n) = X[k, n]$ 
25:      $\forall l : \Delta J_{l,l+1}^{(i)}(t_n) \leftarrow \frac{S_{l,n}}{\lambda_l} \text{Im} \sum_k \langle \chi_k^{(i)}(t_n) | \frac{\partial H}{\partial J_{l,l+1}} | \phi_k^{(i+1)}(t_n) \rangle$ 
26:      $\forall l : J_{l,l+1}^{(i+1)}(t_n) \leftarrow J_{l,l+1}^{(i)}(t_n) + \Delta J_{l,l+1}^{(i)}(t_n)$ 
27:   end for
28:    $U(t_n) \leftarrow H(t_n) \leftarrow J_{l,l+1}^{(i+1)}(t_n)$ 
29:    $\forall k : \Phi_1[k, n+1] \leftarrow \phi_k^{(i+1)}(t_{n+1}) \leftarrow U(\phi_k^{(i+1)}(t_n))$ 
30: end for
31:  $\Phi_0 \leftarrow \Phi_1$ 
32:  $J_{T,\text{re}} \leftarrow \Phi_1[k, N_T]$ 
33: if  $J_{T,\text{re}} < \epsilon$  then break
34: end if
35:  $i \leftarrow i + 1$ 
36: end while
37:  $\{J_{l,l+1}^{\text{opt}}(t)\} \leftarrow \{J_{l,l+1}^{(i)}(t)\}$ 

```

For our specific $J_T = J_{T,\text{re}}$, this boundary condition gives

$$|\chi_k(T)\rangle = \frac{1}{2N} |\phi_k^{\text{target}}\rangle, \quad (22)$$

which is independent of $\phi_k(T)$. The pseudocode of this procedure is shown in Algorithm 1.

Figure 8 demonstrates that when the total number of time steps N_T is restricted to a relatively small value, both GRAPE and Krotov's methods perform poorly in the Toffoli gate search task and fail to achieve low infidelity. Here, we set $N_T = 25$ for both algorithms. Because each time step contains four exchange unitaries (cf. Fig. 6), this leads to a 100-pulse sequence—already longer than the 92-pulse sequence found by our JK algorithm. However, as illustrated in Fig. 8, after 100 iterations, the gate infidelity for GRAPE only approaches around 0.3, while Krotov's method achieves better results but still plateaus at approximately 0.22. Both results are far from the infidelity achieved by the 92-pulse sequence found by our

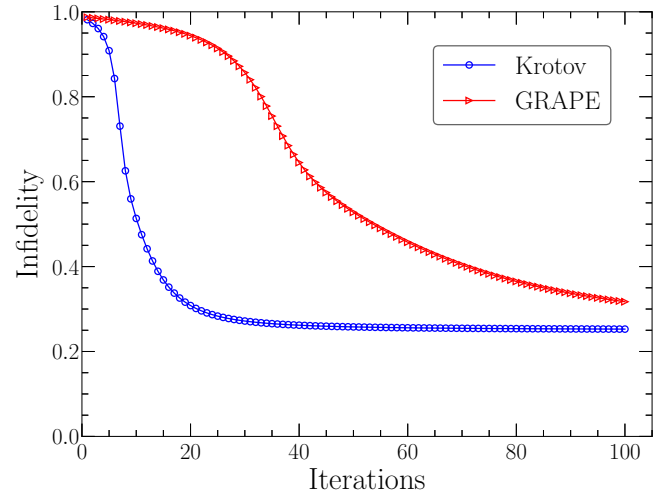


FIG. 8. Fidelity convergence of GRAPE and Krotov's methods in the task of synthesizing a Toffoli gate using exchange unitaries. The optimization for both GRAPE and Krotov's methods was carried out over 100 iterations with $N_T = 25$, which is too short to achieve lower infidelity. The learning rate for GRAPE was set to $l_r = 0.02$.

JK algorithm, which indicates that a single gradient-based algorithm alone is insufficient for this task under constrained N_T . In contrast, our JK algorithm circumvents this limitation by first performing gradient-based optimization with a much larger N_T (e.g., $N_T = 55$, resulting in a 220-pulse sequence), and then pruning most of the unnecessary unitaries. Without the restriction of small N_T , it is always possible to use gradient methods (mainly Krotov's method in this work) to optimize from multiple random initializations. If, for a large number of random initializations, it remains difficult to reduce the infidelity below the required threshold of 10^{-8} , then N_T can be further increased until solutions meeting the threshold can be reliably found with only a small number of random initializations.

D. Jenga-Krotov algorithm

Krotov's method, discussed in the previous section, has proven valuable for quantum-gate optimization. Yet our preliminary tests reveal that, when the final-gate infidelity is used directly as the cost functional, the original algorithm exhibits serious shortcomings for our problem. Because Krotov's update scheme is strictly local in parameter space, its performance is highly sensitive to the initial guess and it frequently becomes trapped in poor local minima, leaving the residual infidelity unacceptably high for practical use.

To overcome these limitations, we introduce the JK algorithm, which retains the strengths of Krotov's framework while adding two complementary strategies that promote global convergence.

1. *Parameter-space expansion.* We first lengthen the initial pulse sequence, giving the optimizer a higher-dimensional search space in which to escape shallow minima and locate a configuration whose final infidelity falls below a chosen threshold. One practical approach is to initialize with a randomly generated, sufficiently long pulse train and use the Krotov algorithm to iteratively adjust it toward the

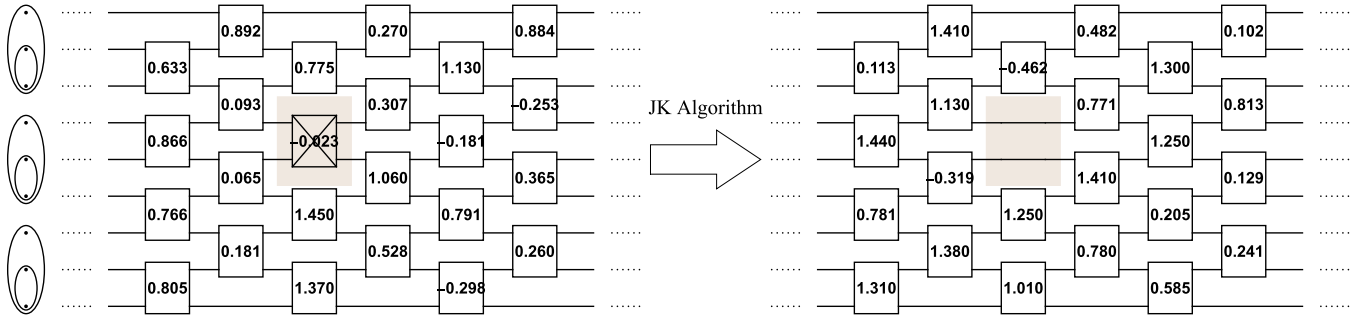


FIG. 9. Illustration of the JK algorithm. The left panel shows an initial dense pulse sequence, where one exchange unitary (highlighted by the shaded box) is randomly selected and removed. The sequence is then reoptimized, allowing the durations of the remaining exchange unitaries (indicated by the numbers in the boxes) to adjust. The right panel shows the resulting optimized sequence, now with one fewer exchange unitary, but maintaining the same level of gate fidelity. This iterative pruning-and-refinement process is key to the JK algorithm’s ability to yield compact, high-fidelity pulse sequences.

target unitary. This step encourages exploration of wider neighborhoods in the control landscape, avoiding barren plateaus.

2. *Systematic pulse reduction.* Once a high-fidelity pulse sequence is obtained, we initiate a pruning process: Individual exchange unitaries (treated as a “brick”) are tentatively removed, one at a time, and the resulting sequence is reoptimized. During this reoptimization, the locations of all remaining exchange unitaries are kept fixed, but the durations of the corresponding exchange unitaries are allowed to vary. An exchange unitary is eliminated if its removal increases the final gate infidelity by less than a predetermined threshold. By iterating this “brick-removal” process (as illustrated in Fig. 9), we obtain a significantly more compact pulse sequence that retains the fidelity of its denser predecessor.

The name Jenga-Krotov reflects the core strategy of the algorithm: Much like the game Jenga, which begins with a fully constructed tower and gradually removes blocks without compromising its stability, the JK algorithm starts with a dense, high-fidelity pulse sequence and systematically prunes away redundant exchange unitaries while maintaining the target performance.

This two-stage approach, i.e., first expanding the search space to escape local minima, then reducing complexity through guided pruning, not only mitigates the limitations of local gradient optimization but also significantly lowers computational overhead. Using JK, we are able to generate a Toffoli sequence that meets practical fidelity thresholds with substantially fewer exchange unitaries than those produced by conventional Krotov or GRAPE methods.

To address the issue of sensitivity to initialization in the Krotov method, we systematically investigated different parameter settings, including step size and other hyperparameters, and explored adaptive tuning strategies. Our results show that while such adjustments can marginally improve performance, they do not fundamentally resolve the sensitivity inherent to the optimization landscape. Notably, we found that expanding the parameter space by increasing the sequence length N_T is a much more effective approach. By dynamically adjusting N_T based on preliminary optimization results, the Krotov method is more likely to reach a global optimum from various random initializations, as evidenced by achieving infidelity below a threshold. This reflects the existence of multiple

global optima corresponding to different pulse sequences that can implement the same quantum gate.

Our strategy involves incrementally increasing N_T and conducting a preoptimization sampling, where we run a limited number of Krotov iterations from several random initializations. If most runs fail to reach the desired infidelity, it signals the need for a larger N_T . This procedure is computationally efficient and enables rapid identification of suitable sequence lengths.

However, simply increasing N_T often leads to unnecessarily long and inefficient gate sequences. To address this, our JK algorithm incorporates a postoptimization pruning step, which refines the raw sequence into a more compact and efficient form. The combination of adaptive expansion of the parameter space and subsequent pruning is central to the robustness and efficiency of our approach, distinguishing the JK algorithm from straightforward modifications of the Krotov method.

The complete procedure is summarized in Algorithm 2.

IV. RESULTS

A. JK-based efficient pulse optimization for Toffoli gate

In this section, we present our results on efficiently decomposing the Toffoli gate using the JK algorithm. We begin by generating random initial control pulses that fully populate a dense, overcomplete control structure—activating all allowable exchange couplings across both spatial and temporal channels, regardless of whether they ultimately contribute to the target gate. This initial sequence comprises 192 exchange unitaries over 55 time steps. Krotov’s method is then applied to optimize the pulse parameters, driving the system toward a Toffoli operation; however, it does not alter the sequence length or structure. The resulting high-fidelity but uncompressed sequence is detailed in the Supplemental Material [53]. Although this uncompressed sequence successfully realizes the desired gate, its length and complexity render it impractical for direct use in scalable architectures.

To address this, we applied the JK algorithm to systematically remove pulses and further optimize the pulse parameters. All optimizations were performed on a workstation equipped with an Intel Core Ultra 7 155H processor (1.40 GHz, 16 cores) and 32 GB RAM, with a typical total computational time of about 20 h. This process resulted in a significantly

ALGORITHM 2. Jenga-Krotov algorithm.

```

1: Initialize: Set a long initial pulse sequence of length  $L$  and an
   infidelity threshold  $\epsilon$ .
2: Use Krotov's method to find the optimal  $\{J_{i,l+1}^L(t)\}$  such that the
   infidelity is less than  $\epsilon$ . If this fails, return to step 1 with a larger  $L$ .
3: Allocate a  $4 \times L$  matrix  $M$  with all elements set to  $-1$ ,
   indicating unexplored entries. At this stage, the number of  $-1$ 
   in  $M$  is  $4L$ , which equals the number of exchange unitaries  $N_{\text{ex}}$ .
4: while there exists  $e = -1$  in  $M$  do
5:   Randomly select a position  $(i, j)$  such that  $M_{i,j} = -1$ .
6:    $(i, j)$  refers to the  $i$ th control at time step  $j$ .
7:   Set  $J_{i,i+1}^L(t_j) = 0$ .
8:   Use Krotov's method to find a new optimum  $\{J_{i,l+1}^L(t)\}$ .
9:   if  $J_{T,\text{re}}(\{J_{i,l+1}^L(t)\}) < \epsilon$  then
10:     $\{J_{i,l+1}^L(t)\} \leftarrow \{J_{i,l+1}^L(t)\}$ 
11:     $M_{i,j} \leftarrow 0$ 
12:     $N_{\text{ex}} \leftarrow N_{\text{ex}} - 1$ 
13:   else
14:     $M_{i,j} \leftarrow 1$ 
15:   end if
16: end while
17:  $N_{\text{ex}}^{\text{opt}} \leftarrow N_{\text{ex}}$ 
18: for  $j = 1$  to  $L$  do
19:   if  $M_{i,j} = 0$  for all  $i$  then
20:     $L \leftarrow L - 1$ 
21:   end if
22: end for
23:  $L^{\text{opt}} \leftarrow L$ 
24:  $\{J_{i,l+1}^{\text{opt}}(t)\} \leftarrow \{J_{i,l+1}^L(t)\}$ 

```

refined sequence consisting of 92 pulses distributed across 50 time steps, as detailed in the Appendix. The optimized sequence is visually represented in Fig. 10, where it is divided into five sections based on their active qubit subsets and functional roles:

U_1 : A unitary acting on qubits A and B, spanning time steps 1–15, consisting of 24 pulses.

U_2 : A unitary acting on all three qubits, spanning time steps 16–27, consisting of 29 pulses.

U_3 : A unitary acting on qubits B and C, spanning time steps 28–38, consisting of 18 pulses.

U_4 : A unitary acting on all three qubits, spanning time steps 39–40, consisting of 6 pulses.

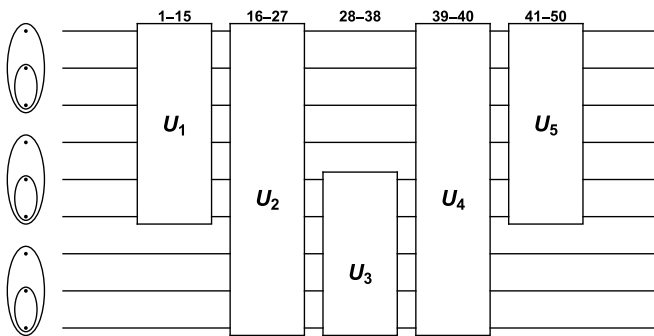


FIG. 10. Illustration of the final optimized pulse sequence for the Toffoli gate obtained by the JK algorithm in five sections of unitaries.

TABLE III. Comparison of key properties between DIR sequence and the JK-optimized sequence.

Property	DIR sequence	JK sequence
Total time steps	162	50
Total exchange unitaries	216	92
Maximum J/J_0	1	1.342
Minimum J/J_0	$-1/2$	-1.592

U_5 : A unitary acting on qubits A and B, spanning time steps 41–50, consisting of 15 pulses.

The final unitary operation implemented by this sequence is expressed as

$$U_{\text{final}} = U_5 U_4 U_3 U_2 U_1, \quad (23)$$

where each segment U_i is defined as the product of individual unitary operations over its respective time steps:

$$U_1 = \prod_{k=15}^1 U^{(k)}, \quad U_2 = \prod_{k=27}^{16} U^{(k)}, \quad U_3 = \prod_{k=38}^{28} U^{(k)},$$

$$U_4 = \prod_{k=40}^{39} U^{(k)}, \quad U_5 = \prod_{k=50}^{41} U^{(k)}, \quad (24)$$

and the individual unitary operations $U^{(k)}$ are defined as

$$U^{(k)} = \begin{cases} e^{-i\pi(p_{2,3}^{(k)} \mathcal{H}_{2,3}^{\text{ex}} + p_{4,5}^{(k)} \mathcal{H}_{4,5}^{\text{ex}} + p_{6,7}^{(k)} \mathcal{H}_{6,7}^{\text{ex}} + p_{8,9}^{(k)} \mathcal{H}_{8,9}^{\text{ex}})} & (\text{for odd } k), \\ e^{-i\pi(p_{1,2}^{(k)} \mathcal{H}_{1,2}^{\text{ex}} + p_{3,4}^{(k)} \mathcal{H}_{3,4}^{\text{ex}} + p_{5,6}^{(k)} \mathcal{H}_{5,6}^{\text{ex}} + p_{7,8}^{(k)} \mathcal{H}_{7,8}^{\text{ex}})} & (\text{for even } k), \end{cases} \quad (25)$$

with

$$\mathcal{H}_{i,j}^{\text{ex}} = \frac{1}{4} \sigma_i \cdot \sigma_j - \frac{1}{4} \quad (i, j = 1, 2, 3). \quad (26)$$

Here, the labels are in the computational order. The 92-pulse sequence obtained via the JK algorithm offers significant advantages over the DIR sequence. As summarized in Table III, the JK approach yields a markedly shorter and more efficient pulse sequence for realizing the Toffoli gate, substantially reducing both the total number of exchange unitaries and time steps while preserving high fidelity. In our optimization, the maximum and minimum values of the exchange coupling J are selected to match the experimentally accessible range for exchange-only qubits. Modern semiconductor quantum dot devices allow J to be tuned from below 1 MHz up to approximately 10 GHz [35]. Our simulations ensure that all J values remain within these practical limits, with the minimum J set to a small finite value for numerical stability. For example, with $J_0 = 100$ MHz, each time step in the JK sequence is approximately 31.4 ns, and a 50-step sequence results in a total gate time of about 1.57 μs . This duration is well below the typical decoherence time (T_2^*) of exchange-only qubits, which often exceeds 2 μs with advanced control techniques. Thus, all pulse amplitudes and gate durations used in our optimization are compatible with current experimental capabilities. These improvements demonstrate the JK algorithm's effectiveness in addressing the inefficiencies of direct decomposition methods

and affirm its potential for broader applications in high-performance quantum control in EO qubit systems.

To further illustrate the flexibility and efficiency of the JK algorithm, we have extended our study to multiqubit gates and complex quantum circuits, such as the quantum Fourier transform (QFT) circuit and the Fredkin gate. Direct decomposition methods for these circuits typically result in long pulse sequences with a large number of time steps and exchange unitaries—for example, the QFT circuit requires 180 time steps and 237 exchange unitaries, while the Fredkin gate requires 200 time steps and 276 exchange unitaries. In contrast, the JK algorithm can directly optimize the entire circuit, yielding significantly shorter and more efficient pulse sequences: only 80 time steps and 202 exchange unitaries for the QFT circuit, and 104 time steps and 172 exchange unitaries for the Fredkin gate. This demonstrates that the JK algorithm can efficiently optimize pulse sequences for complex quantum algorithms and multiqubit gates, not just individual cases. Detailed results and comparisons are provided in the Supplemental Material [53] and our GitHub repository [59].

B. Performance under noisy environment

In this section, we evaluate the robustness of the optimized 92-pulse Toffoli gate sequence under realistic noise conditions. Our noise model incorporates both control-dependent and quasistatic charge noise [60–63], which are characterized by fluctuations in the exchange coupling strength:

$$p_{i,i+1} \rightarrow (1 + \alpha)p_{i,i+1}, \quad (27)$$

where α represents a quasistatic noise parameter that is assumed to be identical for all $p_{i,i+1}$ in a given realization. We also consider crosstalk effects between neighboring spins, modeled by the transformation

$$p_{i,i+1} \mathcal{H}_{i,i+1}^{\text{ex}} \rightarrow p_{i,i+1} [\mathcal{H}_{i,i+1}^{\text{ex}} + \beta(\mathcal{H}_{i-1,i}^{\text{ex}} + \mathcal{H}_{i+1,i+2}^{\text{ex}})], \quad (28)$$

where β represents the strength of crosstalk, which is also assumed to be identical for all $p_{i,i+1}$ in a given realization.

In our analysis, we model α and β as random variables drawn from normal distribution $N(\bar{\alpha}, 0.1\bar{\alpha})$ and $N(\bar{\beta}, 0.1\bar{\beta})$, respectively, where the mean value $\bar{\alpha}$ and $\bar{\beta}$ are varied from 10^{-8} to 10^{-1} .

Typical magnitudes of quasistatic charge noise and crosstalk in exchange-only qubit systems have been well characterized in recent studies. Quasistatic charge noise is usually modeled as Gaussian fluctuations in the exchange coupling, with standard deviation σ_J commonly in the range 0.01–0.1J [64]. Current devices can achieve $\sigma_J \approx 0.01J$, though higher values may occur. This noise significantly affects gate fidelity, especially for longer sequences, making optimized pulse design essential. Crosstalk, arising from unwanted exchange interactions between neighboring qubits, is typically below 0.5% of the maximal exchange coupling and can be further reduced to below 0.1% with advanced compensation and pulse engineering [65]. At these levels, the impact on gate fidelity is negligible. Continued improvements in device design and control are expected to further suppress both charge noise and crosstalk.

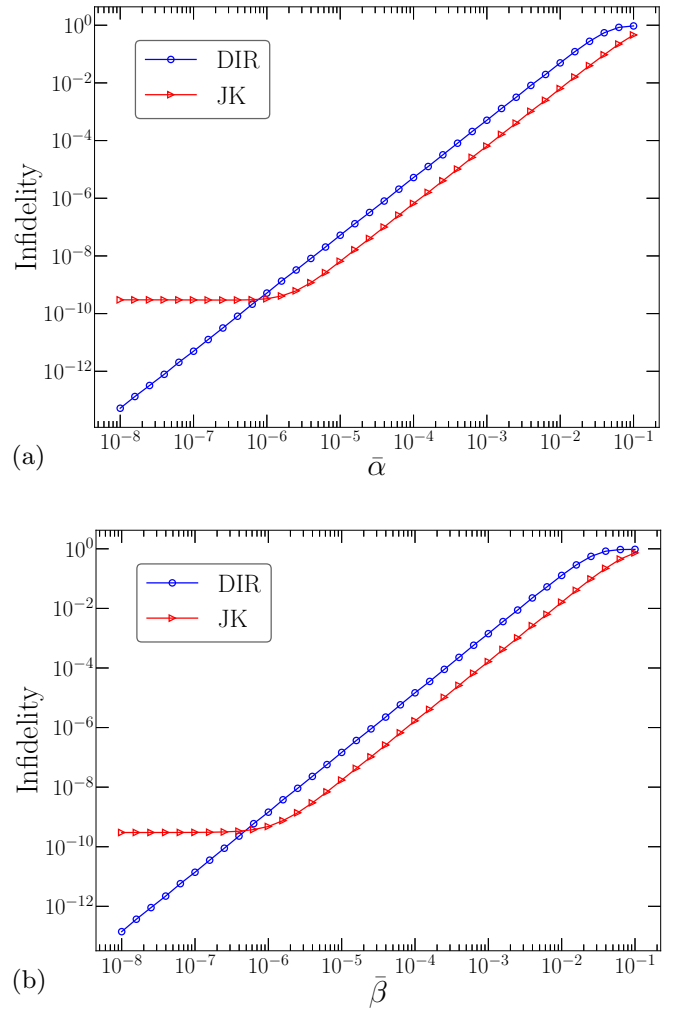


FIG. 11. Comparison of the infidelity from DIR and JK sequences as functions of (a) the mean value of charge noise α and (b) the mean value of gate crosstalk β .

To quantify the effect of noise, we compute the gate fidelity using the standard formula

$$F = \frac{d + |\text{Tr}(U_{\text{ideal}}^\dagger U_{\text{actual}})|^2}{d(d+1)}, \quad (29)$$

where U_{ideal} denotes the target Toffoli operation, U_{actual} is the noisy implementation of the JK sequence, and $d = 24$ is the total dimension of the three 8×8 blocks.

The noise robustness of the gate sequences is demonstrated in Fig. 11, where panel (a) shows the effect of charge noise and panel (b) illustrates the effect of crosstalk noise. In both panels, the blue curves correspond to the DIR sequence, while the red curves represent the JK sequence. For each mean value of the noise strength parameter [α in panel (a) and β in panel (b)], the infidelity is obtained by averaging over 100 Monte Carlo samples, where the noise values are sampled from their respective normal distributions.

As the noise strength increases, both sequences exhibit a monotonic rise in infidelity, reflecting their sensitivity to control imperfections or crosstalk. Notably, when the noise strength exceeds 10^{-6} , the JK sequence (red curve)

TABLE IV. Optimal compilation of Toffoli gate: a sequence with 92 exchange unitaries in 50 time steps.

Step	$p_{1,2}$	$p_{2,3}$	$p_{3,4}$	$p_{4,5}$	$p_{5,6}$	$p_{6,7}$	$p_{7,8}$	$p_{8,9}$
1	0	-0.361 356	0	0	0	0	0	0
2	0	0	0.466 94	0	0	0	0	0
3	0	0.474 996	0	-0.680 8	0	0	0	0
4	0	0	0.463 992	0	0.683 111	0	0	0
5	0	-0.921 766	0	-0.352 727	0	0	0	0
6	0	0	-0.647 173	0	-1.290 74	0	0	0
7	0	0	0	-0.783 947	0	0	0	0
8	0	0	0.289 479	0	0.463 229	0	0	0
9	0	0.236 16	0	-1.032 36	0	0	0	0
10	0	0	-0.517 465	0	-0.341	0	0	0
11	0	0.445 978	0	-0.591 699	0	0	0	0
12	0	0	0.757 532	0	1.292 82	0	0	0
13	0	0	0	-0.580 782	0	0	0	0
14	0	0	0	0	-0.355 991	0	0	0
15	0	0	0	-0.547 022	0	0	0	0
16	0	0	-0.761 73	0	0	0	-0.401 742	0
17	0	0.400 511	0	0	0	0.561 134	0	-0.985 982
18	0	0	-0.490 934	0	0	0	-0.490 32	0
19	0	0	0	0.613 639	0	-0.240 236	0	0.782 226
20	0	0	-0.562 063	0	0	0	-0.574 828	0
21	0	0	0	0.708 219	0	-1.472 77	0	-0.025 134 9
22	0	0	0.448 977	0	-0.754 719	0	0	0
23	0	0.780 178	0	0	0	-0.409 208	0	0
24	0	0	-0.279 016	0	0	0	-1.079 02	0
25	0	-0.585 912	0	0	0	-0.313 122	0	0
26	0	0	-1.077 95	0	0.812 474	0	0.370 976	0
27	0	0.768 569	0	0	0	0.201 01	0	-1.447 53
28	0	0	0	0	0	0	-0.571 905	0
29	0	0	0	0	0	1.318 95	0	0.349 844
30	0	0	0	0	0.724 231	0	-0.647 533	0
31	0	0	0	0	0	1.830 8	0	0
32	0	0	0	0	0	0	1.341 69	0
33	0	0	0	0	0	0	0	0.428 598
34	0	0	0	0	0.481 629	0	-0.807 275	0
35	0	0	0	0	0	-1.592 16	0	0
36	0	0	0	0	0.650 856	0	-0.028 411 5	0
37	0	0	0	-0.692 437	0	1.291 84	0	0.450 61
38	0	0	0.510 52	0	0	0	-0.529 812	0
39	0	0.610 313	0	-0.089 599 8	0	0.577 097	0	0
40	0	0	-0.538 592	0	-0.649 904	0	1.469 32	0
41	0	0.671 8	0	-0.648 449	0	0	0	0
42	0	0	0.586 782	0	-0.353 683	0	0	0
43	0	0.669 013	0	0.378 554	0	0	0	0
44	0	0	0	0	-1.185 46	0	0	0
45	0	0	0	-1.442 03	0	0	0	0
46	0	0	-1.016 47	0	0	0	0	0
47	0	0	0	-0.617 632	0	0	0	0
48	0	0	0.823 515	0	-0.376 096	0	0	0
49	0	-0.475 713	0	1.317 82	0	0	0	0
50	0	0	-0.556 544	0	0	0	0	0

consistently yields lower infidelity than the DIR sequence (blue curve), indicating its superior noise resilience across a wide range of noise levels. In the low-noise regime ($\bar{\alpha}$ or $\bar{\beta} < 10^{-6}$), the infidelity of the JK sequence saturates at approximately 10^{-10} , which corresponds to the intrinsic limit achieved during the optimization of the 92-pulse JK sequence. This plateau as $\bar{\alpha} \rightarrow 0$ or $\bar{\beta} \rightarrow 0$ indicates that optimization

precision has been reached. In contrast, the DIR sequence does not exhibit such a plateau and continues to decrease with diminishing noise.

In conclusion, the optimized JK sequence not only achieves higher fidelity and greater robustness against both charge noise and crosstalk compared to the DIR sequence, but also features a reduced pulse count and shorter operation time.

These advantages make it a more efficient and reliable choice for quantum gate implementation in semiconductor quantum dot systems, showing its potential for robust and scalable quantum computing applications.

V. CONCLUSION

In this work, we systematically constructed the angular momentum structure of a three-spin EO qubit system and identified the DFS bases. By reformulating the gate implementation problem as a quantum optimal control task, we utilized Krotov's method and its improved JK algorithm to discover a globally optimized 92-pulse sequence for the Toffoli gate. This sequence is significantly shorter than the conventional direct decomposition using single- and two-qubit gates. The noise performance of the JK-optimized sequence was analyzed under charge noise and crosstalk, demonstrating superior robustness compared to the DIR approach. Additionally, the reduced pulse count and shorter operation time further enhance its practicality, achieving higher computational efficiency and reduced cumulative error.

This study not only provides an efficient and robust solution for the Toffoli gate but also establishes a framework for designing high-fidelity quantum gates in noisy environments. The methodologies developed here can be extended to other multiqubit gates, contributing to the advancement of scalable and fault-tolerant quantum computing. The results demonstrate the potential of optimal control techniques in addressing hardware-specific challenges and optimizing quantum gate implementations.

More broadly, this work reveals a fundamental challenge in EO qubit design: The very encoding scheme that enables universal quantum computation using only exchange interactions, by mapping logical qubits into three-spin total angular momentum subspaces, also introduces substantial complexity into gate construction, particularly for multiqubit gates.

For instance, even the most efficient two-qubit CNOT sequence (e.g., the Fong-Wandzura construction) already exhibits significant operational overhead. Our analysis of the

Toffoli gate shows that such complexity scales unfavorably with system size, rendering naïve decompositions using only single- and two-qubit gates highly inefficient. This highlights an open challenge in EO architectures: how to maintain the robustness of exchange-only encoding while achieving efficient, compact realizations of multiqubit gates. Our JK algorithm provides a promising path forward, but further advances in control design and circuit optimization will be crucial for the long-term scalability of EO-based quantum computing.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grants No. 11874312 and No. 12474489), the Research Grants Council of Hong Kong (CityU 11304920), Shenzhen Fundamental Research Program (Grant No. JCYJ20240813153139050), the Guangdong Provincial Quantum Science Strategic Initiative (Grants No. GDZX2203001 and No. GDZX2403001), and the Quantum Science and Technology–National Science and Technology Major Project (Grant No. 2021ZD0302300).

DATA AVAILABILITY

The data that support the findings of this article are openly available [59].

APPENDIX: COMPILATION OF TOFFOLI GATE INTO EXCHANGE UNITARIES BY JK ALGORITHM

In this Appendix, we show the detail of our results. The unitary in step k is given by Eq. (25). The compilation of Toffoli gate into exchange unitaries by JK algorithm, which consists of 92 exchange unitaries and 50 time steps, is explicitly given in Table IV. The pulse sequence is described by a set of optimized parameters p for each segment [cf. Eq. (4)]. This parameter reflects the magnitude of the exchange operation in each segment, integrating both the interaction strength and duration.

-
- [1] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
 - [2] N. P. De Leon, K. M. Itoh, D. Kim, K. K. Mehta, T. E. Northup, H. Paik, B. Palmer, N. Samarth, S. Sangtawesin, and D. W. Steuerman, Materials challenges and opportunities for quantum computing hardware, *Science* **372**, eabb2823 (2021).
 - [3] S. Chen, J. Cotler, H.-Y. Huang, and J. Li, The complexity of NISQ, *Nat. Commun.* **14**, 6001 (2023).
 - [4] M. AbuGhanem and H. Eleuch, NISQ computers: A path to quantum supremacy, *IEEE Access* **12**, 102941 (2024).
 - [5] F. Leymann and J. Barzen, The bitter truth about gate-based quantum algorithms in the NISQ era, *Quantum Sci. Technol.* **5**, 044007 (2020).
 - [6] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, Semiconductor spin qubits, *Rev. Mod. Phys.* **95**, 025003 (2023).
 - [7] D. Loss and D. P. DiVincenzo, Quantum computation with quantum dots, *Phys. Rev. A* **57**, 120 (1998).
 - [8] M. Friesen, P. Rugheimer, D. E. Savage, M. G. Lagally, D. W. van der Weide, R. Joynt, and M. A. Eriksson, Practical design and simulation of silicon-based quantum-dot qubits, *Phys. Rev. B* **67**, 121301 (2003).
 - [9] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard, Coherent manipulation of coupled electron spins in semiconductor quantum dots, *Science* **309**, 2180 (2005).
 - [10] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, Spins in few-electron quantum dots, *Rev. Mod. Phys.* **79**, 1217 (2007).
 - [11] B. M. Maune, M. G. Borselli, B. Huang, T. D. Ladd, P. W. Deelman, K. S. Holabird, A. A. Kiselev, I. Alvarado-Rodriguez, R. S. Ross, A. E. Schmitz *et al.*, Coherent singlet-triplet oscillations in a silicon-based double quantum dot, *Nature (London)* **481**, 344 (2012).

- [12] D. M. Zajac, T. M. Hazard, X. Mi, E. Nielsen, and J. R. Petta, Scalable gate architecture for a one-dimensional array of semiconductor spin qubits, *Phys. Rev. Appl.* **6**, 054013 (2016).
- [13] J. M. Nichol, L. A. Orona, S. P. Harvey, S. Fallahi, G. C. Gardner, M. J. Manfra, and A. Yacoby, High-fidelity entangling gate for double-quantum-dot spin qubits, *npj Quantum Inf.* **3**, 3 (2017).
- [14] A. Chatterjee, P. Stevenson, S. De Franceschi, A. Morello, N. P. de Leon, and F. Kuemmeth, Semiconductor qubits in practice, *Nat. Rev. Phys.* **3**, 157 (2021).
- [15] A. J. Weinstein, M. D. Reed, A. M. Jones, R. W. Andrews, D. Barnes, J. Z. Blumoff, L. E. Euliss, K. Eng, B. H. Fong, S. D. Ha *et al.*, Universal logic with encoded spin qubits in silicon, *Nature (London)* **615**, 817 (2023).
- [16] D. Schröer, A. D. Greentree, L. Gaudreau, K. Eberl, L. C. L. Hollenberg, J. P. Kotthaus, and S. Ludwig, Electrostatically defined serial triple quantum dot charged with few electrons, *Phys. Rev. B* **76**, 075306 (2007).
- [17] K. Eng, T. D. Ladd, A. Smith, M. G. Borselli, A. A. Kiselev, B. H. Fong, K. S. Holabird, T. M. Hazard, B. Huang, P. W. Deelman *et al.*, Isotopically enhanced triple-quantum-dot qubit, *Sci. Adv.* **1**, e1500214 (2015).
- [18] J. Z. Blumoff, A. S. Pan, T. E. Keating, R. W. Andrews, D. W. Barnes, T. L. Brecht, E. T. Croke, L. E. Euliss, J. A. Fast, C. A. Jackson *et al.*, Fast and high-fidelity state preparation and measurement in triple-quantum-dot spin qubits, *PRX Quantum* **3**, 010352 (2022).
- [19] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley, Universal quantum computation with the exchange interaction, *Nature (London)* **408**, 339 (2000).
- [20] J. Levy, Universal quantum computation with spin-1/2 pairs and Heisenberg exchange, *Phys. Rev. Lett.* **89**, 147902 (2002).
- [21] 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 (2010).
- [22] J. Medford, J. Beil, J. Taylor, S. Bartlett, A. Doherty, E. Rashba, D. DiVincenzo, H. Lu, A. Gossard, and C. M. Marcus, Self-consistent measurement and state tomography of an exchange-only spin qubit, *Nat. Nanotechnol.* **8**, 654 (2013).
- [23] W. Ha, S. D. Ha, M. D. Choi, Y. Tang, A. E. Schmitz, M. P. Levendorf, K. Lee, J. M. Chappell, T. S. Adams, D. R. Hulbert *et al.*, A flexible design platform for Si/SiGe exchange-only qubits with low disorder, *Nano Lett.* **22**, 1443 (2022).
- [24] D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley, Universal fault-tolerant quantum computation on decoherence-free subspaces, *Phys. Rev. Lett.* **85**, 1758 (2000).
- [25] G. Hu, W. W. Huang, R. Cai, L. Wang, C. H. Yang, G. Cao, X. Xue, P. Huang, and Y. He, Single-electron spin qubits in silicon for quantum computing, *Intell. Comput.* **4**, 0115 (2025).
- [26] Z. Shi, C. B. Simmons, J. R. Prance, J. K. Gamble, T. S. Koh, Y.-P. Shim, X. Hu, D. E. Savage, M. G. Lagally, M. A. Eriksson, M. Friesen, and S. N. Coppersmith, Fast hybrid silicon double-quantum-dot qubit, *Phys. Rev. Lett.* **108**, 140503 (2012).
- [27] G. Cao, H.-O. Li, G.-D. Yu, B.-C. Wang, B.-B. Chen, X.-X. Song, M. Xiao, G.-C. Guo, H.-W. Jiang, X. Hu, and G.-P. Guo, Tunable hybrid qubit in a GaAs double quantum dot, *Phys. Rev. Lett.* **116**, 086801 (2016).
- [28] M. Russ and G. Burkard, Three-electron spin qubits, *J. Phys.: Condens. Matter* **29**, 393001 (2017).
- [29] B. Thorgrimsson, D. Kim, Y.-C. Yang, L. Smith, C. Simmons, D. R. Ward, R. H. Foote, J. Corrigan, D. Savage, M. Lagally *et al.*, Extending the coherence of a quantum dot hybrid qubit, *npj Quantum Inf.* **3**, 32 (2017).
- [30] A. Frees, S. Mehl, J. K. Gamble, M. Friesen, and S. Coppersmith, Adiabatic two-qubit gates in capacitively coupled quantum dot hybrid qubits, *npj Quantum Inf.* **5**, 73 (2019).
- [31] J. Medford, J. Beil, J. M. Taylor, E. I. Rashba, H. Lu, A. C. Gossard, and C. M. Marcus, Quantum-dot-based resonant exchange qubit, *Phys. Rev. Lett.* **111**, 050501 (2013).
- [32] J. M. Taylor, V. Srinivasa, and J. Medford, Electrically protected resonant exchange qubits in triple quantum dots, *Phys. Rev. Lett.* **111**, 050502 (2013).
- [33] A. C. Doherty and M. P. Wardrop, Two-qubit gates for resonant exchange qubits, *Phys. Rev. Lett.* **111**, 050503 (2013).
- [34] M. Russ, J. R. Petta, and G. Burkard, Quadrupolar exchange-only spin qubit, *Phys. Rev. Lett.* **121**, 177701 (2018).
- [35] H. Qiao, Y. P. Kandel, K. Deng, S. Fallahi, G. C. Gardner, M. J. Manfra, E. Barnes, and J. M. Nichol, Coherent multispin exchange coupling in a quantum-dot spin chain, *Phys. Rev. X* **10**, 031006 (2020).
- [36] A. Sala, J. H. Qvist, and J. Danon, Highly tunable exchange-only singlet-only qubit in a GaAs triple quantum dot, *Phys. Rev. Res.* **2**, 012062 (2020).
- [37] B. H. Fong and S. M. Wandzura, Universal quantum computation and leakage reduction in the 3-qubit decoherence free subsystem, *Quantum. Inf. Comput.* **11**, 1003 (2011).
- [38] D. Zeuch and N. E. Bonesteel, Simple derivation of the Fong-Wandzura pulse sequence, *Phys. Rev. A* **93**, 010303 (2016).
- [39] D. Zeuch and N. E. Bonesteel, Efficient two-qubit pulse sequences beyond CNOT, *Phys. Rev. B* **102**, 075311 (2020).
- [40] M. AbuGhanem, A Toffoli gate decomposition via echoed cross-resonance gates, *Quantum Stud.: Math. Found.* **12**, 24 (2025).
- [41] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
- [42] D. Maslov and G. Dueck, Reversible cascades with minimal garbage, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **23**, 1497 (2004).
- [43] P. M. Q. Cruz and B. Murta, Shallow unitary decompositions of quantum Fredkin and Toffoli gates for connectivity-aware equivalent circuit averaging, *APL Quantum* **1**, 016105 (2024).
- [44] F. Setiawan, H.-Y. Hui, J. P. Kestner, X. Wang, and S. D. Sarma, Robust two-qubit gates for exchange-coupled qubits, *Phys. Rev. B* **89**, 085314 (2014).
- [45] E. Acuna, J. D. Broz, K. Shyamsundar, A. B. Mei, C. P. Feeney, V. Smetanka, T. Davis, K. Lee, M. D. Choi, B. Boyd, J. Suh, W. Ha, C. Jennings, A. S. Pan, D. S. Sanchez, M. D. Reed, and J. R. Petta, Coherent control of a triangular exchange-only spin qubit, *Phys. Rev. Appl.* **22**, 044057 (2024).
- [46] T. D. Ladd, Hyperfine-induced decay in triple quantum dots, *Phys. Rev. B* **86**, 125408 (2012).
- [47] L. Gaudreau, G. Granger, A. Kam, G. Aers, S. Studenikin, P. Zawadzki, M. Pioro-Ladriere, Z. Wasilewski, and A. Sachrajda, Coherent control of three-spin states in a triple quantum dot, *Nat. Phys.* **8**, 54 (2012).
- [48] G. T. Hickman, X. Wang, J. P. Kestner, and S. Das Sarma, Dynamically corrected gates for an exchange-only qubit, *Phys. Rev. B* **88**, 161303 (2013).

- [49] R. W. Andrews, C. Jones, M. D. Reed, A. M. Jones, S. D. Ha, M. P. Jura, J. Kerckhoff, M. Levendorf, S. Meenehan, S. T. Merkel *et al.*, Quantifying error and leakage in an encoded Si/SiGe triple-dot qubit, *Nat. Nanotechnol.* **14**, 747 (2019).
- [50] D. Zeuch, R. Cipri, and N. E. Bonesteel, Analytic pulse-sequence construction for exchange-only quantum computation, *Phys. Rev. B* **90**, 045306 (2014).
- [51] Y. Makhlin, Nonlocal properties of two-qubit gates and mixed states, and the optimization of quantum computations, *Quantum Inf. Process.* **1**, 243 (2002).
- [52] Y. Kawano, K. Kimura, H. Sekigawa, M. Noro, K. Shirayanagi, M. Kitagawa, and M. Ozawa, Existence of the exact CNOT on a quantum computer with the exchange interaction, *Quantum Inf. Process.* **4**, 65 (2005).
- [53] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/1x1x-j1xl> for details of basis states and additional multiqubit gate and circuit results.
- [54] V. V. Shende and I. L. Markov, On the CNOT-cost of Toffoli gates, *Quantum. Inf. Comput.* **9**, 461 (2009).
- [55] D. Basilewitsch, H. Yuan, and C. P. Koch, Optimally controlled quantum discrimination and estimation, *Phys. Rev. Res.* **2**, 033396 (2020).
- [56] O. V. Morzhin and A. N. Pechen, Krotov method for optimal control of closed quantum systems, *Russ. Math. Surv.* **74**, 851 (2019).
- [57] M. H. Goerz, D. Basilewitsch, F. Gago-Encinas, M. G. Krauss, K. P. Horn, D. M. Reich, and C. P. Koch, Krotov: A Python implementation of Krotov's method for quantum optimal control, *SciPost Phys.* **7**, 080 (2019).
- [58] M. E. F. Fernandes, F. F. Fanchini, E. F. de Lima, and L. K. Castelano, Effectiveness of the Krotov method in finding controls for open quantum systems, *J. Phys. A: Math. Theor.* **56**, 495303 (2023).
- [59] J. Wu, G. He, W. Zhuo, Q. Fu, and X. Wang, Data for the manuscript "Jenga-Krotov algorithm: Efficient compilation of multiqubit gates for exchange-only qubits", <https://github.com/WuJiahao555/JK-Algorithm-in-EO-Qubit>.
- [60] X. Hu and S. Das Sarma, Charge-fluctuation-induced dephasing of exchange-coupled spin qubits, *Phys. Rev. Lett.* **96**, 100501 (2006).
- [61] J. P. Kestner, X. Wang, L. S. Bishop, E. Barnes, and S. Das Sarma, Noise-resistant control for a spin qubit array, *Phys. Rev. Lett.* **110**, 140502 (2013).
- [62] M. D. Reed, B. M. Maune, R. W. Andrews, M. G. Borselli, K. Eng, M. P. Jura, A. A. Kiselev, T. D. Ladd, S. T. Merkel, I. Milosavljevic, E. J. Pritchett, M. T. Rakher, R. S. Ross, A. E. Schmitz, A. Smith, J. A. Wright, M. F. Gyure, and A. T. Hunter, Reduced sensitivity to charge noise in semiconductor spin qubits via symmetric operation, *Phys. Rev. Lett.* **116**, 110402 (2016).
- [63] D. Keith, S. Gorman, Y. He, L. Kranz, and M. Simmons, Impact of charge noise on electron exchange interactions in semiconductors, *npj Quantum Inf.* **8**, 17 (2022).
- [64] R. E. Throckmorton and S. Das Sarma, Crosstalk- and charge-noise-induced multiqubit decoherence in exchange-coupled quantum dot spin qubit arrays, *Phys. Rev. B* **105**, 245413 (2022).
- [65] M. T. Madzik, F. Luthi, G. G. Guerreschi, F. A. Mohiyaddin, F. Borjans, J. D. Chadwick, M. J. Curry, J. Ziegler, S. Atanasov, P. L. Bavdaz *et al.*, Operating two exchange-only qubits in parallel, [arXiv:2504.01191](https://arxiv.org/abs/2504.01191).