

# Simple and High-Precision Hamiltonian Simulation by Compensating Trotter Error with Linear Combination of Unitary Operations

Pei Zeng<sup>1,\*</sup>, Jinzhao Sun<sup>2,3,†</sup>, Liang Jiang<sup>1,‡</sup> and Qi Zhao<sup>4,§</sup>

<sup>1</sup>*Pritzker School of Molecular Engineering, The University of Chicago, Illinois 60637, USA*

<sup>2</sup>*Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom*

<sup>3</sup>*Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom*

<sup>4</sup>*QICI Quantum Information and Computation Initiative, Department of Computer Science, School of Computing and Data Science, The University of Hong Kong, Pokfulam Road, Hong Kong*



(Received 2 July 2024; revised 5 November 2024; accepted 19 February 2025; published 27 March 2025)

Trotter and linear combination of unitary (LCU) operations are two popular Hamiltonian simulation methods. The Trotter method is easy to implement and enjoys good system-size dependence endowed by commutator scaling, while the LCU method admits high-accuracy simulation with a smaller gate cost. We propose Hamiltonian simulation algorithms using LCU to compensate Trotter error, which enjoy both of their advantages. By adding few gates after the  $K$ th-order Trotter formula, we realize a better time scaling than  $2K$ th-order Trotter. Our first algorithm exponentially improves the accuracy scaling of the  $K$ th-order Trotter formula. For a generic Hamiltonian, the estimated gate counts of the first algorithm can be 2 orders of magnitude smaller than the best analytical bound of fourth-order Trotter formula. In the second algorithm, we consider the detailed structure of Hamiltonians and construct LCU for Trotter errors with commutator scaling. Consequently, for lattice Hamiltonians, the algorithm enjoys almost linear system-size dependence and quadratically improves the accuracy of the  $K$ th-order Trotter. For the lattice system, the second algorithm can achieve 3 to 4 orders of magnitude higher accuracy with the same gate costs as the optimal Trotter algorithm. These algorithms provide an easy-to-implement approach to achieve a low-cost and high-precision Hamiltonian simulation.

DOI: [10.1103/PRXQuantum.6.010359](https://doi.org/10.1103/PRXQuantum.6.010359)

## I. INTRODUCTION

Hamiltonian simulation, i.e., to simulate the real-time evolution  $U(t) = e^{-iHt}$  of a physical Hamiltonian  $H = \sum_l H_l$ , is considered to be a natural and powerful application of quantum computing [1]. It can also be used as an important subroutine in many other quantum algorithms like ground-state preparation [2,3], optimization problems [4,5], and quantum linear solvers [6]. To pursue real-world applications of Hamiltonian simulation with near-term

quantum devices, we need to design feasible algorithms with small space complexity (i.e., qubit number) and time complexity (i.e., circuit depth and gate number).

One of the most natural Hamiltonian simulation methods is based on Trotter formulas [7–22], which approximate the real-time evolution of  $H = \sum_{l=1}^L H_l$  by the product of the simple evolution of its summands  $e^{-iH_l t}$ . Besides its prominent advantage of simple realization without ancillas, Trotter methods are recently rigorously shown to enjoy commutator scaling [13,20], i.e., the Trotter error is only related to the nested commutators of the Hamiltonian summands  $\{H_l\}$ . This is very helpful for the Hamiltonians with strong locality constraints. For example, when we consider  $n$ -qubit lattice Hamiltonians, the gate cost of high-order Trotter methods is almost linear to the system size  $n$ , which is nearly optimal [13]. The major drawback of the Trotter methods is its polynomial gate cost to the inversed accuracy  $1/\epsilon$ ,  $\text{Poly}(1/\epsilon)$ . This is unfavorable in many applications where high-precision simulation is demanded to obtain practical advantages over the existing classical algorithms [23].

\*Contact author: [peizeng@uchicago.edu](mailto:peizeng@uchicago.edu), [qubitpei@gmail.com](mailto:qubitpei@gmail.com)

†Contact author: [jinzhaosun.phys@gmail.com](mailto:jinzhaosun.phys@gmail.com)

‡Contact author: [liang.jiang@uchicago.edu](mailto:liang.jiang@uchicago.edu)

§Contact author: [zhaoyiqi@cs.hku.hk](mailto:zhaoyiqi@cs.hku.hk)

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

In recent years, we have seen the developments of “post-Trotter” algorithms with exponentially improved accuracy dependence [24–29]. Due to the smart choice of the expansion formulas (i.e., Taylor series [25] or Jacobi-Anger expansion [28]), these post-Trotter methods are able to catch the dominant terms in the time-evolution operator  $U(t)$  with polynomially increasing gate resources, leading to a logarithmic gate-number dependence on the accuracy requirement  $1/\varepsilon$ . Unlike Trotter methods, these advanced algorithms are not able to utilize the specific structure of Hamiltonians due to the lack of commutator-based error form. Consequently, for instance, for  $n$ -qubit lattice Hamiltonians, their gate complexities are  $\mathcal{O}(n^2)$ , which is worse than those in Trotter algorithms  $\mathcal{O}(n^{1+o(1)})$ . Furthermore, these post-Trotter algorithms require the implementation of linear combination of unitary (LCU) formulas [30,31] or block encoding of Hamiltonians [27], which often costs many ancillary qubits and multicontrolled Toffoli gates. This is still experimentally challenging in a near-term or early fault-tolerant quantum computer [32]. To reduce the hardware requirement of compiling the LCU formulas, recent studies focus on a random-sampling implementation of a LCU formula [30,33–35], where the elementary unitaries are sampled to realize the LCU formula statistically. In this case, the Hamiltonian simulation is not performed by coherently implementing the unitary  $U = e^{-iHt}$ , but is instead realized through random sampling. This method remains effective for common applications, such as estimating the properties of the final state. Similar ideas have also been studied in the ground-state preparation algorithms [36–38].

Here, we propose composite algorithms that combine the inherent advantages of Trotter and LCU methods—easy implementation, high precision, and commutator scaling—by performing the Trotter method and then compensating the Trotter error with the LCU formulas we construct. We primarily focus on the random-sampling implementation of the LCU formula [30,33–35], with the goal of estimating the properties of the target state after real-time evolution. We demonstrate that optimal performance can be achieved by allowing the Trotter circuit to handle the majority of the simulation, with the LCU method completing the remainder.

In Sec. II, we provide a summary of our construction and results, aimed at a general audience. We explain the key ideas behind the constructions with intuitive examples. For readers interested in the technical aspects, we introduce the necessary preliminary knowledge of Hamiltonian simulation in Sec. III to facilitate understanding of the technical results. Next, in Secs. IV and V, we present a detailed construction and gate complexity analysis of the two Trotter-LCU algorithms. Finally, in Sec. VI, we conclude our discussion and outline possible future directions.

## II. SUMMARY OF RESULTS

### A. General idea

The major idea of the proposed Trotter-LCU algorithm is illustrated in Fig. 1. In a normal  $K$ th-order Trotter circuit, we decompose the time evolution  $U(t) = e^{-iHt}$  to  $\nu$  segments, each with a small evolution time  $x = t/\nu$ . For consistency, we denote the 0th-order Trotter formula as  $S_0(x) = I$ . After we perform the  $K$ th-order Trotter formula  $S_K(x)$  ( $K = 0, 1, 2k, k \in \mathbb{N}_+$ ), there will be a remaining Trotter error  $V_K(x) := U(x)S_K(x)^\dagger$ , which affects the simulation accuracy. To address this problem, we introduce a random LCU formula to compensate the Trotter error using one ancilla and simple gates, which achieves a high-precision Hamiltonian simulation with low cost.

Consider the following LCU formula of an operator  $V$ ,

$$\tilde{V} = \mu \sum_{i=0}^{\Gamma-1} \text{Pr}(i) V_i, \quad (1)$$

such that the spectral norm distance  $\|V - \tilde{V}\| \leq \varepsilon$ . Here,  $\mu > 0$  is the 1-norm (i.e.,  $l_1$ -norm) of the coefficient vector,  $\text{Pr}(i)$  is a probability distribution over different unitaries  $V_i$ , and  $\{V_i\}_{i=0}^{\Gamma-1}$  is a set of unitaries. There are usually two ways to implement the LCU formula: the coherent implementation [24,25] and the random-sampling implementation [33–35]. Our major focus is on the random-sampling implementation, where we can estimate the properties of the target state  $\rho = U(t)\rho_0 U(t)^\dagger$  with only one ancillary qubit. In Appendix H, we discuss the potential use of the coherent implementation of our algorithm. In the random-sampling implementation, we can use Eq. (1) to estimate an arbitrary observable value  $O$  on  $\rho$ ,

$$\text{Tr}(O\rho) \approx \mu^2 \sum_{i,j} p_i p_j \text{Tr}(O V_i \rho_0 V_j^\dagger). \quad (2)$$

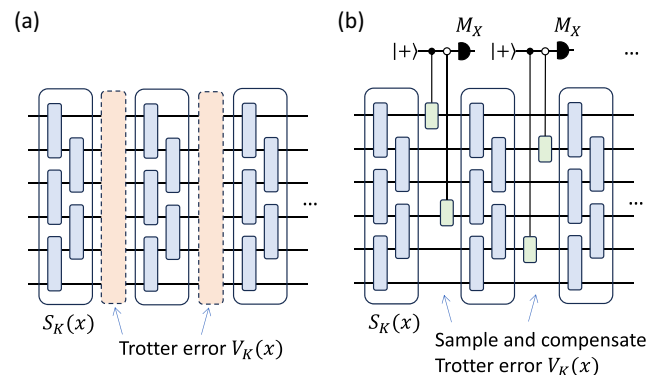


FIG. 1. (a) In the normal  $K$ th-order Trotter circuit, there will be a remaining Trotter error  $V_K(x)$  after each segment. (b) We introduce random LCU formulas to compensate  $V_K(x)$  with a single ancilla qubit and simple gates.

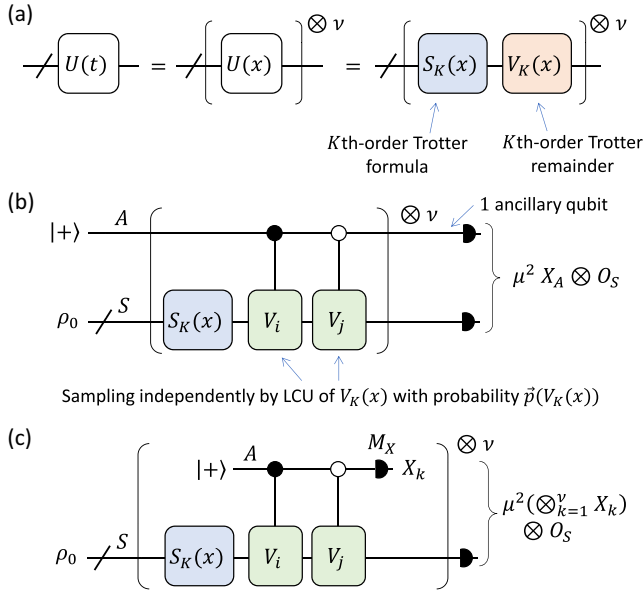


FIG. 2. (a) In the  $K$ th-order Trotter-LCU algorithm, we first implement  $K$ th-order Trotter formula, then compensate the remainder using the LCU formulas we construct. (b) Random-sampling implementation of the LCU formula. We sample the elementary unitaries  $V_i$  and  $V_j$  independently based on the LCU formula of  $V_K(x)$  and implement the controlled  $V_i$  and  $V_j$  gate. Then the Trotter remainder  $V_K(x)$  can be compensated in a Hadamard-test-type circuit. (c) A variant of the implementation where the ancillary qubit is measured and reset in each segment. The detailed sampling procedure of  $V_i$  and  $V_j$  is shown in Figs. 5 and 7.

As is shown in Fig. 2(b), since the estimation of  $\text{Tr}(O V_i \rho_0 V_j^\dagger)$  can be implemented using Hadamard-test-type circuits [39], we only need to sample  $V_i$  and  $V_j$  based on the LCU formula in Eq. (1) to estimate  $\text{Tr}(O\rho)$  with  $\varepsilon$  accuracy using  $\mathcal{O}(\mu^4/\varepsilon^2)$  sampling resource, which owns an extra  $\mu^4$  overhead compared to the normal Hamiltonian simulation algorithms [35]. To make the algorithm efficient, we need to set  $\mu$  to be a constant. We also provide a variant in Fig. 2(c) where the ancillary qubit is measured and reset in each segment, which is equivalent to Fig. 2(b) for the observable estimation. In this case, the expectation value of  $\mu^2 X_A^{(\text{tot})} \otimes O_S$  provides an unbiased estimation of  $\text{Tr}(O\rho)$  where  $X^{(\text{tot})} := \bigotimes_{k=1}^v X_k$  is the multiplication of all the ancillary measurement values. This variant reduces the need to store the ancilla qubit, simplifying the implementation on a fault-tolerant quantum computer.

The construction of an appropriate LCU formula for the  $K$ th-order Trotter remainder,  $V_K(x)$ , is crucial for developing an efficient Hamiltonian simulation algorithm. In the following subsections, we briefly introduce two approaches for constructing the LCU formula for  $V_K(x)$ . The resulting composite Trotter-LCU algorithms are referred to as paired Taylor-series compensation (PTSC)

and nested-commutator compensation (NCC), respectively. Detailed analysis and performance proofs for these two algorithms can be found in Secs. IV and V.

## B. Paired Taylor-series compensation: overview

Without loss of generality, we focus on the case of an  $n$ -qubit Hamiltonian  $H$ , which can be written as

$$H = \sum_{l=1}^L H_l = \sum_{l=1}^L \alpha_l P_l = \lambda \sum_{l=1}^L p_l P_l, \quad (3)$$

where  $\{P_l\}_l$  are different  $n$ -qubit Pauli operators. We set all the coefficients  $\{\alpha_l\}_l$  to be positive and absorb the signs into Pauli operators  $\{P_l\}_l$ .  $\lambda := \sum_l \alpha_l$  is the  $l_1$ -norm of the Hamiltonian coefficient vector. We consider the Hamiltonians where  $\lambda$  increases polynomially with respect to  $n$ .

We first consider to construct the LCU formulas for  $V_K(x)$  from Taylor-series expansion [25]. In the 0th-order case, when no Trotter formula is introduced, the Trotter remainder  $V_0(x)$  is the short-time evolution  $U(x)$  itself, which can be expanded as

$$\begin{aligned} V_0(x) &= e^{-ixH} = \sum_{s=0}^{\infty} F_{0,s}(x) = \sum_{s=0}^{\infty} \frac{(-ix)^s}{s!} H^s \\ &= e^{\lambda x} \sum_{s=0}^{\infty} \text{Poi}(s; \lambda x) \sum_{l_1, \dots, l_s} p_{l_1} \cdots p_{l_s} P_{l_1} \cdots P_{l_s}. \end{aligned} \quad (4)$$

Here,  $\text{Poi}(s; a) := e^{-a} a^s / s!$  is the Poisson distribution.  $F_{0,s}(x)$  denotes the  $s$ -order expansion term. Equation (4) illustrated in Fig. 3(a) is a LCU formula of  $V_0(x)$  with 1-norm  $\mu_x = e^{\lambda x}$ . The 1-norm of the overall evolution  $U(t) = U(x)^v$  is then  $\mu = e^{\lambda t}$ , which, unfortunately, grows exponentially with respect to  $t$  regardless of how much we increase the segment number  $v$ . This implies that the direct random-sampling implementation of the LCU formula  $V_0(x)$  in Eq. (4) is not feasible. In Ref. [25], the authors discuss the coherent implementation of  $V_0(x)$  instead and find that one can achieve good time and accuracy dependence in that scenario.

When focusing on the random-sampling implementation, we need to suppress the 1-norm  $\mu_x$  of each segment. To this end, we first consider the usage of the Trotter formula. For example, if we apply first-order Trotter formula  $S_1(x) = \prod_{l=1}^L e^{-ixH_l}$  in each segment, the first-order remainder  $V_1(x) := U(x)S_1(x)^\dagger$  can be expanded as

$$\begin{aligned} V_1(x) &= e^{-ixH} \prod_{l=L}^1 e^{ixH_l} = \sum_{s=0}^{\infty} F_{1,s}(x) \\ &= \sum_{r:r_1, r_2, \dots, r_L} \frac{(-ix)^r}{r!} H^r \prod_{l=L}^1 \frac{(ix)^{r_l}}{r_l!} H^{r_l}, \end{aligned} \quad (5)$$

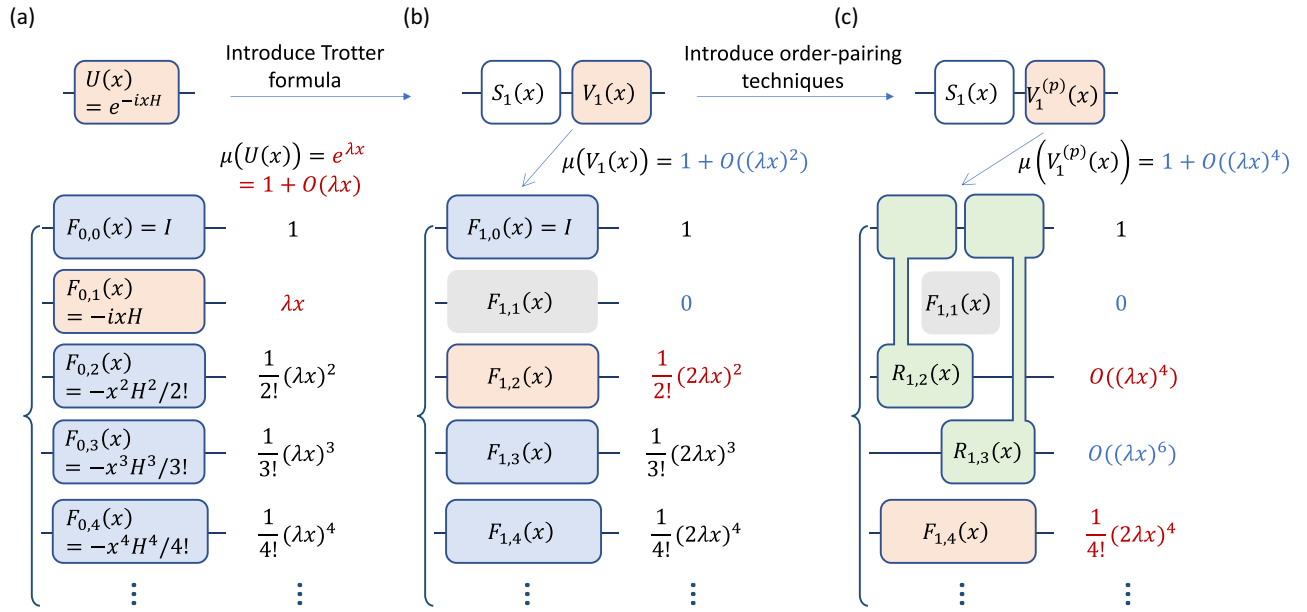


FIG. 3. Illustration of the idea of paired Taylor-series compensation (PTSC) algorithm. We take the first-order algorithm as an example. (a) Taylor-series expansion of the small-time evolution  $U(x)$ . The 1-norm is  $\lambda(U(x)) = e^{\lambda x} = 1 + \mathcal{O}(\lambda x)$ . The dominant term is contributed from the first-order expansion  $F_{0,1}(x)$ . (b) By introducing the first-order Trotter formula  $S_1(x)$ , the first-order expansion  $F_{1,1}$  in the Trotter remainder  $V_1(x)$  becomes 0. As a result, the 1-norm of  $V_1(x)$  is suppressed to  $1 + \mathcal{O}((\lambda x)^2)$ , limited by the second-order expansion term  $F_{1,2}(x)$ . (c) By further noticing that  $F_{1,2}$  and  $F_{1,3}$  are anti-Hermitian, we introduce Euler's formula in Eq. (7) to pair the leading-order terms to  $F_{1,0} := I$ . This will double the order of  $\mu_x - 1$  and further suppress the 1-norm  $\mu$  of the overall evolution  $U(t)$ .

where  $\{r_1, \dots, r_L\}$  denotes  $L$  expansion variables related to the Trotter formula and  $s := r + \sum_{l=1}^L r_l$ .  $F_{1,s}$  denotes the  $s$ -order expansion term of  $V_1(x)$ . The 1-norm of  $V_1(x)$  in Eq. (5) is  $e^{2\lambda x}$ , which seems to be even larger than the 0th-order case. However, since  $V_1(x)$  denotes the (multiplicative) Trotter error, we have  $F_{1,1}(x) = 0$ . Using this condition, we can rewrite  $V_1(x)$  as

$$V_1(x) = I + \sum_{s=2}^{\infty} F_{1,s}(x), \quad (6)$$

as illustrated in Fig. 3. From the Taylor-series expansion, we can bound the 1-norm of the new LCU formula in Eq. (6) by  $\mu_x = e^{2\lambda x} - (2\lambda x) \leq e^{(2\lambda x)^2}$ . In this way, we reduce  $\mu_x$  from  $1 + \mathcal{O}(\lambda x)$  to  $1 + \mathcal{O}((\lambda x)^2)$ . The 1-norm of the overall time evolution becomes  $\mu = \mu_x^\nu \leq \exp((2\lambda t)^2/\nu)$ . As a result, by increasing the segment number  $\nu$  – or equivalently reducing the unit evolution time  $x$  – we can decrease the 1-norm  $\mu$ , leading to a lower sampling cost. If we choose the segment number as  $\nu = \mathcal{O}((\lambda t)^2)$ ,  $\mu$  will remain constant.

From the above discussion, it is clear that, to reduce the 1-norm  $\mu$  of the overall LCU formula for  $U = e^{-iHt}$ , our main objective is to suppress the leading-order term of the 1-norm remainder  $\mu_x - 1$  for each segment, which determines the number of segments  $\nu$  and hence circuit depth when  $\mu$  is set to be a constant.

We can further reduce the 1-norm  $\mu_x$  of  $V_1(x)$  by taking advantage of the structure of Trotter errors. For an anti-Hermitian Pauli operator  $\pm iP$  where  $P \in \{I, X, Y, Z\}^{\otimes n}$ , we have the following Euler's formula,

$$I \pm iyP = \sqrt{1 + y^2} e^{\pm i\theta P}. \quad (7)$$

Here,  $\theta = \tan^{-1}(y)$  and we suppose  $0 < y < 1$ . The 1-norm of the left-hand side of Eq. (7) is  $1 + y$ , while the 1-norm of the right-hand side is  $\sqrt{1 + y^2} < 1 + y^2/2 = 1 + \mathcal{O}(y^2)$ . As a result, the exponent of  $\mu_x - 1$  is effectively doubled. In Sec. IV B, we prove that the expansion terms  $F_{1,2}$  and  $F_{1,3}$  in the LCU formula Eq. (5) are anti-Hermitian. As a result, we can further suppress  $\mu_x$  by pairing the terms in  $F_{1,s}(x)$  with  $F_{1,0} = I$  using Euler's formula in Eq. (7). When  $y = x^s$ , the paired formula  $R_{1,s}(x)$  as a summation of Pauli rotation unitaries owns the 1-norm of  $\mu_x = 1 + \mathcal{O}(x^{2s})$ , whose  $x$  dependence is doubled, as illustrated in Fig. 3(c).

To generalize the discussion, for the  $K$ th-order Trotter error  $V_K(x)$  ( $K = 2k, k \in \mathbb{N}_+$ ), we have  $F_{K,1} = \dots = F_{K,K} = 0$  [8,20]. Moreover, we can show that  $F_{K,K+1}, F_{K,K+2}, \dots, F_{K,2K+1}$  are anti-Hermitian. We call the term with  $s = K + 1$  to  $2K + 1$  the leading-order terms. The algorithm utilizing  $K$ th-order Trotter formula and the paired idea in the LCU construction is called the  $K$ th-order PTSC

algorithm. We provide the detailed algorithm description and gate-complexity analysis in Sec. IV.

The PTSC algorithm is generic for the  $L$ -sparse Hamiltonian  $H = \sum_{l=1}^L H_l = \lambda \sum_{l=1}^L P_l P_l$  with  $\lambda = \sum_l \|H_l\|$  and  $\{P_l\}$  are Pauli matrices. It can be implemented using a simple and universal classical random-sampling procedure: first, we sample the order  $s$  from the Taylor-series expansion, and then we sample a Pauli string based on the Hamiltonian coefficients. With the random-sampling implementation, we prove that by appending few gates after the  $K$ th-order Trotter formula with only one ancillary qubit, one can improve the time scaling from  $1 + 1/K$  to  $1 + 1/(2K + 1)$  and exponentially improves the accuracy scaling of the  $K$ th-order Trotter formula compared to the  $K$ th-order Trotter. We have the following theorem.

*Theorem 1.* (Informal, see Theorem 3 in Sec. IV) In a  $K$ th-order paired Taylor-series compensation algorithm ( $K = 0, 1$  or  $2k, k \in \mathbb{N}_+$ ), the gate complexity in a single round is

$$\mathcal{O}\left((\lambda t)^{1+\frac{1}{2K+1}}\left(\kappa_K L + \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)}\right)\right),$$

where  $\lambda = \sum_l \|H_l\|$ ,  $\kappa_K = K$  when  $K = 0$  or  $1$ ,  $\kappa_K = 2 \times 5^{K/2-1}$  otherwise.

From Theorem 1, we observe that setting  $K = 0$ , i.e., not using Trotter formulas, still yields a valid PTSC algorithm by pairing  $F_{0,1}$  with  $I$ . In this case, the gate complexity is independent of the sparsity  $L$  but quadratically dependent on  $t$ , similar to the algorithm in Ref. [34]. Conversely, when using a  $K$ th-order Trotter formula, the PTSC algorithm becomes  $L$  dependent with an almost linear dependence on  $t$ . In both cases, the PTSC algorithms achieve high simulation accuracy  $\varepsilon$ . We expect the 0th-order algorithm to be particularly useful for quantum chemistry Hamiltonians with large  $L$ , while higher-order algorithms are better suited for generic  $L$ -sparse Hamiltonians with long simulation times  $t$ .

### C. Nested-commutator compensation: overview

The PTSC algorithms above are generic and applicable to any Hamiltonian. When we consider the detailed structure of Hamiltonians, we could make the compensation algorithms more efficient by taking advantage of the commutation relationship of the terms in the Hamiltonians, which was formerly also studied in the Trotter algorithms [20].

We will take the first-order Trotter remainder  $V_1(x)$  as an illustrative example. Following the Taylor-series expansion in Eq. (5), the second-order term  $F_{1,2}(x)$  in  $V_1(x)$  can

be written as

$$F_{1,2}(x) = \sum_{\substack{r, \vec{r} \\ r + \sum \vec{r} = 2}} \frac{(-ixH)^r}{r!} \prod_{l=L}^1 \frac{(ixH_l)^{r_l}}{r_l!}. \quad (8)$$

Since  $F_{1,2}(x)$  is anti-Hermitian, all the Hermitian expansion terms in Eq. (8) will cancel out. We can then simplify  $F_{1,2}(x)$  as,

$$\begin{aligned} F_{1,2}(x) &= (ix)^2 \left( \sum_{l=1}^L \frac{H_l^2}{2!} + \sum_{l, l': l > l'} H_l H_{l'} \right) \\ &\quad + (-ixH) \sum_{l=1}^L (ixH_l) + \frac{(-ixH)^2}{2!} \\ &= \frac{x^2}{2} \sum_{l, l': l > l'} [H_{l'}, H_l], \end{aligned} \quad (9)$$

which is a summation of  $L(L-1)$  commutators. Since the commutators of Hermitian operators are always anti-Hermitian, this implies that the nested-commutator expansion of  $F_{1,2}(x)$  in Eq. (9) is compact enough since there is no Hermitian expansion terms in it.

For a common physical Hamiltonian with locality constraints, we can take advantage of the commutator-form expression like Eq. (9). For example, for an  $n$ -qubit lattice Hamiltonian with the form,

$$H = \sum_{j=0}^{n-1} H_{j,j+1}, \quad (10)$$

where the summand  $H_{j,j+1}$  acts on the  $j$ th and  $(j+1)$ th vertices. We can split the Hamiltonian to two components  $H = A + B$  where  $A := \sum_{j:\text{even}} H_{j,j+1}$ ,  $B := \sum_{j:\text{odd}} H_{j,j+1}$ , so that the summands  $H_{j,j+1}$  commute with each other in each component. We denote the norm of each Hamiltonian summand as

$$\Lambda = \max_j \|H_{j,j+1}\|, \quad \Lambda_1 = \max_j \|H_{j,j+1}\|_1. \quad (11)$$

Now, suppose we estimate the 1-norm of  $F_{1,2}(x)$  of the lattice Hamiltonian based on Eq. (9), we can see that there are only  $n$  nonzero terms: for any given Hamiltonian component  $H_{j,j+1}$ , only  $H_{j-j,j}$  and  $H_{j+1,j+2}$  do not commute with it. Then, the norm of  $F_{1,2}(x)$  is bounded by

$$\|F_{1,2}(x)\|_1 \leq n \Lambda_1 \frac{x^2}{2} = \mathcal{O}(nx^2). \quad (12)$$

Comparing with the original bound in Eq. (62),  $\|F_{1,2}(x)\|_1 \leq \eta_2 = \mathcal{O}((\lambda x)^2) = \mathcal{O}(n^2 x^2)$ , we improve the

system-size-related factor  $n$ . The improved system-size  $n$  dependence of the 1-norm  $\|F_{1,2}(x)\|_1$  suggests a corresponding improvement in the system-size dependence of the gate complexity for the nested commutator algorithm.

Now, we are going to generalize the idea above. In Sec. V, we show how to expand the second- and third-order terms of  $V_1(x)$  as a summation of nested commutators,

$$F_{1,s}(x) = \frac{x^s}{s!} C_{s-1} = \frac{(-ix)^s}{s!} \left( \mathcal{A}_{com}^{(s-1)}(H_L, \dots, H_1; H) - \sum_{l=1}^L \mathcal{A}_{com}^{(s-1)}(H_L, \dots, H_{l+1}; H_l) \right), \quad s = 2, 3, \quad (13)$$

where  $\mathcal{A}_{com}^{(s)}(A_L, \dots, A_1; B)$  is defined to be

$$\sum_{m_1 + \dots + m_L = s} \binom{s}{m_1, \dots, m_L} \text{ad}_{A_L}^{m_L} \dots \text{ad}_{A_1}^{m_1} B. \quad (14)$$

We also use the adjoint notation  $\text{ad}_{A_L} \dots \text{ad}_{A_1} B := [A_L, \dots [A_1, B] \dots]$ . It is easy to check that the form of  $F_{1,s}(x)$  ( $s = 2$  or  $3$ ) in Eq. (13) is anti-Hermitian, which is consistent with the discussion in the paired Taylor-series algorithm in the previous section. We can also generalize the method to the case of  $K$ th-order Trotter remainder, that is, to express the expansion terms  $F_{K,s}$  of the  $K$ th-order Trotter remainder based on the nested commutators.

Based on the nested-commutator expansion, we propose the  $K$ th-order NCC algorithm. As is illustrated in Fig. 4, in the construction of NCC, we first utilize the nested-commutator forms of Trotter error terms from  $K + 1$  to  $2K + 1$  order, i.e., the leading-order terms; then we apply the order-pairing techniques similar to PTSC in Fig. 3(c) to further suppress the 1-norm of  $\mu_x$ . A key difference between NCC and PTSC algorithms is that in PTSC algorithms, we compensate the Trotter error  $V_K(x)$  up to arbitrary order; while in NCC algorithms, we only compensate  $V_K(x)$  for leading-order terms, which shrinks the error from  $\mathcal{O}(x^{K+1})$  to  $\mathcal{O}(x^{2K+2})$  in one slice with the sampling cost  $\mu = 1 + \mathcal{O}(\|C_K\|_1 / ((K + 1)!) x^{2K+2})$ . The gate complexity estimation is then converted to the calculation of the 1-norm of the commutator  $\|C_K\|_1$ . For instance, if we consider the  $n$ -qubit lattice Hamiltonian models in Eq. (10), then we can prove that  $\|C_K\|_1 = \mathcal{O}(n)$ . We can then provide the following performance guarantee for the NCC algorithms.

*Theorem 2.* (Informal, see Theorem 4 in Sec. V) In a  $K$ th-order NCC algorithm ( $K = 1$  or  $2k$ ) with  $n$ -qubit lattice Hamiltonians, the gate complexity in a single round is  $\mathcal{O}(n^{1 + \frac{2}{2K+1}} t^{1 + \frac{1}{2K+1}} \varepsilon^{-\frac{1}{2K+1}})$ .

Compared to the performance of  $K$ th-order Trotter algorithm  $\mathcal{O}((nt)^{1+1/K} \varepsilon^{-1/K})$  [13], we achieve  $t$ - and  $\varepsilon$ -dependence better than  $2K$ th-order Trotter using only  $K$ th-order Trotter formula with simple compensation gates

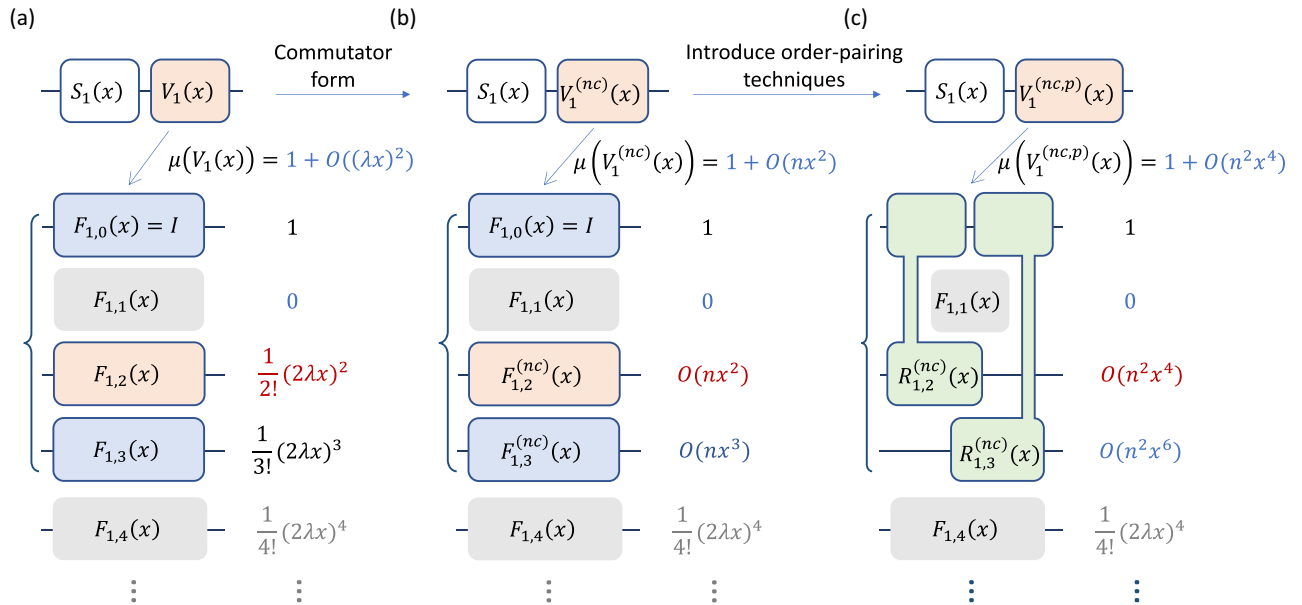


FIG. 4. Illustration of the idea of nested-commutator compensation (NCC) algorithm. We take the first-order algorithm  $K = 1$  as an example. (a) The Taylor-series expansion of the first-order Trotter remainder. We consider a truncation such that the higher-order terms with order  $s \geq 2K + 2$  will not be compensated. (b) We derive the nested-commutator form for the leading-order expansion terms  $F_{1,2}^{(nc)}(x)$  and  $F_{1,3}^{(nc)}(x)$ , which own better system-size dependence. (c) Since  $F_{1,2}^{(nc)}$  and  $F_{1,3}^{(nc)}$  are anti-Hermitian, we introduce Euler's formula in Eq. (7) to pair the leading-order terms to  $F_{1,0} := I$ . This will double the order of  $\mu_x - 1$  and further suppress the 1-norm  $\mu$  of the overall evolution  $U(t)$ .

of Pauli-rotation operators. To generalize the result in Theorem 2, we also study the performance of NCC algorithms when applied to a general Hamiltonian in Sec. VC.

#### D. Efficient random-sampling implementation

A simple implementation of the Trotter-LCU algorithm in Fig. 2(b) or Fig. 2(c) requires not only an easy-to-implement quantum circuit but also efficient classical random sampling of Pauli operators from the Trotter remainder  $V_K(x)$ . We now briefly discuss how to realize an efficient classical random sampling in PTSC and NCC algorithms, that is, with a space resource of  $\mathcal{O}(\kappa_K K)$  and time resource of  $\mathcal{O}(K(\log L + \log \kappa_K))$  where  $L$  is the sparsity of the Hamiltonian.

A key idea for achieving efficient sampling is to use a multistage hierarchical sampling algorithm. Rather than fully expanding the Trotter remainder into a direct summation of Pauli operators, we structure the LCU formula into multiple layers. This allows us to decompose the overall Pauli operator sampling process into a series of simpler, more manageable sampling tasks.

In the PTSC algorithm, the Trotter remainder in Eq. (5) is derived by expanding the time evolution  $e^{-ixH}$  and each Hamiltonian summand term  $e^{ixH_l}$  by Taylor series independently. As a result, the sampling can be done by first sampling the overall expansion order  $s$ , then sample the individual expansion order  $r$  of Hamiltonian  $H$  or the expansion order  $\vec{r} := [r_1, r_2, \dots, r_L]$  of the summands  $\{H_l\}$ . The sampling of  $r$  and  $\vec{r}$ , following the analysis in Sec. IV, can be done based on a multinomial distribution  $\text{Mul}(r, \vec{r}; \{\frac{1}{2}, \frac{\vec{p}}{2}\}; s)$  where  $\vec{p} := [p_1, p_2, \dots, p_L]$  denotes the normalized coefficient factor of  $H$  defined in Eq. (3). For the sampled Hamiltonian  $H$ , we further sample the summands  $H_l$  inside based on  $\vec{p}$ . We summarize the sampling algorithm in Fig. 5.

In the NCC algorithm, the Trotter remainder is expanded based on the adjoint operators. For example, for the lattice Hamiltonian  $H = A + B$  in Eq. (10), we can write the second- and third-order Trotter remainder of  $V_1(x)$  as,

$$\begin{aligned} F_{1,2}^{(nc)}(x) &= -\frac{x^2}{2!} \text{ad}_A B, \\ F_{1,3}^{(nc)}(x) &= i\frac{x^3}{3!} (2\text{ad}_B \text{ad}_A B + \text{ad}_A^2 B). \end{aligned} \quad (15)$$

We can first sample the expansion order  $s = 2$  or  $3$ . If  $s = 3$ , we then sample the specific commutator, i.e.,  $\text{ad}_B \text{ad}_A B$  or  $\text{ad}_A^2 B$ . For the given commutator, for example,  $\text{ad}_B \text{ad}_A B = [B, [A, B]]$ , we first randomly sample a summand  $H_{j,j+1}$  for the rightmost  $B$  as the starting point of the adjoint operator. The action of the subsequent  $\text{ad}_A$  and  $\text{ad}_B$  will enlarge the support of  $H_{j,j+1}$ , but within a ‘‘light-cone’’ region shown in Fig. 6. We then sample the Hamiltonian summand  $H_{j_1, j_1+1}$  and  $H_{j_2, j_2+1}$  for the adjoint operators  $\text{ad}_B$

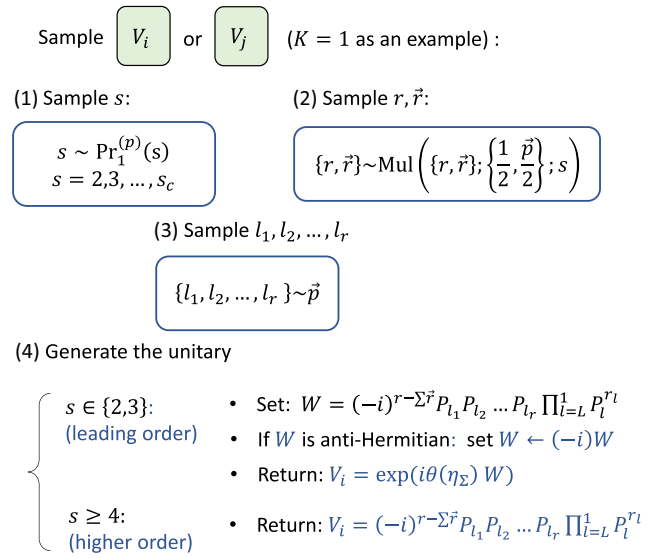


FIG. 5. The sampling procedure of  $V_i$  or  $V_j$  in Figs. 2(b) and 2(c) in the PTSC algorithm. We set  $K = 1$  as an example.  $\text{Pr}_1^{(p)}(s)$  is a discrete probability distribution given in Eq. (77).  $\eta_\Sigma := \eta_2 + \eta_3$  and  $\eta_s := (2\lambda x)^s / s!$ .  $\theta(y) := \tan^{-1}(y)$ .

and  $\text{ad}_A$ , but within the light-cone region. This will ensure our sampling to be efficient and with nested-commutator scaling.

A problem of sampling the Hamiltonian summands  $H_{j,j+1}$  in the commutator is that, since the Hamiltonian  $H$  may be nonhomogeneous, the 1-norm of  $H_{j,j+1}$  with different  $j$  may be different. This will complicate the sampling algorithm, since we need to first calculate the 1-norm of all the elementary commutators with the form of

$$\text{ad}_{H_{j_2, j_2+1}} \text{ad}_{H_{j_1, j_1+1}} H_{j, j+1}, \quad (16)$$

for all  $j, j_1$ , and  $j_2$ . When the Trotter order  $K$  and the expansion order  $s = K + 1, \dots, 2K + 1$  gets larger, the number of elementary adjoint operators will increase exponentially. Consequently, we cannot estimate the 1-norm of all the elementary commutator.

To solve this problem but still keep the advantage of the NCC algorithm, we introduce the following Hamiltonian ‘‘padding’’ technique to ensure all the elementary nested commutator own the same 1-norm. Consider a Hamiltonian summand  $H_{j,j+1}$ , which can be expanded to some Pauli operators,

$$H_{j,j+1} = \sum_{\omega} \alpha_j^{(\omega)} P_{j,j+1}^{(\omega)}, \quad (17)$$

where  $\alpha_j^{(\omega)}$  is a positive number and  $P_{j,j+1}^{(\omega)}$  is a normalized Pauli operator whose support is on qubit  $j$  and  $j + 1$ . Recall that  $\Lambda_j := \max_j \|H_{j,j+1}\|_1$  and  $\|H_{j,j+1}\|_1 :=$

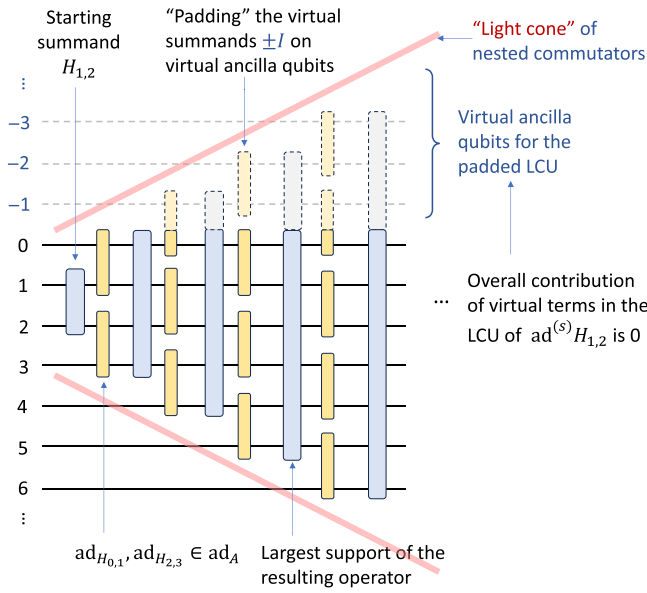


FIG. 6. Illustration of the generation of elementary commutators in a given commutator form  $\text{ad}_B \text{ad}_A \text{ad}_B \text{ad}_A H_{j,j+1}$ . Here we set  $j = 1$ . Start from  $H_{1,2}$ , we sequentially apply the adjoint operation of elementary summands  $H_{j_l, j_{l+1}}$  (the yellow blocks) on the operator. The blue blocks indicates the largest possible support of the resulting operators. When the support reach the boundary of the Hamiltonian, we introduce extra virtual ancillary qubits to construct padded LCU. The operators on the virtual qubits are always  $\pm I$ , so that we only need to implement a phase gate on the control qubit without the need to perform operations on the virtual qubits.

$\sum_{\omega} \alpha_j^{(\omega)}$ . When  $\|H_{j,j+1}\|_1$  is smaller than  $\Lambda_1$ , we add extra trivial terms  $\pm I$  in the Pauli decomposition of  $H_{j,j+1}$ ,

$$\begin{aligned} \bar{H}_{j,j+1} &= \sum_{\omega} \alpha_j^{(\omega)} P_{j,j+1}^{(\omega)} + \frac{\delta \Lambda_1}{2} I + \frac{\delta \Lambda_1}{2} (-I) \\ &= \Lambda_1 \sum_{\omega} \bar{p}_j^{(\omega)} P_{j,j+1}^{(\omega)}, \end{aligned} \quad (18)$$

where  $\delta \Lambda_1 := \Lambda_1 - \|H_{j,j+1}\|_1$ . Equation (18) holds naturally, but now with a manually predetermined 1-norm value  $\Lambda_1$ . Similarly, we can pad all the elementary commutators with the form

$$\text{ad}_{H_{j_1, j_1+1}} H_{j,j+1} = H_{j_1, j_1+1} H_{j,j+1} - H_{j,j+1} H_{j_1, j_1+1}, \quad (19)$$

so that their 1-norms are all  $2\Lambda_1^2$ . In this way, we ignore the commutator relationship between  $H_{j,j+1}$  and  $H_{j_1, j_1+1}$  as long as they are in the light-cone region.

After the padding procedure described above, all elementary nested commutators of the same order will have the same 1-norm. This property allows us to uniformly sample these commutators: the starting summand  $H_{j,j+1}$  is sampled uniformly from those in  $B$ , and the subsequent

$H_{j_1, j_1+1}$  and  $H_{j_2, j_2+1}$  are sampled uniformly within the light cone region. However, when the starting summand  $H_{j,j+1}$  is near the boundary, applying a few adjoint operators may cause it to touch the boundary. This reduces the number of possible elementary nested commutators compared to those starting from the center, resulting in different 1-norms for  $\text{ad}_A H_{j,j+1}$  depending on  $j$ . This complicates the sampling of the starting summand  $H_{j,j+1}$ . To resolve this issue and ensure uniform sampling of  $H_{j,j+1}$ , we introduce virtual qubits at the boundary, as illustrated in Fig. 6, and pad the virtual qubits with 0-summed  $\pm I$  terms. Since we only perform  $\pm I$  operation on the virtual qubits, we do not need to introduce it in the real experiments.

We remark that, our padding method preserve the locality structure. As a result, the performance guarantee in Theorem 4 still holds. We summarize the sampling algorithm in Fig. 7. If we consider the Heisenberg Hamiltonian

$$H = \sum_i \vec{\sigma}_i \vec{\sigma}_{i+1} + \sum_i Z_i, \quad (20)$$

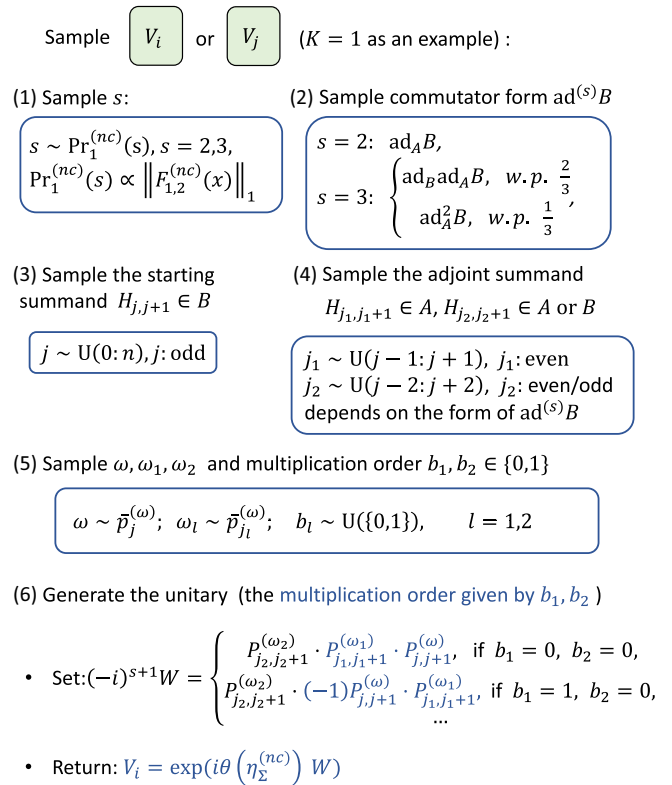


FIG. 7. The sampling procedure of  $V_i$  or  $V_j$  in Fig. 2(b) in the NCC algorithm. We set  $K = 1$  as example. The notation  $l : l + k$  indicates the number array of  $l, l + 1, \dots, l + k$ .  $b_l$  determines whether to multiply  $P_{j_l, j_l+1}^{(\omega_l)}$  to the left or the right side of the current Pauli operator.  $U(\bullet)$  refers to a uniform distribution in the set.  $\eta_{\Sigma}^{(nc)} := \|F_{1,2}^{(nc)}\|_1 + \|F_{1,3}^{(nc)}\|_1$ .  $\theta(y) := \tan^{-1}(y)$ . The probability  $\bar{p}_j^{(\omega)}$  is given by the Hamiltonian information in Eq. (18).

as an example, where  $\vec{\sigma}_i := (X_i, Y_i, Z_i)$  is the vector of Pauli operators on the  $i$ th qubit, we can define the summand  $H_{j,j+1}$  to be

$$H_{j,j+1} = 4 \left( \frac{1}{4} X_j X_{j+1} + \frac{1}{4} Y_j Y_{j+1} + \frac{1}{4} Z_j Z_{j+1} + \frac{1}{4} Z_j \right). \quad (21)$$

In this case,  $\Lambda_1 = \|H_{j,j+1}\| = 4$ . The probability distribution  $\bar{p}_j^{(\omega)}$  in Fig. 7 is to uniformly sample the  $XX$ ,  $YY$ ,  $ZZ$ , and  $ZI$  term. As a demonstration, we explicitly present the algorithm to sample the Pauli-rotation operator in the first-order NCC algorithm for the Heisenberg Hamiltonian in Eq. (20) in Algorithm 1.

As a final remark, the sampling procedure in both PTSC and NCC algorithms are independent of the implementation of the quantum circuit and the measurement outcome. Thanks to this property, we can perform the classical sampling during the quantum circuit implementation or even generate the sampled Pauli matrices before the implementation of the quantum circuits.

### E. Performance comparison

In Table I, we compare the implementation complexity and the gate complexity in a single round of experiment of the 0th-order PTSC,  $K$ th-order PTSC, and  $K$ th-order NCC algorithms to previous Hamiltonian simulation algorithms. For a fair comparison, we set the 1-norm of all LCU formulas  $\mu$  to be constant. In this case, the sample complexity of PTSC and NCC algorithms incurs a  $\mu^4$  overhead compared to standard sampling from the  $K$ th-order Trotter or post-Trotter algorithms.

We show that by inserting a few randomly sampled Pauli-rotation gates after each Trotter segment, as illustrated in Fig. 1, both PTSC and NCC achieve improved accuracy and time dependence. The gate counts for PTSC exhibit logarithmic dependence on accuracy,  $\log(1/\varepsilon)$ , while the NCC gate counts show improved system-size dependence.

To demonstrate how the Trotter-LCU algorithms can help reduce gate costs in practical scenarios, we estimate the single-shot gate count of random-sampling Trotter-LCU algorithms and compare it with the state-of-the-art Trotter algorithm, i.e., fourth-order Trotter [9,12,20]. To ensure a fair comparison, we set 1-norm of the LCU formula to be  $\mu = 2$ . Based on Proposition 1, this implies that the random-sampling Trotter-LCU algorithm will require an additional factor of 16 in the sample number to estimate the properties of the target state  $U\rho U^\dagger$  where  $U := e^{iHt}$  to a given precision compared to the normal Trotter or coherent implementation of LCU algorithms.

We compile their quantum circuits to CNOT gates, single-qubit Clifford gates, and single-qubit  $Z$ -axis rotation gates  $R_z(\theta) = e^{i\theta Z}$ . Here, we mainly compare the

ALGORITHM 1. Demonstration: sampling of  $V_i$  or  $V_j$  of first-order NCC algorithm for the Heisenberg Hamiltonian in Eq. (20)

---

**Input:** An  $n$ -qubit Heisenberg Hamiltonian  $H$  in Eq. (20); unit evolution time  $0 < x < 1$  for each Trotter segment;

**Output:** Sampling of a Pauli-rotation operator  $V_i$  from the Trotter remainder  $\tilde{V}_1^{(nc)}(x)$ .

- 1: Sample the expansion order  $s \in \{2, 3\}$  with the probability  $\{\frac{1}{1+24x}, \frac{24x}{1+24x}\}$ .
- 2: **if**  $s = 3$  **then**
- 3:   Sample the adjoint form  $\text{ad}^{(2)}B$  from  $\{\text{ad}_B \text{ad}_A B, \text{ad}_A^2 B\}$  with the probability  $\{1/3, 2/3\}$ .
- 4:   **end if**
- 5:   Sample the starting index  $j$  uniformly from all odd indices in  $0, \dots, n-1$ . Sample the Pauli operator  $P_{j,j+1}$  uniformly from  $\{XX, YY, ZZ, ZI\}$  on qubit  $j$  and  $j+1$ .
- 6:   Set  $W := P_{j,j+1}$ .
- 7:   Sample the adjoint index  $j_1$  uniformly from  $\{j-1, j+1\}$ . Sample the Pauli operator  $P_{j_1,j_1+1}$  uniformly from  $\{XX, YY, ZZ, ZI\}$  on qubit  $j_1$  and  $j_1+1$ . Sample the multiplication order  $b_1$  uniformly from  $\{0, 1\}$ .<sup>a</sup>
- 8:   **if**  $b_1 = 0$  **then**
- 9:     Set  $W := P_{j_1,j_1+1}W$ .
- 10:   **else**  $\triangleright b_1 = 1$
- 11:     Set  $W := -WP_{j_1,j_1+1}$ .
- 12:   **end if**
- 13:   **if**  $s = 3$  **then**
- 14:     **if**  $\text{ad}^{(2)}B = \text{ad}_B \text{ad}_A B$  **then**
- 15:       Sample the adjoint index  $j_2$  uniformly from  $\{j-2, j, j+2\}$ . Sample the Pauli operator  $P_{j_2,j_2+1}$  uniformly from  $\{XX, YY, ZZ, ZI\}$  on qubit  $j_2$  and  $j_2+1$ . Sample the multiplication order  $b_2$  uniformly from  $\{0, 1\}$ .
- 16:       **else**  $\triangleright \text{ad}^{(2)}B = \text{ad}_A^2 B$
- 17:       Sample the adjoint index  $j_2$  uniformly from  $\{j-3, j-1, j+1\}$ . Sample the Pauli operator  $P_{j_2,j_2+1}$  uniformly from  $\{XX, YY, ZZ, ZI\}$  on qubit  $j_2$  and  $j_2+1$ . Sample the multiplication order  $b_2$  uniformly from  $\{0, 1\}$ .
- 18:       **end if**
- 19:       **if**  $b_2 = 0$  **then**
- 20:         Set  $W := P_{j_2,j_2+1}W$ .
- 21:       **else**  $\triangleright b_2 = 1$
- 22:         Set  $W := -WP_{j_2,j_2+1}$ .
- 23:       **end if**
- 24:   **end if**
- 25: Return  $V_i := \exp(i\theta W)$  as the sampled Pauli-rotation operator. Here  $\theta := \tan^{-1}(16nx^2(1+24x))$ .

---

number of  $R_z(\theta)$  gates since they are the most resource-consuming part on a fault-tolerant quantum computer [40]. The CNOT gate-number comparison results can be found in Appendix G, which is similar to the  $R_z(\theta)$  gate comparison. We also compare the gate counts of the Trotter-LCU algorithms to the coherent-implementation of LCU [24,25] and QSP [27,28] in Appendix G.

In the first comparison, we consider the simulation of generic Hamiltonians without the usage of commutator information, for which we choose the two-local Hamiltonian,  $H = \sum_{i,j} X_i X_j + \sum_i Z_i$  where  $X_i$  and  $Z_i$  are the Pauli matrices on the  $i$ th qubit. Figures 8(a) and 8(b) show the gate counts for the fourth-order Trotter formula and the

TABLE I. Comparison of the implementation hardness and gate complexity in a single round of the circuit for Trotter-LCU methods versus previous algorithms. Here,  $\tilde{\log}(x) := \log(x)/\log \log(x)$ . The (implementation) hardness refers to whether one needs to implement multicontrolled gates with plenty of ancillary qubits. In the comparison for the system-size scaling of lattice Hamiltonians, we use the fact that  $\lambda = \mathcal{O}(n)$  and  $L = \mathcal{O}(n)$ .

Algorithm	Implementation hardness	Accuracy	Size scaling (lattice Hamiltonian)	Time dependence
$K$ th-order Trotter [8]	Easy	$\mathcal{O}(\varepsilon^{-1/K})$	$\mathcal{O}(n^{1+\frac{1}{K}})$	$\mathcal{O}(t^{1+\frac{1}{K}})$
Post-Trotter [25,27]	Hard	$\mathcal{O}(\tilde{\log}(1/\varepsilon))$	$\mathcal{O}(n^2)$	$\mathcal{O}(t)$
0th-order PTSC	Easy	$\mathcal{O}(\tilde{\log}(1/\varepsilon))$	$\mathcal{O}(n^2)$	$\mathcal{O}(t^2)$
$K$ th-order PTSC	Easy	$\mathcal{O}(\tilde{\log}(1/\varepsilon))$	$\mathcal{O}(n^{2+\frac{1}{2K+1}})$	$\mathcal{O}(t^{1+\frac{1}{2K+1}})$
$K$ th-order NCC	Easy	$\mathcal{O}(\varepsilon^{-1/(2K+1)})$	$\mathcal{O}(n^{1+\frac{2}{2K+1}})$	$\mathcal{O}(t^{1+\frac{1}{2K+1}})$

PTSC Trotter-LCU algorithms with different orders with an increasing time  $t$  and increasing system size  $n$ , respectively. The gate counting method for fourth-order Trotter with random permutation is based on the analytical bounds in Ref. [12].

From Figs. 8(a) and 8(b), we can clearly see the advantage of composition of Trotter and LCU formulas: if we do not use LCU and merely apply Trotter formulas, the gate resource of fourth-order Trotter suffers from a large overhead that is 2 orders of magnitudes larger than the PTSC algorithms. Moreover, if we increase  $\varepsilon$  from  $10^{-3}$  to  $10^{-5}$ , we can see a clear increase of the gate resources for the Trotter algorithm. For the PTSC algorithms, however, the gate number is almost not affected by  $\varepsilon$  since they enjoy a logarithmic  $\varepsilon$ -dependence.

On the other hand, if we do not use Trotter formula, the 0th-order PTSC algorithm owns a quadratically worse  $t$ -dependence ( $\mathcal{O}(t^2)$ ) than the fourth-order Trotter ( $\mathcal{O}(t^{1.25})$ ), second-order PTSC ( $\mathcal{O}(t^{1.2})$ ), and fourth-order PTSC ( $\mathcal{O}(t^{1.11})$ ) algorithms in Fig. 8(a). For a short-time evolution  $t = n$ , the system-size dependence of second- or fourth-order PTSC in Fig. 8(b) outperforms 0th-order

PTSC algorithm. In the case when long-time Hamiltonian simulation is required, for example,  $t$  should be set to  $10^3$  for the phase estimation [11], the advantage of second- or fourth-order PTSC to 0th-order PTSC will be more obvious. The composition of Trotter and LCU formulas enables second- or fourth-order PTSC to enjoy good  $t$  and  $\varepsilon$  dependence and small gate-resource overhead simultaneously.

Next, we compare the gate count when simulating the lattice models, where the commutator analysis will help remarkably reduce the gate count. We consider the Heisenberg Hamiltonian  $H = \sum_i \vec{\sigma}_i \vec{\sigma}_{i+1} + \sum_i Z_i$  using the nested commutator bounds. In Fig. 8(c), we choose  $n = t = 12$  and 50 and show the gate number with respect to the accuracy requirement  $\varepsilon$ . The fourth-order Trotter error analysis is based on the nested commutator bound (Proposition M.1 in Ref. [20]), which is currently the tightest Trotter error analysis. The performance of our second-order NCC algorithm is based on the analytical bound in Sec. H in the Appendix. We mainly present results for the second-order NCC algorithm due to its simplicity and leave precise higher-order NCC gate count analysis for future study.

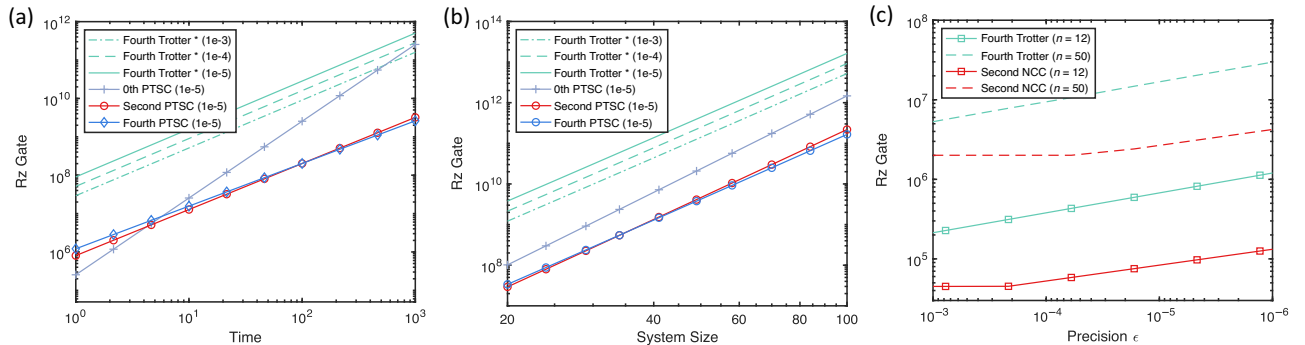


FIG. 8. Non-Clifford  $R_z(\theta)$  gate-number estimation for simulating real-time dynamics with Trotter and Trotter-LCU algorithms. (a) and (b) show the  $R_z(\theta)$  gate counts when simulating the Hamiltonian  $H = \sum_{i,j} X_i X_j + \sum_i Z_i$ . (a) shows the  $R_z(\theta)$  gate count with an increasing time and fixed system-size  $n = 20$ . (b) shows the  $R_z(\theta)$  gate count with an increasing system size with the time  $t = n$ . We list the performance of 0th, second, and fourth-order PTSC algorithms with  $\mu = 2$ . The fourth-order Trotter analytical bound is from Ref. [12]. (c) The  $R_z(\theta)$  gate count for the nearest-interaction Heisenberg model using the nested commutator bound with  $n = 12$  and  $n = 50$ . The fourth-order Trotter commutator bound is from Ref. [20].

From Fig. 8(c), we can see that, while enjoying near-optimal system-size scaling similar to the fourth-order Trotter algorithm, which is currently the best one for lattice Hamiltonians [13,20], the second-order NCC algorithm shows better accuracy dependence than fourth-order Trotter algorithm. Particularly, using the same gate number as the fourth-order Trotter, we are able to achieve a 3 to 4 orders of magnitudes higher accuracy  $\varepsilon$ .

### III. PRELIMINARIES

In this section, we review the Hamiltonian simulation algorithms based on Trotter and LCU formulas.

In all the Hamiltonian simulation algorithms discussed in this work, we divide the real-time evolution  $U(t)$  into  $\nu$  segments,

$$U(t) = e^{-iHt} = (U(x))^\nu = (e^{-iHx})^\nu, \quad (22)$$

with  $x := t/\nu$ , and consider the construction of each small segment  $U(x)$ . Without loss of generality, we assume  $x > 0$ .

#### A. Trotter formulas

The most natural way to approximate  $U(x)$  is to apply the Lie-Trotter-Suzuki formulas [8,9]. Hereafter, we refer to them as Trotter formulas. The first-order Trotter formula is

$$S_1(x) = \prod_{l=1}^L e^{-ixH_l} = e^{-ixH_L} \dots e^{-ixH_2} e^{-ixH_1} =: \overleftarrow{\prod} e^{-ix\vec{H}}. \quad (23)$$

Here,  $\vec{H} := [H_1, H_2, \dots, H_L]$ . In Eq. (23), we simplify the notation of the sequential products from  $l = 1$  to  $L$  with the same form, using the arrow to denote the ascending direction of the dummy index  $l$ . The Hermitian conjugation of  $S_1(x)$  can similarly be written as  $S_1(x)^\dagger = \overrightarrow{\prod} e^{ix\vec{H}}$ .

The second-order Trotter formula is

$$\begin{aligned} S_2(x) &= D_2(-x)^\dagger D_2(x) = S_1(-\frac{x}{2})^\dagger S_1(\frac{x}{2}) \\ &= \overrightarrow{\prod} e^{-i\frac{1}{2}x\vec{H}} \overleftarrow{\prod} e^{-i\frac{1}{2}x\vec{H}}, \end{aligned} \quad (24)$$

where  $D_2(x) := S_1(x/2)$ . We have  $S_2(-x)^\dagger = S_2(x)$ .

The general  $2k$ th-order Trotter formula is [9]

$$\begin{aligned} S_{2k}(x) &= D_{2k}(-x)^\dagger D_{2k}(x) \\ &= [S_{2k-2}(u_k x)]^2 S_{2k-2}((1-4u_k)x) [S_{2k-2}(u_k x)]^2, \end{aligned} \quad (25)$$

with  $u_k := 1/(4-4^{1/(2k-1)})$  for  $k \geq 1$ . The operator  $D_{2k}(x)$  is defined recursively,

$$D_{2k}(x) := D_{2k-2}((1-4u_k)x) S_{2k-2}(u_k x)^2. \quad (26)$$

By induction from Eqs. (25) and (26) we can show that  $S_{2k}(-x)^\dagger = S_{2k}(x)$ .

We also denote the zeroth-order Trotter formula to be  $S_0(x) = I$  for consistency. We denote the multiplicative remainder of the Trotter formulas as

$$V_K(x) = U(x) S_K(x)^\dagger, \quad (27)$$

for  $K = 0, 1, 2k$ . In what follows, we name this Trotter error term  $V_K(x)$  as the  $K$ th-order Trotter remainder to avoid ambiguity to other error effects such as truncation error.

In Ref. [8], Suzuki proves the following order condition for the Trotter formulas:

$$S_K(x) = U(x) + \mathcal{O}(x^{K+1}) = e^{-iHx + \mathcal{O}(x^{K+1})}, \quad (28)$$

for  $K = 1$  or even positive  $K$ . As a result, the remainder  $V_K(x)$  will only contain the terms of  $x^q$  with  $q \geq K + 1$ . We will use the following order condition of Trotter formulas in the later discussion.

*Lemma 1 (Order condition of Trotter formulas, Theorem 4 in Ref. [20]).* Let  $H$  be an Hermitian operator and  $U(x) := e^{-iHx}$  to be the time-evolution operator with  $x \in \mathbb{R}$ . For the Trotter formula  $S_K(x)$  defined in Eqs. (23) and (25), where  $K = 1$  or positive even number, we have the following:

- (1) Additive error:  $A_K(x) := S_K(x) - U(x) = \mathcal{O}(x^{K+1})$ .
- (2) Multiplicative error:  $M_K(x) := U(x) S_K(x)^\dagger - I = \mathcal{O}(x^{K+1})$ .
- (3) For the exponentiated error  $E_K(x)$  such that  $S_K(x) = \mathcal{T} \exp(\int_0^x d\tau (-iH + E_K(\tau)))$ , we have  $E_K(x) = \mathcal{O}(x^K)$ .

#### B. LCU formulas

Instead of decomposing  $U(x)$  as a product of elementary unitaries, another way is to decompose  $U(x)$  to a summation of elementary unitaries. We now provide a formal definition of a LCU formula.

*Definition 1 ([30]).* A  $(\mu, \varepsilon)$  LCU formula of an operator  $V$  is defined to be

$$\tilde{V} = \sum_{i=0}^{\Gamma-1} c_i V_i = \mu \sum_{i=0}^{\Gamma-1} \text{Pr}(i) V_i, \quad (29)$$

such that the spectral norm distance  $\|V - \tilde{V}\| \leq \varepsilon$ . Here,  $\mu > 0$  is the  $l_1$ -norm of the coefficient vector,  $\text{Pr}(i)$  is

a probability distribution over different unitaries  $V_i$ , and  $\{V_i\}_{i=0}^{\Gamma-1}$  is a set of unitaries. Here, we assume  $c_i > 0$  for all  $i$  and absorb the phase into the unitaries  $V_i$ . We call  $\mu$  the 1-norm of this  $(\mu, \varepsilon)$ -LCU formula.

In what follows, we define the 1-norm  $\|V\|_1$  of an operator  $V$  to be its smallest 1-norm over all possible  $(\mu, 0)$ -LCU formulas for  $V$ . Note that,  $\|U\|_1 = 1$  for any unitary  $U$ . One can easily check the validity of this norm definition.

We may consider two ways to implement the LCU formula of the operator  $V$ . In the first way, we coherently implement the LCU formula by introducing an ancillary system  $A$  with the dimension  $\Gamma$ , which costs  $\lceil \log_2 \Gamma \rceil$  qubits. For the LCU lemma defined in Eq. (29), we define the amplitude-encoding unitary  $U_{AE}$  and the select gate  $\text{Sel}(\tilde{V})$  to be,

$$\begin{aligned} U_{AE} |0\rangle_A &:= \sum_{i=0}^{\Gamma-1} \sqrt{\text{Pr}(i)} |i\rangle, \\ \text{Sel}(\tilde{V}) &:= \sum_{i=0}^{\Gamma-1} |i\rangle \langle i| \otimes V_i. \end{aligned} \quad (30)$$

Then, the following controlled-gate  $W_{AB}$  acting on ancillary  $A$  and system  $B$  defines a way to realize LCU coherently when we prepare  $|0\rangle$  on  $A$  and measure it on computational basis to get  $|0\rangle$ ,

$$W := (U_{AE}^\dagger \otimes I) \text{Sel}(\tilde{V}) (U_{AE} \otimes I). \quad (31)$$

More precisely, if we set the initial state on  $B$  to be  $|\psi\rangle$ , then we have

$$W |0\rangle_A |\psi\rangle_B = \frac{1}{\mu} |0\rangle_A (\tilde{V} |\psi\rangle)_B + \sqrt{1 - \frac{1}{\mu^2}} |\Phi_\perp\rangle_{AB}, \quad (32)$$

where  $|\Phi_\perp\rangle_{AB}$  is a state whose ancillary state is supported in the subspace orthogonal to  $|0\rangle$ .

If we perform computation-basis measurement directly on  $A$ , the successful probability to obtain  $\tilde{V} |\psi\rangle$  is  $1/\mu^2$ . To boost the successful probability to nearly deterministic, we can introduce the amplitude amplification techniques [24]. To this end, we consider the following isometry:

$$\mathcal{V} := -WRW^\dagger RW (|0\rangle \otimes I)_{AB}. \quad (33)$$

Here,  $R$  is a reflection over  $|0\rangle$  on the system  $A$ ,

$$R = I - 2 |0\rangle_A \langle 0|. \quad (34)$$

Consider the case of  $\mu = 2$  and  $\varepsilon$  is small for the LCU formula. If  $\tilde{V}$  is a unitary, then we have  $(|0\rangle \otimes I)\mathcal{V} = \tilde{V}$ .

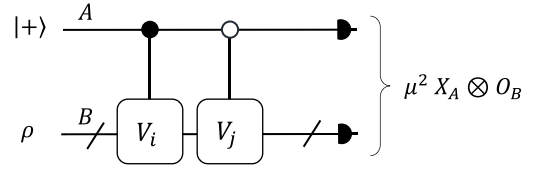


FIG. 9. Estimate the observable value  $\text{Tr}(V\rho V^\dagger O)$  based on the random sampling implementation of  $(\mu, \varepsilon)$ -LCU formula of the  $V$  defined in Eq. (29).

For a general  $\tilde{V}$ , we can verify the resulting operator

$$(|0\rangle \otimes I)\mathcal{V} = \frac{3}{2}\tilde{V} - \frac{1}{2}\tilde{V}\tilde{V}^\dagger\tilde{V}, \quad (35)$$

is close to  $V$  with the spectral norm distance bound  $\xi := (\varepsilon^2 + 3\varepsilon + 4)\varepsilon/2$ . Also, the successful probability to project the ancilla to  $|0\rangle$  is larger than  $(1 - \xi)^2$ .

In the second way to implement the LCU formula, we randomly sample the terms  $\{V_i\}$  in Eq. (29) based on the probability  $\text{Pr}(i)$  [30]. This way saves the ancillary qubit number and the cost to implement the multiqubit Toffoli gates in  $U_{AE}$  and  $\text{Sel}(\tilde{V})$  in Eq. (30), which is more suitable to implement in the near-term devices. Instead of implementing the operator  $V$  directly, we now focus on the task to estimate the properties of the target state  $\sigma = V\rho V^\dagger$  where  $\rho$  is the initial state of the Hamiltonian simulation task. Suppose we want to estimate the expectation value of a given observable  $O$  on  $\sigma$ , we can then embed the task to the Hadamard test [39] shown in Fig. 9.

We first prepare a  $|+\rangle$  state on a single ancillary qubit. After that, we implement two controlled operations  $|0\rangle_A \langle 0| \otimes I_B + |1\rangle_B \langle 1| \otimes (V_i)_B$  and  $|0\rangle_A \langle 0| \otimes (V_j)_B + |1\rangle_B \langle 1| \otimes I_B$  where  $V_i$  and  $V_j$  are sampled independently from the LCU formula in Eq. (29). The measured expectation value of  $\langle \mu^2 X_A \otimes O_B \rangle$  is then a nearly unbiased estimation of  $\text{Tr}(V\rho V^\dagger O)$ . We will use the following performance guarantee for the observable estimation.

*Proposition 1 (Performance of the random-sampling LCU implementation, Theorem 2 from Ref. [35]).* For a target operator  $V$  and its  $(\mu, \varepsilon)$ -LCU formula defined in Definition 1, if we estimate the value  $\langle O \rangle_V := \text{Tr}(V\rho V^\dagger O)$  with an initial state  $\rho$  and observable  $O$  using the circuit in Fig. 9 for  $N$  times, then the distance between the mean estimator value  $\hat{O}$  and the true value  $\langle O \rangle_V$  is bounded by

$$|\hat{O} - \langle O \rangle_V| \leq \|O\|(3\varepsilon + \varepsilon_n), \quad (36)$$

with successful probability  $1 - \delta$ . Here,  $N = 2\mu^4 \ln(2/\delta) / \varepsilon_n^2$ ,  $\|O\|$  is the spectral norm of  $O$ .

From Proposition 1 we can see that, the 1-norm  $\mu$  of the LCU formula affects the sample complexity while the

accuracy factor  $\varepsilon$  introduces extra bias in the observable estimation. To estimate  $\langle O \rangle_V$  using Hadamard-test-type circuits with  $\varepsilon$  accuracy, we need  $\mathcal{O}(\mu^4/\varepsilon^2)$  sampling resource, which owns an extra  $\mu^4$  overhead compared to the normal Hamiltonian simulation algorithms [35]. To make the algorithm efficient, we need to set  $\mu$  to be a constant.

In the later discussion, we will construct new LCU formulas from the product of LCU formulas. We will use following proposition.

*Proposition 2 (Product of LCU formulas).* Suppose we have a  $(\mu, \varepsilon)$ -LCU formula  $\tilde{V}$  for an operator  $V$  with the form of Eq. (29). Then the product formula

$$\tilde{V}^\nu = \mu^\nu \sum_{i_1, i_2, \dots, i_\nu} \Pr(i_1) \Pr(i_2) \cdots \Pr(i_\nu) V_1 V_2 \cdots V_\nu, \quad \nu \in \mathbb{N}, \quad (37)$$

is a  $(\mu', \varepsilon')$ -LCU formula for  $V^\nu$  with

$$\mu' = \mu^\nu, \quad \varepsilon' \leq \nu \mu' \varepsilon. \quad (38)$$

*Proof.* The 1-norm is obvious. We now bound the distance  $\varepsilon'$  between  $\tilde{U}^\nu$  and  $U^\nu$ .

$$\begin{aligned} \|U^\nu - \tilde{U}^\nu\| &= \left\| \sum_{k=1}^{\nu} U^{k-1} (U - \tilde{U}) \tilde{U}^{\nu-k} \right\| \\ &\leq \sum_{k=1}^{\nu} \|U^{k-1}\| \|U - \tilde{U}\| \|\tilde{U}^{\nu-k}\| \\ &\leq \nu \|U - \tilde{U}\| \sum_{k=1}^{\nu} \max\{\|U\|, \|\tilde{U}\|\}^{\nu-1} \\ &\leq \nu \varepsilon \mu^{\nu-1} \leq \nu \mu' \varepsilon. \end{aligned} \quad (39)$$

■

We remark that, when there are common unitary components for all  $\{V_i\}$  in the LCU formula Eq. (29), we can simplify the circuit in Fig. 9 by removing the ancillary control on the common unitary components. Suppose we have the following LCU formula for an operator  $V$ ,

$$\tilde{V} = \mu \sum_{i_1, i_2, \dots, i_\nu} \Pr(i_1, i_2, \dots, i_\nu) V_{i_\nu} W_\nu \cdots V_{i_2} W_2 V_{i_1} W_1, \quad (40)$$

such that  $\|V - \tilde{V}\| \leq \varepsilon$ , where  $W_1, W_2, \dots, W_\nu$  are some fixed unitaries. Then according to Definition 1, Eq. (40) is a  $(\mu, \varepsilon)$ -LCU formula of  $V$  with the elementary unitaries to be  $V_{\vec{i}} = V_{i_\nu} W_\nu \cdots V_{i_2} W_2 V_{i_1} W_1$  for  $\vec{i} = \{i_1, i_2, \dots, i_\nu\}$ . Instead of naively apply the Hadamard test circuit in

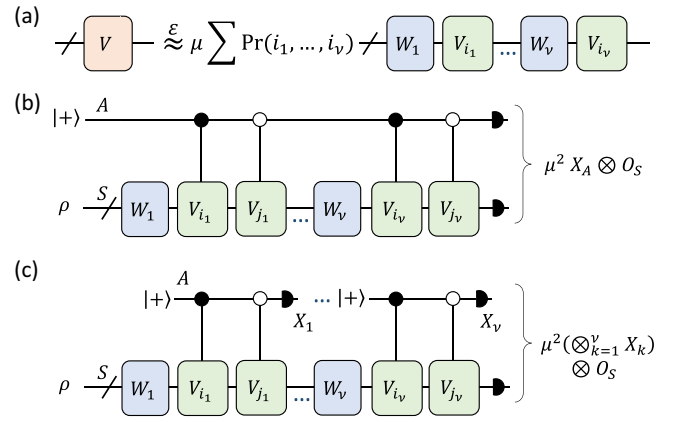


FIG. 10. Improve the random-sampling implementation of Hamiltonian simulation based on  $(\mu, \varepsilon)$ -LCU formula in Eq. (40). (a) The LCU formula of  $V$  in Eq. (40) with deterministic components. (b) The improved application of Hadamard test. We remove the control on the fixed unitaries and apply them only once. (c) A variant of Hadamard test, where the ancillary qubit is measured and reset for each randomly sampled unitary  $V_{i_k}$  for  $k = 1, \dots, \nu$ .

Fig. 9, we can introduce an equivalent circuit implementation shown in Fig. 10(b). The performance guarantee in Proposition 1 also holds for the improved implementation.

Furthermore, we notice that  $\text{Tr}(O \tilde{V} \rho \tilde{V}^\dagger)$  can be written as,

$$\mathbb{E}_{i_1, j_1, \dots, i_\nu, j_\nu} \text{Tr}[O \mathcal{E}_{i_\nu, j_\nu} \circ \mathcal{W}_\nu \circ \dots \circ \mathcal{E}_{i_1, j_1} \circ \mathcal{W}_1(\rho)], \quad (41)$$

where  $\mathcal{W}_k(\bullet) := W_k \bullet W_k^\dagger$  and  $\mathcal{E}_{i_k, j_k}(\bullet) := \frac{1}{2}(V_{i_k} \bullet V_{j_k}^\dagger + V_{j_k} \bullet V_{i_k}^\dagger)$  for  $k = 1, 2, \dots, \nu$ . As a result, we can implement each channel  $\mathcal{W}_k$  by a unitary and each map  $\mathcal{E}_{i_k, j_k}$  by a Hadamard-test-type circuit. This leads to a variant circuit shown in Fig. 10(c), where the ancillary qubit is measured and reset for every segment. While this circuit owns the same gate complexity as Fig. 10(b), it is beneficial in a fault-tolerant quantum computer since we do not need to store the ancillary qubit for a long time—the ancillary qubit is activated only in the compensation stage and is quickly measured.

#### IV. TROTTER-LCU ALGORITHMS WITH PAIRED TAYLOR-SERIES COMPENSATION

In this section, we construct a  $(\mu, \varepsilon)$ -LCU formula for the remainder  $V_K(x)$  of the  $K$ -th order Trotter formula based on the idea to perform Taylor-series (TS) expansion on all the exponential terms in  $V_K(x)$ . Although TS expansion is naturally a LCU formula of  $V_K(x)$ , it usually owns poor  $\mu$  performance. To further suppress the 1-norm of the expansion, we will modify the introduce a “pairing” idea to combine the terms that correspond to different TS expansion orders. We first consider a simple case without

Trotter formula in Sec. IV A to illustrate the major idea to construct TS-based LCU formula. Then we take the construction of LCU for the first-order Trotter formula as an example in Sec. IV B. Finally, in Sec. IV C, we discuss the random-sampling implementation of the algorithm and analyze its sample and gate complexity.

### A. Zeroth-order case

We begin our discussion from the case where the Trotter formula is trivial, i.e.,  $S_0(x) = I$  for each segment. In this case, the Trotter remainder is  $V_0(x) = U(x) = e^{-ixH}$ . To construct LCU formula, we expand  $V_0(x)$  by Taylor series [25],

$$\begin{aligned} V_0(x) &= \sum_{s=0}^{\infty} \frac{(-i\lambda x)^s}{s!} \sum_{l_1, \dots, l_s} p_{l_1} p_{l_2} \cdots p_{l_s} P_{l_1} P_{l_2} \cdots P_{l_s} \\ &= \sum_{s=0}^{\infty} F_{0,s}(x), \end{aligned} \quad (42)$$

where

$$F_{0,s}(x) = \frac{(-i\lambda x)^s}{s!} \sum_{l_1, \dots, l_s} p_{l_1} p_{l_2} \cdots p_{l_s} P_{l_1} P_{l_2} \cdots P_{l_s}. \quad (43)$$

The 1-norm of  $V_0(x)$  is

$$\begin{aligned} \|V_0(x)\|_1 &= \sum_{s=0}^{\infty} \|F_{0,s}(x)\|_1 \\ &= 1 + (\lambda x) + \frac{1}{2!}(\lambda x)^2 + \cdots = e^{\lambda x}. \end{aligned} \quad (44)$$

That is, the 1-norm of  $V_0(x)$  is exponentially large with respect to  $\lambda x$ . Suppose we use  $V_0(x)$  directly for the random-sampling implementation of LCU following Fig. 2(b), the composite LCU formula for  $U(t) = V_0(x)^v$  is the product of  $V_0(x)$ . Based on Proposition 2, the 1-norm of the product formula is  $\mu = (e^{\lambda x})^v = e^{\lambda t}$ , which increases exponentially with the simulation time  $t$ . Based on Proposition 1, this implies an exponentially increasing sample cost  $N = \mathcal{O}(\mu^4)$ . To make the TS expansion practical for the random-sampling implementation, we need to reduce 1-norm of  $V_0(x)$ .

When  $\lambda x$  is a small value, the major contribution to  $\|V_0(x)\|_1$  comes from the low-order terms of  $F_{0,s}(x)$ . Note that the first-order term  $F_{0,1}(x)$  is anti-Hermitian. This allows us to utilize the following Euler's formula on Pauli operators: for a Pauli matrix  $P$  and  $y \in \mathbb{R}$ ,

$$I + iyP = \sqrt{1 + y^2} e^{i\theta(y)P}, \quad (45)$$

where  $\theta(y) := \tan^{-1}(y)$ .

To suppress  $\|V_0(x)\|_1$ , we rewrite  $V_0(x)$  as follows:

$$\begin{aligned} V_0(x) &= \sum_{s=0}^{\infty} F_{0,s}(x) \\ &= I - i\lambda x \sum_{l=1}^L p_l P_l + \sum_{s=2}^{\infty} F_{0,s}(x) \\ &= \sum_{l=1}^L p_l (I - i\lambda x P_l) + \sum_{s=2}^{\infty} F_{0,s}(x). \end{aligned} \quad (46)$$

Then, we can apply Eq. (45) on Eq. (46) to convert the first-order term to Pauli rotation unitaries

$$V_0^{(p)}(x) = \sqrt{1 + (\lambda x)^2} \sum_{l=1}^L p_l e^{i\theta_0 P_l} + \sum_{s=2}^{\infty} F_{0,s}(x), \quad (47)$$

where  $\theta_0 := \tan^{-1}(-\lambda x)$ . We call the formula in Eq. (47) the 0th-order PTS formula.

In practice, to avoid the sampling in the infinitely large space, we introduce a truncation  $s_c \geq 2$  to the expansion order of  $x$ . After this truncation, the approximated LCU formula of  $V_0^{(p)}(x)$  is

$$\begin{aligned} \tilde{V}_0^{(p)}(x) &:= \tilde{U}_0^{(p)}(x) = \sqrt{1 + (\lambda x)^2} \sum_{l=1}^L p_l e^{i\theta_0 P_l} + \sum_{s=2}^{s_c} F_{0,s}(x) \\ &= \mu_0^{(p)}(x) \left( \Pr_0^{(p)}(1) V_{0,1}^{(p)} + \sum_{s=2}^{s_c} \Pr_0^{(p)}(s) V_{0,s}^{(p)} \right), \end{aligned} \quad (48)$$

where

$$\begin{aligned} \mu_0^{(p)}(x) &:= \sqrt{1 + (\lambda x)^2} + \sum_{s=2}^{s_c} \|F_{0,s}(x)\|_1 \\ &\leq \sqrt{1 + (\lambda x)^2} + (e^{\lambda x} - 1 - \lambda x), \\ \Pr_0^{(p)}(s) &:= \frac{1}{\mu_0^{(p)}(x)} \begin{cases} \sqrt{1 + (\lambda x)^2}, & s = 1, \\ \|F_{0,s}(x)\|_1 = \frac{(\lambda x)^s}{s!}, & s = 2, 3, \dots, s_c, \end{cases} \\ V_{0,s}^{(p)} &:= \begin{cases} \sum_l p_l e^{i\theta_0 P_l}, & s = 1, \\ \sum_{l_1, s} p_{l_1, s}^{(s)} (-i)^s P_{l_1, s}^{(s)}, & s = 2, 3, \dots, s_c. \end{cases} \end{aligned} \quad (49)$$

After ‘‘pairing’’ the terms with  $s = 0$  and  $s = 1$ , we obtain Eq. (48), which is a new LCU formula with the 1-norm  $\mu_0^{(p)}(x)$ . We have the following proposition to characterize the LCU formula in Eq. (48).

*Proposition 3 (0th-order Trotter-LCU formula by paired Taylor-series compensation).* For  $x \geq 0$  and  $s_c \geq 2$ ,

$\tilde{V}_0^{(p)}(x)$  in Eq. (48) is a  $(\mu_0^{(p)}(x), \varepsilon_0^{(p)}(x))$ -LCU formula of  $V_0(x) = U(x)$  with

$$\mu_0^{(p)}(x) \leq e^{\frac{3}{2}(\lambda x)^2}, \quad \varepsilon_0^{(p)}(x) \leq \left( \frac{e\lambda x}{s_c + 1} \right)^{s_c+1}. \quad (50)$$

*Proof.* For the normalization factor, we have

$$\begin{aligned} \mu_0^{(p)}(x) &\leq \sqrt{1 + (\lambda x)^2} + (e^{\lambda x} - 1 - \lambda x) \\ &\leq 1 + \frac{1}{2}(\lambda x)^2 + (e^{\lambda x} - 1 - \lambda x) \\ &\leq e^{(\lambda x)^2} + \frac{1}{2}(\lambda x)^2 \leq e^{\frac{3}{2}(\lambda x)^2}. \end{aligned} \quad (51)$$

The fourth inequality is due to  $e^x - x \leq e^{x^2}$  for  $x \in \mathbb{R}$ . For the distance bound, we have

$$\begin{aligned} \|\tilde{V}_0(x) - V_0(x)\| &\leq \sum_{s>s_c} \|F_{0,s}(x)\| \\ &\leq \sum_{s>s_c} \frac{(\lambda x)^s}{s!} \|V_{0,s}(x)\| \\ &\leq \sum_{s>s_c} \frac{(\lambda x)^s}{s!} \leq \left( \frac{e\lambda x}{s_c + 1} \right)^{s_c+1}. \end{aligned} \quad (52)$$

In the second inequality, we use the fact that

$$\|V_{0,s}\| \leq \sum_{l_1, l_2, \dots, l_s} p_{l_1} p_{l_2} \cdots p_{l_s} \|(-i)^s P_{l_1} P_{l_2} \cdots P_{l_s}\| \leq 1. \quad (53)$$

In the fourth inequality of Eq. (52), we apply the following Poisson tail bound formula,

$$\sum_{s=k+1}^{\infty} \frac{x^s}{s!} \leq \left( \frac{ex}{k+1} \right)^{k+1}, \quad (54)$$

which can be proven from Theorem 1 in Ref. [41].  $\blacksquare$

From Proposition 3 we see that, the 1-norm of the LCU formula  $V_0^{(p)}(x)$  in Eq. (47) is  $\exp(\frac{3}{2}(\lambda x)^2) \approx 1 + \mathcal{O}((\lambda x)^2)$ , whose leading term is quadratically smaller than the one of  $\|V_0^{(p)}(x)\|_1 = \exp(\lambda x) \approx 1 + \mathcal{O}(\lambda x)$  when  $\lambda x \ll 1$ . We will later see that  $V_0^{(p)}(x)$  provides us an efficient random-sampling implementation of the Trotter-LCU algorithm.

### B. First-order case

Following similar ideas in Sec. IV A, we now study the PTS compensation of the Trotter remainder  $V_K(x)$ . We will

take first-order case as an example. The Trotter remainder  $V_1(x)$  is

$$V_1(x) = U(x)S_1(x)^\dagger. \quad (55)$$

From Eqs. (23) and (42) we have

$$\begin{aligned} S_1(x)^\dagger &= \overrightarrow{\prod} e^{ix\tilde{H}} = \left( \prod e^{\tilde{\alpha}x} \right) \sum_{\vec{r}} \text{Poi}(\vec{r}; \vec{\alpha}x) \overrightarrow{\prod} (i\vec{P})^{\vec{r}} \\ U(x) &= \sum_{r=0}^{\infty} \frac{(\lambda x)^r}{r!} \sum_{l_1, \dots, l_r} p_{l_1} p_{l_2} \cdots p_{l_r} (-i)^r P_{l_1} P_{l_2} \cdots P_{l_r} \\ &= e^{\lambda x} \sum_{r=0}^{\infty} \text{Poi}(r; \lambda x) \sum_{l_{1:r}} p_{l_{1:r}}^{(r)} (-i)^r P_{l_{1:r}}^{(r)}. \end{aligned} \quad (56)$$

Here, we adopt the vector notations introduced in Sec. III A to simplify the expressions. In Eq. (56), we also extend the notation of Poisson distribution,

$$\text{Poi}(\vec{r}; \vec{\alpha}x) := \prod_{l=1}^L \text{Poi}(r_l; \alpha_l x). \quad (57)$$

Based on Eq. (56), we then write the Taylor-series expansion of first-order remainder as follows:

$$\begin{aligned} V_1(x) &= U(x)S_1(x)^\dagger \\ &= e^{2\lambda x} \sum_{r; \vec{r}} \text{Poi}(r, \vec{r}; \lambda x, \vec{\alpha}x) \sum_{l_{1:r}} p_{l_{1:r}}^{(r)} (-i)^{r-\sum \vec{r}} P_{l_{1:r}}^{(r)} \overrightarrow{\prod} \vec{P}^{\vec{r}}, \end{aligned} \quad (58)$$

which is a LCU formula with 1-norm  $e^{2\lambda x}$ . Here,  $r$  and  $\vec{r}$  are two groups of independent variables.

Now, we utilize the order condition of the Trotter formula in Lemma 1 to reduce 1-norm of Eq. (58). To this end, we first rewrite Eq. (58) by classifying the terms based on the power of  $x$ , which is determined by the value  $s = r + \sum \vec{r}$ ,

$$V_1(x) = \sum_{s=0}^{\infty} F_{1,s}(x) = \sum_{s=0}^{\infty} \|F_{1,s}(x)\|_1 V_{1,s}. \quad (59)$$

Here,  $\|F_{1,s}(x)\|_1$  is the 1-norm of the  $s$ -order expansion formula  $F_{1,s}(x)$ .  $V_{1,s}$  denotes the normalized LCU formula for

the  $s$ -order terms. We have

$$F_{1,s}(x) := \sum_{r+\sum \vec{r}=s} \frac{(\lambda x)^r}{r!} \left( \prod \frac{(\vec{\alpha}x)^{\vec{r}}}{\vec{r}!} \right).$$

$$\sum_{l_{1:r}} p_{l_{1:r}}^{(r)} (-i)^{2r-s} P_{l_{1:r}}^{(r)} \prod \vec{P}^{\vec{r}},$$

$$V_{1,s} := \sum_{r;\vec{r}} \Pr(r, \vec{r}|s) \sum_{l_{1:r}} p_{l_{1:r}}^{(r)} (-i)^{2r-s} P_{l_{1:r}}^{(r)} \prod \vec{P}^{\vec{r}}. \quad (60)$$

In the expression of  $V_{1,s}$ , we use  $r - \sum \vec{r} = r - (s - r) = 2r - s$ . The conditional probability  $\Pr(r, \vec{r}|s)$  indicates the probability to sample  $r$  and  $\vec{r}$  when their summation  $s$  is given,

$$\Pr(r, \vec{r}|s) = \frac{\Pr\{\hat{r} = r, \hat{\vec{r}} = \vec{r} \cap \{\hat{s} = s\}\}}{\Pr\{\hat{s} = s\}}$$

$$= \frac{s!}{r! \prod \vec{r}!} \left( \frac{1}{2} \right)^r \prod \left( \frac{\vec{p}}{2} \right)^{\vec{r}}$$

$$= \text{Mul} \left( \{r, \vec{r}\}; \left\{ \frac{1}{2}, \frac{\vec{p}}{2} \right\}; s \right), \quad (61)$$

which is a multinomial distribution  $\text{Mul}(\cdot; \cdot; s)$  with  $s$  trials and  $(L + 1)$  outcomes. In each trial, the  $(L + 1)$  outcomes  $\{r; r_1, r_2, \dots, r_L\}$  occur with the corresponding probability  $\{\frac{1}{2}, \frac{1}{2}p_1, \frac{1}{2}p_2, \dots, \frac{1}{2}p_L\}$ . Recall that  $\vec{p} = \{p_1, p_2, \dots, p_L\}$  is the normalized Hamiltonian coefficients defined in Eq. (3).

We first estimate the normalization cost  $\|F_{1,s}(x)\|_1$  for different  $s$  orders,

$$\|F_{1,s}(x)\|_1 = \sum_{s;\vec{r}} \mathbb{1} \left[ r + \sum \vec{r} = s \right] \frac{(\lambda x)^r}{r!} \left( \prod \frac{(\vec{\alpha}x)^{\vec{r}}}{\vec{r}!} \right)$$

$$= \frac{(\lambda x + \sum \vec{\alpha}x)^s}{s!} = \frac{(2\lambda x)^s}{s!} =: \eta_s. \quad (62)$$

In the second inequality, we use the following equation:

$$\sum_{s_1, s_2=0}^r \mathbb{1}[s_1 + s_2 = r] \frac{(x_1)^{s_1}}{s_1!} \frac{(x_2)^{s_2}}{s_2!} = \frac{(x_1 + x_2)^r}{r!}. \quad (63)$$

We denote  $\eta_s := (2\lambda x)^s/s!$ , which will be frequently used in the following discussion.

From Eq. (62) we can see that, similar to the expansion of  $V_0(x)$ , the 1-norm of the expansion terms with different orders of  $x$  in  $V_1(x)$  follow the Poisson distribution. This motivates us to eliminate the low-order terms such as

$F_{1,1}(x)$ . Based on Lemma 1, we have

$$F_{1,1}(x) = 0. \quad (64)$$

As a result, we can directly remove the term  $F_{1,1}(x)$  in Eq. (59). The resulting formula is,

$$V_1(x) = I + \sum_{s=2}^{\infty} \eta_s V_{1,s}. \quad (65)$$

After the elimination of the first-order term, we now introduce Euler's formula to suppress higher-order terms in  $V_{1,2}$  and  $V_{1,3}$ . For the convenience of later discussion, we simplify the notation of  $F_{1,s}(x)$  in Eq. (60) as follows:

$$F_{1,s}(x) = \eta_s \sum_{r,\gamma} \Pr(r, \gamma|s) P_1(r, \gamma), \quad (66)$$

where  $\gamma$  is used to denote all the expansion variables  $\{\vec{r}, l_{1:r}\}$  besides  $r$ .  $\Pr(r, \gamma|s)$  and  $P_1(r, \gamma)$  are then defined to be

$$\Pr(r, \gamma|s) := \Pr(r, \vec{r}|s) p_{l_{1:r}}^{(r)}, \quad (67)$$

$$P_1(r, \gamma) := (-i)^{2r-s} P_{l_{1:r}}^{(r)} \prod \vec{P}^{\vec{r}}.$$

To apply Euler's formula in Eq. (45) to Eq. (65), we need to make sure that the Pauli operator  $P$  is Hermitian. To this end, we classify the Pauli terms of  $F_{1,s}$  in Eq. (66) into Hermitian and anti-Hermitian types,

$$F_{1,s}(x) = \eta_s \sum_{\{r,\gamma\} \in \text{Her}} \Pr(r, \gamma|s) P_{1,s}(r, \gamma)$$

$$+ i\eta_s \sum_{\{r,\gamma\} \in \text{anti-Her}} \Pr(r, \gamma|s) (-i) P_{1,s}(r, \gamma), \quad (68)$$

where  $\{r, \gamma\} \in \text{Her}$  and  $\{r, \gamma\} \in \text{anti-Her}$ , respectively, indicates the set of  $\{r, \gamma\}$  such that  $P_1(r, \gamma)$  is Hermitian and anti-Hermitian. When  $\{r, \gamma\} \in \text{Her}$ , the corresponding Pauli operator owns a real coefficient, which cannot be paired with  $I$  based on Euler's formula.

It seems that we cannot eliminate the Hermitian terms in Eq. (68) by Euler's formula. However, by taking advantage of the properties of Trotter remainder  $V_1(x)$ , we can show that  $V_{1,2}$  and  $V_{1,3}$  are actually anti-Hermitian. Recall that we can write the exponential form of  $V_1(x)$  by applying the BCH formula on the definition of  $V_1(x)$  in Eq. (27),

$$V_1(x) = \exp(i(E_{1,2}x^2 + E_{1,3}x^3 + E_{1,4}x^4 + \dots)), \quad (69)$$

where  $\{E_{1,s}\}$  are some Hermitian operators determined by the BCH formula. The first-order term  $E_{1,1}$  vanishes due to the order condition in Lemma 2.

*Lemma 2 (Lemma 1 in Ref. [13]).* Let  $F(x)$  be an operator-valued function that is infinitely differentiable. Let  $K \geq 1$  be a non-negative integer. The following two conditions are equivalent.

- (1) Asymptotic scaling:  $F(x) = \mathcal{O}(x^{K+1})$ .
- (2) Derivative condition:  $F(0) = F'(0) = \dots = F^{(K)}(0) = 0$ .

From Eq. (69) we then have

$$\begin{aligned} F_{1,2}(x) &= ix^2 E_{1,2}, \\ F_{1,3}(x) &= ix^3 E_{1,3} \end{aligned} \quad (70)$$

by the Taylor-series expansion on Eq. (69).

Comparing Eqs. (68) and (70), we can see that

$$\sum_{\{r,\gamma\} \in \text{Her}} \Pr(r, \gamma|s) P_{1,s}(r, \gamma) = 0, \quad s = 2, 3. \quad (71)$$

This is because  $F_{1,2}(x)$  and  $F_{1,3}(x)$  are anti-Hermitian from Eq. (70).

We can then modify the form of  $F_{1,s}(x)$  ( $s = 2$  or  $3$ ) in Eq. (68) as follows:

$$\begin{aligned} F_{1,s}(x) &= i\eta_s \sum_{\{r,\gamma\} \in \text{Her}} \Pr(r, \gamma|s) P_{1,s}(r, \gamma) \\ &\quad + i \sum_{\{r,\gamma\} \in \text{anti-Her}} \Pr(r, \gamma|s) (-i) P_{1,s}(r, \gamma), \\ &= i\eta_s \sum_{r,\gamma} \Pr(r, \gamma|s) (-i)^{\mathbb{1}[P_{1,s}(r,\gamma):\text{anti-Her}]} P_{1,s}(r, \gamma). \end{aligned} \quad (72)$$

In Eq. (72), we intentionally add an extra phase  $i$  on the Hermitian terms, which has no effect on  $F_{1,s}(x)$  as they own zero summation value. In this way, all the Pauli expansion terms in  $F_{1,2}(x)$  and  $F_{1,3}(x)$  are with imaginary coefficient, which can be paired with  $I$  using the Euler's formula in Eq. (45). We call the second- and third-order terms  $V_{1,s}$  ( $s = 2, 3$ ) the leading TS expansion orders of  $V_1(x)$ . The major reason to the Hermitian terms with zero summation value is to simplify the sampling procedure of  $V_{1,s}$ , which will be clarified later.

Now, we are going to eliminate the leading order terms in  $V_1(x)$ ,

$$\begin{aligned} V_1(x) &= I + \sum_{s=2}^3 i\eta_s V'_{1,s} + \sum_{s>3} \eta_s V_{1,s}, \\ V'_{1,s} &= \sum_{r,\gamma} \Pr(r, \gamma|s) P'_{1,s}(r, \gamma), \quad s = 2, 3 \\ V_{1,s} &= \sum_{r,\gamma} \Pr(r, \gamma|s) (-i)^{2r-s} P_{1,s}(r, \gamma), \quad s \geq 4. \end{aligned} \quad (73)$$

Here, the Pauli operator for the leading-order term  $P'_{1,s}(r, \gamma) := (-i)^{\mathbb{1}[P_{1,s}(r,\gamma):\text{anti-Her}]} P_{1,s}(r, \gamma)$  is always with a real coefficient. We can then pair  $I$  with the Pauli operators in  $F_{1,2}(x)$  and  $F_{1,3}(x)$ ,

$$\begin{aligned} V_1^{(p)}(x) &= I + i\eta_2 V'_{1,2} + i\eta_3 V'_{1,3} + \sum_{s=4}^{\infty} \eta_s V_{1,s} \\ &= \frac{\eta_2}{\eta_\Sigma} (I + \eta_\Sigma V'_{1,2}(x)) + \frac{\eta_3}{\eta_\Sigma} (I + \eta_\Sigma V'_{1,3}(x)) + \sum_{s=4}^{\infty} \eta_s V_{1,s} \\ &= \sqrt{1 + \eta_\Sigma^2} \left( \frac{\eta_2}{\eta_\Sigma} R_{1,2}(\eta_\Sigma) + \frac{\eta_3}{\eta_\Sigma} R_{1,3}(\eta_\Sigma) \right) + \sum_{s=4}^{\infty} \eta_s V_{1,s}. \end{aligned} \quad (74)$$

Here,  $\eta_\Sigma := \eta_2 + \eta_3$ . The third line of Eq. (74) is the final LCU formula used for the first-order PTSC algorithm. In the third line, we apply the following pairing procedure:

$$\begin{aligned} I + \eta_\Sigma V'_{1,s} &= I + i\eta_\Sigma \sum_{r,\gamma} \Pr(r, \gamma|s) P'_{1,s}(r, \gamma) \\ &= \sum_{r,\gamma} \Pr(r, \gamma|s) (I + \eta_\Sigma P'_{1,s}(r, \gamma)) \\ &= \sqrt{1 + \eta_\Sigma^2} \sum_{r,\gamma} \Pr(r, \gamma|s) \exp(i\theta(\eta_\Sigma) P'_{1,s}(r, \gamma)) \\ &=: \sqrt{1 + \eta_\Sigma^2} R_{1,s}(\eta_\Sigma), \quad s = 2, 3. \end{aligned} \quad (75)$$

In practice, we introduce a truncation  $s_c$  on the expansion formula in Eq. (74). For the convenience of analysis, we set  $s_c > 3$ . The truncated LCU formula for  $V_1^{(p)}(x)$  is then

$$\tilde{V}_1^{(p)}(x) = \mu_1^{(p)}(x) \left( \sum_{s=2,3} \Pr_1^{(p)}(s) R_{1,s}(\eta_\Sigma) + \sum_{s=4}^{s_c} \Pr_1^{(p)}(s) V_{1,s} \right), \quad (76)$$

with the 1-norm  $\mu_1^{(p)}(x)$  and probabilities  $\Pr_1^{(p)}(s)$  to sample the  $s$ -order term

$$\begin{aligned} \mu_1^{(p)}(x) &= \sqrt{1 + \eta_\Sigma^2} + \sum_{s=4}^{s_c} \eta_s, \\ \Pr_1^{(p)}(s) &= \frac{1}{\mu_1^{(p)}(x)} \begin{cases} \sqrt{1 + \eta_\Sigma^2} \frac{\eta_s}{\eta_\Sigma}, & s = 2, 3, \\ \eta_s, & s = 4, 5, \dots, s_c. \end{cases} \end{aligned} \quad (77)$$

Combined with the deterministic first-order Trotter formula, the overall LCU formula for  $U(x)$  is

$$\tilde{U}_1^{(p)}(x) = \tilde{V}_1^{(p)}(x) S_1(x). \quad (78)$$

The following proposition gives the performance characterization of  $\tilde{U}_1^{(p)}(x)$  in Eq. (78) to approximate  $U(x)$ .

*Proposition 4 (First-order Trotter-LCU formula by paired Taylor-series compensation).* For  $0 < x < 1/(2\lambda)$  and  $s_c \geq 3$ ,  $\tilde{V}_1(x)$  in Eq. (76) is a  $(\mu_1^{(p)}(x), \varepsilon_1^{(p)}(x))$ -LCU formula of the first-order Trotter remainder  $V_1(x)$  with

$$\mu_1^{(p)}(x) \leq e^{(e+\frac{2}{9})(2\lambda x)^4}, \quad \varepsilon_1^{(p)}(x) \leq \left(\frac{2e\lambda x}{s_c + 1}\right)^{s_c+1}. \quad (79)$$

As a result,  $\tilde{U}_1(x)$  in Eq. (78) is a  $(\mu_1^{(p)}(x), \varepsilon_1^{(p)}(x))$ -LCU formula of  $U(x)$ .

*Proof.* We first bound the normalization factor  $\mu_1^{(p)}(x)$ . When  $2\lambda x < 1$  we have

$$\begin{aligned} \mu_1^{(p)}(x) &\leq \sqrt{1 + (\eta_\Sigma)^2} + \sum_{s=4}^{\infty} \eta_s \\ &\leq 1 + \frac{1}{2}\eta_\Sigma^2 + \left(e^{2\lambda x} - \sum_{s=0}^3 \eta_s\right) \\ &= \frac{1}{2}(2\lambda x)^4 \left(\frac{1}{2!} + \frac{(2\lambda x)}{3!}\right)^2 + \left(e^{2\lambda x} - \sum_{s=1}^3 \eta_s\right) \\ &\leq \frac{1}{2}(2\lambda x)^4 \left(\frac{1}{2} + \frac{1}{6}\right)^2 + \left(e^{2\lambda x} - \sum_{s=1}^3 \eta_s\right) \\ &\leq \frac{2}{9}(2\lambda x)^4 + e^{e(2\lambda x)^4} \leq e^{(e+\frac{2}{9})(2\lambda x)^4}. \end{aligned} \quad (80)$$

For the distance bound, from Eqs. (58) and (76) we have

$$\begin{aligned} \|V_1(x) - \tilde{V}_1^{(p)}(x)\| &\leq \sum_{s>s_c} \eta_s \|V_{1,s}\| \\ &= \sum_{s>s_c} \frac{(2\lambda x)^s}{s!} \leq \left(\frac{2e\lambda x}{s_c + 1}\right)^{s_c+1} \end{aligned} \quad (81)$$

In the second line, we use the fact that  $\|V_{1,s}\| \leq 1$ . In the third line, we apply Eq. (54). ■

From Proposition 4 we have shown that, by introducing first-order Trotter formula, we can further suppress the 1-norm of the LCU formula  $V_1^{(p)}(x)$  to  $\exp(c(\lambda x)^4) \approx 1 + \mathcal{O}(\lambda x)^4$ , where  $c$  is a constant. In Appendix A, we discuss the generalized LCU formula construction of higher-order Trotter remainder  $V_K(x)$  with  $K = 2k, k \in \mathbb{N}_+$ . Under such constructions, we have,

$$\|V_K^{(p)}(x)\|_1 = \exp(c(\lambda x)^{2K+2}). \quad (82)$$

In Sec. IV C we will see how this can help us to improve the time scaling of the whole simulation algorithm.

### C. Random-sampling implementation and performance

We have now derived the LCU formulas for Trotter remainders  $V_K(x)$  and hence the small time evolution  $U(x) = V_K(x)S_K(x)$  based on the idea to utilize the Trotter order condition and pair the leading-order terms in  $V_K(x)$  to suppress the normalization factors. We now discuss the practical random-sampling implementation of them, taking the first-order case as an example.

Suppose we want to perform Hamiltonian evolution  $e^{iHt}$  on an initial state  $\rho$ . As is illustrated in Fig. 2, in each segment, we first implement the Trotter circuit  $S_1(x)$  and then compensate the remainder  $V_1(x)$  by LCU. In the random-sampling implementation of LCU, we embed the LCU sampling into a modified Hadamard test. In Fig. 5, we show the detailed sampling procedure of  $V_i$  and  $V_j$ . In stage (1), we sample the Taylor-expansion order  $s$  from a finite probability distribution  $\text{Pr}_1^{(p)}(s)$ . Afterwards, in stage (2) and (3), we randomly sample Pauli string indices  $\{r, \bar{r}\}$  and  $l_1, \dots, l_r$  based on the LCU formula of  $F_{1,s}$ . The variables  $\{r, \bar{r}\}$  obey a multinomial distribution  $\text{Mul}(r, \bar{r}; \{\frac{1}{2}, \frac{\bar{r}}{2}\}; s)$  defined in Eq. (61) while  $\{l_1, l_2, \dots, l_r\}$  are sampled identically and independently from normalized Hamiltonian coefficients  $\bar{p}$  defined in Eq. (3). Finally, in stage (4), depending on whether  $s$  is the leading order ( $s = 2, 3$  for  $K = 1$ ) or not, we determine the sampled unitary  $V_i$  to be a Pauli-rotation unitary  $e^{i\theta P}$  or just Pauli operators  $P$ .

For the gate complexity of the random-sampling implementation, we have the following theorem.

*Theorem 3 (Gate complexity of the  $K$ th-order random-sampling Trotter-LCU algorithm by paired Taylor-series compensation).* To realize a (probabilistic) Hamiltonian simulation of  $e^{iHt}$  with accuracy  $\varepsilon$ , the gate complexity of random-sampling  $K$ th-order Trotter-LCU algorithm ( $K = 0, 1$  or  $2k$ ) based on paired Taylor-series compensation is

$$\mathcal{O}\left((\lambda t)^{1+\frac{1}{2K+1}} \left(\kappa_K L + \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)}\right)\right). \quad (83)$$

Here,  $\kappa_K = K$  if  $K = 0$  or  $1$ , or  $\kappa_K = 2 \times 5^{K/2-1}$  if  $K = 2k$ .

*Proof.* Without loss of generality, we focus on the case when  $K = 1$ . The case of  $K = 0$  and  $K = 2k, k \in \mathbb{N}_+$  can be analyzed similarly following Propositions 3 and 10, respectively.

For the random-sampling implementation, the overall LCU formula for  $U(t)$  is to repeat the sampling of  $\tilde{U}_1^{(p)}(x)$  for  $\nu$  times,  $\tilde{U}_{1,\text{tot}}^{(p)}(t) = \tilde{U}_1^{(p)}(x)^\nu$ . Using Propositions 2 and 4, when  $0 < x < 1/2\lambda$  and  $s_c \geq 3$ , we conclude that

$\tilde{U}_{1,\text{tot}}^{(p)}(t)$  is a  $(\mu_{1,\text{tot}}^{(p)}(t), \varepsilon_{1,\text{tot}}^{(p)}(t))$ -LCU formula of  $U(t)$  with

$$\begin{aligned} \mu_{1,\text{tot}}^{(p)}(t) &= \mu_1^{(p)}(x)^\nu \leq e^{(e+c_1)\frac{(2\lambda t)^4}{\nu^3}}, \\ \varepsilon_{1,\text{tot}}^{(p)}(t) &\leq \nu \mu_{1,\text{tot}}^{(p)}(t) \varepsilon_1^{(p)}(x) \leq \nu e^{(e+c_1)\frac{(2\lambda t)^4}{\nu^3}} \left( \frac{2e\lambda x}{s_c + 1} \right)^{s_c+1}. \end{aligned} \quad (84)$$

Here,  $c_1 = 2/9$ .

To realize a  $(\mu, \varepsilon)$ -LCU formula for  $U(t)$ , we only need to set the segment number  $\nu$  and the truncation order  $s_c$  to satisfy

$$\begin{aligned} \nu &\geq \max \left\{ \nu_1^{(p)}(t), 2\lambda t \right\}, \\ s_c &\geq \max \left\{ \left\lceil \frac{\ln \left( \frac{\mu}{\varepsilon} \nu_1^{(p)}(t) \right)}{W_0 \left( \frac{1}{2e\lambda t} \nu_1^{(p)}(t) \ln \left( \frac{\mu}{\varepsilon} \nu_1^{(p)}(t) \right) \right)} - 1 \right\rceil, 3 \right\}, \end{aligned} \quad (85)$$

we can then realize a  $(\mu, \varepsilon)$ -LCU formula for  $U(t)$  based on  $\nu$  segments of  $\tilde{U}_1^{(p)}$  in Eq. (78). Here,  $\nu_1^{(p)}(t) := (2(e+c_1)\lambda t / \ln \mu)^{1/3} 2\lambda t$ .  $W_0(y)$  is the principle branch of the Lambert  $W$  function whose scaling is approximately  $\ln(y)$  according to the tight bound in Lemma 7 in Appendix F. To derive the bound for  $s_c$  in the second line of Eq. (85), we use Lemma 6 in Appendix F.

The gate complexity of the random-sampling implementation of the Trotter-LCU algorithm is determined by the segment number  $\nu$ , Trotter order  $K$  and the gate complexity of each elementary gate in the LCU formula. As shown in Fig. 5, to construct controlled- $U(t)$ , we split it to  $\nu$  segments. In each segment, we need to implement  $K$ th-order Trotter circuits and LCU circuits. The gate complexity of each elementary gate in the LCU circuit is determined by the truncation order  $s_c$  of the Taylor-series compensation. Specifically, we consider the compilation of the controlled-Pauli gate and controlled-Pauli rotation gate, as shown in Fig. 11. The number of gates is determined by the weight of the sampled Pauli matrices  $\text{wt}(P_l)$ , which is upper bounded by  $\mathcal{O}(s_c)$ .

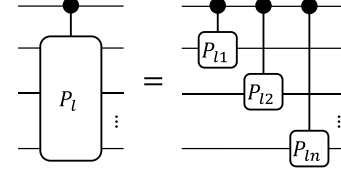
Therefore, the gate complexity of the overall algorithm using  $K$ th Trotter formula ( $K = 0, 1, 2k$ ) is given by

$$N_K^{(RS)} = \mathcal{O}(\nu(\kappa_K L + s_c)) \quad (86)$$

where

$$\kappa_K = \begin{cases} 0, & K = 0, \\ 1, & K = 1, \\ 2 \cdot 5^{K/2-1}, & K = 2k, k \in \mathbb{N}_+. \end{cases} \quad (87)$$

(a)  $V_i/V_j$ :  $n$ -qubit Pauli matrices  $P_l$



(b)  $V_i/V_j$ :  $n$ -qubit Pauli rotation unitary  $e^{i\theta_s P_l}$

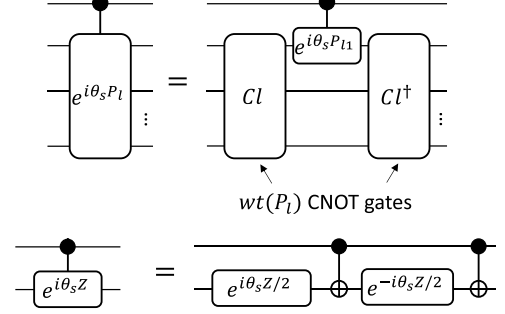


FIG. 11. (a) Compilation of the LCU circuit when  $V_i(V_j)$  is a  $n$ -qubit Pauli gate. (b) Compilation of the LCU circuit when  $V_i(V_j)$  is a Pauli rotation unitary. Here,  $Cl$  is a Clifford gate with  $\text{wt}(P_l)$  CNOT gates.

Based on Eq. (86), the gate complexity of the  $2k$ th-order Trotter-LCU algorithm is then

$$\mathcal{O}(\nu(\kappa_K L + s_c)) = \mathcal{O} \left( (\lambda t)^{1+\frac{1}{4k+1}} (\kappa_K L + \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)}) \right). \quad (88)$$

Here,  $\kappa_K$  is the stage number of the Trotter formula. When  $K = 0$ ,  $\kappa_K = 0$ . When  $K = 2k$ ,  $\kappa_K = 2 \times 5^{K/2-1}$ . ■

From Theorem 3 we can see that, by introducing the LCU compensation, the time scaling of the  $K$ th-order Trotter-LCU algorithm improve the bare  $K$ th-order Trotter time scaling from  $1 + 1/K$  to  $1 + 1/(2K + 1)$ . Moreover, the accuracy scaling is exponentially improved. This allows us to achieve optimal gate complexity with lower-order Trotter implementation. For example, the time dependence of using first-order (respectively, second-order) Trotter formula can be improved from  $t^2$  (respectively,  $t^{1+1/2}$ ) to  $t^{1+1/3}$  (respectively,  $t^{1+1/5}$ ) by adding LCU compensation.

## V. TROTTER-LCU FORMULA WITH NESTED-COMMUTATOR COMPENSATION

In this section, we provide detailed construction of the nested-commutator compensation Trotter-LCU algorithms and the gate complexity analysis. We first sketch the procedure to derive the nested-commutator form LCU

formula in Sec. VA. Then we construct the LCU formula of the first-order Trotter remainder of the lattice Hamiltonians in Sec. VB as an example. In Sec. VC, we describe the random-sampling implementation of the nested-commutator compensation algorithm and analyze its gate complexity.

### A. Derivation of the nested-commutator formula

Our aim is to expand the LCU formula for the  $K$ th-order Trotter remainder  $V_K(x) := U(x)S_K(x)^\dagger$  ( $K = 1$  or  $2k$ ,  $k \in \mathbb{N}_+$ ) in the following form:

$$V_K(x) = I + \sum_{s=K+1}^{2K+1} F_{K,s}^{(nc)}(x) + F_{K,\text{res}}^{(nc)}(x), \quad (89)$$

where the leading-order terms  $F_{K,s}^{(nc)}(x)$  are written as a summation of nested commutators. We will use the following lemma of the operator-valued differential equation.

*Lemma 3 (Lemma A.1 in Ref. [20]).* Let  $H(x)$ ,  $R(x)$  be continuous operator-valued functions defined for  $x \in \mathbb{R}$ . Then, the first-order differential equation

$$\frac{d}{dx}W(x) = H(x)W(x) + R(x), \quad W(0) \text{ known}, \quad (90)$$

has a unique solution given by

$$W(x) = \mathcal{T} \exp\left(\int_0^x d\tau H(\tau)\right) W(0) + \int_0^x d\tau_1 \mathcal{T} \exp\left(\int_{\tau_1}^x d\tau_2 H(\tau_2)\right) R(\tau_1). \quad (91)$$

Here,  $\mathcal{T}$  is the time-ordering operator.

In Eq. (90), if we set  $W(x)$  to be the real-time evolution  $U(x) = e^{-iHx}$ , we can find that  $H(x) = -iH$  and  $R(x) = 0$ . Therefore,  $R(x)$  reflects the derivation of  $W(x)$  from the exponential function. We are going to apply Lemma 3 to  $S_K(x)^\dagger$ , i.e., we set  $W(x) := S_K(x)^\dagger$  and  $H(x) = iH$ . The deviation of  $S_K(x)^\dagger$  from  $U(x)^\dagger = e^{ixH}$  is characterized by the following function:

$$R_K(x) = \frac{d}{dx}S_K(x)^\dagger - iHS_K(x)^\dagger. \quad (92)$$

Applying Lemma 3, we have

$$S_K(x)^\dagger = e^{ixH} + \int_0^x d\tau e^{i(x-\tau)H} R_K(\tau). \quad (93)$$

The Trotter remainder  $V_K(x)$  can then be expressed as

$$\begin{aligned} V_K(x) &= U(x)S_K(x)^\dagger = I + \int_0^x d\tau e^{-i\tau H} R_K(\tau) \\ &= I + \int_0^x d\tau U(\tau)S_K(\tau)^\dagger S_K(\tau)R_K(\tau) \\ &= I + \int_0^x d\tau V_K(\tau)J_K(\tau), \end{aligned} \quad (94)$$

where

$$J_K(x) := S_K(x)R_K(x). \quad (95)$$

Equation (94) provides a recurrence formula to solve the expansion terms in  $V_K(x)$ . To be more explicit, if we expand  $V_K(x)$  and  $J_K(x)$  based on the operator-valued Taylor series,

$$V_K(x) = \sum_{s=0}^{\infty} G_s \frac{x^s}{s!}, \quad J_K(x) = \sum_{s=0}^{\infty} C_s \frac{x^s}{s!}, \quad (96)$$

where  $G_s$  and  $C_s$  denotes the respective  $s$ -order term, then we have

$$\begin{aligned} G_{s+1} &= \frac{(s+1)!}{x^{s+1}} \int_0^x d\tau \sum_{m=0}^s G_m C_{s-m} \frac{\tau^s}{m!(s-m)!} \\ &= \sum_{m=0}^s \binom{s}{m} G_m C_{s-m} \end{aligned} \quad (97)$$

since Eq. (94) holds for all  $x \in \mathbb{R}$ . This is a recurrence formula, which can be used to solve all the expansion terms  $G_s$  of the remainder  $V_K(x)$  from the expansion terms  $C_r$  of  $J_K(x)$ .

To solve the explicit form of  $G_s$ , we need to study the form of expansion terms  $\{C_r\}_{r=0}^{s-1}$  for the function  $J_K(x)$ . We will use the following proposition derived from Lemma 2.

*Proposition 5 (Order condition).* For the  $K$ th-order Trotter formula  $S_K(x)$ , the multiplicative remainder  $V_K(x)$  and derivative remainder  $J_K(x)$  defined in Eqs. (89), (92), and (95), respectively, the following statements are equivalent.

- (1)  $S_K(x) = U(x) + \mathcal{O}(x^{K+1})$ ,
- (2)  $(S_K(0))^{(j)} = (-iH)^j$ , for  $0 \leq j \leq K$ ,
- (3)  $J_K(x) = \mathcal{O}(x^K)$ ,
- (4)  $(J_K(0))^{(j)} = 0$ , for  $0 \leq j \leq K-1$ ,
- (5)  $V_K(x) = I + \mathcal{O}(x^{K+1})$ ,
- (6)  $V_K(0) = I$  and  $(V_K(0))^{(j)} = 0$ , for  $1 \leq j \leq K$ .

Here,  $f(x)^{(j)}$  is the  $j$ th-derivative of the function  $f(x)$ .

*Proof.* First, we note that (1)  $\Leftrightarrow$  (2), (3)  $\Leftrightarrow$  (4), and (5)  $\Leftrightarrow$  (6) by applying Lemma 2 and setting  $F(x)$  to be

$S_K(x) - U(x)$ ,  $J_K(x)$ , and  $V_K(x) - I$ , respectively. So we only need to prove (2)  $\Leftrightarrow$  (4) and (2)  $\Leftrightarrow$  (6).

We first prove (2)  $\Leftrightarrow$  (4). From (2) we also have  $(S_K(x)^\dagger)^{(j)} = (iH)^j$  for  $0 \leq j \leq K$ . Based on Eqs. (92) and (95), the derivatives of  $R_K(x)$  and  $J_K(x)$  are

$$\begin{aligned} (R_K(x))^{(j)} &= (S_K(x)^\dagger)^{(j+1)} - iH(S_K(x)^\dagger)^{(j)}, \\ (J_K(x))^{(j)} &= \sum_{l=0}^j \binom{j}{l} (S_K(x))^{(l)} (R_K(x))^{(j-l)}. \end{aligned} \quad (98)$$

Based on (2) we have  $(R_K(0))^{(j)} = 0$  for  $0 \leq j \leq K-1$ . Hence  $(J_K(0))^{(j)} = 0$  for  $0 \leq j \leq K-1$ .

For the reverse direction, we notice that  $R_K(x) = J_K(x)S_K(x)^\dagger$ . This implies

$$(R_K(x))^{(j)} = \sum_{l=0}^j \binom{j}{l} (J_K(x))^{(l)} (S_K(x)^\dagger)^{(j-l)}. \quad (99)$$

If we have  $(J_K(0))^{(j)} = 0$  for  $0 \leq j \leq K-1$ , then  $(R_K(0))^{(j)} = 0$  for  $0 \leq j \leq K-1$ . Then, from Eq. (98) we have  $(S_K(x)^\dagger)^{(j)} = (iH)^j$  for  $0 \leq j \leq K$ , which is equivalent to (2).

Now, we prove (2)  $\Leftrightarrow$  (6). Based on Eq. (89) we have

$$(V_K(x))^{(j)} = \sum_{l=0}^j \binom{j}{l} (U(x))^{(l)} (S_K(x)^\dagger)^{(j-l)}. \quad (100)$$

From (2) we have

$$(V_K(0))^{(j)} = \sum_{l=0}^j \binom{j}{l} (-iH)^l (iH)^{j-l} = 0, \quad (101)$$

for  $1 < j < K$ . The reverse direction can be proven similarly based on the derivative of the formula  $S_K(x) = V_K(x)^\dagger U(x)$ . ■

From Proposition 5, we have the following order condition for the  $K$ th-order Trotter formula and remainders:

$$\begin{aligned} S_K(x) &= U(x) + \mathcal{O}(x^{K+1}), \\ J_K(x) &= \mathcal{O}(x^K), \\ V_K(x) &= I + \mathcal{O}(x^{K+1}). \end{aligned} \quad (102)$$

Compare the recurrence formula in Eq. (94) and the order condition in Eq. (102), we will obtain the following relationship for the Taylor-series expansion terms  $G_s$  and  $C_s$

for  $V_K(x)$  and  $J_K(x)$ , respectively,

$$\begin{aligned} G_s &= 0, \quad s = 1, 2, \dots, K, \\ C_s &= 0, \quad s = 0, 1, \dots, K-1, \\ G_s &= C_{s-1}, \quad s = K+1, K+2, \dots, 2K+1. \end{aligned} \quad (103)$$

Based on Eq. (103), we are going to expand  $J_K(x)$  based on the operator-valued Taylor-series expansion with integral remainders,

$$J_K(x) = J_{K,L}(x) + J_{K,\text{res},2K}(x) = \sum_{s=K}^{2K} C_s \frac{x^s}{s!} + J_{K,\text{res},2K}(x), \quad (104)$$

where

$$\begin{aligned} C_s &= J_K^{(s)}(0), \quad s = K, K+1, \dots, 2K, \\ J_{K,\text{res},s}(x) &= \int_0^x d\tau \frac{(x-\tau)^s}{s!} J_K^{(s+1)}(\tau). \end{aligned} \quad (105)$$

Here,  $J_{K,L}(x) := \sum_{s=K}^{2K} C_s (x^s)/(s!)$  denotes the leading-order terms in  $J_K(x)$ . Then, from Eqs. (94) and (103), the  $K$ th-order Trotter remainder can be expressed as

$$\begin{aligned} V_K(x) &= I + M_K(x) = I + \sum_{s=K+1}^{2K+1} F_{K,s}^{(nc)}(x) + F_{K,\text{res}}^{(nc)}(x), \\ F_{K,s}^{(nc)}(x) &= C_{s-1} \frac{x^s}{s!}, \quad s = K+1, K+2, \dots, 2K+1, \\ F_{K,\text{res}}^{(nc)}(x) &= \int_0^x d\tau (M_K(\tau)J_{K,L}(\tau) + V_K(\tau)J_{K,\text{res},2K}(\tau)). \end{aligned} \quad (106)$$

Here,  $M_K(x) := V_K(x) - I$ .

From Eq. (106) we can see that, the leading-order expansion terms  $F_{K,s}^{(nc)}(x)$  of  $V_K(x)$  with  $s = K+1, K+2, \dots, 2K+1$  owns a simple expression related to  $C_{s-1} := J_K^{(s-1)}(0)$ . Later we will show that,  $C_s$  can be simply written as a summation of nested commutators with the form,

$$\text{ad}_{H_{l_j}}^{m_j} \cdots \text{ad}_{H_{l_2}}^{m_2} \text{ad}_{H_{l_1}}^{m_1} H_l, \quad (107)$$

where  $\sum_j m_j = s$ ,  $H_l, H_{l_1}, \dots, H_{l_j}$  are different summands in  $H$ . Furthermore, we will show that  $\{C_s\}$  are all anti-Hermitian. As a result, the expansion-order-pairing based on Euler's formula introduced in Sec. IV can also be applied here. Since  $\{C_s\}$  are anti-Hermitian, we can expand them by the Pauli operators as follows:

$$C_s = i \|C_s\|_1 \sum_{\gamma} \text{Pr}(\gamma|s) P_s^{(nc)}(\gamma), \quad (108)$$

where  $\{P_s^{(nc)}(\gamma)\}$  are Hermitian Pauli operators with coefficients 1 or  $-1$ .  $\|C_s\|_1$  denotes the 1-norm of this expansion.

The leading-order expansion term  $F_{K,s}^{(nc)}(x)$  can then be expressed as

$$\begin{aligned} F_{K,s}^{(nc)}(x) &= i \|C_{s-1}\|_1 \frac{x^s}{s!} \sum_{\gamma} \Pr(\gamma|s-1) P_{s-1}^{(nc)}(\gamma) \\ &=: \eta_s^{(nc)} V_{K,s}^{(nc)}. \end{aligned} \quad (109)$$

Here,  $\eta_s^{(nc)} := \|C_{s-1}\|_1 x^s / s!$  is the 1-norm of the LCU formula of  $F_{K,s}^{(nc)}(x)$  in Eq. (109).  $V_{K,s}^{(nc)}$  is the normalized LCU formula.

The residue term  $F_{K,\text{res}}^{(nc)}$  in Eq. (106), however, is complicated and hard to be expressed simply using nested commutators. Due to the hardness to compensate  $F_{K,\text{res}}^{(nc)}$ , we will remove it in the truncated Trotter remainder formula,

$$\begin{aligned} \tilde{V}_K^{(nc)}(x) &= I + \sum_{s=K+1}^{2K+1} F_{K,s}^{(nc)}(x) \\ &= \sum_{s=K}^{2K} \frac{\eta_s^{(nc)}}{\eta_{\Sigma}^{(nc)}} \left( I + \eta_{\Sigma}^{(nc)} V_{K,s}^{(nc)} \right) \\ &= \sqrt{1 + (\eta_{\Sigma}^{(nc)})^2} \sum_{s=K}^{2K} \frac{\eta_s^{(nc)}}{\eta_{\Sigma}^{(nc)}} R_{K,s}^{(nc)}(\eta_{\Sigma}). \end{aligned} \quad (110)$$

Here,  $\eta_{\Sigma}^{(nc)} := \sum_{s=K+1}^{2K+1} \|C_{s-1}\|_1 x^s / s!$ . In the forth line, we apply the following pairing procedure based on the Euler's formula, similar to Eq. (75),

$$\begin{aligned} I + \eta_{\Sigma}^{(nc)} V_{K,s}^{(nc)} &= I + i \eta_{\Sigma}^{(nc)} \sum_{\gamma} \Pr(\gamma|s-1) P_{s-1}^{(nc)}(\gamma) \\ &= \sqrt{1 + (\eta_{\Sigma}^{(nc)})^2} \sum_{\gamma} \Pr(\gamma|s-1) \exp\left(i\theta(\eta_{\Sigma}^{(nc)}) P_{s-1}^{(nc)}(\gamma)\right) \\ &=: \sqrt{1 + (\eta_{\Sigma}^{(nc)})^2} R_{2,s}^{(nc)}(\eta_{\Sigma}^{(nc)}), \end{aligned} \quad (111)$$

for  $s = K+1, K+2, \dots, 2K+1$ . Recall that  $\theta(x) := \tan^{-1}(x)$ .

$\tilde{V}_K^{(nc)}(x)$  in Eq. (110) is the final LCU formula for the nested-commutator compensation of  $V_K(x)$ . In what follows, we estimate the 1-norm  $\mu_K^{(nc)}(x)$  and distance  $\varepsilon_K^{(nc)}(x)$  of this LCU formula,

$$\begin{aligned} \mu_K^{(nc)}(x) &:= \sqrt{1 + (\eta_{\Sigma}^{(nc)})^2}, \\ \varepsilon_K^{(nc)}(x) &:= \|\tilde{V}_K^{(nc)}(x) - V_K(x)\| = \|F_{K,\text{res}}^{(nc)}(x)\|. \end{aligned} \quad (112)$$

*Proposition 6 (Bound the 1-norm and error of nested-commutator expansion formula).*  $\tilde{V}_K^{(nc)}(x)$  in Eq. (110) is a  $(\mu_K^{(nc)}(x), \varepsilon_K^{(nc)}(x))$  formula of the  $K$ th-order Trotter

remainder  $V_K(x)$  with

$$\begin{aligned} \mu_K^{(nc)}(x) &= \sqrt{1 + (\eta_{\Sigma}^{(nc)})^2} = \sqrt{1 + \left( \sum_{s=K}^{2K} \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2} \\ \varepsilon_K^{(nc)}(x) &\leq \int_0^x d\tau \left( \|M_K(\tau)\| \|J_{K,L}(\tau)\| + \|J_{K,\text{res},2K}(\tau)\| \right), \end{aligned} \quad (113)$$

where

$$\begin{aligned} \|J_{K,L}(\tau)\| &\leq \sum_{s=K}^{2K} \|C_s\| \frac{x^{s+1}}{(s+1)!}, \\ \|J_{K,\text{res},2K}(\tau)\| &\leq \int_0^x d\tau \frac{(x-\tau)^s}{s!} \|J_K^{(s+1)}(\tau)\|, \\ \|M_K(\tau)\| &\leq \int_0^{\tau} d\tau_1 \int_0^{\tau_1} d\tau_2 \frac{(\tau_1 - \tau_2)^{(K-1)}}{(K-1)!} \|J_K^{(K)}(\tau_2)\|. \end{aligned} \quad (114)$$

*Proof.* The value of  $\mu_K^{(nc)}(x)$  is derived based on Eqs. (109) and (110). To calculate  $\varepsilon_K^{(nc)}(x)$ , we will use the following bound:

$$\begin{aligned} \varepsilon_K^{(nc)}(x) &= \|F_{K,\text{res}}^{(nc)}(x)\| \\ &\leq \int_0^x d\tau \left( \|M_K(\tau)\| \|J_{K,L}(\tau)\| + \|V_K(\tau)\| \|J_{K,\text{res},2K}(\tau)\| \right) \\ &= \int_0^x d\tau \left( \|M_K(\tau)\| \|J_{K,L}(\tau)\| + \|J_{K,\text{res},2K}(\tau)\| \right). \end{aligned} \quad (115)$$

In the second equality, we use the fact that  $V_K(\tau)$  is a unitary. To bound  $\|J_{K,L}(\tau)\|$ , we have

$$\|J_{K,L}(\tau)\| \leq \sum_{s=K}^{2K} \|C_s\| \frac{x^{s+1}}{(s+1)!}. \quad (116)$$

To bound  $J_{K,\text{res},2K}(\tau)$  in Eq. (115), from Eq. (105) we have

$$\|J_{K,\text{res},2K}(\tau)\| \leq \int_0^x d\tau \frac{(x-\tau)^s}{s!} \|J_K^{(s+1)}(\tau)\|. \quad (117)$$

Finally, to bound  $\|M_K(\tau)\|$  in Eq. (115), we have

$$\begin{aligned} \|M_K(\tau)\| &= \left\| \int_0^{\tau} d\tau_1 V_K(\tau_1) J_K(\tau_1) \right\| \\ &\leq \int_0^{\tau} d\tau_1 \|J_K(\tau_1)\| = \int_0^{\tau} d\tau_1 \|J_{K,\text{res},K-1}(\tau_1)\| \\ &\leq \int_0^{\tau} d\tau_1 \int_0^{\tau_1} d\tau_2 \frac{(\tau_1 - \tau_2)^{(K-1)}}{(K-1)!} \|J_K^{(K)}(\tau_2)\|. \end{aligned} \quad (118)$$

In the first line, we use the definition of  $M_K(\tau) = V_K(\tau) - I$  and the recurrence formula in Eq. (94). In the second line,

we use the property that  $V_K(\tau)$  is a unitary. In the third line, we use the order condition in Proposition 5. In the final line, we apply the operator-valued Taylor-series expansion on  $J_K(\tau)$  and set the truncation order  $s_c = K - 1$ . ■

From Proposition 6 we can see that, to study the performance of the truncated LCU formula  $\tilde{V}_K^{(nc)}(x)$  in Eq. (110), we only need to study the property of the derivatives of  $J_K(x)$ , including  $\{C_s\}$ . In the following section, we are going to derive the explicit formula of  $\tilde{V}_K^{(nc)}(x)$ , taking the lattice Hamiltonian with first-order Trotter formulas as an example.

### B. Example: first-order lattice Hamiltonian

Now, we focus on the lattice Hamiltonians with the form  $H = A + B$  in Eq. (10). For the lattice Hamiltonian, the first-order Trotter formula is  $S_1(x) = e^{-ixB}e^{-ixA}$ . Here, we assume that the time-evolution of each two-qubit component  $e^{-ixH_{j,j+1}}$  is easy to be implement on the quantum computer.

To derive the explicit form of the LCU formula  $\tilde{V}_1^{(nc)}(x)$  in Eq. (110), we first derive  $J_1(x)$  defined in Eq. (95) and its derivatives  $C_s := J_1^{(s)}(x)$ . From Eqs. (92) and (95) we have

$$\begin{aligned} R_1(x) &= \frac{d}{dx} S_1(x)^\dagger - iHS_1(x)^\dagger = i[e^{ixA}, B]e^{ixB}, \\ J_1(x) &= S_1(x)R_1(x) = ie^{-ixB}e^{-ixA}[e^{ixA}, B]e^{ixB} \\ &= i(B - e^{-ixad_B}e^{-ixad_A}B). \end{aligned} \quad (119)$$

Applying the general Libniz formula to  $J_1(x)$  we have,

$$J_1^{(s)}(x) = (-i)^{s+1} \sum_{\substack{m_1, n_1; \\ m_1 + n_1 = s}} \binom{s}{m_1} e^{-ixad_B} \text{ad}_B^{m_1} e^{-ixad_A} \text{ad}_A^{n_1} B. \quad (120)$$

If we set the truncation of  $J_1^{(s)}(x)$  to be  $s_c$  and apply the following operator Taylor-series expansion formula,

$$Q(x) = \sum_{s=0}^{s_c} \frac{x^s}{s!} Q^{(s)}(0) + \int_0^x d\tau \frac{(x-\tau)^{s_c}}{s_c!} Q^{(s_c+1)}(\tau), \quad (121)$$

we can expand  $J_1(x)$  as

$$J_1(x) = \sum_{s=0}^{s_c} C_s \frac{x^s}{s!} + J_{1,\text{res}}(x), \quad (122)$$

where

$$C_s = J_1^{(s)}(0) = (-i)^{s+1} \sum_{\substack{m_1, n_1; \\ m_1 + n_1 = s}} \binom{s}{m_1} \text{ad}_B^{m_1} \text{ad}_A^{n_1} B, \quad (123)$$

$$J_{1,\text{res},s_c}(x) = \int_0^x d\tau \frac{(x-\tau)^{s_c}}{s_c!} J_1^{(s_c+1)}(\tau).$$

We can see that,  $C_s$  can be written as the summation of nested commutators with concise form  $\text{ad}_B^{m_1} \text{ad}_A^{n_1} B$ . Note that,  $(-i)^{s+1} \text{ad}_B^{m_1} \text{ad}_A^{n_1} B$  with  $m_1 + n_1 = s$  is always anti-Hermitian when  $A$  and  $B$  are Hermitian. On the other hand, these nested commutators are all with the nice property that their spectral norm and 1-norm is linear to the system size  $n$ . To be more specific, we have the following norm bound.

*Proposition 7.* Consider a lattice Hamiltonian  $H = A + B$  with the form in Eq. (10). Suppose the spectral norm and 1-norm of its components  $H_{j,j+1}$  are upper bounded by  $\Lambda$  and  $\Lambda_1$ . Then for the nested commutators appearing in Eq. (123), we have the following bound:

$$\|e^{-i\tau ad_B} \text{ad}_B^{m_1} e^{-i\tau ad_A} \text{ad}_A^{n_1} B\| \leq \frac{n}{2} 3^{m_1} 2^{n_1} 2^s \Lambda^{s+1}, \quad (124)$$

where  $m_1, n_1$  are non-negative integers satisfying  $m_1 + n_1 = s$ . As a result, we can bound the spectral norm of  $J_1^{(s)}$  as  $\|J_1^{(s)}(x)\| \leq \frac{n}{2} 10^s \Lambda^{s+1}$ . The 1-norm upper bound is to simply replace  $\Lambda$  by  $\Lambda_1$ .

*Proof.* We first focus on one Hamiltonian term  $H_{j,j+1}$  contained in  $B$  and bound the norm,

$$\|e^{-i\tau ad_B} \text{ad}_B^{m_1} e^{-i\tau ad_A} \text{ad}_A^{n_1} (H_{j,j+1})\| \leq 3^{m_1} 2^{n_1} 2^s \Lambda^{s+1}. \quad (125)$$

To do this, we are going to decompose commutator to the elementary nested commutators in the following form:

$$\begin{aligned} &e^{-i\tau ad_B} \text{ad}_{H_{j_s, j_{s+1}}} \cdots \text{ad}_{H_{j_{n_1+1}, j_{n_1+1+1}}} \\ &e^{-i\tau ad_A} \text{ad}_{H_{j_1, j_{n_1+1}}} \cdots \text{ad}_{H_{j_1, j_1+1}} H_{j, j+1}, \end{aligned} \quad (126)$$

where  $j_1, j_2, \dots, j_s$  are the possible vertices' indices. For each elementary nested commutator, the spectral norm can be easily bounded by  $(2\Lambda)^s \Lambda$  by simply expanding all the commutators and applying triangle inequality. Here, we use the property that the spectral norm of all the exponential operators with anti-Hermitian exponent is 1.

Now, we count the number of the possible elementary commutators with the form in Eq. (126). We will check the action of each adjoint operator  $\text{ad}_A$  or  $\text{ad}_B$  from the right to the left. For the first location, if we expand  $\text{ad}_A$ , there will be only two possible elementary nonzero components,  $\text{ad}_{H_{j-1, j}}$  and  $\text{ad}_{H_{j+1, j+2}}$ . If the next ad is still  $\text{ad}_A$ ,

the support will still be on the four qubits:  $j - 1, j, j + 1$ , and  $j + 2$ . As a result, there will still be only two possible components,  $\text{ad}_{H_{j-1,j}}$  and  $\text{ad}_{H_{j+1,j+2}}$ . Similarly, the exponential operator  $e^{-ix\text{ad}_A}$  will not enlarge the support since one can expand it to the power of  $\text{ad}_A$ . The support will be enlarged when  $\text{ad}_B$  comes. In this layer, the support of the operator will be expanded to six qubits, and there will be three elementary components. The number of possible elementary commutators is then

$$3^{m_1} 2^{n_1}. \quad (127)$$

Combining the number of elementary nested commutators and the norm bound for each commutator and applying triangle inequality, we will obtain Eq. (125). Finally, in the operator  $B$ , there are  $n/2$  possible summands. This finishes the proof of Eq. (124).

Now, we apply Eq. (124) to bound the norm of  $J_1^{(s)}(x)$ . From Eq. (120) we have

$$\begin{aligned} \|J_1^{(s)}(x)\| &\leq \sum_{\substack{m_1, n_1; \\ m_1+n_1=s}} \binom{s}{m_1} \|e^{-ix\text{ad}_B} \text{ad}_B^{m_1} e^{-ix\text{ad}_A} \text{ad}_A^{n_1} B\| \\ &\leq \frac{n}{2} 2^s \Lambda^{s+1} \sum_{\substack{m_1, n_1; \\ m_1+n_1=s}} \binom{s}{m_1} 3^{m_1} 2^{n_1} \\ &= \frac{n}{2} 10^s \Lambda^{s+1}. \end{aligned} \quad (128)$$

In the third line, we apply the binomial theorem.

Since 1-norm can be estimated based on the same logic by counting the number of elementary nested commutators and the 1-norm of each nest commutator, the derivation for the 1-norm is similar by replacing  $\Lambda$  to  $\Lambda_1$ . ■

As introduced in Sec. V A, when  $K = 1$ , we set  $F_{1,s}^{(nc)}(x)$  with  $s = 2, 3$  to be the leading-order terms and set the truncation order  $s_c = 3$ . That is, we only compensate the second- and third-order error using LCU methods. While we are not able to achieve the logarithmic accuracy similar to PTSC algorithms, we can achieve a high accuracy of  $\mathcal{O}(\varepsilon^{-1/3})$ , which is cubically improved comparing to the bare first-order Trotter result  $\mathcal{O}(\varepsilon^{-1})$ .

Based on Eq. (110), the truncated nested-commutator LCU formula for  $V_1(x)$  can be written as

$$\begin{aligned} \tilde{V}_1(x) &= I + \sum_{s=2}^3 F_{1,s}^{(nc)}(x), \\ &= \sqrt{1 + (\eta_\Sigma^{(nc)})^2} \sum_{s=2}^3 \frac{\eta_s^{(nc)}}{\eta_\Sigma^{(nc)}} R_{1,s}^{(nc)}(\eta_\Sigma). \end{aligned} \quad (129)$$

Here,  $\eta_\Sigma^{(nc)} := \sum_{s=2}^3 \|C_{s-1}\|_1 \frac{x^s}{s!}$ . The explicit form of  $R_{1,s}^{(nc)}(\eta_\Sigma)$  can be obtained by the definitions in Eqs. (109),

(111) and the Pauli operator decomposition based on the nested-commutator form in Eq. (123).

Combined with the deterministic first-order Trotter formula, the overall LCU formula for  $U(x)$  is

$$\tilde{U}_1^{(nc)}(x) = \tilde{V}_1^{(nc)}(x) S_1(x). \quad (130)$$

Following Propositions 6 and 7, we can bound the 1-norm and error of the LCU formula in Eqs. (129) and (130) as follows.

*Proposition 8 (First-order Trotter-LCU formula by nested-commutator compensation for lattice Hamiltonians).* Consider a lattice Hamiltonian with the form in Eq. (10). For  $\min\{1/4\Lambda, 3/20\Lambda_1\} > x > 0$ , Eq. (129) is a  $(\mu_1^{(nc)}(x), \varepsilon_1^{(nc)}(x))$ -LCU formula of  $V_1(x)$  with

$$\begin{aligned} \mu_1^{(nc)}(x) &\leq \exp(10n^2(\Lambda_1 x)^4), \\ \varepsilon_1^{(nc)}(x) &\leq 15n^2(\Lambda x)^4. \end{aligned} \quad (131)$$

As a result,  $\tilde{U}_1^{(nc)}(x)$  in Eq. (130) is a  $(\mu_1^{(nc)}(x), \varepsilon_1^{(nc)}(x))$ -LCU formula of  $U(x)$ . Here,  $\Lambda_1$  and  $\Lambda$  are defined in Eq. (11).

*Proof.* We start from bounding the 1-norm  $\mu_1^{(nc)}(x)$ . From Proposition 6 we have

$$\begin{aligned} \mu_1^{(nc)}(x) &\leq \sqrt{1 + \left( \sum_{s=1}^2 \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2} \\ &\leq 1 + \frac{1}{2} \left( \sum_{s=1}^2 \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2 \\ &= 1 + n^2 \left( \frac{25}{8} (\Lambda_1 x)^4 + \frac{250}{12} (\Lambda_1 x)^5 \right. \\ &\quad \left. + \frac{1250}{3!3!} (\Lambda_1 x)^6 \right) \\ &\leq 1 + 3n^2 \frac{25}{8} (\Lambda_1 x)^4 \leq \exp(10n^2(\Lambda_1 x)^4). \end{aligned} \quad (132)$$

In the third line, we use the fact that  $C_s = J_1^{(s)}(0)$  and Proposition 7. In the fourth line, we use the assumption that  $\Lambda_1 x \leq 3/20$ .

Now, we bound the spectral norm distance  $\varepsilon_1^{(nc)}(x) = \|F_{1,\text{res}}^{(nc)}\|$ . From Proposition 6 we know that we only need to bound  $\|J_{1,L}(\tau)\|$ ,  $\|J_{1,\text{res},2}(\tau)\|$ , and  $M_1(\tau)$  based on

Eq. (114). For  $\|J_{1,L}(\tau)\|$ , from Proposition 7 we have

$$\|J_{1,L}(\tau)\| \leq \sum_{s=1}^2 \|C_s\| \frac{\tau^s}{s!} \leq \frac{n}{2} \sum_{s=1}^2 10^s \Lambda^{s+1} \frac{\tau^s}{s!}. \quad (133)$$

For  $\|J_{1,\text{res},2}(\tau)\|$  and  $\|M_1(\tau)\|$ , from Eq. (114) and Proposition 7 we have

$$\begin{aligned} \|J_{1,\text{res},2}(\tau)\| &\leq \int_0^\tau d\tau_1 \frac{(\tau - \tau_1)^2}{2!} \|J_1^{(3)}(\tau_1)\| \\ &\leq \frac{n}{2} 10^3 \Lambda^4 \frac{\tau^3}{3!}, \\ \|M_1(\tau)\| &\leq \int_0^\tau d\tau_1 \int_0^{\tau_1} d\tau_2 \|J_1^{(1)}(\tau_2)\| \\ &\leq \frac{n}{2} 10 \Lambda^2 \frac{\tau^2}{2!}. \end{aligned} \quad (134)$$

Based on Eqs. (133), (134) and Proposition 6,

$$\begin{aligned} \|F_{1,\text{res}}^{(nc)}\| &\leq \int_0^x d\tau (\|M_1(\tau)\| \|J_{1,L}(\tau)\| + \|J_{1,\text{res},2}(\tau)\|) \\ &\leq \int_0^x d\tau \left( \frac{n^2}{4} \sum_{s=1}^2 10^{s+1} \Lambda^{s+3} \frac{\tau^{s+2}}{s!2!} + \frac{n}{2} 10^3 \Lambda^4 \frac{\tau^3}{3!} \right) \\ &= \frac{n^2}{4} \sum_{s=1}^2 10^{s+1} \frac{(\Lambda x)^{s+2}}{s!2!(s+3)} + \frac{n}{2} 10^3 \frac{(\Lambda x)^4}{4!} \\ &\leq 2 \cdot \frac{n^2}{4} 10^2 \frac{(\Lambda x)^4}{8} + \frac{n}{2} 10^3 \frac{(\Lambda x)^4}{4!} \\ &\leq 15n^2 (\Lambda x)^4. \end{aligned} \quad (135)$$

In the fourth line, we use the assumption that  $x \leq 1/4\Lambda$ .  $\blacksquare$

### C. General construction and performance

We can easily extend the first-order analysis above to the higher-order case. In Appendix B, we provide an explicit construction for second-order nested-commutator expansion, which will be used for the later numerical results. For the general LCU formula for the  $K$ th-order Trotter remainder, we have the following proposition to characterize the LCU formulas  $\tilde{V}_K(x)$  in Eq. (110).

*Proposition 9 (Trotter-LCU formula by nested-commutator compensation for lattice Hamiltonians).* Consider a lattice Hamiltonian with the form in Eq. (10). We set  $\beta := 2(4\kappa + 5)$  where  $\kappa$  is the stage number of the  $K$ th-order Trotter formula ( $K = 1$  or  $2k$ ). For  $\min\{(K+1)/\beta\Lambda, (K+2)/\beta\Lambda_1\} > x > 0$ ,  $\tilde{V}_K^{(nc)}(x)$  in

Eq. (110) is a  $(\mu_K^{(nc)}(x), \varepsilon_K^{(nc)}(x))$ -LCU formula of  $V_K(x)$  with

$$\begin{aligned} \mu_K^{(nc)} &\leq \exp\left(\frac{n^2 \kappa^2 \beta^{2(K+1)}}{2(K!)^2} (\Lambda_1 x)^{2K+2}\right), \\ \varepsilon_K^{(nc)} &\leq (n\kappa)^2 \frac{\beta^{2K+1}}{K!(K+1)!} (\Lambda x)^{2K+2}. \end{aligned} \quad (136)$$

As a result,  $\tilde{U}_K^{(nc)}(x)$  in Eq. (110) is a  $(\mu_K^{(nc)}(x), \varepsilon_K^{(nc)}(x))$ -LCU formula of  $U(x)$ . Here,  $\Lambda$  and  $\Lambda_1$  are, respectively, the largest spectral norm and 1-norm of the lattice Hamiltonian components defined in Eq. (11).

The proof of Proposition 9 is in Appendix C.

The circuit of random-sampling implementation of the NCC algorithm is similar to the PTSC algorithm, which is illustrated in Fig. 7. The only difference is that we sample the Pauli operators based on the nested-commutator expansion formula. In Appendix D, we generalize our random-sampling algorithm to the  $K$ th-order situation to demonstrate its scalability. Specifically, the space and time cost of the sampling algorithm are  $\mathcal{O}(K\kappa_K)$  and  $\mathcal{O}(K(\log \kappa_K + \log n))$ , respectively. In practice, we need to expand the leading-order terms  $\{F_{K,s}\}$  to a summation of different adjoint operators based on the methods in Sec. VA first, and then calculate the corresponding sampling probability  $\Pr(\gamma|s)$ . We have the following theorem to characterize the gate complexity of the  $K$ th-order NCC algorithm with random-sampling implementation.

*Theorem 4 (Gate complexity of the  $K$ th-order random-sampling Trotter-LCU algorithm by nested-commutator compensation for lattice Hamiltonians).* In a  $K$ th-order Trotter-LCU algorithm ( $K = 1$  or  $2k$ ) based on nested-commutator compensation, if the segment number  $\nu$  satisfy all the requirements below,

$$\begin{aligned} \nu &\geq \max\left\{\frac{\beta\Lambda}{K+1}t, \frac{\beta\Lambda_1}{K+2}t\right\}, \\ \nu &\geq \left(\frac{\kappa^2}{2 \ln \mu(K!)^2}\right)^{\frac{1}{2K+1}} \beta^{1+\frac{1}{2K+1}} n^{\frac{2}{2K+1}} (\Lambda_1 t)^{1+\frac{1}{2K+1}}, \\ \nu &\geq \left(\frac{\mu\kappa^2}{K!(K+1)!}\right)^{\frac{1}{2K+1}} \beta \varepsilon^{-\frac{1}{2K+1}} n^{\frac{2}{2K+1}} (\Lambda t)^{1+\frac{1}{2K+1}}, \end{aligned} \quad (137)$$

we can then realize a  $(\mu, \varepsilon)$ -LCU formula for  $U(t)$  based on  $\nu$  segments of  $\tilde{U}_K^{(nc)}(x)$  in Eq. (110). As a result, the gate complexity of random-sampling  $K$ th-order Trotter-LCU algorithm based on nested commutator compensation

for the lattice Hamiltonian is

$$\mathcal{O}\left(n^{1+\frac{2}{2K+1}}t^{1+\frac{1}{2K+1}}\varepsilon^{-\frac{1}{2K+1}}\right). \quad (138)$$

Here,  $\beta := 2(4\kappa + 5)$  where  $\kappa$  is the stage number of the Trotter formula.  $\Lambda$  and  $\Lambda_1$  are, respectively, the largest spectral norm and 1-norm of the lattice Hamiltonian components defined in Eq. (11).

*Proof.* For the random-sampling implementation, the overall LCU formula for  $U(t)$  is to repeat the sampling of  $\tilde{U}_K^{(nc)}(x)$  for  $\nu$  times,  $\tilde{U}_{K,\text{tot}}^{(nc)}(t) = \tilde{U}_K^{(nc)}(x)^\nu$ . Using Propositions 2 and 9, when  $0 < x < \min\{(K+1)/\beta\Lambda, (K+2)/\beta\Lambda_1\}$ , we conclude that  $\tilde{U}_{K,\text{tot}}^{(nc)}(t)$  is a  $(\mu_{K,\text{tot}}^{(nc)}(t), \varepsilon_{K,\text{tot}}^{(nc)}(t))$ -LCU formula of  $U(t)$  with

$$\begin{aligned} \mu_{K,\text{tot}}^{(nc)}(t) &= \mu_K(x)^\nu \leq \exp\left(\frac{n^2\kappa^2\beta^{2(K+1)}(\Lambda_1 t)^{2K+2}}{2(K!)^2\nu^{2K+1}}\right), \\ \varepsilon_{K,\text{tot}}^{(nc)}(t) &\leq \nu\mu_{K,\text{tot}}^{(nc)}(t)\varepsilon_K^{(nc)}(x) \\ &\leq \nu\mu_{K,\text{tot}}^{(nc)}(t)\left((n\kappa)^2\frac{\beta^{2K+1}}{K!(K+1)!}\frac{(\Lambda t)^{2K+2}}{\nu^{2K+2}}\right). \end{aligned} \quad (139)$$

To realize a  $(\mu, \varepsilon)$ -LCU formula for  $U(t)$ , we only need the segment number  $\nu$  satisfy all the requirements in Eq. (137). It suffices to choose

$$\nu = \mathcal{O}\left(n^{\frac{2}{2K+1}}t^{1+\frac{1}{2K+1}}\varepsilon^{-\frac{1}{2K+1}}\right). \quad (140)$$

Based on Eq. (86), the gate complexity of the  $K$ th-order Trotter-LCU algorithm is then

$$\mathcal{O}(\nu(\kappa_K L + s_c)) = \mathcal{O}\left(n^{1+\frac{2}{2K+1}}t^{1+\frac{1}{2K+1}}\varepsilon^{-\frac{1}{2K+1}}\right). \quad (141)$$

Here we use the fact that  $s_c = 2K + 1 = \mathcal{O}(1)$  and  $L = \mathcal{O}(n)$  for lattice Hamiltonians. ■

So far, we have restricted our construction of the nested-commutator expansion to lattice Hamiltonians. In practice, various physical Hamiltonians, including those for the electronic structure of quantum materials [42], quantum chemistry Hamiltonians [43–45], and power-law interaction Hamiltonians [20], also possess sparse properties. As a result, the methods for the nested-commutator expansion of the Trotter remainder  $V_K(x)$  introduced in Sec. V A can also be applied to a general Hamiltonian  $H$ .

In Appendix E, we discuss how to perform the nested-commutator expansion of  $V_K(x)$  for general Hamiltonians, and discuss the performance of the resulting LCU formula in Proposition 13. We find that the 1-norm of the LCU formula based on nested-commutator expansion is closely

related to the following nested commutator norm of the Hamiltonian  $H = \sum_{l=1}^L H_l$ :

$$\tilde{\alpha}_{\text{com}}(H) := \sum_{l_{s+1}=1}^L \cdots \sum_{l_2=1}^L \|[H_{l_{s+1}}, \dots [H_{l_2}, H_1]] \cdots\|. \quad (142)$$

$\tilde{\alpha}_{\text{com}}(H)$  is originally defined in Ref. [20] to analyze the performance of Trotter methods. In Ref. [20], the authors estimate the values of  $\alpha_H^{(s)}$  for typical Hamiltonian models like plane-wave-basis quantum chemistry models,  $k$ -local Hamiltonian, and Hamiltonians with power-law interactions. Following similar estimation methods, we can also calculate  $\tilde{\alpha}_{\text{com}}(H)$  for different models and consider their explicit nested-commutator expansions. We will leave the explicit evaluation of other typical Hamiltonians for a future work.

## VI. CONCLUSION AND OUTLOOK

We study the Hamiltonian simulation algorithms based on the composition of Trotter and LCU algorithms. In both theoretical and numerical studies, we show that the 0th-order PTSC algorithm,  $2k$ th-order PTSC algorithm and the  $2k$ th-order NCC algorithm enjoy different advantages and will be useful in different scenarios. Taking the  $n$ -qubit lattice Hamiltonian as an example: the 0th-order PTSC algorithm performs the best when  $t$  is small compared with  $n$  and  $1/\varepsilon$ ; the  $2k$ th-order PTSC algorithm performs the best when  $n$  is small compared with  $t$  and  $1/\varepsilon$ ; while the  $2k$ th-order NC algorithm performs the best when  $1/\varepsilon$  is small compared with  $n$  and  $t$ . In practice, with finite system size  $n$ , simulation time  $t$  and inverse accuracy  $1/\varepsilon$ , we can think about a hybrid implementation of different algorithms. For example, when the sparsity  $L$  of a given Hamiltonian is large, we can first split the Hamiltonian to two parts,

$$H = H_1 + H_2 = \sum_{j=1}^{L_1} H_j + \sum_{k=L_1+1}^L H_k, \quad (143)$$

where the summands in  $H_1$  are the few dominant terms with large coefficients. We can then perform second-order Trotter only for  $H_1$ , and use PTS to expand the remainder

$$V_2(x; H_1) = \prod_{j_1=L_1}^1 e^{iH_{j_1}x} e^{-iHx} \prod_{j_2=1}^{L_1} e^{iH_{j_2}x}. \quad (144)$$

If the number of dominant terms  $L_1$  is small, we can then reduce the  $L$  dependence of the algorithm similar to 0th-order PTSC algorithm while keep the good  $t$  dependence of second-order PTSC algorithm. As another example, we can hybridize second-order PTSC and NCC algorithms:

we apply the nested-commutator compensation for the leading-order terms (i.e., the terms with  $s = 3, 4$ , and  $5$ ), and normal Taylor-series compensation for higher-order terms. In this case, we can find an optimal truncation location  $s_c$ , which fulfills the high simulation accuracy requirement and keeps the nested-commutator scaling for the leading-order compensation terms.

The design of Trotter-LCU algorithms is based on a series connection of Trotter and LCU algorithms. A similar composition method can also be exploited for other Hamiltonian simulation algorithms [46]. For example, we may replace the deterministic Trotter with the ones with random permutation [12]. Recently, Cho *et al.* [47] consider similar idea to compensate the Trotter error using randomized unitary operators. Using anticommutative cancellation [48], we can further reduce the compensation terms.

### ACKNOWLEDGMENTS

We thank Xiaoming Zhang, Xiao Yuan, Min-Hsiu Hsieh, Yuan Su, Kaiwen Gui, Ming Yuan, Senrui Chen, and Ying Li for helpful discussion and suggestions. We would like to especially thank Wenjun Yu for highlighting the validity of the variant where the ancillary qubit is measured and reset for each segment. P.Z. and L.J. acknowledge support from the ARO MURI (W911NF-21-1-0325), AFOSR MURI (FA9550-19-1-0399, FA9550-21-1-0209), AFRL (FA8649-21-P-0781), NSF (OMA-1936118, ERC-1941583, OMA-2137642), NTT Research, and the Packard Foundation (2020-71479). J.S. would like to thank support from the Innovate UK (Project No.10075020) and support through Schmidt Sciences, LLC. Q.Z. acknowledges funding from National Natural Science Foundation of China (NSFC) via Project No. 12347104 and No. 12305030, Guangdong Natural Science Fund via Project 2023A1515012185, Hong Kong Research Grant Council (RGC) via No. 27300823, N\_HKU718/23, and R6010-23, Guangdong Provincial Quantum Science Strategic Initiative No. GDZX2303007, HKU Seed Fund for Basic Research for New Staff via Project 2201100596.

### APPENDIX A: PAIRED TAYLOR-SERIES COMPENSATION WITH HIGHER-ORDER TROTTER FORMULAS

Following the same idea in Sec. IV B, we now generalize it to the case with  $2k$ th-order Trotter formula. Expanding the  $2k$ th-order Trotter remainder, we have

$$V_{2k}(x) = \sum_{s=0}^{\infty} F_{2k,s}(x) = \sum_{s=0}^{\infty} \eta_s V_{2k,s}, \quad (\text{A1})$$

where

$$\begin{aligned} F_{2k,s}(x) &= \eta_s \sum_{r,\gamma} \Pr(r, \gamma | s) P_{2k}(r, \gamma), \\ \Pr(r, \gamma | s) &:= \Pr(r, \vec{r}_{1:k}, \vec{r}'_{1:k} | s) \sum_{l_{1:r}} P_{l_{1:r}}^{(r)}, \\ P_{2k}(r, \gamma) &:= (-i)^{2r-s} \overleftarrow{\prod} \vec{P}^{\vec{r}_k} \dots \overleftarrow{\prod} \vec{P}^{\vec{r}'_1} \left( P_{l_{1:r}}^{(r)} \right) \\ &\quad \overrightarrow{\prod} \vec{P}^{\vec{r}'_1} \dots \overrightarrow{\prod} \vec{P}^{\vec{r}_k}. \end{aligned} \quad (\text{A2})$$

Here, we use  $\gamma$  to denote all the expansion variables  $\{\vec{r}_{1:k}, \vec{r}'_{1:k}, l_{1:r}\}$  besides  $r$ .

We ignore the derivation and provide the general form of the LCU formula for  $\tilde{V}_{2k}^{(p)}(x)$ ,

$$\begin{aligned} \tilde{V}_{2k}^{(p)}(x) &= \mu_{2k}^{(p)}(x) \left( \sum_{s=2k+1}^{4k+1} \Pr_{2k}^{(p)}(s) R_{2k,s}^{(p)}(\eta_\Sigma) \right. \\ &\quad \left. + \sum_{s=4k+2}^{s_c} \Pr_{2k}^{(p)}(s) V_{2k,s}^{(p)} \right), \end{aligned} \quad (\text{A3})$$

where

$$\begin{aligned} \mu_{2k}^{(p)}(x) &= \sqrt{1 + (\eta_\Sigma)^2} + \sum_{s=4k+2}^{s_c} \eta_s, \quad \eta_\Sigma = \sum_{s=2k+1}^{4k+1} \eta_s, \\ \Pr^{(p)}(s) &= \frac{1}{\mu_{2k}^{(p)}(x)} \begin{cases} \sqrt{1 + (\eta_\Sigma)^2} \frac{\eta_s}{\eta_\Sigma}, & s = 2k+1, 2k+2, \dots, 4k+1, \\ \eta_s, & s = 4k+2, 4k+3, \dots, s_c. \end{cases} \end{aligned} \quad (\text{A4})$$

The Pauli rotation unitary  $R_{2k,2q+1}^{(p)}(y) := \exp(i\theta(y)P'_{2k}(r, \gamma)')$  where  $\theta(y) := \tan^{-1}(1 + y^2)$  and  $P'_{2k}(r, \gamma) := (-i)^{\mathbb{1}[P_{2k,s}(r,\gamma):\text{anti-Her}]} P_{2k}(r, \gamma)$ .

Combined with the deterministic Trotter formula, the overall LCU formula for  $U(x)$  is

$$\tilde{U}_{2k}^{(p)}(x) = \tilde{V}_{2k}^{(p)}(x) S_{2k}(x). \quad (\text{A5})$$

The following proposition gives the performance characterization of  $\tilde{U}_{2k}^{(p)}(x)$  to approximate  $U(x)$ .

*Proposition 10 (2kth-order Trotter-LCU formula by paired Taylor-series compensation).* For  $0 < x < 1/(2\lambda)$

and  $s_c \geq 4k + 1$ ,  $\tilde{V}_{2k}^{(p)}(x)$  in Eq. (A3) is a  $(\mu_{2k}^{(p)}(x), \varepsilon_{2k}^{(p)}(x))$ -LCU formula of  $V_{2k}(x)$  with

$$\begin{aligned} \mu_{2k}^{(p)}(x) &\leq e^{(e+c_k)(2\lambda x)^{4k+2}}, \\ \varepsilon_{2k}^{(p)}(x) &\leq \left(\frac{2e\lambda x}{s_c+1}\right)^{s_c+1}. \end{aligned} \quad (\text{A6})$$

Here,

$$c_k := \frac{1}{2} \left(\frac{e}{2k+1}\right)^{4k+2}, \quad (\text{A7})$$

so that  $0.3 > c_k > 0$ . As a result,  $\tilde{U}_{2k}^{(p)}(x)$  in Eq. (A5) is a  $(\mu_{2k}^{(p)}(x), \varepsilon_{2k}^{(p)}(x))$ -LCU formula of  $U(x)$ .

*Proof.* We will focus on the case with  $k = 1$ . The proof for a general  $k$  is similar. We first bound the normalization factor  $\mu_2^{(p)}(x)$ . When  $2\lambda x < 1$  we have

$$\begin{aligned} \mu_2^{(p)}(x) &\leq \sqrt{1 + (\eta_\Sigma)^2} + \sum_{s=6}^{\infty} \eta_s \\ &\leq 1 + \frac{1}{2}\eta_\Sigma^2 + \left(e^{2\lambda x} - \sum_{s=0}^5 \eta_s\right) \\ &= \frac{1}{2}(2\lambda x)^6 \left(\frac{1}{3!} + \frac{2\lambda x}{4!} + \frac{(2\lambda x)^2}{5!}\right)^2 + \left(e^{2\lambda x} - \sum_{s=1}^5 \eta_s\right) \\ &\leq \frac{1}{2}(2\lambda x)^6 \left(\sum_{s=3}^{\infty} \frac{1}{3!}\right)^2 + \left(e^{2\lambda x} - \sum_{s=1}^5 \eta_s\right) \\ &\leq \frac{1}{2}\left(\frac{e}{3}\right)^6 (2\lambda x)^6 + e^{e(2\lambda x)^6} \leq e^{(e+c_1)(2\lambda x)^6}. \end{aligned} \quad (\text{A8})$$

In the fifth line, we use Corollary 1.

The distance bound can be derived in the same manner as the one in Proposition 4.  $\blacksquare$

## APPENDIX B: TIGHT NESTED-COMMUTATOR ANALYSIS FOR SECOND-ORDER TROTTER-LCU ALGORITHM

We can easily extend the methods for the first-order analysis in Sec. VB to higher order case. Taking the second-order case as an example, the second-order Trotter formula of the lattice Hamiltonian is  $S_2(x) = e^{-i\frac{x}{2}A} e^{-ixB} e^{-i\frac{x}{2}A}$ .

To derive the explicit LCU formula  $\tilde{V}_2^{(nc)}(x)$  in Eq. (110), we first derive  $J_2(x)$  defined in Eq. (95) and its derivatives

$C_s := J_2^{(s)}(x)$ . From Eqs. (92) and (95) we have

$$\begin{aligned} S_2(x)^\dagger &= e^{ixH} + \int_0^x d\tau e^{i(x-\tau)H} R_2(\tau), \\ R_2(\tau) &= i[e^{i\frac{x}{2}A}, B]e^{ixB} e^{i\frac{x}{2}A} + \frac{i}{2}[e^{i\frac{x}{2}A} e^{ixB}, A]e^{i\frac{x}{2}A}, \\ J_2(x) &= S_2(x)R_2(x) = \frac{i}{2} \left( A - e^{-i\frac{x}{2}ad_A} e^{-ixad_B} A \right) \\ &\quad + i \left( e^{-i\frac{x}{2}ad_A} B - e^{-i\frac{x}{2}ad_A} e^{-ixad_B} e^{-i\frac{x}{2}ad_A} B \right). \end{aligned} \quad (\text{B1})$$

Following the approach in Sec. VA, we expand  $V_2(x)$  and  $J_2(x)$  by

$$V_2(x) = \sum_{s=0}^{\infty} G_s \frac{x^s}{s!}, \quad J_2(x) = \sum_{s=0}^{\infty} C_s \frac{x^s}{s!}, \quad (\text{B2})$$

then based on Proposition 5 and the recurrence formula Eq. (94) we have,

$$\begin{aligned} G_1 = G_2 = 0, \quad C_0 = C_1 = 0, \\ G_s = C_{s-1}, \quad s = 3, 4, 5. \end{aligned} \quad (\text{B3})$$

Combining Eqs. (B1) and (B3), we can show that

$$G_s = \mathcal{O}(n^{\lfloor \frac{s}{3} \rfloor}). \quad (\text{B4})$$

Therefore, the first three nontrivial terms  $G_3, G_4, G_5 = \mathcal{O}(n)$ . These terms will be set as the leading-order terms.

Based on Eq. (B3), we are going to expand  $J_K(x)$  based on the operator-valued Taylor-series expansion with integral remainders,

$$J_2(x) = J_{2,L}(x) + J_{2,\text{res},4}(x) = \sum_{s=2}^4 C_s \frac{x^s}{s!} + J_{2,\text{res},4}(x), \quad (\text{B5})$$

where

$$\begin{aligned} C_s &= J_2^{(s)}(0), \quad s = 2, 3, 4, \\ J_{2,\text{res},s}(x) &= \int_0^x d\tau \frac{(x-\tau)^s}{s!} J_2^{(s+1)}(\tau). \end{aligned} \quad (\text{B6})$$

$J_{2,L}(x) := \sum_{s=2}^4 C_s \frac{x^s}{s!}$  denotes the leading-order terms in  $J_K(x)$ . Then, from Eqs. (94) and (B3), the second-order

Trotter remainder can be expressed as

$$\begin{aligned} V_2(x) &= I + \sum_{s=3}^5 F_{2,s}^{(nc)}(x) + F_{2,\text{res}}^{(nc)}(x), \\ F_{2,s}^{(nc)}(x) &= C_{s-1} \frac{x^s}{s!}, \quad 3, 4, 5, \\ F_{2,\text{res}}^{(nc)}(x) &= \int_0^x d\tau (M_2(\tau)J_{2,L}(\tau) + V_2(\tau)J_{2,\text{res},4}(\tau)). \end{aligned} \quad (\text{B7})$$

We put the explicit nested-commutator expressions of the leading-order terms  $F_{2,s}^{(nc)}(x)$  ( $s = 3, 4$ , or  $5$ ) in Sec. B.

In practice, we truncate the formula with the order  $s_c = 5$ . Based on Eq. (110), the truncated nested-commutator LCU formula for  $V_2(x)$  can be written as

$$\begin{aligned} \tilde{V}_2(x) &= I + \sum_{s=3}^5 F_{2,s}^{(nc)}(x), \\ &= \sqrt{1 + (\eta_\Sigma^{(nc)})^2} \sum_{s=3}^5 \frac{\eta_s^{(nc)}}{\eta_\Sigma^{(nc)}} R_{2,s}^{(nc)}(\eta_\Sigma). \end{aligned} \quad (\text{B8})$$

Here,  $\eta_\Sigma^{(nc)} := \sum_{s=3}^5 \|C_{s-1}\|_1 \frac{x^s}{s!}$ . The explicit form of  $R_{2,s}^{(nc)}(\eta_\Sigma)$  can be obtained by the definitions in Eqs. (109), (111) and the Pauli operator decomposition based on the nested-commutator form in Eq. (B12).

For a tight numerical estimation, we seek for a tighter bound of the 1-norm  $\mu_2^{(nc)}(x)$  and spectral norm accuracy  $\varepsilon_2^{(nc)}(x)$  for the LCU formula in Eq. (B8).

### 1. Bound the 1-norm of recurrence function

Hereafter, we define  $A' := -\frac{i}{2}A$  and  $B' := -iB$  to simplify the notation. We have

$$\begin{aligned} J_2(x) &= J_a(x) + J_b(x), \\ J_a(x) &= (e^{\text{rad}_{A'}} e^{\text{rad}_{B'}} A' - A'), \\ J_b(x) &= (e^{\text{rad}_{A'}} e^{\text{rad}_{B'}} e^{\text{rad}_{A'}} B' - e^{\text{rad}_{A'}} B'). \end{aligned} \quad (\text{B9})$$

Apply the Libniz rule to  $J_a(x)$  and  $J_b(x)$ , we have

$$\begin{aligned} J_a^{(s)}(x) &= \sum_{l=0}^s \binom{s}{l} e^{\text{rad}_{A'}} \text{ad}_{A'}^l e^{\text{rad}_{B'}} \text{ad}_{B'}^{s-l} A', \\ J_b^{(s)}(x) &= \sum_{\substack{l_1, l_2, l_3 \\ l_1+l_2+l_3=s, l_2+l_3 \neq 0}} \binom{s}{l_1, l_2, l_3} \\ &\quad e^{\text{rad}_{A'}} \text{ad}_{A'}^{l_1} e^{\text{rad}_{B'}} \text{ad}_{B'}^{l_2} e^{\text{rad}_{A'}} \text{ad}_{A'}^{l_3} B'. \end{aligned} \quad (\text{B10})$$

Then,

$$\begin{aligned} C_s &= \sum_{l=0}^{s-1} \text{ad}_{A'}^l \text{ad}_{B'}^{s-l} A \\ &+ \sum_{\substack{l_1, l_2, l_3 \\ l_1+l_2+l_3=s, l_2+l_3 \neq 0}} \binom{s}{l_1, l_2, l_3} \text{ad}_{A'}^{l_1} \text{ad}_{B'}^{l_2} \text{ad}_{A'}^{l_3} B. \end{aligned} \quad (\text{B11})$$

We can solve the explicit form of the leading-order terms with  $s = 2, 3$ , and  $4$ ,

$$\begin{aligned} C_2 &= (\text{ad}_{B'}^2 A' + 2\text{ad}_{A'} \text{ad}_{B'} A') + (3\text{ad}_{A'}^2 B' + 2\text{ad}_{B'} \text{ad}_{A'} B'), \\ C_3 &= (\text{ad}_{B'}^3 A' + 3\text{ad}_{A'} \text{ad}_{B'}^2 A' + 3\text{ad}_{A'}^2 \text{ad}_{B'} A') + (7\text{ad}_{A'}^3 B' \\ &\quad + 3\text{ad}_{B'} \text{ad}_{A'}^2 B' + 6\text{ad}_{A'} \text{ad}_{B'} \text{ad}_{A'} B' + 3\text{ad}_{B'}^2 \text{ad}_{A'} B'), \\ C_4 &= (\text{ad}_{B'}^4 A' + 4\text{ad}_{A'} \text{ad}_{B'}^3 A' + 6\text{ad}_{A'}^2 \text{ad}_{B'}^2 A' + 4\text{ad}_{A'}^3 \text{ad}_{B'} A') \\ &\quad + (15\text{ad}_{A'}^4 B' + 4\text{ad}_{B'} \text{ad}_{A'}^3 B' + 12\text{ad}_{A'}^2 \text{ad}_{B'} \text{ad}_{A'} B' \\ &\quad + 12\text{ad}_{A'} \text{ad}_{B'} \text{ad}_{A'}^2 B' + 6\text{ad}_{B'}^2 \text{ad}_{A'}^2 B' \\ &\quad + 12\text{ad}_{A'} \text{ad}_{B'}^2 \text{ad}_{A'} B' + 4\text{ad}_{B'}^3 \text{ad}_{A'} B'). \end{aligned} \quad (\text{B12})$$

We can then expand the operators  $A'$  and  $B'$  to Pauli operators and solve the 1-norm of  $C_2, C_3$ , and  $C_4$  under the Pauli decomposition based on Eq. (B12). The 1-norm of  $\tilde{V}_2^{(nc)}(x)$  is given by

$$\tilde{\mu}_2^{(nc)}(x) = \sqrt{1 + \left( \sum_{s=2}^4 \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2}. \quad (\text{B13})$$

We summarize the procedure to estimate the segment number  $\nu$  of second-order Trotter-LCU under nested-commutator compensation as follows.

- (1) For a specific lattice Hamiltonian, get the explicit Pauli expansion form of  $C_2, C_3, C_4$  by Eq. (B12).
- (2) Get the expression of the 1-norm  $\tilde{\mu}^{(nc)}(x)$  by Eq. (B13). Calculate  $x_U$  based on  $\tilde{\mu}^{(nc)}(x) = 2$ .
- (3) Solve the following residue operator:

$$V_{2,\text{res}}(x) = V_2(x) - \tilde{V}_2(x) = U(x)S_2(x)^\dagger, \quad (\text{B14})$$

calculate its spectral norm  $\|V_{2,\text{res}}(x)\|$  numerically. Check whether either the following requirement are satisfied:

$$\|V_{2,\text{res}}(x)\| \frac{t}{x} \leq \varepsilon, \quad (\text{B15})$$

where  $\varepsilon$  is a preset accuracy requirement. If not, we search the largest  $x$  value in the region  $0 < x < x_U$  by dichotomy which makes Eq. (B15) satisfied.

- (4) The number of segments is given by  $\nu := t/x$ . We then calculate the number of gates based on methods in Appendix B.2.

We have a tighter count bound for the norm of  $J_a^{(s)}(x)$  and  $J_b^{(s)}(x)$ ,

*Proposition 11.* Consider a lattice Hamiltonian  $H = A + B$  with the form in Eq. (10). Suppose the spectral norm and 1-norm of its components  $H_{j,j+1}$  are bounded by  $\Lambda$  and  $\Lambda_1$ . Then for the recurrence function  $J_2(x)$ , we have the following norm bound for its derivatives:

$$\begin{aligned} \|J_2^{(s)}(x)\| &\leq \frac{n}{2}(7^s + 12^s)\Lambda^{s+1}, \\ \|J_2^{(s)}(x)\|_1 &\leq \frac{n}{2}(7^s + 12^s)\Lambda_1^{s+1}. \end{aligned} \quad (\text{B16})$$

*Proof.* From Eq. (B10) we have,

$$\begin{aligned} \|J_a^{(s)}(x)\| &\leq \sum_{l=0}^s \binom{s}{l} \|e^{\text{xad}_{A'}} \text{ad}_{A'}^l e^{\text{xad}_{B'}} \text{ad}_{B'}^{s-l} A'\|, \\ \|J_b^{(s)}(x)\| &\leq \sum_{\substack{l_1, l_2, l_3 \\ l_1+l_2+l_3=s, l_2+l_3 \neq 0}} \binom{s}{l_1, l_2, l_3} \\ &\quad \|e^{\text{xad}_{A'}} \text{ad}_{A'}^{l_1} e^{\text{xad}_{B'}} \text{ad}_{B'}^{l_2} e^{\text{xad}_{A'}} \text{ad}_{A'}^{l_3} B'\|. \end{aligned} \quad (\text{B17})$$

To bound the norm of nested commutators, we use the same methods in Proposition 7. We have

$$\begin{aligned} &\|e^{\text{xad}_{A'}} \text{ad}_{A'}^l e^{\text{xad}_{B'}} \text{ad}_{B'}^{s-l} A\| \\ &\leq (3^l 2^{s-l})(\Lambda^l (2\Lambda)^{s-l}) \cdot \left(\frac{n}{2}\Lambda\right), \\ &\|e^{\text{xad}_{A'}} \text{ad}_{A'}^{l_1} e^{\text{xad}_{B'}} \text{ad}_{B'}^{l_2} e^{\text{xad}_{A'}} \text{ad}_{A'}^{l_3} B\| \\ &\leq (4^{l_1} 3^{l_2} 2^{l_3})(\Lambda^{l_1} (2\Lambda)^{l_2} \Lambda^{l_3}) \cdot \left(\frac{n}{2}\Lambda\right). \end{aligned} \quad (\text{B18})$$

Here, the first bracket of each bound corresponds to possible nest commutators, while the second bracket of each bound indicates the norm enlargement of each nested commutator.

Combining Eqs. (B17) and (B18), we have

$$\begin{aligned} \|J_a^{(s)}(x)\| &\leq \sum_{l=0}^s \binom{s}{l} (3\Lambda)^l (2 \cdot 2\Lambda)^{s-l} \cdot \left(\frac{n}{2}\Lambda\right) \\ &\leq (7\Lambda)^s \left(\frac{n}{2}\Lambda\right), \end{aligned}$$

$$\begin{aligned} \|J_b^{(s)}(x)\| &\leq \sum_{\substack{l_1, l_2, l_3 \\ l_1+l_2+l_3=s, l_2+l_3 \neq 0}} \binom{s}{l_1, l_2, l_3} \\ &\quad (4\Lambda)^{l_1} (3 \cdot 2\Lambda)^{l_2} (2\Lambda)^{l_3} \cdot \left(\frac{n}{2}\Lambda\right) \\ &\leq (12\Lambda)^s \left(\frac{n}{2}\Lambda\right). \end{aligned} \quad (\text{B19})$$

The 1-norm bound can be derived in the same way.  $\blacksquare$

We can derive the bounds for  $\tilde{\mu}_2^{(nc)}(x)$  and  $\varepsilon_2^{(nc)}(x)$  as follows:

$$\tilde{\mu}_2^{(nc)}(x) = \sqrt{1 + \left(\sum_{s=2}^4 \|C_s\|_1 \frac{x^{s+1}}{(s+1)!}\right)^2} \quad (\text{B20})$$

$$\leq \sqrt{1 + \frac{n^2}{4} \left(\sum_{s=2}^4 (7^s + 12^s) \frac{(\Lambda_1 x)^{s+1}}{(s+1)!}\right)^2} \quad (\text{B21})$$

$$\leq \exp((35n)^2 (\Lambda_1 x)^6), \quad (\text{B22})$$

$$\begin{aligned} \varepsilon_2^{(nc)}(x) &\leq \frac{n^2}{4} (7^2 + 12^2) \sum_{s=2}^4 (7^s + 12^s) \frac{(\Lambda x)^{s+4}}{3!s!(s+4)} \\ &\quad + \frac{n}{2} (7^5 + 12^5) \frac{(\Lambda x)^6}{6!} \end{aligned} \quad (\text{B23})$$

$$\leq (3n^2 + 185n)(\Lambda x)^6 \leq 370n^2 (\Lambda x)^6, \quad (\text{B24})$$

Here, Eq. (B22) holds when  $\Lambda_1 x \leq 1/3$ , Eq. (B24) holds when  $\Lambda x \leq 21/72$ .

## 2. Estimating the gate counts

To estimate the performance, i.e., gate complexity of the Trotter-LCU algorithm based on nested-commutator compensation, we need to first estimate the number of segments  $\nu = t/x$  in the algorithm. This is determined by the following three constraints:

$$\begin{aligned} \Lambda_1 x, \Lambda x \leq 1 &\Leftrightarrow \nu \geq \max\{\Lambda_1 t, \Lambda t\}, \\ (\tilde{\mu}^{(nc)})^\nu \leq 2 &\Leftrightarrow \nu \ln \tilde{\mu}^{(nc)} \leq \ln 2, \\ \nu \tilde{\mu}^\nu (\varepsilon^{(nc)}) \leq \varepsilon &\Leftrightarrow \nu \varepsilon^{(nc)} \leq \frac{1}{2}\varepsilon. \end{aligned} \quad (\text{B25})$$

After we solve the required segment number  $\nu$  by Eq. (B25), we can estimate the gate number accordingly. Here, we introduce the method to estimate the gate number of the LCU part. The gate number in the implementation of second-order Trotter formula can be evaluated following the methods in Ref. [49]. In the worst-case scenario, the Pauli weight of the gate is determined by the weight of Pauli operators contained in  $C_4$  in Eq. (B12). The largest

Pauli weight is 6. As a result, a controlled-Pauli gate will cost at most six CNOT gates and no non-Clifford gate.

If we consider the paired algorithm, when we sample the third-order term, it will be a Pauli rotation unitary on four qubits. In this case, it will cost eight CNOT gates and two single-qubit Pauli rotation gates  $R_z(\theta)$ .

We summarize the whole procedure to estimate the gate complexity.

- (1) Input: Hamiltonian parameters:  $H = A + B$ ,  $n$ ,  $\Lambda$ ,  $\Lambda_1$ . Normalization requirements  $\mu = 2$ , accuracy requirements  $\varepsilon$ , time requirements  $t$ .
- (2) Normalization factor estimation.
  - (a) Analytical way (scalable). Using Eq. (B21) to get the function  $\tilde{\mu}^{(nc)}(x)$ .
  - (b) Numerical way (scalable). Using Eq. (B12) to get the form of  $C_2$ ,  $C_3$ , and  $C_4$ . Then get the function  $\tilde{\mu}^{(nc)}(x)$  by Eq. (B20).
- (3) Accuracy estimation.
  - (a) Analytical way (scalable). Using Eq. (B23) to get the function  $\varepsilon^{(nc)}(x)$ .
  - (b) Numerical way (unscalable). Calculate  $V_{2,\text{res}}(x) := U(x)S_2(x)^\dagger - I - \sum_{s=2}^4 C_s(x^{s+1})/((s+1)!)$  numerically. Solve its largest singular value, which is an upper bound of  $\varepsilon^{(nc)}(x)$ .
- (4) Based on the constraints in Eq. (B25), calculate the segment number  $\nu$ .
- (5) Analyze the CNOT and  $R_z(\theta)$  gate number.

### APPENDIX C: EXPLICIT NESTED-COMMUTATOR COMPENSATION FOR HIGHER-ORDER TROTTER REMAINDERS

In this section, we provide detailed results for the nested-commutator compensation for higher-order Trotter remainders introduced in Sec. V. Here, we will focus on the lattice model Hamiltonians in Eq. (10). The results for general Hamiltonians will be presented in Appendix E.

As is introduced in Sec. V A, for the (asymmetric) multiplicative remainder  $V_K(x)$ , our aim is to construct a LCU formula with the following form:

$$V_K(x) = I + \sum_{s=K+1}^{2K+1} F_{K,s}^{(nc)}(x) + F_{K,\text{res}}^{(nc)}(x), \quad (\text{C1})$$

where  $F_{K,\text{res}}^{(nc)}(x)$  denotes the term with  $x$ -order  $\mathcal{O}(x^{2K+2})$ . In practice, we will use the truncated LCU formula with

paired leading-order terms, derived in Eq. (110),

$$\begin{aligned} \tilde{V}_K^{(nc)}(x) &= I + \sum_{s=K+1}^{2K+1} F_{K,s}^{(nc)}(x) \\ &= I + i \sum_{s=K}^{2K} \eta_s^{(nc)} V_{K,s}^{(nc)} \\ &= \sum_{s=K}^{2K} \frac{\eta_s^{(nc)}}{\eta_\Sigma^{(nc)}} \left( I + \eta_\Sigma^{(nc)} V_{K,s}^{(nc)} \right) \\ &= \sqrt{1 + (\eta_\Sigma^{(nc)})^2} \sum_{s=K}^{2K} \frac{\eta_s^{(nc)}}{\eta_\Sigma^{(nc)}} R_{K,s}^{(nc)}(\eta_\Sigma), \end{aligned} \quad (\text{C2})$$

where  $\eta_\Sigma^{(nc)} = \sum_{s=K+1}^{2K+1} \eta_s^{(nc)} = \sum_{s=K}^{2K} \|C_{s-1}\|_1 (x^{s+1}) / ((s+1)!)$ .

We are going to finish the following tasks:

- (1) (Sec. C1) Derive the explicit formulas for the leading-order expansion terms  $F_{K,s}^{(nc)}(x)$  with  $s = K+1, \dots, 2K+1$ .
- (2) (Sec. C2) Prove the 1-norm  $\mu_K^{(nc)}(x)$  and error bound  $\varepsilon_K^{(nc)}(x)$  in Proposition 9.

#### 1. Derivation of LCU formula with nested-commutator form

We first introduce the canonical expression of the  $K$ th-order Trotter formula,

$$S_K(x) := W(x) = \prod_{j=1}^{\kappa} W_j(x) = W_\kappa(x) \cdots W_2(x) W_1(x), \quad (\text{C3})$$

where

$$W_j(x) := e^{-ixb_j B} e^{-ixa_j A}, \quad (\text{C4})$$

is the  $j$ th stage of the Trotter formula.  $\kappa$  is the stage number. We have  $\kappa = 1$  for  $K = 1$  and  $\kappa \leq 2 \times 5^{k-1}$  when  $K = 2k$ . The stage lengths  $a_j$  and  $b_j$  are determined based on Eqs. (24) and (25). We have

$$\begin{aligned} 0 \leq a_j, b_j \leq 1, \quad \forall j = 1, 2, \dots, \kappa, \\ \sum_{j=1}^{\kappa} a_j = \sum_{j=1}^{\kappa} b_j = 1. \end{aligned} \quad (\text{C5})$$

For example, for the second-order Trotter formula  $S_2(x) = e^{-ix\frac{A}{2}} e^{-ixB} e^{-ix\frac{A}{2}}$ , we set the stage number  $\kappa = 2$  with  $a_1 = \frac{1}{2}$ ,  $b_1 = 1$ ;  $a_2 = \frac{1}{2}$ ,  $b_2 = 0$ .

The Hermitian conjugate of  $W(x)$  is

$$W(x)^\dagger = \prod_{j=\kappa}^1 W_j(x)^\dagger = W_1(x)^\dagger W_2(x)^\dagger \cdots W_\kappa(x)^\dagger. \quad (\text{C6})$$

As is discussed in Sec. VA, to derive the nested-commutator form of  $V_K(x)$ , we first solve  $J_K(x)$  defined in Eq. (95). We have

$$\begin{aligned} R_K(x) &= \frac{d}{dx} W(x)^\dagger - iHW(x)^\dagger, \\ J_K(x) &:= W(x)R_K(x). \end{aligned} \quad (\text{C7})$$

From Proposition 5, we can write the  $K$ th-Trotter remainder  $V_K(x)$  and  $J_K(x)$  as the following form:

$$\begin{aligned} V_K(x) &:= I + M_K(x) = I + \sum_{s=K+1}^{2K+1} C_{s-1} \frac{x^s}{s!} + F_{K,\text{res}}(x), \\ J_K(x) &= \sum_{s=K}^{2K} C_s \frac{x^s}{s!} + J_{K,\text{res},2K}(x), \end{aligned} \quad (\text{C8})$$

where  $F_{K,\text{res}}(x) = \mathcal{O}(x^{2K+2})$  and  $J_{K,\text{res},2K}(x) = \mathcal{O}(x^{2K+1})$  are the higher-order remaining terms to be analyzed later. We also denote

$$\begin{aligned} F_L(x) &:= \sum_{s=K+1}^{2K+1} C_{s-1} \frac{x^s}{s!}, \\ J_L(x) &:= \sum_{s=K}^{2K} C_s \frac{x^s}{s!}, \end{aligned} \quad (\text{C9})$$

as the leading orders whose explicit forms will be calculated in this section. We will show that  $F_L(x)$  contains  $\mathcal{O}(n)$  term where  $n$  is the lattice size.

Now, we are going to solve the exact form of  $C_s$  for the leading orders. We first try to solve the succinct form of  $R_K(x)$  based on its definition in Eq. (C7). Taking the derivative for each Trotter stage, we have

$$\begin{aligned} R_K(x) &= \frac{d}{dx} \left[ \prod_{j=\kappa}^1 W_j(x)^\dagger \right] - iH \prod_{j=\kappa}^1 W_j(x)^\dagger \\ &= \sum_{j=\kappa}^1 \prod_{l=j-1}^1 W_l(x)^\dagger \frac{d}{dx} W_j(x)^\dagger \prod_{l=\kappa}^{j+1} W_l(x)^\dagger \\ &\quad - \sum_{j=\kappa}^1 (ia_j A + ib_j B) \prod_{l=\kappa}^1 W_l(x)^\dagger. \end{aligned} \quad (\text{C10})$$

Here we assume  $1 \leq j-1 < j+1 \leq \kappa$ . When  $j=1$  (or  $\kappa$ ), the value of the product  $\prod_{l=j-1}^1 W_l(x)^\dagger$  (or

$\prod_{l=\kappa}^{j+1} W_l(x)^\dagger$ ) will be regarded as  $I$ . Recall that  $W_j(x)^\dagger = e^{ix_j A} e^{ix_j B}$ . We further expand  $d/(dx)W_j(x)^\dagger$  and merge the two terms together,

$$\begin{aligned} R_K(x) &= \sum_{j=\kappa}^1 \prod_{l=j-1}^1 W_l(x)^\dagger (ia_j A W_j(x)^\dagger + W_j(x)^\dagger ib_j B) \cdot \\ &\quad \prod_{l=\kappa}^{j+1} W_l(x)^\dagger - \sum_{j=\kappa}^1 (ia_j A + ib_j B) \prod_{l=\kappa}^1 W_l(x)^\dagger \\ &= \sum_{j=\kappa}^1 \left[ \prod_{l=j}^1 W_l(x)^\dagger, ia_{j+1} A + ib_j B \right] \prod_{l=\kappa}^{j+1} W_l(x)^\dagger. \end{aligned} \quad (\text{C11})$$

Here, we assume  $a_{\kappa+1} = 0$ . Now, we simplify the commutator by splitting the product and then change the summation order,

$$\begin{aligned} R_K(x) &= \sum_{j=\kappa}^1 \sum_{s=j}^1 \left( \prod_{m=s-1}^1 W_m(x)^\dagger [W_s(x)^\dagger, ia_{j+1} A + ib_j B] \cdot \right. \\ &\quad \left. \prod_{m=j}^{s+1} W_m(x)^\dagger \right) \prod_{l=\kappa}^{j+1} W_l(x)^\dagger \\ &= \sum_{s=\kappa}^1 \sum_{j=\kappa}^s \left( \prod_{m=s-1}^1 W_m(x)^\dagger [W_s(x)^\dagger, ia_{j+1} A + ib_j B] \cdot \right. \\ &\quad \left. \prod_{m=j}^{s+1} W_m(x)^\dagger \right) \prod_{l=\kappa}^{j+1} W_l(x)^\dagger \\ &= \sum_{s=\kappa}^1 \prod_{m=s-1}^1 W_m(x)^\dagger [W_s(x)^\dagger, ic_{j+1} A + id_s B] \prod_{l=\kappa}^{s+1} W_l(x)^\dagger, \end{aligned} \quad (\text{C12})$$

where

$$c_s := \sum_{j=\kappa}^s a_j, \quad d_s = \sum_{j=\kappa}^s b_j, \quad s \leq \kappa. \quad (\text{C13})$$

We also set  $c_{\kappa+1} = 0$ .

Based on Eq. (C12), we now derive the succinct form of  $J_K(x)$ ,

$$\begin{aligned} J_K(x) &= W(x)R_K(x) = \prod_{l=1}^{\kappa} W_l(x)R_K(x) \\ &= \sum_{s=\kappa}^1 \prod_{l=s}^{\kappa} W_l(x) [W_s(x)^\dagger, ic_{s+1} A + id_s B] \prod_{m=\kappa}^{s+1} W_m(x)^\dagger. \end{aligned} \quad (\text{C14})$$

We expand each stage of the Trotter formula  $W_l(x)$  in the formula,

$$\begin{aligned}
J_K(x) &= i \sum_{j=\kappa}^1 \left( \prod_{l=j+1}^{\kappa} W_l(x)^\dagger (c_{j+1}A + d_jB) \prod_{l=\kappa}^{j+1} W_l(x)^\dagger - \prod_{l=j}^{\kappa} W_l(x)^\dagger (c_{j+1}A + d_jB) \prod_{l=\kappa}^j W_l(x)^\dagger \right) \\
&= -i \sum_{j=\kappa}^1 \left( \prod_{l=j}^{\kappa} e^{-i\tau b_l \text{ad}_B} e^{-i\tau a_l \text{ad}_A} (c_{j+1}A + d_jB) - \prod_{l=j+1}^{\kappa} e^{-i\tau b_l \text{ad}_B} e^{-i\tau a_l \text{ad}_A} (c_{j+1}A + d_jB) \right). \quad (\text{C15})
\end{aligned}$$

Finally, we apply the following operator-valued Taylor expansion formula with integral form of the remainder:

$$Q(x) = \sum_{s=0}^k \frac{x^s}{s!} Q^{(s)}(0) + \int_0^x d\tau \frac{(x-\tau)^k}{k!} Q^{(k+1)}(\tau). \quad (\text{C16})$$

By the general Libniz formula, we obtain the derivatives of  $J_K(x)$  as follows:

$$\begin{aligned}
J_K^{(s)}(x) &= (-i)^{s+1} \sum_{j=\kappa}^1 \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ m_j, n_j \neq 0; \sum_{l=j}^{\kappa} m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \\
&\quad \prod_{l=j}^{\kappa} (b_l^{m_l} a_l^{n_l}) \prod_{l=j}^{\kappa} (e^{-i\tau b_l \text{ad}_B} \text{ad}_B^{m_l} e^{-i\tau a_l \text{ad}_A} \text{ad}_A^{n_l}) (c_{j+1}A + d_jB). \quad (\text{C17})
\end{aligned}$$

We can then expand  $J_K(x)$  around  $t = 0$  as follows:

$$J_K(x) = \sum_{s=K}^{2K} C_s \frac{x^s}{s!} + J_{K, \text{res}, 2K}(x). \quad (\text{C18})$$

Here, we use the order condition in Proposition 5 so that the terms with expansion order from 0 to  $K-1$  are all zeros. The  $s$ th-order term  $C_s$  and the  $2K$ th-order residue  $J_{K, \text{res}, 2K}(x)$  can then be expressed as

$$\begin{aligned}
C_s &= J_K^{(s)}(0), \\
J_{K, \text{res}, 2K}(x) &= \int_0^x d\tau \frac{(x-\tau)^{2K}}{(2K)!} J_K^{(2K+1)}(\tau). \quad (\text{C19})
\end{aligned}$$

## 2. Norm bounds for LCU formula

In Sec. C 1 we have derived the explicit form of the LCU formula for the  $K$ th-order Trotter remainder  $V_K(x)$ . Based on Eqs. (C15), (C17), (C19), and Proposition 6, we are going to prove the 1-norm bound  $\mu_K^{(nc)}(x)$  and error bound  $\varepsilon_K^{(nc)}(x)$  in Proposition 9.

First of all, we need to estimate the norms of  $J_K^{(s)}(x)$ . We have the following results.

*Proposition 12 (Upper bound of the norm of nested commutators).* Consider a lattice Hamiltonian  $H = A + B$  with the form in Eq. (10). Suppose the spectral norm and 1-norm of its components  $H_{j,j+1}$  are bounded by  $\Lambda$  and  $\Lambda_1$ .

Then for the nested commutators appearing in Eq. (C19), we have the following bound:

$$\begin{aligned}
&\left\| \prod_{l=j}^{\kappa} e^{-i\tau b_l \text{ad}_B} \text{ad}_B^{m_l} e^{-i\tau a_l \text{ad}_A} \text{ad}_A^{n_l} (c_{j+1}A + d_jB) \right\| \\
&\leq \frac{n}{2} 2^s \Lambda^{s+1} \left( c_{j+1} \prod_{l=j}^{\kappa} (2(l-j) + 2)^{m_l} (2(l-j) + 1)^{n_l} \right. \\
&\quad \left. + d_j \prod_{l=j}^{\kappa} (2(l-j) + 3)^{m_l} (2(l-j) + 2)^{n_l} \right), \quad (\text{C20})
\end{aligned}$$

where  $\{m_l, n_l\}_{l=1}^{\kappa}$  are non-negative integers satisfying  $\sum_{l=j}^{\kappa} (m_l + n_l) = s$ .  $\kappa$  is the stage number determined by the Trotter formula in Eq. (C3).  $\{a_l, b_l\}_{l=1}^{\kappa}$  are defined by the Trotter formula in Eq. (C4).  $\{c_l, d_l\}_{l=1}^{\kappa}$  are defined in Eq. (C13). As a result, we can bound the norm of  $J_K^{(s)}(x)$  defined in Eq. (C17) as follows:

$$\|J_K^{(s)}(x)\| \leq n\kappa(4\kappa + 5)^s 2^s \Lambda^{s+1}. \quad (\text{C21})$$

The 1-norm upper bound for  $J_K^{(s)}(x)$  is to simply replace  $\Lambda$  by  $\Lambda_1$ .

*Proof.* To begin with, we now focus on one Hamiltonian term  $H_{j,j+1}$  in  $A$  and bound the norm

$$\begin{aligned} & \left\| \prod_{l=j}^{\kappa} e^{-i\tau b_l \text{ad}_B} \text{ad}_B^{m_l} e^{-i\tau a_l \text{ad}_A} \text{ad}_A^{n_l} (H_{j,j+1}) \right\| \\ & \leq 2^s \Lambda^{s+1} \prod_{l=j}^{\kappa} (2(l-j) + 2)^{m_l} (2(l-j) + 1)^{n_l}. \quad (\text{C22}) \end{aligned}$$

To this end, we decompose operator to the elementary nested commutators with the form,

$$\begin{aligned} & e^{-i\tau \text{ad}_B} \text{ad}_{H_{j_s j_s + 1}} \cdots \text{ad}_{H_{j_s - m_\kappa + 1 j_s - m_\kappa + 1 + 1}} \\ & \cdots \\ & e^{-i\tau \text{ad}_B} \text{ad}_{H_{j_{m_j} + n_j j_{m_j} + n_j + 1}} \cdots \text{ad}_{H_{j_{n_j} + 1 j_{n_j} + 1 + 1}} \\ & e^{-i\tau \text{ad}_A} \text{ad}_{H_{j_1 j_1 + 1}} \cdots \text{ad}_{H_{j_1 j_1 + 1}} H_{j,j+1}, \end{aligned} \quad (\text{C23})$$

where  $j_1, j_2, \dots, j_s$  are the possible vertex indices. For each elementary nested commutator, the spectral norm can be bounded by

$$(2\Lambda)^s \Lambda. \quad (\text{C24})$$

This can be done by expand all the commutators and apply triangle inequality. Here, we use the property that the spectral norm of all the exponential operator with anti-Hermitian exponent is 1.

Now, we count the number of the possible elementary commutators with the form in Eq. (C23). We will check the action of the adjoint operators from right to left. For the first location, we know that  $\text{ad}_A H_{j,j+1} = 0$ . For simplicity, we keep the term  $\text{ad}_{H_{j,j+1}} H_{j,j+1}$  in the counting of elementary commutators. If the next ad is still  $\text{ad}_A$ , the support will still be on the two qubits  $j$  and  $j+1$ . As a result, there will be one possible elementary term  $\text{ad}_{H_{j,j+1}}$  left. Similarly, the exponential operator  $e^{-i\tau \text{ad}_A}$  will not enlarge

the support since one can expand it to the power of  $\text{ad}_A$ . The support will be enlarged when  $\text{ad}_B$  comes into play. In this layer, the support of the operator will be expanded to four qubits:  $j-1, j, j+1$ , and  $j+2$ . We can decompose  $\text{ad}_B$  to 2 nonzero elementary elements,  $\text{ad}_{H_{j-1,j}}$  and  $\text{ad}_{H_{j+1,j+2}}$ . If the next operator is still  $\text{ad}_B$  or  $e^{-i\tau \text{ad}_B}$ , it will not enlarge the support. Following this logic, we can see that the number of possible elementary commutators is bounded by

$$\begin{aligned} & (2(\kappa - j) + 2)^{m_\kappa} (2(\kappa - j) + 1)^{n_\kappa} \cdots 4^{m_j + 1} 3^{n_j + 1} 2^{m_j} 1^{n_j} \\ & = \prod_{l=j}^{\kappa} (2(l-j) + 2)^{m_l} (2(l-j) + 1)^{n_l}. \quad (\text{C25}) \end{aligned}$$

We remark that, the elementary nested commutators with  $n_j \neq 0$  is actually 0. Here, we keep these commutators for the simplicity of counting.

Combining the number of elementary nested commutators and the norm bound for each commutator and applying triangle inequality, we will obtain Eq. (C22).

Similarly, we can check one Hamiltonian term  $H_{j,j+1}$  in  $B$  and bound the norm with

$$\begin{aligned} & \left\| \prod_{l=j}^{\kappa} e^{-i\tau b_l \text{ad}_B} \text{ad}_B^{m_l} e^{-i\tau a_l \text{ad}_A} \text{ad}_A^{n_l} (H_{j,j+1}) \right\| \\ & \leq 2^s \Lambda^{s+1} \prod_{l=j}^{\kappa} (2(l-j) + 3)^{m_l} (2(l-j) + 2)^{n_l}. \quad (\text{C26}) \end{aligned}$$

The counting logic is similar to the case for  $H_{j,j+1}$  in  $A$ . The only difference is that when counting the number of elementary nested commutators, the action of the first  $\text{ad}_A$  will enlarge the operator space to four qubits.

Applying Eqs. (C22) and (C26) for all the components  $H_{j,j+1}$  in  $H$ , we will obtain Eq. (C20).

Now, we apply Eq. (C20) to bound the norm of  $J_K^{(s)}(x)$  in Eq. (C17). We have

$$\begin{aligned} \|J_K^{(s)}(x)\| & \leq \sum_{j=\kappa}^1 \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ \sum_{l=j}^{\kappa} m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \prod_{l=j}^{\kappa} (b_l^{m_l} a_l^{n_l}) \left\| \prod_{l=j}^{\kappa} (e^{-i\tau b_l \text{ad}_B} \text{ad}_B^{m_l} e^{-i\tau a_l \text{ad}_A} \text{ad}_A^{n_l}) (c_{j+1} A + d_j B) \right\| \\ & \leq 2^s \Lambda^{s+1} \frac{n}{2} \sum_{j=\kappa}^1 \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ \sum_{l=j}^{\kappa} m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \prod_{l=j}^{\kappa} (b_l^{m_l} a_l^{n_l}) \\ & \quad \times \left( c_{j+1} \prod_{l=j}^{\kappa} (2(l-j) + 2)^{m_l} (2(l-j) + 1)^{n_l} + d_j \prod_{l=j}^{\kappa} (2(l-j) + 3)^{m_l} (2(l-j) + 2)^{n_l} \right) \end{aligned}$$

$$\begin{aligned}
&\leq 2^s \Lambda^{s+1} \frac{n}{2} \sum_{j=\kappa}^1 c_{j+1} \left( \sum_{l=j}^{\kappa} b_l (2(l-j) + 2) + a_l (2(l-j) + 1) \right)^s + d_j \left( \sum_{l=j}^{\kappa} b_l (2(l-j) + 3) + a_l (2(l-j) + 2) \right)^s \\
&\leq 2^s \Lambda^{s+1} \frac{n}{2} \sum_{j=\kappa}^1 c_{j+1} (4(\kappa - j) + 3)^s + d_j (4(\kappa - j) + 5)^s \\
&\leq 2^s \Lambda^{s+1} \frac{n}{2} \sum_{j=\kappa}^1 (4(\kappa - j) + 3)^s + (4(\kappa - j) + 5)^s \\
&\leq 2^s \Lambda^{s+1} \frac{n}{2} \cdot 2\kappa (4\kappa + 5)^s = n\kappa (4\kappa + 5)^s 2^s \Lambda^{s+1}. \tag{C27}
\end{aligned}$$

In the third line, we use the multinomial theorem. In the fourth line, we use the fact that  $a_l, b_l > 0$  and  $\sum_l a_l, \sum_l b_l \leq 1$ . The fifth line is due to  $0 \leq c_j, d_j < 1$  for all  $j$ . In the sixth line, we use the following bound:

$$\begin{aligned}
&\sum_{j=1}^{\kappa} (4(\kappa - j) + a)^s = \sum_{l=0}^{\kappa-1} (4l + a)^s \\
&\leq \int_0^{\kappa} (4x + a)^s dx = \frac{1}{4} \frac{(4\kappa + a)^{s+1} - a^{s+1}}{s+1} \\
&= \kappa \frac{1}{s+1} \sum_{m=0}^s (4\kappa + a)^m a^{s-m} \leq \kappa (4\kappa + a)^s. \tag{C28}
\end{aligned}$$

Here,  $a$  is any real number.

Since 1-norm can be estimated based on the same logic by counting the number of nested commutators and the 1-norm of each nest commutator, the derivation for the 1-norm is similar by replacing  $\Lambda$  to  $\Lambda_1$ . ■

Now, we prove the 1-norm bound  $\mu_K^{(nc)}(x)$  of  $\tilde{V}_K(x)$ . From Proposition 6 we have

$$\begin{aligned}
\|\tilde{V}_K(x)\|_1 &= \sqrt{1 + \left( \sum_{s=K}^{2K} \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2} \\
&\leq 1 + \frac{1}{2} \left( \sum_{s=K}^{2K} \|C_s\|_1 \frac{x^{s+1}}{(s+1)!} \right)^2 \\
&\leq 1 + \frac{1}{2} n^2 \kappa^2 \left( \sum_{s=K}^{2K} \beta^s \frac{(\Lambda_1 x)^{s+1}}{(s+1)!} \right)^2
\end{aligned}$$

$$\begin{aligned}
&\leq 1 + \frac{1}{2} n^2 \kappa^2 \left( (K+1) \beta^{K+1} \frac{(\Lambda_1 x)^{K+1}}{(K+1)!} \right)^2 \\
&\leq 1 + \frac{n^2 \kappa^2 \beta^{2(K+1)}}{2(K!)^2} (\Lambda_1 x)^{2K+2} \\
&\leq \exp \left( \frac{n^2 \kappa^2 \beta^{2(K+1)}}{2(K!)^2} (\Lambda_1 x)^{2K+2} \right). \tag{C29}
\end{aligned}$$

Here, we set  $\beta := 2(4\kappa + 5)$ . In the third line, we use Proposition 12. In the fourth line, we use the assumption that  $\beta \Lambda_1 x \leq (K+2)$ .

Then, we prove the distance bound  $\varepsilon_K^{(nc)}(x) = \|F_{K,\text{res}}\|$ . From Proposition 6 we know that we only need to bound  $\|J_{K,L}(\tau)\|$ ,  $\|J_{K,\text{res},2K}(\tau)\|$ , and  $M_K(\tau)$  based on Eq. (114).

We start from  $\|J_{K,L}(\tau)\|$ . From Proposition 7 we have

$$\begin{aligned}
\|J_{K,L}(\tau)\| &\leq \sum_{s=K}^{2K} \|C_s\| \frac{\tau^s}{s!} \\
&\leq n\kappa \sum_{s=K}^{2K} \beta^s \Lambda^{s+1} \frac{\tau^s}{s!}. \tag{C30}
\end{aligned}$$

Then, we bound  $\|J_{K,\text{res},2K}(\tau)\|$  and  $\|M_K(\tau)\|$ . From Eq. (114) and Proposition 12 we have

$$\begin{aligned}
\|J_{K,\text{res},2K}(\tau)\| &\leq \int_0^{\tau} d\tau_1 \frac{(\tau - \tau_1)^{2K}}{(2K)!} \|J_K^{(2K+1)}(\tau_1)\| \\
&\leq n\kappa \frac{(\beta\tau)^{2K+1}}{(2K+1)!} \Lambda^{2K+2}, \\
\|M_K(\tau)\| &\leq \int_0^{\tau} d\tau_1 \int_0^{\tau_1} d\tau_2 \frac{(\tau_1 - \tau_2)^{(K-1)}}{(K-1)!} \|J_K^{(K)}(\tau_2)\| \\
&\leq n\kappa \frac{\beta^K \tau^{K+1}}{(K+1)!} \Lambda^{K+1}. \tag{C31}
\end{aligned}$$

Based on Eqs. (C30), (C31) and Proposition 6 we have

$$\begin{aligned}
\|F_{K,\text{res}}(x)\| &\leq \int_0^\tau (\|M_K(\tau)\| \|J_L(\tau)\| + \|J_{K,\text{res},2K}(\tau)\|) \\
&\leq \int_0^x d\tau (n\kappa)^2 \sum_{s=K}^{2K} \frac{\beta^{s+K} \Lambda^{s+K+2} \tau^{s+K+1}}{(K+1)!s!} \\
&\quad + (n\kappa) \frac{\beta^{2K+1} \Lambda^{2K+2} \tau^{2K+1}}{(2K+1)!} \\
&= (n\kappa)^2 \sum_{s=K}^{2K} \frac{\beta^{s+K} (\Lambda x)^{s+K+2}}{(K+1)!s!(s+K+2)} \\
&\quad + (n\kappa) \frac{\beta^{2K+1} (\Lambda x)^{2K+2}}{(2K+2)!} \\
&\leq (n\kappa)^2 (K+1) \frac{\beta^{2K} (\Lambda x)^{2K+2}}{(2K+2)K!(K+1)!} \\
&\quad + (n\kappa) \frac{\beta^{2K+1} x^{2K+2}}{(2K+2)!} \\
&\leq (n\kappa)^2 \frac{\beta^{2K+1}}{K!(K+1)!} (\Lambda x)^{2K+2}. \tag{C32}
\end{aligned}$$

In the fourth line, we assume  $\beta \Lambda x \leq K+1$ . In the fifth line, we use the fact that  $n\kappa > 1$ .

To summarize, from Eqs. (C29) and (C32), we have proven that,

$$\begin{aligned}
\mu_K^{(nc)} &:= \|V_K(x)\|_1 \leq \exp\left(\frac{n^2 \kappa^2 \beta^{2(K+1)}}{2(K!)^2} (\Lambda_1 x)^{2K+2}\right), \\
\varepsilon_K^{(nc)} &:= \|\tilde{V}_K(x) - V_K(x)\| = \|F_{K,\text{res}}(x)\| \\
&\leq (n\kappa)^2 \frac{\beta^{2K+1}}{K!(K+1)!} (\Lambda x)^{2K+2}. \tag{C33}
\end{aligned}$$

Here,  $\beta := 2(4\kappa + 5)$ . We assume that  $\beta x \leq \max\{(K+1)/\Lambda, (K+2)/\Lambda_1\}$ . This finish the proof of Proposition 9.

#### APPENDIX D: EFFICIENT SAMPLING FOR THE HIGHER-ORDER NESTED-COMMUTATOR COMPENSATION ALGORITHM

Following the analysis in Appendix C, we now design a general sampling method for the general  $K$ th-order Trotter remainder  $\tilde{V}_K^{(nc)}(x)$ . We consider the general  $K$ th-order Trotter formula in the canonical form in Eq. (C3).

The truncated Trotter remainder can be expanded as

$$\begin{aligned}
V_K(x) &:= I + \sum_{s=K+1}^{2K+1} F_{K,s}, \\
F_{K,s} &= C_{s-1} \frac{x^s}{s!}.
\end{aligned} \tag{D1}$$

Based on Eq. (C17), we have shown that  $C_s$  can be written as

$$C_s = \sum_{j=\kappa}^1 C_{s,j}^{(A')} + C_{s,j}^{(B')}, \tag{D2}$$

where  $A' := -iA, B' := -iB$ ,

$$\begin{aligned}
C_{s,j}^{(A')} &:= c_{j+1} \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ m_j, n_j \neq 0; \sum_{l=j}^\kappa m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \\
&\quad \prod_{l=j}^\kappa (b_l^{m_l} a_l^{n_l}) (\text{ad}_{B'}^{m_l} \text{ad}_{A'}^{n_l}) A', \\
C_{s,j}^{(B')} &:= d_j \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ m_j, n_j \neq 0; \sum_{l=j}^\kappa m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \\
&\quad \prod_{l=j}^\kappa (b_l^{m_l} a_l^{n_l}) (\text{ad}_{B'}^{m_l} \text{ad}_{A'}^{n_l}) B'. \tag{D3}
\end{aligned}$$

The values  $c_s$  and  $d_s$  are defined in Eq. (C13).

We now construct efficient random sampling of  $C_{s,j}^{(A')}$  and  $C_{s,j}^{(B')}$  Eq. (D3) based on LCU formulas with the 1-norm of

$$\begin{aligned}
\mu_{s,j}^{(A')} &= 2^s \Lambda_1^{s+1} \frac{n}{2} c_{j+1} \chi_{A',j}^s, \\
\mu_{s,j}^{(B')} &= 2^s \Lambda_1^{s+1} \frac{n}{2} d_j \chi_{B',j}^s,
\end{aligned} \tag{D4}$$

respectively. Here,

$$\begin{aligned}
\chi_{A',j} &:= \sum_{l=j}^\kappa b_l(2(l-j)+2) + a_l(2(l-j)+1), \\
\chi_{B',j} &:= \sum_{l=j}^\kappa b_l(2(l-j)+3) + a_l(2(l-j)+2).
\end{aligned} \tag{D5}$$

There are  $n/2$  summands  $\{H_{q,q+1}\}$  in  $A'$ . We now focus on a generic summand  $H_{q,q+1}$  in  $A'$  and check the action of the

adjoint operators on it,

$$\begin{aligned} & \text{ad}_{s,j;\vec{m},\vec{n}}^{(A')} H_{q,q+1} := \\ & (-i) \text{ad}_{B'}^{m_\kappa} \text{ad}_{A'}^{n_\kappa} \cdots \text{ad}_{B'}^{m_{j+1}} \text{ad}_{A'}^{n_{j+1}} \text{ad}_{B'}^{m_j} \text{ad}_{A'}^{n_j} H_{q,q+1}. \end{aligned} \quad (\text{D6})$$

Recall that  $m_\kappa + n_\kappa + \cdots + m_{j+1} + n_{j+1} + m_j + n_j = s$ .

We would like to follow that construction in Proposition 12, where we first decompose the adjoint operator in Eq. (D6) into the elementary operators with the form,

$$\text{ad}_{q_1,s}^{(s)} H_{q,q+1} := (-i)^{s+1} \text{ad}_{H_{q_s,q_s+1}} \cdots \text{ad}_{H_{q_1,q_1+1}} H_{q,q+1}. \quad (\text{D7})$$

For each elementary nested commutator, the 1-norm is

$$\Lambda_1^{(s)} = (2\Lambda_1)^s \Lambda_1. \quad (\text{D8})$$

Here, we have assumed that all the summands  $\{H_{q,q+1}\}$  in  $A$  or  $B$  have been padded similar to Eq. (18) so that their 1-norms are all  $\Lambda_1$ .

Now, we count the number of elementary nested commutators with the form of Eq. (D7) in the adjoint operator  $\text{ad}_{s,j}^{(A')}$  in Eq. (D6). In the proof of Proposition 12, we show the number of possible elementary nested commutator is upper bounded by

$$\begin{aligned} & N(\text{ad}_{s,j;\vec{m},\vec{n}}^{(A')} H_{q,q+1}) \\ &= (2(\kappa - j) + 2)^{m_\kappa} (2(\kappa - j) + 1)^{n_\kappa} \cdots 4^{m_{j+1}} 3^{n_{j+1}} 2^{m_j} 1^{n_j} \\ &= \prod_{l=j}^{\kappa} (2(l - j) + 2)^{m_l} (2(l - j) + 1)^{n_l}. \end{aligned} \quad (\text{D9})$$

We now discuss how to achieve this bound by ‘padding’ zero-valued elementary nested commutators into the decomposition of  $\text{ad}_{s,j}^{(A')}$ . We notice that, after the sequential action of  $\text{ad}_{A'}^{n_j}, \text{ad}_{B'}^{m_j}, \text{ad}_{A'}^{n_{j+1}}, \text{ad}_{B'}^{m_{j+1}}, \dots$ , the support, i.e., the index of the qubit where the operators act nontrivially, of the resulting operator is given by the ‘light cone’ of  $q : (q + 1), (q - 1) : (q + 2), (q - 2) : (q + 3), (q - 3) : (q + 4), \dots$ , as is illustrated in Fig. 6. We keep track of the largest possible support of the resulting adjoint operators and pad the LCU formula in the following situations:

- (1) (Padding small summands) When the 1-norm of the summand  $H_{q_l,q_{l+1}}$  is smaller than  $\Lambda_1$ , we add extra  $\pm I$  terms in the LCU formula of  $H_{q_l,q_{l+1}}$  to make its 1-norm to  $\Lambda_1$ , similar to Eq. (18).
- (2) (Padding 0-valued commutators) Many nested commutators with the form of Eq. (D7) may be 0 due to the commutation relationship, for example  $\text{ad}_{H_{q+2,q+3}} \text{ad}_{H_{q-1,q}} H_{q,q+1} = 0$ , since the support of

$\text{ad}_{H_{q-1,q}} H_{q,q+1}$  is on qubits  $q - 1, q$  and  $q + 1$ , which commutes with  $H_{q+2,q+3}$ . However, we keep all these 0-valued terms as long as the support of the nested commutator is in the ‘light-cone’ range of the operator.

- (3) (Padding boundary terms) For the summands  $H_{q,q+1}$  which are close to the boundary, after a few action of the adjoint operators, the ‘light cone’ will touch the boundary of the system. In this case, we introduce virtual padding ancillary qubits and extra padding nested commutators on it:
  - (a) For the summand  $H_0$  or  $H_n$ , which own only one-qubit support on the boundary, we redefine the Pauli terms in the LCU of them. For example, if  $H_0 = \Lambda_1 \sum_{\omega} p_{\omega} P_0^{(\omega)}$ , we then redefined it to  $H_{-1,0} = \sum_{\omega} p_{\omega} (I_{-1} \otimes P_0)^{(\omega)}$ .
  - (b) We define the ‘virtual’ summand  $H_{q_l,q_{l+1}}$  where qubits  $q_l, q_{l+1}$  are all virtual qubits as

$$H_{q_l,q_{l+1}} := \frac{\Lambda_1}{2} (I_{q_l,q_{l+1}} + (-I_{q_l,q_{l+1}})). \quad (\text{D10})$$

Since all the operations on the virtual qubits are  $\pm I$ , we do not need to introduce these qubits in the real implementation.

After these padding, we have now construct the LCU formula of  $\text{ad}_{s,j}^{(A')}$  with the following form:

$$\begin{aligned} & \text{ad}_{s,j;\vec{m},\vec{n}}^{(A')} H_{q,q+1} \\ &= N(\text{ad}_{s,j}^{(A')} H_{q,q+1}) \cdot \\ & \sum_{q_1, \dots, q_s} (-i)^{s+1} \text{ad}_{H_{q_s,q_s+1}} \cdots \text{ad}_{H_{q_1,q_1+1}} H_{q,q+1} \\ &=: N(\text{ad}_{s,j}^{(A')} H_{q,q+1}) \sum_{q_1, \dots, q_s} \text{ad}_{q_1,s}^{(s)} H_{q,q+1} \end{aligned} \quad (\text{D11})$$

where  $q_1, \dots, q_s$  are indices in the light cone region. Recall that all the elementary operators with the form Eq. (D7) are with the same 1-norm of  $(2\Lambda_1)^s \Lambda_1$ , irrelevant of the qubit index  $q$  and the rank number  $\{m_l, n_l\}_{l=j}^{\kappa}$ . Therefore, the 1-norm of  $\text{ad}_{s,j}^{(A')} H_{q,q+1}$  is

$$\mu\{\text{ad}_{s,j;\vec{m},\vec{n}}^{(A')} H_{q,q+1}\} = N\left(\text{ad}_{s,j;\vec{m},\vec{n}}^{(A')}\right) (2\Lambda_1)^s \Lambda_1, \quad (\text{D12})$$

which is irrelevant of the qubit index  $q$ .

Now, based on Eq. (D11), we can write  $C_{s,j}^{(A')}$  in Eq. (D3) as

$$\begin{aligned} \frac{C_{s,j}^{(A')}}{C_{j+1}} &= \sum_{\substack{1 \leq q \leq n \\ q:\text{odd}}} \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ \sum_{l=j}^\kappa m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \\ &\quad \prod_{l=j}^\kappa (b_l^{m_l} a_l^{n_l}) \text{ad}_{s,j; \vec{m}, \vec{n}}^{(A')} H_{q,q+1} \\ &= \sum_{\substack{1 \leq q \leq n \\ q:\text{odd}}} \sum_{\substack{m_j, n_j, \dots, m_\kappa, n_\kappa \\ \sum_{l=j}^\kappa m_l + n_l = s}} \binom{s}{m_j, n_j, \dots, m_\kappa, n_\kappa} \\ &\quad \prod_{l=j}^\kappa (b_l^{m_l} a_l^{n_l}) N(\text{ad}_{s,j}^{(A')} H_{q,q+1}) \sum_{q_1, \dots, q_s} \text{ad}_{q_1:s}^{(s)} H_{q,q+1} \\ &= \chi_{A',j}^s \text{Mul}(\{m_j, n_j, \dots, m_\kappa, n_\kappa\}; \{\vec{p}_{A',b}, \vec{p}_{A',a}\}; s) \cdot \\ &\quad \sum_{\substack{1 \leq q \leq n \\ q:\text{odd}}} \sum_{q_1, \dots, q_s} \text{ad}_{q_1:s}^{(s)} H_{q,q+1}, \end{aligned} \quad (\text{D13})$$

where

$$p_{A',b;l} = \frac{b_l(2(l-j)+2)}{\chi_{A',j}}, \quad p_{A',a;l} = \frac{a_l(2(l-j)+2)}{\chi_{A',j}}, \quad (\text{D14})$$

for  $l = j, j+1, \dots, \kappa$  and  $\text{Mul}(v; \vec{p}; s)$  denotes a multinomial distribution where we sample the variable value  $v$  based on the probability distribution  $\vec{p}$  for  $s$  times. Based on Eqs. (D12) and (D13), we can easily check that the 1-norm of  $C_{s,j}^{(A')}$  is given by  $\mu_{s,j}^{(A')}$  in Eq. (D4). More importantly, since the 1-norm of  $\text{ad}_{q_1:s}^{(s)} H_{q,q+1}$  is independent of  $q$  and  $q_1, \dots, q_s$ , the sampling of  $q$  from all odd qubits and  $q_1, \dots, q_s$  from the light-cone region follows uniform distribution.

Similarly, we can decompose and pad extra terms in  $C_{s,j}^{(B')}$  to construct LCU with the following form:

$$\begin{aligned} \frac{C_{s,j}^{(B')}}{d_j} &= \chi_{B',j}^s \text{Mul}(\{m_j, n_j, \dots, m_\kappa, n_\kappa\}; \{\vec{p}_{B',b}, \vec{p}_{B',a}\}; s) \cdot \\ &\quad \sum_{\substack{1 \leq q \leq n \\ q:\text{even}}} \sum_{q_1, \dots, q_s} \text{ad}_{q_1:s}^{(s)} H_{q,q+1}, \\ p_{B',b;l} &= \frac{b_l(2(l-j)+3)}{\chi_{B',j}}, \quad p_{B',a;l} = \frac{a_l(2(l-j)+2)}{\chi_{B',j}}, \end{aligned} \quad (\text{D15})$$

for  $l = j, j+1, \dots, \kappa$ , so that the 1-norm of  $C_{s,j}^{(B')}$  is given by  $\mu_{s,j}^{(B')}$  in Eq. (D4). Again, the 1-norm of the operator

$\sum_{q_1, \dots, q_s} \text{ad}_{q_1:s}^{(s)} H_{q,q+1}$  is  $(2\Lambda_1)^s \Lambda_1$ , independent of  $q$  and  $q_1, \dots, q_s$ . Therefore, the 1-norm of  $C_s$  in Eq. (D2) is

$$\mu\{C_s\} = \sum_{j=\kappa}^1 \mu_{s,j}^{(A')} + \mu_{s,j}^{(B')}, \quad (\text{D16})$$

and the 1-norm of  $F_{K,s}$  is

$$\mu\{F_{K,s}\} = \mu\{C_{s-1}\} \frac{\chi^s}{s!}. \quad (\text{D17})$$

Based on the above analysis, we summarize the overall sampling procedure in Fig. 12 and Algorithm 2. We consider a multistage sampling: first, we sample the expansion order  $s$  based on  $\mu\{F_{K,s}\}$ ; second, for a given expansion order  $s$ , we sample the terms  $C_{s-1}^{(A')}$  or  $C_{s-1}^{(B')}$  in the nested commutator  $C_{s-1}$  based on Eqs. (D2) and (D4); third, we sample the power of the adjoint operators  $\vec{m}$  and  $\vec{n}$  based on the multinomial distribution in Eqs. (D13) and (D15); fourth, we uniformly sample the specific Hamiltonian summands  $H_{q,q+1}$  and uniformly sample the adjoint Hamiltonian summands  $\{H_{q_1, q_1+1}, \dots, H_{q_{s-1}, q_{s-1}+1}\}$ , each from the light-cone region; finally, if there are multiple terms in the Hamiltonian summands, we then uniformly sample the specific Pauli terms in the summands.

Now, we analyze the space and time cost of the whole sampling algorithm in Algorithm 2. The calculation of the 1-norms of  $C_s$  and its components  $C_{s,j}^{(A')}$  and  $C_{s,j}^{(B')}$  requires  $\mathcal{O}(K\kappa)$  spacetime resources. Consider a parallel calculation, we need  $\mathcal{O}(K\kappa)$  spatial resources and  $\mathcal{O}(1)$  time resources. We store all the above 1-norm coefficients in the memory with the size  $\mathcal{O}(K\kappa)$ . The sampling of  $F_{K,s}$  from  $K$  discrete values requires  $\mathcal{O}(\log K)$  steps [50]. For a given nested commutator  $C_{s-1}$ , the sampling of  $C_{s,j}^{(A')}$  and  $C_{s,j}^{(B')}$  from  $\kappa$  discrete values requires  $\mathcal{O}(\log \kappa)$  steps. The multinomial sampling of the power of adjoint operators  $\vec{m}$  and  $\vec{n}$  requires  $\mathcal{O}(s \log \kappa)$  steps. Finally, the uniform sampling of the Hamiltonian summands from the light-cone region requires  $\mathcal{O}(s \log n)$  steps. To summarize, the space and time cost of Algorithm 2 are  $\mathcal{O}(K\kappa)$  and  $\mathcal{O}(K(\log \kappa + \log n))$ , respectively.

## APPENDIX E: NESTED-COMMUTATOR COMPENSATION FOR TROTTER FORMULAS OF GENERAL HAMILTONIANS

We now extend the methods to analyze the lattice model Hamiltonian to a general Hamiltonian. Consider a  $L$ -sparse Hamiltonian with the form  $H = \sum_{l=1}^L H_l$ . The  $K$ th-order Trotter formula ( $K = 1$  or  $2k$ ,  $k \in \mathbb{N}_+$ ) for  $U(x) = e^{-ixH}$  can be written as

$$S_K(x) = \prod_{j=1}^\kappa \prod_{l=1}^L e^{-ix a_{(j,l)} H_{\pi_j(l)}}. \quad (\text{E1})$$

Here,  $\kappa$  is the number of stages in the Trotter formula, that is, how many times each Hamiltonian component  $H_l$  is

**ALGORITHM 2.** Nested-commutator compensation algorithm: sampling of Pauli operators.

**Input:** An  $n$ -qubit Hamiltonian  $H$ ; unit evolution time  $0 < x < 1$  for each  $K$ th-order Trotter segment; the canonical form of the  $K$ th-order Trotter method with coefficients  $\{a_j, b_j\}$  for  $j = 1, \dots, \kappa$ ;

**Output:** Sampling of a Pauli operator  $P^{(\omega_j; \dots; \omega_{j_s}, b_s)}$  from the Trotter remainder  $\tilde{V}_K^{(nc)}(x)$ .

- 1: Calculate the 1-norm of the nested commutator  $C_s$  for  $s = K, \dots, 2K$  and its  $j$ th-stage components  $C_{s,j}^{(A')}$  and  $C_{s,j}^{(B')}$  for  $j = 1, \dots, \kappa$  based on Eq. (D4) and Eq. (D16).
- 2: Sample the expansion order  $s \in \{K, \dots, 2K\}$  based on the 1-norm  $\mu\{F_{K,s+1}\} = \mu\{C_s\} \frac{x^{s+1}}{(s+1)!}$ . This determines the sampled nested commutator  $C_s$ .
- 3: From  $C_s$ , sample the  $j$ th-stage components  $C_{s,j}^{(A')}$  or  $C_{s,j}^{(B')}$  based on the 1-norm  $\mu_{s,j}^{(A')}$  and  $\mu_{s,j}^{(B')}$  in Eq. (D4).
- 4: From  $C_{s,j}^{(A')}$  (or  $C_{s,j}^{(B')}$ ) with a given  $j$ , sample the power of the sequential adjoint operators  $m_j, n_j; \dots; m_\kappa, n_\kappa$  based on a multinomial distribution  $\text{Mul}(\{\bar{m}, \bar{n}\}; \{\bar{p}_{A'(B'),b}, \bar{p}_{A'(B'),a}\}; s)$  defined in Eq. (D13) and Eq. (D15).
- 5: Sample the index of the starting Hamiltonian summand  $q$  uniformly from  $A'$  ( $B'$ ). Sample the index  $q_1, \dots, q_s$  of the subsequent adjoint Hamiltonian summands, each from the “light cone” region of that location.
- 6: For Hamiltonian summands indexed by  $q$  and  $\{q_1, \dots, q_s\}$ , sample the Pauli operators  $P^{(\omega_j)}$  and  $\{P^{(\omega_{j_1})}, \dots, P^{(\omega_{j_s})}\}$  independently based on the padded LCU formula for each Hamiltonian summand in Eq. (18). For each adjoint location  $q_1, \dots, q_s$ , uniformly and independently sample the multiplication order  $b_1, \dots, b_s \in \{0, 1\}$ , which indicates the multiplication order of the Pauli operators.
- 7: Set  $P := P^{(\omega_j)}$
- 8: **for**  $l = 1$  **to**  $s$  **do** ▷ Calculate the output Pauli operator
- 9:     **if**  $b_l = 0$  **then**
- 10:         Set  $P := P^{(\omega_{j_1})} \cdot P$ ,
- 11:     **else**
- 12:         Set  $P := -P \cdot P^{(\omega_{j_1})}$ ,
- 13:     **end if**
- 14: **end for**
- 15: Output  $P$  as the sampled Pauli operator.

repeated in the implementation. We have  $\kappa = 1$  for  $K = 1$  and  $\kappa = 2 \times 5^{k-1}$  when  $K = 2k$ . The stage length coefficients  $a_{(j,l)}$  are determined based on Eqs. (24) and (25). The permutation  $\pi_j$  indicates the ordering of the summands  $\{H_l\}$  within the  $j$ th stage in the Trotter formula. In Suzuki’s constructions [8] of Trotter formulas considered in this work, we alternately reverse the ordering of summands between neighboring stages.

In what follows, we omit the subscript  $K$  in  $S_K(x)$  for simplicity. To further simplify the notation, we introduce the lexicographical order [20] for the pair of tuples  $(j, l)$  in Eq. (E1). For two pairs of tuples  $(j, l)$  and  $(j', l')$ , we have

- (1)  $(j, l) \geq (j', l')$  if  $j > j'$ , or if  $j = j'$  and  $l > l'$ .
- (2)  $(j, l) \succ (j', l')$  if  $(j, l) \geq (j', l')$  and  $(j, l) \neq (j', l')$ .

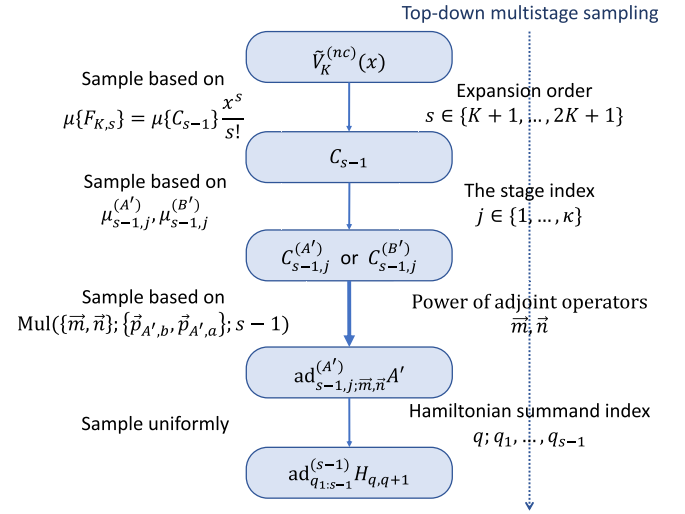


FIG. 12. A general procedure to obtain the sampling weights and perform the sampling in the nested-commutator compensation algorithm.

We can also define  $(j, l) \leq (j', l')$  and  $(j, l) \prec (j', l')$  in the same way. We denote the number of different tuples  $(j, l)$  in  $S_K(x)$  as  $\Upsilon$ , which is usually equal to  $\kappa L$ . We can then express  $S_K(x)$  and  $S_K(x)^\dagger$  in the following way:

$$S_K(x) = \prod_{(j,l)}^{\leftarrow} e^{-ixa_{(j,l)} H_{\pi_j(l)}}, \quad S_K(x)^\dagger = \prod_{(j,l)}^{\rightarrow} e^{ixa_{(j,l)} H_{\pi_j(l)}}. \quad (\text{E2})$$

Similar to the case of lattice Hamiltonians, based on Eqs. (94) and (111), we can expand the Trotter remainder  $V_K(x)$  to the following form:

$$V_K(x) = I + F_L(x) + F_{K,\text{res}}(x), \quad (\text{E3})$$

where

$$F_L(x) := \sum_{s=K}^{2K} C_s \frac{x^{s+1}}{(s+1)!},$$

$$F_{K,\text{res}}(x) := \int_0^x d\tau (M_K(\tau) J_L(\tau) + V_K(\tau) J_{K,\text{res},2K}(\tau)). \quad (\text{E4})$$

Here,  $F_L(x)$  indicates the leading-order terms to be compensated by LCU formula,  $F_{K,\text{res}}(x) = \mathcal{O}(x^{2K+2})$  is the high-order part. In practice, we remove the high-order part with order  $s > 2K + 1$  and implement only the leading-order terms

$$\tilde{V}_K^{(nc)}(x) = I + F_L(x) = I + \sum_{s=K+1}^{2K+1} C_{s-1} \frac{x^s}{s!}$$

$$= \sqrt{1 + (\eta_\Sigma^{(nc)})^2} \sum_{s=K+1}^{2K+1} \frac{\eta_s^{(nc)}}{\eta_\Sigma^{(nc)}} R_{2,s}^{(nc)}(\eta_\Sigma), \quad (\text{E5})$$

where  $\eta_{\Sigma}^{(nc)} := \sum_{s=K+1}^{2K+1} \|C_{s-1}\|_1 x^s / s!$ . The explicit form of  $R_{2,s}^{(nc)}(\eta_{\Sigma})$  can be obtained by the definitions in Eqs. (109), (111) and the Pauli operator decomposition based on the nested-commutator form in Eq. (B12).

We are going to finish the following tasks:

- (1) (Appendix E 1) Derive the explicit formulas for the leading-order expansion terms  $C_s$  with  $s = K, \dots, 2K$ .
- (2) (Appendix E 2) Prove the 1-norm  $\mu_K^{(nc)}(x)$  and error bound  $\varepsilon_K^{(nc)}(x)$  of the LCU formula  $\tilde{V}_K^{(nc)}(x)$  in Eq. (E5).

## 1. Derivation of nested-commutator form for general Hamiltonians

Based on the Trotter formulas in Eq. (E2), we expand  $R_K(x)$  in Eq. (92) as follows:

$$R_K(x) = i \sum_{\gamma=1}^{\Upsilon} \prod_{\gamma'=\gamma-1}^1 e^{ix_{\gamma'} H_{\gamma'}} (a_{\gamma} H_{\gamma}).$$

$$\prod_{\gamma'=\Upsilon}^{\gamma+1} e^{ix_{\gamma'} H_{\gamma'}} - iH \prod_{\gamma'=\Upsilon}^1 e^{ix_{\gamma'} H_{\gamma'}}, \quad (\text{E6})$$

$J_K(x)$  in Eq. (95) can then be written as,

$$J_K(x) = i \sum_{\gamma=1}^{\Upsilon} \prod_{\gamma'=\gamma+1}^{\Upsilon} e^{-ix_{\gamma'} \text{ad}_{H_{\gamma'}}} (a_{\gamma} H_{\gamma}) - i \prod_{\gamma'=1}^{\Upsilon} e^{-ix_{\gamma'} \text{ad}_{H_{\gamma'}}} H. \quad (\text{E7})$$

The derivative of  $J_K(x)$  is

$$J_K^{(s)}(x) = i \sum_{\gamma=1}^{\Upsilon} \sum_{\substack{m_{\gamma+1} \dots m_{\Upsilon} \\ \sum_{v=\gamma+1}^{\Upsilon} m_v = s}} \binom{s}{m_{\gamma+1}, \dots, m_{\Upsilon}} \left( \prod_{\gamma'=\gamma+1}^{\Upsilon} (-ia_{\gamma'} \text{ad}_{H_{\gamma'}})^{m_{\gamma'}} e^{-ix_{\gamma'} \text{ad}_{H_{\gamma'}}} \right) (a_{\gamma} H_{\gamma})$$

$$- i \sum_{\substack{m_1 \dots m_{\Upsilon} \\ \sum_{v=1}^{\Upsilon} m_v = s}} \binom{s}{m_1, \dots, m_{\Upsilon}} \left( \prod_{\gamma'=1}^{\Upsilon} (-ia_{\gamma'} \text{ad}_{H_{\gamma'}})^{m_{\gamma'}} e^{-ix_{\gamma'} \text{ad}_{H_{\gamma'}}} \right) H. \quad (\text{E8})$$

Using the operator-valued Taylor-series expansion in Eq. (121), we have

$$C_s = i \sum_{\gamma=1}^{\Upsilon} \sum_{\substack{\sum_{(j',l')} m_{(j',l')} = s \\ (j',l') > (j,l)}} \binom{s}{\{m_{(j',l')}\}} \left( \overleftarrow{\prod}_{(j',l') > (j,l)} (-ia_{(j',l')} \text{ad}_{H_{\pi_{j'}(l')}})^{m_{(j',l')}} \right) (a_{(j,l)} H_{\pi_j(l)})$$

$$- i \sum_{\sum_{(j',l')} m_{(j',l')} = s} \binom{s}{\{m_{(j',l')}\}} \left( \overleftarrow{\prod}_{(j',l')} (-ia_{(j',l')} \text{ad}_{H_{\pi_{j'}(l')}})^{m_{(j',l')}} \right) H \quad (\text{E9})$$

which is the nested-commutator form. Here,  $\{m_{(j',l')}\}$  are a group of integers whose summation is  $s$ . Their corresponding multinomial coefficient is given by

$$\binom{s}{\{m_{(j',l')}\}} := \frac{s!}{\prod_{(j',l')} m_{(j',l')}!}. \quad (\text{E10})$$

If we define

$$A_{(j,l)} = A_{\gamma} = -ia_{\gamma} H_{\gamma} = -ia_{(j,l)} H_{\pi_j(l)}, \quad (\text{E11})$$

we can simplify Eq. (E9) as

$$C_s = - \sum_{\gamma=1}^{\Upsilon} \sum_{\substack{m_{\gamma+1} \dots m_{\Upsilon} \\ \sum_{v=\gamma+1}^{\Upsilon} m_v = s}} \binom{s}{m_{\gamma+1}, \dots, m_{\Upsilon}} \prod_{\gamma'=\gamma+1}^{\Upsilon} \text{ad}_{A_{\gamma'}}^{m_{\gamma'}} A_{\gamma}$$

$$+ \sum_{\substack{m_1 \dots m_{\Upsilon} \\ \sum_{v=1}^{\Upsilon} m_v = s}} \binom{s}{m_1, \dots, m_{\Upsilon}} \prod_{\gamma'=1}^{\Upsilon} \text{ad}_{A_{\gamma'}}^{m_{\gamma'}} (-iH), \quad (\text{E12})$$

which is the summation of nested commutators. Therefore, one can still pair the leading-order terms  $F_L(x)$  defined in Eq. (E4) with  $I$  to suppress the 1-norm, shown in Eq. (E5).

## 2. Norm bounds for the nested-commutator expansion

Now, we are going to bound the 1-norm  $\mu_K^{(nc)}(x)$  and distance  $\varepsilon_K^{(nc)}(x)$  of the truncated LCU formula  $\tilde{V}_K^{(nc)}(x)$  in Eq. (110) for a general Hamiltonian. We will use the following formula in the derivation.

*Lemma 4 (Theorem 5 in Ref. [20]).* Let  $A_1, A_2, \dots, A_r$  and  $B$  be operators. Then, the conjugation has the expansion,

$$\begin{aligned} e^{\tau \text{ad}_{A_r}} \dots e^{\tau \text{ad}_{A_2}} e^{\tau \text{ad}_{A_1}} B \\ = G_0 + G_1 \tau + \dots + G_{s-1} \frac{\tau^{s-1}}{(s-1)!} + G_{\text{res},s}(\tau). \end{aligned} \quad (\text{E13})$$

Here,  $G_0, G_1, \dots, G_{s-1}$  are operators independent of  $\tau$ . The operator-valued function  $G_{\text{res},s}(\tau)$  is given by

$$\begin{aligned} G_{\text{res},s}(\tau) &:= \sum_{\gamma=1}^r \sum_{\substack{m_1+\dots+m_\gamma=s \\ m_\gamma \neq 0}} e^{\tau \text{ad}_{A_r}} \dots e^{\tau \text{ad}_{A_{\gamma+1}}} \\ &\int_0^\tau d\tau_2 \frac{(\tau - \tau_2)^{m_\gamma-1} \tau^{m_1+\dots+m_{\gamma-1}}}{(m_\gamma-1)! m_{\gamma-1}! \dots m_1!} e^{\tau_2 \text{ad}_{A_\gamma}} \text{ad}_{A_\gamma}^{m_\gamma} \dots \text{ad}_{A_1}^{m_1} B. \end{aligned} \quad (\text{E14})$$

Furthermore, we have the spectral-norm bound,

$$\|G_{\text{res},s}(\tau)\| \leq \alpha_{\text{com}}^{(s)}(A_r, \dots, A_1; B) \frac{|\tau|^s}{s!} e^{2|\tau| \sum_{\gamma=1}^r \|A_\gamma\|}, \quad (\text{E15})$$

for general operators and

$$\|G_{\text{res},s}(\tau)\| \leq \alpha_{\text{com}}^{(s)}(A_r, \dots, A_1; B) \frac{|\tau|^s}{s!}, \quad (\text{E16})$$

when  $A_1, \dots, A_r$  are anti-Hermitian.  $\alpha_{\text{com}}^{(s)}(A_r, \dots, A_1; B)$  is defined in Eq. (E18).

We have the following proposition.

*Proposition 13 (Trotter-LCU formula by nested-commutator compensation for general Hamiltonians).* For  $x > 0$ ,  $\tilde{V}_K(x)$  in Eq. (E5) is a  $(\mu^{(nc)}(x), \varepsilon^{(nc)}(x))$ -LCU formula of

$K$ th-order Trotter remainder  $V_K(x) = U(x)S_K(x)$  with

$$\begin{aligned} \mu^{(nc)} &\leq \sqrt{1 + 4\kappa^2 \left( \sum_{s=K}^{2K} \alpha_{H,1}^{(s)} \frac{x^{s+1}}{(s+1)!} \right)^2}, \\ \varepsilon^{(nc)} &\leq 4\kappa^2 \alpha_H^{(K)} \sum_{s=K}^{2K} \alpha_H^{(s)} \frac{x^{s+K+2}}{s!(K+1)!(s+K+2)} \\ &\quad + 2\kappa \frac{x^{2K+2}}{(2K+2)!} \alpha_H^{(2K+1)}. \end{aligned} \quad (\text{E17})$$

Here, we define the nested-commutator norms to be

$$\begin{aligned} \alpha_H^{(s)} &:= \sum_{l=1}^L \alpha_{\text{com}}^{(s)}(\overleftarrow{\{H_{\pi_j(l')}\}}; H), \\ \alpha_{\text{com}}^{(s)}(A_r, \dots, A_1; B) &:= \\ &\sum_{m_1+\dots+m_r=s} \binom{s}{m_1, \dots, m_r} \|\text{ad}_{A_r}^{m_r} \dots \text{ad}_{A_1}^{m_1} B\|, \end{aligned} \quad (\text{E18})$$

where  $\overleftarrow{\{H_{\pi_j(l')}\}}$  indicates a sequence of  $\Upsilon = \kappa L$  summands with the lexicographical order given by the  $K$ th-order Trotter formula in Eq. (E2).  $\alpha_{H,1}^{(s)}$  is defined similarly by replacing the spectral norm to 1-norm in Eq. (E18).

*Proof.* From Eq. (E12) we can bound the 1-norm and spectral norm of  $C_s$  by

$$\begin{aligned} \|C_s\| &\leq \sum_{\gamma=1}^{\Upsilon} \alpha_{\text{com}}^{(s)}(A_\Upsilon, \dots, A_{\gamma+1}; A_\gamma) \\ &\quad + \alpha_{\text{com}}^{(s)}(A_\Upsilon, \dots, A_1; H), \\ \|C_s\|_1 &\leq \sum_{\gamma=1}^{\Upsilon} \alpha_{\text{com},1}^{(s)}(A_\Upsilon, \dots, A_{\gamma+1}; A_\gamma) \\ &\quad + \alpha_{\text{com},1}^{(s)}(A_\Upsilon, \dots, A_1; H). \end{aligned} \quad (\text{E19})$$

Now, we are going to bound  $\|J_K(x)\|$  and  $\|J_{K,\text{res},2K}(x)\|$ . From Eq. (E7) and using Lemma 4 we have

$$\begin{aligned} J_K(x) &= - \sum_{\gamma=1}^{\Upsilon} \prod_{\gamma'=\gamma+1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (A_\gamma) + \prod_{\gamma'=1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (-iH) \\ &= \sum_{\gamma=1}^{\Upsilon} G_{\text{res},K}^{(\gamma)}(x) + G_{\text{res},K}^{(0)}(x). \end{aligned} \quad (\text{E20})$$

In the second line, we use Lemma 4 to expand all the conjugate matrix exponentials to the following form:

$$\begin{aligned}
\prod_{\gamma'=\gamma+1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (A_{\gamma}) &= G_0^{(\gamma)} + G_1^{(\gamma)} x + \dots \\
&+ G_{K-1}^{(\gamma)} \frac{x^{K-1}}{(K-1)!} + G_{\text{res},K}^{(\gamma)}, \quad \gamma = 1, 2, \dots, \Upsilon, \\
\prod_{\gamma'=1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (-iH) &= G_0^{(0)} + G_1^{(0)} x + \dots \\
&+ G_{K-1}^{(0)} \frac{x^{K-1}}{(K-1)!} + G_{\text{res},K}^{(0)}. \tag{E21}
\end{aligned}$$

The low-order terms  $G_s^{(\gamma)}$  with  $s = 0, 1, \dots, K-1$  in Eq. (E20) cancel out due to the order condition in Proposition 5. Therefore, we have

$$\begin{aligned}
\|J_K(x)\| &\leq \sum_{\gamma=1}^{\Upsilon} \|G_{\text{res},K}^{(\gamma)}(x)\| + \|G_{\text{res},K}^{(0)}(x)\| \\
&\leq \frac{x^K}{K!} \left( \sum_{\gamma=1}^{\Upsilon} \alpha_{\text{com}}^{(K)}(A_{\Upsilon}, \dots, A_{\gamma+1}; A_{\gamma}) \right. \\
&\quad \left. + \alpha_{\text{com}}^{(K)}(A_{\Upsilon}, \dots, A_1; H) \right) \\
&\leq 2 \frac{x^K}{K!} \sum_{\gamma=1}^{\Upsilon} \alpha_{\text{com}}^{(K)}(A_{\Upsilon}, \dots, A_1; H_{\gamma}) \\
&= 2\kappa \frac{x^K}{K!} \sum_{l=1}^L \alpha_{\text{com}}^{(K)}(\overleftarrow{\{a_{(j',l')} H_{\pi_{j'}(l')}\}}; H_l) \\
&\leq 2\kappa \frac{x^K}{K!} \sum_{l=1}^L \alpha_{\text{com}}^{(K)}(\overleftarrow{\{H_{\pi_{j'}(l')}\}}; H_l) =: 2\kappa \frac{x^K}{K!} \alpha_H^{(K)}. \tag{E22}
\end{aligned}$$

Following the same way, we can bound  $\|C_s\|$  and  $\|C_s\|_1$  in Eq. (E19) as

$$\begin{aligned}
\|C_s\| &\leq 2\kappa \alpha_H^{(s)}, \\
\|C_s\|_1 &\leq 2\kappa \alpha_{H,1}^{(s)}. \tag{E23}
\end{aligned}$$

Now, we are going to bound  $\|J_{K,\text{res},2K}(x)\|$ . Similar to Eq. (E20), we expand all the conjugate matrix exponentials to  $2K$ th-order,

$$\begin{aligned}
J_K(x) &= \sum_{\gamma=1}^{\Upsilon} \prod_{\gamma'=\gamma+1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (A_{\gamma}) + \prod_{\gamma'=1}^{\Upsilon} e^{\text{ad}_{A_{\gamma'}}} (-iH) \\
&= \sum_{s=K}^{2K} C_s \frac{x^s}{s!} + \sum_{\gamma=1}^{\Upsilon} G_{\text{res},2K+1}^{(\gamma)}(x) + G_{\text{res},2K+1}^{(0)}(x), \tag{E24}
\end{aligned}$$

in the second line, we use the expansion of  $J_K(x)$  in Eq. (104). Based on Eq. (E22) we then have

$$\begin{aligned}
J_{K,\text{res},2K}(x) &= \sum_{\gamma=1}^{\Upsilon} G_{\text{res},2K+1}^{(\gamma)}(x) + G_{\text{res},2K+1}^{(0)}(x) \\
\Rightarrow \|J_{K,\text{res},2K}(x)\| &\leq \sum_{\gamma=1}^{\Upsilon} \|G_{\text{res},2K+1}^{(\gamma)}(x)\| + \|G_{\text{res},2K+1}^{(0)}(x)\| \\
&\leq 2\kappa \frac{x^{2K+1}}{(2K+1)!} \alpha_H^{(2K+1)}. \tag{E25}
\end{aligned}$$

We omit the derivation to the second line since it is the same as the one in Eq. (E22).

Then we can bound  $\|M_K(x)\|$  by

$$\|M_K(x)\| \leq \int_0^x d\tau \|J_K(\tau)\| \leq 2\kappa \frac{x^{K+1}}{(K+1)!} \alpha_H^{(K)}. \tag{E26}$$

Finally, by applying Proposition 6 and using Eq. (E23), we can bound the 1-norm  $\mu_K^{(nc)}(x)$  as follows:

$$\mu_K^{(nc)}(x) \leq \sqrt{1 + 4\kappa^2 \left( \sum_{s=K}^{2K} \alpha_{H,1}^{(s)} \frac{x^{s+1}}{(s+1)!} \right)^2}, \tag{E27}$$

while the accuracy  $\varepsilon_K^{(nc)}(x)$  can be bounded using Eqs. (E19), (E25), and (E26),

$$\begin{aligned}
\varepsilon_K^{(nc)}(x) &= \|F_{K,\text{res}}(x)\| \leq \int_0^x d\tau (\|M_K(\tau)\| \|J_L(\tau)\| + \|J_{K,\text{res},2K}(\tau)\|) \\
&\leq \int_0^x d\tau \left( 4\kappa^2 \frac{\tau^{K+1}}{(K+1)!} \alpha_H^{(K)} \sum_{s=K}^{2K} \alpha_H^{(s)} \frac{\tau^s}{s!} + 2\kappa \frac{\tau^{2K+1}}{(2K+1)!} \alpha_H^{(2K+1)} \right) \\
&\leq 4\kappa^2 \alpha_H^{(K)} \sum_{s=K}^{2K} \alpha_H^{(s)} \frac{x^{s+K+2}}{s!(K+1)!(s+K+2)} + 2\kappa \frac{x^{2K+2}}{(2K+2)!} \alpha_H^{(2K+1)}. \tag{E28}
\end{aligned}$$

■

From Proposition 13 we can see that, to characterize the performance of the Trotter-LCU algorithm, we only need to estimate the values of  $\alpha_H^{(s)}$  and  $\alpha_{H,1}^{(s)}$  for a given Hamiltonian. We can further simplify the form of  $\alpha_H^{(s)}$  by the following upper bound [20]:

$$\begin{aligned} \alpha_H^{(s)} &\leq \kappa^s \sum_{l_{s+1}=1}^L \cdots \sum_{l_2=1}^L \|[H_{l_{s+1}}, \dots [H_{l_2}, H_l]] \cdots \| \\ &=: \kappa^s \tilde{\alpha}_{\text{com}}(H), \end{aligned} \quad (\text{E29})$$

which is because the commutator terms in the left-hand side must be of the form on the right. Moreover, if we fix one term  $\|[H_{l_{s+1}}, \dots [H_{l_2}, H_l]] \cdots \|$  on the right-hand side, we can find at most  $\kappa^s$  times of this term on the left-hand side.

## APPENDIX F: SOME USEFUL FORMULAS IN THE PROOF

*Lemma 5 (Tail bound for the Poisson distribution (Theorem 1 in Ref. [41])).* Suppose  $\hat{X}$  is a random variable with Poisson distribution so that  $\Pr(\hat{X} = s) = \text{Poi}(s; x) = e^{-x} x^s / s!$ , where  $x > 0$  is the expectation value. Then, for any  $\epsilon > 0$ , we have,

$$\Pr(\hat{X} \geq x + \epsilon) \leq e^{-\frac{\epsilon^2}{2x} h(\frac{\epsilon}{x})}, \quad (\text{F1})$$

and, for any  $0 < \epsilon < x$ ,

$$\Pr(\hat{X} \leq x - \epsilon) \leq e^{-\frac{\epsilon^2}{2x} h(\frac{\epsilon}{x})}. \quad (\text{F2})$$

Here,  $h(u) := 2 \frac{(1+u) \ln(1+u) - u}{u^2}$  for  $u \geq -1$ .

From Lemma 5 we have the following corollaries.

*Corollary 1.* For  $x > 0$  and positive integer  $k$  such that  $x < k + 1$ , we have

$$\sum_{s=k+1}^{\infty} \frac{x^s}{s!} \leq \left( \frac{ex}{k+1} \right)^{k+1}. \quad (\text{F3})$$

*Proof.* We set  $\epsilon = (k + 1) - x$ . From Lemma 5 we have

$$\begin{aligned} \Pr(\hat{X} \geq k + 1) &\leq \exp\left(-\frac{(k+1-x)^2}{2x} h\left(\frac{k+1-x}{x}\right)\right) \\ &= e^{-x} \left( \frac{ex}{k+1} \right)^{k+1}. \end{aligned} \quad (\text{F4})$$

Therefore,

$$\begin{aligned} \Pr(\hat{X} \geq k + 1) &= e^{-x} \sum_{s=k+1}^{\infty} \frac{x^s}{s!} \leq e^{-x} \left( \frac{ex}{k+1} \right)^{k+1} \\ \Rightarrow \sum_{s=k+1}^{\infty} \frac{x^s}{s!} &\leq \left( \frac{ex}{k+1} \right)^{k+1}. \end{aligned} \quad (\text{F5})$$

■

*Corollary 2.* For  $x > 0$  and positive integer  $k$  such that  $x < k + 1$ , we have

$$e^x - \sum_{s=1}^k \frac{x^s}{s!} \leq e^{ex^{k+1}}. \quad (\text{F6})$$

*Proof.* When  $x < k + 1$ , from Corollary 1 we have

$$\begin{aligned} e^{-x} \sum_{s=k+1}^{\infty} \frac{x^s}{s!} &\leq e^{-x} \left( \frac{ex}{k+1} \right)^{k+1} \\ \Rightarrow 1 - e^{-x} - e^{-x} \sum_{s=1}^k \frac{x^s}{s!} &\leq e^{-x} \left( \frac{ex}{k+1} \right)^{k+1} \\ \Rightarrow e^x - \sum_{s=1}^k \frac{x^s}{s!} &\leq 1 + \left( \frac{ex}{k+1} \right)^{k+1} \\ &\leq 1 + \frac{(ex)^{k+1}}{(k+1)!} \leq e^{ex^{k+1}}. \end{aligned} \quad (\text{F7})$$

■

*Lemma 6 (Proposition 9 in Ref. [34]).* For any  $\beta > 0$ ,  $1 > \epsilon > 0$ , we have  $(e\beta/s)^s \leq \epsilon$ , for all  $s \geq f(\beta, \epsilon) := \ln(1/\epsilon) / W_0(1/(e\beta) \ln(1/\epsilon))$ . Here,  $W_0(y)$  is the principle branch of the Lambert  $W$  function.

*Lemma 7 (Theorem 2.7 in Ref. [51]).* When  $y \geq e$  we have

$$\begin{aligned} \ln(y) - \ln \ln(y) + \frac{1}{2} \frac{\ln \ln(y)}{\ln(y)} &\leq W_0(y) \\ &\leq \ln(y) - \ln \ln(y) + \frac{e}{e-1} \frac{\ln \ln(y)}{\ln(y)}. \end{aligned} \quad (\text{F8})$$

## APPENDIX G: ADDITIONAL NUMERICAL RESULTS

In this section, we provide more numerical results by comparing the gate costs of Trotter-LCU algorithms with other typical algorithms, especially the Trotter algorithm and ‘‘post-Trotter’’ algorithm with best performance, i.e., the fourth-order Trotter algorithm and the quantum signal processing (QSP) algorithm. We will mainly consider

two different scenarios—generic  $L$ -sparse Hamiltonians and lattice Hamiltonians. In the former case, the previously known best result is given by the QSP algorithm [28]; in the latter case, since we can take advantage of the spacial locality and commutator information, the fourth-order Trotter algorithm is known to have the best performance [13,49].

For the Trotter algorithms and Trotter-LCU algorithms, we first compile the circuit to CNOT gates, single-qubit Clifford gates and non-Clifford  $Z$ -axis rotation gate  $R_z(\theta) = e^{i\theta Z}$ . On the other hand, the circuit compilation for the QSP algorithm is more complicated: we need to decompose the state-preparation oracles and the select- $H$  gates. We follow the QROM design in Ref. [52] based on a sawtooth structure. The detailed gate resource analysis can be found in a companion work [53]. Based on the qROM design, we decompose all the state preparation oracles and the select- $H$  gates to Toffoli gates, which can be further decomposed to CNOT gates, single-qubit Clifford gates, and  $T$  gates.

For a fair comparison between the gate costs of Trotter, Trotter-LCU, and QSP algorithms, we need to further compile the  $Z$ -axis rotation gate  $R_z(\theta)$  to  $T$  gate. We follow the gate compilation work in Ref. [54], where the expected  $T$ -gate number to compile  $R_z(\theta)$  with random  $\theta$  is about

$$c_T = 1.149 \log_2(1/\epsilon) + 9.2, \quad (\text{G1})$$

where  $\epsilon$  is the compilation accuracy. We set the accuracy  $\epsilon = 10^{-15}$  in the later resource estimation. In this case,  $c_T \approx 66$ .

### 1. Generic $L$ -sparse Hamiltonians

For the generic  $L$ -sparse Hamiltonian, we choose the two-local Hamiltonian

$$H = \sum_{ij} J_{ij} X_i X_j + \sum_i Z_i, \quad (\text{G2})$$

with  $J_{ij} = 1$  as an example, in which case we ignore the commutator information between different Hamiltonian summands. This simple model works as a representative of many generic Hamiltonians where the commutator information is not helpful or too complicated to count on, e.g., quantum chemistry Hamiltonian for the molecules. Without the commutator information, the former best Hamiltonian simulation algorithm is QSP [28].

In Figs. 13 and 14, we estimate and compare the CNOT and  $T$ -gate counts for the PTSC algorithms, QSP, and fourth-order Trotter algorithm with respect to the evolution time. We choose fourth-order Trotter algorithm with random permutation since it performs the best over all the Trotter algorithms. The gate counting method for fourth-order Trotter with random permutation is based on the analytical bounds in Ref. [12]. For the QSP algorithm, we

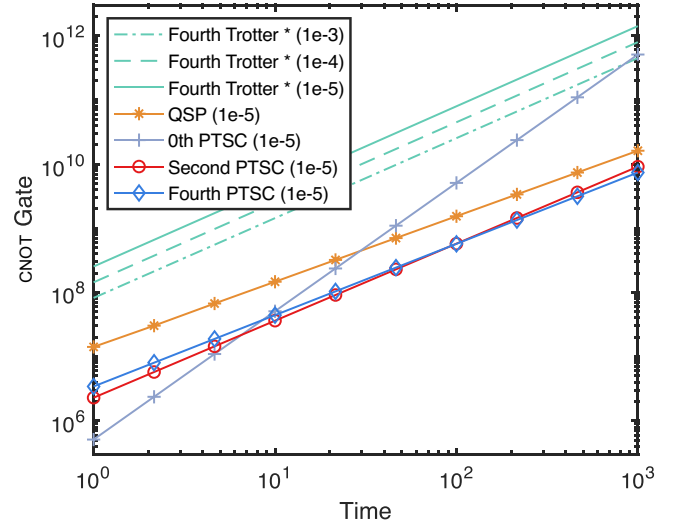


FIG. 13. CNOT-gate number estimation for simulating the generic  $L$ -sparse Hamiltonian with an increasing time. The system size is set as  $n = 20$ . The simulation is exemplified with the two-local Hamiltonian in Eq. (G2).

estimate the gate count based on the qROM construction in Ref. [52]. Only the high-accuracy results  $\epsilon = 10^{-5}$  for our method and quantum signal processing are presented since our method and quantum signal processing has a logarithmic dependence and hence are less prone to the accuracy while the Trotter formulae has polynomial dependence on the accuracy.

From Figs. 13 and 14 we can see that, the PTSC algorithm owns a lower CNOT-gate cost than the QSP algorithm because it does not need the qROM for classical

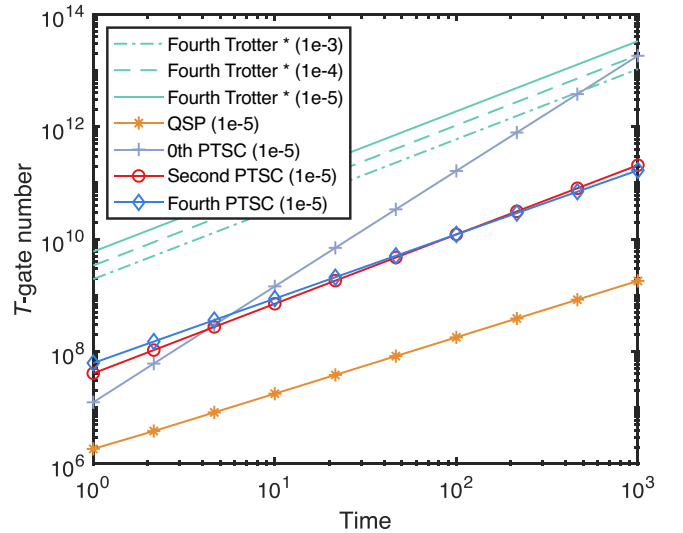


FIG. 14.  $T$ -gate number estimation for simulating the generic  $L$ -sparse Hamiltonian with an increasing time. The system size is set as  $n = 20$ . The simulation is exemplified with the two-local Hamiltonian in Eq. (G2).

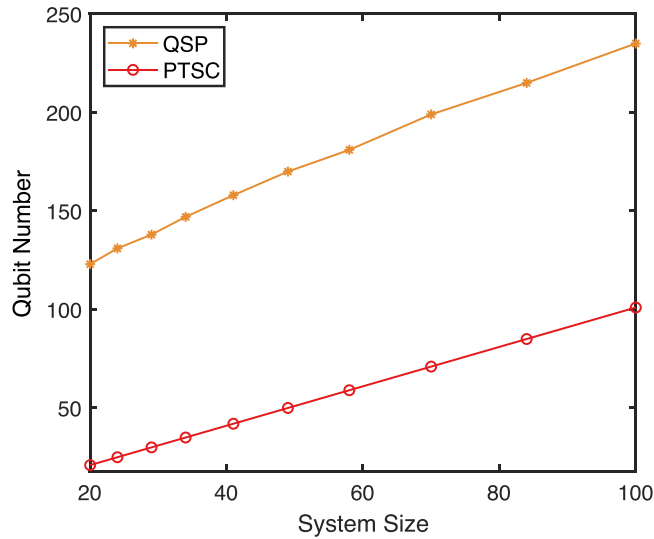


FIG. 15. Qubit number required for simulating the generic  $L$ -sparse Hamiltonian with an increasing system size. The simulation is exemplified with the 2-local Hamiltonian in Eq. (G2).

data loading. On the other hand, the PTSC algorithms have a larger  $T$ -gate cost than QSP algorithm. This is mainly due to the compilation cost of arbitrary  $Z$ -axis rotation gate  $R_z(\theta)$  to  $T$  gates: for each  $R_z(\theta)$  gate with a random phase  $\theta$ , we need  $c_T = 66$   $T$  gates on average to compile it to the accuracy of  $\varepsilon = 10^{-15}$ . In the Trotter or Trotter-LCU algorithms, the non-Clifford gate cost mainly originates from the  $R_z(\theta)$  gates: in each segment, there are roughly  $\kappa_K L$   $R_z(\theta)$  gates, leading to about  $\kappa_K c_T L$  compiled  $T$  gates. As a comparison, there are only few  $R_z(\theta)$  gates in QSP algorithm used for the phase interaction procedure to realize the Jacobi-Anger polynomials. The major non-Clifford gate cost lies in the compilation of state-preparation oracles and the select- $H$  gates, each of which can be realized by about  $4L$   $T$  gates based on the qROM design in Ref. [52].

In Fig. 15 we compare the qubit number required to implement the QSP and the PTSC algorithms, based on the two-local Hamiltonian model. We can see a clear advantage of the spacial resource cost of PTSC to QSP algorithm.

## 2. Lattice Hamiltonians

Now, we consider the case of lattice Hamiltonians. We consider the Heisenberg model,  $H = J \sum_i \vec{\sigma}_i \vec{\sigma}_{i+1} + h \sum_i Z_i$ , where  $\vec{\sigma}_i := (X_i, Y_i, Z_i)$  is the vector of Pauli operators on the  $i$ th qubit and  $J = h = 1$ .

We compare the CNOT and  $T$ -gate number of the second-order NCC algorithm, QSP and fourth-order Trotter algorithm in Figs. 16 and 17, respectively. The fourth-order Trotter error analysis is based on the nested-commutator bound (Proposition M.1 in Ref. [20]), which is currently the tightest Trotter error bound. The performance of our

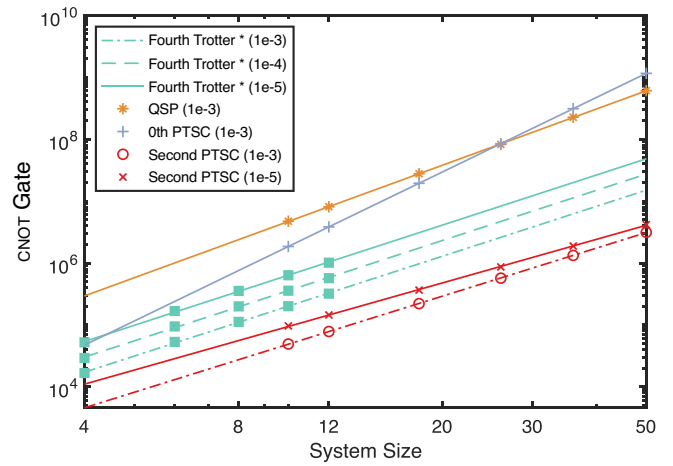


FIG. 16. CNOT-gate number estimation for simulating the Heisenberg Hamiltonian using the nested commutator bound with an increasing system size  $n$ . The simulation time is set as  $t = n$ . The fourth-order Trotter formula uses the nested-commutator bound shown in Proposition M.1 in Ref. [20]. Our result is based on second-order NCC algorithm with the analysis in Sec. B.

second-order NCC algorithm is analyzed based on the detailed analysis in Sec. B. We explicitly calculate the 1-norm of the LCU formula and use the analytical bound for the accuracy analysis. Here, we do not introduce the empirical analysis for a fair comparison. Since the explicit evaluation of our fourth-order NCC algorithm is complicated, we mainly present the results for our second-order algorithm.

As addressed in Refs. [13,20], fourth-order Trotter formula shows a near-optimal scaling with respect to the system size for the lattice model, which is clearly shown in Fig. 16. From Fig. 16 we can see that, our second-order

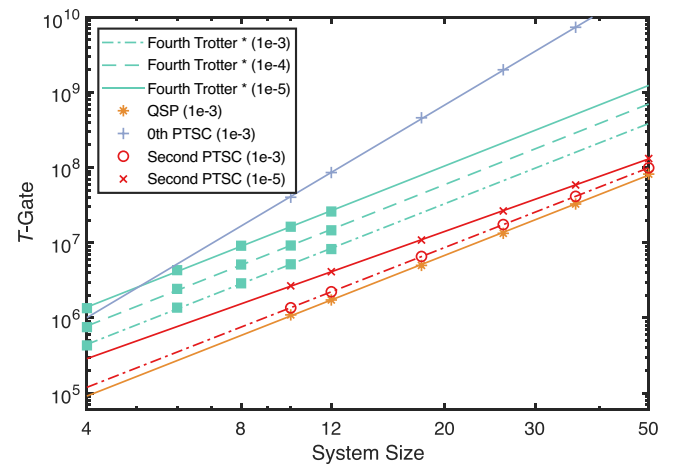


FIG. 17.  $T$ -gate number estimation for simulating the Heisenberg Hamiltonian using the nested-commutator bound with an increasing system size  $n$ . The stage is set up the same as that in Fig. 16.

NCC algorithm shows an advantage over the fourth-order Trotter algorithm. Furthermore, the  $n$  and  $t$  scaling of our second-order NCC algorithm is similar to the fourth-order Trotter algorithm, which is near optimal.

## APPENDIX H: COHERENT IMPLEMENTATION

In the main text, we primarily focus on the random-sampling implementation of the Trotter-LCU algorithms due to its simplicity. When the block encoding of an LCU formula of  $V$  is feasible on a fault-tolerant quantum computer, we can also consider the coherent implementation of the Trotter-LCU algorithms.

The gate complexity of the coherent implementation of the Trotter-LCU algorithm is determined by the segment number  $\nu$ , the Trotter order  $K$ , the number of elementary unitaries  $\Gamma$  and the gate complexity of each elementary unitary in the LCU formula. The value  $\Gamma$  is related to the specific compensation method we use, which is usually proportional to  $L$  and the truncation order  $s_c$ . The gate complexity is

$$N_K = \mathcal{O}(\nu(\kappa_K L + \Gamma)). \quad (\text{H1})$$

To estimate the segment number  $\nu$ , we first find a proper evolution time  $x$  such that the 1-norm  $\mu_x \leq 2$ . The segment number is then  $\nu = t/x$ .

For the PTSC formula, based on the 1-norm bound in Theorem 3, we find that to ensure  $\mu_x \leq 2$ , it is sufficient to set  $x = 1/2\lambda(\ln 2/(e + c_k))^{1/(2K+2)} = \mathcal{O}(1/\lambda)$ . As a result, the number of segment  $\nu = t/x = \mathcal{O}(\lambda t)$ . From Eq. (H1), the overall gate complexity of the algorithm is

$$N_K^{(p)} = \mathcal{O}(\nu(\kappa_K L + \Gamma)) = \mathcal{O}\left(\lambda t L \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)}\right), \quad (\text{H2})$$

which is the same as the case when no Trotter formula is applied [25].

For the NCC formula, based on the 1-norm bound in Theorem 4, we find that to ensure  $\mu_x \leq 2$ , we need to set

$$x = \frac{1}{\beta \Lambda_1} \left( \frac{2 \ln 2 (K!)^2}{(n\kappa)^2} \right)^{\frac{1}{2K+2}} \Rightarrow \nu = \mathcal{O}(n^{\frac{1}{K+1}} t).$$

To achieve an overall simulation accuracy  $\varepsilon$ , we need

$$\nu \varepsilon_K^{(nc)}(x) \leq \varepsilon \Rightarrow \nu = \mathcal{O}(n^{\frac{2}{2K+1}} t^{1+\frac{1}{2K+1}} \varepsilon^{-\frac{1}{2K+1}}). \quad (\text{H3})$$

Therefore, it suffice to choose  $\nu$  to be

$$\nu = \mathcal{O}(n^{\frac{2}{2K+1}} t^{1+\frac{1}{2K+1}} \varepsilon^{-\frac{1}{2K+1}}). \quad (\text{H4})$$

In each segment, we need to implement  $K$ th-order Trotter formula and the  $(K+1)$ th- to  $(2K+1)$ th-order LCU

formula. The number of elementary unitaries is  $\mathcal{O}(n)$ . Therefore, the overall gate complexity is

$$N_K^{(nc)} = \mathcal{O}(n^{1+\frac{2}{2K+1}} t^{1+\frac{1}{2K+1}} \varepsilon^{-\frac{1}{2K+1}}). \quad (\text{H5})$$

Although it does not achieve the logarithmic accuracy dependence seen in standard post-Trotter algorithms [25, 27], it has the unique advantage of a system-size dependence of  $\mathcal{O}(n)$ , which could be advantageous in large-scale coherent Hamiltonian simulations.

- 
- [1] R. P. Feynman *et al.*, Simulating physics with computers, *Int. J. Theor. Phys* **21**, 467 (1982).
  - [2] D. S. Abrams and S. Lloyd, Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, *Phys. Rev. Lett.* **83**, 5162 (1999).
  - [3] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, *Science* **309**, 1704 (2005).
  - [4] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
  - [5] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
  - [6] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
  - [7] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
  - [8] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, *Phys. Lett. A* **146**, 319 (1990).
  - [9] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, *J. Math. Phys.* **32**, 400 (1991).
  - [10] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, *Commun. Math. Phys.* **270**, 359 (2007).
  - [11] E. Campbell, Random compiler for fast Hamiltonian simulation, *Phys. Rev. Lett.* **123**, 070503 (2019).
  - [12] A. M. Childs, A. Ostrander, and Y. Su, Faster quantum simulation by randomization, *Quantum* **3**, 182 (2019).
  - [13] A. M. Childs and Y. Su, Nearly optimal lattice simulation by product formulas, *Phys. Rev. Lett.* **123**, 050503 (2019).
  - [14] S. Endo, Q. Zhao, Y. Li, S. Benjamin, and X. Yuan, Mitigating algorithmic errors in a Hamiltonian simulation, *Phys. Rev. A* **99**, 012334 (2019).
  - [15] M. Heyl, P. Hauke, and P. Zoller, Quantum localization bounds Trotter errors in digital quantum simulation, *Sci. Adv.* **5**, eaau8342 (2019).
  - [16] C.-F. Chen, H.-Y. Huang, R. Kueng, and J. A. Tropp, Concentration for random product formulas, *PRX Quantum* **2**, 040305 (2021).
  - [17] B. Şahinoğlu and R. D. Somma, Hamiltonian simulation in the low-energy subspace, *Npj Quantum Inf.* **7**, 1 (2021).

- [18] Y. Su, H.-Y. Huang, and E. T. Campbell, Nearly tight Trotterization of interacting electrons, *Quantum* **5**, 495 (2021).
- [19] M. C. Tran, S.-K. Chu, Y. Su, A. M. Childs, and A. V. Gorshkov, Destructive error interference in product-formula lattice simulation, *Phys. Rev. Lett.* **124**, 220502 (2020).
- [20] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, *Phys. Rev. X* **11**, 011020 (2021).
- [21] D. Layden, [arXiv:2107.08032](https://arxiv.org/abs/2107.08032).
- [22] Q. Zhao, Y. Zhou, A. F. Shaw, T. Li, and A. M. Childs, Hamiltonian simulation with random inputs, *Phys. Rev. Lett.* **129**, 270502 (2022).
- [23] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, *Proc. Natl. Acad. Sci.* **114**, 7555 (2017).
- [24] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing (Association for Computing Machinery, New York, NY, USA, 2014)*, pp. 283–292.
- [25] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, *Phys. Rev. Lett.* **114**, 090502 (2015).
- [26] D. W. Berry, A. M. Childs, and R. Kothari, *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (IEEE, 2015)*, pp. 792–809.
- [27] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [28] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [29] G. H. Low, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (Association for Computing Machinery, New York, NY, USA, 2019)*, pp. 491–502.
- [30] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, *Quantum Inf. Comput.* **12**, 0901 (2012).
- [31] G. L. Long, Duality quantum computing and duality quantum information processing, *Int. J. Theor. Phys.* **50**, 1305 (2011).
- [32] L. Lin and Y. Tong, Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers, *PRX Quantum* **3**, 010318 (2022).
- [33] Y. Yang, B.-N. Lu, and Y. Li, Accelerated quantum Monte Carlo with mitigated error on noisy quantum computer, *PRX Quantum* **2**, 040361 (2021).
- [34] K. Wan, M. Berta, and E. T. Campbell, Randomized quantum algorithm for statistical phase estimation, *Phys. Rev. Lett.* **129**, 030503 (2022).
- [35] P. K. Faehrmann, M. Steudtner, R. Kueng, M. Kieferova, and J. Eisert, Randomizing multi-product formulas for Hamiltonian simulation, *Quantum* **6**, 806 (2022).
- [36] L. Lin and Y. Tong, Near-optimal ground state preparation, *Quantum* **4**, 372 (2020).
- [37] P. Zeng, J. Sun, and X. Yuan, Universal quantum algorithmic cooling on a quantum computer, [arXiv:2109.15304](https://arxiv.org/abs/2109.15304).
- [38] R. Zhang, G. Wang, and P. Johnson, Computing ground state properties with early fault-tolerant quantum computers, *Quantum* **6**, 761 (2022).
- [39] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026).
- [40] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, *Quantum* **3**, 128 (2019).
- [41] C. Canonne, *A short note on Poisson tail bounds* (2016), <http://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf>.
- [42] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-depth quantum simulation of materials, *Phys. Rev. X* **8**, 011044 (2018).
- [43] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, *PRX Quantum* **2**, 030305 (2021).
- [44] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, *Quantum* **3**, 208 (2019).
- [45] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, *Phys. Rev. Res.* **3**, 033055 (2021).
- [46] M. Hagan and N. Wiebe, Composite quantum simulations, *Quantum* **7**, 1181 (2023).
- [47] C.-H. Cho, D. W. Berry, and M.-H. Hsieh, Doubling the order of approximation via the randomized product formula, *Phys. Rev. A* **109**, 062431 (2024).
- [48] Q. Zhao and X. Yuan, *Quantum* **5**, 534 (2021).
- [49] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proc. Natl. Acad. Sci.* **115**, 9456 (2018).
- [50] K. Bringmann and K. Panagiotou, in *Automata, Languages, and Programming*, edited by A. Czumaj, K. Mehlhorn, A. Pitts, and R. Wattenhofer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), p. 133.
- [51] A. Hoorfar and M. Hassani, *J. Inequal. Pure Appl. Math* **9**, 5 (2008).
- [52] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, *Phys. Rev. X* **8**, 041015 (2018).
- [53] J. Sun, P. Zeng, T. Gur, and M. S. Kim, High-precision and low-depth eigenstate property estimation: theory and resource estimation, [arXiv:2406.04307](https://arxiv.org/abs/2406.04307).
- [54] A. Bocharov, M. Roetteler, and K. M. Svore, Efficient synthesis of universal repeat-until-success quantum circuits, *Phys. Rev. Lett.* **114**, 080502 (2015).