High-Order Randomized Compiler for Hamiltonian Simulation

Kouhei Nakaji,[1,2,3,*] Mohsen Bagherimehrab,[1,4,†] and Alán Aspuru-Guzik[1,4,5,6,7,8]

[1] *Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Ontario, Canada M5S 3H6*

[2] *Research Center for Emerging Computing Technologies, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan*

[3] *Quantum Computing Center, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan*

[4] *Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 2E4*

[5] *Vector Institute for Artificial Intelligence, Toronto, Ontario, Canada M5G 1M1*

[6] *Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, Ontario, Canada M5S 3E5*

[7] *Department of Materials Science & Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3E4*

[8] *Canadian Institute for Advanced Research (CIFAR), Toronto, Ontario, Canada M5S 1M1*

(Received 3 May 2023; revised 24 January 2024; accepted 2 April 2024; published 8 May 2024)

Hamiltonian simulation is known to be one of the fundamental building blocks of a variety of quantum algorithms such as its most immediate application, that of simulating many-body systems to extract their physical properties. In this work, we present qSWIFT, a high-order randomized algorithm for Hamiltonian simulation. In qSWIFT, the required number of gates for a given precision is independent of the number of terms in the Hamiltonian, while the systematic error is exponentially reduced with regard to the order parameter. In this respect, our qSWIFT is a higher-order counterpart of the previously proposed quantum stochastic drift protocol (qDRIFT), the number of gates in which scales linearly with the inverse of the precision required. We construct the qSWIFT channel and establish a rigorous bound for the systematic error quantified by the diamond norm. qSWIFT provides an algorithm to estimate given physical quantities by using a system with one ancilla qubit, which is as simple as other product-formula-based approaches such as regular Trotter-Suzuki decompositions and qDRIFT. Our numerical experiment reveals that the required number of gates in qSWIFT is significantly reduced compared to qDRIFT. In particular, the advantage is significant for problems where high precision is required; e.g., to achieve a systematic relative propagation error of 10^{-6}, the required number of gates in third-order qSWIFT is 1000 times smaller than that of qDRIFT.

DOI: 10.1103/PRXQuantum.5.020330

I. INTRODUCTION

Hamiltonian simulation is a key subroutine of quantum algorithms for simulating quantum systems. Given a Hamiltonian $H = \sum_{\ell=1}^{L} h_\ell H_\ell$, where $h_\ell \geq 0$ and L is the number of terms, the task in Hamiltonian simulation is to construct a quantum circuit that approximately emulates time evolution $U(t) := \exp(-\mathrm{i}Ht)$ of the system for time t. Several approaches have been established for this task. The conventional approach uses the Trotter-Suzuki decompositions that provide a deterministic method for Hamiltonian simulation [1–3]. The gate count of this approach scales at least linearly with the number of terms L in H [3]; we note that the gate count in Ref. [2] scales at least quadratically with L. Although this scaling is formally efficient it is impractical for many applications of interest, particularly for the electronic structure problem in quantum chemistry, where the number of terms in a Hamiltonian is prohibitively large. An alternative approach is to randomly permute the order of terms in the Trotter-Suzuki decompositions [4]. This randomized compilation provides a slightly better scaling for the gate count over Ref. [2] but the gate count still depends quadratically on the number of Hamiltonian terms.

The quantum stochastic drift protocol (qDRIFT) [5] is another randomized Hamiltonian-simulation approach but

*Corresponding author: kohei.nakaji@utoronto.ca

†Corresponding author: mohsen.bagherimehrab@utoronto.ca

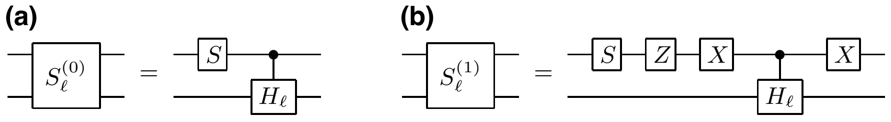

FIG. 1. Quantum circuits for implementing the swift operators $\tilde{\mathcal{S}}_\ell^{(b)} := S_\ell^{(b)} \rho S_\ell^{(b)\dagger}$ ($b \in \{0, 1\}$). The top line corresponds to an ancilla qubit and the bottom line corresponds to the qubits in the system. The S gate is defined by the operator $e^{i\pi/2}e^{i\sigma_z}$, where σ_z is the Pauli-Z operator. (a) The quantum circuit for $\tilde{\mathcal{S}}_\ell^{(0)}(\rho)$. (b) The quantum circuit for $\tilde{\mathcal{S}}_\ell^{(1)}(\rho)$.

is independent of the number of terms. In qDRIFT, gates of the form $\exp(-iH_\ell\tau)$ with a small interval τ are applied randomly with a probability proportional to the strength h_ℓ of the corresponding term in the Hamiltonian. qDRIFT improves upon the Trotter-Suzuki approach in that its gate count is independent of L and $\Lambda := \max_\ell h_\ell$ (the magnitude of the strongest term in the Hamiltonian) and instead depends on $\lambda := \sum_{\ell=1}^{L} h_\ell$. However, qDRIFT has poor scaling with respect to the precision ε, in contrast to that in the Trotter-Suzuki approach. We note that there are other approaches to Hamiltonian simulation with asymptotically better performance as a function of various parameters [6–11]. Still, the approaches based on product formulas, e.g., Trotter-Suzuki decompositions and qDRIFT, are preferred for their superior performance in practice [12] and their predominant usage in experimental implementations [13–15] due to their simplicity and the fact that they do not require any ancilla qubits. From this perspective, we focus on an approach based on product formulas, while having better gate scaling than the previous methods.

In this paper, we propose the *quantum swift protocol (qSWIFT)* [16], a high-order randomized algorithm having (i) better scaling with respect to the precision ε and (ii) the same scaling with respect to λ compared to qDRIFT. Specifically, the gate count of qSWIFT scales as $\mathcal{O}((\lambda t)^2/\varepsilon^{1/K})$, where K is the order parameter, while that of qDRIFT scales as $\mathcal{O}((\lambda t)^2/\varepsilon)$. For example, with respect to the precision ε, the gate count of qSWIFT scales as $\mathcal{O}(1/\sqrt{\varepsilon})$ for the second order and as $\mathcal{O}(1/\varepsilon^{1/3})$ for the third order. Our qSWIFT algorithm shares its simplicity with the other approaches based on product formulas. It works in the system with one ancilla qubit (we refer to the qubits other than the ancilla qubit simply as the system qubits). We can construct all gate operations with $\exp(iH_\ell\tau)$ and the *swift operators* $\tilde{\mathcal{S}}_\ell^{b} := S_\ell^{(b)} \rho S_\ell^{(b)\dagger}$, where $S_\ell^{(b)}$ is a unitary transformation; the swift operators can be constructed if we can efficiently implement controlled-H_ℓ gates, as shown in Fig. 1. In the case of qDRIFT, the entire time evolution is divided into segments and a sampled time evolution $\exp(iH_\ell\tau)$ is performed in each segment [see Fig. 2(a)]. In qSWIFT, we utilize the *swift circuit* in addition to the circuit for qDRIFT. The swift circuit also has segments; in most segments, a sampled time evolution $\exp(iH_\ell\tau)$ is performed to the system qubits but in the other segments, a sequence of the swift operators is performed [see Fig. 2(b)]. The number of swift operators is upper bounded by about twice the order parameter. Therefore, the qSWIFT algorithm can be performed with almost no additional resources compared to qDRIFT.

We will now describe in more detail how qSWIFT is carried out. First, we build the qSWIFT channel that simulates the ideal time evolution. We then establish a bound for the distance between the qSWIFT channel and the ideal channel, quantified with the diamond norm, which exponentially decreases by increasing the order parameter. The established bound yields the desired scaling for the gate count of qSWIFT. It should be noted that the qSWIFT channel itself is not physical in the sense that it is not a completely positive and trace-preserving (CPTP) map. Nevertheless, we can employ the qSWIFT channel to develop a procedure for measuring a physical quantity of interest, i.e., computing the expectation value of some given observable that is exponentially more precise than the original qDRIFT with respect to the order parameter.

Our numerical analysis also reveals the advantage of our qSWIFT algorithm. We show the asymptotic behavior of qSWIFT by using electronic molecular Hamiltonians with the number of qubits being approximately 50 and we compare the performance with the other approaches based on product formulas. Specifically, we compute the required number of gates to approximate the time evolution with the molecule Hamiltonians with a given systematic error ε. We show that the number of gates in the third-order version of qSWIFT is 10 times smaller than that of qDRIFT when $\varepsilon = 0.001$ for every time region. A significant reduction of the number of gates is observed when $\varepsilon = 10^{-6}$; the required number of gates in third-order

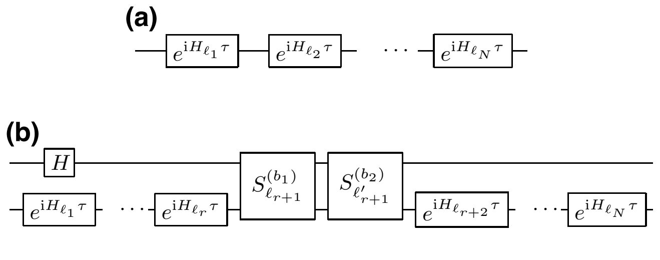

FIG. 2. Examples of quantum circuits used for qDRIFT and qSWIFT. (a) A quantum circuit for qDRIFT with N segments. (b) A swift circuit with N segments, where two swift operators are performed in the $(r+1)$th segment.

(sixth-order) qSWIFT is 1000 (10 000) times smaller than that of qDRIFT. We also simulate our qSWIFT algorithm and the other product-formula-based algorithms by using a quantum circuit simulator with the small-size (eight-qubit) molecular Hamiltonian. Its result is consistent with the result of the asymptotic behavior analysis.

The rest of the paper is organized as follows. In Sec. II, we briefly review approaches for Hamiltonian simulation based on product formulas. Sections III and IV are dedicated to proposing and analyzing our qSWIFT algorithm. In Sec. III, we introduce the way of constructing the second-order qSWIFT algorithm. Then, we generalize the algorithm to the higher order in Sec. IV. In Sec. V, we validate our algorithm by numerical experiments. Finally, in Sec. VI, we conclude with some discussions.

II. BACKGROUND

This section covers the key background pertinent to the following sections. We begin with a brief description of Hamiltonian simulation and the Trotter-Suzuki formulas in Sec. II A. Then, we review the qDRIFT algorithm for Hamiltonian simulation in Sec. II B.

A. Hamiltonian simulation by Trotter-Suzuki formulas

We begin with a brief description of Hamiltonian simulation. For a given time-independent Hamiltonian of the form $H=\sum_{\ell=1}^{L}h_\ell H_\ell$, where $h_\ell>0$ and H_ℓ are Hermitian operators with $\|H_\ell\|=1$, the task in Hamiltonian simulation is to find a good approximation of the transformation

$$\mathcal{U}(t):\rho\to U(t)\rho U^\dagger(t), \tag{1}$$

where $U(t):=\mathrm{e}^{\mathrm{i}Ht}$ and t is a real parameter. We assume that we can efficiently implement each $\mathrm{e}^{\mathrm{i}H_\ell t'}$ by quantum gates with t' as a real number. For example, if we decompose H to the sum of the tensor products of the Pauli operators, we can efficiently implement each $\mathrm{e}^{\mathrm{i}H_\ell t'}$.

The conventional approach for Hamiltonian simulation is the Trotter-Suzuki decomposition [1,2]. In the first-order Trotter-Suzuki decomposition for Hamiltonian simulation, the entire simulation for time t is divided into r segments of simulations for time t/r as $U(t)=(U(t/r))^r$ and $U(t/r)$ is approximated as $U(t/r)\approx U_{\mathrm{TS}}^{(1)}(t/r)$, with

$$U_{\mathrm{TS}}^{(1)}(t):=\prod_{\ell=1}^{L}\mathrm{e}^{\mathrm{i}h_\ell H_\ell t}, \tag{2}$$

which yields the approximation $U(t)\approx(U_{\mathrm{TS}}^{(1)}(t/r))^r$ for the entire simulation. The second-order Trotter-Suzuki decomposition is given by $U(t)\approx(U_{\mathrm{TS}}^{(2)}(t/r))^r$, with

$$U_{\mathrm{TS}}^{(2)}(t):=\prod_{\ell'=L}^{1}\mathrm{e}^{\mathrm{i}h_{\ell'}H_{\ell'}t/2}\prod_{\ell=1}^{L}\mathrm{e}^{\mathrm{i}h_\ell H_\ell t/2}, \tag{3}$$

which serves as the base case for the recursive formula

$$U_{\mathrm{TS}}^{(2k)}(t):=\left[U_{\mathrm{TS}}^{(2k-2)}(p_k t)\right]^2U_{\mathrm{TS}}^{(2k-2)}((1-4p_k)t)\times\left[U_{\mathrm{TS}}^{(2k-2)}(p_k t)\right]^2 \tag{4}$$

for the 2kth-order decomposition, where $p_k:=1/(4-4^{1/(2k-1)})$.

Let us discuss the Trotter-Suzuki decomposition in the channel representation. The 2kth-order Trotter-Suzuki channel $\mathcal{U}_{\mathrm{TS}}^{(2k)}(t):\rho\to U_{\mathrm{TS}}^{(2k)}(t)\rho U_{\mathrm{TS}}^{(2k)}(t)$ is used to approximate the channel $\mathcal{U}(\rho)$ as $\mathcal{U}(\rho)\approx(\mathcal{U}_{\mathrm{TS}}^{(2k)}(t/r))^r$. For a given channel $\mathcal{C}$, we denote by $\mathcal{C}^{r'}$ the r' repetition of $\mathcal{C}$. Previous analytic work [2,17,18] has shown that

$$||\mathcal{U}(t)-(\mathcal{U}_{\mathrm{TS}}^{(2k)}(t/r))^r||_\diamond\le\varepsilon \tag{5}$$

for $r\in\mathcal{O}(\alpha L\Lambda t(\alpha L\Lambda t/\varepsilon)^{1/2k})$ with $\alpha:=2\cdot5^{k-1}$, where $||\cdot||_\diamond$ is the diamond norm. We note that α here is defined so that $r\alpha L$ is the number of gates used in the 2kth-order decomposition. Hence the gate count for the 2kth-order Trotter-Suzuki decomposition, denoted by G_{TS}, is

$$G_{\mathrm{TS}}=r\alpha L\in O\left(\frac{\alpha^2L^2\Lambda t(\alpha L\Lambda t)^{\frac{1}{2k}}}{\varepsilon^{\frac{1}{2k}}}\right). \tag{6}$$

Note that the gate count approaches $\mathcal{O}(L^2\Lambda t)$ by increasing the order parameter 2k but the prefactor scales exponentially with 2k. Because of this rapidly growing prefactor, Trotter-Suzuki decompositions of finite orders, typically second ($k=1$) or fourth order ($k=2$), are used in practice [5].

We note that Ref. [3] demonstrates that the gate-count scaling can be more rigorously bounded using the commutator bounds, where the upper bound is represented as the commutation relation. However, we use the gate-count scaling in Eq. (6) to compare qSWIFT against qDRIFT [5], particularly for comparing the numerical experiments in Ref. [5].

B. Hamiltonian simulation by qDRIFT

Developed by Campbell [5], qDRIFT is an algorithm for Hamiltonian simulation using a randomized procedure. While the procedure is randomized, with many repetitions the evolution stochastically drifts toward the target unitary. Specifically, the exact time evolution is approximated by N

repetitions of the qDRIFT channel $\mathcal{E}_N$ as $\mathcal{U} \approx \mathcal{E}_N^N$, with the qDRIFT channel defined as

$$\mathcal{E}_N(\rho) := \sum_{\ell=1}^{L} p_\ell \mathcal{T}_\ell(\rho), \tag{7}$$

where

$$\mathcal{T}_\ell(\rho) = \mathrm{e}^{\mathrm{i}H_\ell \tau} \rho \mathrm{e}^{-\mathrm{i}H_\ell \tau}, \tag{8}$$

is the unitary channel that we call the *time operator* and

$$p_\ell := h_\ell/\lambda, \quad \lambda = \sum_\ell h_\ell, \quad \tau := \lambda t/N, \tag{9}$$

are three variables used through the paper. To realize the qDRIFT channel $\mathcal{E}_N$, the index ℓ is sampled according to the probability p_ℓ and the quantum state ρ is evolved through the channel associated with the operator $\mathrm{e}^{\mathrm{i}H_\ell \tau}$.

For evaluating the systematic error of the approximation, they define the exact short-time evolution $\mathcal{U}_N$ as

$$\mathcal{U}_N(\rho) := e^{\mathrm{i}Ht/N} \rho e^{-\mathrm{i}Ht/N}. \tag{10}$$

Then, they show

$$d_\diamond(\mathcal{U}_N, \mathcal{E}_N) \le \frac{2(\lambda t)^2}{N^2} e^{2\lambda t/N}, \tag{11}$$

where the diamond distance is defined as

$$d_\diamond(\mathcal{U}', \mathcal{E}') := \frac{1}{2}||\mathcal{U}' - \mathcal{E}'||_\diamond. \tag{12}$$

They utilize the diamond distance $d_\diamond(\mathcal{U}, \mathcal{E}_N^N)$ as the measure of the systematic error. By using the subadditive feature of the diamond distance, they obtain the bound for the diamond distance as

$$\begin{aligned} d_\diamond(\mathcal{U}, \mathcal{E}_N^N) &\le N d_\diamond(\mathcal{U}_N, \mathcal{E}_N) \\ &\le \frac{2(\lambda t)^2}{N} e^{2\lambda t/N} \\ &\in \mathcal{O}\left(\frac{(\lambda t)^2}{N}\right). \end{aligned} \tag{13}$$

In other words, to reduce the systematic error within ε, we need to set $N \in \mathcal{O}((\lambda t)^2/\varepsilon)$.

In most of the applications of Hamiltonian simulation, our interest lies in computing the expectation value of an observable after applying the time-evolution operator $\mathcal{U}$. Let us write the expectation value as

$$q := \mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}})), \tag{14}$$

where Q is an observable and ρ_{init} is an input quantum state. By using the qDRIFT algorithm, we can approximately compute the value of Q as

$$q^{(1)} := \mathrm{Tr}\left(Q\mathcal{E}_N^N(\rho_{\mathrm{init}})\right), \tag{15}$$

where the systematic error is bounded as

$$|q - q^{(1)}| \le 2||Q||_\infty d_\diamond\left(\mathcal{U}, \mathcal{E}_N^N\right) \in \mathcal{O}\left(||Q||_\infty \left(\frac{(\lambda t)^2}{N}\right)\right). \tag{16}$$

III. SECOND-ORDER QSWIFT

In this section, we describe our second-order qSWIFT as a preparation for introducing the general high-order qSWIFT in Sec. IV. To elucidate our algorithm, we use a "mixture function" in our second- and higher-order qSWIFT. We begin by describing this function in Sec. III A. Next, we construct the second-order qSWIFT channel and discuss its error bound in Sec. III B. Finally, in Sec. III C, we explain how to apply the constructed qSWIFT channel for computing physical quantities.

A. Mixture function

In constructing our qSWIFT channels, we make use of a mixture function. As a preparation, let us first define the following *sorting function*.

Definition 1 (Sorting function). Let S_N be the permutation group. For positive integers k and N with $k < N$, let $\vec{\mathcal{A}} := (\mathcal{A}_1, \ldots, \mathcal{A}_k)$ and $\vec{\mathcal{B}} := (\mathcal{B}_1, \ldots, \mathcal{B}_{N-k})$. We define the sorting function as

$$f_{\sigma,k,N-k}(\vec{\mathcal{A}}, \vec{\mathcal{B}}) := \mathcal{X}_{\sigma(1)}\mathcal{X}_{\sigma(2)} \cdots \mathcal{X}_{\sigma(N)}, \tag{17}$$

where $\sigma \in S_N$ and

$$\mathcal{X}_j = \begin{cases} \mathcal{A}_j & j \le k, \\ \mathcal{B}_{j-k} & j \ge k+1. \end{cases} \tag{18}$$

The mixture function is defined by using the sorting function as follows.

Definition 2 (Mixture function). For positive integers k and N with $k < N$, let $\vec{\mathcal{A}} := (\mathcal{A}_1, \ldots, \mathcal{A}_k)$ and $\vec{\mathcal{B}} := (\mathcal{B}_1, \ldots, \mathcal{B}_{N-k})$. Then, we define the mixture function as

$$M_{k,N-k}(\vec{\mathcal{A}}, \vec{\mathcal{B}}) = \sum_{\sigma \in S_{N,k}^{\mathrm{sub}}} f_{\sigma,k,N-k}(\vec{\mathcal{A}}, \vec{\mathcal{B}}), \tag{19}$$

where the set $S_{N,k}^{\mathrm{sub}}$ is the subgroup of the permutation group S_N comprised of all elements $\sigma \in S_N$ that satisfies the following condition: if both $\mathcal{X}_i, \mathcal{X}_j \in \vec{\mathcal{A}}$ or $\in \vec{\mathcal{B}}$, then $\sigma(i) < \sigma(j)$ for any $i < j$. We remark that the number of elements in $S^{\mathrm{sub}}_{N,k}$ is $\binom{N}{k}$.

For simplicity, if the elements of $\vec{\mathcal{B}}$ are identical, we denote the sorting and mixture functions as $f_{\sigma,k,N-k}(\vec{\mathcal{A}}, \mathcal{B})$ and $M_{k,N-k}(\vec{\mathcal{A}}, \mathcal{B})$, respectively. In this case, $\mathcal{X}_j = \mathcal{B}$ for $j \geq k+1$. Similarly, we use the notation $f_{\sigma,k,N-k}(\mathcal{A}, \mathcal{B})$ and $M_{k,N-k}(\mathcal{A}, \mathcal{B})$ if elements of $\vec{\mathcal{A}}$, and also elements of $\vec{\mathcal{B}}$, are identical. In this case, $\mathcal{X}_j = \mathcal{A}$ for $j \leq k$ and $\mathcal{X}_j = \mathcal{B}$ for $j \geq k+1$.

Note that the sorting function in Eq. (17) and the mixture function in Eq. (19) are bilinear functions. For example, if the ℓth element of $\vec{\mathcal{A}}$ is a linear combination of elements of another vector $\vec{\mathcal{F}}$, i.e., if $\mathcal{A}_\ell = \sum_n c_n \mathcal{F}_n$ for $c_n \in \mathbb{C}$, then we have

$$M_{k,N-k}\left((\mathcal{A}_1, \ldots, \mathcal{A}_{\ell-1}, \sum_n c_n \mathcal{F}_n, \mathcal{A}_{\ell+1}, \ldots, \mathcal{A}_k), \vec{\mathcal{B}}\right) = \sum_n c_n M_{k,N-k}\Big((\mathcal{A}_1, \ldots, \mathcal{A}_{\ell-1}, \mathcal{F}_n, \mathcal{A}_{\ell+1}, \ldots, \mathcal{A}_k), \vec{\mathcal{B}}\Big). \tag{20}$$

In general, if $\mathcal{A}_\ell = \sum_{n_\ell} c_{n_\ell} \mathcal{F}_{n_\ell}$ for any ℓ, then the identity

$$M_{k,N-k}\left(\vec{\mathcal{A}}, \vec{\mathcal{B}}\right) = \sum_{n_1} c_{n_1} \sum_{n_2} c_{n_2} \cdots \sum_{n_k} c_{n_k} \times M_{k,N-k}\left((\mathcal{F}_{n_1}, \ldots, \mathcal{F}_{n_k}), \vec{\mathcal{B}}\right) \tag{21}$$

holds.

B. Second-order qSWIFT channel

To construct the qSWIFT channel, let us define

$$\mathcal{L}_\ell(\rho) := \mathrm{i}[H_\ell, \rho], \tag{22}$$

$$\mathcal{L}(\rho) := \frac{\mathrm{i}}{\lambda}[H, \rho] = \sum_\ell p_\ell \mathcal{L}_\ell(\rho), \tag{23}$$

where the variables λ, p_ℓ, and τ are defined in Eq. (9). We then have

$$\mathcal{U}_N = \mathrm{e}^{\mathcal{L}\tau} = \mathbb{I} + \tau\mathcal{L} + \Delta^{(2)}\mathcal{U}_N, \tag{24}$$

$$\mathcal{E}_N = \sum_\ell p_\ell \mathrm{e}^{\mathcal{L}_\ell \tau} = \mathbb{I} + \tau\mathcal{L} + \Delta^{(2)}\mathcal{E}_N, \tag{25}$$

for the ideal time-evolution channel in Eq. (10) and the qDRIFT channel in Eq. (7), where

$$\Delta^{(k)}\mathcal{U}_N = \sum_{n=k}^{\infty} \frac{\tau^n}{n!}\mathcal{L}^n, \tag{26}$$

$$\Delta^{(k)}\mathcal{E}_N = \sum_{n=k}^{\infty} \frac{\tau^n}{n!} \sum_{\ell=1}^{L} p_\ell \mathcal{L}_\ell^n. \tag{27}$$

Let $\Delta_k := \Delta^{(k)}\mathcal{U}_N - \Delta^{(k)}\mathcal{E}_N$. Then,

$$\Delta_k = \sum_{n=k}^{\infty} \frac{\tau^n}{n!}\mathcal{L}^{(n)}, \quad \mathcal{L}^{(n)} := \mathcal{L}^n - \sum_{\ell=1}^{L} p_\ell \mathcal{L}_\ell^n. \tag{28}$$

Using the definition of Δ_k and Eqs. (24) and (25), we have $\mathcal{U}_N = \mathcal{E}_N + \Delta_2$, which we use to expand $\mathcal{U} = \mathcal{U}_N^N$ as

$$\begin{aligned}\mathcal{U} = (\mathcal{E}_N + \Delta_2)^N &= \mathcal{E}_N^N + \sum_{k=1}^{N} M_{k,N-k}(\Delta_2, \mathcal{E}_N) \\ &= \mathcal{E}_N^N + \frac{\tau^2}{2} M_{1,N-1}\left(\mathcal{L}^{(2)}, \mathcal{E}_N\right) + M_{1,N-1}(\Delta_3, \mathcal{E}_N) \\ &\quad + \sum_{k=2}^{N} M_{k,N-k}(\Delta_2, \mathcal{E}_N),\end{aligned} \tag{29}$$

where we have used $\Delta_2 = (\tau^2/2)\mathcal{L}^{(2)} + \Delta_3$ and linearity of $M_{1,N-1}$ to obtain the last equality. Let us denote the first two terms as

$$\mathcal{E}^{(2)} := \mathcal{E}_N^N + \frac{\tau^2}{2} M_{1,N-1}\left(\mathcal{L}^{(2)}, \mathcal{E}_N\right). \tag{30}$$

We refer to $\mathcal{E}^{(2)}$ as the *second-order qSWIFT* channel. In the following lemma, we provide a bound for the error in approximating the ideal channel $\mathcal{U}$ in Eq. (1) by the second-order qSWIFT channel $\mathcal{E}^{(2)}$, where the error is quantified as the diamond norm of their difference.

Lemma 1. Let $\mathcal{U}$ be the ideal channel in Eq. (1) and let $\mathcal{E}^{(2)}$ be the second-order qSWIFT channel in Eq. (30). Then, in the region $\lambda t \geq 1$,

$$d_\diamond\left(\mathcal{U}, \mathcal{E}^{(2)}\right) \in \mathcal{O}\left(\left(\frac{(\lambda t)^2}{N}\right)^2\right), \tag{31}$$

provided that $N \leq 2\sqrt{2}e(\lambda t)^2$.

We provide the proof in Appendix A 1. By this lemma, when we consider the reasonable parameter region $\lambda t \geq 1$, if $N \in \mathcal{O}\left((\lambda t)^2/\sqrt{\varepsilon}\right)$ for $\varepsilon > 0$, then $d_\diamond\left(\mathcal{U}, \mathcal{E}^{(2)}\right) \leq \varepsilon$. This provides a quadratic improvement over the original qDRIFT with respect to ε.

C. Implementation of the second-order qSWIFT channel

The second-order qSWIFT channel that we have constructed is not a physical channel, as it is not a CPTP map. Therefore, we cannot directly implement the qSWIFT channel itself. However, in most applications of Hamiltonian simulation, our interest is in computing physical quantities, i.e., the expectation value of an observable after applying the time-evolution operator as described in Sec. II B. Thus, in the following, we focus on how to compute $q^{(2)} := \mathrm{Tr}(Q\mathcal{E}^{(2)}(\rho_{\mathrm{init}}))$, for a given observable Q and the input ρ_{init}. By using $q^{(2)}$, the systematic error is reduced to

$$|q - q^{(2)}| \le 2||Q||_\infty d_\diamond\left(\mathcal{U}, \mathcal{E}^{(2)}\right) \in \mathcal{O}\left(||Q||_\infty \left(\frac{(\lambda t)^2}{N}\right)^2\right), \tag{32}$$

where we use Eq. (31), which is the quadratic improvement in terms of $(\lambda t)^2/N$ from the one in qDRIFT, given in Eq. (16).

Here, we provide a way of computing $q^{(2)}$ by quantum circuits. We can expand $q^{(2)}$ as

$$q^{(2)} = q^{(1)} + \delta q, \tag{33}$$

where

$$\delta q = \frac{\tau^2}{2}\mathrm{Tr}\left(QM_{1,N-1}\left(\mathcal{L}^{(2)}, \mathcal{E}_N\right)(\rho_{\mathrm{init}})\right). \tag{34}$$

For computing $q^{(1)}$, we just need to apply the original qDRIFT channel and compute the expectation value. Thus, we focus on how to compute δq in the following. More specifically, we will show that δq is computable by using the swift circuits, composed of the time evolution $\exp(iH_\ell\tau)$ and the swift operators shown in Fig. 1.

By using

$$M_{1,N-1}\left(\mathcal{L}^{(2)}, \mathcal{E}_N\right) = \sum_{r=0}^{N-1}\mathcal{E}_N^{N-1-r}\mathcal{L}^{(2)}\mathcal{E}_N^r, \tag{35}$$

which can be derived from the definition in Eq. (19), we obtain

$$\delta q = \frac{\tau^2}{2}\sum_{r=0}^{N-1}\delta q_r, \tag{36}$$

where $\delta q_r := \mathrm{Tr}\left(Q\mathcal{E}_N^{N-1-r}\mathcal{L}^{(2)}\mathcal{E}_N^r(\rho_{\mathrm{init}})\right)$. Now let us move on to the evaluation of δq_r. We transform δq_r as

$$\begin{aligned}\delta q_r &= \mathrm{Tr}\left(Q\mathcal{E}_N^{N-1-r}\mathcal{L}^{(2)}\mathcal{E}_N^r(\rho_{\mathrm{init}})\right),\\ &= \sum_{\ell,k=1}^{L}p_\ell p_k \mathrm{Tr}\left(Q\mathcal{E}_N^{N-1-r}\mathcal{L}_\ell\mathcal{L}_k\mathcal{E}_N^r(\rho_{\mathrm{init}})\right)\\ &\quad - \sum_{\ell=1}^{L}p_\ell \mathrm{Tr}\left(Q\mathcal{E}_N^{N-1-r}\mathcal{L}_\ell^2\mathcal{E}_N^r(\rho_{\mathrm{init}})\right),\end{aligned} \tag{37}$$

where we use Eq. (28) in the second equality. Let two probability distributions be

$$P_0^{(2)}(\vec{\ell}) = p_{\ell_2}p_{\ell_1}, \quad P_1^{(2)}(\vec{\ell}) = \begin{cases} p_\ell, & \ell_1 = \ell_2 = \ell,\\ 0, & \ell_1 \ne \ell_2,\end{cases} \tag{38}$$

where the input $\vec{\ell}$ is a vector with two elements (ℓ_1 and ℓ_2). Also, let

$$\mathcal{L}_2(\vec{\ell}) := \mathcal{L}_{\ell_2}\mathcal{L}_{\ell_1}. \tag{39}$$

Then, it holds that

$$\delta q_r := \sum_{s=0}^{1}(-1)^s\sum_{\vec{\ell}}P_s^{(2)}(\vec{\ell})\mathrm{Tr}\left(Q\mathcal{K}(r,\vec{\ell})(\rho_{\mathrm{init}})\right), \tag{40}$$

where we define a channel $\mathcal{K}(r,\vec{\ell})$ as

$$\mathcal{K}(r,\vec{\ell}) := \mathcal{E}_N^{N-1-r}\mathcal{L}_2(\vec{\ell})\mathcal{E}_N^r. \tag{41}$$

To evaluate each term of Eq. (40), we utilize a system with one ancilla qubit. Let us write the density matrix for a given system with one ancilla qubit as the matrix form

$$\begin{pmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{pmatrix} := |0\rangle\langle 0|\otimes\rho_{00} + |0\rangle\langle 1|\otimes\rho_{01} + |1\rangle\langle 0|\otimes\rho_{10} + |1\rangle\langle 1|\otimes\rho_{11}. \tag{42}$$

We define the operation of a quantum channel $\tilde{\mathcal{K}}(r,\vec{\ell})$ that transforms the initial state $\tilde{\rho}_{\mathrm{init}} = |+\rangle\langle +|\otimes\rho_{\mathrm{init}}$ into a final state as

$$\begin{aligned}\tilde{\rho}_{\mathrm{init}} &= \begin{pmatrix}\rho_{\mathrm{init}}/2 & \rho_{\mathrm{init}}/2\\ \rho_{\mathrm{init}}/2 & \rho_{\mathrm{init}}/2\end{pmatrix}\\ &\xmapsto{\tilde{\mathcal{K}}(r,\vec{\ell})}\begin{pmatrix}\cdot & \mathcal{K}(r,\vec{\ell})(\rho_{\mathrm{init}})/2\\ \mathcal{K}(r,\vec{\ell})(\rho_{\mathrm{init}})/2 & \cdot\end{pmatrix},\end{aligned} \tag{43}$$

where the dot ("·") in the diagonal element denotes a matrix in which we do not have any interest at this stage.

Then, it holds that

$$\mathrm{Tr}(Q\mathcal{K}(r,\vec{\ell})(\rho_{\mathrm{init}})) = \mathrm{Tr}\left(\tilde{Q}\tilde{\mathcal{K}}(r,\vec{\ell})(\tilde{\rho}_{\mathrm{init}})\right) \tag{44}$$

and, consequently,

$$\delta q_r := \sum_{s=0}^{1}(-1)^s \sum_{\vec{\ell}} P_s^{(2)}(\vec{\ell})\mathrm{Tr}\left(\tilde{Q}\tilde{\mathcal{K}}(r,\vec{\ell})(\tilde{\rho}_{\mathrm{init}})\right), \tag{45}$$

where

$$\tilde{Q} = X \otimes Q, \tag{46}$$

where X is the Pauli-X observable for the ancilla qubit. Therefore, we can evaluate δq_r if we implement the channel $\tilde{\mathcal{K}}(r,\vec{\ell})$ by using quantum circuits.

We can specify the channel $\tilde{\mathcal{K}}(r,\vec{\ell})$ as follows:

$$\tilde{\mathcal{K}}(r,\vec{\ell}) := \tilde{\mathcal{E}}_N^{N-1-r}\tilde{\mathcal{L}}_{\ell_2}\tilde{\mathcal{L}}_{\ell_1}\tilde{\mathcal{E}}_N^r, \tag{47}$$

where we define $\tilde{\mathcal{E}}_N := \mathbf{1}\otimes\mathcal{E}_N$ and $\vec{\ell} = (\ell_1,\ell_2)$ and where the operation of the channel $\tilde{\mathcal{L}}_\ell$ is specified for the input having the identical nondiagonal elements as follows:

$$\begin{pmatrix} \cdot & \rho \\ \rho & \cdot \end{pmatrix} \xmapsto{\tilde{\mathcal{L}}_\ell} \begin{pmatrix} \cdot & \mathcal{L}_\ell(\rho) \\ \mathcal{L}_\ell(\rho) & \cdot \end{pmatrix}. \tag{48}$$

The channel $\tilde{\mathcal{L}}_\ell$ can be written as the sum of two swift operators, $\tilde{\mathcal{S}}_\ell^{(0)}$ and $\tilde{\mathcal{S}}_\ell^{(1)}$, introduced in Fig. 1 as

$$\tilde{\mathcal{L}}_\ell = \tilde{\mathcal{S}}_\ell^{(0)} + \tilde{\mathcal{S}}_\ell^{(1)}, \tag{49}$$

which is easily checked by using

$$\begin{aligned}
&\begin{pmatrix} \cdot & \rho \\ \rho & \cdot \end{pmatrix} \xmapsto{\tilde{\mathcal{S}}_\ell^{(0)}} \begin{pmatrix} \cdot & -\mathrm{i}\rho H_\ell \\ \mathrm{i}H_\ell\rho & \cdot \end{pmatrix},\\
&\begin{pmatrix} \cdot & \rho \\ \rho & \cdot \end{pmatrix} \xmapsto{\tilde{\mathcal{S}}_\ell^{(1)}} \begin{pmatrix} \cdot & \mathrm{i}H_\ell\rho \\ -\mathrm{i}\rho H_\ell & \cdot \end{pmatrix}.
\end{aligned} \tag{50}$$

It can be pedagogically shown that the channel in Eq. (47) reproduces the transformation in Eq. (43); with $\rho'_{\mathrm{init}} = \rho_{\mathrm{init}}/2$, it holds that

$$\begin{aligned}
\begin{pmatrix} \rho'_{\mathrm{init}} & \rho'_{\mathrm{init}} \\ \rho'_{\mathrm{init}} & \rho'_{\mathrm{init}} \end{pmatrix} &\xmapsto{\tilde{\mathcal{E}}_N^r} \begin{pmatrix} \mathcal{E}_N^r(\rho'_{\mathrm{init}}) & \mathcal{E}_N^r(\rho'_{\mathrm{init}}) \\ \mathcal{E}_N^r(\rho'_{\mathrm{init}}) & \mathcal{E}_N^r(\rho'_{\mathrm{init}}) \end{pmatrix}\\
&\xmapsto{\tilde{\mathcal{L}}_{\ell_1}} \begin{pmatrix} \cdot & \mathcal{L}_{\ell_1}\mathcal{E}_N^r(\rho'_{\mathrm{init}}) \\ \mathcal{L}_{\ell_1}\mathcal{E}_N^r(\rho'_{\mathrm{init}}) & \cdot \end{pmatrix}\\
&\xmapsto{\tilde{\mathcal{L}}_{\ell_2}} \begin{pmatrix} \cdot & \mathcal{L}_{\ell_2}\mathcal{L}_{\ell_1}\mathcal{E}_N^r(\rho'_{\mathrm{init}}) \\ \mathcal{L}_{\ell_2}\mathcal{L}_{\ell_1}\mathcal{E}_N^r(\rho'_{\mathrm{init}}) & \cdot \end{pmatrix}\\
&\xmapsto{\tilde{\mathcal{E}}_N^{N-1-r}} \begin{pmatrix} \cdot & \tilde{\mathcal{K}}(r,\vec{\ell})(\rho'_{\mathrm{init}}) \\ \tilde{\mathcal{K}}(r,\vec{\ell})(\rho'_{\mathrm{init}}) & \cdot \end{pmatrix}.
\end{aligned} \tag{51}$$

By substituting Eqs. (49)–(47), we obtain

$$\tilde{\mathcal{K}}(r,\vec{\ell}) = \sum_{b_1,b_2=0}^{1} \tilde{\mathcal{E}}_N^{N-1-r}\tilde{\mathcal{S}}_{\ell_2}^{(b_2)}\tilde{\mathcal{S}}_{\ell_1}^{(b_1)}\tilde{\mathcal{E}}_N^r. \tag{52}$$

Finally, from Eqs.(36), (45), and (52),

$$\begin{aligned}
\delta q &= \frac{\tau^2}{2}\sum_{s=0}^{1}(-1)^s \sum_{b_1,b_2=0}^{1}\sum_{r=0}^{N-1}\sum_{\vec{\ell}} P_s^{(2)}(\vec{\ell})\mathrm{Tr}\\
&\quad\times\left(\tilde{Q}\tilde{\mathcal{E}}_N^{N-1-r}\tilde{\mathcal{S}}_{\ell_2}^{(b_2)}\tilde{\mathcal{S}}_{\ell_1}^{(b_1)}\tilde{\mathcal{E}}_N^r(\rho_{\mathrm{init}})\right),\\
&= \frac{(\lambda t)^2}{2N}\sum_{s=0}^{1}(-1)^s\sum_{b_1,b_2=0}^{1}\delta q(s,b_1,b_2),
\end{aligned} \tag{53}$$

where in the second equality, we define

$$\begin{aligned}
\delta q(s,b_1,b_2) &:= \frac{1}{N}\sum_{r=0}^{N-1}\sum_{\vec{\ell}} P_s^{(2)}(\vec{\ell})\mathrm{Tr}\\
&\quad\times\left(\tilde{Q}\tilde{\mathcal{E}}_N^{N-1-r}\tilde{\mathcal{S}}_{\ell_2}^{(b_2)}\tilde{\mathcal{S}}_{\ell_1}^{(b_1)}\tilde{\mathcal{E}}_N^r(\tilde{\rho}_{\mathrm{init}})\right).
\end{aligned} \tag{54}$$

We can evaluate Eq. (54) using Monte Carlo sampling. To this end, let

$$P_{\mathrm{MDRIFT}}(\vec{k},n) = p_{k_1}p_{k_2}\cdots p_{k_n} \tag{55}$$

be the probability distribution for product of multiple qDRIFT probability distributions, where $n\in\mathbb{Z}^+$ specifies the number of qDRIFT distributions and k_j, for each j, goes from 1 to L. Then,

$$\tilde{\mathcal{E}}_N^n = \sum_{k_1,k_2,\ldots k_n}^{L} P_{\mathrm{MDRIFT}}(\vec{k},n)\tilde{\mathcal{T}}_n(\vec{k}), \tag{56}$$

where $\tilde{\mathcal{T}}_n(\vec{k})$ is the unitary channel defined as the sequence of the time operators

$$\tilde{\mathcal{T}}_n(\vec{k}) := (\mathbf{1}\otimes\mathcal{T}_{k_n}\cdots\mathcal{T}_{k_2}\mathcal{T}_{k_1}). \tag{57}$$

We obtain an unbiased estimator of $\delta q(s,b_1,b_2)$ as follows:

(1) Sample ℓ uniformly from $\{0,1,\ldots,N-1\}$.
(2) With probability $P_s^{(2)}(\vec{\ell})$, sample $\vec{\ell}=(\ell_1,\ell_2)$. With probability $P_{\mathrm{MDRIFT}}(\vec{k},r)$ and $P_{\mathrm{MDRIFT}}(\vec{k}',N-1-r)$, sample $\vec{k}$ and $\vec{k}'$.
(3) Estimate

$$\mathrm{Tr}\left(\tilde{Q}\tilde{\mathcal{T}}_{N-1-r}(\vec{k}')\tilde{\mathcal{S}}_{\ell_2}^{(b_2)}\tilde{\mathcal{S}}_{\ell_1}^{(b_1)}\tilde{\mathcal{T}}_r(\vec{k})(\tilde{\rho}_{\mathrm{init}})\right) \tag{58}$$

with N_{shot} measurements and set the resulting value to $\delta\hat{q}(s,b_1,b_2)$, which is an unbiased estimator of

$\delta q(s, b_1, b_2)$. The value in Eq. (58) can be evaluated by applying a swift circuit composed of the time evolution and the swift operators and estimating the expectation value of $\tilde{Q}$ by measurements.

We repeat the above process N_{sample} times and the estimate of $\delta q(s, b_1, b_2)$ is computed as the sample average. By substituting each estimate of $\delta q(s, b_1, b_2)$ into Eq. (54), we obtain the estimate of δq as $\delta\hat{q}$. It should be noted that the swift circuit for evaluating each term of Eq. (58) has the structure that two swift operators are tucked in between $N-1$ time operators as in Fig. 2(b). Therefore, the number of gates in the swift circuit is almost the same as the original qDRIFT, which requires N time operators.

IV. HIGHER-ORDER QSWIFT

In this section, we generalize the second-order qSWIFT channel introduced in Sec. III B to an arbitrary high-order channel. First, we construct the higher-order qSWIFT channel and discuss the error bound in Sec. IV A. Then, in Sec. IV B, we construct an algorithm to apply the qSWIFT channel for computing physical quantities.

A. Higher-order qSWIFT channel

To construct a high-order qSWIFT channel, we retain higher orders of τ in the right-hand side of

$$\mathcal{U} = (\mathcal{E}_N + \Delta_2)^N = \mathcal{E}_N^N + \sum_{k=1}^{N} M_{k,N-k}(\Delta_2, \mathcal{E}_N), \tag{59}$$

where $M_{k,N-k}$ is the mixture function defined in Eq. (19). We note that Δ_2 is a linear combination of $\mathcal{L}^{(n)}$ as per Eq. (28). Thus, by Eq. (21), we obtain

$$\begin{aligned} M_{k,N-k}(\Delta_2, \mathcal{E}_N) &= \sum_{n_1=2}^{\infty} \frac{\tau^{n_1}}{n_1!} \sum_{n_2=2}^{\infty} \frac{\tau^{n_2}}{n_2!} \cdots \sum_{n_k=2}^{\infty} \frac{\tau^{n_k}}{n_k!} M_{k,N-k} \\ &\quad \left(\left(\mathcal{L}^{(n_1)}, \ldots, \mathcal{L}^{(n_k)}\right), \mathcal{E}_N\right) \\ &= \sum_{n_1,n_2,\ldots,n_k=2}^{\infty} \frac{\tau^{\sum_{j=1}^{k} n_j}}{n_1! n_2! \cdots n_k!} \sum_{\xi=2}^{\infty} \delta\left[\xi, \sum_{\ell=1}^{k} n_\ell\right] \\ &\quad M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)}, \ldots, \mathcal{L}^{(n_k)}\right), \mathcal{E}_N\right) \\ &= \sum_{\xi=2}^{\infty} \tau^{\xi} \sum_{n_1,n_2,\ldots,n_k=2}^{\xi} \frac{1}{n_1! n_2! \cdots n_k!} \delta\left[\xi, \sum_{j=1}^{k} n_j\right] \\ &\quad M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)}, \ldots, \mathcal{L}^{(n_k)}\right), \mathcal{E}_N\right), \end{aligned} \tag{60}$$

where $\delta[i,j]$ here is the Kronecker δ. To show the second equality, we use the identity

$$\sum_{\xi=2}^{\infty} \delta\left[\xi, \sum_{j=1}^{k} n_j\right] = 1, \tag{61}$$

which holds for a fixed set of integers $\{n_j\}_{j=1}^{k}$ with $n_j \geq 2$. In the last equality, we use the fact that the Kronecker δ is zero if any of $n_1, n_2, \ldots, n_k$ is larger than ξ. Truncating the upper limit of ξ yields a high-order qSWIFT channel. Specifically, we define the high-order qDRIFT as follows.

Definition 3 (Higher-order qSWIFT). We define the Kth-order qSWIFT channel ($K \leq N$) as

$$\begin{aligned} \mathcal{E}^{(K)} := \mathcal{E}_N^N + \sum_{\xi=2}^{2K-2} \tau^{\xi} \sum_{k=1}^{N} \sum_{n_1,\ldots,n_k=2}^{\xi} \frac{1}{n_1! n_2! \cdots n_k!} \delta \\ \times \left[\xi, \sum_{j=1}^{k} n_j\right] M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)}, \ldots, \mathcal{L}^{(n_k)}\right), \mathcal{E}_N\right). \end{aligned} \tag{62}$$

Here, we note that the upper limit N in the second summation can be replaced with K for $K \leq N$ because the terms with $k > K$ become zero by the Kronecker δ. We remark that setting $K = 2$ yields the second-order qSWIFT channel in Eq. (30). Also, by setting $K = 1$, we reproduce the qDRIFT channel.

We now provide a bound for the error in approximating the ideal channel $\mathcal{U}$ in Eq. (1) by the high-order qSWIFT channel $\mathcal{E}^{(K)}$ in the following lemma.

Lemma 2. Let $\mathcal{U}$ be the ideal channel in and let $\mathcal{E}^{(K)}$ be the Kth-order qSWIFT channel. Then, in the region $\lambda t \geq 1$,

$$d_\diamond\left(\mathcal{U}, \mathcal{E}^{(K)}\right) \in \mathcal{O}\left(\left(\frac{(\lambda t)^2}{N}\right)^K\right). \tag{63}$$

as far as $N \leq 2\sqrt{2}e(\lambda t)^2$.

The proof is given in Appendix A 2. As in the case of the second order, when we consider the reasonable parameter region $\lambda t \geq 1$, if $N \in \mathcal{O}\left((\lambda t)^2/\sqrt{\varepsilon}\right)$ for $\varepsilon > 0$, then $d_\diamond\left(\mathcal{U}, \mathcal{E}^{(K)}\right) \leq \varepsilon$.

B. Implementation of the higher-order qSWIFT channel

As we discuss in Sec. III C, the qSWIFT channel is not a physical channel but we can apply it to computing the physical quantities. Specifically, we discuss how to compute

$$q^{(K)} := \mathrm{Tr}\left(Q\mathcal{E}^{(K)}\right). \tag{64}$$

Then, the systematic error is bounded as

$$|q - q^{(K)}| \le 2||Q||_\infty d_\diamond\left(\mathcal{U}, \mathcal{E}^{(K)}\right)$$
$$\in \mathcal{O}\left(||Q||_\infty \left(\frac{(\lambda t)^2}{N}\right)^K\right), \tag{65}$$

which can be exponentially small with the order parameter.

Here, we provide the way to compute $q^{(K)}$. We can expand $q^{(K)}$ as

$$q^{(K)} = q^{(1)} + \sum_{\xi=2}^{2K-2}\sum_{k=1}^{N}\sum_{n_1,\ldots,n_k=2}^{\xi} \delta\left[\xi, \sum_{j=1}^{k} n_j\right]\delta q^{(k)}(\vec{n}) \tag{66}$$

$$= q^{(1)} + \sum_{\xi=2}^{2K-2}\sum_{k=1}^{K}\sum_{\vec{n}\in G_2(k,\xi)} \delta q^{(k)}(\vec{n}), \tag{67}$$

where

$$\delta q^{(k)}(\vec{n}) := \frac{\tau^{\sum_{j=1}^{k} n_j}}{n_1!n_2!\cdots n_k!} \times \mathrm{Tr}\left(QM_{k,N-k}\left(\left(\mathcal{L}^{(n_1)},\ldots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)(\rho_{\mathrm{init}})\right), \tag{68}$$

with $\vec{n} = \{n_1, n_2, \ldots, n_k\}$ and where we replace N in the second summation with K in the second equality (since the terms with $k > K$ become zero by the Kronecker δ.). To obtain Eq. (67), we define the set $G_2(k, \xi)$ composed of all vectors with k integer elements $\{n_j\}_{j=1}^{k}$ that satisfies $n_j \ge 2$ and $\sum_{j=1}^{k} n_j = \xi$. As an example, when using the second-order ($K = 2$) qSWIFT, the summation of the second term in Eq. (68) includes only one term, as

$$q^{(2)} = q^{(1)} + \delta q^{(1)}(\{2\}). \tag{69}$$

We see that δq in Eq. (33) corresponds to $\delta q^{(1)}(\{2\})$, where we write $\{a_1, \ldots, a_k\}$ as the vector having elements $a_1, \ldots, a_k$. As another example, when using the third-order ($K = 3$) qSWIFT,

$$q^{(3)} = q^{(1)} + \delta q^{(1)}(\{2\}) + \delta q^{(1)}(\{3\}) + \delta q^{(1)}(\{4\})$$
$$+ \delta q^{(2)}(\{2, 2\}). \tag{70}$$

Since $q^{(1)}$ is again evaluable by using the original qDRIFT, we focus on the evaluation of $\delta q^{(k)}(\vec{n})$ in the following.

Similar to the second-order case, we will transform $\delta q^{(k)}(\vec{n})$ as a sum of the terms that are evaluable with quantum circuits. The mixture function $M_{k,N-k}(\cdot)$ included in Eq. (68) is written as the summation of $\binom{N}{k}$ terms as

$$\delta q^{(k)}(\vec{n}) = \frac{\tau^{\sum_{j=1}^{k} n_j}}{n_1!n_2!\cdots n_k!}\sum_{\sigma\in S_{N,k}^{\mathrm{sub}}} \delta q_\sigma^{(k)}(\vec{n}), \tag{71}$$

where

$$\delta q_\sigma^{(k)}(\vec{n}) := \mathrm{Tr}\left(Qf_{\sigma,k,N-k}\left(\left(\mathcal{L}^{(n_1)},\ldots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)(\rho_{\mathrm{init}})\right), \tag{72}$$

where $f_{\sigma,k,N-k}$ is the sorting function defined in Eq. (17).

We now move on to the calculation of $\delta q_\sigma^{(k)}(\vec{n})$. To this end, let

$$\mathcal{D}_0^{(n)} := \mathcal{L}^n, \quad \mathcal{D}_1^{(n)} := \sum_\ell p_\ell \mathcal{L}_\ell^n. \tag{73}$$

We can write $\mathcal{L}^{(n)}$ as a linear combination of $\mathcal{D}_s^{(n)}$:

$$\mathcal{L}^{(n)} = \sum_{s=0}^{1}(-1)^s \mathcal{D}_s^{(n)}, \tag{74}$$

as per Eq. (28). We define two probability distributions:

$$P_0^{(n)}(\vec{\ell}) = P_{\mathrm{MDRIFT}}(\vec{\ell}, n), \tag{75}$$

$$P_1^{(n)}(\vec{\ell}) = \begin{cases} p_\ell, & \ell_1 = \cdots = \ell_n = \ell, \\ 0, & \ell_a \ne \ell_b \text{ for } \exists(a,b), \end{cases} \tag{76}$$

where n specifies the number of vectors in $\vec{\ell}$ and we write the ath element of $\vec{\ell}$ as ℓ_a. We see that for $n = 2$, Eq. (75) is consistent with Eq. (38). Then, it holds that

$$\mathcal{D}_s^{(n)} = \sum_{\vec{\ell}} P_s^{(n)}(\vec{\ell})\mathcal{L}_n(\vec{\ell}), \tag{77}$$

with

$$\mathcal{L}_n(\vec{\ell}) := \mathcal{L}_{\ell_n}\cdots\mathcal{L}_{\ell_1}, \tag{78}$$

which is consistent with Eq. (39) for $n = 2$. By using the bilinearity of the sorting function, we obtain

$$
\begin{aligned}
& f_{\sigma,k,N-k}\left(\left(\mathcal{L}^{(n_1)},\ldots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right) \\
&= \sum_{s_1,\ldots,s_k=0}^{1} (-1)^{\sum_c s_c} f_{\sigma,k,N-k}\left(\left(\mathcal{D}_{s_1}^{(n_1)},\ldots,\mathcal{D}_{s_k}^{(n_k)}\right),\mathcal{E}_N\right) \\
&= \sum_{s_1,\ldots,s_k=0}^{1} (-1)^{\sum_c s_c} \sum_{\vec{\ell}_1,\ldots\vec{\ell}_k} P_{s_1}^{(n_1)}(\vec{\ell}_1)\cdots P_{s_k}^{(n_k)}(\vec{\ell}_k) \\
&\quad f_{\sigma,k,N-k}\left(\left(\mathcal{L}_{n_1}(\vec{\ell}_1),\ldots,\mathcal{L}_{n_k}(\vec{\ell}_k)\right),\mathcal{E}_N\right),
\end{aligned}
\tag{79}
$$

where we use Eq. (74) in the first equality and Eq. (77) in the second equality. Substituting the above into Eq. (72), we obtain

$$
\begin{aligned}
\delta q_\sigma^{(k)}(\vec{n}) = {} & \sum_{s_1,\ldots,s_k=0}^{1} (-1)^{\sum_c s_c} \sum_{\vec{\ell}_1,\ldots\vec{\ell}_k} P_{s_1}^{(n_1)}(\vec{\ell}_1)\cdots P_{s_k}^{(n_k)}(\vec{\ell}_k) \\
& \delta q_\sigma^{(k)}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right)\right),
\end{aligned}
\tag{80}
$$

where

$$
\begin{aligned}
\delta q_\sigma^{(k)}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right)\right) := {} & \mathrm{Tr}\Big(Q f_{\sigma,k,N-k} \\
& \times \left(\left(\mathcal{L}_{n_1}(\vec{\ell}_1),\ldots,\mathcal{L}_{n_k}(\vec{\ell}_k)\right),\mathcal{E}_N\right)(\rho_{\text{init}})\Big).
\end{aligned}
\tag{81}
$$

As in Sec. III C, we compute the right-hand side of Eq. (81) by using the system with one ancilla qubit. Let

$$
\tilde{\mathcal{L}}_n(\vec{\ell}) := \tilde{\mathcal{L}}_{\ell_n}\cdots\tilde{\mathcal{L}}_{\ell_1}. \tag{82}
$$

By repeatedly operating Eq. (48) with $j = \ell_1,\ldots,\ell_n$, we obtain the operation of $\tilde{\mathcal{L}}_n(\vec{\ell})$ as

$$
\begin{pmatrix} \cdot & \rho \\ \rho & \cdot \end{pmatrix} \xmapsto{\tilde{\mathcal{L}}_n(\vec{\ell})} \begin{pmatrix} \cdot & \mathcal{L}_n(\vec{\ell})(\rho) \\ \mathcal{L}_n(\vec{\ell})(\rho) & \cdot \end{pmatrix} \tag{83}
$$

for a given density operator ρ. Recall the definition of the sorting function in Eq. (17):

$$
f_{\sigma,k,N-k}\left(\left(\mathcal{L}_{n_1}(\vec{\ell}_1),\ldots,\mathcal{L}_{n_k}(\vec{\ell}_k)\right),\mathcal{E}_N\right) = \mathcal{X}_{\sigma(1)}\cdots\mathcal{X}_{\sigma(N)}, \tag{84}
$$

with

$$
\mathcal{X}_a = \begin{cases} \mathcal{L}_{n_a}(\vec{\ell}_a), & a \le k, \\ \mathcal{E}_N, & a \ge k+1. \end{cases} \tag{85}
$$

Then, in the system with one ancilla qubit, it holds that

$$
f_{\sigma,k,N-k}\left(\left(\tilde{\mathcal{L}}_{n_1}(\vec{\ell}_1),\ldots,\tilde{\mathcal{L}}_{n_k}(\vec{\ell}_k)\right),\tilde{\mathcal{E}}_N\right) = \tilde{\mathcal{X}}_{\sigma(1)}\cdots\tilde{\mathcal{X}}_{\sigma(N)}, \tag{86}
$$

with

$$
\tilde{\mathcal{X}}_a = \begin{cases} \tilde{\mathcal{L}}_{n_a}(\vec{\ell}_a), & a \le k, \\ \tilde{\mathcal{E}}_N, & a \ge k+1. \end{cases} \tag{87}
$$

Since it holds that

$$
\begin{pmatrix} \cdot & \rho \\ \rho & \cdot \end{pmatrix} \xmapsto{\tilde{\mathcal{X}}_a} \begin{pmatrix} \cdot & \mathcal{X}_a(\rho) \\ \mathcal{X}_a(\rho) & \cdot \end{pmatrix}, \tag{88}
$$

the operation of $\tilde{\mathcal{X}}_{\sigma(1)}\cdots\tilde{\mathcal{X}}_{\sigma(N)}$ to the input state $\tilde{\rho}_{\text{init}} = |+\rangle\langle+| \otimes \rho_{\text{init}}$ reproduces the operation of $\mathcal{X}_{\sigma(1)}\cdots\mathcal{X}_{\sigma(N)}$ as

$$
\tilde{\rho}_{\text{init}} = \begin{pmatrix} \rho_{\text{init}}/2 & \rho_{\text{init}}/2 \\ \rho_{\text{init}}/2 & \rho_{\text{init}}/2 \end{pmatrix} \xmapsto{\tilde{\mathcal{X}}_{\sigma(1)}\cdots\tilde{\mathcal{X}}_{\sigma(N)}} \begin{pmatrix} \cdot & \mathcal{X}_{\sigma(1)}\cdots\mathcal{X}_{\sigma(N)}(\rho_{\text{init}})/2 \\ \mathcal{X}_{\sigma(1)}\cdots\mathcal{X}_{\sigma(N)}(\rho_{\text{init}})/2 & \cdot \end{pmatrix}. \tag{89}
$$

Therefore, the estimation value of the observable $\tilde{Q} = X \otimes Q$ with the final state in Eq. (89) gives $\delta q_\sigma^{(k)}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right)\right)$, i.e.,

$$
\delta q_\sigma^{(k)}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right)\right) = \mathrm{Tr}\left(\tilde{Q} f_{\sigma,k,N-k}\left(\left(\tilde{\mathcal{L}}_{n_1}(\vec{\ell}_1),\ldots,\tilde{\mathcal{L}}_{n_k}(\vec{\ell}_k)\right),\tilde{\mathcal{E}}_N\right)(\tilde{\rho}_{\text{init}})\right). \tag{90}
$$

Next, we discuss the way of evaluating the right-hand side of Eq. (90). By substituting Eqs. (49)–(90), we obtain

$$
\tilde{\mathcal{L}}_n(\vec{\ell}) = \sum_{\vec{b}} \tilde{\mathcal{S}}_n^{(\vec{b})}(\vec{\ell}), \tag{91}
$$

with $\vec{b} \in \{0,1\}^{\otimes n}$, where

$$
\tilde{\mathcal{S}}_n^{(\vec{b})}(\vec{\ell}) = \tilde{\mathcal{S}}_{\ell_n}^{(b_n)}\cdots\tilde{\mathcal{S}}_{\ell_1}^{(b_1)}. \tag{92}
$$

Then with $\vec{b}_j \in \{0,1\}^{\otimes n_j}$ $(j=1,\ldots,k)$, we obtain

$$\begin{aligned}
&f_{\sigma,k,N-k}\left(\left(\tilde{\mathcal{L}}_{n_1}(\vec{\ell}_1),\ldots,\tilde{\mathcal{L}}_{n_k}(\vec{\ell}_k)\right),\tilde{\mathcal{E}}_N\right)\\
&\quad=\sum_{\vec{b}_1\cdots\vec{b}_k} f_{\sigma,k,N-k}\left(\left(\tilde{\mathcal{S}}^{(\vec{b}_1)}_{n_1}(\vec{\ell}_1),\ldots,\tilde{\mathcal{S}}^{(\vec{b}_k)}_{n_k}(\vec{\ell}_k)\right),\tilde{\mathcal{E}}_N\right)\\
&\quad=\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{r}} P_{\mathrm{MDRIFT}}(\vec{r},N-k)\mathcal{C}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}\\
&\qquad\times\left(\vec{n},\left(\vec{\ell}_1,\ldots\vec{\ell}_k\right),\vec{r}\right),
\end{aligned}\tag{93}$$

where $\mathcal{C}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right),\vec{r}\right)$ is an unitary channel defined by

$$\begin{aligned}
\mathcal{C}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}&\left(\vec{n},\left(\vec{\ell}_1,\ldots\vec{\ell}_k\right),\vec{r}\right)=f_{\sigma,k,N-k}\\
&\times\left(\left(\tilde{\mathcal{S}}^{(\vec{b}_1)}_{n_1}(\vec{\ell}_1),\ldots,\tilde{\mathcal{S}}^{(\vec{b}_k)}_{n_k}(\vec{\ell}_k)\right),\left(\tilde{\mathcal{T}}_{r_1},\ldots,\tilde{\mathcal{T}}_{r_{N-k}}\right)\right).
\end{aligned}\tag{94}$$

We use Eq. (91) in the first equality of Eq. (93) and we use

$$\tilde{\mathcal{E}}_N:=\sum_{r=1}^{L}p_r\tilde{\mathcal{T}}_r \tag{95}$$

and

$$P_{\mathrm{MDRIFT}}(\vec{r},N-k)=p_{r_1}\cdots p_{r_{N-k}} \tag{96}$$

with $\vec{\ell}=\{\ell_1\cdots\ell_{N-k}\}$ in the second equality. By substituting Eqs. (90)–(93), we obtain

$$\begin{aligned}
&\delta q^{(k)}_{\sigma}\left(\vec{n},\left(\vec{\ell}_1,\ldots,\vec{\ell}_k\right)\right)\\
&\quad=\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{r}}P_{\mathrm{MDRIFT}}(\vec{r},N-k)\\
&\qquad\times\mathrm{Tr}\left(\tilde{\mathcal{QC}}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}\left(\vec{n},\left(\vec{\ell}_1,\ldots\vec{\ell}_k\right),\vec{r}\right)(\tilde{\rho}_{\mathrm{init}})\right).
\end{aligned}\tag{97}$$

Finally, combining Eqs. (71), (72), and (80) with Eq. (97),

$$\begin{aligned}
\delta q^{(k)}(\vec{n})&=\binom{N}{k}\frac{\tau^{\sum_{j=1}^{k}n_j}}{n_1!n_2!\cdots n_k!}\frac{1}{\binom{N}{k}}\sum_{\sigma\in S^{\mathrm{sub}}_{N,k}}\delta q^{(k)}_{\sigma}(\vec{n})\\
&=c^{(k)}(\vec{n})\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{s}}(-1)^{\sum_{c=1}^{k}s_c}\delta q^{(k)}\\
&\quad\times\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right),
\end{aligned}\tag{98}$$

where

$$\begin{aligned}
&\delta q^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)\\
&\quad:=\frac{1}{\binom{N}{k}}\sum_{\sigma\in S^{\mathrm{sub}}_{N,k}}\sum_{\vec{\ell}_1,\ldots\vec{\ell}_k}P^{(n_1)}_{s_1}(\vec{\ell}_1)\cdots P^{(n_k)}_{s_k}(\vec{\ell}_k)\\
&\quad\times\sum_{\vec{r}}P_{\mathrm{MDRIFT}}(\vec{r},N-k)\mathrm{Tr}\\
&\quad\times\left(\tilde{\mathcal{QC}}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}\left(\vec{n},\left(\vec{\ell}_1,\ldots\vec{\ell}_k\right),\vec{r}\right)(\tilde{\rho}_{\mathrm{init}})\right),
\end{aligned}\tag{99}$$

and we define the coefficient as

$$c^{(k)}(\vec{n}):=\binom{N}{k}\frac{\tau^{\sum_{j=1}^{k}n_j}}{n_1!n_2!\cdots n_k!}. \tag{100}$$

We can obtain an unbiased estimator of Eq. (99) using Monte Carlo sampling. More specifically, we repeat the following procedure and compute the average of the output; the pth operation works as follows:

(1) Sample σ uniformly from all elements of $S^{\mathrm{sub}}_{N,k}$.
(2) Sample $\vec{\ell}_1,\ldots\vec{\ell}_k$ according to $P^{n_1}_{s_1}(\vec{\ell}_1)\cdots P^{n_k}_{s_k}(\vec{\ell}_k)$. Sample $\vec{r}$ according to $P_{\mathrm{MDRIFT}}(\vec{r},N-k)$.
(3) Evaluate

$$\mathrm{Tr}\left(\tilde{\mathcal{QC}}^{(\vec{b}_1,\ldots,\vec{b}_k)}_{\sigma,k,N-k}\left(\vec{n},\left(\vec{\ell}_1,\ldots\vec{\ell}_k\right),\vec{r}\right)(\tilde{\rho}_{\mathrm{init}})\right), \tag{101}$$

with N_{shot} measurements. We set the result to $\delta\hat{q}^{(k)}_{\sigma_p}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)$.

As in the second-order case, we repeat the above process N_{sample} times and compute the average of $\delta\hat{q}^{(k)}_{\sigma_p}\left(\left(\vec{b}_1,\ldots,\vec{b}_k\right),\vec{s}\right)$ as $\delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots,\vec{b}_k\right),\vec{s}\right)$, which gives the estimate of $\delta q^{(k)}\left(\left(\vec{b}_1,\ldots,\vec{b}_k\right),\vec{s}\right)$. Substituting the estimate into Eq. (98), we obtain the estimate of $\delta q^{(k)}(\vec{n})$ as

$$\begin{aligned}
\delta\hat{q}^{(k)}(\vec{n})&=c^{(k)}(\vec{n})\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{s}}(-1)^{\sum_{c=1}^{k}s_c}\delta\hat{q}^{(k)}\\
&\quad\times\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right).
\end{aligned}\tag{102}$$

We write the above algorithm to estimate $\delta q^{(k)}(\vec{n})$ as the **Evalcorrection** and summarize it in Algorithm 1. By using the **Evalcorrection**, we can construct the algorithm to compute $q^{(K)}$ according to Eq. (66). We write the algorithm as **qSWIFT** and summarize it in Algorithm 2. Since we can tune N_{sample} and N_{shot} depending on $\delta\hat{q}^{(k)}(\vec{n})$, we parametrize them as $N_{\mathrm{sample}}(\vec{n})$ and $N_{\mathrm{shot}}(\vec{n})$. We also write the number of circuits sampled and the number of measurements for calculating $q^{(1)}$ as N^0_{sample} and N^0_{shot}.

ALGORITHM 1. Evalcorrection.

```
Input: $k$, $\vec{n}$, $N$, $N_{\text{sample}}$, and $N_{\text{shot}}$
1: for $\vec{s}$ in $\{0,1\}^{\otimes k}$ do
2:   for $(\vec{b}_1, ..., \vec{b}_k)$ in $(\{0,1\}^{\otimes n_1}, ..., \{0,1\}^{\otimes n_k})$ do
3:     for $p = 1$ to $N_{\text{sample}}$ do
4:       Sample $\sigma$ uniformly from $S^{\text{sub}}_{N,k}$.
5:       Sample $\vec{\ell}_1, \cdots \vec{\ell}_k$ according to $P^{n_1}_{s_1}(\vec{\ell}_1) \cdots P^{n_k}_{s_k}(\vec{\ell}_k)$. Sample $\vec{r}$ according to $P_{\text{MDRIFT}}(\vec{r}, N-k)$.
6:       With $N_{\text{shot}}$ measurements, estimate the following by the quantum circuit and set it to $\delta\hat{q}^{(k)}_{\sigma_p}\left(\left(\vec{b_1}, \cdots \vec{b_k}\right), \vec{s}\right)$:
           $$\mathrm{Tr}\left(\tilde{Q}\mathcal{C}^{(\vec{b}_1,\cdots,\vec{b}_k)}_{\sigma,k,N-k}\left(\vec{n}, \left(\vec{\ell}_1, \cdots \vec{\ell}_k\right), \vec{r}\right)(\tilde{\rho}_{\text{init}})\right),$$
7:     end for
8:     Set $\delta\hat{q}^{(k)}\left(\left(\vec{b_1}, \cdots \vec{b_k}\right), \vec{s}\right) = \frac{1}{N_{\text{sample}}}\sum_{p=1}^{N_{\text{sample}}} \delta\hat{q}^{(k)}_{\sigma_p}\left(\left(\vec{b_1}, \cdots \vec{b_k}\right), \vec{s}\right)$.
9:   end for
10: end for
11: Set
       $$\delta\hat{q}^{(k)}(\vec{n}) = c^{(k)}(\vec{n}) \sum_{\vec{b_1}\cdots\vec{b_k}} \sum_{\vec{s}} (-1)^{\sum_{c=1}^{k} s_c} \delta\hat{q}^{(k)}\left(\left(\vec{b_1}, \cdots \vec{b_k}\right), \vec{s}\right).$$
Output: $\delta\hat{q}^{(k)}(\vec{n})$
```

It should be noted that there is a statistical error in $\hat{q}^{(K)}$ due to the use of Monte Carlo sampling and the limited number of measurements. To reduce the statistical error, we need to increase $N_{\text{sample}}(\vec{n})$, $N_{\text{shot}}(\vec{n})$, N^0_{sample}, and N^0_{shot}. We show how the total number of quantum circuit runs scales with the order in Appendix B, even though our focus in this paper is on discussing the reduction of systematic error and not the statistical error. In the discussion, we show that the dominant source of the quantum circuit runs comes from the estimation of $\Delta\hat{q}^{(k)}(\vec{n})$ with limited $\vec{n}$ and that the cost for $\Delta\hat{q}^{(k)}(\vec{n})$ with other $\vec{n}$ is negligible when N is large; consequently, the total number of quantum circuit runs scales only less than quadratically with K.

In this section, we present the high-order qSWIFT method by expanding the unitary channel $\mathcal{U}$, where the systematic error decreases exponentially with the order parameter. We wish to highlight that it is possible to construct an all-order qSWIFT, which completely eliminates systematic error, as shown in Appendix C. To avoid exponentially large statistical errors, we must set $N \in \mathcal{O}\left((\lambda t)^2\right)$. The primary drawback of the all-order qSWIFT is the lack of an upper bound for the number of swift operators. Therefore, the higher-order qSWIFT is more appropriate for use in quantum devices with limited gate-operation capacity. The details of the all-order qSWIFT are described in Appendix C.

C. Note on the previous higher-order randomized method

We mention that the work by Wan *et al.* [19] introduces a higher-order randomized technique for phase estimation, where the task is to compute $\mathrm{Tr}[\rho e^{\mathrm{i}Ht}]$ with ρ as the initial state. They propose a randomized approach to estimate $\mathrm{Tr}\left[\rho e^{\mathrm{i}Ht}\right]$ that leverages the linear combination of unitaries (LCU) scheme. Methodologically, they express $e^{\mathrm{i}Ht}$ as a weighted sum of unitary operations and select a term for sampling based on a specific probability distribution.

While the study in Ref. [19] is primarily focused on phase estimation, it could be extended to a higher-order

ALGORITHM 2. qSWIFT.

```
Input: $K$, $N$, $N_{\text{sample}}(\vec{n})$, $N_{\text{shot}}(\vec{n})$, $N^0_{\text{sample}}$, and $N^0_{\text{shot}}$
1: Estimate $\mathrm{Tr}(Q\mathcal{E}^N_N(\rho_{\text{init}}))$ by sampling $N^0_{\text{sample}}$ circuits and measuring each circuit $N^0_{\text{shot}}$ times; set the result to $\hat{q}^{(1)}$.
2: Set delta = 0.
3: for $\xi = 2, ..., 2K-2$ do
4:   for $k = 1, ..., K$ do
5:     for $\vec{n} \in G_2(k, \xi)$ do
6:       Add the result of Evalcorrection $(k, \vec{n}, N, N_{\text{sample}}(\vec{n}), N_{\text{shot}}(\vec{n}))$ to delta.
7:     end for
8:   end for
9: end for
10: Set $\hat{q}^{(K)} = \hat{q}^{(1)}$ + delta.
Output: $\hat{q}^{(K)}$
```

randomization approach to estimate $q = \mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}})) = \mathrm{Tr}(Qe^{\mathrm{i}Ht}\rho_{\mathrm{init}}e^{-\mathrm{i}Ht})$ by expanding $e^{\mathrm{i}Ht}$ and $e^{-\mathrm{i}Ht}$ in a similar fashion to the LCU technique, which we refer to as the LCU-based method. Detailed in Appendix D, this method, as in the all-order qSWIFT presented in Appendix C, is free from systematic error. Nonetheless, the LCU-based method demands that all the $\mathcal{O}((\lambda t)^2)$ time-evolution operations must be controlled by the ancilla qubit. In contrast, qSWIFT only requires swift operators to interact with the ancilla qubit. Appendix D illustrates how the demand for $\mathcal{O}((\lambda t)^2)$ time evolutions in the LCU-based method could lead to a substantial increase in the number of control gates in comparison to qSWIFT for certain problems.

V. NUMERICAL EXPERIMENTS

In this section, we present two numerical simulations of qSWIFT. In Sec. V A, we show the asymptotic behavior of qSWIFT and compare it with Trotter-Suzuki decomposition and qDRIFT. In Sec. V B, we perform a numerical experiment with the hydrogen-molecule Hamiltonian and compare its performance with those of the previous algorithms.

A. Asymptotic behavior

We compute the required number of gates N to achieve a given systematic error for each evolution time and each algorithm: qSWIFT, Trotter-Suzuki decomposition, and qDRIFT. To clarify the difference from the original qDRIFT, we use the three molecules used in the numerical experiment of the original paper [5]: propane, ethane, and carbon dioxide.

For the qSWIFT algorithm, we utilize the third-order ($K = 3$) qSWIFT. We also use the sixth-order qSWIFT ($K = 6$) as a reference. For the Trotter-Suzuki decomposition, we use both the deterministic and the randomized methods of the first, second, and fourth order. For calculating the systematic error of qSWIFT, we use the bound in Eq. (A22), derived in Appendix A 2. For the error of qDRIFT and the Trotter-Suzuki decompositions, we utilize the same upper-bound formulas as in the literature [5] (see Appendixes B and C of Ref. [5]).

In Fig. 3, we show the asymptotic behavior of N for each time to achieve the systematic error $\varepsilon = 0.001$, which includes three subfigures corresponding to the three molecules: propane with the STO-3G basis [Fig. 3(a)], ethane with the 6-31G basis [Fig. 3(b)], and carbon dioxide with the 6-31G basis [Fig. 3(c)]. To generate each molecule Hamiltonian, we use OpenFermion [20]. For each figure, we show the third-order qSWIFT by the pink line (qSWIFT-3), the sixth-order qSWIFT by the pink dotted line (qSWIFT-6), and the qDRIFT by the black line (qDRIFT). For the Trotter-Suzuki decomposition, the best of the deterministic methods among the first, second, and fourth orders is shown by the gray dotted line [TS (Best)] and the best of the randomized methods is shown by the green dotted line [RTS (Best)].

We see that the qSWIFT algorithms outperform qDRIFT for all t in the sense that the required number of gates, N, is more than 10 times smaller in the qSWIFT algorithms than in qDRIFT. While the original qDRIFT reduces the number of gates in the region $t \lesssim 10^8$ compared to the Trotter-Suzuki decompositions, the qSWIFT algorithms realize a further reduction of the gates. Note that the improvement from the third-order qSWIFT to the sixth-order qSWIFT is not as large as that from qDRIFT to the third-order qSWIFT. Since there is a trade-off between the order and the number of quantum circuit runs, as we

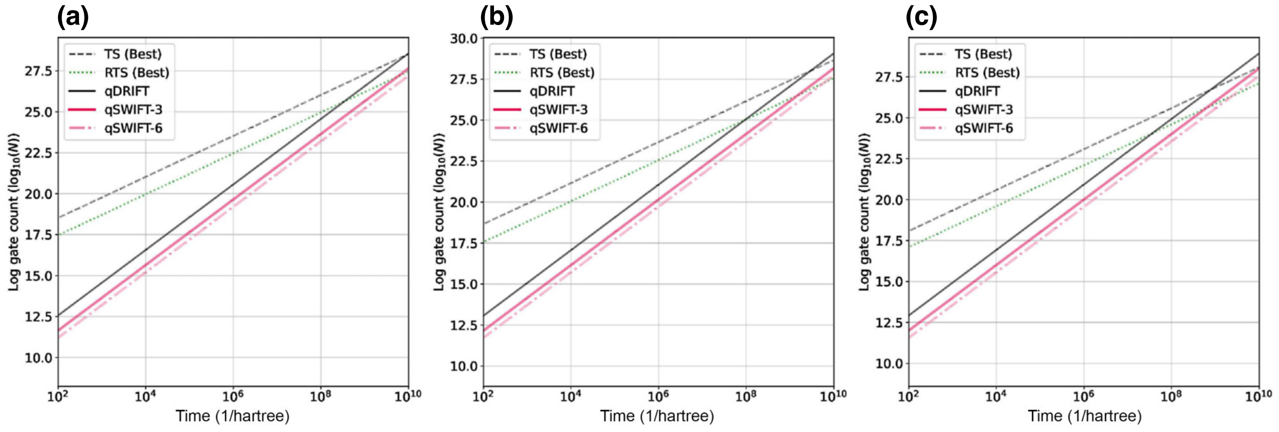

FIG. 3. The asymptotic behavior of N for each time for each molecule to achieve the systematic error $\varepsilon = 0.001$. The three subfigures correspond to the three molecules: (a) propane with the STO-3G basis, (b) ethane with the 6-31G basis, and (c) carbon dioxide with the 6-31G basis. We show the third-order qSWIFT by the pink line (qSWIFT-3), the sixth-order qSWIFT by the pink dotted line (qSWIFT-6), and the qDRIFT by the black line (qDRIFT). For the Trotter-Suzuki decomposition, the best of the deterministic methods among the first, second, and fourth orders is shown by the gray dotted line [TS (Best)], and the best of the randomized methods is shown by the green dotted line [RTS (Best)].

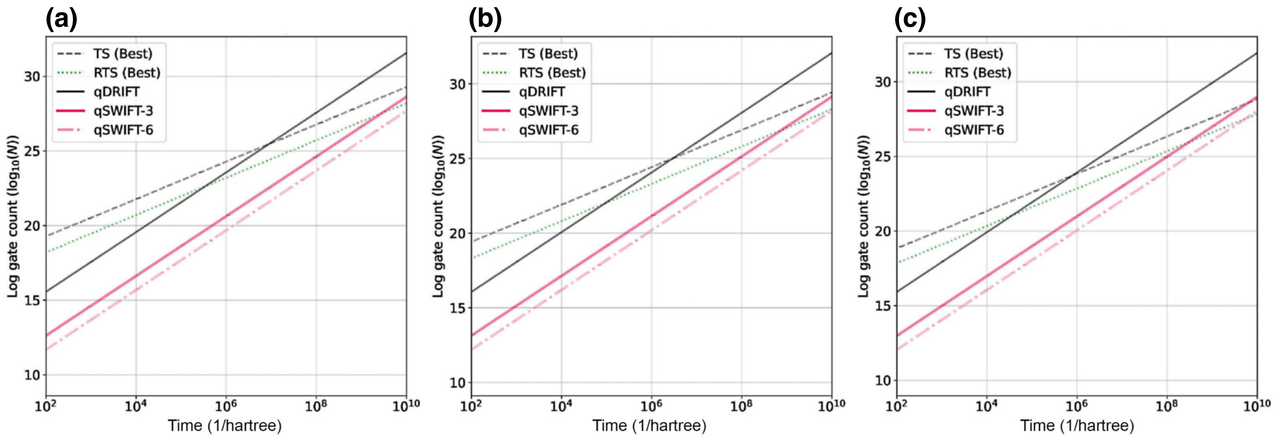

FIG. 4. The asymptotic behavior to achieve $\varepsilon = 10^{-6}$. The other settings are the same as in Fig. 3.

discuss in Appendix B, it may be better to utilize the third-order qSWIFT rather than the sixth-order qSWIFT, though it depends on the features of the quantum devices.

Figure 4 is the same as Fig. 3 except that the required systematic error is $\varepsilon = 10^{-6}$. In this case, where more precise time evolution is necessary, the merit of using qSWIFT is much clearer. In qSWIFT-3 (qSWIFT-6), the required number of gates for each time is almost 1000 (10 000) times smaller than that of qDRIFT. The region in which qDRIFT has an advantage over the Trotter-Suzuki decomposition is very limited ($t \lesssim 10^5$–10^6) due to the bad scaling of qDRIFT in terms of ε. In contrast, the region in which qSWIFT has the advantage over the Trotter-Suzuki decomposition does not change much from the case of $\varepsilon = 0.001$ ($t \lesssim 10^9$–10^{10}). This result shows the merit of our algorithm; in the case of qDRIFT, we need to increase the number of gates to reduce the systematic error but in the case of our qSWIFT, it can be reduced just by increasing the order parameter of the algorithm.

We note that as we state at the end of Sec. II A, we can further improve the bound for the deterministic Trotter-Suzuki (TS) decomposition by using the commutator bounds [3], represented as the commutator relation. It should also be noted that the derivation of upper bounds for qDRIFT and qSWIFT involves many triangular inequalities that are loosely bounded. Therefore, the upper bounds for the randomized compiling methods could also be improved by considering relations between operators, presenting a promising direction for future research.

B. Simulation of the hydrogen molecule

We estimate $\mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}}))$ by using the qSWIFT algorithm described in Algorithm 2 with an observable Q, a time evolution $\mathcal{U}$, and an input state ρ_{init}. For $\mathcal{U}$, we implement the time evolution with the hydrogen-molecule Hamiltonian with the 6-31G basis and $t = 1$. Again, we use OpenFermion [20] to generate the molecule Hamiltonian. We utilize the Bravyi-Kitaev transformation [21] for transforming the fermionic operators to Pauli operators. The number of terms in the Hamiltonian L is 184. The generated Hamiltonian has eight qubits and, therefore, we use a system with nine qubits (including one ancilla qubit). As for the observable Q, we choose $Q = ZIIIIIII$ and, as for the input state, we choose $\rho_{\mathrm{init}} = |+\rangle^{\otimes 8}$. For comparison, we also estimate $\mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}}))$ by using qDRIFT and the Trotter-Suzuki decomposition. As the input parameters of qSWIFT, we set $N^0_{\mathrm{shot}} = N_{\mathrm{shot}}(\vec{n}) = 100$, $N^0_{\mathrm{sample}} = 400\,000$, and $N_{\mathrm{sample}}(\vec{n}) = C(\vec{n}) \times N^0_{\mathrm{sample}}$. For qDRIFT and the randomized Trotter-Suzuki decomposition, we sample N^0_{sample} quantum circuits and perform N^0_{shot} measurements for each circuit. For the deterministic Trotter-Suzuki decomposition, we perform $N^0_{\mathrm{shot}} \times N^0_{\mathrm{sample}}$ measurements. For the quantum circuit simulation, we use the QULACS software package [22].

In Fig. 5, we show the estimation error of $\mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}}))$ for each number of gates, N, for each method. In plotting each point, we perform six trials and show the mean value and the standard deviation of the mean. For qSWIFT, we show the result of the second order (qSWIFT-2) with the purple line and the third order (qSWIFT-3) with the pink line. The result of qDRIFT (qDRIFT) is plotted with the black line. For the Trotter-Suzuki decomposition, we show the first- and second-order results. The minimum N values for the first and second Trotter-Suzuki decompositions are 184 and 368, respectively (1840 for the fourth order). For the deterministic Trotter-Suzuki decomposition, the first-order result (TS-1st) is plotted with the dotted gray line and the second-order result (TS-2nd) is plotted with the dotted green line. For the randomized Trotter-Suzuki decomposition, the first-order result (RTS-1st) is plotted with the dotted yellow line and the second-order result (RTS-2nd) is plotted with the dotted blue line.

We see that the qSWIFT algorithms outperform the other methods. In particular, the required N value for the third-order qSWIFT for achieving $\varepsilon \sim 0.001$ is almost 10 times smaller than that for qDRIFT, which is consistent

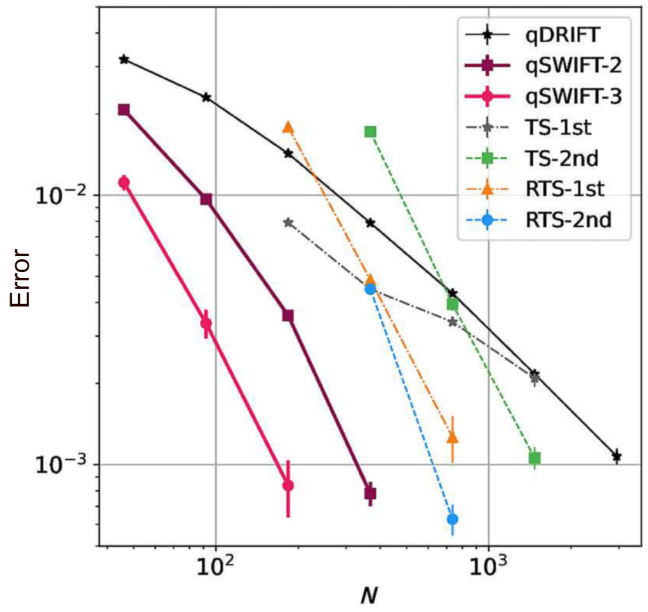

FIG. 5. The estimation error of $\mathrm{Tr}(Q\mathcal{U}(\rho_{\mathrm{init}}))$ for each number of gates N for each method with $Q = ZIIIIIII$ and $\rho_{\mathrm{init}} = |+\rangle^{\otimes 8}$. The time evolution is performed by the hydrogen-molecule Hamiltonian with the 6-31G basis transformed by the Bravyi-Kitaev transformation. The evolution time is set to be $t = 1$. In plotting each point, we perform six trials and show the mean value and the standard deviation of the mean. For qSWIFT, we show the result of the second order (qSWIFT-2) with the purple line and the third order (qSWIFT-3) with the pink line. The result of the qDRIFT algorithm (qDRIFT) is plotted with the black line. For the deterministic Trotter-Suzuki decomposition, the first-order result (TS-1st) is plotted with the dotted gray line and the second-order result (TS-2nd) is plotted with the dotted green line. For the randomized Trotter-Suzuki decomposition, the first-order result (RTS-1st) is plotted with the dotted yellow line and the second-order result (RST-2nd) is plotted with the dotted blue line.

with the asymptotic behavior shown in Fig. 3. Consequently, even in the region in which the Trotter-Suzuki decomposition works better than qDRIFT, the qSWIFT algorithms outperform the Trotter-Suzuki decompositions.

VI. DISCUSSION AND CONCLUSIONS

Hamiltonian simulation is a crucial subroutine of various quantum algorithms. Approaches based on product formulas are favored in practice, due to their simplicity and ancilla-free nature. There are two representative product-formula-based methods: Trotter-Suzuki decompositions and qDRIFT [5]. The Trotter-Suzuki decompositions have the issue that the number of gates depends on the number of terms in the Hamiltonian, at least linearly. In contrast, qDRIFT avoids the dependency on the number of terms but has the issue that the number of gates is dependent on the systematic error ε as $O(1/\varepsilon)$.

In this paper, we propose qSWIFT, a high-order randomized algorithm having the advantages of both the Trotter Suzuki decompositions and qDRIFT, in the sense that its gate count is independent of the number of terms and that the gate count is asymptotically optimal with respect to the precision. We construct the qSWIFT channel and bound the systematic error by the diamond norm. We prove that the qSWIFT channel satisfies the required precision, which decreases exponentially with the order parameter in terms of the diamond norm. Then we construct the algorithm by applying the qSWIFT channel to estimate given physical quantities. The algorithm requires a system as simple as qDRIFT; it requires just one ancilla qubit and only the time-evolution operators and the swift operators constructible with the gate in Fig. 2(b) are necessary to construct the quantum circuits. Our numerical demonstration shows that qSWIFT outperforms qDRIFT with respect to the number of gates for the required precision. In particular, when high precision is required, there is a significant advantage of using qSWIFT; the number of gates in the third-order (sixth-order) qSWIFT is 1000 (10 000) times smaller than qDRIFT to achieve the systematic error of $\varepsilon = 10^{-6}$.

As a future direction, it is beneficial to perform case studies to investigate the performance of qSWIFT in specific problems. In particular, the literature [23] points out that qDRIFT does not perform well in phase-estimation problems, contrary to the originally expectations [5], due to the relatively large systematic error for a given number of gates. In contrast, since qSWIFT can successfully reduce systematic error by increasing the order parameter, we expect that the performance in phase estimation is significantly improved with qSWIFT. Also, as we note in Sec. I, there are algorithms using many ancilla qubits [6–11] that achieve a better asymptotic gate scaling with respect to λ and t, though their constant prefactors is relatively large compared to qDRIFT and qSWIFT. Comparing qSWIFT with those algorithms and discussing the advantages of each algorithm in specific problems is a promising direction for future work.

Whether we can improve the scaling with respect to λt within the framework of qDRIFT and qSWIFT is another important open question. Unlike qDRIFT and qSWIFT, where the number of gates required for a given error scales quadratically with λt, the scaling with t can be improved by increasing the order parameter in the Trotter-Suzuki decomposition. The development of methods to enhance the λt scaling would further expedite the realization of practical Hamiltonian simulations.

ACKNOWLEDGMENTS

We thank Nathan Wiebe for helpful discussions. K.N. acknowledges the support of Grant-in-Aid for the Japan Society for the Promotion of Science (JSPS) Research

Fellow Grant No. 22J01501. A.A.-G. acknowledges support from the Canada 150 Research Chairs program and the Canadian Institute for Advanced Research (CIFAR). A.A.-G. acknowledges the generous support of Anders G. Frøseth.

APPENDIX A: BOUND FOR THE SYSTEMATIC ERRORS IN QSWIFT CHANNELS

1. Error bound for the second-order qSWIFT channel

In this subsection, we prove the error bound given in Lemma 1 for our second-order qSWIFT channel. To this end, we utilize the triangle and submultiplicative properties of the diamond norm. Namely, we utilize

$$||\mathcal{A}+\mathcal{B}||_\diamond \le ||\mathcal{A}||_\diamond + ||\mathcal{B}||_\diamond, \tag{A1}$$

$$||\mathcal{A}\mathcal{B}||_\diamond \le ||\mathcal{A}||_\diamond ||\mathcal{B}||_\diamond, \tag{A2}$$

for given channels $\mathcal{A}$ and $\mathcal{B}$.
The diamond distance between the exact time evolution and the qSWIFT channel can be evaluated as

$$\begin{aligned} d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right) &= \frac{1}{2}||M_{1,N-1}\left(\Delta_3,\mathcal{E}_N\right) + \sum_{k=2}^{N} M_{k,N-k}\left(\Delta_2,\mathcal{E}_N\right)||_\diamond \\ &\le \frac{1}{2}||M_{1,N-1}\left(\Delta_3,\mathcal{E}_N\right)||_\diamond + \frac{1}{2}\sum_{k=2}^{N}||M_{k,N-k}\left(\Delta_2,\mathcal{E}_N\right)||_\diamond \\ &\le \frac{1}{2}\sum_{n=3}^{\infty}\frac{\tau^n}{n!}||M_{1,N-1}(\mathcal{L}^{(n)},\mathcal{E}_N)||_\diamond + \frac{1}{2}\sum_{k=2}^{N}\sum_{n_1,\dots n_k=2}^{\infty}\frac{\tau^{\sum_j^k n_j}}{n_1!\cdots n_k!}||M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)},\dots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)||_\diamond, \end{aligned} \tag{A3}$$

where we use Eqs. (A1) and (A2) to show the inequality. By the definition of the mixture function $M_{k,N-k}$ in Eq. (19) and using the triangle and submultiplicative properties of the diamond norm, we obtain

$$\begin{aligned} ||M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)},\dots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)||_\diamond &\le \sum_{\sigma\in S_{N,k}^{\mathrm{sub}}}||f_{\sigma,k,N-k}\left(\left(\mathcal{L}^{(n_1)},\dots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)||_\diamond \\ &\le \binom{N}{k}\prod_{j=1}^{k}||\mathcal{L}^{(n_j)}||_\diamond ||\mathcal{E}_N||_\diamond^{N-k} \\ &\le \binom{N}{k}2^{k+\sum_{j=1}^k n_j}. \end{aligned} \tag{A4}$$

To show the last inequality, we use

$$||\mathcal{L}^{(n)}||_\diamond \le ||\mathcal{L}||_\diamond^n + \sum_{\ell=1}^{L} p_\ell ||\mathcal{L}_\ell||_\diamond^n \le 2^{n+1}, \tag{A5}$$

which holds since $||\mathcal{L}||_\diamond \le 2$ and $||\mathcal{L}_\ell||_\diamond \le 2$.
By using Eq. (A4),

$$\begin{aligned} d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right) &\le \frac{1}{2}\sum_{n=3}^{\infty}\frac{\tau^n}{n!}\binom{N}{1}2^{n+1} + \frac{1}{2}\sum_{k=2}^{N}\sum_{n_1,\dots n_k=2}^{\infty}\frac{\tau^{\sum_j^k n_j}}{n_1!\cdots n_k!}\binom{N}{k}2^{k+\sum_{j=1}^k n_j} \\ &= \frac{1}{2}\sum_{n=3}^{\infty}\frac{N\tau^n}{n!}2^{n+1} + \frac{1}{2}\sum_{k=2}^{N}\sum_{n_1,\dots n_k=2}^{\xi}\sum_{\xi=4}^{\infty}\delta\left[\xi,\sum_{j=1}^{k}n_j\right]\frac{\tau^{\xi}}{n_1!\cdots n_k!}\binom{N}{k}2^{k+\xi}, \end{aligned} \tag{A6}$$

where, in the second line, we use

$$\sum_{\xi=4}^{\infty}\delta\left[\xi,\sum_{j=1}^{k}n_j\right]=1, \tag{A7}$$

which holds for a fixed set of integers $\{n_j\}_{j=1}^{k}$ with $n_j\geq 2$ and $k\geq 2$. By using that $1/n\leq 1/2$ in the summand of the first term and $1/n_j\leq 1/2$ in the summand of the second term, we obtain

$$\begin{aligned}d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right)&\leq\frac{1}{2}\sum_{n=3}^{\infty}(2\tau)^nN+\frac{1}{2}\sum_{\xi=4}^{\infty}\sum_{k=2}^{N}(2\tau)^\xi\binom{N}{k}\\&\quad\times\sum_{n_1,\ldots n_k=2}^{\xi}\delta\left[\xi,\sum_{j=1}^{k}n_j\right]\\&=\frac{1}{2}\sum_{n=3}^{\infty}(2\tau)^nN+\frac{1}{2}\sum_{\xi=4}^{\infty}\sum_{k=2}^{\lfloor\xi/2\rfloor}(2\tau)^\xi\binom{N}{k}\\&\quad\times\sum_{n_1,\ldots n_k=2}^{\xi}\delta\left[\xi,\sum_{j=1}^{k}n_j\right],\end{aligned} \tag{A8}$$

with $\lfloor x\rfloor$ as the largest integer that does not exceed a real value x, where we use the fact that the summand of the second term vanishes if $k\geq\lfloor\xi/2\rfloor$. Using

$$\sum_{n_1,\ldots,n_k=2}^{\xi}\delta\left[\xi,\sum_{j=1}^{k}n_j\right]\leq\xi^k, \tag{A9}$$

we obtain

$$\begin{aligned}d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right)&\leq\frac{1}{2}\sum_{n=3}^{\infty}(2\tau)^n\binom{N}{1}+\frac{1}{2}\sum_{\xi=4}^{\infty}\sum_{k=2}^{\lfloor\xi/2\rfloor}(2\tau)^\xi\binom{N}{k}\xi^k\\&\leq\frac{1}{2}\sum_{n=3}^{\infty}(2\tau)^nN+\frac{1}{2}\sum_{\xi=4}^{\infty}(2\tau)^\xi N^{\lfloor\xi/2\rfloor}\sum_{k=2}^{\lfloor\xi/2\rfloor}\frac{\xi^k}{k!}\\&\leq\frac{1}{2}\sum_{n=3}^{\infty}(2\tau)^nN+\frac{1}{2}\sum_{\xi=4}^{\infty}(2e\tau)^\xi N^{\lfloor\xi/2\rfloor}\\&\leq\frac{1}{2}\sum_{\xi=3}^{\infty}(2e\tau)^\xi N^{\lfloor\xi/2\rfloor},\end{aligned} \tag{A10}$$

where to show the third inequality, we use

$$\sum_{k=1}^{\lfloor\xi/2\rfloor}\frac{\xi^k}{k!}\leq\sum_{k=0}^{\infty}\frac{\xi^k}{k!}=e^\xi. \tag{A11}$$

Now let us evaluate $\frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2e\tau)^\xi N^{\lfloor\xi/2\rfloor}$ (in the current case, $K=2$). It can be evaluated, depending on whether ξ is odd ($\xi=2m$ with m as an integer) or even ($\xi=2m-1$), as

$$\begin{aligned}&\sum_{\xi=2K-1}^{\infty}\frac{1}{2}(2e\tau)^\xi N^{\lfloor\xi/2\rfloor}\\&\quad\leq\frac{1}{2}\sum_{m=K}^{\infty}(2e\tau)^{2m}N^m+\frac{1}{2}\sum_{m=K}^{\infty}(2e\tau)^{2m-1}N^{m-1}\\&\quad=\frac{1}{2}\left(1+\frac{1}{2eN\tau}\right)\sum_{m=K}^{\infty}(2e\tau)^{2m}N^m\\&\quad=\eta(\lambda t,N)\left(\frac{(2e\lambda t)^2}{N}\right)^K,\end{aligned} \tag{A12}$$

as far as

$$\frac{(2e\lambda t)^2}{N}<1, \tag{A13}$$

where

$$\eta(x,N):=\frac{1}{2}\left(1+\frac{1}{2ex}\right)\frac{1}{1-(2ex)^2/N}. \tag{A14}$$

By setting $K=2$, we obtain the bound

$$d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right)\leq\eta(\lambda t,N)\left(\frac{(2e\lambda t)^2}{N}\right)^2. \tag{A15}$$

In the reasonable parameter range $1\leq\lambda t\leq\sqrt{N}/2\sqrt{2}e$ (where the latter inequality holds by choosing suitable N), it holds that $\eta(\lambda t,N)\leq 3/2$ and therefore

$$d_\diamond\left(\mathcal{U},\mathcal{E}^{(2)}\right)\leq\frac{3}{2}\left(\frac{(2e\lambda t)^2}{N}\right)\in\mathcal{O}\left(\left(\frac{(\lambda t)^2}{N}\right)^2\right). \tag{A16}$$

2. Error bound for the higher-order qSWIFT channels

We now prove the error bound given in Lemma 2 for the Kth-order qSWIFT channel. By the definition of this channel in Eq. (62), we have

$$d_\diamond\left(\mathcal{U},\mathcal{E}^{(K)}\right)=\frac{1}{2}||\Phi^{(\infty)}-\mathcal{E}^{(K)}||_\diamond \tag{A17}$$

$$\begin{aligned}&=\frac{1}{2}\sum_{\xi=2K-1}^{\infty}\tau^\xi\sum_{k=1}^{N}\sum_{n_1,\ldots,n_k=2}^{\xi}\frac{1}{n_1!n_2!\cdots n_k!}\delta\left[\xi,\sum_{j=1}^{k}n_j\right]\\&\quad\times M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)},\ldots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)\end{aligned} \tag{A18}$$

$$\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}\tau^{\xi}\sum_{k=1}^{N}\sum_{n_1,\ldots,n_k=2}^{\xi}\frac{1}{n_1!n_2!\cdots n_k!}\delta\left[\xi,\sum_{j=1}^{k}n_j\right] \times ||M_{k,N-k}\left(\left(\mathcal{L}^{(n_1)},\ldots,\mathcal{L}^{(n_k)}\right),\mathcal{E}_N\right)||_{\diamond}, \tag{A19}$$

where we have used the triangular inequality Eq. (A1).

By substituting Eq. (A4) into Eq. (A17), we obtain

$$\begin{aligned} d_{\diamond}\left(\mathcal{U},\mathcal{E}^{(K)}\right) &\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}\tau^{\xi}\sum_{k=1}^{N}\sum_{n_1,\ldots,n_k=2}^{\xi}\frac{1}{n_1!n_2!\cdots n_k!}\delta \times\left[\xi,\sum_{j=1}^{k}n_j\right]\binom{N}{k}2^{\xi+k} \\ &\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2\tau)^{\xi}\sum_{k=1}^{N}\binom{N}{k}\sum_{n_1,\ldots,n_k=2}^{\xi}\delta \times\left[\xi,\sum_{j=1}^{k}n_j\right] \\ &= \frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2\tau)^{\xi}\sum_{k=1}^{\lfloor\xi/2\rfloor}\binom{N}{k}\sum_{n_1,\ldots,n_k=2}^{\xi}\delta \times\left[\xi,\sum_{j=1}^{k}n_j\right]. \end{aligned} \tag{A20}$$

We use $1/n_j \leq 1/2$ for ($n_j \geq 2$) in the second inequality and to show the third equality, we use the fact that summands with $k > \lfloor\xi/2\rfloor$ vanish. Using Eq. (A9), we obtain

$$\begin{aligned} d_{\diamond}\left(\mathcal{U},\mathcal{E}^{(K)}\right) &\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2\tau)^{\xi}\sum_{k=1}^{\lfloor\xi/2\rfloor}\binom{N}{k}\xi^{k} \\ &\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2\tau)^{\xi}N^{\lfloor\xi/2\rfloor}\sum_{k=1}^{\lfloor\xi/2\rfloor}\frac{\xi^{k}}{k!} \\ &\leq \frac{1}{2}\sum_{\xi=2K-1}^{\infty}(2e\tau)^{\xi}N^{\lfloor\xi/2\rfloor}, \end{aligned} \tag{A21}$$

where to show the last inequality, we use Eq. (A11). By using Eq. (A12), we obtain

$$d_{\diamond}\left(\mathcal{U},\mathcal{E}^{(K)}\right) \leq \eta(\lambda t,N)\left(\frac{(2e\lambda t)^2}{N}\right)^{K}. \tag{A22}$$

In the reasonable parameter range, $1 \leq \lambda t \leq \sqrt{N}/2\sqrt{2}e$ again, it holds that $\eta(\lambda t,N) \leq 3/2$ and therefore

$$d_{\diamond}\left(\mathcal{U},\mathcal{E}^{(K)}\right) \leq \frac{3}{2}\left(\frac{(2e\lambda t)^2}{N}\right) \in \mathcal{O}\left(\left(\frac{(\lambda t)^2}{N}\right)^{K}\right). \tag{A23}$$

Note that in drawing Figs. 3 and 4, we have used the formula given in Eq. (A22).

APPENDIX B: STATISTICAL ERROR

Due to the sampling error and the shot noise, there is a statistical error in $\delta\hat{q}^{(k)}(\vec{n})$. Let us estimate the statistical error in the following. For simplicity, we set $N_{\text{shot}}(\vec{n}) = 1$. Let $\Delta_q\left((\vec{b}_1,\vec{b}_2,\ldots,\vec{b}_k),\vec{s}\right)$ as the statistical error of $\delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)$. Then, from the central limit theorem, it behaves as

$$\Delta_q\left((\vec{b}_1,\vec{b}_2,\ldots,\vec{b}_k),\vec{s}\right) \sim \frac{\mathrm{Var}\left(\delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)\right)}{\sqrt{N_{\text{sample}}(\vec{n})}}, \tag{B1}$$

where Var(A) denotes the variance of the variable A. In the rest of the section, we use "$\sim$" with the same meaning. Since

$$-1 \leq \delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right) \leq 1, \tag{B2}$$

from Popoviciu's inequality on variances, it holds that $\mathrm{Var}\left(\delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)\right) \leq 1$, and

$$\Delta_q\left((\vec{b}_1,\vec{b}_2,\ldots,\vec{b}_k),\vec{s}\right) \lesssim \frac{1}{\sqrt{N_{\text{sample}}(\vec{n})}}. \tag{B3}$$

Using the fact that each $\delta\hat{q}^{(k)}\left(\left(\vec{b}_1,\ldots\vec{b}_k\right),\vec{s}\right)$ is independent, we can estimate the statistical error of $\delta\hat{q}^{(k)}(\vec{n})$ as

$$\begin{aligned} |\delta q^{(k)}(\vec{n}) - \delta\hat{q}^{(k)}(\vec{n})| &\leq c^{(k)}(\vec{n})\sqrt{\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{s}}\left[\Delta_q\left((\vec{b}_1,\vec{b}_2,\ldots,\vec{b}_k),\vec{s}\right)\right]^2} \\ &\lesssim c^{(k)}(\vec{n})\sqrt{\frac{2^{\sum_j n_j+k}}{N_{\text{sample}}(\vec{n})}}, \end{aligned} \tag{B4}$$

where to show the second inequality, we use $\sum_{\vec{b}_1\cdots\vec{b}_k}\sum_{\vec{s}} 1 = 2^{\sum_j n_j+k}$. In other words, the statistical error is bounded as

$$|\delta q^{(k)}(\vec{n}) - \delta\hat{q}^{(k)}(\vec{n})| \le \varepsilon, \tag{B5}$$

if we set

$$N_{\text{sample}}(\vec{n}) \sim \frac{[c^{(k)}(\vec{n})]^2 \times 2^{\sum_j n_j+k}}{\varepsilon^2}. \tag{B6}$$

Let $N^{(k)}_{\text{total}}(\vec{n})$ be the total circuit run for calculating $\delta\hat{q}^{(k)}(\vec{n})$. Then, it holds that

$$N_{\text{total}}(\vec{n}) = 2^{\sum_j n_j+k} \times N_{\text{sample}}(\vec{n}) \sim \frac{[c^{(k)}(\vec{n})]^2 \times 2^{2\sum_j n_j+2k}}{\varepsilon^2}. \tag{B7}$$

Let us further clarify the implication of Eq. (B7). Recall that in the qSWIFT algorithm, we only compute $\delta\hat{q}^{(k)}(\vec{n})$ if

$$\sum_{j=1}^{k} n_j = \xi, \quad n_j \ge 2\ (\forall j). \tag{B8}$$

When the condition in Eq. (B8) is satisfied, it holds that

$$c^{(k)}(\vec{n}) \le N^k \left(\frac{\tau^\xi}{2^k}\right), \tag{B9}$$

where $\xi \ge 2k$ (the equality holds when $n_j = 2$ for all j). Conversely,

$$c^{(k)}(\vec{n}) \le \begin{cases} \left(\frac{(\lambda t)^2}{2N}\right)^{\frac{\xi}{2}}, & \text{if } n_j = 2 \text{ for all } j, \\ \sqrt{\frac{2}{N}}\left(\frac{(\lambda t)^2}{2N}\right)^{\frac{\xi}{2}}, & \text{otherwise,} \end{cases} \tag{B10}$$

meaning that $c^{(k)}(\vec{n})$ is suppressed by the factor $\sqrt{2/N}$ unless $n_j = 2$ is satisfied for all j. Therefore, the dominant source of the quantum circuit runs comes from the calculation of $\delta\hat{q}^{(k)}(\vec{n})$ with $n_j = 2$ for all j; and the number of measurements for calculating $\delta\hat{q}^{(k)}(\vec{n})$ with other $\vec{n}$ asymptotically becomes negligible as N becomes large. In the asymptotic limit, the total number of quantum circuit runs N_{total} for computing all terms in $q^{(K)}$ within the error ε in $2K$th-order qSWIFT can be estimated as

$$\begin{aligned} N_{\text{total}} &\approx N^0_{\text{sample}} + N^{(1)}_{\text{total}}(\{2\}) + N^{(2)}_{\text{total}}(\{2,2\}) \\ &\quad + \cdots N^{(K)}_{\text{total}}(\{2,\ldots 2\}) \end{aligned} \tag{B11}$$

$$\sim \frac{1}{\varepsilon^2}\sum_{k=0}^{K}\left(\frac{(2\lambda t)^2}{N}\right)^\xi, \tag{B12}$$

which includes the number of samples N^0_{sample} to compute $q^{(1)}$. The approximation in the first line denotes the asymptotic limit. To obtain the last expression, we use Eq. (B7) and also use that if $N^0_{\text{shot}} = 1$, the required number of samples to reduce the statistical error of $q^{(1)}$ within ε is

$$N^0_{\text{sample}} \sim \frac{1}{\varepsilon^2} \tag{B13}$$

from the central limit theorem. To reduce the statistical error of $q^{(K)}$ to less than $\varepsilon_{\text{total}}$, we should set $\varepsilon = \varepsilon_{\text{total}}/\sqrt{K+1}$, and therefore,

$$N_{\text{total}} \sim \frac{K+1}{\varepsilon^2_{\text{total}}}\sum_{k=0}^{K}\left(\frac{(2\lambda t)^2}{N}\right)^\xi. \tag{B14}$$

Thus, if we fix $\varepsilon_{\text{total}}$, N_{total} scales less than quadratically with K as far as $(2\lambda t)^2/N < 1$.

APPENDIX C: ALL-ORDER QSWIFT

In the main text, we have constructed the second-order and higher-order versions of qSWIFT, which include systematic errors dependent on the order parameter. In this section, we demonstrate that we can create an all-order version of qSWIFT that has no systematic error.

The expectation value of an observable can be expanded as

$$\begin{aligned} \mathrm{Tr}\,(Q\mathcal{U}(\rho_{\text{init}})) &= \mathrm{Tr}\left(Q(\mathcal{E}_N + \Delta_2)^N(\rho_{\text{init}})\right) = \mathrm{Tr}\left(Q\left(\mathcal{E}_N + \sum_{n=2}^{\infty}\frac{\tau^n}{n!}\mathcal{L}^{(n)}\right)^N(\rho_{\text{init}})\right) \\ &= \mathrm{Tr}\left(Q\left(\mathcal{E}_N + \sum_{n=2}^{\infty}\frac{\tau^n}{n!}\sum_{\vec{\ell}}\sum_{s=0}^{1}(-1)^s P^{(n)}_s(\vec{\ell})\mathcal{L}_n(\vec{\ell})\right)^N(\rho_{\text{init}})\right) \\ &= \mathrm{Tr}\left(\tilde{Q}\left(\tilde{\mathcal{E}}_N + \sum_{n=2}^{\infty}\frac{\tau^n}{n!}\sum_{\vec{\ell}}\sum_{s=0}^{1}(-1)^s P^{(n)}_s(\vec{\ell})\tilde{\mathcal{L}}_n(\vec{\ell})\right)^N(\tilde{\rho}_{\text{init}})\right) = \mathrm{Tr}\left(\tilde{Q}\tilde{\mathcal{U}}^N_N(\tilde{\rho}_{\text{init}})\right), \end{aligned} \tag{C1}$$

where we use Eq. (28) in the second equality and we use Eqs. (74) and (77) in the third equality. We can readily show the fourth equality by using Eq. (89). In the last equality, we define

$$\tilde{\mathcal{U}}_N := \tilde{\mathcal{E}}_N + \sum_{n=2}^{\infty} \frac{\tau^n}{n!} \sum_{s=0}^{1} \sum_{\vec{b}\in\{0,1\}^{\otimes n}} \sum_{\vec{\ell}} (-1)^s P_s^{(n)}(\vec{\ell}) \tilde{S}_n^{(b)}(\vec{\ell}). \tag{C2}$$

We can rewrite $\tilde{\mathcal{U}}_N$ by using the physical channel as follows:

$$\begin{aligned}\tilde{\mathcal{U}}_N &= \tilde{\mathcal{E}}_N + \sum_{n=2}^{\infty} \frac{2^{n+1}\tau^n}{n!} \frac{1}{2} \sum_{s=0}^{1} \frac{1}{2^n} \sum_{\vec{b}\in\{0,1\}^{\otimes n}} \\ &\quad \times \sum_{\vec{\ell}} (-1)^s P_s^{(n)}(\vec{\ell}) \tilde{S}_n^{(b)}(\vec{\ell}) \\ &= \tilde{\mathcal{E}}_N + \sum_{n=2}^{\infty} \beta(n) \tilde{\mathcal{W}}_n \\ &= B\mathcal{E}_N^{(\mathrm{all})}.\end{aligned} \tag{C3}$$

In the second equality of Eq. (C3), we define

$$\beta(n) := \frac{2^{n+1}\tau^n}{n!}, \quad \tilde{\mathcal{W}}_n := \frac{1}{2} \sum_{s=0}^{1} \frac{1}{2^n} \sum_{\vec{b}\in\{0,1\}^{\otimes n}} \sum_{\vec{\ell}} (-1)^s P_s^{(n)}(\vec{\ell}) \tilde{S}_n^{(b)}(\vec{\ell}), \tag{C4}$$

where $\tilde{\mathcal{W}}_n (n \geq 2)$ is the physical channel; we can implement the process by sampling s and $\vec{b}$ uniformly, sampling $\vec{\ell}$ according to $P_s^{(n)}(\vec{\ell})$, and applying $(-1)^s \tilde{S}_n^{(b)}(\vec{\ell})$. In the last equality, we define

$$B := 1 + \sum_{n=2}^{\infty} \beta(n) = e^{(2\ln 2)\tau} - 4\tau - 1, \tag{C5}$$

and the channel as $\mathcal{E}_N^{(\mathrm{all})}$, the physical channel implemented by applying $\mathcal{E}_{\mathcal{N}}$ with the probability $1/B$ and $\mathcal{W}_n (n \geq 2)$ with the probability $\beta(n)/B$. Consequently, we obtain

$$\mathrm{Tr}\left(Q\mathcal{U}(\rho_{\mathrm{init}})\right) = B^N \mathrm{Tr}\left(\tilde{Q}\left(\mathcal{E}_N^{(\mathrm{all})}\right)^N (\tilde{\rho}_{\mathrm{init}})\right). \tag{C6}$$

We see that there is no systematic error on the right-hand side. For a given number of samples, denoted as N_{sample}, the statistical error ϵ_{st} scales as

$$\epsilon_{\mathrm{st}} \in \mathcal{O}\left(\frac{B^N}{\sqrt{N_{\mathrm{sample}}}}\right). \tag{C7}$$

Since $B^N < e^{(2\ln 2)(\lambda t)^2/N}$, we obtain $N_{\mathrm{sample}} \in \mathcal{O}(1/\epsilon_{\mathrm{st}}^2)$ by setting $N \in \mathcal{O}\left((\lambda t)^2\right)$.

We note that even though N is upper bounded, there is no theoretical upper bound on the number of gates, as the count of swift operators in $\tilde{\mathcal{W}}_n$ is determined by n. Here, n is not bounded and can take on a large value with probability $\beta(n)/B$. Conversely, the count of swift operators in the second- or higher-order qSWIFT is upper bounded by the order parameter and, therefore, the number of gates is also upper bounded. Thus, in practical situations where the number of operational gates is limited, the second- or higher-order qSWIFT may be more advantageous than the previously introduced all-order qSWIFT.

APPENDIX D: NOTE ON THE LCU-BASED RANDOMIZED APPROACH

The literature [19] provides a higher-order randomized method for phase estimation. The calculation includes the evaluation of $\mathrm{Tr}\left[\rho e^{iHt}\right]$ with $H = \sum_{\ell=1}^{L} h_\ell H_\ell$, where again we define $\{H_\ell\}_{\ell=1}^{L}$ so that $h_\ell > 0$ and we define $\lambda := \sum_\ell h_\ell$. For that, the authors propose an all-order randomized method for estimating $\mathrm{Tr}\left[\rho e^{iHt}\right]$ based on the linear combination of the unitary (LCU) approach. In this appendix, we show that by extending the randomized method, we can construct another all-order randomized method for estimating $\mathrm{Tr}(Qe^{\mathrm{i}Ht}\rho e^{-\mathrm{i}Ht})$, which is the target of our qSWIFT algorithm.

We begin by reviewing the method given in Ref. [19] and discuss how to extend this method to estimate the expectation value of an observable. Then, we discuss the difference between the LCU-based approach and the all-order qSWIFT introduced in Appendix C.

1. The LCU-based randomized approach

To estimate $\mathrm{Tr}[\rho e^{iHt}]$, the authors of Ref. [19] utilize the expansion

$$e^{iHt/N} = \sum_m c_m W_m, \tag{D1}$$

where

$$\begin{aligned} W_m &\to (\mathrm{i}\,\mathrm{sign}(t))^n H_{\ell_1} \ldots H_{\ell_n} V_{\ell'}^{(n)}, \\ c_m &\to \frac{1}{n!}\left(\frac{\lambda|t|}{N}\right)^n \sqrt{1 + \left(\frac{\lambda t}{N(n+1)}\right)^2}\, p_{\ell_1} \ldots p_{\ell_n} p_{\ell'} > 0, \end{aligned} \tag{D2}$$

with

$$V_{\ell'}^{(n)} = \exp(\mathrm{i}\theta_n H_{\ell'}), \quad \text{with } \theta_n := \arccos \times \left(\left[1 + \left(\frac{\lambda t}{N(n+1)}\right)^2\right]^{-1/2}\right). \tag{D3}$$

The multi-index m denotes the indices $(n, \vec{\ell}, \ell')$. Then,

$$e^{iHt} = \sum_{m_1,\ldots,m_N} c_{m_1} \ldots c_{m_N} W_{m_1} \ldots W_{m_N} = C^N \sum_{m_1,\ldots,m_N} q_{m_1} \cdots q_{m_N} W_{m_1} \ldots W_{m_N}, \tag{D4}$$

where $q_m = c_m/C$ with $C = \sum_m c_m$. Since $q_m > 0$ and $\sum_m q_m = 1$, $\{q_m\}$ can be interpreted as a probability distribution.

By using the expansion, we obtain

$$\mathrm{Tr}\left[\rho e^{iHt}\right] = C^N \sum_{m_1,\ldots,m_N} q_{m_1} \cdots q_{m_N} \times \left(\mathrm{Re}\left[\mathrm{Tr}\left(\rho W_{m_1} \cdots W_{m_N}\right)\right] + \mathrm{i}\,\mathrm{Im}\left[\mathrm{Tr}\left(\rho W_{m_1} \cdots W_{m_N}\right)\right]\right). \tag{D5}$$

We can estimate the right-hand side, by sampling $(m_1 \cdots m_N)$ according to $q_{m_1} \cdots q_{m_N}$, and evaluate the real part and the imaginary part of $\mathrm{Tr}\left(\rho W_{m_1} \cdots W_{m_N}\right)$ by the Hadamard test repeatedly. There is no systematic error. The statistical error ϵ_{st} scales as

$$\epsilon_{\mathrm{st}} \in \mathcal{O}\left(\frac{C^N}{\sqrt{N_{\mathrm{sample}}}}\right), \tag{D6}$$

where N_{sample} is the number of sampling the set of indices $(m_1, \ldots, m_N)$. In Ref. [19], $C \in \mathcal{O}\left(\exp\left((\lambda t)^2/N^2\right)\right)$, i.e., $C^N = \mathcal{O}\left(\exp\left((\lambda t)^2/N\right)\right)$ in our notation. Thus, we need $N = O\left((\lambda t)^2\right)$ so that $N_{\mathrm{sample}} \in \mathcal{O}\left(1/\epsilon_{\mathrm{st}}^2\right)$.

2. Application to the Hamiltonian-simulation problem

We can utilize the expansion in Eq. (D4) to calculate the expectation value after applying the time evolution:

$$\mathrm{Tr}\left(Qe^{iHt}\rho e^{-iHt}\right) = C^{2N} \sum_{m'_1 \ldots m'_N} \sum_{m_1 \ldots m_N} q_{m'_1} \ldots q_{m'_N} q_{m_1} \ldots q_{m_N} \mathrm{Tr}\left(QW_{m'_1} \ldots W_{m'_N} \rho W_{m_1}^\dagger \ldots W_{m_N}^\dagger\right) \tag{D7}$$

$$= C^{2N} \sum_{m'_1 \ldots m'_N} \sum_{m_1 \ldots m_N} q_{m'_1} \ldots q_{m'_N} q_{m_1} \ldots q_{m_N} \mathrm{Re}\Big[\mathrm{Tr}\left(QW_{m'_1} \ldots W_{m'_N} \rho W_{m_1}^\dagger \ldots W_{m_N}^\dagger\right)\Big], \tag{D8}$$

where we use that the left-hand side is the real value in the second equality. As in the case of Eq. (D5), we can estimate the value of Eq. (D8), by sampling $(m_1, \ldots, m_N)$ and $(m'_1, \ldots, m'_N)$ and estimating $\mathrm{Re}\Big[\mathrm{Tr}\left(QW_{m'_1} \cdots W_{m'_N} \rho W_{m_1}^\dagger \cdots W_{m_N}^\dagger\right)\Big]$ by the Hadamard test. Again, there is no systematic error and by setting $N = O((\lambda t)^2)$, the number of samples N_{sample} scales as $N_{\mathrm{sample}} \in \mathcal{O}(1/\epsilon_{\mathrm{st}}^2)$. Therefore, the behavior of both the systematic error and the statistical error with respect to N is the same as that of the all-order qSWIFT introduced in Appendix C.

However, the LCU-based method and our qSWIFT approach have a key distinction: the former expands the unitary operation, while the latter expands the unitary channel. This difference in expansion methodology results in distinct quantum circuits. In the LCU-based approach, the Hadamard test is necessary for calculating the real part of the trace function, requiring the implementation of controlled-W_m operations. For instance, the quantum circuit for evaluating Eq. (D8) is depicted in Fig. 6 (we note that the circuit is also used in another higher-order randomization protocol [24]). Given that each W_m operation involves a rotational operator of the form $e^{\mathrm{i}H_\ell t'}$ (with t' as a real value), a total of $2N$ controlled-$e^{\mathrm{i}H_\ell t'}$ operations are needed for the circuit. Conversely, in the qSWIFT approach, as illustrated in Fig. 2, no controlled-$e^{\mathrm{i}H_\ell t'}$ operations are necessary. The only interactions with the ancilla qubit involve swift operators, as all exponential time-evolution operators are encapsulated within the qDRIFT channel, which can be implemented without the need for an ancilla qubit.

The implementation of the controlled-$e^{\mathrm{i}H_\ell t'}$ operation requires additional controlled gates in the LCU-based

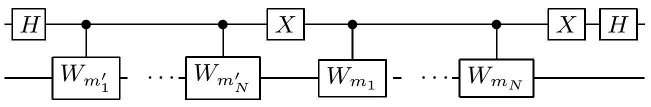

FIG. 6. The quantum circuits for evaluating Eq. (D8) by the LCU-based approach.

methods compared to the implementation of $e^{\mathrm{i}H_\ell t'}$ operation. In particular, in the simulation where the one-body term is dominant, the LCU-based method needs many more controlled gates than in qSWIFT. To demonstrate this, let us consider the following Hamiltonian, written as the summation of the multibody terms (the first summation) and the one-body terms (the second summation):

$$H = \frac{J}{h'_{\mathrm{sum}} n} \sum_{\ell=1}^{L'} h'_\ell P_\ell + \frac{h}{c_{\mathrm{sum}}} \sum_{j=1}^{n} \sum_{k=x,y,z} c_j^k \sigma_j^k. \tag{D9}$$

The first summation corresponds to L' multibody terms, where P_ℓ is the tensor product of the Pauli operators that act nontrivially on multiple qubits, with h'_ℓ as a real coefficient, $h'_{\mathrm{sum}} := \sum_\ell |h'_\ell|$, and J is a positive value. The second summation corresponds to $3n$ one-body terms, with n as the number of qubits, σ_j^k as one of the Pauli operators acting on the jth qubit, c_j^k and h'_ℓ as real coefficients, $c_{\mathrm{sum}} = \sum_{j=1}^{b} \sum_{k=x,y,z} |c_j^k|$, and h as a positive value. We assume the case in which the one-body term is dominant in the sense that $J \ll h$. The sum of all absolute values of the coefficients is given as $\lambda = J/n + h$.

Our objective of the simulation is approximately to compute $\mathrm{Tr}\left(Q e^{\mathrm{i}Ht} \rho e^{-\mathrm{i}Ht}\right)$ using directly implementable time evolutions: $e^{iP_\ell t'}$ and $e^{i\sigma_j^k t'}$. Note that the collective neutrino oscillation problem is an example of Hamiltonian simulation where the one-body term is dominant and has been already discussed in Ref. [25]. On the one hand, in the LCU-based approach, the required number of controlled time-evolution gates is $O((\lambda t)^2)$ and since each controlled time evolution requires at least one controlled gate, the total number of controlled gates is also $O((\lambda t)^2)$. On the other hand, the number of time-evolution gates is also $O((\lambda t)^2)$ in the all-order qSWIFT. However, most of the time-evolution gates sampled are $e^{i\sigma_j^k t'}$, which do not need the controlled gate; the number of $e^{iP_\ell t'}$ sampled is $O(J/(\lambda n))$ on average. Since each $e^{iP_\ell t'}$ requires at most $O(n)$ controlled gates, the total number of controlled gates in qSWIFT is at most

$$\mathcal{O}\left((\lambda t)^2 \times \frac{J}{\lambda n} \times n\right) \sim \mathcal{O}\left((\lambda t)^2 \times \frac{J}{h}\right), \tag{D10}$$

where to show the expression on the right-hand side, we use $\lambda \sim h$ since $J \ll h$. Therefore, for the simulation with the Hamiltonian in Eq. (D9), qSWIFT achieves a significant reduction of the controlled-NOT (CNOT) gates by the factor J/h.

[1] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Phys. Lett. A **146**, 319 (1990).

[2] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Commun. Math. Phys. **270**, 359 (2007).

[3] 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).

[4] A. M. Childs, A. Ostrander, and Y. Su, Faster quantum simulation by randomization, Quantum **3**, 182 (2019).

[5] E. Campbell, Random compiler for fast Hamiltonian simulation, Phys. Rev. Lett. **123**, 070503 (2019).

[6] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, Quantum Inf. Comput. **12**, 29 (2012).

[7] 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).

[8] D. W. Berry, A. M. Childs, and R. Kothari, in *56th Annual IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society, Berkeley, California, USA, 2015), p. 792.

[9] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, Phys. Rev. Lett. **118**, 010501 (2017).

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

[11] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, USA, 2019), p. 193.

[12] 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).

[13] K. R. Brown, R. J. Clark, and I. L. Chuang, Limitations of quantum simulation examined by simulating a pairing Hamiltonian using nuclear magnetic resonance, Phys. Rev. Lett. **97**, 050504 (2006).

[14] B. P. Lanyon, C. Hempel, D. Nigg, M. Müller, R. Gerritsma, F. Zähringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, M. Hennrich, P. Zoller, R. Blatt, and C. F. Roos, Universal digital quantum simulation with trapped ions, Science **334**, 57 (2011).

[15] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, and J. Y. Mutus, *et al.*, Digital quantum simulation of fermionic models with a superconducting circuit, Nat. Commun. **6**, 1 (2015).

[16] The code for the qSWIFT algorithm is available at https://github.com/konakaji/qswift.

[17] A. M. Childs, A. Ostrander, and Y. Su, Faster quantum simulation by randomization, Quantum **3**, 182 (2019).

[18] 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).

[19] K. Wan, M. Berta, and E. T. Campbell, Randomized quantum algorithm for statistical phase estimation, Phys. Rev. Lett. **129**, 030503 (2022).

[20] J. R. McClean, N. C. Rubin, K. J. Sung, I. D. Kivlichan, X. Bonet-Monroig, Y. Cao, C. Dai, E. S. Fried, C. Gidney,

and B. Gimby, *et al.*, Openfermion: The electronic structure package for quantum computers, Quantum Sci. Technol. **5,** 034014 (2020).

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

[22] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, and S. Tamiya, *et al.*, QULACS: A fast and versatile quantum circuit simulator for research purpose, Quantum **5**, 559 (2021).

[23] 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).

[24] P. K. Faehrmann, M. Steudtner, R. Kueng, M. Kieferova, and J. Eisert, Randomizing multi-product formulas for Hamiltonian simulation, Quantum **6**, 806 (2022).

[25] A. Rajput, A. Roggero, and N. Wiebe, Hybridized methods for quantum simulation in the interaction picture, Quantum **6**, 780 (2022).