Faster quantum simulation by randomization

Andrew M. Childs[1,2,3], Aaron Ostrander[2,3,4], and Yuan Su[1,2,3]

[1]Department of Computer Science, University of Maryland

[2]Institute for Advanced Computer Studies, University of Maryland

[3]Joint Center for Quantum Information and Computer Science, University of Maryland

[4]Department of Physics, University of Maryland

Product formulas can be used to simulate Hamiltonian dynamics on a quantum computer by approximating the exponential of a sum of operators by a product of exponentials of the individual summands. This approach is both straightforward and surprisingly efficient. We show that by simply randomizing how the summands are ordered, one can prove stronger bounds on the quality of approximation for product formulas of any given order, and thereby give more efficient simulations. Indeed, we show that these bounds can be asymptotically better than previous bounds that exploit commutation between the summands, despite using much less information about the structure of the Hamiltonian. Numerical evidence suggests that the randomized approach has better empirical performance as well.

1 Introduction

Simulating quantum dynamics is one of the major potential applications of quantum computers. The apparent intractability of simulating quantum dynamics with a classical computer led Feynman [19] and others to propose the idea of quantum computation. Lloyd gave the first explicit quantum algorithm for simulating the dynamics of local Hamiltonians [25], and later work showed that the more general class of sparse Hamiltonians can also be simulated efficiently [1]. Quantum simulation can be applied to understand the behavior of various physical systems—including many-body physics [31], quantum chemistry [2, 30, 36], and quantum field theory [23]—and designing new quantum algorithms [9, 11, 16, 18, 21].

The main ingredient in Lloyd's algorithm is the Lie product formula, which provides a first-order approximation to the exponential of a sum as a product of exponentials of the summands. Given Hermitian operators $H_1, \ldots, H_L$ (which we refer to as the summands of the Hamiltonian $H = \sum_{j=1}^{L} H_j$) and a complex number λ, the Lie product formula

$$S_1(\lambda) := \prod_{j=1}^{L} \exp(\lambda H_j), \tag{1}$$

approximates the exponentiation

$$V(\lambda) := \exp\left(\lambda \sum_{j=1}^{L} H_j\right) \tag{2}$$

in the sense that $V(\lambda) \approx S_1(\lambda/r)^r$ for large r. Suzuki systematically extended this formula to give a $(2k)$th-order approximation S_{2k}, defined recursively by

$$S_2(\lambda) := \prod_{j=1}^{L} \exp\left(\frac{\lambda}{2} H_j\right) \prod_{j=L}^{1} \exp\left(\frac{\lambda}{2} H_j\right)$$

$$S_{2k}(\lambda) := S_{2k-2}(p_k\lambda)^2 \, S_{2k-2}((1 - 4p_k)\lambda) \, S_{2k-2}(p_k\lambda)^2 \tag{3}$$

with $p_k := 1/(4 - 4^{1/(2k-1)})$ [33]. Again we have $V(\lambda) \approx S_{2k}(\lambda/r)^r$ for large r, and the approximation obtained with a given value of r improves as k increases (albeit with a prefactor that grows exponentially in k). We refer to all such formulas as *product formulas*. When they are used for quantum simulation, H is chosen to be the Hamiltonian and $\lambda = -it$, where t is the evolution time. Although other approaches to quantum simulation have better proven asymptotic performance as a function of various parameters [4, 6–8, 20, 26, 27], product formulas perform well in practice [17] and are widely used in experimental implementations [3, 12, 24] due to their simplicity and the fact that they do not require any ancilla qubits.

The main challenge in applying product formulas to quantum simulation is to choose the number of segments r to ensure the simulation error is at most some allowed threshold ϵ. To simulate $H = \sum_{j=1}^{L} H_j$ for time t, rigorous error analysis shows that

$$r_{1,\text{det}} = O\left(\frac{(t\Lambda L)^2}{\epsilon}\right) \tag{4}$$

suffices to ensure error at most ϵ for the first-order formula and

$$r_{2k,\text{det}} = O\left(\frac{(t\Lambda L)^{1+\frac{1}{2k}}}{\epsilon^{\frac{1}{2k}}}\right) \tag{5}$$

suffices for $(2k)$th order [5], where $\Lambda := \max_j \|H_j\|$ is a spectral-norm upper bound on the summands and det indicates that these formulas are constructed deterministically. However, numerical simulations suggest that the product formula algorithm can perform significantly better in practice than the best proven error bounds demonstrate [2, 17, 31, 32]. Indeed, recent work suggests that it can even asymptotically outperform more sophisticated simulation algorithms with better proven running times [17]. This dramatic gap between the provable and the actual behavior of product formula simulation suggests that it may be possible to significantly improve their analysis, and thereby give more efficient algorithms for quantum simulation.

It is sometimes possible to improve the analysis of product formulas using further information about the form of the Hamiltonian. In particular, the cost of simulation can be reduced when many pairs of summands commute [2, 17, 25]. However, this approach can only be applied for structured Hamiltonians that contain many commuting summands. Furthermore, the best known bounds of this type give only modest improvement, remaining orders of magnitude away from the empirical performance even in cases where many summands commute [17].

Randomization can be a powerful tool for improving the performance of quantum simulation algorithms. For example, Poulin et al. gave improved simulations of time-dependent Hamiltonians by sampling the Hamiltonian at random times [29]. Closer in spirit to the present paper, Zhang studied the effect of randomizing the ordering and/or duration of evolutions in a product formula, showing in particular that randomly ordering the summands in the first-order formula in either forward or reverse order can give an improved algorithm [37].

In this paper, we explore a closely related approach for higher-order product formulas, which can achieve significantly better asymptotic performance. Specifically, we analyze the effect of randomly permuting the summands. The resulting algorithm is not much more complicated than a deterministic product formula, but the savings in the simulation cost are substantial. For any permutation $\sigma \in \text{Sym}(L)$ of the L summands, let

$$S_2^\sigma(\lambda) := \prod_{j=1}^{L} \exp\left(\frac{\lambda}{2} H_{\sigma(j)}\right) \prod_{j=L}^{1} \exp\left(\frac{\lambda}{2} H_{\sigma(j)}\right)$$

$$S_{2k}^\sigma(\lambda) := [S_{2k-2}^\sigma(p_k\lambda)]^2 S_{2k-2}^\sigma((1-4p_k)\lambda)[S_{2k-2}^\sigma(p_k\lambda)]^2. \tag{6}$$

We show that the $(2k)$th-order randomized simulation has error

$$\left\|\mathcal{V}(-it) - \left(\frac{1}{L!} \sum_{\sigma \in \text{Sym}(L)} \mathcal{S}_{2k}^\sigma(-it/r)\right)^r\right\|_\diamond = O\left(\frac{(\Lambda tL)^{4k+2}}{r^{4k+1}} + \frac{(\Lambda t)^{2k+1}L^{2k}}{r^{2k}}\right). \tag{7}$$

where $\mathcal{V}(-it)$ and $\mathcal{S}_{2k}^\sigma(-it/r)$ are quantum channels describing the unitary transformation $V(-it)$ and the random unitary $S_{2k}^\sigma(-it/r)$, respectively, and $\|\cdot\|_\diamond$ is the diamond norm (defined in Section 2).

Our analysis uses a mixing lemma of Campbell and Hastings [13, 22] to bound the diamond norm distance from the ideal evolution. (Even for the first-order case, this improves over the analysis of Zhang, which uses similar methods but only bounds the trace distance from the ideal final state [37], a metric that does not account for entanglement with a reference system.) Informally, the lemma of [13, 22] states that if we can approximate a desired operation as the average over some set of operations, then the overall error depends linearly on the error in the average operation but only quadratically on the error in any individual operation. Standard error bounds for product formulas do not depend on how the summands are ordered, but we show that randomizing the ordering gives a more accurate average evolution. We motivate this approach in Section 2, where we consider the effect of randomizing how the summands are ordered in the simple case of the first-order formula. Assuming $\Lambda := \max_j \|H_j\|$ is constant, the randomized first-order algorithm has gate complexity $g_1^{\text{rand}} = O(t^{1.5}L^{2.5}/\epsilon^{0.5})$, improving over $g_1^{\text{det}} = O(t^2L^3/\epsilon)$ in the deterministic case.

Analyzing the effect of randomization on higher-order formulas is more challenging. For terms of order at most L in the Taylor expansion of a product formula, the majority of the error comes from terms in which no summands are repeated. We call such contributions *nondegenerate terms*. In Section 3, we give a combinatorial argument to compute nondegenerate terms of the average evolution $\frac{1}{L!} \sum_{\sigma \in \text{Sym}(L)} S_{2k}^\sigma(\lambda)$ in closed form. (In fact, we prove a more general result that applies to the average evolution as a special case.) As a corollary, we show that the nondegenerate terms completely cancel in the randomized product formula.

Section 4 presents our main technical result, an upper bound on the error in a randomized higher-order product formula simulation. This bound follows by using the mixing lemma to combine an error bound for the average evolution operator with standard product formula error bounds for the error of the individual terms. Section 5 discusses the overall performance of the resulting algorithm and compares it with deterministic approaches. For the $(2k)$th-order product formula, assuming $\Lambda := \max_j \|H_j\|$ is constant, our randomized Hamiltonian simulation algorithm has complexity

$$g_{2k}^{\text{rand}} = \max\left\{O\left(tL^2\left(\frac{tL}{\epsilon}\right)^{\frac{1}{4k+1}}\right), O\left(tL^2\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right)\right\}, \tag{8}$$

compared to $g_{2k}^{\text{det}} = O\big(tL^2(tL/\epsilon)^{\frac{1}{2k}}\big)$ in the deterministic case. Thus our algorithm always improves the dependence on L and sometimes achieves better dependence on t and ϵ as well.

We also show in Section 5 that our bound can outperform a previous bound that takes advantage of the structure of the Hamiltonian. Specifically, we compare our randomized product formula algorithm with the deterministic algorithm using the commutator bound of [17] for a one-dimensional Heisenberg model in a random magnetic field. We find that over a significant range of parameters, the randomized algorithm has better proven performance, despite using less information about the form of the Hamiltonian.

In light of the large gap between proven and empirical performance of product formulas, it is natural to ask whether randomized product formulas still offer an improvement under the best possible error bounds. To address this question, we present numerical comparisons of the deterministic and randomized product formulas in Section 6. In particular, we show that the randomized approach can sometimes outperform the deterministic approach even with respect to their empirical performance.

Finally, we conclude in Section 7 with a brief discussion of the results and some open questions.

2   The power of randomization

To see how randomness can improve a product formula simulation, consider a simple Hamiltonian expressed as a sum of two operators, $H = H_1 + H_2$. The Taylor expansion of the first-order formula as a function of $\lambda \in \mathbb{C}$ is

$$S_1(\lambda) = \exp(\lambda H_1)\exp(\lambda H_2) = I + \lambda(H_1 + H_2) + \frac{\lambda^2}{2}(H_1^2 + 2H_1 H_2 + H_2^2) + O(\lambda^3), \quad (9)$$

whereas the Taylor series of the ideal evolution is

$$V(\lambda) = \exp((H_1 + H_2)\lambda) = I + \lambda(H_1 + H_2) + \frac{\lambda^2}{2}(H_1^2 + H_1 H_2 + H_2 H_1 + H_2^2) + O(\lambda^3). \quad (10)$$

Using the triangle inequality, we can bound the spectral-norm error as

$$\|V(\lambda) - S_1(\lambda)\| \leq \|[H_1, H_2]\| \frac{|\lambda|^2}{2} + O((\Lambda|\lambda|)^3), \quad (11)$$

where $\Lambda := \max\{\|H_1\|, \|H_2\|\}$. Since H_1 and H_2 need not commute, $S_1(\lambda)$ approximates $V(\lambda)$ to first order in λ, as expected.

It is clearly impossible to approximate $V(\lambda)$ to second order using a product of only two exponentials of H_1 and H_2: any such product can have only one of the products $H_1 H_2$ and $H_2 H_1$ in its Taylor expansion, whereas $V(\lambda)$ contains both of these products in its second-order term. However, we can obtain both products by taking a uniform mixture of $S_1(\lambda)$ and

$$S_1^{\text{rev}}(\lambda) := \exp(\lambda H_2)\exp(\lambda H_1). \quad (12)$$

Indeed, a simple calculation shows that

$$\left\|V(\lambda) - \frac{1}{2}(S_1(\lambda) + S_1^{\text{rev}}(\lambda))\right\| = O((\Lambda|\lambda|)^3). \quad (13)$$

However, $\big(S_1(-it) + S_1^{\text{rev}}(-it)\big)/2$ is not a unitary operation in general. We could in principle implement a linear combination of unitaries using the techniques of [6], but such

an approach would use ancillas and could have high cost, especially when the Hamiltonian contains many summands. A simpler approach is to apply one of the two operations $S_1(-it)$ and $S_1^{\text{rev}}(-it)$ chosen uniformly at random (as in Algorithm 2 of [37]), thereby implementing a quantum channel that gives a good approximation to the desired evolution.

We now introduce some notation that is useful to analyze the performance of randomized product formulas. Let X be a matrix acting on a finite-dimensional Hilbert space $\mathcal{H}$. We write $\|X\|$ for its spectral norm (the largest singular value) and $\|X\|_1$ for its trace norm (the sum of its singular values, i.e., its Schatten 1-norm). Let $\mathcal{E}\colon X \mapsto \mathcal{E}(X)$ be a linear map on the space of matrices on $\mathcal{H}$. The diamond norm of $\mathcal{E}$ is

$$\|\mathcal{E}\|_\diamond := \max\{\|(\mathcal{E} \otimes \mathbb{1}_\mathcal{H})(Y)\|_1 : \|Y\|_1 \leq 1\}, \tag{14}$$

where the maximization is taken over all matrices Y on $\mathcal{H} \otimes \mathcal{H}$ satisfying $\|Y\|_1 \leq 1$.

The following mixing lemma bounds how well we can approximate a unitary operation using a random unitary channel. Specifically, the error is linear in the distance between the target unitary and the average of the random unitaries, and only quadratic in the distance between the target unitary and each individual random unitary.

Lemma 1 (Mixing lemma [13, 22]). *Let V and $\{U_j\}$ be unitary matrices, with associated quantum channels $\mathcal{V}\colon \rho \mapsto V\rho V^\dagger$ and $\mathcal{U}_j\colon \rho \mapsto U_j\rho U_j^\dagger$, and let $\{p_j\}$ be a collection of positive numbers satisfying $\sum_j p_j = 1$. Suppose that*

(i) $\quad \|U_j - V\| \leq a$ *for all j and*

(ii) $\quad \|(\sum_j p_j U_j) - V\| \leq b$.

Then the average evolution $\mathcal{E} := \sum_j p_j \mathcal{U}_j$ satisfies $\|\mathcal{E} - \mathcal{V}\|_\diamond \leq a^2 + 2b$.

To simulate the Hamiltonian $H = H_1 + H_2$ for time t, we divide the evolution into r segments of duration t/r and implement each segment via the random unitary operation

$$\frac{1}{2}\big(\mathcal{S}_1(-it/r) + \mathcal{S}_1^{\text{rev}}(-it/r)\big) \tag{15}$$

using one bit of randomness per segment, where $\mathcal{S}_1$ and $\mathcal{S}_1^{\text{rev}}$ are the quantum channels associated with S_1 and S_1^{rev}. Invoking the mixing lemma with $a = O((\Lambda t)^2/r^2)$ and $b = O((\Lambda t)^3/r^3)$, we find that

$$\left\|\mathcal{V}(-it/r) - \frac{1}{2}\big(\mathcal{S}_1(-it/r) + \mathcal{S}_1^{\text{rev}}(-it/r)\big)\right\|_\diamond = O\left(\frac{(\Lambda t)^3}{r^3}\right). \tag{16}$$

Since the diamond norm distance between quantum channels is subadditive under composition [35, p. 178], the error of the entire simulation is

$$\left\|\mathcal{V}(-it) - \frac{1}{2^r}\big(\mathcal{S}_1(-it/r) + \mathcal{S}_1^{\text{rev}}(-it/r)\big)^r\right\|_\diamond = O\left(\frac{(\Lambda t)^3}{r^2}\right). \tag{17}$$

Thus the randomized first-order formula is effectively a second-order formula.

This approach easily extends to a sum of L operators, again effectively making the first-order formula accurate to second order (cf. [37], which shows the same result with respect to trace distance of the output state). Keeping track of all the prefactors, we find the following error bound for the randomized first-order formula.

Theorem 1 (Randomized first-order error bound). *Let $\{H_j\}_{j=1}^L$ be Hermitian matrices. Let*

$$V(-it) := \exp\left(-it \sum_{j=1}^L H_j\right) \tag{18}$$

be the evolution induced by the Hamiltonian $H = \sum_{j=1}^{L} H_j$ for time $t \in \mathbb{R}$. Define

$$S_1(\lambda) := \prod_{j=1}^{L} \exp(\lambda H_j) \quad and \quad S_1^{\mathrm{rev}}(\lambda) := \prod_{j=L}^{1} \exp(\lambda H_j). \tag{19}$$

Let $r \in \mathbb{N}$ be a positive integer and $\Lambda := \max \|H_j\|$. Then

$$\left\| \mathcal{V}(-it) - \frac{1}{2^r} \left(\mathcal{S}_1(-it/r) + \mathcal{S}_1^{\mathrm{rev}}(-it/r) \right)^r \right\|_{\diamond} \leq \frac{(\Lambda|t|L)^4}{r^3} \exp\left(2\frac{\Lambda|t|L}{r} \right) + \frac{2(\Lambda|t|L)^3}{3r^2} \exp\left(\frac{\Lambda|t|L}{r} \right) \tag{20}$$

where, for $\lambda = -it$, we associate channels $\mathcal{V}(\lambda)$, $\mathcal{S}_1(\lambda)$, and $\mathcal{S}_1^{\mathrm{rev}}(\lambda)$ with the unitaries $V(\lambda)$, $S_1(\lambda)$, and $S_1^{\mathrm{rev}}(\lambda)$, respectively.

To guarantee that the simulation error is at most ϵ, we upper bound the right-hand side of (20) by ϵ and solve for r. Assuming $\Lambda := \max_j \|H_j\|$ is constant, we find that it suffices to choose $r_1^{\mathrm{rand}} = O((tL)^{1.5}/\epsilon^{0.5})$, giving a simulation algorithm with gate complexity $g_1^{\mathrm{rand}} = O(t^{1.5}L^{2.5}/\epsilon^{0.5})$. In comparison, the gate complexity in the deterministic case is $g_1^{\mathrm{det}} = O(t^2 L^3/\epsilon)$. Therefore, the randomized first-order product formula algorithm improves over the deterministic algorithm with respect to all parameters of interest.

It is natural to ask whether a similar randomization strategy can improve higher-order product formulas (as defined in (3)). While it turns out that randomization does not improve the order of the formula, it does result in a significant reduction of the error, and in particular, lowers the dependence on the number of summands in the Hamiltonian. The more complicated structure of higher-order formulas makes this analysis more involved than in the first-order case (in particular, we randomly permute the L summands instead of simply choosing whether or not to reverse them, so we use $\Theta(L \log L)$ bits of randomness per segment instead of only a single bit). As discussed at the end of Section 1, our proof is based on a randomization lemma (established in the next section) that evaluates the dominant contribution to the Taylor series of the randomized product formula in closed form.

3 Randomization lemma

In this section, we study the Taylor expansion of the average evolution operator obtained by randomizing how the summands of a Hamiltonian are ordered. We consider a formula of the form

$$
\begin{aligned}
&\exp(q_1 \lambda H_{\pi_1(1)}) \exp(q_1 \lambda H_{\pi_1(2)}) \cdots \exp(q_1 \lambda H_{\pi_1(L)}) \\
&\exp(q_2 \lambda H_{\pi_2(1)}) \exp(q_2 \lambda H_{\pi_2(2)}) \cdots \exp(q_2 \lambda H_{\pi_2(L)}) \\
&\cdots \\
&\exp(q_\kappa \lambda H_{\pi_\kappa(1)}) \exp(q_\kappa \lambda H_{\pi_\kappa(2)}) \cdots \exp(q_\kappa \lambda H_{\pi_\kappa(L)})
\end{aligned}
\tag{21}
$$

for real numbers $q_1, \ldots, q_\kappa \in \mathbb{R}$, a complex number $\lambda \in \mathbb{C}$, Hermitian matrices $H_1, \ldots, H_L$, and permutations $\pi_1, \ldots, \pi_\kappa \in \mathrm{Sym}(L)$. By choosing appropriate values of $q_1, \ldots, q_\kappa \in \mathbb{R}$ and ordering $H_1, \ldots, H_L$ in both forward and backward directions, we can write any product formula $S_{2k}(\lambda)$ in this form.

6

We now permute the summands to get the average evolution

$$\frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \exp(q_1 \lambda H_{\sigma(\pi_1(1))}) \exp(q_1 \lambda H_{\sigma(\pi_1(2))}) \cdots \exp(q_1 \lambda H_{\sigma(\pi_1(L))})$$

$$\exp(q_2 \lambda H_{\sigma(\pi_2(1))}) \exp(q_2 \lambda H_{\sigma(\pi_2(2))}) \cdots \exp(q_2 \lambda H_{\sigma(\pi_2(L))}) \qquad (22)$$

$$\cdots$$

$$\exp(q_\kappa \lambda H_{\sigma(\pi_\kappa(1))}) \exp(q_\kappa \lambda H_{\sigma(\pi_\kappa(2))}) \cdots \exp(q_\kappa \lambda H_{\sigma(\pi_\kappa(L))}).$$

In its Taylor expansion, we call the sum of the form

$$\sum_{\substack{m_1,\ldots,m_s \\ \text{pairwise different}}} \alpha_{m_1 \ldots m_s} \lambda^s H_{m_1} \cdots H_{m_s}, \qquad (23)$$

with coefficients $\alpha_{m_1 \ldots m_s} \in \mathbb{C}$, the sth-order *nondegenerate term*. This term contributes $\Theta(L^s)$ to the sth-order error, whereas the remaining (degenerate) terms only contribute $O(L^{s-1})$.

The following lemma shows how to compute the sth-order nondegenerate term for an arbitrary average evolution.

Lemma 2 (Randomization lemma). *Define an average evolution operator as in (22) and let $s \leq L$ be a positive integer. The sth-order nondegenerate term of this operator is*

$$\frac{[(q_1 + \cdots + q_\kappa)\lambda]^s}{s!} \sum_{\substack{m_1,\ldots,m_s \\ \text{pairwise different}}} H_{m_1} \cdots H_{m_s}. \qquad (24)$$

Proof. We take all possible products of s terms from the Taylor expansion of (22). Observe that the exponentials in (22) are organized in an array with κ rows and L columns. We use $\kappa_1,\ldots,\kappa_s$ and $l_1,\ldots,l_s$ to label the row and column indices, respectively, of the exponentials from which the terms are chosen. To avoid double counting, we take terms with smaller row indices first (i.e., $\kappa_1 \leq \cdots \leq \kappa_s$). Within each row, we take terms with smaller column indices first. To get the sth-order nondegenerate term, we require that $\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s)$ are pairwise different. The sth-order nondegenerate term of (22) can then be expressed as

$$\frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \sum_{\kappa_1 \leq \cdots \leq \kappa_s} \sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s) \\ \text{pairwise different}}} (q_{\kappa_1} \lambda H_{\sigma(\pi_{\kappa_1}(l_1))}) \cdots (q_{\kappa_s} \lambda H_{\sigma(\pi_{\kappa_s}(l_s))}). \qquad (25)$$

A direct calculation shows that

$$\frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \sum_{\kappa_1 \leq \cdots \leq \kappa_s} \sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s) \\ \text{pairwise different}}} (q_{\kappa_1} \lambda H_{\sigma(\pi_{\kappa_1}(l_1))}) \cdots (q_{\kappa_s} \lambda H_{\sigma(\pi_{\kappa_s}(l_s))})$$

$$= \frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \sum_{\kappa_1 \leq \cdots \leq \kappa_s} \sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s) \\ \text{pairwise different}}} \sum_{\substack{m_1 = \sigma(\pi_{\kappa_1}(l_1)),\ldots, \\ m_s = \sigma(\pi_{\kappa_s}(l_s))}} (q_{\kappa_1} \lambda H_{m_1}) \cdots (q_{\kappa_s} \lambda H_{m_s})$$

$$= \frac{1}{L!} \sum_{\substack{m_1,\ldots,m_s \\ \text{pairwise different}}} \sum_{\kappa_1 \leq \cdots \leq \kappa_s} \sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s) \\ \text{pairwise different}}} \sum_{\substack{\sigma \in \mathrm{Sym}(L): \\ \sigma(\pi_{\kappa_1}(l_1)) = m_1,\ldots, \\ \sigma(\pi_{\kappa_s}(l_s)) = m_s}} (q_{\kappa_1} \lambda H_{m_1}) \cdots (q_{\kappa_s} \lambda H_{m_s})$$

$$= \frac{(L-s)!}{L!} \sum_{\substack{m_1,\ldots,m_s \\ \text{pairwise different}}} \left[\sum_{\kappa_1 \leq \cdots \leq \kappa_s} \sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s) \\ \text{pairwise different}}} (q_{\kappa_1} \lambda) \cdots (q_{\kappa_s} \lambda) \right] H_{m_1} \cdots H_{m_s}.$$

$$(26)$$

Now observe that the summand $(q_{\kappa_1}\lambda)\cdots(q_{\kappa_s}\lambda)$ depends only on the row indices. Letting $r_1,\ldots,r_\kappa$ denote the number of terms picked from row $1,\ldots,\kappa$, respectively, we can re-express this summand as $(q_1\lambda)^{r_1}\cdots(q_\kappa\lambda)^{r_\kappa}$. We determine the coefficient of this term as follows. The number of ways of choosing $l_1,\ldots,l_s$ pairwise different is $L(L-1)\cdots(L-s+1)$. However, when we apply permutations $\pi_{\kappa_1},\ldots,\pi_{\kappa_s}$, we may double count some terms. In particular, if $\kappa_i=\kappa_{i+1}$, we are to pick terms from the same row κ_i and we must have $l_i < l_{i+1}$. This implies that the ordering of $\pi_{\kappa_i}(l_i)$ and $\pi_{\kappa_{i+1}}(l_{i+1})$ is uniquely determined. Altogether, we see that we have overcounted by a factor of $(r_1!)\cdots(r_\kappa!)$. Therefore, we have

$$
\sum_{\kappa_1\leq\cdots\leq\kappa_s}\sum_{\substack{\pi_{\kappa_1}(l_1),\ldots,\pi_{\kappa_s}(l_s)\\ \text{pairwise different}}}(q_{\kappa_1}\lambda)\cdots(q_{\kappa_s}\lambda)=\sum_{\substack{r_1,\ldots,r_\kappa:\\ r_1+\cdots+r_\kappa=s}}\frac{L(L-1)\cdots(L-s+1)}{(r_1!)\cdots(r_\kappa!)}(q_1\lambda)^{r_1}\cdots(q_\kappa\lambda)^{r_\kappa}
$$

$$
= L(L-1)\cdots(L-s+1)\frac{[(q_1+\cdots+q_\kappa)\lambda]^s}{s!},
\tag{27}
$$

where the last equality follows by the multinomial theorem.

Substituting (27) into (26) completes the proof. $\qquad\square$

As an immediate corollary, we compute the sth-order nondegenerate term of the average evolution operator $\frac{1}{L!}\sum_{\sigma\in\mathrm{Sym}(L)}S_{2k}^\sigma(\lambda)$.

Corollary 1. *Let $\{H_j\}_{j=1}^L$ be Hermitian operators; let $\lambda\in\mathbb{C}$, $k,s\in\mathbb{N}$, and $s\leq L$. Then the sth-order nondegenerate term of the average evolution $\frac{1}{L!}\sum_{\sigma\in\mathrm{Sym}(L)}S_{2k}^\sigma(\lambda)$, with $S_{2k}^\sigma(\lambda)$ defined in (6), is*

$$
\frac{\lambda^s}{s!}\sum_{\substack{m_1,\ldots,m_s\\ \text{pairwise different}}}H_{m_1}\cdots H_{m_s}.
\tag{28}
$$

Proof. The fact that $S_{2k}^\sigma(\lambda)$ is at least first-order accurate implies that $q_1+\cdots+q_\kappa=1$ in (24). $\qquad\square$

Observe that the sth-order nondegenerate term of $V(\lambda)=\exp(\lambda\sum_{j=1}^L H_j)$ is also given by (28). Therefore, the sth-order nondegenerate term completely cancels in

$$
V(\lambda)-\frac{1}{L!}\sum_{\sigma\in\mathrm{Sym}(L)}S_{2k}^\sigma(\lambda).
\tag{29}
$$

4 Error bounds

In this section we establish our main result, an upper bound on the error of a randomized product formula simulation. To apply the mixing lemma, we need to bound the error of the average evolution. We now present an error bound for an arbitrary fixed-order term in the Taylor expansion of the average evolution operator.

Lemma 3. *Let $\{H_j\}_{j=1}^L$ be Hermitian operators; let $\lambda\in\mathbb{C}$ and $k,s\in\mathbb{N}$. Define the target evolution $V(\lambda)$ as in (2), and define the permuted $(2k)$th-order formula $S_{2k}^\sigma(\lambda)$ as in (6). Then the sth-order error of the approximation*

$$
V(\lambda)-\frac{1}{L!}\sum_{\sigma\in\mathrm{Sym}(L)}S_{2k}^\sigma(\lambda)
\tag{30}
$$

is at most

$$\begin{cases} 0 & 0 \le s \le 2k, \\ \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{(s-2)!} L^{s-1} & s > 2k, \end{cases} \tag{31}$$

where $\Lambda := \max \|H_j\|$.

The proof of this error bound uses the following estimate of a fixed-order degenerate term in the average evolution operator.

Lemma 4. *Let* $\{H_j\}_{j=1}^L$ *be Hermitian operators with* $\Lambda := \max_j \|H_j\|$; *let* $q_1, \ldots, q_\kappa \in \mathbb{R}$ *with* $\max_k |q_k| \le 1$; *and let* $s \le L$ *be a positive integer. Then the norm of the sth-order degenerate term of the ideal evolution operator* $V(\lambda)$ *as in (2) is at most*

$$\frac{(\Lambda |\lambda|)^s}{s!} \big[L^s - L(L-1) \cdots (L-s+1) \big] \tag{32}$$

and the norm of the sth-order degenerate term of the average evolution operator as in (22) is at most

$$\frac{(\kappa \Lambda |\lambda|)^s}{s!} \big[L^s - L(L-1) \cdots (L-s+1) \big]. \tag{33}$$

Proof. The sth-order term of $V(\lambda)$ is

$$\frac{\left(\lambda \sum_{j=1}^L H_j \right)^s}{s!} = \frac{\lambda^s}{s!} \sum_{m_1, \ldots, m_s} H_{m_1} \cdots H_{m_s} \tag{34}$$

and its nondegenerate term is

$$\frac{\lambda^s}{s!} \sum_{\substack{m_1, \ldots, m_s \\ \text{pairwise different}}} H_{m_1} \cdots H_{m_s}. \tag{35}$$

We use the following strategy to bound the norms of these terms: (i) bound the norm of a sum of terms by summing the norms of each term; (ii) bound the norm of a product of terms by multiplying the norms of each term; (iii) bound the norm of each summand by Λ; and (iv) replace λ by $|\lambda|$. Applying this strategy, we find that the norm of the sth-order term is at most $(L\Lambda|\lambda|)^s/s!$, where the nondegenerate term contributes precisely $L(L-1) \cdots (L-s+1)(\Lambda|\lambda|)^s/s!$. Taking the difference gives the desired bound (32).

According to Lemma 2, the sth-order nondegenerate term of the average evolution is

$$\frac{[(q_1 + \cdots + q_\kappa)\lambda]^s}{s!} \sum_{\substack{m_1, \ldots, m_s \\ \text{pairwise different}}} H_{m_1} \cdots H_{m_s}. \tag{36}$$

Following the same strategy as for $V(\lambda)$ and also upper bounding the norm of each q_k by 1 as part of step (iv), we find that the norm of this term is at most

$$\frac{(\kappa \Lambda |\lambda|)^s}{s!} L(L-1) \cdots (L-s+1). \tag{37}$$

It remains to find an upper bound for the entire sth-order term of the average evolution. To this end, we start with the average evolution (22) and apply the following strategy: (i′) replace each summand of the Hamiltonian by Λ; (ii′) replace each q_k by 1 and each λ

by $|\lambda|$; and (iii') expand all exponentials into their Taylor series and extract the sth-order term. In other words, we extract the sth-order term of $\sum_{\sigma \in \mathrm{Sym}(L)} \exp(\kappa L \Lambda |\lambda|)/L!$ to get

$$\frac{(\kappa L \Lambda |\lambda|)^s}{s!}. \tag{38}$$

The equivalence of strategies (i)–(iv) and (i')–(iii') can be seen from [17, Eq. (57)]. Finally, taking the difference between (38) and (37) gives the desired bound (33). $\qquad\square$

Proof of Lemma 3. We first prove a stronger bound, namely that the sth-order error is at most

$$\begin{cases} 0 & 0 \leq s \leq 2k, \\ 2\frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!}[L^s - L(L-1) \cdots (L-s+1)] & 2k < s \leq L, \\ 2\frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!} L^s & s > L. \end{cases} \tag{39}$$

The first and third cases in this expression are straightforward. The formula S_{2k}^σ is exact for terms with order $0 \leq s \leq 2k$ (this is what it means for the formula to have order $2k$), so the error is zero in this case. When $s > L$, the randomization lemma is not applicable and the error can be bounded as in [17, Proof of Proposition F.3].

To handle the remaining case $2k < s \leq L$, we apply Lemma 4 with $\kappa = 2 \cdot 5^{k-1}$. This choice of κ follows from the definition of the $(2k)$th-order formula (3). The norm of the sth-order degenerate terms can be upper bounded by

$$\frac{(\Lambda |\lambda|)^s}{s!}\left[L^s - L(L-1) \cdots (L-s+1)\right] + \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!}\left[L^s - L(L-1) \cdots (L-s+1)\right]. \tag{40}$$

According to Corollary 1, the sth-order nondegenerate term of (30) cancels, which proves (39) for $2k < s \leq L$.

To finish the proof, we need a unified error expression for order $s > 2k$. When $2k < s \leq L$, we have

$$\begin{aligned} L^s &- L(L-1) \cdots (L-s+1) \\ &= \#\{(l_1, \ldots, l_s) \in [L]^s\} - \#\{(l_1, \ldots, l_s) \in [L]^s : \forall i,j,\ l_i \neq l_j\} \\ &= \#\{(l_1, \ldots, l_s) \in [L]^s\} - \#\bigcap_{i<j}\{(l_1, \ldots, l_s) \in [L]^s :\ l_i \neq l_j\} \\ &= \#\bigcup_{i<j}\{(l_1, \ldots, l_s) \in [L]^s :\ l_i = l_j\} \\ &\leq \binom{s}{2} L^{s-1}, \end{aligned} \tag{41}$$

with $\#\{\cdot\}$ denoting the size of a set and $[L] := \{1, \ldots, L\}$, where the inequality follows from the union bound. Therefore, we have

$$\begin{aligned} 2\frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!}[L^s - L(L-1) \cdots (L-s+1)] &\leq \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!} s(s-1) L^{s-1} \\ &= \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{(s-2)!} L^{s-1}. \end{aligned} \tag{42}$$

If $s > L \in \mathbb{N}$, we have $s(s-1) \geq (L+1)L \geq 2L$ and

$$2\frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!} L^s \leq \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{(s-2)!} L^{s-1}. \tag{43}$$

This completes the proof. $\qquad\square$

We also use the following standard tail bound on the exponential function [17, Lemma F.2].

Lemma 5. *For any $x \in \mathbb{C}$ and $\kappa \in \mathbb{N}$, we have*

$$\left| \sum_{s=\kappa}^{\infty} \frac{x^s}{s!} \right| \leq \frac{|x|^{\kappa}}{\kappa!} \exp(|x|). \tag{44}$$

We now establish the main theorem, which upper bounds the error of a higher-order randomized product formula.

Theorem 2 (Randomized higher-order error bound). *Let $\{H_j\}_{j=1}^{L}$ be Hermitian matrices. Let*

$$V(-it) := \exp\left(-it \sum_{j=1}^{L} H_j\right) \tag{45}$$

be the evolution induced by the Hamiltonian $H = \sum_{j=1}^{L} H_j$ for time t. For any permutation $\sigma \in \mathrm{Sym}(L)$, define the permuted $(2k)$th-order formula recursively by

$$S_2^{\sigma}(\lambda) := \prod_{j=1}^{L} \exp\left(\frac{\lambda}{2} H_{\sigma(j)}\right) \prod_{j=L}^{1} \exp\left(\frac{\lambda}{2} H_{\sigma(j)}\right)$$
$$S_{2k}^{\sigma}(\lambda) := [S_{2k-2}^{\sigma}(p_k\lambda)]^2 S_{2k-2}^{\sigma}((1-4p_k)\lambda)[S_{2k-2}^{\sigma}(p_k\lambda)]^2, \tag{46}$$

with $p_k := 1/(4 - 4^{1/(2k-1)})$ for $k > 1$. Let $r \in \mathbb{N}$ and $\Lambda := \max \|H_j\|$. Then

$$\left\| \mathcal{V}(-it) - \left(\frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \mathcal{S}_{2k}^{\sigma}(-it/r) \right)^r \right\|_{\diamond}$$
$$\leq 4 \frac{(2 \cdot 5^{k-1} \Lambda |t| L)^{4k+2}}{((2k+1)!)^2 r^{4k+1}} \exp\left(4 \cdot 5^{k-1} \frac{\Lambda |t| L}{r}\right) + 2 \frac{(2 \cdot 5^{k-1} \Lambda |t|)^{2k+1} L^{2k}}{(2k-1)! r^{2k}} \exp\left(2 \cdot 5^{k-1} \frac{\Lambda |t| L}{r}\right) \tag{47}$$

where, for $\lambda = -it$, we associate quantum channels $\mathcal{V}(\lambda)$ and $\mathcal{S}_{2k}^{\sigma}(\lambda)$ with the unitaries $V(\lambda)$ and $S_{2k}^{\sigma}(\lambda)$, respectively.

Proof. We first prove that

$$\left\| \mathcal{V}(\lambda) - \frac{1}{L!} \sum_{\sigma \in \mathrm{Sym}(L)} \mathcal{S}_{2k}^{\sigma}(\lambda) \right\|_{\diamond}$$
$$\leq 4 \frac{(2 \cdot 5^{k-1} \Lambda |\lambda| L)^{4k+2}}{((2k+1)!)^2} \exp(4 \cdot 5^{k-1} \Lambda |\lambda| L) + 2 \frac{(2 \cdot 5^{k-1} \Lambda |\lambda|)^{2k+1} L^{2k}}{(2k-1)!} \exp(2 \cdot 5^{k-1} \Lambda |\lambda| L). \tag{48}$$

To this end, note that the sth-order error of $V(\lambda) - S_{2k}^{\sigma}(\lambda)$ is at most

$$\begin{cases} 0 & 0 \leq s \leq 2k, \\ \frac{2(2 \cdot 5^{k-1} \Lambda |\lambda|)^s}{s!} L^s & s > 2k \end{cases} \tag{49}$$

(as before, this follows as in [17, Proof of Proposition F.3]). Thus Lemma 5 gives

$$\|V(\lambda) - S_{2k}^{\sigma}(\lambda)\| \leq 2 \frac{(2 \cdot 5^{k-1} \Lambda |\lambda| L)^{2k+1}}{(2k+1)!} \exp(2 \cdot 5^{k-1} \Lambda |\lambda| L). \tag{50}$$

On the other hand, Lemma 3 implies that the sth-order error of $V(\lambda) - \frac{1}{L!}\sum_{\sigma\in\text{Sym}(L)} S_{2k}^{\sigma}(\lambda)$ is at most

$$
\begin{cases}
0 & 0 \le s \le 2k, \\
\frac{(2\cdot 5^{k-1}\Lambda|\lambda|)^s}{(s-2)!} L^{s-1} & s > 2k,
\end{cases}
\tag{51}
$$

so again Lemma 5 gives

$$
\left\| V(\lambda) - \frac{1}{L!}\sum_{\sigma\in\text{Sym}(L)} S_{2k}^{\sigma}(\lambda) \right\| \le \frac{(2\cdot 5^{k-1}\Lambda|\lambda|)^{2k+1} L^{2k}}{(2k-1)!} \exp(2\cdot 5^{k-1}\Lambda|\lambda|L).
\tag{52}
$$

Equation (48) now follows from Lemma 1 by setting

$$
\begin{aligned}
a &= 2\frac{(2\cdot 5^{k-1}\Lambda|\lambda|L)^{2k+1}}{(2k+1)!} \exp(2\cdot 5^{k-1}\Lambda|\lambda|L), \\
b &= \frac{(2\cdot 5^{k-1}\Lambda|\lambda|)^{2k+1} L^{2k}}{(2k-1)!} \exp(2\cdot 5^{k-1}\Lambda|\lambda|L).
\end{aligned}
\tag{53}
$$

To simulate the evolution for time t, we divide it into r segments. The error within each segment is obtained from (48) by setting $\lambda = -it/r$. Then subadditivity of the diamond norm distance gives

$$
\left\| \mathcal{V}(-it) - \left(\frac{1}{L!}\sum_{\sigma\in\text{Sym}(L)} \mathcal{S}_{2k}^{\sigma}(-it/r)\right)^r \right\|_{\diamond} \le r \left\| \mathcal{V}(-it/r) - \frac{1}{L!}\sum_{\sigma\in\text{Sym}(L)} \mathcal{S}_{2k}^{\sigma}(-it/r) \right\|_{\diamond},
\tag{54}
$$

which completes the proof. $\qquad\square$

5 Algorithm performance and comparisons

We now analyze the complexity of our randomized product formula algorithm. Assume that $k \in \mathbb{N}$ is fixed, $\Lambda = O(1)$ is constant, and $r > tL$. By Theorem 2, the asymptotic error of the $(2k)$th-order randomized product formula is

$$
\left\| \mathcal{V}(-it) - \left(\frac{1}{L!}\sum_{\sigma\in\text{Sym}(L)} \mathcal{S}_{2k}^{\sigma}(-it/r)\right)^r \right\|_{\diamond} \le O\left(\frac{(tL)^{4k+2}}{r^{4k+1}} + \frac{t^{2k+1}L^{2k}}{r^{2k}}\right).
\tag{55}
$$

To guarantee that the simulation error is at most ϵ, we upper bound the right-hand side of (55) by ϵ and solve for r. We find that it suffices to use

$$
\begin{aligned}
r_{2k}^{\text{rand}} &= \max\left\{ O\left(\frac{(tL)^{\frac{4k+2}{4k+1}}}{\epsilon^{\frac{1}{4k+1}}}\right), O\left(\frac{t^{\frac{2k+1}{2k}}L}{\epsilon^{\frac{1}{2k}}}\right) \right\} \\
&= \max\left\{ O\left(tL\left(\frac{tL}{\epsilon}\right)^{\frac{1}{4k+1}}\right), O\left(tL\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right) \right\}
\end{aligned}
\tag{56}
$$

segments, giving a simulation algorithm with

$$
g_{2k}^{\text{rand}} = O(Lr_{2k}^{\text{rand}}) = \max\left\{ O\left(tL^2\left(\frac{tL}{\epsilon}\right)^{\frac{1}{4k+1}}\right), O\left(tL^2\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right) \right\}
\tag{57}
$$

elementary gates.

For comparison, the error in the $(2k)$th-order deterministic formula algorithm is at most [17, Proposition F.4]

$$\left\| V(-it) - [S_{2k}(-it/r)]^r \right\| \leq O\left(\frac{(tL)^{2k+1}}{r^{2k}} \right). \tag{58}$$

While this bound quantifies the simulation error in terms of the spectral-norm distance, it can easily be adapted to the diamond-norm distance using either Lemma 1 or [8, Lemma 7]. This translation introduces only constant-factor overhead, so we have

$$\left\| \mathcal{V}(-it) - [\mathcal{S}_{2k}(-it/r)]^r \right\|_{\diamond} \leq O\left(\frac{(tL)^{2k+1}}{r^{2k}} \right). \tag{59}$$

Therefore, the number of segments that suffice to ensure error at most ϵ satisfies

$$r_{2k}^{\mathrm{det}} = O\left(tL\left(\frac{tL}{\epsilon} \right)^{\frac{1}{2k}} \right), \tag{60}$$

giving an algorithm with

$$g_{2k}^{\mathrm{det}} = O(Lr_{2k}^{\mathrm{det}}) = O\left(tL^2\left(\frac{tL}{\epsilon} \right)^{\frac{1}{2k}} \right) \tag{61}$$

elementary gates. Comparing to (57), we see that the randomized product formula strictly improves the complexity as a function of L. Indeed, the $(2k)$th-order randomized approach either provides an improvement with respect to all parameters of interest over the $(2k)$th order deterministic approach (if the first term of (57) obtains the maximum), or has better dependence on the number of terms in the Hamiltonian than any deterministic formula (if the second term dominates).

We can also compare our result to the commutator bound of [17], which depends on the specific structure of the Hamiltonian. For concreteness, we consider a one-dimensional nearest-neighbor Heisenberg model with a random magnetic field, as studied in [17]. Specifically, let

$$H = \sum_{j=1}^{n} (\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j \sigma_j^z) \tag{62}$$

with periodic boundary conditions (i.e., $\vec{\sigma}_{n+1} = \vec{\sigma}_1$), and $h_j \in [-h, h]$ chosen uniformly at random, where $\vec{\sigma}_j = (\sigma_j^x, \sigma_j^y, \sigma_j^z)$ denotes a vector of Pauli x, y, and z matrices on qubit j. The $(2k)$th-order deterministic formula with the commutator bound has error at most [17, Eq. (146)]

$$\left\| \mathcal{V}(-it) - [\mathcal{S}_{2k}(-it/r)]^r \right\|_{\diamond} \leq O\left(\frac{(tL)^{2k+2}}{r^{2k+1}} + \frac{t^{2k+1}L^{2k}}{r^{2k}} \right), \tag{63}$$

where we have again used Lemma 1 (or [8, Lemma 7]) to relate the spectral-norm distance to the diamond-norm distance. To guarantee that the simulation error is at most ϵ, it suffices to choose

$$
\begin{aligned}
r_{2k}^{\mathrm{comm}} &= \max\left\{ O\left(\frac{(tL)^{\frac{2k+2}{2k+1}}}{\epsilon^{\frac{1}{2k+1}}} \right), O\left(\frac{t^{\frac{2k+1}{2k}}L}{\epsilon^{\frac{1}{2k}}} \right) \right\} \\
&= \max\left\{ O\left(tL\left(\frac{tL}{\epsilon} \right)^{\frac{1}{2k+1}} \right), O\left(tL\left(\frac{t}{\epsilon} \right)^{\frac{1}{2k}} \right) \right\}
\end{aligned}
\tag{64}
$$

segments, giving an algorithm with

$$g_{2k}^{\mathrm{comm}} = O(Lr_{2k}^{\mathrm{comm}}) = \max\left\{ O\left(tL^2\left(\frac{tL}{\epsilon}\right)^{\frac{1}{2k+1}}\right), O\left(tL^2\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right)\right\} \tag{65}$$

elementary gates. Comparing to the corresponding bound (57) for randomized product formulas, we see that the only difference is that the exponent $1/(2k+1)$ for the commutator bound becomes $1/(4k+1)$ in the randomized case. Thus the randomized approach can provide a slightly faster algorithm despite using less information about the structure of the Hamiltonian. More specifically, the relationship between t and L determines whether the randomized approach offers an improvement. If $t = \Omega(L^{2k})$, then the second term of (65) achieves the maximum, and both approaches have asymptotic complexity $O\left(tL^2\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right)$. However, if $t = o(L^{2k})$, then the randomized formula is advantageous.

6 Empirical performance

While randomization provides a useful theoretical handle for establishing better provable bounds, those bounds may still be far from tight. As described in Section 1, our original motivation for considering randomization was the observation that product formulas appear to perform dramatically better in practice than the best available proven bounds would suggest. To investigate the empirical behavior of product formulas, we numerically evaluate their performance for simulations of the Heisenberg model (62) with $t = n$ and $h = 1$, targeting error $\epsilon = 10^{-3}$, as previously considered in [17]. We collect data for the first-, fourth-, and sixth-order formulas as the latter two orders have the best performance in practice for small n and the first-order formula offers a qualitatively better theoretical improvement.

For the deterministic formula, we order the operators of the Hamiltonian in the same way as [17], namely

$$\sigma_1^x\sigma_2^x,\ldots,\sigma_{n-1}^x\sigma_n^x,\sigma_n^x\sigma_1^x,\ \sigma_1^y\sigma_2^y,\ldots,\sigma_{n-1}^y\sigma_n^y,\sigma_n^y\sigma_1^y,\ \sigma_1^z\sigma_2^z,\ldots,\sigma_{n-1}^z\sigma_n^z,\sigma_n^z\sigma_1^z,\ \sigma_1^z,\ldots,\sigma_n^z. \tag{66}$$

We compute the error in terms of the spectral-norm distance and convert it to the diamond-norm distance using Lemma 7 of [8] (i.e., we multiply by 2). To analyze the randomized formula, we would like to numerically evaluate the diamond-norm distances

$$\left\| \mathcal{V}(-it) - \frac{1}{2^r}\left(\mathcal{S}_1(-it/r) + \mathcal{S}_1^{\mathrm{rev}}(-it/r)\right)^r \right\|_\diamond \tag{67}$$

and

$$\left\| \mathcal{V}(-it) - \left(\frac{1}{L!}\sum_{\sigma\in\mathrm{Sym}(L)} \mathcal{S}_{2k}^\sigma(-it/r)\right)^r \right\|_\diamond. \tag{68}$$

While the diamond norm can be computed using a semidefinite program [34], direct computation is prohibitive as the channel contains $(L!)^r$ Kraus operators. Instead, we use Lemma 1 to estimate the error. We randomly choose the ordering of the summands in each of the r segments, exponentiate each individual operator, and construct a unitary operator by concatenating the exponentials according to the given product formula. We follow this procedure to obtain a Monte Carlo estimate of the average error

$$\left\| V(-it) - \frac{1}{M}\sum_{m=1}^M S_{2k}^{\sigma_{m,r}}(-it/r)\cdots S_{2k}^{\sigma_{m,1}}(-it/r) \right\| \tag{69}$$
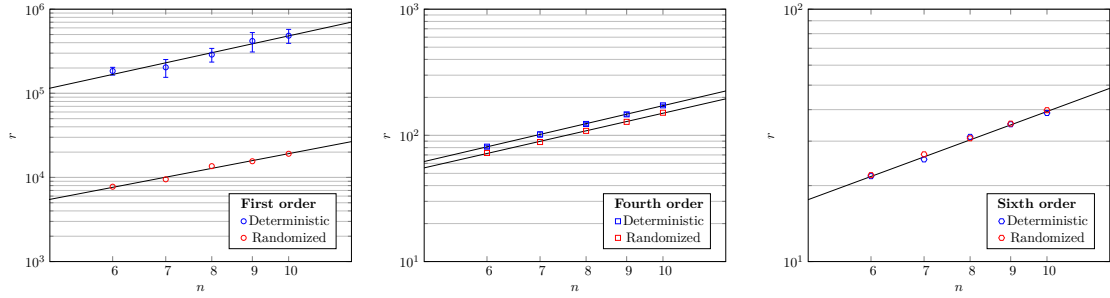
Figure 1: Comparison of the values of r between deterministic and randomized product formulas. Error bars are omitted when they are negligibly small on the plot. Straight lines show power-law fits to the data.

for the $(2k)$th-order formula and similarly for the first-order case. Here, M is the number of samples in the Monte Carlo estimation, which can be increased to get more accurate estimate. In practice, we find that it suffices to take only three samples, as the standard deviations are already negligibly small (about 10^{-5}). We then invoke Lemma 1 to bound the diamond-norm error in (68). To the extent that the bound of Lemma 1 is loose, we expect the empirical performance to be better in practice.

Using five randomly generated instances for each value of n, we apply binary search to determine the smallest number of segments r that suffices to give error at most 10^{-3}. Figure 1 shows the resulting data for the first-, fourth-, and sixth-order formulas, which are well-approximated by power laws. Fitting the data, we estimate that

$$r_1^{\text{remp}} = 300.0n^{1.806} \qquad r_4^{\text{remp}} = 5.458n^{1.439} \qquad r_6^{\text{remp}} = 2.804n^{1.152} \qquad (70)$$

segments should suffice to give error at most 10^{-3}. We thus observe that the empirical complexity of the randomized algorithm is still significantly better than the provable performance

$$r_1^{\text{rand}} = O(n^3) \qquad r_4^{\text{rand}} = O(n^{2.25}) \qquad r_6^{\text{rand}} = O(n^{2.17}). \qquad (71)$$

For comparison, analogous empirical fits for deterministic formulas give the comparable values

$$r_1^{\text{demp}} = 4143n^{2.066} \qquad r_4^{\text{demp}} = 5.821n^{1.471} \qquad r_6^{\text{demp}} = 2.719n^{1.160}, \qquad (72)$$

(cf. [17, Eq. (147)], but note that we have generated new data using [8, Lemma 7] to bound the diamond-norm distance in terms of the spectral-norm distance), whereas the rigorous commutator bound gives the larger exponents [17]

$$r_1^{\text{comm}} = O(n^3) \qquad r_4^{\text{comm}} = O(n^{2.4}) \qquad r_6^{\text{comm}} = O(n^{2.28}). \qquad (73)$$

We see that the randomized bound offers significantly better empirical performance at first order, consistent with the observation that randomization improves the order of approximation in this case. The fourth-order formula slightly improves both the exponent and the constant factor. While this improvement is small, it is nevertheless notable since it involves only a minor change to the algorithm. At sixth order we see negligible improvement. Since the proven bounds give less improvement with each successive order, it is perhaps not surprising to see that the empirical performance shows similar behavior.

To illustrate the effect of using different formulas and different error bounds to simulate larger systems, Figure 2 compares the cost of simulating our model system for sizes up to
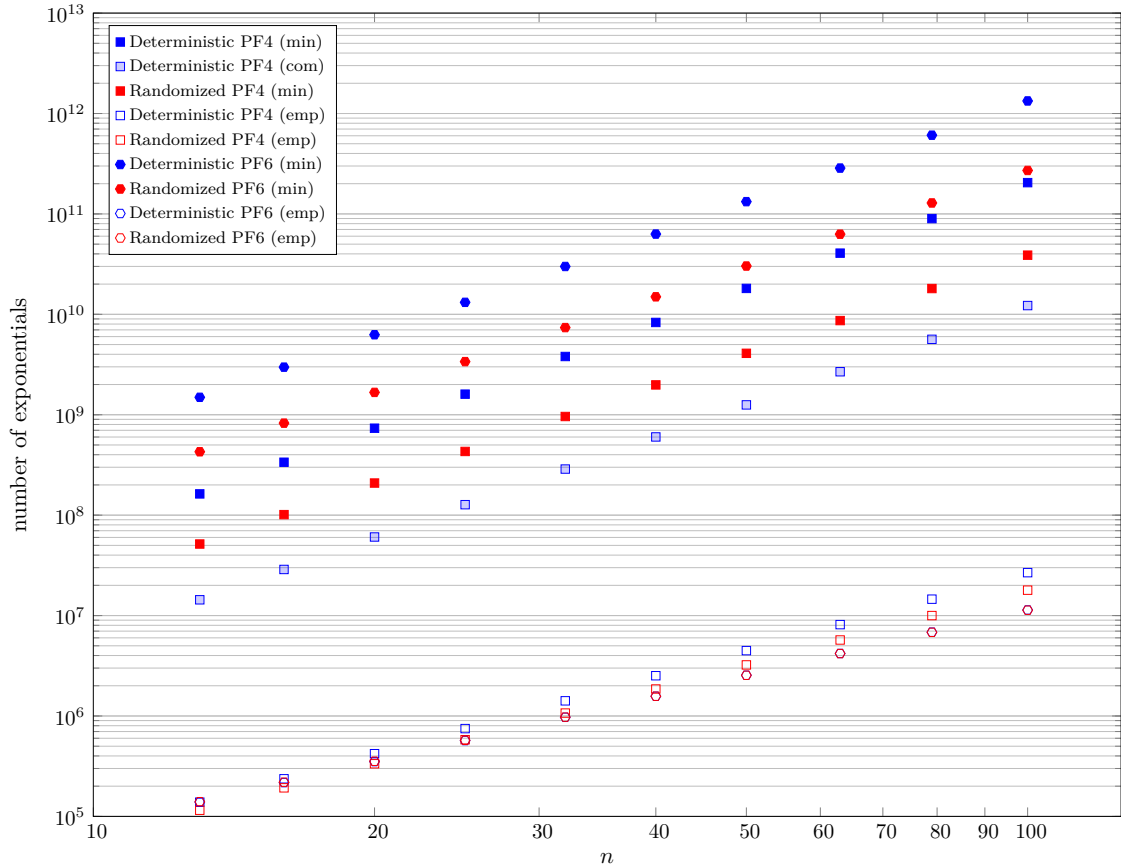
Figure 2: Comparison of the total number of elementary exponentials for product formula simulations of the Heisenberg model using deterministic and randomized product formulas of fourth and sixth order with both rigorous and empirical error bounds. Note that since the empirical performance of deterministic and randomized sixth-order product formulas is almost the same, the latter data points are obscured by the former.

$n = 100$ with deterministic and randomized formulas of orders 4 and 6, using both proven error bounds and the above empirical estimates. (We omit the first-order formula since it is not competitive even at such small sizes.) We give rigorous bounds for deterministic formulas using the minimized bound of [17], and for fourth order we also show the result of using the commutator bound. We see that randomization gives a significant improvement over the deterministic formula using the minimized bound, although the commutator bound outperforms the randomized bound at the system sizes shown here. For sufficiently large n, the randomized bound gives lower complexity, but this requires a fairly large n since the difference in exponents is small and the commutator bound achieves a favorable constant prefactor. Empirical estimates of the error improve the performance by several orders of magnitude, with randomization giving a small advantage for the fourth-order formula as indicated above. However, for systems of size larger than about $n = 25$, the sixth-order bound prevails, and in this case randomization no longer offers a significant advantage.

7 Discussion

We have shown that randomization can be used to establish better performance for quantum simulation algorithms based on product formulas. By simply randomizing how the

summands in the Hamiltonian are ordered, we introduce terms in the average evolution that could not appear in any deterministic product formula approximation of the same order, and thereby give a more efficient algorithm. Indeed, this approach can outperform the commutator bound even though that method uses more information about the structure of the Hamiltonian. A randomized product formula simulation algorithm is not much more complicated than the corresponding deterministic formula, using only $O(L \log L)$ bits of randomness per segment and no ancilla qubits. Furthermore, we showed that randomization can even offer improved empirical performance in some cases.

While randomization has allowed us to make some progress on the challenge of proving better bounds on the performance of product formulas, our strengthened bounds remain far from the apparent empirical performance. We expect that other ideas will be required to improve the product-formula approach [15, 28]. Although our bounds have better asymptotic n-dependence than the previous commutator bound, they only offer an improvement if the system is sufficiently large. It could be fruitful to establish bounds for randomized product formulas that take advantage of the structure of the Hamiltonian, perhaps offering better performance both asymptotically and for small system sizes. More generally, it may be of interest to investigate other scenarios in which random choices can be used to improve the analysis of quantum simulation [10, 14] and other quantum algorithms.

Acknowledgments

References

[1] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 20–29, 2003. DOI: 10.1145/780542.780546. arXiv:quant-ph/0301023.

[2] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation. *Physical Review A*, 91:022311, 2015. DOI: 10.1103/PhysRevA.91.022311. arXiv:1410.8159.

[3] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. G. Fowler, A Megrant, E Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, E. Solano, and John M. Martinis. Digital quantum simulation of fermionic models with a superconducting circuit. *Nature Communications*, 6:7654, 2015. DOI: 10.1038/ncomms8654. arXiv:1501.07703.

[4] Dominic W. Berry and Andrew M. Childs. Black-box Hamiltonian simulation and unitary implementation. *Quantum Information and Computation*, 12(1-2):29–62, 2012. arXiv:0910.4157.

[5] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007. DOI: 10.1007/s00220-006-0150-x. arXiv:quant-ph/0508139.

[6] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the 46th ACM Symposium on Theory of Computing*, pages 283–292, 2014. DOI: 10.1145/2591796.2591854. arXiv:1312.1414.

[7] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, 2015. DOI: 10.1103/PhysRevLett.114.090502. arXiv:1412.4687.

[8] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science*, pages 792–809, 2015. DOI: 10.1109/FOCS.2015.54. arXiv:1501.01715.

[9] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356:1057–1081, 2017. DOI: 10.1007/s00220-017-3002-y. arXiv:1701.03684.

[10] Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling, 2019. arXiv:1906.07115.

[11] Fernando G. S. L. Brandao and Krysta M. Svore. Quantum speed-ups for solving semidefinite programs. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 415–426, 2017. DOI: 10.1109/FOCS.2017.45. arXiv:1609.05537.

[12] Kenneth R. Brown, Robert J. Clark, and Isaac L. Chuang. Limitations of quantum simulation examined by simulating a pairing Hamiltonian using nuclear magnetic resonance. *Physical Review Letters*, 97:050504, 2006. DOI: 10.1103/PhysRevLett.97.050504. arXiv:quant-ph/0601021.

[13] Earl Campbell. Shorter gate sequences for quantum computing by mixing unitaries. *Physical Review A*, 95:042306, Apr 2017. DOI: 10.1103/PhysRevA.95.042306. arXiv:1612.02689.

[14] Earl Campbell. Random compiler for fast Hamiltonian simulation. *Physical Review Letters*, 123:070503, Aug 2019. DOI: 10.1103/PhysRevLett.123.070503. arXiv:1811.08017.

[15] Andrew M. Childs and Yuan Su. Nearly optimal lattice simulation by product formulas. *Physical Review Letters*, 123:050503, Aug 2019. DOI: 10.1103/PhysRevLett.123.050503. arXiv:1901.00564.

[16] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by quantum walk. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 59–68, 2003. DOI: 10.1145/780542.780552. arXiv:quant-ph/0209131.

[17] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, 2018. DOI: 10.1073/pnas.1801723115. arXiv:1711.10980.

[18] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008. DOI: 10.4086/toc.2008.v004a008.

[19] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982. DOI: 10.1007/BF02650179.

[20] Jeongwan Haah, Matthew B. Hastings, Robin Kothari, and Guang Hao Low. Quantum

algorithm for simulating real time evolution of lattice Hamiltonians. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 350–360, Oct 2018. DOI: 10.1109/FOCS.2018.00041. arXiv:1801.03922.

[21] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. DOI: 10.1103/PhysRevLett.103.150502. arXiv:0811.3171.

[22] Matthew B. Hastings. Turning gate synthesis errors into incoherent errors. *Quantum Information and Computation*, 17(5-6):488–494, 2017. arXiv:1612.01011.

[23] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012. DOI: 10.1126/science.1217069. arXiv:1111.3633.

[24] 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(6052):57–61, 2011. DOI: 10.1126/science.1208001. arXiv:1109.1512.

[25] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996. DOI: 10.1126/science.273.5278.1073.

[26] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118:010501, 2017. DOI: 10.1103/PhysRevLett.118.010501. arXiv:1606.02685.

[27] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019. DOI: 10.22331/q-2019-07-12-163. arXiv:1610.06546.

[28] Guang Hao Low, Vadym Kliuchnikov, and Nathan Wiebe. Well-conditioned multi-product Hamiltonian simulation, 2019. arXiv:1907.11679.

[29] David Poulin, Angie Qarry, Rolando D. Somma, and Frank Verstraete. Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space. *Physical Review Letters*, 106(17):170501, 2011. DOI: 10.1103/PhysRevLett.106.170501. arXiv:1102.1360.

[30] David Poulin, Matthew B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doherty, and Matthias Troyer. The Trotter step size required for accurate quantum simulation of quantum chemistry. *Quantum Information and Computation*, 15(5-6): 361–384, 2015. arXiv:1406.4920.

[31] Sadegh Raeisi, Nathan Wiebe, and Barry C. Sanders. Quantum-circuit design for efficient simulations of many-body quantum dynamics. *New Journal of Physics*, 14: 103017, 2012. DOI: 10.1088/1367-2630/14/10/103017. arXiv:1108.4318.

[32] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 114(29):7555–7560, 2017. DOI: 10.1073/pnas.1619152114. arXiv:1605.03590.

[33] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991. DOI: 10.1063/1.529425.

[34] John Watrous. Simpler semidefinite programs for completely bounded norms. *Chicago Journal of Theoretical Computer Science*, 2013(8), 2013. DOI: 10.4086/cjtcs.2013.008.

[35] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. DOI: 10.1017/9781316848142.

[36] Dave Wecker, Bela Bauer, Bryan K. Clark, Matthew B. Hastings, and Matthias Troyer. Gate count estimates for performing quantum chemistry on small quantum

computers. *Physical Review A*, 90:022305, 2014. DOI: 10.1103/PhysRevA.90.022305. arXiv:1312.1695.

[37] Chi Zhang. Randomized algorithms for Hamiltonian simulation. In Leszek Plaskota and Henryk Woźniakowski, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 709–719, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-27440-4. DOI: 10.1007/978-3-642-27440-4_42.