Journal of Physics: Conference Series **2970** (2025) 012004 doi:10.1088/1742-6596/2970/1/012004

Efficient Quantum Circuit Implementations of The qDRIFT Algorithm via Linear Combinations of Unitaries and Quantum Forking

I. J. David[1,2], I. Sinayskiy[1,2] and F. Petruccione[2,3]

[1]School of Chemistry and Physics, University of KwaZulu-Natal, Durban 4001, South Africa.
[2]National Institute for Theoretical and Computational Sciences (NITheCS), Stellenbosch, South Africa.
[3]School of Data Science and Computational Thinking, Department Of Physics, Stellenbosch University, Stellenbosch 7604, South Africa.

E-mail: ian.david@nithecs.ac.za, sinayskiy@ukzn.ac.za, petruccione@sun.ac.za

Abstract.

Quantum computers can be used to simulate the evolution of quantum systems via the discretization of this evolution into unitary gate sequences. While Trotter-Suzuki (TS) methods stand as the most prevalent approach for quantum simulation, their effectiveness is contingent upon the sparsity of the Hamiltonian system. This is due to the precision and gate complexity of the TS method being dependent on the number of terms within the Hamiltonian. Despite attempts to mitigate this dependence through the random permutation of terms within TS methods, scalability issues persist. To address this, Campbell introduced the qDRIFT algorithm, leveraging random compilation to alleviate the reliance on the Hamiltonian's term count. However, the initial implementation of this algorithm necessitated the involvement of classical computing resources for gate set compilation. Our study presents two novel quantum circuit implementations of the qDRIFT algorithm, eliminating the need for classical computing resources. We employ the Linear Combination of Unitaries (LCU) approach and the Quantum Forking (QF) procedure to construct efficient circuits to implement the qDRIFT algorithm. Furthermore, we conduct a comparative analysis of the gate complexities between our proposed circuits and TS methods, identifying the scenarios in which our circuits exhibit superior performance.

1 Introduction

The idea that Quantum computers can be used to simulate the dynamics of other quantum systems was proposed by Feynman and Manin independently [1, 2]. The idea was that quantum computers could provide a computational advantage over classical computers in simulating these systems to help us better understand physical systems. Digital quantum computation relies on breaking down a computation into a sequence of discrete single and two-qubit gates [3]. Therefore, if we want to simulate the dynamics of quantum systems, which are described by continuous unitary evolution, we need to be able to approximate this continuous unitary evolution as a discrete sequence of quantum gates. This procedure is called quantum simulation [4]. The standard way to approximate the dynamics is by the Trotter-Suzuki (TS) product formulas [5, 6, 7], which break down the unitary evolution into a sequence of simpler unitaries that are generated by decomposing the Hamiltonian of the system in a specific

manner. Although there are many novel methods [8, 9, 10, 11, 12], to do quantum simulation TS product formulas are still found to be a good method both empirically [13] and theoretically [11]. However, some Hamiltonians, such as electronic structure Hamiltonians, contain a large number of terms. This presents a problem for TS product formulas as the gate count scales poorly for Hamiltonians with many terms. Therefore, a simulation method without this scaling problem would have many significant applications. In [10], the authors develop a simulation algorithm that implements TS product formulas with randomly permuted terms. This greatly improved the gate counts and the dependence on the number of terms in the Hamiltonian. However, it was still a problem for systems with a larger number of terms in the Hamiltonian because the gate count depended on the number of terms. Campbell [14] developed a quantum simulation algorithm that uses random compilation to build a gate set that will simulate the dynamics up to a given precision, with the gate count not dependent on the number of terms in the Hamiltonian. This was a big improvement over TS product formulas as we were no longer restricted to simulating systems with only sparse interactions to have a good gate count. The algorithm was called the qDRIFT algorithm [14]. With this algorithm, one can better simulate electronic struct Hamiltonians and other problems with Hamiltonians with many terms, such as three-dimensional Jaynes-Cummings-Hubbard Model [15]. One of the problems that the qDRIFT algorithm has, is that it relies on classical random sampling to perform the compilation of the gate set. This may be a problem for some quantum algorithms where quantum simulation must be performed as a sub-routine, and accessing a classical computer is impossible. Therefore, designing an efficient quantum circuit implementation for the qDRIFT algorithm is important, so we do not require classical sampling. In this work we prose the use of the Linear Combinations of Unitaries (LCU) approach [9] and Quantum Forking (QF) [16], to implement the qDRIFT algorithm without classical sampling. We shall construct the circuits to implement qDRIFT and analyse their gate complexities. We demonstrate that the gate complexities for both approaches scale linearly in the number of terms in the Hamiltonian, which is better than higher-order TS, which has quadratic scaling in the large order limit. We shall also discuss how considering the cost of the classical sampling leads to the algorithm in [14] having the same scaling as the circuit implementations done in this work. This will show that these circuit implementations are as efficient as the algorithm in [14]. The rest of this article is outlined as follows. In section 2, we shall discuss some preliminaries, outline the problem of Hamiltonian simulation, and discuss the details of the TS product formula method for simulation. Section 3, will outline the qDRIFT protocol from [14], and we outline how qDRIFT approximates continuous unitary evolution. In section 4 we present our quantum circuit implementations for the qDRIFT algorithm for both the LCU and QF approach and we compute the gate complexities. Section 5 compares the gate complexities of our circuit implementations with TS product formulas, and we illustrate the performance by plotting the gate complexities. Lastly, in section 6 we make some concluding remarks and outline some open problems.

2 Preliminaries

The state space of a d-dimensional quantum system is the d-dimensional Hilbert space $\mathscr{H} \cong \mathbb{C}^d$. The system's state can be described by a d-dimensional unit vector $|\psi\rangle \in \mathscr{H}$, called a state vector. This state is time-dependent when we are in the Schrödinger picture i.e. $|\psi(t)\rangle$ is the state of the system at some time $t \geq 0$. The dynamics of our quantum system is governed by the Schrödinger equation,

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle\,, \tag{1}$$

where $H = H^\dagger$ is the Hamiltonian of the system, which can be represented as a $d \times d$ matrix for finite dimensional systems and we use natural units where $\hbar = 1$. Formally, one solves the Schrödinger equation for the initial state $|\psi(t=0)\rangle = |\psi(0)\rangle$,

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle\,. \tag{2}$$

The operator $\exp(-iHt)$ is unitary and called the time evolution operator. We will usually write $U(t) = \exp(-iHt)$ so that we have,

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle\,, \tag{3}$$

where the fact that $U(t)$ is unitary implies $U(t)^\dagger U(t) = \mathbb{1} = U(t)U(t)^\dagger$. One can equivalently describe our quantum system and its evolution in the language of density operators and quantum channels. This will be useful later on for our description of the qDRIFT protocol (see Section 3). In this description,

the state of our system is described by a density operator $\rho \in \mathcal{B}(\mathscr{H})$, with $\mathcal{B}(\mathscr{H})$ being the space of bounded linear operators acting on the Hilbert space $\mathscr{H}$. Since our system is d-dimensional, the density operator ρ is represented by a $d \times d$ matrix, which we call a density matrix. The term density operator and density matrix are used interchangeably for finite-dimensional systems. The density matrix ρ is a Hermitian and positive matrix that has a trace of 1 i.e.

$$\rho = \rho^\dagger, \qquad \rho \geq 0, \qquad \mathrm{tr}(\rho) = 1. \tag{4}$$

The density matrix can also be time-dependent, i.e. $\rho(t)$ is the state of the system at some time $t \geq 0$. Then, the dynamics of the system are governed by the Schrödinger equation, for the density matrix,

$$\frac{\partial}{\partial t}\rho(t) = -i\left[H, \rho(t)\right] = \mathcal{H}\left(\rho(t)\right), \tag{5}$$

where $\mathcal{H}(*) = -i[H, *]$ is called a superoperator (because it is an operator that acts on other operators) and H is again the Hamiltonian of the system. The formal solution of this equation is,

$$\rho(t) = e^{-iHt}\rho(0)e^{iHt} = U(t)\rho(0)U(t)^\dagger, \tag{6}$$

which describes the evolution of the system by the unitary operator $U(t)$. This tells us that if we can compute $U(t)$, we can fully describe the dynamics of our system. However, this is highly non-trivial as computing exponentials of operators is computationally hard. This leads us to the problem of quantum simulation, which we shall discuss formally. Consider the Hamiltonian H with the following decomposition,

$$H = \sum_{j=1}^{M} h_j H_j, \tag{7}$$

where $h_j \in \mathbb{R}$ and each H_j is Hermitian operator with the same dimensions as H and it is normalised so that $\|H_j\| = 1$, for $j = 1, ..., M$. Where $\|\cdot\|$ denotes the operator norm, which is equal to the largest singular value of an operator. One should note that we can always choose H_j so that the weighting h_j are all positive and real numbers, and we can bound the norm of the Hamiltonian as follows,

$$\|H\| = \left\| \sum_{j=1}^{M} h_j H_j \right\| \leq \sum_{j=1}^{M} h_j \|H_j\| \leq \sum_{j=1}^{M} h_j, \tag{8}$$

this bound will be important throughout this text so we define for convenience $\lambda = \sum_{j=1}^{M} h_j$. The decomposition of the Hamiltonian H should be chosen so that for each H_j the unitary $\exp(-iH_j\tau)$ can be easily implemented on a quantum computer for any $\tau \in \mathbb{R}$. In quantum simulation, the goal then is to find an approximation of $\exp(-iHt)$ into a sequence of gates $\exp(-iH_j\tau)$ up to some desired precision $\epsilon > 0$. We use the number of unitaries $\exp(-iH_j\tau)$ to quantify the cost of the quantum computation. We aim to minimise the number of unitaries used in the approximation. A common way to approximate $\exp(-iHt)$ is to use a Trotter-Suzuki (TS) product formula [5, 6, 7]. The TS product formula uses products of unitaries $\exp(-iH_j\tau)$ to approximate $\exp(-iHt)$ up to any desired precision. Here we summarise briefly how TS product formulas can be used. First, we consider a simulation time $t \geq 0$, and we divide the time interval from 0 to t into r segments, i.e. t/r. Then, define the first-order TS product formula as,

$$S_1(\tau) := \prod_{j=1}^{M} e^{-iH_j\tau}, \tag{9}$$

where $\tau \in \mathbb{R}$ and this formula approximates $\exp(-iH\tau)$ up to first order. Then if we consider $S_1(t/r)^r$ then this will approximate $\exp(-iHt)$ in the large r limit i.e.

$$e^{-iHt} \approx S_1(t/r)^r = \left(\prod_{j=1}^{M} e^{-iH_j t/r} \right)^r. \tag{10}$$

Since we want to approximate $\exp(-iHt)$ up to some desired precision $\epsilon > 0$ we usually need to show,

$$\left\| e^{-iHt} - S_1(t/r)^r \right\| \leq \epsilon. \tag{11}$$

In [5], the precision to which the first order TS product formula approximates $\exp(-iHt)$ is calculated and it is shown that,

$$\left\| e^{-iHt} - S_1(t/r)^r \right\| = O\left(\frac{(M\Lambda t)^2}{r} \right) \tag{12}$$

where $\Lambda = \max_j h_j$. From this one can determine upper bound on the number of segments r given a precision ϵ as,

$$r = O\left(\frac{(M\Lambda t)^2}{\epsilon} \right). \tag{13}$$

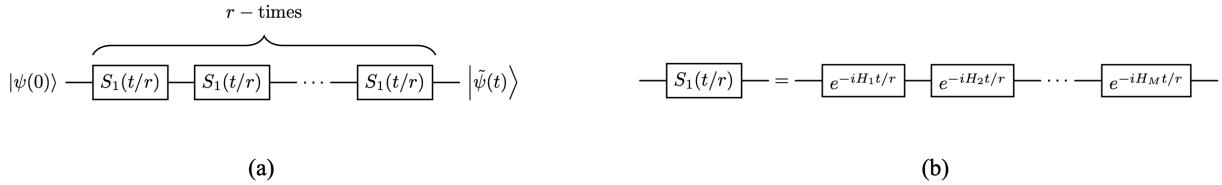

Figure 1: (a) The quantum circuit that implements the first order TS product formula $S_1(t/r)^r$ to an initial state $|\psi(0)\rangle$ with the output state $|\tilde{\psi}(0)\rangle$ which approximates the state $|\psi(t)\rangle$ to a precision ϵ. (b) This circuit shows how we can express $S_1(t/r)$ in terms of the the exponentials $\exp(-iH_j t/r)$.

Now to simulate $\exp(-iHt)$ using the first order TS product formula, we decompose the Hamiltonian H into the form (7), we specify a precision $\epsilon > 0$ and a simulation time $t \geq 0$, then we compute the number of segments $r = \lceil (M\Lambda t)^2/\epsilon \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. Using r, we dived the time t into r segments, i.e. t/r, then we constructed a quantum circuit that implements $S_1(t/r)$ as seen in Figure 1. When constructing any quantum circuit, one always wants to analyse the gate complexity of the circuit. Here, we define the gate complexity as an upper bound on the number of sequential unitary gate layers that act on the input register of our quantum computer, where a unitary gate layer is defined as a collection of gates that can be implemented in parallel at the same step of our circuit. For example, if we have a quantum computer with an input register of $|0\rangle \otimes |0\rangle$ and we apply $U_1 = X$ and $U_2 = H \otimes H$ to this register, the circuit is shown in Figure 2, we see that in total there are three quantum gates however there are only two gate layers being applied to the input register U_1 and U_2.

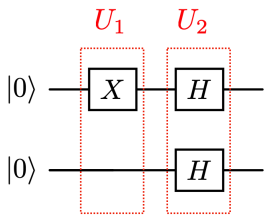

Figure 2: Quantum Circuit showing two gate layers U_1 and U_2, with a total of three single qubit gates. This circuit illustrates our notion of gate complexity as we shall only count sequential gate layers and not gates on individual registers.

We can then compute the gate complexity of the circuit by counting the number of exponentials $\exp(-iH_j t/r)$ that we need to implement, as these will be the sequential unitary gate layers applied to

the register $|\psi(0)\rangle$. We see that the gate complexity g is then M times the number of segments r, i.e.

$$g = O\left(\frac{M^3(\Lambda t)^2}{\epsilon}\right). \tag{14}$$

The output of the circuit is shown in Figure 1. (a) is some quantum state $|\tilde{\psi}(t)\rangle$ which approximates the state $|\psi(t)\rangle$ up to a precision ϵ, this can easily be seen by using the properties of the norm,

$$\begin{aligned}
\left\||\psi(t)\rangle - |\tilde{\psi}(t)\rangle\right\| &= \left\|e^{-iHt}|\psi(0)\rangle - S_1(t/r)^r|\psi(0)\rangle\right\|, &(15)\\
&= \left\|\left(e^{-iHt} - S_1(t/r)^r\right)|\psi(0)\rangle\right\|, &(16)\\
&\leq \left\|e^{-iHt} - S_1(t/r)^r\right\|\||\psi(0)\rangle\|, &(17)\\
&= \left\|e^{-iHt} - S_1(t/r)^r\right\|, &(18)\\
&\leq \epsilon. &(19)
\end{aligned}$$

Now that we have seen how we can use the first order TS product formula to simulate the evolution of a system with the Hamiltonian H up to a precision ϵ, the natural question is, can we simulate the evolution with higher precision and an improved gate complexity?
In [6, 7], Suzuki systematically extended the product formulas so that they can be used to approximate up to an arbitrarily large order, which increases the precision. The second-order product formula is then defined as,

$$S_2(\tau) := \prod_{j=1}^{M} e^{-iH_j\frac{\tau}{2}} \prod_{j'=M}^{1} e^{-iH_{j'}\frac{\tau}{2}}, \tag{20}$$

using this formula, we can recursively construct product formulas up to the $2k$-th order as follows,

$$S_{2k}(\tau) := S_{2k-2}(z_k\tau)^2 S_{2k-2}((1-4z_k)\tau)S_{2k-2}(z_k\tau)^2, \tag{21}$$

where $z_k := 1/(4 - 4^{1/(2k-1)})$. This formula approximates e^{-iHt} to an order of $2k$, with an upper bound on the precision of the approximation, which was found to be [5],

$$\left\|e^{-iHt} - S_{2k}(t/r)^r\right\| = O\left(\frac{(M\Lambda t)^{2k+1}}{r^{2k}}\right). \tag{22}$$

The number of segments r can then be bounded as,

$$r = O\left(\frac{(M\Lambda t)^{1+\frac{1}{2k}}}{\epsilon^{1/2k}}\right). \tag{23}$$

From this, the gate count is easily obtained and is shown to be bounded as,

$$g = O\left(\frac{M^{2+\frac{1}{2k}}(\Lambda t)^{1+\frac{1}{2k}}}{\epsilon^{1/2k}}\right). \tag{24}$$

From the gate count in (24), we see that as we increase the order by increasing k, the scaling approaches $O(M^2\Lambda t)$. However, the constant pre-factors become worse for higher orders, so in practice, the optimal choice is usually fourth or sixth order. Now that we have reviewed the product formula approach to quantum simulation, we observe that the dependence on M, the number of terms in H, never improved below quadratic scaling. Therefore, TS product formulas are limited to systems simulation with sparse interactions. In the next section, we discuss the qDRIFT protocol, which will eliminate this dependence on M.

3 Overview of The qDRIFT Protocol For Hamiltonian Simulation

The qDRIFT algorithm, as described in [14], constructs a gate sequence by selecting an exponential $\exp(-iH_j\tau)$ from an identical distribution (i.i.d sampling). Where the probability p_j, of selecting an exponential $\exp(-iH_j\tau)$, is $p_j = h_j/\lambda$ and it is dependent on the strength h_j of each term in the Hamiltonian. This quantum process is random, however built into the probabilities is a bias so that with many repetitions the evolution stochastically drifts towards $\exp(-iHt)$. For this reason, the algorithm is called qDRIFT. Since each exponential is sampled independently, the process is Markovian, and we can consider the evolution as a result of a single random operation. A quantum channel can then describe this evolution,

$$\mathcal{E}_\tau(\rho) = \sum_{j=1}^{M} p_j e^{-iH_j\tau}\rho e^{iH_j\tau}, \tag{25}$$

which applies each unitary $\exp(-iH_j\tau)$ with a probability p_j and for $\tau = t\lambda/N$, with N being the number of repetitions of the qDRIFT channel needed. From this point onwards, we will refer to the channel $\mathcal{E}_\tau$ as the qDRIFT channel. Theorem 1. will outline the how the qDRIFT channel approximates the evolution $\exp(-iHt)$ and it will provide a bound on N.

Theorem 1. Given a Hamiltonian H for a quantum system in the form (7), the initial density matrix of the system $\rho(0)$ and a time $t \geq 0$, the evolution of the system is described by the quantum channel,

$$\mathcal{U}_t(\rho(0)) = e^{-iHt}\rho(0)e^{iHt}, \tag{26}$$

where $\rho(t) = \mathcal{U}_t(\rho(0))$. Then, given a precision $\epsilon > 0$ and $N \in \mathbb{N}$. The diamond norm of the difference between qDRIFT channel $\mathcal{E}_\tau^N$ in (25) and $\mathcal{U}_t$ can be bounded as,

$$\left\|\mathcal{U}_t - \mathcal{E}_\tau^N\right\|_\diamond \leq \epsilon. \tag{27}$$

where,

$$\epsilon = O\left(\frac{\lambda^2 t^2}{N}\right), \qquad N = O\left(\frac{\lambda^2 t^2}{\epsilon}\right). \tag{28}$$

Proof. To prove the bounds on the precision ϵ and the number of repetitions N we need to use the diamond norm to compute how accurately the qDRIFT channel approximates $\mathcal{U}_t$. The diamond norm [17] is a superoperator norm defined on the space of quantum channels, and more information about it can be found in Appendix A. Now, if we consider the diamond norm of the difference between $\mathcal{E}_\tau^N$ and $\mathcal{U}_t$,

$$\left\|\mathcal{U}_t - \mathcal{E}_\tau^N\right\|_\diamond = \left\|\mathcal{U}_{t/N}^N - \mathcal{E}_\tau^N\right\|_\diamond, \tag{29}$$

$$\leq N\left\|\mathcal{U}_{t/N} - \mathcal{E}_\tau\right\|_\diamond, \tag{30}$$

where in (30) we have used Lemma 3. from Appendix A. From (30) we see that it suffices to find a bound on $\left\|\mathcal{U}_{t/N} - \mathcal{E}_\tau\right\|_\diamond$. We do this by making use of the Liouvillian representation of the quantum channel so that for the unitary evolution $\mathcal{U}_{t/N}$ we can write,

$$\mathcal{U}_{t/n}(\rho) = e^{\mathcal{H}t/N}(\rho) = \sum_{l=0}^{\infty} \frac{t^l \mathcal{H}^l(\rho)}{l! N^l}, \tag{31}$$

where $\mathcal{H}(\rho) = -i[H, \rho]$, for any density matrix ρ. For the qDRIFT channel, we can also use this representation,

$$\mathcal{E}_\tau(\rho) = \sum_{j=1}^{M} p_j e^{\mathcal{H}_j\tau}(\rho) = \sum_{j=1}^{M} \frac{h_j}{\lambda} e^{\mathcal{H}_j\tau}(\rho), \tag{32}$$

where ρ is once again any density matrix and we have defined $\mathcal{H}_j(\rho) = -i[H_j, \rho]$. We will use the Liouville representation of the channels for the rest of this proof. Now consider the difference of the

Taylor expansion of $\mathcal{U}_{t/N}$ and $\mathcal{E}_\tau$ and observe that the zeroth and first-order terms cancel, leaving us with the following remainder terms,

$$\mathcal{U}_{t/N} - \mathcal{E}_\tau = \sum_{l=2}^{\infty} \frac{t^l \mathcal{H}^l}{l! N^l} - \sum_{j=1}^{M} \frac{h_j}{\lambda} \sum_{l=2}^{\infty} \frac{\lambda^l t^l \mathcal{H}_j^l}{l! N^l}. \tag{33}$$

Now, to bound the diamond norm of the difference, we make use of the subadditive and sub-multiplicative property of the diamond norm, and we find that,

$$\left\| \mathcal{U}_{t/N} - \mathcal{E}_\tau \right\|_\diamond \leq \sum_{l=2}^{\infty} \frac{t^l}{l! N^l} \|\mathcal{H}\|_\diamond^l + \sum_{j=1}^{M} \frac{h_j}{\lambda} \sum_{l=2}^{\infty} \frac{\lambda^l t^l}{l! N^l} \|\mathcal{H}_j\|_\diamond^l. \tag{34}$$

Now we observe that,

$$\|\mathcal{H}\|_\diamond \leq 2\|H\| \leq 2\lambda, \tag{35}$$

and we have that

$$\|\mathcal{H}_j\|_\diamond \leq 2\|H_j\| \leq 2. \tag{36}$$

Substituting into (34) yields,

$$\left\| \mathcal{U}_{t/N} - \mathcal{E}_\tau \right\|_\diamond \leq \sum_{l=2}^{\infty} \frac{t^l}{l! N^l} (2\lambda)^l + \sum_{j=1}^{M} \frac{h_j}{\lambda} \sum_{l=2}^{\infty} \frac{\lambda^l t^l}{l! N^l} 2^l, \tag{37}$$

$$= 2 \sum_{l=2}^{\infty} \frac{1}{l!} \left(\frac{2\lambda t}{N} \right)^l. \tag{38}$$

The last equality uses the fact that $\sum_{j=1}^{M} h_j = \lambda$, and it collects together the pair of equal summations. Next, we use the exponential tail bound (see Lemma F.2. from the supplementary information of [13]) that states,

$$\sum_{n=2}^{\infty} \frac{x^n}{n!} \leq \frac{x^2}{2} e^x, \tag{39}$$

and we use $x = 2\lambda t/N$ so that,

$$\left\| \mathcal{U}_{t/N} - \mathcal{E}_\tau \right\|_\diamond \leq \left(\frac{2\lambda t}{N} \right)^2 \exp\left(\frac{2\lambda t}{N} \right) \approx \left(\frac{2\lambda t}{N} \right)^2, \tag{40}$$

where the last approximation is very accurate in the large N limit. Using this bound and substituting into (30) we have,

$$\left\| \mathcal{U}_t - \mathcal{E}_\tau^N \right\|_\diamond \leq \frac{(2\lambda t)^2}{N}, \tag{41}$$

which leads us to the bound on ϵ by choosing $\epsilon > 0$ such that,

$$\epsilon \geq \frac{(2\lambda t)^2}{N}, \tag{42}$$

which implies that,

$$N \geq \frac{(2\lambda t)^2}{\epsilon}. \tag{43}$$

From this we can obtain the upper bound on ϵ and N as,

$$\epsilon = O\left(\frac{\lambda^2 t^2}{N} \right), \qquad N = O\left(\frac{\lambda^2 t^2}{\epsilon} \right), \tag{44}$$

which completes the proof.

Now that we have shown that the qDRIFT channel can approximate $\mathcal{U}_t$ up to a precision ϵ, then by the definition of the diamond norm (see Appendix A) we observe that if we define $\tilde{\rho}(t) = \mathcal{E}_\tau^N(\rho(0))$ as being the state after applying the qDRIFT channel and $\rho(t) = \mathcal{U}_t(\rho(0))$ by the state of the exact state of the system at some time t then,

$$\|\rho(t) - \tilde{\rho}(t)\|_1 = \left\|\mathcal{U}_t(\rho(0)) - \mathcal{E}_\tau^N(\rho(0))\right\|_1, \tag{45}$$

$$\leq \left\|\mathcal{U}_t - \mathcal{E}_\tau^N\right\|_\diamond \leq \epsilon, \tag{46}$$

where $\|\cdot\|_1$ is the trace norm. In [14], the author presents an algorithm for constructing a quantum circuit to implement the qDRIFT protocol using classical sampling to construct a gate set. One can refer to Fig. 1. of [14] for pseudocode that outlines how to use classical sampling to construct the quantum circuit. The algorithm consists of two parts, the first part uses classical sampling to construct a gate set which is a sequence of exponentials $e^{-iH_j\tau}$. It does this by appending $e^{-iH_j\tau}$, where each j is sampled from the distribution p_j, to an empty gate set and it repeats this process N times. The second part of the algorithm applies the gate set to the initial state $|\psi(0)\rangle$, which tells us that in this circuit, there will be N gates. The output of this circuit will be the state $|\tilde{\psi}(t)\rangle$, which approximates the exact state $|\psi(t)\rangle$ to a precision ϵ. This tells us that the qDRIFT protocol in [14], is a hybrid algorithm which relies on a classical part. In the next section, we present a way to efficiently implement the qDRIFT channel on a quantum computer without classical sampling using quantum forking. We will also show how this is more efficient than standard approaches for implementing a random unitary channel [9].

4 Quantum Circuit Implement For qDRIFT Channnel

This section presents two ways to implement the qDRIFT channel on a quantum computer. The first way is a naive implementation that takes inspiration from the Linear Combination of Unitaries (LCU) approach in [9]. The second implementation will use Quantum Forking (QF) [16] to construct an efficient quantum circuit to implement the qDRIFT channel by applying each exponential in parallel. In both cases we are assumed that we have already written the Hamiltonian H in the form (7), we are given a time $t \geq 0$, a precision $\epsilon > 0$ and we have computed λ as well as $N = \lceil \lambda^2 t^2/\epsilon \rceil$.

4.1 Implementation of qDRIFT Channel inspired by LCU Approach

We shall first construct a quantum circuit inspired by the LCU approach [9], that can implement the qDRIFT channel $\mathcal{E}_\tau^N$. We begin by constructing a circuit that impelments $\mathcal{E}_\tau$. This circuit is shown in Figure 3. The first register in this circuit encodes the probabilities p_j into the state $\sum_j p_j |j\rangle$, where $\{|j\rangle\}$ are all orthonormal. The controlled gates shown in the circuit apply $\exp(-iH_j\tau)$ only when the first register is in the state $|j\rangle$ for $j = 1, ..., M$ and they apply identity otherwise. The measurement and discard operation at the end of the first register is the same as tracing out the first register. Lemma 1 will prove that the circuit correctly implements $\mathcal{E}_\tau$.

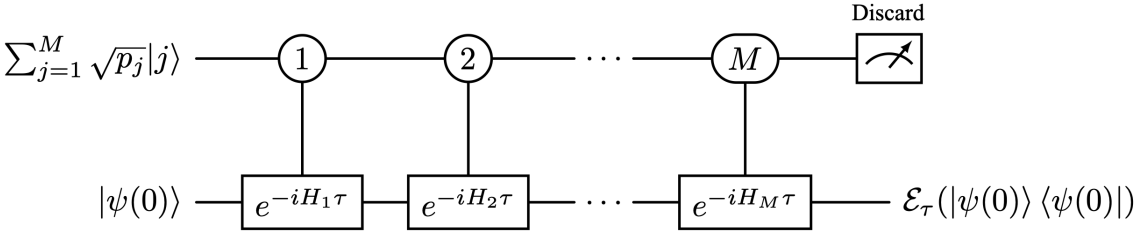

Figure 3: Quantum circuit for implementing a single step of the qDRIFT algorithm $\mathcal{E}_\tau$ using the LCU approach. The measurement and discard procedure is equivalent to tracing out the ancillary register using the partial trace. The controlled-$\exp(-iH_j\tau)$ gates are applied only when the first register is in the state $|j\rangle$.

Lemma 1. The quantum circuit shown in Figure 3 implements the qDRIFT channel $\mathcal{E}_\tau(|\psi(0)\rangle\langle\psi(0)|)$ for $\tau = t\lambda/N$.

Proof. We need to use the density matrix representation to analyse the circuit to prove this. The initial density matrix of both registers in the circuit is in Figure 3. is,

$$\sum_{j,j'=1}^{M} \left(\sqrt{p_j}\sqrt{p_{j'}}\,|j\rangle\langle j'| \otimes |\psi(0)\rangle\langle\psi(0)|\right). \tag{47}$$

We can rewrite this initial density matrix by splitting the sum into two parts: the first part is terms where $j = j'$ and the second part where $j \neq j'$,

$$\sum_{j=1}^{M} \left(p_j\,|j\rangle\langle j| \otimes |\psi(0)\rangle\langle\psi(0)|\right) + \sum_{j,j'=1}^{M} \left(\sqrt{p_j}\sqrt{p_{j'}}\,|j\rangle\langle j'| \otimes |\psi(0)\rangle\langle\psi(0)|\right). \tag{48}$$

Since we will eventually trace out the first register we see that the second sum in (48) will become zero as $\mathrm{tr}(|j\rangle\langle j'|) = \langle j|j'\rangle = 0$ since $\{|j\rangle\}_{j=1}^{M}$ is are orthonormal. Therefore, we will ignore the second sum in (48) and perform the rest of the operations in the circuit only to the first sum of (48). Next, we apply $\exp(-iH_j\tau)$ to the second register only when the first register is in the state $|j\rangle$,

$$\sum_{j=1}^{M} p_j\,|j\rangle\langle j| \otimes e^{-iH_j\tau}\,|\psi(0)\rangle\langle\psi(0)|\,e^{iH_j\tau}. \tag{49}$$

Lastly, we trace out the first register by performing the partial trace,

$$\sum_{j=1}^{M} p_j \mathrm{tr}(|j\rangle\langle j|) \otimes e^{-iH_j\tau}\,|\psi(0)\rangle\langle\psi(0)|\,e^{iH_j\tau} = \sum_{j=1}^{M} p_j e^{-iH_j\tau}\,|\psi(0)\rangle\langle\psi(0)|\,e^{iH_j\tau} = \mathcal{E}_\tau(|\psi(0)\rangle\langle\psi(0)|). \tag{50}$$

Hence, we have shown the circuit in Figure 3. implements $\mathcal{E}_\tau$.

To implement $\mathcal{E}_\tau^N$ we need to repeat the circuit in Figure 3. N times, but after each time we measure and discard, we need to reset the first register to $\sum_{j=1}^{M}\sqrt{p_j}\,|j\rangle$. To determine the gate complexity of the circuit, we must count the sequential operations in the circuit in Figure 3. There are M sequential operations in the circuit. Because we repeat this N times, the number of sequential operations is NM. Therefore if we denote the gate complexity for the LCU circuit by g_{LCU} then,

$$g_{\mathrm{LCU}} = O(MN). \tag{51}$$

One of the downsides to the LCU-inspired circuit implementation for the qDRIFT channel is that it requires the implementation of controlled-$\exp(-iH_j\tau)$ gates, which in practice may be hard to implement on a quantum computer. The QF approach to implementing the qDRIFT channel shall circumvent this issue.

4.2 Implementation of qDRIFT Channel By Quantum Forking

We use QF [16], to construct a quantum circuit that will implement $\mathcal{E}_\tau^N$. The circuit that will implement $\mathcal{E}_\tau$ is shown in Figure 4, where the first register in this circuit encodes the probabilities p_k into the state $\sum_k p_k\,|k\rangle$, where $\{|k\rangle\}$ are all orthonormal. The second register is the initial state of the system $|\psi(0)\rangle$ and the remaining registers are all 'junk states' with the same dimension as $|\psi(0)\rangle$. They are called 'junk states' as they do not affect the outcome of the circuit and can be any state that is easy to prepare. The controlled-SWAP gates in the circuit work by applying the swap between $|\psi(0)\rangle$ and the state $|\phi_{k-1}\rangle$ when the first register is in the state $|k\rangle$. Here again the measurement and discard step at the end is equivalent to tracing out all other registers besides the register corresponding to the system's state. Lemma 2. will show that the circuit implements $\mathcal{E}_\tau$.

Lemma 2. The circuit in Figure 4 implements the qDRIFT channel $\mathcal{E}_\tau(|\psi(0)\rangle\langle\psi(0)|)$ for $\tau = t\lambda/N$.

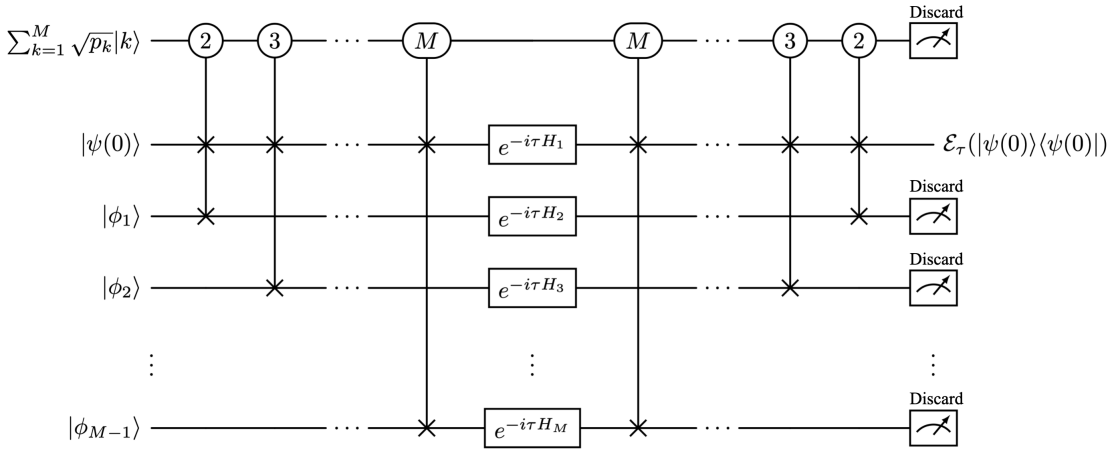

Figure 4: This quantum circuit implements a single step of the qDRIFT algorithm $\mathcal{E}_\tau$. The j-th controlled-SWAP gate is applied when the first register is in the state $|j\rangle$. The measurement and discard procedures correspond to tracing out the ancillary systems. The states $|\phi_l\rangle$ for $l = 1, ..., M-1$ can be in any easy to prepare state.

Proof. Once again, we make use of the density matrix representation to analyse this circuit. The initial density matrix can be written in the following way,

$$\sum_{k=1}^{M} p_k \left|k\rangle\langle k\right| \otimes \left|\psi(0)\rangle\langle\psi(0)\right| \otimes \left|\phi_1\rangle\langle\phi_1\right| \otimes ... \otimes \left|\phi_{M-1}\rangle\langle\phi_{M-1}\right| + \tag{52}$$

$$\sum_{k,k'=1}^{M} \sqrt{p_k}\sqrt{p_{k'}} \left|k\rangle\langle k'\right| \otimes \left|\psi(0)\rangle\langle\psi(0)\right| \otimes \left|\phi_1\rangle\langle\phi_1\right| \otimes ... \otimes \left|\phi_{M-1}\rangle\langle\phi_{M-1}\right|.$$

Since we trace out the first register at the end we observe that the second term of the sum will be zero, therefore for the rest of the proof we only apply operations to the first term in (52). For the next step we need to apply the controlled-SWAP operations. This gives the following density matrix,

$$\begin{aligned} &p_1 \left|1\rangle\langle 1\right| \otimes \left|\psi(0)\rangle\langle\psi(0)\right| \otimes \left|\phi_1\rangle\langle\phi_1\right| \otimes ... \otimes \left|\phi_{M-1}\rangle\langle\phi_{M-1}\right| + \\ &p_2 \left|2\rangle\langle 2\right| \otimes \left|\phi_1\rangle\langle\phi_1\right| \otimes \left|\psi(0)\rangle\langle\psi(0)\right| \otimes ... \otimes \left|\phi_{M-1}\rangle\langle\phi_{M-1}\right| + \\ &\vdots \qquad (53) \\ &+ p_M \left|M\rangle\langle M\right| \otimes \left|\phi_{M-1}\rangle\langle\phi_{M-1}\right| \otimes \left|\phi_1\rangle\langle\phi_1\right| \otimes ... \otimes \left|\psi(0)\rangle\langle\psi(0)\right|. \qquad (54) \end{aligned}$$

We can then apply in parallel all of the exponentials $\exp(-iH_k\tau)$ to the respective register. To make this easier, we define a map $\mathcal{V}_j(\rho) = e^{-iH_j\tau}\rho e^{iH_j}$ yielding,

$$\begin{aligned} &p_1 \left|1\rangle\langle 1\right| \otimes \mathcal{V}_1(\left|\psi(0)\rangle\langle\psi(0)\right|) \otimes \mathcal{V}_2(\left|\phi_1\rangle\langle\phi_1\right|) \otimes ... \otimes \mathcal{V}_M(\left|\phi_{M-1}\rangle\langle\phi_{M-1}\right|) + \\ &p_2 \left|2\rangle\langle 2\right| \otimes \mathcal{V}_1(\left|\phi_1\rangle\langle\phi_1\right|) \otimes \mathcal{V}_2(\left|\psi(0)\rangle\langle\psi(0)\right|) \otimes ... \otimes \mathcal{V}_M(\left|\phi_{M-1}\rangle\langle\phi_{M-1}\right|) + \\ &\vdots \qquad (55) \\ &+ p_M \left|M\rangle\langle M\right| \otimes \mathcal{V}_1(\left|\phi_{M-1}\rangle\langle\phi_{M-1}\right|) \otimes \mathcal{V}_2(\left|\phi_1\rangle\langle\phi_1\right|) \otimes ... \otimes \mathcal{V}_M(\left|\psi(0)\rangle\langle\psi(0)\right|). \qquad (56) \end{aligned}$$

Then we perform the controlled-SWAPs once more, which yields,

$$p_1 |1\rangle\langle 1| \otimes \mathcal{V}_1(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathcal{V}_2(|\phi_1\rangle\langle\phi_1|) \otimes ... \otimes \mathcal{V}_M(|\phi_{M-1}\rangle\langle\phi_{M-1}|)+$$
$$p_2 |2\rangle\langle 2| \otimes \mathcal{V}_2(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathcal{V}_1(|\phi_1\rangle\langle\phi_1|) \otimes ... \otimes \mathcal{V}_M(|\phi_{M-1}\rangle\langle\phi_{M-1}|)+$$
$$\vdots \tag{57}$$
$$+ p_M |M\rangle\langle M| \otimes \mathcal{V}_M(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathcal{V}_2(|\phi_1\rangle\langle\phi_1|) \otimes ... \otimes \mathcal{V}_1(|\phi_{M-1}\rangle\langle\phi_{M-1}|). \tag{58}$$

Now, we trace out all the ancillary registers by performing the partial trace,

$$p_1 \mathrm{tr}(|1\rangle\langle 1|) \otimes \mathcal{V}_1(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathrm{tr}(\mathcal{V}_2(|\phi_1\rangle\langle\phi_1|)) \otimes ... \otimes \mathrm{tr}(\mathcal{V}_M(|\phi_{M-1}\rangle\langle\phi_{M-1}|))+$$
$$p_2 \mathrm{tr}(|2\rangle\langle 2|) \otimes \mathcal{V}_2(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathrm{tr}(\mathcal{V}_1(|\phi_1\rangle\langle\phi_1|)) \otimes ... \otimes \mathrm{tr}(\mathcal{V}_M(|\phi_{M-1}\rangle\langle\phi_{M-1}|))+$$
$$\vdots \tag{59}$$
$$+ p_M \mathrm{tr}(|M\rangle\langle M|) \otimes \mathcal{V}_M(|\psi(0)\rangle\langle\psi(0)|) \otimes \mathrm{tr}(\mathcal{V}_2(|\phi_1\rangle\langle\phi_1|)) \otimes ... \otimes \mathrm{tr}(\mathcal{V}_1(|\phi_{M-1}\rangle\langle\phi_{M-1}|)). \tag{60}$$

We observe that $\mathcal{V}_j$ is a unitary transformation and therefore preserves the trace of $|\psi(0)\rangle\langle\psi(0)|$, which implies that $\mathrm{tr}(\mathcal{V}_j(|\phi_l\rangle\langle\phi_l|) = 1$ for all j and l. We also observe that $\mathrm{tr}(|k\rangle\langle k|) = \langle k|k\rangle = 1$. Hence, the final density matrix is,

$$\sum_{j=1}^{M} p_j \mathcal{V}_j(|\psi(0)\rangle\langle\psi(0)|) = \sum_{j=1}^{M} p_j e^{-iH_j\tau} |\psi(0)\rangle\langle\psi(0)| e^{iH_j\tau} = \mathcal{E}_\tau(|\psi(0)\rangle\langle\psi(0)|). \tag{61}$$

Therefore, the circuit completes the proof and implements $\mathcal{E}_\tau$.

To be able to implement $\mathcal{E}_\tau^N$, we need to repeat the circuit in Figure 4 N times, but after each time we measure and discard, we need to reset each ancillary register to their initial state. To obtain the gate complexity for this circuit, we need to count the sequential operations in the circuit. We observe that there are $2(M-1)$ controlled-SWAP operations and 1 unitary gate layer, which applies in parallel the unitary $\exp(-iH_1\tau) \otimes ... \otimes \exp(-iH_M\tau)$. So, the total number of gate layers is $2M-1$. Since we repeat this circuit N times, the total number of sequential gate layers is $2MN - N$. Then if the gate complexity is denoted by g_{QF} we see that,

$$g_{\mathrm{QF}} = O(MN). \tag{62}$$

Here we see that the gate complexity of this circuit has the same scaling as the LCU approach, however it circumvents the need to implement controlled-$\exp(-iH_j\tau)$ gates by applying all the exponentials in parallel. This is possible as the exponentials $\exp(-iH_j\tau)$ are, by design, easy to implement on a quantum computer. Although the gate complexities for both circuits have the same scaling in practice, the QF circuit may be more costly to implement due to the number of controlled-SWAPS needed, however, the simplicity of not needing to implement controlled-$\exp(-iH_j\tau)$ gates and the parallel application of exponentials makes this additional cost worthwhile.

5 Comparison Of Gate Complexities

Now that we have presented the qDRIFT protocol and the circuit implementations for it, we can compare the gate complexities with TS product formulas.

Table 1 summarises the gate complexities for the TS product formulas for first, second, fourth and $2k$-th orders as well as the gate complexities for the qDRIFT circuit implementation and the hybrid qDRIFT algorithm in [14], with classical sampling. We observe that for the circuit implementations of the qDRIFT channel both the LCU approach and QF approach scale as $O(MN)$. Using the bound on N in Theorem 1. we have that the gate complexity for the qDRIFT circuit implementation is,

$$O(MN) = O\left(\frac{M\lambda^2 t^2}{\epsilon}\right). \tag{63}$$

The circuit implementation of qDRIFT contains a linear factor of M in the complexity. In comparison the the hybrid qDRIFT algorithm in [14], that uses classical sampling, the gate complexity is just $O(\lambda^2 t^2/\epsilon)$. Although the gate complexity in this case does not depend explicitly on M, the full

Table 1: This table shows the gate complexities for various orders of TS formulas, the qDRIFT circuit implementations, and the hybrid qDRIFT algorithm.

Algorithm	**Gate Complexity**
First order Trotter-Suzuki	$O\left(M^3(\Lambda t)^2/\epsilon\right)$
Second order Trotter-Suzuki	$O\left(M^{5/2}(\Lambda t)^{3/2}/\epsilon^{1/2}\right)$
Fourth Order Trotter-Suzuki	$O\left(M^{9/4}(\Lambda t)^{5/4}/\epsilon^{1/4}\right)$
Higher Order Trotter-Suzuki	$O\left(M^{2+\frac{1}{2k}}(\Lambda t)^{1+\frac{1}{2k}}/\epsilon^{1/2k}\right)$
qDRIFT Circuit Implementation	$O(M\lambda^2 t^2/\epsilon)$
qDRIFT with Classical Sampling	$O(\lambda^2 t^2/\epsilon)$

algorithm in [14], will also scale as $O(MN)$ as the classical sampling procedure used will scale as $O(M)$ [18]. However, for most practical purposes, this scaling is negligible, so we shall not consider the computational cost of sampling when talking about the hybrid qDRIFT algorithm.
Although the qDRIFT algorithms scale better in terms of their dependence on M, they do scale quadratically in time, which means that for longer times, we expect the higher-order TS product formulas to outperform the qDRIFT circuit implementation. To see this we have used the parameters from the molecular simulations in [14], to illustrate the performance of the qDRIFT circuit implementation. In Figure 5, we plot the log of the gate counts for first-order TS, second-order TS, fourth-order TS, qDRIFT circuit and qDRIFT hybrid for two sets of parameters taken from [14]. We see in Figure 5 (a) that both the qDRIFT circuit and qDRIFT hybrid have a better gate complexity than first and second-order TS. However, for fourth-order TS, we see that for $t > 1244$ s, the gate complexity is better than the qDRIFT circuit because of the quadratic scaling in time. Therefore, the qDRIFT circuit is better for shorter times in this regard. In Figure 5 (b) we observe a similar behaviour for the first and second order TS in comparison to both qDRIFT algorithms but for the fourth order TS it outperforms qDRIFT circuit for $t > 32$ as λ is much larger in this case and its quadratic scaling gives us a poor gate complexity. Therefore we see that the qDRIFT circuit implementations perform a lot better for shorter times.
To see for what values of M the qDRIFT circuit will perform better than TS, we plot the gate complexities of all the simulation methods and fix t while varying M; figure 6 shows two plots for two sets of parameters with varying number of terms M. In Figure 6 (a), we see that for $M > 363673$, the qDRIFT circuit has better gate complexity than the fourth-order TS. We also observe that both qDRIFT implementations scale better than first and second-order TS. In Figure 6 (b) we observe similar behaviour for the gate complexities, however the qDRIFT circuit will only perform better than fourth order TS when $M > 848485$ this is because $t = 1000$ s and we know the quadratic scaling in t gives poor gate complexities for qDRIFT circuit. It is obvious that for fixed t and varying M hybrid qDRIFT will not change as it does not depend on M. Overall, we observe that the qDRIFT hybrid is still the best-performing in terms of gate complexity. However, if we need to implement qDRIFT directly on a quantum computer without a classical sub-routine, then it will work best for shorter times and a large number of terms.

6 Conclusion

In this work, we construct two efficient quantum circuit implementations for the qDRIFT algorithm. One implementation used the LCU [9] approach to construct the circuit, while the other used the QF procedure [16]. Both of these circuits had a gate complexity of $O(MN)$, which scales better than both TS [5] and randomised TS [10] in the number of terms of the Hamiltonian. While the LCU approach requires no controlled-SWAP gates, it does require controlled-$\exp(-iH_j\tau)$ gates, which may be costly to implement on the quantum computer. The QF approach, while needing many controlled-SWAP gates,

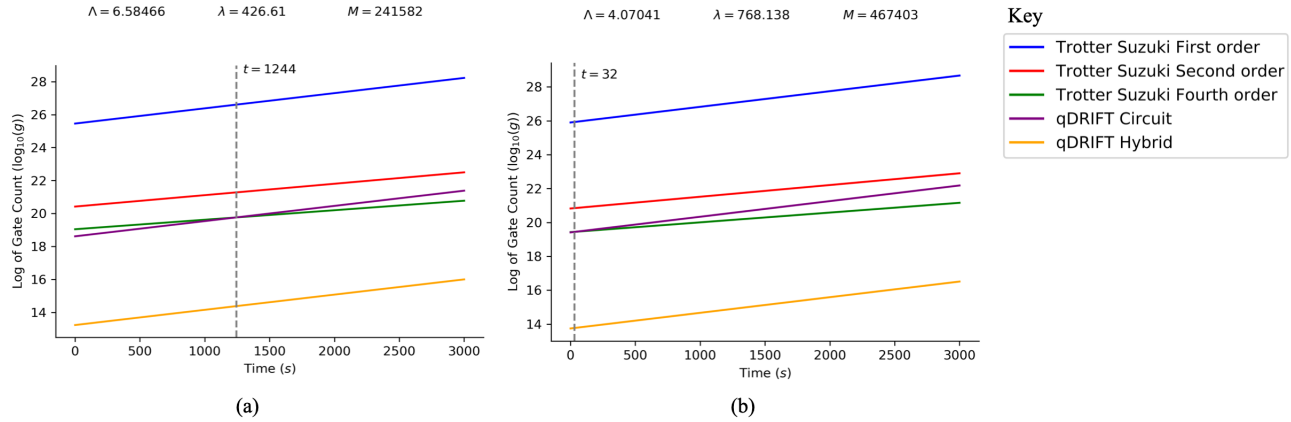

Figure 5: Here we plot the log of the gate complexities with varying time for various orders of TS formulas, the qDRIFT circuit implementations and the hybrid qDRIFT algorithm and various values of Λ, λ and M. (a) We see that the qDRIFT hybrid outperforms all other algorithms, and the qDRIFT circuit outperforms first and second-order TS with the fourth-order performing better only for a long time. (b) the qDRIFT circuit does not outperform fourth-order TS due to the quadratic scaling on time and λ.

only has to implement $\exp(-iH_j\tau)$, which we can do easily on a quantum computer by design. Therefore, the QF implementation may be a better choice if we can implement $\exp(-iH_j\tau)$ easily on a quantum computer. However, if controlled-SWAP gates are more costly to implement on the quantum computer, then the LCU approach will be a better choice. As an open problem one can consider deriving bounds on the gate complexities for our circuit implementations that take into account the decomposition of the exponentials $\exp(-iH_j\tau)$ into single qubit and two qubit gates using the Solovay-Kiteav theorem [3]. This can provide better insight into which circuit one should use based on the type of elementary gates one can implement on a quantum computer.

Acknowledgements
The authors thank Ms S. M. Pillay for proof-reading this manuscript. This work is based upon research supported by the National Research Foundation of the Republic of South Africa. Support from the NICIS (National Integrated Cyber Infrastructure System) e-research grant QICSA is kindly acknowledged.

Conflict Of Interest
Francesco Petruccione the Chair of the Scientific Board and Co-Founder of QUNOVA computing. The authors declare no other competing interests.

A Information about the Diamond Norm for Quantum Channels

In this section we will introduce the diamond norm and its properties. In the main text we used the operator norm, denoted by $\|\cdot\|$, to analyse the TS product formulas. However, when we analyse the qDRIFT protocol we require a way to compute the error in the approximation of a quantum channel. This means that we need a norm that can be used on the space of channels (or superoperators). To do this we use the diamond norm [17] and it is defined for a superoperator T as follows,

$$\|T\|_\diamond := \sup_{\rho;\|\rho\|_1=1} \|(T\otimes\mathbb{1})(\rho)\|_1, \tag{64}$$

where ρ is any density matrix, $\mathbb{1}$ is identity map that acts on the same size space as T and $\|\cdot\|_1$ is the trace norm of an operator and is defined as,

$$\|A\|_1 := \mathrm{tr}(\sqrt{A^\dagger A}). \tag{65}$$

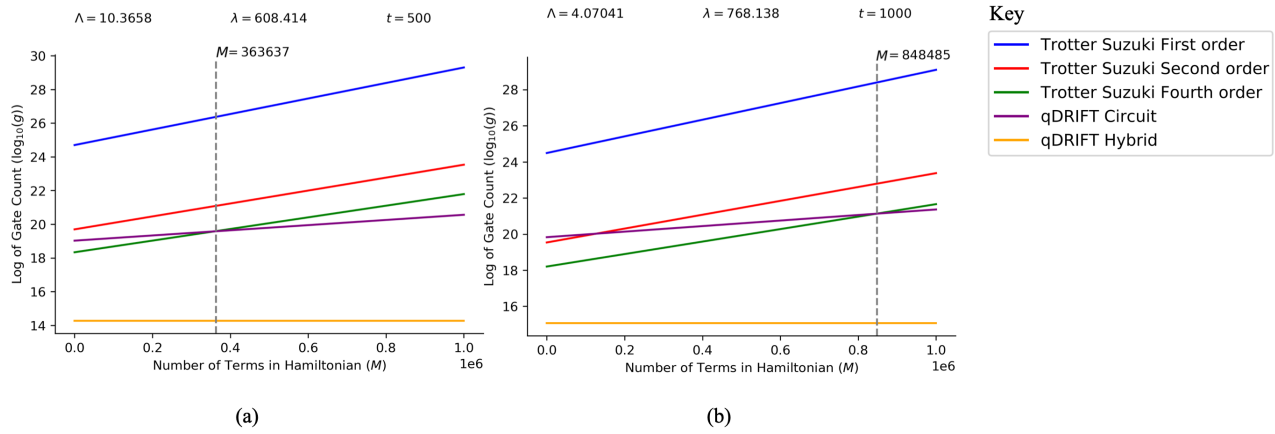

Figure 6: Here we plot the log of the gate complexities with a varying number of terms M, for various orders of TS formulas, the qDRIFT circuit implementations and the hybrid qDRIFT algorithm and various values of Λ, λ and t. (a) We see here that for $M > 363637$, the qDRIFT circuit outperforms fourth-order TS, and for all values of M, it outperforms first and second-order TS. (b) Due to the quadratic scaling on time and $t = 1000$, the qDRIFT circuit only outperforms fourth order TS when $M > 848485$. In both plots, we see that hybrid qDRIFT does not change as it does not depend explicitly on M.

The trace norm of an operator is also a sum of all the singular values. The diamond norm is a good choice of norm because it takes into account entanglement with a reference system [**?**], where as other superoperator norms such as $1 \to 1$ Schatten norms do not. Given two superoperators T and V, the two properties of the diamond norm we use are:

1. The triangle inequality (sub-additivity): $\|T + V\|_\diamond \leq \|T\|_\diamond + \|V\|_\diamond$.
2. Sub-multiplicativity: $\|TV\|_\diamond \leq \|T\|_\diamond \|V\|_\diamond$ and consequently $\|T^n\|_\diamond \leq \|T\|_\diamond^n$ for $n \in \mathbb{N}$.

THe following Lemma will outline a very important property of the diamond norm, it will allow us to break up the evolutions into small steps and bound the difference by the number steps times the norm of the difference of the evolutions for a small time step,

Lemma 3. Given two quantum channels T and V and some $n \in \mathbb{N}$ we have,

$$\|T^n - V^n\|_\diamond \leq n\|T - V\|_\diamond. \tag{66}$$

One should note that if the superoperator T is a quantum channel then $\|T\|_\diamond = 1$. From the definition of the diamond norm it follows that if we have two quantum channels T and V that are applied to a density matrix ρ then,

$$\|T(\rho) - V(\rho)\|_1 \leq \|T - V\|_\diamond. \tag{67}$$

This inequality is very important as it allows us to bound the distance between two quantum states i.e. density matrices, by their respective channels.

References

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys*, vol. 21, no. 6/7, 1982.

[2] Y. Manin, "Computable and uncomputable," *Sovetskoye Radio, Moscow*, vol. 128, 1980.

[3] C. M. Dawson and M. A. Nielsen, "The solovay-kitaev algorithm," *arXiv preprint quant-ph/0505030*, 2005.

[4] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996.

[5] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient quantum algorithms for simulating sparse hamiltonians," *Communications in Mathematical Physics*, vol. 270, pp. 359–371, 2007.

[6] M. Suzuki, "Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations," *Physics Letters A*, vol. 146, no. 6, pp. 319–323, 1990.

[7] Suzuki, "General theory of fractal path integrals with applications to many-body theories and statistical physics," *Journal of Mathematical Physics*, vol. 32, no. 2, pp. 400–407, 1991.

[8] G. H. Low and I. L. Chuang, "Hamiltonian simulation by qubitization," *Quantum*, vol. 3, p. 163, 2019.

[9] A. M. Childs and N. Wiebe, "Hamiltonian simulation using linear combinations of unitary operations," *Quantum Information & Computation*, vol. 12, no. 11-12, pp. 901–924, 2012.

[10] A. M. Childs, A. Ostrander, and Y. Su, "Faster quantum simulation by randomization," *Quantum*, vol. 3, p. 182, 2019.

[11] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, "Theory of trotter error with commutator scaling," *Physical Review X*, vol. 11, no. 1, p. 011020, 2021.

[12] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Simulating hamiltonian dynamics with a truncated taylor series," *Physical Review Letters*, vol. 114, no. 9, p. 090502, 2015.

[13] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, "Toward the first quantum simulation with quantum speedup," *Proceedings of the National Academy of Sciences*, vol. 115, no. 38, pp. 9456–9461, 2018.

[14] E. Campbell, "Random compiler for fast hamiltonian simulation," *Physical Review Letters*, vol. 123, no. 7, p. 070503, 2019.

[15] S. Schmidt and G. Blatter, "Strong coupling theory for the Jaynes-Cummings-Hubbard model," *Physical Review Letters*, vol. 103, no. 8, p. 086403, 2009.

[16] D. K. Park, I. Sinayskiy, M. Fingerhuth, F. Petruccione, and J.-K. K. Rhee, "Parallel quantum trajectories via forking for sampling without redundancy," *New Journal of Physics*, vol. 21, no. 8, p. 083024, 2019.

[17] J. Watrous, "Semidefinite programs for completely bounded norms," *arXiv preprint arXiv:0901.4709*, 2009.

[18] A. J. Walker, "An efficient method for generating discrete random variables with general distributions," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 253–256, 1977.