Noise-aware Quantum Circuit Simulation With Decision Diagrams

Thomas Grurl, *Student Member, IEEE,* Jürgen Fuß, and Robert Wille, *Senior Member, IEEE*

***Abstract*—Since quantum computers can solve important problems faster than classical computers, many resources have gone into the development of this technology in recent decades. Despite the tremendous progress that has already been made towards the development of quantum computers, they are still an emerging technology, which restricts access and reliability. Thus, research on quantum algorithms still heavily relies on quantum circuit simulators that run on classical hardware. However, simulating the execution of a quantum computer on conventional hardware is exponentially difficult, which is also the reason why quantum computing is an interesting technology in the first place. Particularly complex is noise-aware simulation of quantum computers, i.e., the consideration of noise effects that are common in today's quantum hardware during quantum circuit simulation. In this work, we investigate the use of decision diagrams for this task. To this end, we present two distinct approaches for noise-aware quantum circuit simulation, investigate how they can be realized using decision diagrams, and implement decision diagram-based solutions for each of the presented noise-aware simulation schemes. In an extensive evaluation, we unveil potential for further improvements and also demonstrate substantial speed-ups compared to the current state of the art.**

I. Introduction

Quantum computers promise to solve specific tasks that are intractable for conventional computers. They mainly achieve this by exploiting the quantum mechanical effects of *superposition*, i.e., a quantum bit can be in a combination of different states at the same time, and *entanglement*, which is that quantum bits can be connected with one another. Early examples of quantum algorithms that demonstrate the potential of this technology are Shor's algorithm, with which integers can be efficiently factored [1], and Grover's database search algorithm [2]. More recently, other algorithms have been found in the areas of machine learning [3] and chemistry [4], among others. Matching the success in the development of new quantum algorithms, there have also been formidable achievements towards the development of quantum computers, which are driven by big players such as Google, IBM, Intel, Rigetti, and Alibaba.

Despite this, quantum computers are still an emerging technology with restricted reliability and accessibility. Therefore, a substantial amount of research on quantum computing still depends on so-called quantum circuit simulators running on classical hardware. As their name suggests, they simulate the execution of a quantum circuit and, by this, allow for the development and evaluation of quantum circuits without access to actual quantum hardware. Moreover, quantum circuit simulators allow for more insights into the considered quantum applications since their results do not only provide probabilistic measurements but full access to the resulting quantum state.

However, the task of quantum circuit simulation is exponentially hard. This might also be one of the reasons why many quantum circuit simulators (such as [5]–[9]) mimic only *perfect* quantum computers. But, due to the extreme fragility of quantum mechanical effects, today's quantum computers are prone to noise effects (i.e., errors) that obscure their calculations [10]. While error mitigation is constantly improving, errors generated by noise effects are still an overshadowing aspect of quantum computing. Hence, considering those noise effects during quantum circuit simulation is essential to gain important insights into how algorithms behave when executed on real quantum hardware. These insights can be used for developing noise resistant algorithms or the development/evaluation of quantum error correction schemes.

Fortunately, noise effects in quantum computers are well studied and mathematical models for approximating them are available. However, employing those models during quantum circuit simulation makes the hard problem of quantum circuit simulation even harder, limiting corresponding approaches (see, e.g., [11]–[20]). More precisely, the main challenge of quantum circuit simulation is already caused by the exponential nature of the vectors and matrices that are required to describe quantum states and operations. Additionally, considering error effects caused by noise requires extended descriptions and/or methods (covered in more detail later in this paper) that add substantial further complexity. How to efficiently cope with all this complexity is an important research problem.

An interesting approach to address this problem is based on the use of *decision diagrams* (as introduced, e.g., in [6]). They offer a compact data structure for representing quantum states and operations that is often below the exponential size of other solutions. In fact, recent investigations and case studies have shown their potential of being a viable data structure that is promising in handling the complexity of quantum circuit simulation [6]–[8], [21]–[27]. However, thus far their potential for noise-aware quantum circuit simulation is largely unexplored. Hence, it remains unknown whether the compact representation provided by current state-of-the-art quantum decision diagrams also can be utilized when noise is considered during the simulation.

In this work, we address this question[1]. To this end, we present two schemes for considering noise during quantum circuit simulation: a deterministic and a stochastic scheme. For both, we investigate how they can be realized using decision diagrams and what optimizations are possible. Based on that, we implemented decision diagram-based solutions for each of the presented noise-aware simulation schemes in C++ [2] and subjected them to intensive evaluations (comparing them against each other as well as state-of-the-art simulators by IBM and Atos).

The results show that, depending on the scheme used as well as the considered errors, noise-aware quantum circuit simulation

[1] Preliminary versions of this work have been published in [28], [29].

[2] Both implementations are available as open-source at https://github.com/cda-tum/ddsim.

with decision diagrams provides tremendous improvements compared to the state of the art; but also unveils some shortcomings of this approach. More precisely, our investigations show that decision diagrams as used thus far severely struggle with a deterministic consideration of noise. In contrast, when applied to the stochastic scheme, they enable speed-ups in the simulation time of several magnitudes and allow for simulating circuits with more than twice the number of qubits compared to the state of the art.

In the remainder of this paper, these contributions and findings are presented as follows: In Section II, the concepts of quantum computing and noise in quantum computing are reviewed. Section III discusses quantum circuit simulation for both, straightforward and decision diagram-based approaches, and introduces the research question we are considering in this work. In Section IV and Section V, deterministic and stochastic noise-aware quantum circuit simulation are presented, respectively, and it is investigated how those concepts can be implemented in a decision diagram-based simulator. Finally, we present the results of our extensive evaluations in Section VI and conclude the paper in Section VII.

II. Background

In order to keep this work self-contained, this section reviews the basic concepts of quantum computing as well as noise effects. We refer the reader to standard textbooks, e.g., [30], [31], for a more thorough introduction.

A. Quantum Computing

In the classical world the basic unit of information is a bit, which can either assume the state 0 or 1. In the quantum world, the smallest unit of information is called a *quantum bit* or *qubit*. Like a classical bit, a qubit can assume the states 0 and 1, which are called *basis states* and—using Dirac notation—are written as $|0\rangle$ and $|1\rangle$. Additionally, a qubit can also assume a combination of the two basis states, which is then called a *superposition*. More precisely, the state of the qubit $|\psi\rangle$ is written as $|\psi\rangle = \alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle$ with $\alpha_0, \alpha_1 \in \mathbb{C}$ such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The values α_0, α_1 are called *amplitudes* and describe how strongly the qubit is related to each of the basis states. Measuring a qubit yields $|0\rangle$ ($|1\rangle$) with probability $|\alpha_0|^2$ ($|\alpha_1|^2$). By measuring the qubit, any existing superposition is destroyed and the state of the qubit collapses to the measured basis state.

Quantum states containing more than one qubit are often called *quantum registers* and the concepts above can be extended to describe such systems as well. An n qubit register can assume $N = 2^n$ basis states and is described by N amplitudes $\alpha_0, \alpha_1, \ldots \alpha_{N-1}$, which must satisfy the normalization constraint $\sum_{i\in\{0,1\}^n} |\alpha_i|^2 = 1$. Usually quantum states are shortened to state vectors containing only the amplitudes, e.g., $\begin{bmatrix}\alpha_{00} & \alpha_{01} & \alpha_{10} & \alpha_{11}\end{bmatrix}^\top$ for $n = 2$ qubits.

Example 1. *Consider the 2-qubit quantum register*

$$|\psi\rangle = 0 \cdot |00\rangle + 1 \cdot |01\rangle + 0 \cdot |10\rangle + 0 \cdot |11\rangle\,,$$

which is represented by the state vector $\begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^\top$. *This is a valid state, since it satisfies the normalization constraint* $|0|^2 + |1|^2 + |0|^2 + |0|^2 = 1$. *Measuring the system yields* $|01\rangle$, *with probability 1.*

Quantum states can be manipulated using quantum operations. With the exception of the measurement operation, all quantum operations are inherently reversible and represented by unitary matrices, i.e., square matrices whose inverse is their conjugate transpose. Important 1-qubit operations are

$$H = {}^{1}\!/\!\sqrt{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}, X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix},$$
$$Y = iXZ = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}, \text{and } I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},$$

with the H gate transforming a basis state into a superposition, X being the quantum equivalent of the NOT operation, the Z gate flipping the phase of a qubit. The identity "operation" I leaves a state unchanged and is relevant in the context of simulating noise. There are also 2-qubit operations. An important example is the controlled-X (also known as CNOT) operation, which negates the state of a qubit, if the chosen control qubit is $|1\rangle$. Applying an operation to a state can be done by matrix-vector multiplication. For this, the size of both the matrix and the vector have to match. Since quantum states often consist of more than one qubit, this often requires embedding the operation, which is done using the Kronecker product with the I operator.

Example 2. *Consider again the 2-qubit register* $|\psi\rangle$ *from Example 1. In order to apply an* H *operation to the first qubit, the operation first has to be enlarged using the Kronecker product*

$$\underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}}_{H} \otimes \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{I} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}}_{H\otimes I}.$$

Applying this operation to $|\psi\rangle$ *results in*

$$\underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}}_{H\otimes I} \cdot \underbrace{\begin{bmatrix}0\\ 1\\ 0\\ 0\end{bmatrix}}_{|\psi\rangle} = \underbrace{\begin{bmatrix}0\\ \frac{1}{\sqrt{2}}\\ 0\\ \frac{1}{\sqrt{2}}\end{bmatrix}}_{|\psi'\rangle}.$$

Next we are applying a CNOT operation to $|\psi'\rangle$, *which negates the amplitude of the second qubit if the first qubit is set to* $|1\rangle$. *This is given by*

$$\underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}}_{\text{CNOT}} \cdot \underbrace{\begin{bmatrix}0\\ \frac{1}{\sqrt{2}}\\ 0\\ \frac{1}{\sqrt{2}}\end{bmatrix}}_{|\psi'\rangle} = \underbrace{\begin{bmatrix}0\\ \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\\ 0\end{bmatrix}}_{|\psi^\star\rangle}.$$

Measuring $|\psi^\star\rangle$ *either yields* $|01\rangle$ *or* $|10\rangle$, *each with probability* ${}^{1}\!/\!{}_{2}$. *Note that the measurement outcome of the two qubits are strongly correlated—an essential concept in quantum computing known as* entanglement.

B. Noise in Quantum Computing

The formalism presented above can be used to describe how perfect quantum computers behave. However, due to the fragility of quantum mechanical effects, real quantum computers are prone to noise, which causes errors during computations. Such error effects can be viewed as (unwanted) operations on the state. However, while ideal quantum operations are deterministic, errors can have an additional degree of randomness. For instance, whenever a specific qubit is used, an error may occur with probability p while everything works perfectly fine with probability $1 - p$. The outcome of an erroneous quantum computation cannot be described by a single state vector anymore. Instead, it is described by a *mixture* (or ensemble) $\{(p_i, |\psi_i\rangle)\}$ of possible outcomes. Here, the states $|\psi_i\rangle$ label potential outcomes while each weight p_i describes the probability with which outcome $|\psi_i\rangle$ occurs ($p_i \geq 0$ and $\sum_i p_i = 1$).

These noise effects can be classified into two categories [32]:

- *Gate errors* (also known as operational errors)
- *Decoherence errors* (also known as retention errors)

Gate Errors are introduced when operations are executed [32]. They occur since quantum computers are mechanical apparatuses that do not always apply operations perfectly. Instead, an operation may be not executed at all, or in a (slightly) modified fashion. Since gate errors are highly specific for each quantum computer and even vary for qubits within the quantum computer, they are often approximated using depolarization errors [12], [14]. A depolarization error describes that a qubit is set to a completely random state [30]. For publicly available quantum computers from IBM, the error probabilities are on the order of 10^{-3} to 10^{-2} [33].

Example 3. *Consider again the 2-qubit state* $|\psi^\star\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ *from Example 2. Suppose that this state might be affected by a gate error in the first qubit only, depolarizing it. With probability* $1 - p$, *nothing happens and the state remains unchanged. With probability* p, *the first qubit becomes depolarized. We can capture this effect by either applying I, X, Y, or Z—each with probability* $\frac{p}{4}$. *This produces a mixture* $\{(1 - p, \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)), (\frac{p}{4}, \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)), (\frac{p}{4}, (\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)), (\frac{p}{4}, (\frac{i}{\sqrt{2}}(|11\rangle - |00\rangle)), (\frac{p}{4}, \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle))\}$ *which cannot be represented by a single 2-qubit state.*

Decoherence Errors occur due to the fragile nature of quantum systems (qubits). In practice, this leads to the problem that they can hold information for a limited time only. There are two types of decoherence errors that may appear [32]:

- A qubit in a high-energy state ($|1\rangle$) tends to relax into a low energy state ($|0\rangle$). That is, after a certain amount of time, qubits in a quantum system eventually decay to $|0\rangle$. This error is called *amplitude damping error* or *T1 error*.
- In addition to that, when a qubit interacts with the environment, a phase flip effect might occur. This leads to *phase flip error* or *T2 error*.

Developments in the physical realization of quantum computers (e.g., in [34], [35]) show significant improvements in the coherence times of qubits—improving the "lifetime" of qubits before decaying to $|0\rangle$ and reducing the frequency of phase flip errors, respectively. Nevertheless, these errors are still a significant aspect of all quantum computers and, hence, should also be considered during simulation.

Example 4. *Once again, suppose that the 2-qubit state* $|\psi^\star\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ *from Example 2 is affected by an error. More precisely, an amplitude damping error affects the first qubit with probability* p. *This error is described by the matrices* $\mathrm{E}_0 = \left[\begin{smallmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{smallmatrix}\right]$ *and* $\mathrm{E}_1 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{smallmatrix}\right]$ *[30]. Applying* E_0 *and* E_1 *to* $|\psi^\star\rangle$ *mimics the effect of amplitude damping with probability* p *and results in the state mixture* $\{(\frac{p}{2}, |00\rangle), (1 - \frac{p}{2}, \frac{1}{\sqrt{2-p}}\,|01\rangle + \frac{\sqrt{1-p}}{\sqrt{2-p}}\,|10\rangle)\}$.

III. Motivation

This section first reviews, how *perfect* quantum circuit simulation is conducted, i.e., the simulation of a (theoretical) quantum computer that is not subject to any noise effects. This simulation is conceptually simple but quickly requires huge computational resources due to the exponentially large representation of quantum states and operations. In this work, we consider tackling this complexity using decision diagrams whose principles and utilization are reviewed next. All this, however, does not take noise effects into consideration. Hence, finally, the open question of noise aware quantum circuit simulation is discussed and two directions that address this and which are investigated in this work are outlined—providing the motivation of this work.

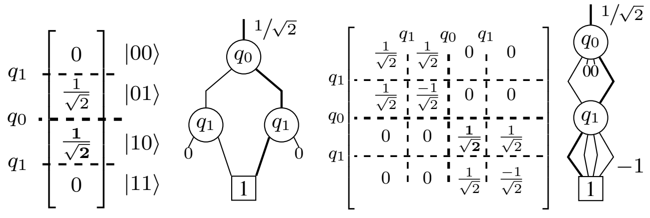
A. Quantum Circuit Simulation

The basic concepts reviewed in Section II are sufficient to simulate the execution of *perfect* quantum computers. More precisely, having states and operations represented by 1-dimensional and 2-dimensional arrays, respectively, the application of operations is simulated by matrix-vector multiplications, as shown in Example 2. Since matrix-vector multiplications can be decomposed into a series of smaller operations, this simulation approach comes with a huge potential for parallelization. More precisely, every multiplication of a matrix M and a vector V can be split into multiple multiplications and additions, i.e.,

$$\begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \cdot \begin{bmatrix} V_0 \\ V_1 \end{bmatrix} = \begin{bmatrix} M_{00} \cdot V_0 + M_{01} \cdot V_1 \\ M_{10} \cdot V_0 + M_{11} \cdot V_1 \end{bmatrix}.$$

This decomposition scheme can be repeated recursively, resulting in a large number of intermediate operations that can be performed independently of one another with little synchronization overhead. This—conceptually simple—concept is the basis for many state-of-the-art quantum circuit simulators (such as, e.g., [9], [11]–[18], [20]).

However, a crucial bottleneck of such a straightforward approach to quantum circuit simulation is the exponential growth of the involved vectors and matrices relative to the number of involved qubits (this is often referred to as the curse of dimensionality). This makes, e.g., the representation of quantum states consisting of more than a few qubits extremely challenging. One might think that considering qubits locally avoids this exponential blow-up, i.e., instead of considering all qubits within a single state at once, we could simply regard them independently. This is often possible when representing quantum operations (in particular for elementary operations that use only few qubits). In these cases, it is not necessary to construct the full operation matrix of size $2^n \times 2^n$ using the Kronecker product (as illustrated in Example 2). However, this does not extend to the representation of quantum states. Due to *entanglement*—an essential concept of quantum computing—individual qubits may affect each other, making it impossible



(a) Representation of $|\psi^{\star}\rangle$

(b) Representation of $H \otimes I$

Fig. 1. Decision diagram representation of a quantum state and a operation

to represent them individually. Accordingly, the matrices representing quantum operations applied to (a larger set of) entangled qubits are subject to exponential growth.

Example 5. *Consider again the quantum state $|\psi^{\star}\rangle$ from Example 2. Measuring the first qubit q_0 collapses it to either $|0\rangle$ or $|1\rangle$ both with 50 % probability. However, since the qubits of $|\psi^{\star}\rangle$ are entangled with one another, this measurement also affects q_1. That is, when, e.g., the measurement of q_0 yields $|0\rangle$, q_1 collapses to the basis state $|1\rangle$ (although not explicitly measured); resulting in $\begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^{\top}$. This shows that, due to entanglement, individual qubits cannot be represented independently of each other.*

Overall, although conceptually simple, this quickly renders quantum circuit simulation significantly complex (each additional qubit to be considered effectively doubles the memory required to represent the vector). Researchers are currently heavily investigating how this exponential complexity can be tackled—leading to different simulation schemes (such as *Feynman-based simulation* [36]), the utilization of approximation (see, e.g., [37]), or alternative data-structures such as *Tensor Networks* [38]–[40], *Matrix Product States* [38], and *Decision Diagrams* [6]–[8], [22]–[27]. In the remainder of this work, we are focusing on approaches based on decision diagrams.

B. Quantum Circuit Simulation With Decision Diagrams

In order to tackle the exponential complexity of quantum circuit simulation, approaches based on decision diagrams (see e.g., [6]–[8], [22]–[27]) have been proven promising. Here, the general idea is to identify data redundancies in the representation of states/operations and to represent them through shared sub-structures. This can result in a very compact representation, which makes it possible to simulate quantum applications that cannot be simulated using a straightforward approach.

More precisely, representing, e.g., a state vector as a decision diagram revolves around recursively splitting the vector into equal-sized sub-vectors, until the sub-vectors only contain a single element. To illustrate this, consider a quantum register $q_0, q_1, \ldots, q_{n-1}$ composed of n qubits, with q_0 representing the most significant qubit. Then, the first 2^{n-1} entries of the corresponding state vector represent amplitudes for basis states where q_0 is $|0\rangle$ and the other entries represent amplitudes where q_0 is $|1\rangle$. In a decision diagram, this is represented by a node labeled q_0 with two successors, where the left (right) successor points to a node (labeled q_1) that represents the sub-vector with amplitudes for basis states with q_0 assigned $|0\rangle$ ($|1\rangle$).

This scheme is repeated recursively until sub-vectors of size 1 (i.e. complex numbers) result. During this procedure, equal sub-vectors are represented by the same node, which reduces the overall size of the decision diagram. Additionally, instead of having explicit terminal nodes for all amplitudes, edge weights are used to store common factors of the amplitudes, resulting in even more compaction. In order to reconstruct the amplitude of a specific state, the edge weights along the corresponding path are multiplied. Furthermore, to aid the readability of the decision diagram, edge weights of 1 are omitted and nodes with an incoming edge weight of 0 are represented as 0-stubs—indicating that amplitudes of all possible states represented by this part of the decision diagram are 0.

Example 6. *In Fig. 1a, the quantum state $|\psi^{\star}\rangle$ from Example 2 is depicted in both, the vector and decision diagram representation. The annotations of the state vector indicate how it is decomposed into the corresponding decision diagram. Reconstructing specific amplitudes from the decision diagram is done by multiplying the edge weights of the corresponding path. For example, reconstructing the amplitude of the state $|10\rangle$ (bold lines in the figure), is done by multiplying the edge weight of the root edge ($1/\sqrt{2}$) with the left edge of q_0 (1) as well as the right edge of q_1 (1), i.e. $1/\sqrt{2} \cdot 1 \cdot 1 = 1/\sqrt{2}$.*

In a similar fashion, quantum operations can be represented by decision diagrams. However, due to their square nature, they are split into four equally sized sub-parts. This is represented in a decision diagram by a node with four successors: the first one representing the sub-matrix in the upper left corner, the second one representing the sub-matrix in the upper right corner, the third one representing the sub-matrix in the lower left corner, and the fourth one representing the sub-matrix in the lower right corner. Other than that, the decomposition process is analogous to the one for vectors.

Example 7. *In Fig. 1b, the decision diagram representation of the operation $H \otimes I$ from Example 2 is provided. Once again, the annotations of the matrix representation indicate how the matrix is decomposed into the resulting decision diagram. Also, reconstructing specific matrix elements from the decision diagram representation is done by multiplying the edge weights along the corresponding path, e.g., the highlighted matrix element can be reconstructed by multiplying the edge weights of the root edge $\frac{1}{\sqrt{2}}$ with the fourth edge of q_0 (1) and the first edge of q_1 (1), i.e., $\frac{1}{\sqrt{2}} \cdot 1 \cdot 1 = \frac{1}{\sqrt{2}}$.*

Having a representation of quantum states and operations through decision diagrams, simulation is conducted by multiplying operations onto states. Yet, because of the different representation, multiplications must be decomposed with respect to the most significant qubit. Consider a quantum register $|v\rangle = q_0, q_1, \ldots, q_{n-1}$ of n qubits, where q_0 represents the most significant qubit and a unitary quantum operation M of size $2^n \times 2^n$. Multiplying the operation M onto the state $|v\rangle$, is done by splitting $|v\rangle$ into two and M into four equally sized parts, leading to two sub-vectors of size 2^{n-1} and four sub-matrices of size $2^{n-1} \times 2^{n-1}$. The process is repeated recursively until matrices of size 2×2 and vectors of size 2 remain, which are multiplied. From the resulting values (i.e., amplitudes), the edge weights are calculated by extracting common factors, and the decision diagram is constructed. Thus, multiplication of decision diagrams mainly revolves around recursive traversals of the involved decision diagrams.

Overall, using those structures often yields representations of states and operations which are much more compact than the exponentially large vectors and matrices. More precisely, as shown for example in [6], [21], the size of decision diagrams representing states and operations does not solely depend on the number of involved qubits (as is the case for vector and matrix representations), but on the complexity of the represented quantum object. Thus, representations of quantum operations stay of sub-exponential size, as long as the number of target qubits is low. The size of quantum states depends on the complexity of the represented state. A good case example is the entanglement circuit, which generates the GHZ state and which can be simulated in linear runtime with increasing number of qubits. On the other hand, Google's quantum supremacy circuits [41], which are designed to be exceptionally hard to simulate classically, are bad case examples for which decision diagrams also grow exponentially with the number of qubits.

C. Open Question: Consideration of Noise

As reviewed above and confirmed by several evaluations in previous works such as [6], [21]–[24], quantum circuit simulation with decision diagrams offers a promising alternative to other approaches that, due to the exploitation of using sharing for redundant sub-structures, may allow for better performance. However, almost all investigations conducted thus far in this direction assumed *perfect* quantum computers, i.e., did not consider noise effects.

In this work, we are investigating whether the promises of quantum circuit simulation with decision diagrams can be kept when noise effects are additionally considered. The basis for this is thereby rather straightforward: In principle, corresponding errors and how they affect the quantum states and operations are well understood. In fact, corresponding models are already available (see for example [30]) and can accordingly be realized on top of decision diagrams as well. However, additionally incorporating them makes the problem of quantum circuit simulation (which is already of exponential complexity) even harder. This raises the questions of what effects (with respect to the performance) the consideration of noise in decision diagram-based quantum circuit simulation has and, if it degrades the performance, what more advanced strategies could be employed to mitigate performance losses as much as possible.

The following sections address these questions in detail. To this end, two directions are investigated:

- A *deterministic* consideration of noise effects: Here, the formalism of (perfect) quantum circuit simulation (i.e., the representation of quantum states and operations as reviewed in Section II-A) is extended to additionally represent all considered occurrences of corresponding errors and their probabilities. This allows to *deterministically* track noise effects during the simulation, but also makes the problem of quantum circuit simulation even harder. This direction is covered in detail in Section IV.
- A *stochastic* consideration of noise effects: Here, the corresponding error effects are considered in a stochastic fashion, i.e., the considered errors are imposed during single simulation runs (based on their probability). By having several such runs, the real final state can be approximated using Monte Carlo sampling. This allows to keep the actual simulation runs more efficient, but may require a substantial number of runs for proper sampling and, after all, remains a stochastic (rather than deterministic) approach. This direction is covered in detail in Section V.

Table I
FORMALISMS FOR DETERMINISTIC SIMULATION OF NOISE

State representation	Let $\|\phi\rangle$ be a state vector representing a quantum system. The corresponding *density matrix* is defined as $$\rho = \|\phi\rangle \langle\phi\| \text{ with } \langle\phi\| := \|\phi\rangle^{\dagger}. \quad (1)$$
Applying operations	Applying an operation specified by the unitary U to a quantum system given by the density matrix ρ yields the density matrix $$\rho^{*} = U\,\rho\,U^{\dagger}. \quad (2)$$
Error representation	Using the operator-sum representation, an error is represented by a tuple $(E_0, E_1, \ldots, E_m)$ of *Kraus matrices* that satisfy the condition $$\sum_{i=0}^{m} E_i^{\dagger} E_i = I. \quad (3)$$
Error application	Applying an error specified by the Kraus matrices $(E_0, E_1, \ldots, E_m)$ to a quantum system given by the density matrix ρ yields the density matrix $$\rho' = \sum_{i=0}^{m} E_i \rho E_i^{\dagger}. \quad (4)$$

Finally, the actual performance of the resulting solutions proposed for both directions is evaluated and compared in Section VI.

IV. Deterministic Simulation of Noise

In this section, we introduce a method to deterministically consider noise effects during quantum circuit simulation with decision diagrams. For this, the formalism introduced in Section II is not sufficient anymore. We therefore extend that formalism to also support the deterministic consideration of noise effects during the simulation. Next, we investigate how this affects decision diagram-based simulation. Finally, based on these observations, we propose an advanced approach to address the identified challenges.

A. Formalism

As reviewed in Section II-B, noise probabilistically affects qubits and leaves the quantum state in a mixture of possible states. The number of possible states increases exponentially with each error and it quickly becomes infeasible to separately track all possible quantum states. Fortunately, quantum mechanics can describe such state mixtures, using so-called *density matrices* [30]. Density matrices can be derived from state vectors as defined by Eq. 1 in Table I. The resulting representation offers a merged state description independently of how many individual pure states the mixed state consists of. The following example illustrates this concept:

Example 8. *Consider the quantum state $|\psi'\rangle$ from Example 2, which is in the state*

$$0 \cdot |00\rangle + \frac{1}{\sqrt{2}} \cdot |01\rangle + 0 \cdot |10\rangle + \frac{1}{\sqrt{2}} \cdot |11\rangle,$$

or, using vector notation, $1/\sqrt{2} \cdot [0\ 1\ 0\ 1]^\top$. *The* density matrix ρ' *representing this state is given by*

$$\rho' = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \boxed{0} & 0 & 0 & 0 \\ 0 & \boxed{\frac{1}{2}} & 0 & \frac{1}{2} \\ 0 & 0 & \boxed{0} & 0 \\ 0 & \frac{1}{2} & 0 & \boxed{\frac{1}{2}} \end{bmatrix}.$$

Like the vector representation, the density matrix contains the probabilities of measuring specific basis states. Now, however, the probabilities are reflected in the diagonal elements (highlighted in gray) of the matrix. More precisely, the diagonal entries from the first element in the upper-left to the last element in the bottom-right represent the probabilities for measuring $|00\rangle, |01\rangle, |10\rangle$, *and* $|11\rangle$*, respectively. Hence, measuring this state would yield* $|01\rangle$ *or* $|11\rangle$*—both with probability* $1/2$.

Since quantum states are now represented by density matrices (rather than vectors), obviously also the application of quantum operations (represented by matrices thus far) needs to be adjusted. Instead of a matrix-vector multiplication, now two matrix-matrix multiplications are required as defined by Eq. 2 in Table I.

Example 9. *Analogous to Example 2, we apply a CNOT operation to* $\rho' = |\psi'\rangle\langle\psi'|$*, that negates the amplitude of the second qubit if the first qubit is set to* $|1\rangle$*. This is given by*

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\rho^\star} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{CNOT}} \cdot \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}}_{\rho'} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{CNOT}^\dagger}$$

The new quantum state $\rho^\star$ *represents the same state as* $|\psi^\star\rangle$ *from Example 2. Measuring measuring this state would also yield* $|01\rangle$ *or* $|10\rangle$*—both with probability* $1/2$.

Next, we look at errors, which can be represented by a tuple of Kraus operators as defined by Eq. 3 in Table I.

Example 10. *The depolarization error from Example 3 is given by* $D = (E_0, E_1, E_2, E_3)$ *with*

$$\begin{gathered} E_0 = \sqrt{1 - \frac{3p}{4}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, E_1 = \sqrt{\frac{p}{4}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ E_2 = \sqrt{\frac{p}{4}} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \text{ and } E_3 = \sqrt{\frac{p}{4}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{gathered} \tag{5}$$

Amplitude damping (illustrated in Example 4), also known as T1 decoherence error can be described by $T1 = (E_0, E_1)$ *with*

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, E_1 = \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix} \tag{6}$$

and, finally, the T2 decoherence error can be described as $T2 = (E_0, E_1)$ *with*

$$E_0 = \sqrt{p} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, E_1 = \sqrt{1-p} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{7}$$

The variable p*, which occurs in all Kraus matrices, represents the probability that an error occurs [30]. This probability is a parameter of the specific quantum computer and has to be provided to the simulator.*

Finally, errors are applied to the density matrix as defined in Eq. 4 of Table I, i.e., each Kraus operator is applied to (a copy) of the density matrix. Afterwards, all individual result matrices are summed up, into a density matrix representing the new quantum state. To illustrate this, consider the following example:

Example 11. *Suppose an amplitude damping (T1) error affects the second qubit of state* $\rho^\star$ *(from Example 9) with a probability of 2 % (p=0.02). The description of the T1 error is given by the Kraus matrices provided in Eq. 6. Applying each Kraus matrix to the second qubit of state* $\rho^\star$ *results in*

$$\underbrace{\begin{bmatrix} 0.01 & 0 & 0 & 0.494 \\ 0 & 0.49 & 0.494 & 0 \\ 0 & 0.494 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\rho^\times} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0.494 \\ 0 & 0.49 & 0.494 & 0 \\ 0 & 0.494 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{(I \otimes E_0)\rho^\star(I \otimes E_0)^\dagger} + \underbrace{\begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{(I \otimes E_1)\rho^\star(I \otimes E_1)^\dagger}$$

The resulting density matrix $\rho^\times$ *contains the effect of the error: While the probability for measuring* $|10\rangle$ *is unchanged, the probability of measuring* $|01\rangle$ *has dropped to 49 % and, now, there is a probability of 1 % to measure* $|00\rangle$*. Hence, the probability that the second qubit is measured* $|0\rangle$ *has increased by 2 %, reflecting the damping error assumed above.*

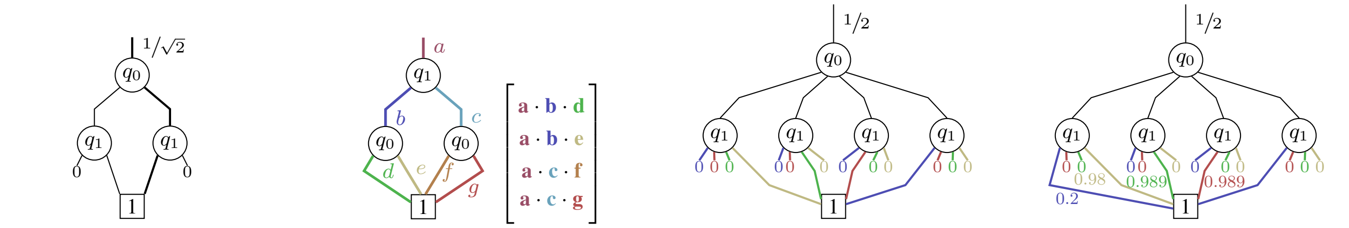
Having this mathematical description of noise, a wide range of errors affecting present-day quantum computers can be considered (for each error, just a corresponding representation in terms of Kraus matrices needs to be provided). Thus, the formalism can be used as a basis for mimicking the execution of real quantum computers for quantum circuit simulation.

B. Effect to Simulation

Using the formalism presented above, approaches for quantum circuit simulation can accordingly be extended. For straightforward approaches as reviewed in Section III-A (representing corresponding vectors and matrices in terms of 1- and 2-dimensional arrays, respectively), this is particularly straightforward and revolves mainly around changing the state representation from vectors to matrices. This can be addressed by extending the underlying data structure (i.e., arrays) accordingly. The newly required operations, i.e., matrix-matrix multiplications and the addition of matrices, can easily be extended as well or are often already supported by the underlying libraries anyway. Thus, many such quantum circuit simulators supporting the consideration of noise are already available (see, e.g., [12]–[18]). However, switching from state vectors to density matrices makes the curse of dimensionality and, by this, the resulting complexity (as discussed in Section III-A) even worse—severely limiting the corresponding approaches.

The question remains, whether and, if so, how quantum circuit simulation based on decision diagrams is affected by extending the formalism. Since compactness constitutes one of the main advantages of decision diagram-based quantum circuit simulation, there still might be potential. On the other hand, the increased complexity caused by the additional formalism certainly constitutes a challenge. The following example sheds some light on this:

Example 12. *Consider again the state* $\rho^\star = |\psi^\star\rangle\langle\psi^\star|$ *in both, the vector (see Example 2) and the density matrix (see Example 9) representation. In Fig. 2a and Fig. 2c the corresponding decision diagram representations are provided. The size of the decision diagrams is rather similar (4 nodes vs. 6 nodes, although the corresponding vector and density matrix are of*



(a) DD of a state vector $|\psi^\star\rangle$ (b) Generic 2-qubit state vector (c) DD rep. of $\rho^\star$ (d) $\rho^\star$ after T1 error to q_1e $(p = 0.02)$

Fig. 2. Decision diagram (DD) representation of state vectors and density matrices

size 2^2 and $2^2 \times 2^2$, respectively). This indicates that, using the density matrix formalism, does not necessarily harm the ability of decision diagrams to represent states/operations in a compact fashion (even though a somewhat larger representation results).

Naturally, this example does not show that decision diagrams *always* provide a efficient representation for a quantum state. In fact, previous works such as e.g., [6], [21] clearly show that the worst-case complexity of decision diagrams is exponential (although polynomial representations are possible). But the example provides a promising indication that the characteristic of offering more compact representations is not automatically lost when this formalism is applied.

However, having a compact representation is not enough. In addition, an efficient realization of the matrix-matrix operations (mainly multiplication and addition), as presented in Section IV-A, is also necessary. In the case of multiplication, related work such as [6] already provides efficient solutions for vector-matrix multiplications—extending this to matrix-matrix multiplications is straightforward. Matrix-matrix addition, however, which is required for applying errors as defined in Eq. 4, has thus far only been a sub-operation and turns out to be particularly challenging.

Recall that adding two matrices is done by adding all elements sharing the same index. Thus, access to *all* matrix elements is required. When working with decision diagrams, this constitutes a bottleneck, since each matrix element is encoded into the tree structure and must be restored for the operation. In Fig. 2b it is illustrated by color and notation how each element within a vector is encoded into a decision diagram (matrices are encoded analogously (see III-B)). Therefore, accessing *all* matrix elements in order to add them, requires to decompress and thereby traverse the entire decision diagram. This is in contrast to multiplication, where, depending on the applied quantum operation, multiplication can in some cases be applied without decompressing any elements at all [42].

Overall, our investigation indicates that the decision diagram structure may also be suitable when used for noise-aware quantum circuit simulation using density matrices. At the same time, due to the extended formalism, new challenges arise which must be addressed, namely how to efficiently apply noise to the state without having to traverse the entire decision diagram.

C. Advanced Simulation Approach

A major challenge when considering noise effects during decision diagram-based quantum circuit simulation is to efficiently realize the necessary operations (particularly Eq. 4). Applying noise heavily revolves around adding matrices, which requires access to *all* matrix elements and, thus, triggers the complete traversal of the involved decision diagrams. This causes an exponential overhead during the simulation, which severely impacts the performance of corresponding approaches. To address this challenge, we investigated alternatives that either completely avoid adding matrices or, at least, involve only (smaller) sub-matrices. Our investigations finally led to an alternative whose main idea is based on the following three observations:

First, adding matrices to apply errors can, in some cases, be avoided altogether. Specifically, the T2 error can be realized by multiplications only, i.e., the effect of this error on a single qubit can be captured by

$$\begin{bmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{bmatrix} \longmapsto \begin{bmatrix} \mathbf{a} & (2p-1)\mathbf{b} \\ (2p-1)\mathbf{c} & \mathbf{d} \end{bmatrix}, \tag{8}$$

where p represents the probability that an error occurs[3]. Applying the T2 error can therefore be realized by applying a factor to specific edges of the decision diagram, for which *no* amplitude has to be decoded at all.

Second, sequentially applying errors, as defined in Eq. 4, requires accessing the decision diagram multiple times. This process can be substantially improved by explicitly enforcing the error effects directly on the corresponding nodes of the affected qubits (as illustrated in Example 13). In doing so, we can apply all desired effects to all qubits with just one traversal of the decision diagram.

Third, due to the tree-like structure of decision diagram representations of quantum states, applying operations to the quantum state often only affects parts of the decision diagram. More precisely, in a decision diagram, every qubit is represented by one or more nodes. Applying an operation to a qubit only modifies the outgoing edges from the nodes representing this specific qubit. Predecessor nodes are only indirectly affected during the normalization process (as described in Section III-B).

Based on the observations above, we are proposing an advanced scheme to realize Eq. 4 for applying errors to quantum decision diagrams. The key idea is to directly modify the decision diagram to reduce the overhead and merge separate operations. More precisely, due to our third observation, we know that applying operations to qubits affects only the outgoing edges of nodes representing this qubit. Using this knowledge, it becomes trivial to predict how each operation will modify the decision diagram. This allows us to aggregate distinct operations—be it matrix-matrix

[3] The colors in the equation illustrate how the matrix elements relate to decision diagram edges of Fig.2c

multiplications/additions, applying separate noise effects, or applying noise operations to different qubits—into a single operation.

Example 13. *To illustrate this idea, consider again Example 11 where we apply an amplitude damping (T1) error with a probability of 2 % to the second qubit (q_1) of the quantum state ρ^*. However, now, we apply the error directly to the decision diagram representation of ρ^* (given in Fig. 2c). The effects of the T1 error with $p = 0.02$ are given by the following transformation*[4]

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \longmapsto \begin{bmatrix} a + 0.02 \cdot d & \sqrt{0.98} \cdot b \\ \sqrt{0.98} \cdot c & 0.98 \cdot d \end{bmatrix}. \qquad (9)$$

Modifying all nodes labeled q_1 accordingly, leads to the new decision diagram shown in Fig. 2d. This new decision diagram represents ρ^* after the T1 error with $p = 0.02$ has been applied and is equal to $\rho^\times$ from Example 11.

Overall, by extending the formalism to density matrices and state mixtures, deterministic consideration of errors is possible. While this comes with an substantially increased complexity of the state description, the main advantage of decision diagram-based state representation—compactness—is not completely lost. To address the additional complexity of applying error operations (compared to standard quantum operations), we proposed an alternative, more efficient scheme. However, despite mitigating the negative effects of deterministically simulating with error effect, they still unavoidably impact the scalability of corresponding simulation approaches due to the increased state complexity. In the next section, another approach for quantum circuit simulation with consideration to errors is proposed which addresses this problem.

V. Stochastic Simulation of Noise

In this section, we introduce a method to stochastically consider noise effects during quantum circuit simulation with decision diagrams. The main idea is as follows: Whenever an error could occur, we mimic the effect of this error with some probability. Simulating in such a fashion generates one possible final state. Sampling enough such final states, opens the door to (Monte-Carlo) approximation to estimate the actual amplitudes of the final quantum state. By this, a more efficient simulation is achieved at the expense of having an approximation (rather than an exact deterministic result). In the following, this approach is described by providing the formalism for stochastic quantum circuit simulation first. Afterwards, we discuss how this affects the quantum circuit simulation before an advanced approach is presented that improves the procedure based on those insights.

A. Formalism

In order to realize a stochastic quantum circuit simulator, we need (1) a formalism that allows one to impose error effects on quantum state representations, and (2) a mathematical model to predict how many samples are necessary for a reliable result.

Addressing (1), noise effects occurring during quantum computing can be viewed as (unwanted) operations on the quantum state. Using this idea allows us to reuse the procedures for quantum circuit simulation introduced in Section II. More precisely, quantum states and operations are represented by vectors and matrices, and operations are applied using matrix-vector multiplications. Then, error effects can be seen as further operations that only occur with some probability p. Such an erroneous operation leaves the state in a mixture of i possible outcomes $|\psi_i\rangle$ each with some probability p_i. This state mixture depends on the probability of the error, as well as the type of error.

Errors can then be represented in terms of Kraus matrices. Gate errors can be mimicked by depolarization errors, as already illustrated in Example 3. Similarly, phase flip (T2) decoherence errors can be realized by applying a Z-operation. Due to their irreversible nature, mimicking the effect of amplitude damping (T1) decoherence errors, cannot be handled so easily. While for other errors the probability that the error is applied only depends on the error probability itself, the probability of applying amplitude damping also depends on the quantum state the error is applied to. How this can be handled is illustrated in the following example.

Example 14. *Consider the state $|\psi^*\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ from Example 2 (shown in Fig 1a) and suppose that it is subject to amplitude damping. If the error occurs with probability p, its effect can be described by the matrices $\mathrm{E}_0 = \left[\begin{smallmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{smallmatrix}\right]$ and $\mathrm{E}_1 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{smallmatrix}\right]$ [30]. But since the probability p of amplitude damping depends not only on p but also the given state of the qubit, E_0 and E_1 have to be applied to $|\psi^\star\rangle$ as well. This either yields a state vector whose squared norm is $\frac{p}{2}$ (in case E_0 is applied to $|\psi^*\rangle$) or $1 - \frac{p}{2}$ (in case E_1 is applied to $|\psi^*\rangle$). These probabilities can eventually be used to randomly choose and normalize one possible output. So, considering this error (amplitude damping on the first qubit) yields the mixture $\{(\frac{p}{2}, |01\rangle), (1 - \frac{p}{2}, \frac{1}{\sqrt{2-p}}\,|00\rangle + \frac{\sqrt{1-p}}{\sqrt{2-p}}\,|11\rangle)\}$.*

Having a scheme to stochastically apply errors, the next question becomes, how many samples must be generated to get a good prediction of the true probabilities. Recall that by simulating in a stochastic fashion, we get *one possible* final state $|\tilde{\psi}\rangle$ from the actual mixture $\{(p_i, |\psi_i\rangle)\}$: $|\tilde{\psi}\rangle = |\psi_i\rangle$ with probability p_i. We can approximate the true distribution of sampled output states by forming empirical averages using Monte-Carlo approximation. This scheme is especially well suited to accurately learn properties of the final state (distribution) without the need to keep track of the complete (exponential) distribution. In quantum computing, interesting properties (e.g. the fidelity with another state or the outcome probability of measurements) can often be described as a quadratic function in the state vector, i.e., $o_l = |\langle\omega_l|\psi\rangle|^2$ with $l \in \{1, \ldots, L\}$ for L properties. In the case of a probabilistic state mixture $\{(p_i, |\psi_i\rangle)\}$, such a quadratic property becomes

$$o_l = \sum_i p_i \, |\langle\omega_l|\psi_i\rangle|^2 \qquad (10)$$

and can be approximated by an empirical average over M samples $|\tilde{\psi}_j\rangle$ from the distribution

$$\hat{o}_l = \tfrac{1}{M} \sum_{j=1}^{M} \left|\langle\omega_l|\tilde{\psi}_j\rangle\right|^2 \quad \text{(Monte Carlo).} \qquad (11)$$

[4] The color coding shows how each matrix element relates to one of the outgoing edges of a decision diagram node. That is, the matrix elements a, b, c, and d, relate to the four outgoing edges of a decision diagram node from left to right.

Furthermore, the same sample collection $\left\{|\tilde{\psi}_1\rangle, \ldots, |\tilde{\psi}_M\rangle\right\}$ can be used to estimate multiple quadratic properties at once.

Using that, a sufficient sample set is motivated by the following theorem (taken from [29]):

Theorem 1. *Fix a collection of* L *(arbitrary) quadratic properties* (10)*, as well as* $\epsilon \in (0,1)$ *(accuracy) and* $\delta \in (0,1)$ *(confidence). Then,* $M = \log(2L/\delta)/(2\epsilon)^2$ *state samples suffice to accurately approximate* all *target properties with high confidence:* $\max_l |\hat{o}_l - o_l| \leq \epsilon$ *with probability at least* $1-\delta$*.*

Proof. Fix a target property $o_l = \sum_i p_i |\langle \omega_l | \psi_i \rangle|^2$. Conducting a single stochastic run yields the correct property in expectation, i.e., $\mathbb{E}|\langle \omega_l | \tilde{\psi}_j \rangle|^2 = o_l$. Standard concentration inequalities, like Hoeffding, imply $\Pr\left[|o_l - \hat{o}_l| \geq \epsilon\right] \leq 2\mathrm{e}^{-2M\epsilon^2}$. The claim follows from taking a union bound over all L target approximations and inserting the advertised value of M. □

The required number of samples M scales inverse quadratically with the target accuracy ϵ—as is typical in Monte Carlo. More interestingly, M depends only logarithmically on the number L of tracked properties and is independent of the size of the system (i.e., the number of qubits of the quantum state). This logarithmic scaling can help to counteract the curse of dimensionality. For example, to approximate all $N = 2^n$ outcome probabilities of a n-qubit system up to ϵ accuracy, only roughly n/ϵ^2 samples are required.

Overall, stochastic quantum circuit simulation allows the consideration of errors without the need to extend the representation of quantum states. This is achieved by viewing error effects as "normal" operations during the quantum circuit simulation, which are randomly applied during the computation, based on their probability. By generating enough samples in such a fashion, the probabilities for measuring specific states can be estimated.

B. Effects on the Simulation

Stochastic quantum circuit simulation is an interesting alternative to deterministic consideration of errors. It avoids the increase in complexity from 2^n-vectors to $2^n \times 2^n$- density matrices and can be implemented in a straightforward fashion on top of an "error-free" quantum circuit simulator. Accordingly, some quantum circuit simulators already make use of this (e.g. [11]–[13], [16], [20]). However, at the same time the deterministic description of the quantum state is lost and interesting properties of the quantum state can only be approximated. In order to get reliable and accurate results, sufficiently many samples have to be generated.

Example 15. *In our experiments, we want to use stochastic quantum circuit simulation to accurately predict properties of a quantum circuit simulation. More precisely, we want at most an error margin of* $\epsilon = 0.01$ *and a confidence of 95 %* ($\delta = 1 - 0.95$)*. Additionally, during the simulation, we want to track 25000 properties (*$L = 25000$*). Using Theorem 1, we can predict the number of required samples:*

$$M = \frac{\log\left(\frac{2L}{\delta}\right)}{2\epsilon^2} = \frac{\log\left(\frac{2 \cdot 25000}{1-0.95}\right)}{2 \cdot 0.01^2} = 30000 \qquad (12)$$

Thus, at least 30000 *samples are necessary to predict* 25000 *properties of the quantum state with 95 % confidence and an error margin of 0.01.*

The question therefore becomes how to generate the required samples—each exponential in size.

C. Advanced Simulation Approach

A major challenge of stochastic quantum circuit simulation is to efficiently generate the required number of samples. Decision diagrams are an interesting candidate to achieve this. They have already proven to be a suitable data structure to efficiently represent quantum states and also support the operations necessary for stochastic consideration of errors. Decision diagrams offer the following further potential for improvement when considering stochastic quantum circuit simulation:

Exploiting Concurrency: Due to the compact representation and heavy use of sharing, decision diagrams have been unsuited for concurrent execution thus far [43]. Because of that, decision diagram-based quantum circuit simulation could not fully exploit the available hardware resources of modern (multi-core) computers yet (in contrast to simulation approaches based, e.g., on arrays, which are easy to parallelize; see, e.g., [9], [11]–[18]). Interestingly, stochastic quantum circuit simulation offers the prospect of resolving the apparent conflict between optimizing memory (by using decision diagrams) and exploiting concurrency (to speed up matrix-vector multiplication) by other means. Now, the full potential of sharing with decision diagrams can be used, since the remaining hardware power can still be put to good use to generate further samples in parallel.

Note that parallelization is, of course, not a new approach. Parallel execution is a well known and general feature of Monte-Carlo-type approximations—and stochastic quantum circuit simulation is merely an interesting use-case. State-of-the-art stochastic simulators such as [11]–[13], however, do not seem to make use of this yet. This is most likely due to the fact that parallel executions are still more beneficial for improving, e.g., the matrix-vector multiplications of single runs—leaving no free resources for parallelizing the generation of samples. This does not constitute a problem for decision diagram-based simulation, as no parallel resources are needed for a single simulation—allowing one to use the hardware power for generating samples in parallel.

Stacking operations: While stochastic quantum circuit simulation can be directly implemented on top of the concepts from Section II, the compressed data structure of decision diagrams allows for further optimization of matrix-vector multiplications by exploiting an idea first suggested in [44]. More precisely, simulating a quantum circuit boils down to applying m quantum operations $U_1, U_2, \ldots, U_m$ to an initial state $|\psi\rangle$, resulting in the final state $|\psi\rangle^m$, i.e.,

$$|\psi\rangle^m = U_m \ldots U_2 U_1 \cdot |\psi\rangle \,. \qquad (13)$$

Matrix-vector multiplications are usually more efficient than matrix-matrix multiplications, which is why Eq. 13 is usually solved by consecutively applying operations onto the vector representing the quantum state. However, due to the compressed data structure of decision diagrams, matrix-matrix multiplications can be considerably faster than matrix-vector multiplications. This has to do with the fact that, when working with decision diagrams, the cost of operations depends on the complexity of the involved decision diagrams. Since a state vector is often more complex than a quantum operation, "stacking" (i.e., multiplying) quantum operations with each other before applying them to the quantum state can improve the simulation speed. But, simply stacking *all* quantum operations before applying them to the quantum state does not work, because whenever two quantum operations are being multiplied, the resulting stacked operation becomes more complex. Therefore, the advantage of this approach relies on finding a good heuristic of how many operations are stacked, before they are applied to the quantum state. Additionally, this scheme only improves the performance when the quantum state is so complex that stacking operations is faster.

When we employ this scheme during stochastic quantum circuit simulation, both those aspects are considered, i.e., whenever an "intentional" quantum operation is applied, the error effects are stacked on top of it. On the one hand, this is a decent heuristic since all operations target the same qubit, which keeps the resulting stacked operations compact. On the other hand, since we simulate with consideration to errors, the quantum state naturally tends to become more complex anyway—increasing the size of the decision diagram representing it and, thus, making operations stacking more viable.

Overall, stochastic quantum circuit simulation allows one to consider errors without the substantial overhead on top of the already exponential problem of quantum circuit simulation. However, using this formalism, the deterministic description of the quantum state is lost and in order to generate reliable results the simulation must be repeated many times. In the next section we will analyze how this affects the performance compared to the deterministic simulation scheme.

VI. Evaluation

In order to empirically evaluate the proposed noise-aware quantum circuit simulation approaches, we implemented the concepts presented in Sections IV and V in C++, on top of the open-source decision diagram package taken from [6], [45]. Afterwards, we compared their performance in different scenarios, i.e., we evaluated the proposed optimizations, compared them against industry-grade state-of-the-art simulators (by IBM and Atos), and finally, compared both noise-aware simulation styles with each other. In this section, the results of these evaluations are summarized and discussed.

A. Setup

To obtain meaningful results, we used a range of different benchmark sets. More specifically, we considered typical use cases incorporating quantum-mechanical effects with an increasing number of qubits. Namely, the *Entanglement* circuit—a quantum algorithm that generates the GHZ state—as well as the *Quantum Fourier Transform* (QFT) [30]. Both benchmarks have the additional benefit of being good case examples for decision diagram-based quantum circuit simulation without consideration of errors [21] and, thus, allow to us directly observe the effects of noise. In addition to this, we also used the *QASMBench* benchmark suite in our experiments which contains a broad range of different quantum circuit algorithms [46]. To ensure a fair comparison, we translated all quantum circuits into QASM basic gates prior to the experiments [47], i.e., arbitrary single-qubit gates, as well as controlled-X and controlled phase shift gates.

In the experiments, we assumed the noise effects reviewed in Section II-B. That is, we approximated gate errors using depolarization with a probability of 0.1 %, amplitude damping error (T1) with 0.2 % probability, and phase flip error (T2) with 0.1 % probability. Each noise effect is (probabilistically) applied to the qubit, whenever it is used. For the stochastic simulators, we generated 30000 samples to predict the final state (according to Theorem 1, this translates to tracking 25000 properties with an error margin of 0.01 and a confidence of 95 %).

All experiments have been conducted on a system with 96 cores running at a clock frequency of 2.2 GHz and 1.5 TB of RAM. As methods, we considered the *LinAlg* simulator of Atos' *Quantum Learning Machine* (QLM) [13] from version 1.5.1, the *density matrix and statevector* simulators of IBM's *Qiskit* [12] in version 0.36.0, as well as the decision diagram-based methods proposed in this paper. While the QLM simulator ran directly on the system, we ported Qiskit and the proposed simulators to the machine using Docker [48]. We choose Docker since its virtualization overhead is negligible [49], which ensures a fair comparison. For all experiments, we applied a timeout of half an hour (1800 seconds). In the following summary, we only list the most relevant benchmarks, i.e., we omit benchmarks that could be simulated by *all* approaches in less than a few seconds, and, benchmarks that no simulator could simulate within the given time limit. For transparency, implementations of both proposed solutions are available online as open-source at https://github.com/cda-tum/ddsim.

B. Comparison Between Implementations

First, we evaluated the advantage of the proposed optimizations for deterministic and stochastic noise-aware quantum circuits simulation. To this end, we implemented two versions for each simulation style: One in which we implemented the necessary concepts in a basic fashion (i.e., as presented in Section IV-A and Section V-A)) and one where we implemented the optimized approaches presented in Section IV-C and Section V-C, respectively.

In Table II, the results for the deterministic approach and the stochastic approach are provided. In both tables, the name of the benchmark, the number of qubits #Q, the number of gates #G, the runtime for the basic and the optimized implementation, as well as the relative runtime with the optimization compared to the basic implementation.

For both approaches, the optimizations substantially improve the runtime for many circuits and do not worsen the runtime for any of the tested algorithms. In the case of the deterministic simulation approach, the performance gains are often on the order of *magnitudes*. For the stochastic simulator, the optimizations improve the simulation by around 20 % for several circuits.

C. Comparison to Related Work

In a second series of evaluations, we compared the proposed simulation approaches (in their optimized version) to other quantum circuit simulators. We used two simulators that constitute good representatives for the current state of the art, i.e., the *LinAlg* simulator of Atos' *Quantum Learning Machine* (QLM) [13], as well as the *density matrix and statevector* simulators of IBM's *Qiskit* [12]. In contrast to the other evaluations—where decoherence and depolarization errors have been considered together—here we additionally considered the scenario with depolarization errors only. This allows us to

Table II
COMPARISON BETWEEN BASIC AND OPTIMIZED DECISION DIAGRAM-BASED SIMULATORS

(a) Optimization for deterministic simulation

Benchmark	#Q/#G	Basic	Optimized	Rel. Runtime
basis_trotter	4/1625	18 s	**1 s**	8 %
qaoa	6/269	22 s	**4 s**	21 %
vqe_uccsd	6/2281	>1800 s	**486 s**	-
QFT	9/192	1432 s	**241 s**	16 %
qpe	9/150	>1800 s	**392 s**	-
entanglement	26/26	1407 s	**174 s**	12 %
entanglement	27/27	>1800 s	**271 s**	-
entanglement	28/28	>1800 s	**432 s**	-

(b) Optimization for stochastic simulation

Benchmark	#Q/#G	Basic	Optimized	Rel. Runtime
basis_trotter	4/1625	28 s	**16 s**	56 %
vqe_uccsd	8/10806	1110 s	**771 s**	69 %
ising	10/479	152 s	**122 s**	80 %
sat	11/678	728 s	**623 s**	85 %
seca	11/281	561 s	**457 s**	81 %
cc	12/48	225 s	**189 s**	83 %
multipler	15/123	392 s	**335 s**	85 %
bigadder	18/284	114 s	**90 s**	78 %

evaluate how the different noise models affect the performance. Additionally, to take into account that two-qubit operations are more error prone compared to single-qubit operations, we double the error probabilities for two-qubit gates.

In Tables IIIa and IIIb, the results of the experiments are provided. The structure of both tables is the same: Each table contains the results for the entanglement and QFT circuit with an increasing number of qubits as well as a selection of the QASMBench circuits for both considered noise models. For each experiment we provide the name of the benchmark, the number of qubits #Q, the number of gates #G, as well as the required runtime for simulating it with the QLM, Qiskit, and the decision diagram-based solution in seconds. Furthermore, we do not list any results from Atos' QLM simulator for the QASMBench benchmarks, due to space issues.

The results clearly show where noise-aware quantum circuit simulation with decision diagrams improves the current state-of-the-art, but also unveils its shortcomings. More precisely, looking at the results of the deterministic noise-aware simulation in Table IIIa, we see that the decision diagram-based solution performs worse than the considered state of the art for most benchmarks. The decision diagram-based simulator is only considerably faster (and more scalable) for simulating the entanglement circuit. Notably, the hardware model considerably affects the performance of the proposed solution, i.e., when only depolarization errors are considered the runtime considerably improves for the entanglement circuit, *QFT*, *sat*, *seca*, and *multiply* (with 13 qubits). The results show that the main advantage of decision diagram-based simulators—namely their compactness—is not completely lost when simulating with density matrices. On the other side, the worse results for the remaining benchmarks also show that the additional complexity introduced by the noise cannot yet completely be mitigated by the current type of decision diagrams. Here, the current state-of-the-art, i.e., Atos' QLM and IBM's Qiskit simulators, still seem to provide the better solutions (or, at least, solutions that are not that much affected by the considered benchmark and/or error model). By this, these results unveil that, although decision diagrams may provide some promises for an efficient deterministic consideration (as seen by the entanglement benchmark), they still struggle with providing a robust solution that works in general. This motivates further research, e.g., towards alternative decision diagram types and/or further optimization dedicated to the compact representation of density matrices.

In contrast, a different picture is seen when the stochastic scheme is considered. Here, the benefits of decision diagrams allow for tremendous improvements in some cases compared to the state of the art and substantially improved scalability for the entanglement and QFT circuit. This confirms that decision diagrams indeed provide a useful alternative to existing noise-aware quantum circuit simulators. As with the deterministic simulation scheme, once again, the complexity of the hardware model strongly affects the simulation time, as reflected in the data. In all cases the runtime considerably improves when only depolarization errors are considered.

D. Comparison Between Deterministic and Stochastic Scheme

Finally, a third series of evaluations considered the difference in the performance of the deterministic and stochastic simulation scheme. To this end, we directly compared both approaches. To gain more insight into the behavior of both simulation schemes, we also tracked the resource consumption of both simulators (in addition to the runtime). Table IV provides the obtained results. Again, the table contains the name of the benchmark, the number of simulated qubits #Q, the number of gates #G, as well as the runtime in seconds for both simulation approaches. Furthermore, the average CPU usage and the maximum memory usage in MB are provided.

While the deterministic simulator is sometimes faster for smaller quantum circuits, the stochastic approach scales much better and manages to simulate circuits that are not feasible with the deterministic approach. The better performance of the stochastic simulator is most likely related to the more efficient use of the available hardware resources (using up to 92 times more processing power compared to the deterministic simulator). Nevertheless, the increased performance of the stochastic simulator comes with losing the deterministic description of the quantum state and, hence, a potential loss of accuracy. In practice, however, this drawback is often negligible since noise-aware quantum circuit simulation is only an approximation of real quantum computers in the first place and, as shown in our discussions in Section V and through Theorem 1, the error margin of the approximated results can be reduced to the desired accuracy by generating more samples. Hence, while having a deterministic approach as a baseline for an exact consideration of noise/errors, the (much more efficient and scalable) stochastic approach is more suited for the simulation of practically relevant instances.

VII. CONCLUSION

Decision diagrams offer a promising data structure for handling the complexity of representing quantum states and operations on classical computers. Accordingly they have found successful application in quantum circuit simulation. But their potential for corresponding simulations that additionally consider noise effects, i.e., the consideration of errors that frequently occur within real quantum computers, remained largely unexplored thus far. In this work, we investigated this

Table III
COMPARISON TO RELATED WORK RESULTS

(a) Comparison to state-of-the-art deterministic quantum circuit simulators

Error	Entanglement Circuit				QFT Circuit				QASMBench Benchmarks			
model	#Q/#G	QLM	Qiskit	Prop.	#Q/#G	QLM	Qiskit	Prop.	Benchmark	#Q/#G	Qiskit	Prop.
Decoherence and Depolarization	16/16	721 s	45 s	**1 s**	9/192	**2 s**	**2 s**	7 s	basis_trotter	4/1625	2 s	**1 s**
	17/17	>1800 s	206 s	**1 s**	10/241	4 s	**2 s**	149 s	qaoa	6/269	2 s	**1 s**
	18/18	>1800 s	776 s	**2 s**	11/291	13 s	**2 s**	>1800 s	vqe_uccsd	6/2281	**2 s**	7 s
	19/19	>1800 s	>1800 s	**3 s**	12/346	52 s	**4 s**	>1800 s	vqe_uccsd	8/10806	**5 s**	1362 s
	⋮	⋮	⋮	⋮	13/410	240 s	**10 s**	>1800 s	ising	10/479	**2 s**	>1800 s
	26/26	>1800 s	>1800 s	**161 s**	14/475	1117 s	**40 s**	>1800 s	sat	11/678	**2 s**	>1800 s
	27/27	>1800 s	>1800 s	**284 s**	15/548	>1800 s	**98 s**	>1800 s	seca	11/281	**2 s**	>1800 s
	28/28	>1800 s	>1800 s	**524 s**	16/624	>1800 s	**394 s**	>1800 s	multiply	13/123	**4 s**	274 s
	29/29	>1800 s	>1800 s	**943 s**	17/704	>1800 s	**1722 s**	>1800 s	multiply	15/573	**397 s**	>1800 s
	30/30	>1800 s	>1800 s	**1335 s**	18/793	>1800 s	>1800 s	>1800 s	cc	18/72	**934 s**	>1800 s
Depolarization	14/14	41 s	3 s	**1 s**	8/153	**2 s**	**2 s**	**2 s**	basis_trotter	4/1625	2 s	**1 s**
	15/15	168 s	13 s	**1 s**	9/192	2 s	**2 s**	9 s	qaoa	6/269	**1 s**	**1 s**
	16/16	710 s	45 s	**1 s**	10/241	4 s	**2 s**	33 s	vqe_uccsd	6/2281	**2 s**	6 s
	17/17	>1800 s	206 s	**2 s**	11/291	12 s	**2 s**	155 s	vqe_uccsd	8/10806	**5 s**	>1800 s
	18/18	>1800 s	774 s	**4 s**	12/346	51 s	**3 s**	>1800 s	ising	10/479	**2 s**	>1800 s
	19/19	>1800 s	>1800 s	**6 s**	13/410	239 s	**10 s**	>1800 s	sat	11/678	**2 s**	3 s
	⋮	⋮	⋮	⋮	14/475	1122 s	**40 s**	>1800 s	seca	11/281	2 s	**1 s**
	30/30	>1800 s	>1800 s	**590 s**	15/548	>1800 s	**101 s**	>1800 s	multiply	13/123	3 s	**1 s**
	31/31	>1800 s	>1800 s	**718 s**	16/624	>1800 s	**368 s**	>1800 s	multiply	15/573	**385 s**	>1800 s
	32/32	>1800 s	>1800 s	**877 s**	17/704	>1800 s	>1800 s	>1800 s	cc	18/72	**921 s**	>1800 s

(b) Comparison to state-of-the-art stochastic quantum circuit simulators

Error	Entanglement Circuit				QFT Circuit				QASMBench Benchmarks			
model	#Q/#G	QLM	Qiskit	Prop.	#Q/#G	QLM	Qiskit	Prop.	Benchmark	#Q/#G	Qiskit	Prop.
Decoherence and Depolarization	20/20	23 s	623 s	**12 s**	11/291	991 s	11 s	**9 s**	basis_trotter	4/1625	135 s	**21 s**
	21/21	42 s	1313 s	**13 s**	12/346	1606 s	22 s	**14 s**	qaoa	6/269	25 s	**10 s**
	⋮	⋮	⋮	⋮	13/410	>1800 s	49 s	**16 s**	vqe_uccsd	6/2281	143 s	**61 s**
	26/26	970 s	>1800 s	**13 s**	⋮	⋮	⋮	⋮	vqe_uccsd	8/10806	**700 s**	830 s
	27/27	>1800 s	>1800 s	**14 s**	17/704	>1800 s	357 s	**16 s**	ising	10/479	**42 s**	145 s
	28/28	>1800 s	>1800 s	**14 s**	18/793	>1800 s	>1800 s	**21 s**	sat	11/678	**23 s**	893 s
	29/29	>1800 s	>1800 s	**15 s**	19/886	>1800 s	>1800 s	**22 s**	seca	11/281	**8 s**	722 s
	30/30	>1800 s	>1800 s	**16 s**	⋮	⋮	⋮	⋮	multiply	13/123	**107 s**	479 s
	31/31	>1800 s	>1800 s	**16 s**	31/2372	>1800 s	>1800 s	**39 s**	multiply	15/573	14 s	**9 s**
	32/32	>1800 s	>1800 s	**16 s**	32/2526	>1800 s	>1800 s	**41 s**	cc	18/72	**197 s**	>1800 s
Depolarization	21/21	14 s	797 s	**12 s**	16/624	1031 s	55 s	**10 s**	basis_trotter	4/1625	19 s	**6 s**
	22/22	26 s	1661 s	**13 s**	17/704	1573 s	87 s	**11 s**	qaoa	6/269	**4 s**	9 s
	23/23	50 s	>1800 s	**13 s**	18/793	>1800 s	166 s	**11 s**	vqe_uccsd	6/2281	**40 s**	45 s
	⋮	⋮	⋮	⋮	19/886	>1800 s	360 s	**19 s**	vqe_uccsd	8/10806	**204 s**	424 s
	26/26	470 s	>1800 s	**13 s**	20/977	>1800 s	761 s	**21 s**	ising	10/479	**9 s**	124 s
	27/27	1071 s	>1800 s	**13 s**	21/1080	>1800 s	1653 s	**22 s**	sat	11/678	**11 s**	22 s
	28/28	>1800 s	>1800 s	**14 s**	22/1186	>1800 s	>1800 s	**22 s**	seca	11/281	**4 s**	6 s
	⋮	>1800 s	>1800 s	**14 s**	⋮	⋮	⋮	⋮	multiply	13/123	36 s	**9 s**
	31/31	>1800 s	>1800 s	**14 s**	31/2372	>1800 s	>1800 s	**26 s**	multiply	15/573	**5 s**	11 s
	32/32	>1800 s	>1800 s	**14 s**	32/2526	>1800 s	>1800 s	**27 s**	cc	18/72	92 s	**10 s**

issue. To this end, we considered two different approaches for noise-aware quantum circuit simulation, namely a deterministic and a stochastic scheme, and investigated corresponding solutions based on decision diagrams. We proposed advanced methods to address challenges caused by considering noise and implemented those concepts in C++. In an extensive evaluation, we investigated the value of the proposed optimizations, how the implemented simulators perform against state-of-the-art solutions from IBM and Atos, and how both noise-aware simulation schemes compare against each other.

The findings of this paper confirm the usefulness of decision diagrams for noise-aware quantum circuit simulation and provide a promising alternative for noise-aware simulation (which has been made available as open-source implementation at https://github.com/cda-tum/ddsim). Additionally, further potential for future work has been unveiled. More precisely, while decision diagrams offer tremendous improvements (speed-ups of several magnitudes and much better scalability) for the stochastic scheme, the deterministic scheme performs worse compared to the state-of-the-art. Alternative types of decision diagrams which are more suited for deterministic noise-aware simulation schemes should be developed. Next, it should be evaluated how approximation of decision diagrams, i.e., the artificial introduction of errors in the representation of the quantum state, affects the reliability and performance of noise-aware quantum circuit simulation. While this has already been discussed for simulation without consideration of errors (see [37]) noise-aware quantum circuit simulation has different requirements and should therefore be considered separately. Additionally, while the proposed solution shows the viability of decision diagram-based noise-aware quantum circuit simulation, further functionality, i.e., allowing for more errors types (e.g., biased noise, idle noise, or crosstalk) and flexibility (e.g., varying the error probability depending on the qubits and quantum operations), would allow for a more accurate approximation of real quantum computers.

ACKNOWLEDGMENTS

This work received funding from the University of Applied Sciences PhD program of the State of Upper Austria (managed by the FFG), from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation

Table IV
OPTIMIZATION FOR STOCHASTIC SIMULATION

Benchmark	#Q/#G	Det. Time.	Stoch. Time	Det. CPU	Stoch. CPU	Det. Memory	Stoch. Memory
error_correctiond	3/114	**<1 s**	5 s	97 %	9174 %	101 mb	3348 mb
cat_state	4/4	**<1 s**	4 s	108 %	821 %	63 mb	433 mb
basis_trotter	4/1625	**1 s**	16 s	107 %	9239 %	165 mb	17236 mb
vqe_uccsd	6/2281	486 s	**54 s**	107 %	9246 %	513 mb	76386 mb
qpe	9/150	392 s	**5 s**	97 %	6684 %	108 mb	1089 mb
QFT	9/192	241 s	**4 s**	107 %	9145 %	297 mb	1323 mb
ising	10/479	>1800 s	**122 s**	107 %	9246 %	417 mb	76641 mb
seca	11/281	>1800 s	**457 s**	107 %	9190 %	225 mb	1490 mb
cc	12/48	>1800 s	**189 s**	108 %	9251 %	117 mb	6060 mb
multipler	15/573	>1800 s	**335 s**	107 %	6941 %	609 mb	1418 mb
entanglement	16/16	**<1 s**	8 s	117 %	6612 %	97 mb	1223 mb
bigadder	18/284	>1800 s	**90 s**	107 %	8245 %	66 mb	1887 mb
entanglement	27/27	271 s	**7 s**	108 %	9051 %	1918 mb	1986 mb

program (grant agreement No. 101001318), was part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and has been supported by the BMWK on the basis of a decision by the German Bundestag through project QuaST.

REFERENCES

[1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Jour. of Comp.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Symp. on Theory of Computing*, 1996, pp. 212–219.

[3] D. Riste *et al.*, “Demonstration of quantum advantage in machine learning,” *npj Quantum Information*, vol. 3, no. 1, pp. 1–5, 2017.

[4] Y. Cao *et al.*, “Quantum chemistry in the age of quantum computing,” *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[5] G. Vidal, “Efficient classical simulation of slightly entangled quantum computations,” *Physical review letters*, vol. 91, no. 14, 2003.

[6] A. Zulehner and R. Wille, “Advanced simulation of quantum computations,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 848–859, 2019.

[7] D. M. Miller, M. A. Thornton, and D. Goodman, “A decision diagram package for reversible and quantum circuit simulation,” in *IEEE World Congress on Computational Intelligence*, 2006, pp. 8597–8604.

[8] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, “QMDDs: Efficient quantum function representation and manipulation,” vol. 35, no. 1, pp. 86–99, 2016.

[9] D. Steiger, T. Häner, and M. Troyer, “ProjectQ: An open source software framework for quantum computing,” *Quantum*, vol. 2, 2018.

[10] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, 2018.

[11] *Forest SDK*, https://pyquil-docs.rigetti.com/en/stable/index.html, Accessed: 2020-07-22, 2020.

[12] H. Abraham *et al.*, *Qiskit: An open-source framework for quantum computing*, 2019.

[13] Atos SE, *Quantum learning machine*, https : / / atos . net / en / products / quantum - learning - machineatos.net/en/products/quantum-learning-machine, Accessed: 2019-11-20, 2016.

[14] N. Khammassi, I. Ashraf, X. Fu, C. Almudever, and K. Bertels, “QX: A high-performance quantum computer simulation platform,” in *Design, Automation and Test in Europe*, 2017.

[15] D. Wecker and K. Svore, “LIQUi|>: A software design architecture and domain-specific language for quantum computing,” *arXiv:1402.4467*, 2014.

[16] *Cirq: A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits.* https : / / github . com / quantumlib / Cirqgithub.com/quantumlib/Cirq, Accessed: 2020-01-22, 2019.

[17] T. Jones, A. Brown, I. Bush, and S. Benjamin, “QuEST and high performance simulation of quantum computers,” *arXiv:1802.08032*, 2018.

[18] M. Smelyanskiy, N. P. D. Sawaya, and A. Aspuru-Guzik, “qHiPSTER: The quantum high performance software testing environment,” *Computing Research Repository*, 2016.

[19] B. Villalonga, S. Boixo, B. Nelson, *et al.*, “A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware,” *npj Quantum Information*, vol. 5, no. 1, 2019.

[20] G. G. Guerreschi, J. Hogaboam, F. Baruffa, and N. P. D. Sawaya, “Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits,” *Quantum Sci. Technol.*, vol. 5, 2020.

[21] T. Grurl, J. Fuß, S. Hillmich, L. Burgholzer, and R. Wille, “Arrays vs. decision diagrams: A case study on quantum circuit simulators,” in *Int’l Symp. on Multi-Valued Logic*, vol. 50, 2020, pp. 176–181.

[22] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo, “An XQDD-based verification method for quantum circuits,” *IEICE Trans. Fundamentals*, vol. 91-A, no. 2, pp. 584–594, 2008.

[23] A. Abdollahi and M. Pedram, “Analysis and synthesis of quantum circuits by using quantum decision diagrams,” in *Design, Automation and Test in Europe*, European Design and Automation Association, 2006, pp. 317–322.

[24] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, “QMDDs: Efficient quantum function representation and manipulation,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 86–99, 2016.

[25] V. Samoladas, “Improved BDD algorithms for the simulation of quantum circuits,” in *European Symp. on Algorithms*, 2008, pp. 720–731.

[26] G. Viamontes, I. Markov, and J. Hayes, “High-performance QuIDD-based simulation of quantum circuits,” in *Design, Automation and Test in Europe*, 2004, pp. 1354–1355.

[27] G. F. Viamontes, I. L. Markov, and J. P. Hayes, “Graph-based simulation of quantum computation in the density matrix representation,” *Quantum Info. Comput.*, vol. 5, no. 2, pp. 113–130, 2005.

[28] T. Grurl, J. Fuß, and R. Wille, “Considering decoherence errors in the simulation of quantum circuits using decision diagrams,” in *Int’l Conf. on CAD*, 2020, pp. 1–7.

[29] T. Grurl, R. Kueng, J. Fuß, and R. Wille, “Stochastic Quantum Circuit Simulation Using Decision Diagrams,” in *Design, Automation and Test in Europe*, 2021.

[30] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[31] J. Watrous, *The Theory of Quantum Information*. Cambridge University Press, 2018.

[32] S. Tannu and M. Qureshi, “Not All Qubits Are Created Equal,” *arXiv:1805.10224*, 2018.

[33] Gambetta, Jay and Sheldon, Sarah, *Cramming more power into a quantum device*, https://www.ibm.com/blogs / research / 2019 / 03 / power - quantum - device/, Accessed: 2020-09-10, 2019.

[34] M. Devoret and R. Schoelkopf, “Superconducting Circuits for Quantum Information: An Outlook,” *Science*, vol. 339, pp. 1169–1174, 2013.

[35] J. Kelly, *A Preview of Bristlecone, Google’s New Quantum Processor*, Accessed: 2019-05-19, 2018.

[36] R. P. Feynman, A. R. Hibbs, and D. F. Styer, *Quantum mechanics and path integrals*. Courier Corporation, 2010.

[37] S. Hillmich, R. Kueng, I. L. Markov, and R. Wille, “As accurate as needed, as efficient as possible: Approximations in DD-based quantum circuit simulation,” in *Design, Automation and Test in Europe*, 2021.

[38] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, pp. 117–158, 2014.

[39] I. L. Markov and Y. Shi, “Simulating quantum computation by contracting tensor networks,” *SIAM J. Comput.*, vol. 38, no. 3, pp. 963–981, 2008.

[40] J. D. Biamonte and V. Bergholm, “Tensor networks in a nutshell,” 2017. arXiv: 1708.00006.

[41] S. Boixo *et al.*, “Characterizing quantum supremacy in near-term devices,” *Nature Physics*, vol. 14, no. 6, p. 595, 2018.

[42] T. Grurl, J. Fuß, and R. Wille, “Lessons learnt in the implementation of quantum circuit simulation using decision diagrams,” in *Int’l Symp. on Multi-Valued Logic*, 2021, pp. 87–92.

[43] S. Hillmich, A. Zulehner, and R. Wille, “Concurrency in DD-based quantum circuit simulation,” in *Asia and South Pacific Design Automation Conf.*, 2020.

[44] A. Zulehner and R. Wille, “Matrix-vector vs. matrix-matrix multiplication: Potential in DD-based simulation of quantum computations,” in *Design, Automation and Test in Europe*, 2019, pp. 90–95.

[45] A. Zulehner, S. Hillmich, and R. Wille, “How to efficiently handle complex values? Implementing decision diagrams for quantum computing,” in *Int’l Conf. on CAD*, 2019.

[46] A. Li and S. Krishnamoorthy, “QASMBench: A Low-level QASM Benchmark Suite for NISQ Evaluation and Simulation,” 2020. arXiv: 2005.13018.

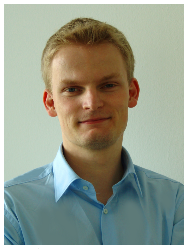
[47] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, “Open quantum assembly language,” *arXiv preprint arXiv:1707.03429*, 2017.

[48] D. Merkel, “Docker: Lightweight Linux containers for consistent development and deployment,” *Linux Jour.*, vol. 2014, no. 239, 2014.

[49] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, “An updated performance comparison of virtual machines and linux containers,” in *2015 IEEE International Symposium on Performance Analysis of Systems and Software*, 2015, pp. 171–172.

Thomas Grurl received his BSc and MSc degree in Secure Information Systems at the University of Applied Sciences Upper Austria in 2015 and 2017, respectively. Since 2016, he is working with the research group of Secure Information Systems at the University of Applied Sciences Upper Austria as a research assistant. 2019 he started with his Ph.D. program at the Institute for Integrated Circuits at the Johannes Kepler University Linz, Austria under the supervision of Robert Wille and Jürgen Fuß. His research interests include information security, quantum computing, and quantum circuit simulation—currently focusing on noise-aware quantum circuit simulation. In these areas, he has published several papers on international conferences such as DATE, ISMVL, and ICCAD.

Jürgen Fuß studied Mathematics in Linz, Austria, and Grenoble, France, and received his Diploma and Dr. techn. degrees in Mathematics from Johannes Kepler University Linz, Austria, in 1995 and 2001, respectively. He was with the near-ring research group at the University of Linz as a researcher from 1995 to 2003. Since 2003, he is full professor with the Department of Secure Information Systems at FH Upper Austria University in Hagenberg, Austria. His current research interests include cryptography, high-performance and quantum computing. In these fields he has served in the Program Committees of numerous journals and conferences and published in various journals including the IEEE Transactions on Mobile Computing (TMC), the Journal on Emerging Technologies in Computing Systems (JETC), the Conference on Computer and Communications Security (CCS), and the Privacy Enhancing Technologies Symposium (PETS).



Robert Wille is a Full and Distinguished Professor at the Technical University of Munich, Germany, and Chief Scientific Officer at the Software Competence Center Hagenberg, Austria. He received the Diploma and Dr.-Ing. degrees in Computer Science from the University of Bremen, Germany, in 2006 and 2009, respectively. Since then, he worked at the University of Bremen, the German Research Center for Artificial Intelligence (DFKI), the University of Applied Science of Bremen, the University of Potsdam, and the Technical University Dresden. From 2015 until 2022, he was Full Professor at the Johannes Kepler University Linz, Austria, until he moved to Munich. His research interests are in the design of circuits and systems for both conventional and emerging technologies. In these areas, he published more than 350 papers and served in editorial boards as well as program committees of numerous journals/conferences such as TCAD, ASP-DAC, DAC, DATE, and ICCAD. For his research, he was awarded, e.g., with Best Paper Awards, e.g., at TCAD and ICCAD, an ERC Consolidator Grant, a Distinguished and a Lighthouse Professor appointment, a Google Research Award, and more.