

Hybrid Schrödinger-Feynman Simulation of Quantum Circuits With Decision Diagrams

Lukas Burgholzer*

Hartwig Bauer*

Robert Wille*[†]

*Institute for Integrated Circuits, Johannes Kepler University Linz, Austria

[†]Software Competence Center Hagenberg GmbH (SCCH), Austria

lukas.burgholzer@jku.at

hartwig.bauer@jku.at

robert.wille@jku.at

<https://iic.jku.at/eda/research/quantum/>

Abstract—Classical simulations of quantum computations are vital for the future development of this emerging technology. To this end, decision diagrams have been proposed as a complementary technique which frequently allows to tackle the inherent exponential complexity of these simulations. In the worst case, however, they still cannot escape this complexity. Additionally, while other techniques make use of all the available processing power, decision diagram-based simulation to date cannot exploit the many processing units of today’s systems. In this work, we show that both problems can be tackled together by employing a hybrid Schrödinger-Feynman scheme for the simulation. More precisely, we show that realizing such a scheme with decision diagrams is indeed possible, we discuss the resulting problems in its realization, and propose solutions how they can be handled. Experimental evaluations confirm that this significantly advances the state of the art in decision diagram-based simulation—allowing to simulate certain hard circuits within minutes that could not be simulated in a whole day thus far.

Index Terms—quantum computing, classical simulation, decision diagrams, hybrid Schrödinger-Feynman

I. INTRODUCTION

Despite actual quantum computers being available in the cloud, the simulation of quantum circuits on classical machines remains crucial for the development and design of future quantum computing applications. Such simulations provide insights into the inner workings of a quantum system and allow, e.g., to analyze quantum algorithms or verify the output of physical quantum computers. To this end, several notions of what such a classical simulation entails exist. In this work, we consider *strong* simulation of quantum circuits, i.e., we want to compute *all* complex amplitudes of the quantum state resulting from the execution of the circuit (as, e.g., in [1]–[15]). This is in contrast to methods based, e.g., on tensor networks that typically only compute individual amplitudes or a small batch of them [16]–[22]. Furthermore, we do not consider *approximate* simulation, which trades computational complexity for accuracy of the result (as, e.g., in [11], [13]).

The most fundamental technique of simulating a quantum circuit is called *Schrödinger-style* simulation. There, a complete representation of the quantum system’s state is stored and manipulated throughout the computation. While straight-forward in principle, this quickly amounts to a complex task, due to the underlying representation requiring the storage and manipulation of 2^n complex amplitudes for an n -qubit system.

While this complexity is frequently tackled by massively parallel computations on *arrays* [1]–[5] using supercomputer clusters with immense amounts of memory and processing power, *decision diagrams* [23]–[26] have been proposed as a complementary technique that aims at compactly representing and efficiently manipulating the 2^n complex amplitudes and, hence, often allows to conduct corresponding simulations on just a single desktop computer [6]–[11]. But, in the worst case, also decision diagrams are subject to the inherent exponential complexity.

Complementary to that, there exists another formalism suited for simulating quantum circuits which reduces the memory complexity of the simulation by breaking it down into simpler parts. This is known as *Feynman-style* path summation [12], [27]. From a high-level point of view, each gate connecting two or more qubits in a quantum circuit introduces a decision point from which the simulation branches (this notion will be made more precise later in Section III-B). As the name implies, *Feynman-style* path summation calculates the result of each path and sums up all the individual contributions—resulting in the final quantum state. Since the number of paths is exponential depending on the number of decision points, this approach requires exponential runtime (but usually avoids too harsh memory requirements).

Over the recent years, mixtures of both schemes emerged [12]–[15]—often referred to as *hybrid Schrödinger-Feynman simulation*. These approaches strive to use all the available memory and processing units in order to efficiently simulate quantum circuits which would (1) run into memory bottlenecks using Schrödinger-style simulation, or (2) take exceedingly long using Feynman-style path summation—eventually trading-off the respective memory and runtime requirements. However, while this hybrid scheme can easily be realized, e.g., using arrays, no solution for decision diagrams exists yet (in fact, there are even discussions that a corresponding realization might not be possible at all [26]). This constitutes a severe drawback as it keeps decision diagram-based simulation stuck with Schrödinger-style simulation that is only suitable if the compact representations of decision diagrams allow to escape the exponential memory requirements (while, in all remaining cases, decision diagrams even impose a severe overhead compared to rather simple arrays).

In this work, we show that realizing a *hybrid Schrödinger-Feynman* scheme with decision diagrams is indeed possible—even if some problems arise when doing so. We describe a possible realization, discuss what problems exactly arise, and propose solutions to overcome them. Eventually, the first hybrid Schrödinger-Feynman quantum circuit simulation approach results which works with decision diagrams. Experimental evaluations confirm that this significantly advances the state of the art in decision diagram-based simulation—allowing to simulate certain hard circuits within minutes that could not be simulated in a whole day thus far. An implementation of the proposed simulation technique is publicly available at <https://github.com/iic-jku/ddsim>.

The remainder of this paper is structured as follows. Section II reviews the basics of (decision diagram-based) quantum circuit simulation. Then, Section III describes the general idea of the hybrid Schrödinger-Feynman technique. Section IV describes the realization of such techniques for decision diagrams, the resulting limitations, and how they can be handled. Afterwards, Section V shows the experimental results before the paper is concluded in Section VI.

II. BACKGROUND

To keep this paper self-contained, this section briefly reviews the basics of quantum circuits and their simulation as well as how this simulation can be conducted using decision diagrams. We refer the interested reader to [6], [28] for a more detailed overview of either topic. Furthermore, an online visualization tool, which makes decision diagrams for quantum computing more accessible [29], is available at https://iic.jku.at/eda/research/quantum_dd/tool/.

A. Quantum Circuits and Their Simulation

In general, the state $|\varphi\rangle$ of an n -qubit quantum system is described as a linear combination of 2^n basis states, i.e.,

$$|\varphi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \text{ with } \alpha_x \in \mathbb{C} \text{ and } \sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1.$$

This is frequently represented as $[\alpha_{0\dots 0}, \dots, \alpha_{1\dots 1}]^\top$ and referred to as *state vector*. Measuring the state of a quantum system probabilistically collapses the system's state to one of the basis states—each with probability $|\alpha_x|^2$ for $x \in \{0,1\}^n$.

Example 1. Consider a quantum system consisting of four qubits that resides in the all-zero initial state $|0000\rangle$. The system's state vector consists of $2^4 = 16$ complex amplitudes and is represented by

$$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^\top.$$

The state of a quantum system is manipulated by quantum operations (frequently called *quantum gates*). Any k -qubit quantum gate can be identified with a unitary matrix U of size $2^k \times 2^k$, which can be extended to the whole system size by forming tensor products with identity matrices. A quantum circuit describes a sequence of gates, that are applied to a quantum system. Applying a quantum gate U to the state $|\varphi\rangle$

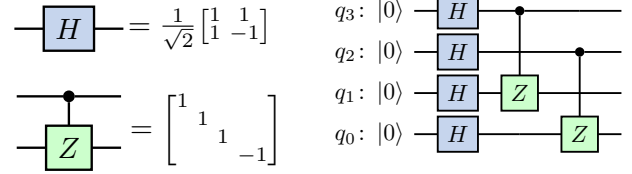


Fig. 1. Quantum gates and quantum circuit G

of a quantum system corresponds to extending its matrix representation to the system size and computing the matrix-vector product of the resulting matrix with the state vector—yielding a new state vector $|\varphi'\rangle$. Thus, simulating a quantum circuit entails the successive application of all individual gates to the initial state of a quantum system in order to obtain the final state (vector). An example illustrates the idea:

Example 2. An example of a single-qubit gate (the Hadamard) and a two-qubit gate (the controlled-Z operation) is shown on the left-hand side of Fig. 1 with their corresponding matrix representations. The right-hand side of Fig. 1 shows a quantum circuit G acting on four qubits that uses these gates. Simulating this circuit with the input state $|0000\rangle$ (see Example 1) results in the final state vector:

$$\frac{1}{4} * [1, 1, 1, 1, 1, -1, 1, -1, 1, 1, -1, -1, 1, -1, -1, 1]^\top.$$

B. Decision Diagram-based Simulation

The size of the state vector is inherently exponential with respect to the number of qubits. Not only does this always incur an exponential memory footprint, but it also implies that all operations have to be conducted on an exponentially large data structure.

Decision diagrams [23]–[26] have been proposed as a complementary approach for efficiently representing and manipulating the state of a quantum system by exploiting redundancies in the underlying representation. They represent quantum states and operations as weighted, directed, acyclic graphs. To this end, a given state vector with its complex amplitudes α_i for $i \in \{0,1\}^n$ is decomposed into sub-vectors

$$\begin{array}{cccc} [\alpha_{00\dots 0}, \dots, \alpha_{1\dots 1}]^\top & & & \\ [\alpha_{0x}]^\top & & & [\alpha_{1x}]^\top \\ [\alpha_{00y}]^\top & [\alpha_{01y}]^\top & [\alpha_{10y}]^\top & [\alpha_{11y}]^\top \end{array}$$

with $x \in \{0,1\}^{n-1}$ and $y \in \{0,1\}^{n-2}$, until only individual amplitudes remain. The resulting decision diagram has n levels of nodes (labeled $n-1$ to 0) where each node i has exactly two successors—indicating whether the corresponding path leads to an amplitude where qubit q_i is in the state $|0\rangle$ or $|1\rangle$. During these decompositions, common factors are extracted as edge weights according to a normalization scheme and equivalent sub-vectors are represented by the same node—allowing to exploit potential redundancies in the representation. The amplitude of a given basis state can then be reconstructed from the multiplication of the edge weights along the path from the root node to the terminal node.

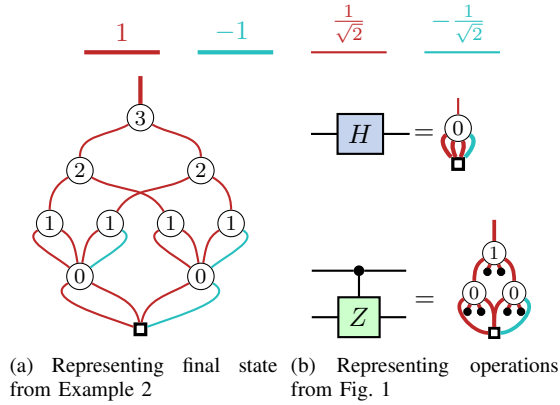


Fig. 2. Decision diagrams

Example 3. Consider again the final quantum state from Example 2. The corresponding decision diagram is shown in Fig. 2a. The amplitude of the state $|1010\rangle$, for example, is obtained by multiplying the root weight with the weights along the path alternating between the right and the left successor, i.e., by multiplying $1 * \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * -\frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} = -\frac{1}{4}$.

A similar decomposition can be employed for the matrices representing quantum gates by recursively splitting the respective matrix into four equally-sized sub-matrices according to the operator basis

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right\} = \{|0\rangle\langle 0|, |0\rangle\langle 1|, |1\rangle\langle 0|, |1\rangle\langle 1|\}.$$

Example 4. Consider again the operations shown on the left-hand side of Fig. 1. The corresponding decision diagrams are shown in Fig. 2b.

As described above, applying a gate to a quantum system entails the matrix-vector multiplication of the corresponding matrix with the current state vector. This operation can be recursively broken down according to

$$\begin{bmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{bmatrix} \cdot \begin{bmatrix} \alpha_{0x} \\ \alpha_{1x} \end{bmatrix} = \begin{bmatrix} (U_{00} \cdot \alpha_{0x} + U_{01} \cdot \alpha_{1x}) \\ (U_{10} \cdot \alpha_{0x} + U_{11} \cdot \alpha_{1x}) \end{bmatrix},$$

with $U_{ij} \in \mathbb{C}^{2^{n-1} \times 2^{n-1}}$ and $\alpha_{ix} \in \mathbb{C}^{2^{n-1}}$ for $i, j \in \{0, 1\}$. Since the U_{ij} and α_{ix} directly correspond to the successors in the respective decision diagrams, matrix-vector multiplication is a native operation on decision diagrams and its complexity scales with the product of the number of nodes of both decision diagrams. Thus, whenever the decision diagrams remain compact throughout the computation, the simulation of quantum circuits can be efficiently conducted using decision diagrams [6]–[11].

III. MOTIVATION AND GENERAL IDEA

Decision diagrams offer a complementary approach for the simulation of quantum circuits. In many cases, they have been shown to compactly represent and efficiently manipulate the state of a quantum system. However, there remain some obstacles and limitations of decision diagram-based simulation.

In the following, we will discuss these limitations and provide the general idea for overcoming them.

A. Limitations of Decision Diagram-based Simulation

While decision diagrams frequently allow to compactly represent the state of a quantum system, in the worst case, their size is still exponential with respect to the number of qubits. Such situations arise when no redundancy in the description of the quantum state can be exploited and, thus, only a few nodes can be shared. This occurs, e.g., during the simulation of quantum circuits whose gates are chosen randomly according to some scheme (e.g., the circuits used by Google in their quantum supremacy experiment [18], [30]). In general, such circuits are designed to make classical simulations as hard as possible, which—in case of decision diagrams—implies that they try hard to not give rise to a particular structure in the corresponding states.

At the same time, decision diagram operations such as matrix-vector multiplication, addition, inner product computation, or sampling, scale polynomially with the number of nodes of the involved decision diagrams. As such, they are highly efficient whenever the underlying decision diagrams actually emit a compact structure. However, if the number of nodes in the decision diagram grows exponentially (which happens in the worst case), their performance degrades significantly. Even worse: In this regime, decision diagrams actually perform worse than, e.g., array-based techniques, which always incur this exponential (memory) complexity, but have a lower overhead of maintaining the underlying data structure.

In one way or another, all Schrödinger-style methods (such as arrays, tensor trains, decision diagrams, etc.) face the problem of exponentially increasing simulation complexity. Many established simulation methods compensate for this limitation by heavily employing parallelization, i.e., making use of the many cores in today’s systems or even large clusters of supercomputers to speed up the computation [1]–[5]. While similar efforts have been conducted towards parallelizing decision diagram-based simulation, e.g., in [31], no “break-through” has been achieved there yet. The main obstacles in this regard are that the shared nature of decision diagrams necessitates inter-process-communication and some kind of locked access to its central data members (such as unique or compute tables) which, in turn, quickly eliminate the benefits of parallelization.

In this work, we entertain an approach that drastically reduces the exponential simulation complexity for certain classes of problems (specifically, depth-limited circuits) at the expense of simulating multiple, independent instances whose contributions are eventually accumulated. Coincidentally, this further allows to fully utilize all available processing power—effectively “killing both birds with one stone”. The proposed approach follows the concepts of a *hybrid Schrödinger-Feynman technique*, which is reviewed next before the general idea of applying this concept to decision diagrams is described.

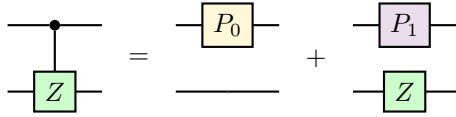


Fig. 3. Schmidt Decomposition of the controlled-Z Gate

B. Hybrid Schrödinger-Feynman Simulation

The hybrid Schrödinger-Feynman simulation style aims at reducing the complexity of the Schrödinger-style simulation by breaking it down into simpler parts. This is accomplished by employing concepts from Feynman-style path summation. To this end, the most important concept is the Schmidt decomposition of a two-qubit gate: Any two-qubit gate (represented by a unitary 4×4 matrix U) can be decomposed into at most four tensor products according to

$$U = |0\rangle\langle 0| \otimes U_{00} + |0\rangle\langle 1| \otimes U_{01} + |1\rangle\langle 0| \otimes U_{10} + |1\rangle\langle 1| \otimes U_{11},$$

with unitary $U_{ij} \in \mathbb{C}^{2 \times 2}$ and $i, j = 0, 1$.

Example 5. Consider the controlled-Z gate whose unitary matrix representation is given by $U = \text{diag}(1, 1, 1, -1)$. Intuitively, the operation leaves both qubits untouched whenever the control qubit is in state $|0\rangle$, while it applies a Z gate to the target qubit in case the control qubit is in state $|1\rangle$. In formulas:

$$|0\rangle \otimes |x\rangle \mapsto |0\rangle \otimes \mathbb{I} |x\rangle \quad \text{and} \quad |1\rangle \otimes |x\rangle \mapsto |1\rangle \otimes Z |x\rangle \quad \text{for } x = 0, 1.$$

Consequently, its Schmidt decomposition consists of two terms and is given by

$$U = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes Z = P_0 \otimes \mathbb{I} + P_1 \otimes Z,$$

with P_0 (P_1) denoting the projection onto $|0\rangle$ ($|1\rangle$). This is illustrated in Fig. 3.

The Schmidt decomposition allows to split the application of any two-qubit gate into separate parts that can be calculated independently. As such, each decomposed gate increases the number of simulations (i.e., the runtime) by the amount of factors in its decomposition. After all individual contributions have been computed, they have to be summed up in order to obtain the full result.

Hybrid Schrödinger-Feynman approaches (such as [12]–[15]) horizontally partition the whole circuit into blocks by introducing *cuts* through the circuit’s list of qubits. Only cross-block gates, i.e. gates acting across such a cut, are decomposed according to their Schmidt decomposition. This way, individual blocks are independent from each other and, thus, can be simulated separately. Then, the total number of necessary simulations depends on the number of cross-block gates. As this dependence is exponential in the number of gates (e.g., doubling on each cross-block controlled-Z gate), such techniques are most efficient for depth-limited circuits. However, this still constitutes a large class of quantum algorithms—especially those targeted at near-term quantum computers, which are inherently depth-limited due to noise.

C. General Idea

While decision diagrams offer a complementary approach to quantum circuit simulation that (exponentially) outperforms, e.g., array-based techniques, whenever the number of nodes only grows polynomially, their performance is significantly worse for “hard” instances (where almost no redundancy can be exploited). The Schmidt decomposition, as introduced above, allows to reduce the complexity of individual simulations by splitting the circuit into independent blocks that are significantly easier to simulate at the expense of runtime. This concept can readily be applied to decision diagrams.

Example 6. Recall the controlled-Z gate and its decision diagram (shown in Fig. 2). As reviewed in Section II-B, the successors of a (matrix) decision diagram node encode its action according to the operator basis $\{P_0, |0\rangle\langle 1|, |1\rangle\langle 0|, P_1\}$. In case of the controlled-Z gate, the left-most successor (corresponding to P_0) leads to a node representing the identity operation while the right-most successor (corresponding to P_1) leads to a node representing the Z operation. Splitting these contributions into individual decision diagrams yields the decomposition shown in Fig. 4, which precisely resembles the Schmidt decomposition of the controlled-Z gate from Example 5 (illustrated in Fig. 3).

Overall, these techniques promise to overcome both obstacles raised above: By drastically reducing the complexity of individual simulations, the efficiency of decision diagrams can be fully exploited. Additionally, no inter-process-communication or locked access is necessary when performing the simulations in parallel since they are independent of another.

Yet, hybrid Schrödinger-Feynman approaches have not been applied to decision diagrams. Some even believe that realizing such “circuit cutting techniques” with decision diagrams (as they are proposed in this work) is not possible at all [26]¹. In the remainder of this paper we show that (1) this is indeed possible, (2) which problems arise in the realization, and (3) how they can be handled.

IV. DECISION DIAGRAM-BASED SCHRÖDINGER-FEYNMAN SIMULATION

Following the general idea outlined above potentially allows to overcome the limitations of decision diagram-based quantum circuit simulation discussed in Section III-A. In this section, we describe the realization of a hybrid Schrödinger-Feynman technique for decision diagrams and discuss the main bottleneck of the resulting scheme. Afterwards, we show how this bottleneck can be addressed by relying on decision diagrams where they are most efficient, while leaving the rest to more suitable techniques.

¹In contrast to quantum circuit simulation as considered in this work, [26] seeks for a complete representation of a quantum circuit’s functionality. Both tasks are related as the functionality of a quantum circuit is obtained from consecutive matrix-matrix multiplication of the individual gate descriptions. Consequently, the results in this work are also applicable in the scenario discussed in [26].

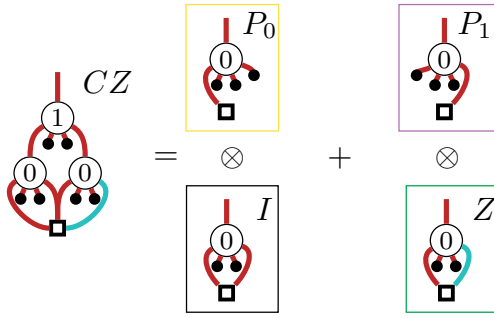


Fig. 4. Schmidt decomposition of the controlled-Z gate decision diagram

A. Realization

In order to employ a hybrid Schrödinger-Feynman approach, the circuit first has to be *partitioned* into blocks as reviewed in Section III-B. In general, there is a large degree of freedom for how to choose such a partitioning, e.g., the number of gates acting across individual blocks (*cross-block gates*). In the following, we consider splitting the circuit into two (almost) equally-sized blocks—ensuring that each block approximately has the same number of qubits. Then, the number of simulations to be performed depends on the number of cross-block gates that need to be decomposed (according to their Schmidt decomposition). Each individual simulation can be assigned a unique identifier that specifies the *decision* (i.e., part of the Schmidt decomposition) to make at each *decision point* (i.e., cross-block gate).

Example 7. Consider again the circuit shown in Fig. 1 and assume it is partitioned into two equally-sized halves. Then, both controlled-Z gates act across the blocks and, hence, need to be decomposed. This yields two decision points with two choices each (the parts of the controlled-Z gate’s Schmidt decomposition)—for a total of four individual parts to be simulated as illustrated on the left-hand side of Fig. 5. To this end, the first (second) term of the Schmidt decomposition is encoded as 0 (1). Therefore, each simulation can be assigned a bitstring of length two (i.e., number of decisions) that indicates which term of the decomposition is to be calculated.

Splitting the circuit in this fashion has three major benefits. Most importantly, the blocks in each individual simulation are independent from another and, thus, can be simulated separately. This reduces the complexity for each simulation from simulating one n -qubit circuit to simulating two $n/2$ -qubit circuits—an exponential improvement since the state of an n -qubit system grows as 2^n . Furthermore, as shown in Fig. 4, the decision diagrams for individual parts of a gate’s Schmidt decomposition are typically much less complex than the full decision diagram—allowing for more compact decision diagrams throughout the simulations. Finally, all individual simulations are independent from another and, hence, can be more efficiently simulated in parallel—even with decision diagrams.

Example 8. Consider again the scenario from Example 7. Then, each of the four individual simulations requires the simulation of two two-qubit circuits. The resulting decision diagrams are shown in the middle of Fig. 5. A maximum of two nodes (the best case for two-qubit decision diagrams) is required during each individual simulation. In contrast, the Schrödinger-style simulation (see Example 3) required the handling of up to nine nodes.

Although this only represents a small example, it already shows the potential that the hybrid Schrödinger-Feynman technique brings to decision diagram-based simulation. However, the computation is not yet finished. As described in Section III-B, the final result of the simulation is obtained as the sum of all individual parts. First, the simulation results of each block have to be combined by forming the tensor product of the corresponding states. Forming the tensor product of two decision diagrams is highly efficient, as it merely requires replacing the terminal node of one decision diagram with the root node of the other.

Example 9. Forming pairwise tensor products of the decision diagrams resulting from the simulations of the circuit from Example 8 yields four decision diagrams of size four each. These decision diagrams (shown in the middle of Fig. 5) represent the state vectors of the individual simulations.

Finally, all the decision diagrams need to be added up to obtain the decision diagram representing the final state vector. As for the simulations, these additions can be computed in parallel without inter-process-communication using a tree-like scheme whose depth corresponds to the logarithm of the number of individual simulations.

Example 10. The right-hand side of Fig. 5 illustrates the process of adding up the individual contributions in order to obtain the final state vector. At the first addition level, two decision diagrams of size six result, while after the final addition, the nine-node decision diagram already seen in the right part of Fig. 2 results.

B. Decision Diagram Addition as a Bottleneck

The combination of all the individual results, i.e., the addition of all resulting decision diagrams, inevitably builds up a potential bottleneck. While the complexity of the decision diagrams throughout the individual simulations might be drastically lower than the complexity of the full Schrödinger-style simulation, the final decision diagram obviously remains the same. Consequently, whenever the final decision diagram grows exponentially, this complexity builds up somewhere along the way of adding up the individual contributions. Since addition of decision diagrams scales linearly with respect to the number of nodes of both decision diagrams, subsequent additions in the “addition hierarchy” take longer and longer.

At some point, this constitutes a severe bottleneck for the hybrid Schrödinger-Feynman simulation using decision diagrams because the overhead of maintaining the data structure becomes larger than the benefit gained from a potentially more

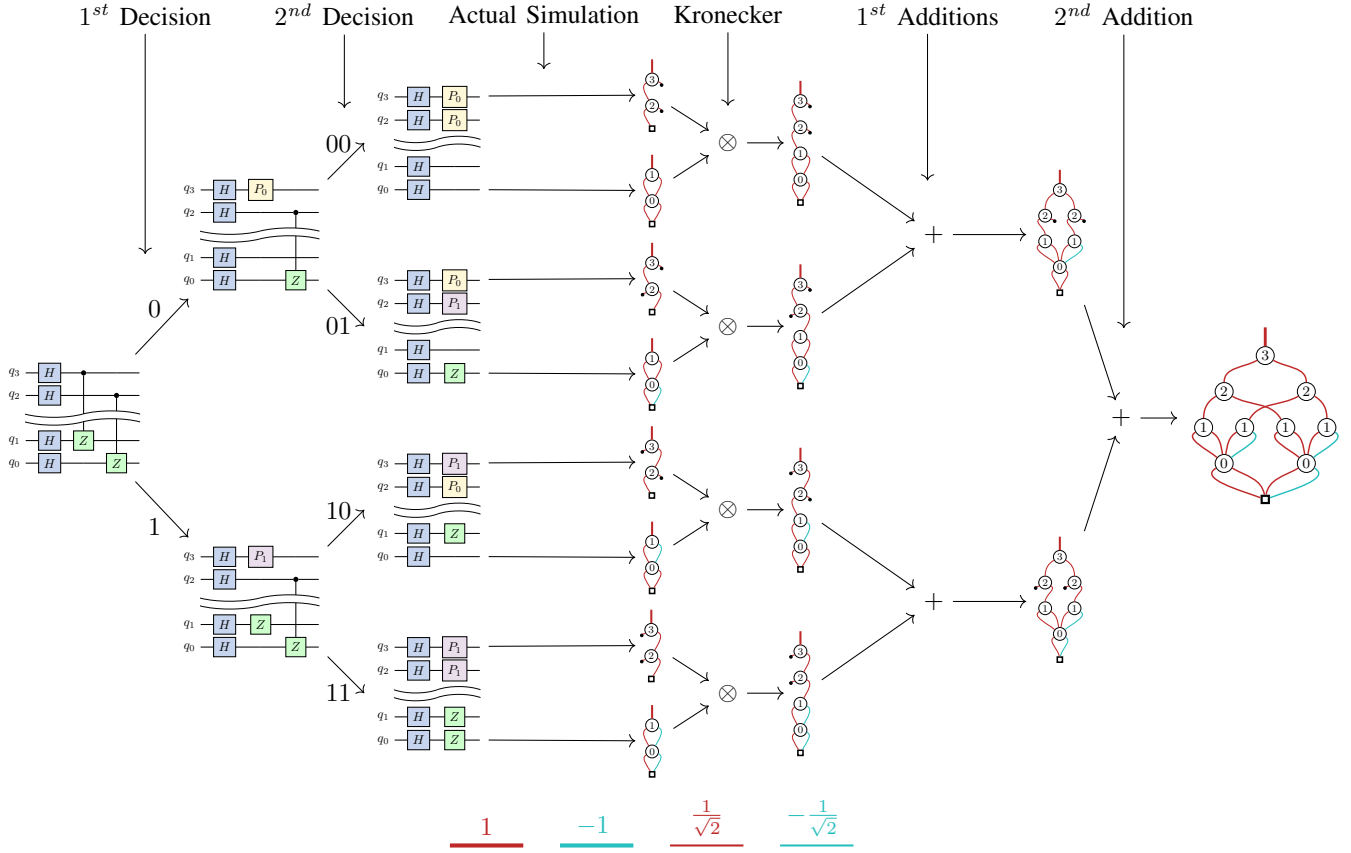


Fig. 5. Hybrid Schrödinger-Feynman simulation of the circuit shown in Fig. 1.

compact representation. As confirmed by our experimental evaluations (which are summarized later in Section V), decision diagrams are highly efficient when it comes to the first part of the hybrid Schrödinger-Feynman scheme (i.e., the simulation of individual parts constituting the overall result), while it might get more challenging in the second part of the computation (i.e., adding up all the individual contributions). In the following, we show how this bottleneck can be addressed whenever the final decision diagram is going to be exponentially large.

C. Avoiding the Final Overhead

Thus far, we used decision diagrams for the first part of the hybrid Schrödinger-Feynman scheme because of their efficiency in simulation. In the second part (when the final results are determined by addition), this benefit might disappear and lead to exponentially large decision diagrams. At this point, decision diagrams do not offer any advantages anymore compared to simpler data-structures such as arrays (in fact, the overhead caused by maintaining a dedicated data-structure will make decision diagrams perform even worse than arrays, which require practically no overhead). That is, in these cases, one is better off extracting the state vector represented by the decision diagram into an array and continue working with that representation. That way, one relies on decision diagrams

where they are most efficient, while substituting more direct representations whenever the limit for decision diagrams has been reached.

To this end, the complete vector represented by a decision diagram is extracted with a single recursive traversal of the decision diagram by accumulating amplitude contributions along the edges. After the extraction, all resulting arrays can be added together to obtain the final state vector. As a consequence, one benefits from the memory locality of array-based representations as well as vectorized instruction support of modern CPUs—completely circumventing the overhead decision diagram-based addition incurs in this regime.

Example 11. Consider again the hybrid Schrödinger-Feynman scheme shown in Fig. 5 for simulating the circuit from Fig. 1. After all four individual simulations have been conducted, the amplitudes of the corresponding decision diagrams representing the state vectors are extracted. For the top-most decision diagram (corresponding to the “00-decision”) this results in the following (recursive) computation:

$$\begin{aligned}
 \frac{1}{2} * [[\dots] 0000 0000]^T &= \frac{1}{2} * [[\dots] 0000 0000 0000]^T \\
 &= \frac{1}{2} * \left[\left[\frac{1}{\sqrt{2}} * [\dots] \frac{1}{\sqrt{2}} * [\dots] \right] 0000 0000 0000 \right]^T \\
 &= \frac{1}{2\sqrt{2}} * \left[\left[\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \right] \left[\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \right] 0000 0000 0000 \right]^T \\
 &= \frac{1}{4} * [1111 0000 0000 0000]^T
 \end{aligned}$$

Overall, the extraction results in the respective amplitude arrays

$$\begin{aligned} & \frac{1}{4} * [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top, \\ & \frac{1}{4} * [0 \ 0 \ 0 \ 0 \ 1-1 \ 1-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top, \\ & \frac{1}{4} * [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1-1-1 \ 0 \ 0 \ 0 \ 0]^\top, \\ & \frac{1}{4} * [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1-1-1 \ 1]^\top, \end{aligned}$$

which are subsequently added up to produce the final state vector

$$\frac{1}{4} * [1 \ 1 \ 1 \ 1 \ 1-1 \ 1-1 \ 1 \ 1-1-1 \ 1-1-1 \ 1]^\top.$$

As our experimental evaluations (which are summarized next) confirm, employing decision diagrams for the individual simulations of the hybrid Schrödinger-Feynman scheme, while handing off the accumulation of individual results to computations on arrays, allows to capitalize on the best of both worlds and mitigates the limitations discussed in Section IV-A.

V. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of decision diagram-based hybrid Schrödinger-Feynman simulation, the scheme proposed in this work has been implemented on top of the state-of-the-art decision diagram-based simulator JKQ DDSIM, which is based on [6] and is part of the JKQ toolset for quantum computing [32]. The corresponding implementation is available at <https://github.com/iic-jku/ddsim>.

As benchmarks, we considered Google’s supremacy circuits [30] which constitute one of the hardest instances for quantum circuit simulation to date and, because they belong to the class of depth-limited circuits, are exactly the kind of circuits for which the proposed scheme is particularly suited for (see Section III-B). The evaluations were conducted on machine equipped with a 16-core AMD Ryzen 9 3950X CPU and 128 GB RAM running Ubuntu 20.04. *Double* precision floating points and a hard timeout of 24h were used for all simulations.

Table I shows the results of our evaluations. Here, the first two columns identify the benchmark circuit and the respective number of decisions (i.e., cross-block gates)². Then, the runtime of the JKQ DDSIM Schrödinger-style simulator is listed, while the remaining four columns contain the runtime and the speedup for the general decision diagram-based hybrid Schrödinger-Feynman scheme (see Section IV-A) and the optimized scheme using arrays for the final additions (see Section IV-C), respectively.

From the results, it can be seen that, the higher the number of qubits in the circuit, the higher the potential gain of employing the hybrid Schrödinger-Feynman scheme. While using the general scheme from Section IV-A only yields a small speedup for the smallest benchmarks (i.e. *inst_4x4_X_Y*), the

²The circuits chosen for this evaluation only use controlled-Z gates as two-qubit gates. Consequently, for a certain number of decisions x , 2^x individual simulations have to be performed in the hybrid Schrödinger-Feynman scheme.

medium-sized benchmarks (i.e. *inst_4x5_X_Y*) show an average speedup of $\approx 6.2\times$. However, neither the JKQ DDSIM Schrödinger-style simulator, nor the general scheme proposed in Section IV-A, were able to simulate the larger benchmarks (i.e. *inst_5x5_X_Y* and *inst_5x6_X_Y*) within 24h. As discussed in Section IV-B, this can be attributed to the fact that decision diagram addition on exponentially growing decision diagrams poses a severe bottleneck.

This problem is addressed by using decision diagrams for the individual simulations and resorting to arrays for the final additions. In fact, the numbers confirm that, then, speedups of several factors and up to several orders of magnitude can be observed across all benchmarks. More impressively, even the biggest circuits in our evaluations that could not be simulated in a whole day using the Schrödinger-style simulator can be simulated in roughly 20 min using this scheme.

VI. CONCLUSIONS

In this work, we showed that a hybrid Schrödinger-Feynman technique can be applied to decision diagram-based quantum circuit simulation, which, for the first time, allows to fully exploit the available hardware resources. Due to the substantially decreased complexity of the individual simulations, decision diagrams are employed in a regime where more redundancy can be exploited. By handing off the accumulation of individual results to computations on arrays, the bottleneck of decision diagram addition observed in this work can be effectively circumvented. The resulting scheme combines the best of both worlds and allows to significantly advance the state of the art in decision diagram-based quantum circuit simulation. An implementation of the proposed simulation technique is publicly available at <https://github.com/iic-jku/ddsim>.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 101001318). It has partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria as well as by the BMK, BMDW, and the State of Upper Austria in the frame of the COMET program (managed by the FFG).

REFERENCES

- [1] T. Häner and D. S. Steiger, “0.5 petabyte simulation of a 45-Qubit quantum circuit,” in *Int’l Conf. for High Performance Computing, Networking, Storage and Analysis*, 2017.
- [2] J. Doi, H. Takahashi, R. Raymond, T. Imamichi, and H. Horii, “Quantum computing simulator on a heterogenous HPC system,” in *Int’l Conf. on Computing Frontiers*, 2019, pp. 85–93.
- [3] T. Jones, A. Brown, I. Bush, and S. C. Benjamin, “QuEST and high performance simulation of quantum computers,” in *Scientific Reports*, 2018.
- [4] 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, p. 034007, 2020.
- [5] X.-C. Wu *et al.*, “Full-state quantum circuit simulation by using data compression,” in *Int’l Conf. for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–24.

TABLE I
EXPERIMENTAL EVALUATIONS

Benchmark		JKQ DDSIM [6]	DD-based (Section IV-A)		Amp. Adding (Section IV-C)	
Name	#Decisions	t_{ref} [s]	t_{DD} [s]	t_{ref}/t_{fullDD}	t_{amp} [s]	t_{ref}/t_{amp}
inst_4x4_10_0	4	0.35	0.28	1.26	0.04	9.17
inst_4x4_10_9	4	0.47	0.24	1.93	0.03	13.48
inst_4x4_14_0	6	2.29	1.37	1.67	0.13	17.50
inst_4x4_14_9	6	3.23	1.33	2.44	0.13	25.03
inst_4x4_15_0	8	3.00	1.25	2.39	0.34	8.74
inst_4x4_15_9	8	4.19	1.38	3.05	0.37	11.36
inst_4x4_19_0	10	5.15	3.01	1.71	1.36	3.79
inst_4x4_19_9	10	5.64	3.12	1.81	1.52	3.72
inst_4x5_10_0	5	464.81	67.21	6.92	0.19	2434.73
inst_4x5_10_9	5	50.25	16.24	3.09	0.20	253.50
inst_4x5_13_0	8	1340.93	266.14	5.04	0.68	1965.64
inst_4x5_13_9	8	1199.97	347.68	3.45	0.81	1484.73
inst_4x5_15_0	10	1859.16	277.02	6.71	2.44	760.63
inst_4x5_15_9	10	1713.83	285.35	6.01	2.89	592.04
inst_4x5_20_0	13	2946.60	272.61	10.81	25.74	114.48
inst_4x5_20_9	13	2763.03	354.49	7.79	29.46	93.78
inst_5x5_10_0	6	>24h	>24h	—	3.79	—
inst_5x5_10_9	6	>24h	>24h	—	4.33	—
inst_5x5_13_0	10	>24h	>24h	—	16.50	—
inst_5x5_13_9	10	>24h	>24h	—	22.69	—
inst_5x5_20_0	15	>24h	>24h	—	999.36	—
inst_5x5_20_9	15	>24h	>24h	—	1211.25	—
inst_5x6_10_0	7	>24h	>24h	—	137.15	—
inst_5x6_10_9	7	>24h	>24h	—	194.92	—

- [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] V. Samoladas, "Improved BDD algorithms for the simulation of quantum circuits," in *Algorithms - ESA*, D. Halperin and K. Mehlhorn, Eds., 2008, pp. 720–731.
- [8] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "High-performance QuIDD-Based simulation of quantum circuits," in *Design, Automation and Test in Europe*, 2004.
- [9] 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.
- [10] 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*, 2020.
- [11] 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*, 2020.
- [12] S. Aaronson and L. Chen. "Complexity-Theoretic Foundations of Quantum Supremacy Experiments." arXiv: 1612.05903 [quant-ph]. (2016).
- [13] I. L. Markov, A. Fatima, S. V. Isakov, and S. Boixo. "Quantum supremacy is both closer and farther than it appears." arXiv: 1807.10749 [quant-ph]. (2018).
- [14] Z.-Y. Chen, Q. Zhou, C. Xue, X. Yang, G.-C. Guo, and G.-P. Guo, "64-Qubit Quantum Circuit Simulation," *Science Bulletin*, vol. 63, no. 15, pp. 964–971, 2018.
- [15] E. Pednault *et al.* "Pareto-efficient quantum circuit simulation using tensor contraction deferral." arXiv: 1710.05867 [quant-ph]. (2020).
- [16] I. L. Markov and Y. Shi, "Simulating quantum computation by contracting tensor networks," *SIAM J. Comput.*, vol. 38, no. 3, pp. 963–981, 2008.
- [17] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, and H. Neven. "Simulation of low-depth quantum circuits as complex undirected graphical models." arXiv: 1712.05384 [quant-ph]. (2018).
- [18] S. Boixo *et al.*, "Characterizing quantum supremacy in near-term devices," *Nature Phys.*, vol. 14, no. 6, pp. 595–600, 2018. arXiv: 1608.00263.
- [19] B. Villalonga *et al.*, "A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware." *Npj Quantum Inf.*, vol. 5, no. 1, pp. 1–16, 2019.
- [20] C. Huang *et al.* "Classical Simulation of Quantum Supremacy Circuits." arXiv: 2005.06787 [quant-ph]. (2020).
- [21] J. Gray and S. Kourtis, "Hyper-optimized tensor network contraction," *Quantum*, vol. 5, p. 410, 2021. arXiv: 2002.01935.
- [22] F. Pan and P. Zhang. "Simulating the Sycamore quantum supremacy circuits." arXiv: 2103.03074 [quant-ph]. (2021).
- [23] L. Chin-Yung, W. Shiou-An, and K. Sy-Yen, "An extended XQDD representation for multiple-valued quantum logic," *IEEE Trans. Comput.*, vol. 60, no. 10, pp. 1377–1389, 2011.
- [24] D. Miller and M. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," in *Int'l Symp. on Multi-Valued Logic*, 2006.
- [25] 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.
- [26] X. Hong, X. Zhou, S. Li, Y. Feng, and M. Ying. "A Tensor Network based Decision Diagram for Representation of Quantum Circuits." arXiv: 2009.02618 [quant-ph]. (2020).
- [27] E. Bernstein and U. Vazirani, "Quantum Complexity Theory," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1411–1473, 1997.
- [28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [29] R. Wille, L. Burgholzer, and M. Artner, "Visualizing decision diagrams for quantum computing," in *Design, Automation and Test in Europe*, 2021.
- [30] S. Boixo and C. Neill. "The Question of Quantum Supremacy," Google AI Blog. (2018), [Online]. Available: <https://github.com/sboixo/GRCS>.
- [31] S. Hillmich, A. Zulehner, and R. Wille, "Concurrency in DD-based Quantum Circuit Simulation," in *Asia and South Pacific Design Automation Conf.*, 2020, pp. 115–120.
- [32] R. Wille, S. Hillmich, and L. Burgholzer, "JKQ: JKU tools for quantum computing," in *Int'l Conf. on CAD*, 2020.