

Bidirectional Path Integral Monte Carlo Simulation of Quantum Circuits

Luis Paulo Santos

psantos@di.uminho.pt

University of Minho

Thomas Bashford-Rogers

University of Warwick

Article

Keywords:

Posted Date: September 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11019572/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Bidirectional Path Integral Monte Carlo Simulation of Quantum Circuits

Luís Paulo Santos^{1,*} and Thomas Bashford-Rogers^{2,†}

¹*HASLab, INESC TEC / Universidade do Minho / INL, Portugal[‡]*

²*Warwick Manufacturing Group, Warwick University, Coventry, United Kingdom[§]*

We propose an approximate classical algorithm for estimating quantum circuits' transition amplitudes, $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$, based on Monte Carlo sums over bidirectionally sampled Feynman paths. The algorithm generates Feynman paths, i.e., sequences of intermediate computational basis states, by combining information from random walks initiated at both the input and output states, Ψ_0 and Ψ_f , respectively. It further combines this bidirectional approach with Multiple Importance Sampling, a powerful variance reduction technique that avoids variance peaks caused by divisions by the very low sampling probabilities associated with hard to reach paths. Empirical data demonstrate the superior convergence rate exhibited by the bidirectional algorithm over a unidirectional, forward-only sampling approach. This superiority is due to the effectiveness of the proposed algorithm in finding and weighting paths with non-zero amplitude in an extremely sparse path space. Sum of paths algorithms exhibit very favourable space complexity. Monte Carlo replaces exhaustive enumeration of the exponentially large path space by stochastic sampling, the computational cost being determined by the variance of the estimator. This work demonstrates these two features by presenting results on very large quantum circuits, with up to 4096 qubits.

I. INTRODUCTION

Current quantum computing platforms are scarce and noisy, with performance limited by decoherence. Execution of quantum circuits is therefore expensive, due to scarcity, and limited to shallow circuits due to short decoherence times. Classical simulation of quantum circuits is a fundamental capability enabling researchers and practitioners to validate algorithms, study quantum advantage, and benchmark quantum solutions. Simulating arbitrary circuits requires time and/or space that grow exponentially with circuit size, given by the number of qubits n and gate depth L , motivating research efforts to efficiently simulate families of quantum circuits. These techniques exploit characteristics exhibited by most quantum circuits, such as structure, redundancy, and low magic, among others.

This paper proposes an approximate classical algorithm for estimating quantum circuits transition amplitudes, $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$, based on Monte Carlo sums over bidirectionally sampled Feynman paths. The algorithm is inspired by current practice in physically based global illumination. Bidirectional Path Tracing (BDPT) is a Monte Carlo algorithm [24, 38] that excels at finding low-probability light paths connecting the light source to the camera. Similarly, this work shows that a bidirectional approach to the Path Integral Monte Carlo (PIMC) problem, within the context of simulating a quantum circuit described by a time-evolution operator $\hat{U}$, efficiently finds low-probability paths connecting $|\Psi_0\rangle$ to $|\Psi_f\rangle$. Challenges raised by the sparsity of the Feynman path space

are efficiently handled by the bidirectional sampling procedure.

Simulation algorithms based on summing over Feynman paths allow for the classical evaluation of transition amplitudes $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$ with $\mathcal{O}(nL)$ space complexity, since the simulator only maintains data on the circuit itself and a running sum over all Feynman paths' amplitudes. The corresponding path space, $\mathcal{P}$, has cardinality $|\mathcal{P}| = 2^{n*(L-1)}$. The exponential advantage in space complexity, compared to Schrödinger-based approaches, comes at the cost of increased time complexity associated with $\mathcal{P}$'s exponentially large number of paths.

Monte Carlo (MC) techniques evaluate the estimator $\tilde{I} \approx \langle \Psi_f | \hat{U} | \Psi_0 \rangle$ using N stochastically generated samples, each sample being a classical path from $|\Psi_0\rangle$ to $|\Psi_f\rangle$, $\bar{z}_i \in \mathcal{P}$, $i = 0 \dots N-1$. The goal is to overcome the curse of dimensionality problem by evaluating many fewer samples than the total number of paths in the path space, $N \ll |\mathcal{P}|$.

For a finite number of samples, the MC estimator will exhibit variance, whose magnitude depends on three major factors: i) how effectively contributing paths, $f(\bar{z}_i) \neq 0$, are found within a often very sparse path space, ii) how well samples' contributions, $f(\bar{z}_i)$, are weighted, and iii) destructive interference among paths with mutually cancelling contributions, referred to as the numerical sign problem.

The proposed Monte Carlo estimator improves the variance associated with path space sparseness and paths' weighting, thus addressing items i) and ii) above. The bidirectional approach samples path space by combining random walks initiated from both $|\Psi_0\rangle$ and $|\Psi_f\rangle$, thus increasing the probability of finding non-zero paths within a highly sparse domain. The paths' contributions are weighted using Multiple Importance Sampling (MIS), further improving the MC convergence rate by avoiding divisions by the very small probabilities associated with hard to find paths in $\mathcal{P}$.

* psantos@di.uminho.pt

† Thomas.Bashford-Rogers@warwick.ac.uk

‡ <https://luisps.github.io/>

§ <https://thomasbashfordrogers.com/>

This paper details the proposed bidirectional sampling (BS) approach and provides empirical evidence supporting the following three hypotheses:

- H1 – Convergence rate::** BS achieves better convergence than a unidirectional forward sampling algorithm.
- H2 – Simulation of large circuits::** BS allows estimating transition amplitudes for very large quantum circuits (e.g., 4096 qubits) because its memory complexity is $\mathcal{O}(nL)$ and it efficiently finds non-zero paths in a very large and sparse path space.
- H3 – Sparsity and numerical sign problem::** BS improvements in the convergence rate, over the unidirectional approach, are due to a more efficient sampling of the sparse path space, rather than to a better handling of destructive interference, i.e., MC’s numerical sign problem.

The document is structured as follows. The next section presents the most important related work; section III reviews Monte Carlo integration, bidirectional and multiple importance sampling. Section IV describes the proposed approach, the following section describes the experimental methodology and section VI analyses experimental results. Section VII concludes the paper.

II. RELATED WORK

Classical simulation of quantum circuits is a central problem in quantum computing, underpinning algorithm validation, benchmarking of near-term devices, and the study of quantum advantage. The computational cost of simulating a quantum system typically grows exponentially with the number of qubits in the worst case [8], motivating a diverse set of simulation techniques that exploit structure, approximations, redundancy, low magic, or alternative representations. Simulation methods can be broadly categorized by the representation of quantum mechanics they are based upon, including the Schrödinger, Feynman, and Heisenberg perspectives.

A. The Schrödinger Picture

In the Schrödinger picture, the quantum state evolves under the action of time evolution unitary operators $\hat{U}$. Simulation methods in this paradigm explicitly represent or approximate the **full state vector** $|\Psi\rangle$ of size 2^n , updating it gate by gate [28]. Single- and two-qubit gates correspond to sparse linear transformations $\hat{U}_g$ applied to $|\Psi\rangle$. This is a conceptually simple and exact approach. Memory requirements, however, grow exponentially with system size ($\mathcal{O}(2^n)$), limiting simulations to roughly 30–50 qubits depending on available resources.

Tensor network approaches exploit the structure inherent in limited-entanglement quantum circuits [9, 26]. The circuit is mapped to a network of tensors, and the simulation reduces to contracting this network. The cost depends on the complexity of the contraction rather than the number of qubits. Tensor networks enable the simulation of systems that are significantly larger than those possible with state-vector approaches. Common techniques include matrix product states (MPS), projected entangled pair states (PEPS), and more general contraction schemes optimized via graph partitioning [15]. Highly entangled circuits, such as random circuits, require large intermediate tensors, bringing back exponential requirements.

Decision diagrams represent quantum states and operators as compressed graph structures that exploit redundancy and regularity in amplitudes [21, 43]. Operations correspond to graph manipulations, often yielding substantial memory savings. Efficiency depends critically on the amount of structure present; highly random or chaotic circuits degrade compression, leading to exponential growth. Nevertheless, decision diagrams provide a powerful alternative when the quantum circuits exhibit repeating patterns or symmetries.

B. Feynman Path Integral Methods

In the Feynman picture, amplitudes are expressed as **sums over computational paths**. Each path $\bar{z} = (s_0 = \Psi_0, s_1, \dots, s_L = \Psi_f)$, with length L , corresponds to a sequence of intermediate basis states and the total amplitude is obtained by summing contributions from all paths $\bar{z} \in \mathcal{P}$:

$$\langle \Psi_f | \hat{U} | \Psi_0 \rangle = \sum_{\bar{z} \in \mathcal{P}} f(\bar{z}) = \sum_{(s_1, \dots, s_{L-1}) \in \mathcal{P}} \left[\langle \Psi_f | \hat{U}_{L-1} | s_{L-1} \rangle \prod_{t=L-2}^1 \left(\langle s_{t+1} | \hat{U}_t | s_t \rangle \right) \langle s_1 | \hat{U}_0 | \Psi_0 \rangle \right]. \quad (1)$$

This formulation avoids storing the full state vector but incurs a combinatorial explosion in the number of paths [1, 13]. Practical algorithms rely on path pruning

or on hybrid Schrödinger–Feynman methods, in which the circuit is partitioned into subcircuits that are simulated independently and recombined. Exact Feynman–

based approaches are particularly effective for circuits with small width or for simulations that trade memory for time. They underpin some of the most powerful classical simulations of random quantum circuits used in benchmarking quantum advantage experiments.

C. Heisenberg Picture Methods

In the Heisenberg picture, operators evolve backward through the circuit while the state remains fixed. A prominent approach is **Pauli propagation** [33]. Observables are decomposed into sums of Pauli operators, and each Pauli term is propagated through the circuit using conjugation rules. For Clifford circuits, Pauli operators remain Pauli under conjugation, enabling efficient simulation via stabilizer techniques. For general circuits, non-Clifford gates cause Pauli operators to expand into linear combinations, leading to exponential growth. Truncation strategies that yield approximate estimates were proposed to mitigate exponential growth [34]. Pauli-based methods are particularly useful for estimating expectation values.

Stabilizer simulation tableau-based methods constitute the canonical Heisenberg approach for Clifford circuits [2, 19]. Operators are represented implicitly via stabilizer generators, with evolution corresponding to symplectic updates of the tableau. The algorithm tracks how a basis of Pauli operators transforms under conjugation, rather than tracking a single observable.

Pauli Transfer Matrix approaches propagate operators using a linear map in operator space [20]. Representing quantum channels as matrices acting on the Pauli basis makes these algorithms particularly suited for simulating noisy circuits. The **Matrix Product Operator** (MPO) Heisenberg Evolution is the Heisenberg analog of tensor networks, with observables represented as MPOs [32]. The rationale is to exploit the fact that operator entanglement often, but not always, grows slower than state entanglement.

D. Monte Carlo and Approximate Simulation

Monte Carlo methods provide a framework for approximate simulation, estimating amplitudes or expectation values via stochastic sampling. General PIMC can be broadly described as stochastically sampling N paths $\bar{z}_i$, with probability $p(\bar{z}_i)$, over path space $\mathcal{P}$, evaluating samples' values $f(\bar{z}_i)$, and linearly combining these with weights $\frac{1}{p(\bar{z}_i)}$. The PIMC estimator, $\tilde{I}_{MC}$, takes the form

$$\langle \Psi_f | \hat{U} | \Psi_0 \rangle \approx \tilde{I}_{MC} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\bar{z}_i)}{p(\bar{z}_i)}. \quad (2)$$

Typical approaches **importance sample Feynman paths** according to some probability distribution, yield-

ing unbiased estimators with variances that depend on the paths' interference structure [35, 36]. Alternative approaches use **quasi-probability decompositions**, expressing non-Clifford gates as linear combinations of Clifford operations with signed weights [11, 30]. Sampling over these decompositions enables simulation using efficient stabilizer methods, at the cost of a variance overhead quantified by the negativity of the Clifford weights. Hybrid methods combine Monte Carlo with tensor networks or decision diagrams, sampling over contraction orders or diagram branches [17]. These approaches can significantly extend the reach of classical simulation, particularly for circuits with moderate non-Clifford content (i.e., low magic). A key challenge is controlling variance, which can grow exponentially with the circuit's depth, non-classicality, and, most importantly, with the **numerical sign problem** that emerges from destructive interference among paths [7]. Techniques such as importance sampling, control variates, and stratified sampling have been proposed to improve efficiency [23], but are ineffective against the exponential sampling requirements imposed by destructive interference patterns.

E. Simulation Hardness

Classical simulation of quantum circuits is widely believed to be intractable in general, meaning that it requires resources (space and/or time) that grow exponentially with circuit size. The metrics used to quantify this hardness depend on the underlying algorithm: the efficiency of the different simulation paradigms depends on distinct structural properties of the quantum state or circuit. Accordingly, a variety of metrics has been proposed, each capturing the challenges faced by a specific class of simulators.

Vanilla full-state Schrödinger approaches' cost scales as $\mathcal{O}(2^n)$ in both space and time. This is a worst-case metric that does not capture potential improvements from exploiting the quantum circuit's structure. Low-entanglement states admit efficient representations (e.g., matrix product states), suggesting that bipartite entropies or the maximum Schmidt rank can serve as proxies for simulation efficiency in tensor network approaches [26, 29, 42]. Entanglement alone, however, does not characterize classical hardness: stabilizer states can exhibit high entanglement while remaining efficiently simulable due to their strong algebraic structure. Additional resources – such as magic, phase complexity, or lack of redundancy – are necessary to capture the full computational difficulty. Additionally, treewidth – a measure of how well the tensor network graph can be decomposed into a tree of small overlapping clusters – is often used to characterize contraction complexity.

Decision diagram simulators exploit redundancy by sharing identical substructures. Their complexity is measured by the compression ratio, $\frac{2^n}{N_{DD}}$, with N_{DD} the number of nodes in the diagram. It quantifies the redundancy

of the amplitude tensor, i.e., the number of distinct sub-tensors in its recursive decomposition [14, 41]. Highly structured states (e.g., stabilizer states) exhibit strong compression, whereas generic states lead to $N_{\text{DD}} \sim 2^n$.

Clifford circuits can be efficiently simulated using stabilizer formalisms. The challenge is the presence of non-Clifford resources, quantified by magic, often defined as the stabilizer extent:

$$\xi(\psi) = \min \left(\sum_k |c_k| \right)^2, \quad |\psi\rangle = \sum_k c_k |\phi_k\rangle, \quad (3)$$

with $|\phi_k\rangle$ being stabilizer states. If $|\psi\rangle$ is a stabilizer state, then $\xi = 1$. Otherwise, ξ grows (often exponentially) with the number of T gates. Since the simulation cost in many stabilizer decomposition algorithms scales polynomially with ξ , it is a key hardness metric [25]. Additional measures of magic include robustness of magic [22] and mana [40].

Approaches based on the sum of Feynman paths express the amplitude $\langle\psi_f|\hat{U}|\psi_0\rangle$ as a sum over exponentially many computational paths. Hardness arises from the number of contributing paths (i.e., path space density) and the lack of degeneracy, i.e., few paths share identical contributions [16]. This connects directly to redundancy. If many paths have identical, eventually zero, contributions, they can be grouped; otherwise, the sum is irreducible.

The complexity of Heisenberg-based methods depends on the growth of the Pauli support and on the number of Pauli terms generated under conjugation as observables are propagated backward through the circuit [18]. Clifford circuits preserve Pauli structure, but non-Clifford gates cause a proliferation of terms. This growth is closely related to magic and can be understood as its operator-space analog.

The efficiency of Path Integral Monte Carlo methods depends on two key factors [30, 36, 37]: how effectively the method finds contributing paths in the highly sparse path space, and how hard the numerical sign problem is, caused by destructive interference among paths. The latter is often quantified by resorting to the inverse squared average phase, $\chi_{\mathcal{P}} = \langle e^{i\phi} \rangle_{\mathcal{P}}^{-2}$ [4, 44].

The amplitude of a path $\bar{z}$ can be written as $f(\bar{z}) = |f(\bar{z})|e^{i\phi_{\bar{z}}}$; trivially, the respective phase is given by $e^{i\phi_{\bar{z}}} = \frac{f(\bar{z})}{|f(\bar{z})|}$. $\mathcal{P}$'s average phase is defined as

$$\langle e^{i\theta} \rangle_{\mathcal{P}} = \frac{|\sum_{\bar{z} \in \mathcal{P}} f(\bar{z})|}{\sum_{\bar{z} \in \mathcal{P}} |f(\bar{z})|} \in [0 \dots 1]. \quad (4)$$

If the phases of contributing paths oscillate, then the numerator above becomes small due to destructive interference and $\langle e^{i\theta} \rangle_{\mathcal{P}}$ approaches 0, indicating a hard numerical sign problem.

The inverse squared average phase [3] is given by

$$\chi_{\mathcal{P}} = \langle e^{i\phi} \rangle_{\mathcal{P}}^{-2} = \left(\frac{\sum_{\bar{z} \in \mathcal{P}} |f(\bar{z})|}{\sum_{\bar{z} \in \mathcal{P}} |f(\bar{z})|} \right)^2 \in [1 \dots \infty]. \quad (5)$$

$\chi_{\mathcal{P}}$ is often used to quantify the Monte Carlo simulation hardness because for highly oscillatory path spaces it is closely related to the estimator's relative variance:

$$RVar(\tilde{I}_{MC}) = \frac{Var(\tilde{I}_{MC})}{(\mathbb{E}[\tilde{I}_{MC}])^2} \approx \frac{1}{N \langle e^{i\theta} \rangle_{\mathcal{P}}^2} = \frac{\chi_{\mathcal{P}}}{N} \quad (6)$$

For fixed relative variance and highly oscillatory path spaces, sampling requirements scale as $N \propto \mathcal{O}(\langle e^{i\theta} \rangle_{\mathcal{P}}^{-2}) = \mathcal{O}(\chi_{\mathcal{P}})$.

These metrics are deeply connected. Magic can be thought to induce phase randomness, destroying redundancy in the amplitude tensor, increasing Pauli complexity, and leading to an exponentially large inverse-squared average phase.

III. MONTE CARLO PATH INTEGRATION

A path integral is an integral over a space of functions, which are referred to as paths, trajectories, histories, etc., depending on the context. The path integral is typically denoted by

$$I = \int_{\mathcal{P}} F(\bar{z}(x)) d\mu(\bar{z}(x)) \quad (7)$$

where $\mathcal{P}$ is the space of paths $\bar{z}(x)$, $F(\bar{z}(x))$ is a functional that assigns a value to the entire path, and $d\mu(\bar{z})$ is a measure over $\mathcal{P}$. In the context of this work, we consider fixed-length discretized paths consisting of L transitions and represented by a sequence of $L + 1$ states $(x_0, x_1, \dots, x_{L-1}, x_L)$. The corresponding integral over paths of length L is

$$I_L = \int_{\mathcal{P}_L} F(x_0, \dots, x_L) dx_0 \dots dx_L. \quad (8)$$

Quantum circuits already entail a discretization of the quantum state time evolution into L time intervals, where L is the circuit's depth. The problem being addressed is therefore of the form depicted by equation 8.

Computing $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$ entails integrating over the path space

$$\mathcal{P}_L = \{(s_0, \dots, s_L) : s_0 = \Psi_0, s_L = \Psi_f, s_1 \dots s_{L-1} \in \mathcal{S}\} \quad (9)$$

with $\mathcal{S}$ being the set of all possible states at each vertex along the path. This work integrates over paths defined in the computational basis, therefore $\mathcal{S} = \{0 \dots 2^n - 1\}$ is discrete, with n being the number of qubits in the circuit. Given that there are $L - 1$ intermediate states (or path vertices, excluding s_0 and s_L), then $\mathcal{P} = \mathcal{S}^{L-1}$ and $|\mathcal{P}| = 2^{n*(L-1)}$. A path is therefore a classical object, denoting a history of the system evolution over classical states along L time intervals, each corresponding to a layer of quantum gates.

The transition amplitude is given by a sum:

$$\langle \Psi_f | \hat{U} | \Psi_0 \rangle = I = \sum_{\bar{z}_i \in \mathcal{P}} f(\bar{z}_i), \quad (10)$$

and $f(\bar{z}_i) : \mathcal{P} \rightarrow \mathbb{C}$ is evaluated according to the summand in equation 1.

Monte Carlo integration enables efficient estimation of equation 10 by stochastically drawing $N \ll |\mathcal{P}|$ samples (paths) from $\mathcal{P}$, as given by:

$$I = \sum_{\bar{z}_i \in \mathcal{P}} f(\bar{z}_i) \approx \tilde{I}_{MC} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\bar{z}_i)}{p(\bar{z}_i)}, \quad (11)$$

where $p(\bar{z}_i)$ is the probability with which the path $\bar{z}_i$ is sampled. The variance of the estimator is given by

$$\text{Var}(\tilde{I}_{MC}) = \frac{1}{N} \left(\sum_{i=0}^{N-1} \frac{f(\bar{z}_i)^2}{p(\bar{z}_i)} - I^2 \right) \quad (12)$$

Since the standard error is the square root of the variance, it decreases with the reciprocal of the square root of the number of samples: $\sigma_{MC} = \sqrt{\text{Var}(\tilde{I}_{MC})} \approx \mathcal{O}(\frac{1}{\sqrt{N}})$.

A. Forward Importance Sampling

Figure 1 depicts an example of a quantum circuit with 3 qubits and 4 layers. It also includes a path connecting the initial computational state 0 to the final state 0, through intermediate states (7, 3, 4).

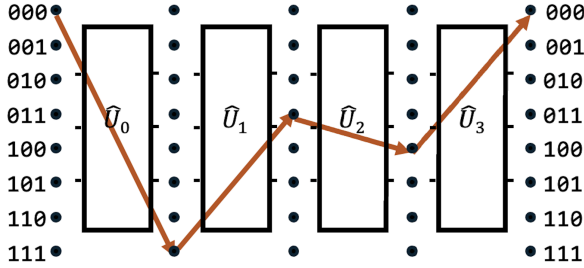


Figure 1: A path from $|000\rangle$ to $|000\rangle$ across a circuit with depth $L = 4$).

Paths can be sampled in many different ways. A relatively simple, yet surprisingly effective, approach is to use forward importance sampling. A path is stochastically generated across $L - 1$ layers, from the initial state $|\Psi_0\rangle$ to the next-to-last state, s_{L-1} . The output classical state $s_l \in \mathcal{S}$ after each intermediate layer $\hat{U}_{l-1}$ is sampled from the probability distribution $p(s_l | s_{l-1}, \hat{U}_{l-1}) = |\langle s_l | \hat{U}_{l-1} | s_{l-1} \rangle|^2$, with $s_0 = \Psi_0$ and $l = 1 \dots L - 1$. Since the final state is fixed, the probability of connecting s_{L-1} to Ψ_f is 1, although its amplitude $\langle \Psi_f | \hat{U}_{L-1} | s_{L-1} \rangle$ might be 0 – see figure 2.

A layer has depth 1, with each gate operating on a subset of 1 or 2 qubits, and each qubit being acted upon, at most, by a single gate. The layer's unitary is given by the tensor product of the gates' unitaries $\hat{U}_l = \bigotimes_j \hat{G}_{l,(j)}$, where j is a tuple identifying the qubits the gate acts upon.

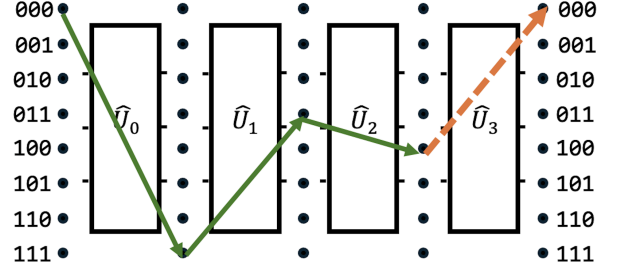


Figure 2: Forward importance sampling: up to layer $L - 1$ (green arrows) the path is guaranteed to have both non-zero amplitude and probability. The deterministic connection to the final state (orange dashed arrow) has probability 1, but its amplitude $\langle \Psi_f | \hat{U}_{L-1} | s_{L-1} \rangle$ can be zero.

The output classical state s_l after layer $\hat{U}_{l-1}$ is sampled on a per gate basis from the probability distribution defined by equation 13 – note that the short notation $p_{v,v'}$ is used to denote $p(v' | v, \hat{U})$, to improve readability.

$$\begin{aligned} p_{s_{l-1,(j)}, s_{l,(j)}} &= p(s_{l,(j)} | s_{l-1,(j)}, \hat{G}_{l-1,(j)}) \\ &= |\langle s_{l,(j)} | \hat{G}_{l-1,(j)} | s_{l-1,(j)} \rangle|^2. \end{aligned} \quad (13)$$

Consider, for example, the Hadamard operator, $\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and input state $s_{l-1} = |1\rangle$, which acts as a column selector in the operator matrix. In this example, that corresponds to the rightmost column, and the amplitudes of the output state are given by $\frac{1}{\sqrt{2}}[1, -1]^T$. The output classical state s_l is sampled from $\{|0\rangle, |1\rangle\}$ using the probability distribution obtained from squaring the magnitudes of the elements of the selected column:

$$p_{|1\rangle, s_l} = p(s_l | |1\rangle, \hat{H}) = [|\frac{1}{\sqrt{2}}|^2, |-\frac{1}{\sqrt{2}}|^2] = [\frac{1}{2}, \frac{1}{2}].$$

Layer $\hat{U}_{l-1}$'s joint sampling distribution is given by

$$p_{s_{l-1}, s_l} = \prod_j p(s_{l,(j)} | s_{l-1,(j)}, \hat{G}_{l-1,(j)}).$$

A path sampling probability $p(\bar{z})$ and amplitude $f(\bar{z})$ are given by the respective products across all layers (except the last for $p(\bar{z})$ since $p_{s_{L-1}, \Psi_f} = 1$):

$$p(\bar{z}) = \prod_{l=1}^{L-1} p(s_l | s_{l-1}, \hat{U}_{l-1}) \quad (14)$$

$$f(\bar{z}) = \prod_{l=1}^L |\langle s_l | \hat{U}_{l-1} | s_{l-1} \rangle|^2, \quad s_0 = \Psi_0, \quad s_L = \Psi_f. \quad (15)$$

The Forward Importance Sampling Monte Carlo (FS) estimator is given by

$$\langle \Psi_f | \hat{U} | \Psi_0 \rangle \approx \tilde{I}_{FS} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\bar{z}_i)}{p(\bar{z}_i)}. \quad (16)$$

Algorithm 1 details Forward Importance Sampling in pseudo-code.

Algorithm 1: Forward Importance Sampling

Data: $U[\cdot], L, n, \Psi_0, \Psi_f, N$
Result: Amplitude (f)

```

1  $s_0 \leftarrow \Psi_0$ ;  $s_L \leftarrow \Psi_f$ ;
2  $f \leftarrow 0. + i0.$ ;  $s \leftarrow 0$ ;
3 while ( $s < N$ ) do           // sample each path  $\bar{z}$ 
4    $p(\bar{z}) \leftarrow 1.$ ;  $f(\bar{z}) \leftarrow 1. + i0.$ ;
5    $\text{isZero} \leftarrow \text{false}$ ;
6    $s_c \leftarrow s_0$ ;           // current state
7   for ( $l \leftarrow 0 \dots L-1$ ) and (not  $\text{isZero}$ ) do
8      $p(s_n) \leftarrow 1.$ ;       //  $s_n$  next state
9     if  $l < L-1$  then         // sample next state
10       $s_{l+1}, p(s_n) \leftarrow \text{SampleU}(U[\cdot], l, s_c)$ ;
11    end
12     $s_c \leftarrow s_l$ ;  $s_n \leftarrow s_{l+1}$ ;
13     $f_l \leftarrow \text{EvalU}(U[\cdot], l, s_c, s_n)$ ;
14    if  $f_l = 0$  then         // early termination
15       $\text{isZero} \leftarrow \text{true}$ ;
16    else
17       $p(\bar{z}) \leftarrow p(\bar{z}) * p(s_n)$ ;
18       $f(\bar{z}) \leftarrow f(\bar{z}) * f_l$ ;
19    end
20  end
21  if not  $\text{isZero}$  then  $f \leftarrow f + \frac{f(\bar{z})}{p(\bar{z})}$ ;
22   $s++$ ;
23 end
24  $f \leftarrow \frac{f}{N}$ ;           // average
25 return  $f$                  // amplitude

```

Forward importance sampling convergence rate depends on how likely it is to forward sample a path segment leading to a next-to-last state, $|s_{L-1}\rangle$, which connects to the final state with non-zero amplitude, i.e., $\langle \Psi_f | \hat{U}_{L-1} | s_{L-1} \rangle \neq 0$. If such segments are very unlikely, then the non-zero paths have very low probability. When they occur, their contribution to the estimator is large because the amplitude is divided by a small probability ($\ll 1$), leading to a high-variance Monte Carlo estimate.

The Bidirectional Sampling Monte Carlo estimator proposed next addresses this challenge, minimizing the impact of zero-amplitude deterministic connections to fixed states, such as Ψ_0 and Ψ_f .

IV. BIDIRECTIONAL MONTE CARLO PATH INTEGRATION

Bidirectional Monte Carlo (BMC) algorithms aim to overcome the challenge of hard-to-access final states by exploiting path reversibility. This paper proposes applying BMC techniques to quantum circuit simulation.

A quantum circuit time evolution operator is described by a unitary matrix $\hat{U}$, implying that $\hat{U}^{-1} = \hat{U}^\dagger$. Paths can therefore be reversed, implying that transition amplitude estimates can be computed based on paths start-

ing from either Ψ_0 or Ψ_f , and, in fact, on combinations thereof.

BMC strives to build paths that successfully connect the initial and the final states by combining both forward (from Ψ_0) and backward (from Ψ_f) sampling.

A. Bidirectional Importance Sampling

A bidirectional sample consists of a pair of two path segments, $(\bar{z}_F, \bar{z}_B)$, forward and backward sampled, respectively, with length $L-1$, each. The forward segment is given by

$$\bar{z}_F = (s_0 = \Psi_0, s_1, \dots, s_{L-1}),$$

each next state, up to s_{L-1} , being sampled according to equation 13. The final deterministic connection from s_{L-1} to Ψ_f across $\hat{U}_{L-1}$ is not included.

The backward segment, labelled using t vertices starting from the final state $t_0 = \Psi_f$, is given by

$$\bar{z}_B = (t_{L-1}, \dots, t_1, t_0 = \Psi_f),$$

each previous state, up to t_{L-1} , being sampled according to equation 17, which uses the layers' unitaries adjoints. The final deterministic connection from t_{L-1} to Ψ_0 across $\hat{U}_0$ is not included.

$$\begin{aligned}
p_{t_{L-1-2}, t_{L-1-1}} &= p(t_{L-1-1} | \hat{U}_{l+1}^\dagger, t_{L-1-2}) \\
&= |\langle t_{L-1-1} | \hat{U}_{l+1}^\dagger | t_{L-1-2} \rangle|^2, l \in \{L-2 \dots 0\}
\end{aligned} \tag{17}$$

Figure 3 presents an example of sampled path segments for $L = 4$. Each of these path segments is guaranteed to have non-zero amplitude and probability.

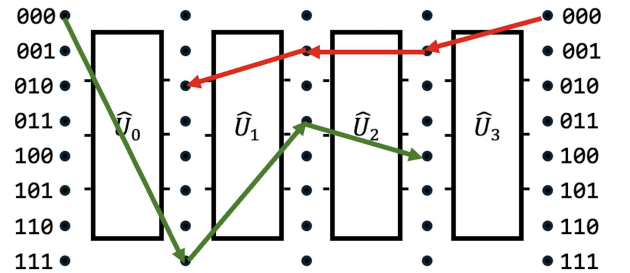


Figure 3: Forward (green) and backward (red) importance sampled path segments for $L = 4$.

L full paths are built by deterministically connecting $\bar{z}_F$'s state s_l to $\bar{z}_B$'s state t_{L-l-1} , with $l = 0 \dots L-1$ and $p_{s_l, t_{L-l-1}} = 1$ since the connection is deterministic. The sampling probability of such path $\bar{z}_l$ is given by

$$p(\bar{z}_l) = \prod_{j=0}^{l-1} (p_{s_j, s_{j+1}}) * \prod_{j=l+1}^{L-1} (p_{t_{L-j}, t_{L-j-1}}), l=0 \dots L-1 \tag{18}$$

and its amplitude is

$$f(\bar{z}_l) = \prod_{j=0}^{l-1} (\langle s_{j+1} | \hat{U}_j | s_j \rangle) * \langle t_{L-l-1} | \hat{U}_l | s_l \rangle * \prod_{j=l+1}^{L-1} (\langle t_{L-j} | \hat{U}_j^\dagger | t_{L-j-1} \rangle), l=0 \dots L-1 \quad (19)$$

Figure 4 illustrates the 4 paths built from the example in figure 3.

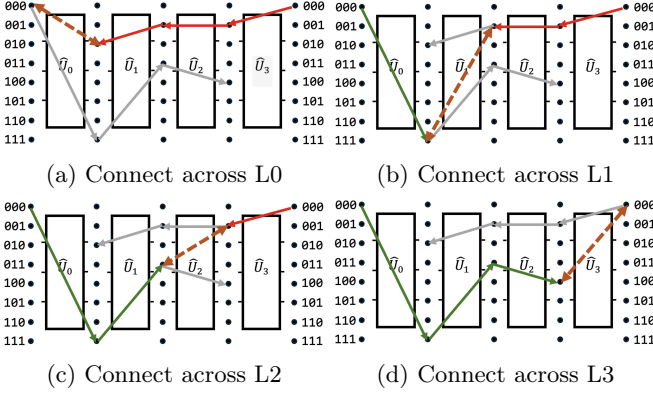


Figure 4: Paths generated by connecting forward and backward path segments. The dashed dark orange arrow represents the deterministic connection. Non-used path segments are dimmed to gray.

The potential advantage of Bidirectional Importance Sampling (BIS) over FS stems from the fact that each of these L paths is more likely to have a non-zero contribution than forward only paths, since it combines bidirectional data. Veach shows that the BIS estimator, given by equation 20, is unbiased [38, Chapter 9].

$$\tilde{I}_{BIS} = \frac{1}{N * L} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} \frac{f(\bar{z}_{i,l})}{p(\bar{z}_{i,l})}. \quad (20)$$

Algorithm 2 depicts this approach in pseudo-code.

B. Multiple Importance Sampling

The previous section generated complete paths by progressively connecting two path segments. While this is effective, variance can be further reduced via the use of Multiple Importance Sampling (MIS) [38]. This is a powerful variance reduction technique that further improves the $\tilde{I}_{BIS}$ estimator by considering all the ways the path could have been generated.

Paths sampled with low probability are the major sources of variance for the previously described Monte Carlo estimators. If a connection within a path is sampled with very low probability, then the whole path sampling probability is very low, $p(\bar{z}) \approx 0$. Vanilla Monte

Algorithm 2: Bidirectional Importance Sampling

Data: $U[\cdot], L, n, \Psi_0, \Psi_f, N$
Result: Amplitude (f)

```

1  $f \leftarrow 0. + i0.$ ;  $s \leftarrow 0$ ;
2 while ( $s < N$ ) do
    // pathF - forward  $\bar{z}_F = (\psi_0, \dots, s_{L-1})$ 
    // pF[l] - probability from  $\psi_0$  to  $s_l$ 
    // aF[l] - amplitude from  $\psi_0$  to  $s_l$ 
3    $pathF, pF, aF \leftarrow \text{GPathF}(U[\cdot], L, n, \Psi_0)$ ;
    // pathB - backward  $\bar{z}_B = (t_{L-1}, \dots, \Psi_f)$ 
    // pB[L-l-1] - probability  $\Psi_f \rightarrow t_{L-l-1}$ 
    // aB[L-l-1] - amplitude  $\Psi_f \rightarrow t_{L-l-1}$ 
4    $pathB, pB, aB \leftarrow \text{GPathB}(U[\cdot], L, n, \Psi_f)$ ;
5   for ( $l \leftarrow 0 \dots L-1$ ) do
    // connect across layer l
6      $aC \leftarrow \text{Connect}(U[\cdot], l, pathF, pathB)$ ;
7      $f(\bar{z}) \leftarrow aF[l] * aC * aB[l]$ ;
8      $p(\bar{z}) \leftarrow pF[l] * 1.0 * pB[l]$ ;
9      $f \leftarrow f + \frac{f(\bar{z})}{p(\bar{z})}$ ; // accumulate
10  end
11   $s++$ ;
12 end
13  $f \leftarrow \frac{f}{N * L}$ ; // average
14 return  $f$  // amplitude

```

Carlo estimators evaluate $\bar{z}$'s contribution as $\tilde{I}_{\bar{z}} = \frac{f(\bar{z})}{p(\bar{z})}$, which overestimates the average value, $\tilde{I}_{\bar{z}} \gg \tilde{I}$, resulting in large variance, thus requiring a large number of samples, N , to converge.

Each bidirectional path is built by deterministically connecting a forward path segment with a backward path segment across a specific layer $\hat{U}_l$. MIS exploits the fact that the same path could have been generated by deterministically connecting across any other layer, which would result in L different sampling strategies, by moving the deterministic connection through the layers. The same path is therefore associated with L different sampling probabilities, $p_d(\bar{z})$, $d=0 \dots L-1$, each resulting from deterministically connecting s_d to t_{L-d-1} . d indexes the layer $\hat{U}_d$ associated with the hypothetical deterministic connection. Each probability $p_d(\bar{z}_l)$ is given by:

$$p_d(\bar{z}_l) = \prod_{v=0}^{d-1} (p_{q_{v+1}, q_v}) * \prod_{v=d+1}^{L-1} (p_{q_v, q_{v+1}}), d=0 \dots L-1 \quad (21)$$

with $\{q_0, \dots, q_L\} = \{s_0, \dots, s_l, t_{L-l-1}, \dots, t_0\}$. Note that the path amplitude $f(\bar{z}_l)$ is independent of d , since only sampling probabilities change.

Figure 5 illustrates, for the path depicted in figure 4b and effectively created by a deterministic connection across $\hat{U}_l$, $l = 1$, the 4 different manners in which it could have been generated. The respective probabilities (equa-

tion 21) are:

$$\begin{aligned} p_{d=0}(\bar{z}) &= 1 * p_{t_2, s_1} * p_{t_2, t_1} * p_{t_1, t_0} \\ p_{d=1}(\bar{z}) &= p_{s_1, s_0} * 1 * p_{t_2, t_1} * p_{t_1, t_0} \\ p_{d=2}(\bar{z}) &= p_{s_1, s_0} * p_{t_2, s_1} * 1 * p_{t_1, t_0} \\ p_{d=3}(\bar{z}) &= p_{s_1, s_0} * p_{t_2, s_1} * p_{t_2, t_1} * 1 \end{aligned}$$

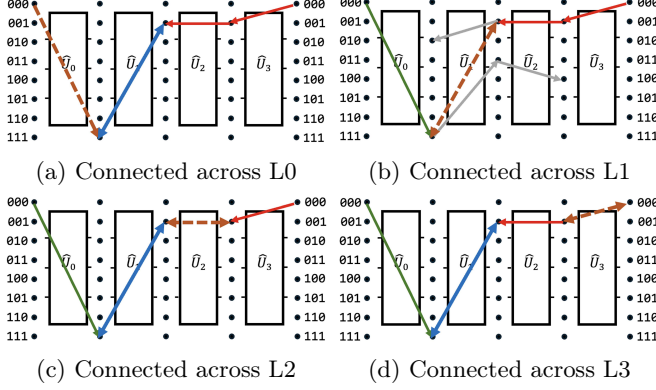


Figure 5: L different manners the same path, originally deterministically connected across L1 as depicted by the blue arrow (see figure 4b), could have been generated by shifting the deterministic connection (dark orange dashed arrow).

The MIS estimator combines the L different sampling probabilities, $p_d(\bar{z})$, for each sampled path $\bar{z}$, to allow for a robust estimate. Equation 22 adds an weight term $w(\bar{z}_i)$ to the MC estimator introduced with equation 2:

$$\langle \Psi_f | \hat{U} | \Psi_0 \rangle \approx \tilde{I}_{MC} = \frac{1}{N} \sum_{i=0}^{N-1} w(\bar{z}_i) \frac{f(\bar{z}_i)}{p(\bar{z}_i)}, \quad (22)$$

with $p(\bar{z}_i)$ standing for the probability with which $\bar{z}_i$ was actually sampled. The balance heuristic [38] proposes weights $w(\bar{z}_i)$ that make the sample contribution independent of $p(\bar{z}_i)$:

$$w(\bar{z}_i) = \frac{p(\bar{z}_i)}{\sum_{d=0}^{L-1} p_d(\bar{z}_i)} \quad (23)$$

By substituting 23 into equation 22 we get the Multiple Importance Sampling (MIS) estimator

$$\begin{aligned} \tilde{I}_{MIS} &= \frac{1}{N} \sum_{i=0}^{N-1} w(\bar{z}_i) \frac{f(\bar{z}_i)}{p(\bar{z}_i)} \\ &= \frac{1}{N} \sum_{i=0}^{N-1} \frac{p(\bar{z}_i)}{\sum_{d=0}^{L-1} p_d(\bar{z}_i)} \frac{f(\bar{z}_i)}{p(\bar{z}_i)} \\ &= \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(\bar{z}_i)}{\sum_{d=0}^{L-1} p_d(\bar{z}_i)} \end{aligned} \quad (24)$$

The denominator $\sum_{d=0}^{L-1} p_d(\bar{z}_i)$ in equation 24 is always larger than or equal to the denominator of the non MIS

estimator (equation 22 with $w(\bar{z}_i) = 1$), therefore $\tilde{I}_{MIS}$ is numerically more stable, exhibiting a better variance convergence rate.

Veatch and Guibas [39] show that the balance heuristic is near-optimal in the sense that its variance is bounded relative to that of other valid MIS estimators.

Applying MIS to the BMC simulation of quantum circuits results in the final Bidirectional Sampling estimator (BS):

$$\tilde{I}_{BS} = \frac{1}{N * L} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} \frac{f(\bar{z}_{i,l})}{\sum_{d=0}^{L-1} p_d(\bar{z}_{i,l})} \quad (25)$$

Algorithm 3 illustrates the approach in pseudo-code. Each of the L paths built by bidirectional sampling (line 6) requires the sum of L different path generation probabilities (line 10).

Algorithm 3: Bidirectional Multiple Importance Sampling (BS)

Data: $U[\cdot], L, n, \Psi_0, \Psi_f, N$

Result: Amplitude (f)

```

1  $s_0 \leftarrow \Psi_0$  ;  $s_L \leftarrow \Psi_f$  ;
2  $f \leftarrow 0. + i0.$  ;  $s \leftarrow 0$  ;
3 while ( $s < N$ ) do
    //  $pF[l]$  - probability from  $\psi_0$  to  $s_l$ 
    //  $aF[l]$  - amplitude from  $\psi_0$  to  $s_l$ 
4    $pathF, pF, aF \leftarrow \text{GPathF}(U[\cdot], L, n, \Psi_0)$  ;
    //  $pB[L-l-1]$  - probability  $\Psi_f \rightarrow t_{L-l-1}$ 
    //  $aB[L-l-1]$  - amplitude  $\Psi_f \rightarrow t_{L-l-1}$ 
5    $pathB, pB, aB \leftarrow \text{GPathB}(U[\cdot], L, n, \Psi_f)$  ;
6   for ( $l \leftarrow 0 \dots L-1$ ) do
    // connect across layer  $l$ 
7      $aC \leftarrow \text{Connect}(U[\cdot], l, pathF, pathB)$  ;
8      $f(\bar{z}) \leftarrow aF[l] * aC * aB[l]$  ;
9      $\text{sumProb} \leftarrow 0$  ;
10    for ( $d \leftarrow 0 \dots L-1$ ) do
11       $p(\bar{z}) \leftarrow pF[d] * 1.0 * pB[d]$  ;
12       $\text{sumProb} \leftarrow \text{sumProb} + p(\bar{z})$  ;
13    end
14     $f \leftarrow f + \frac{f(\bar{z})}{\text{sumProb}}$  ; // accumulate
15  end
16   $s++$  ;
17 end
18  $f \leftarrow \frac{f}{N * L}$  ; // average
19 return  $f$  ; // amplitude

```

MIS plays a fundamental role in increasing BMC convergence rate. Using a sum of probabilities in the estimator's denominator steers the computation away from divisions by small probabilities, which would otherwise yield large ratios and therefore introduce peaks in the variance. It is a robust, unbiased estimator often used by the computer graphics community to render physically based high quality images. We claim that it can be equally applied to the simulation of quantum circuits.

V. EXPERIMENTAL METHODOLOGY

This section describes the experimental methodology used to corroborate or falsify the three hypotheses formulated in section I. The Monte Carlo simulators were written in C++, using gcc 12.3.0 together with OpenMP to exploit multithreading parallelism and MPI to distribute the workload across multiple computing nodes. Jobs were executed on the Deucalion supercomputer, using up to 16 Intel x86 nodes, with 128 physical processing cores each.

A. H1 – Convergence rate

$\tilde{I}_{BS}$ achieves better convergence than $\tilde{I}_{FS}$ when estimating transition amplitudes for quantum circuits.

To corroborate or falsify this hypothesis, experimental results are obtained by running both estimators on a range of circuits.

Monte Carlo's convergence is measured by the variance of the estimator, given by

$$Var_{\tilde{I}} = \frac{1}{E} \sum_{e=0}^{E-1} \left(I - \tilde{I}_e \right)^2 \quad (26)$$

where E is the number of experiments, I is the true value of the integral and $\tilde{I}_e$ is the estimate of the e^{th} experiment. We use $E = 5$.

The two algorithms' convergence rates are compared using two different metrics: variance as a function of the number of samples N , and variance as a function of execution time T .

Data is collected using two different families of circuits, representative of different features and challenges concerning quantum state evolution:

Instantaneous Quantum Polynomial: (IQP) – kernel inversion test for two vectors encoded using the IQP kernel. The overlap is estimated by the probability of measuring $|0\rangle$. IQP circuits are most likely hard to simulate classically [12, 27].

Random: (RND) – generated by randomly sampling gates from the Clifford+T gate set, according to the rules defined in [10]. Random circuits are known to be exponentially hard to simulate classically (in the ideal, noise-free case) [6].

For each circuit family, three different sizes are used, differing in the number of qubits (n) and depth (L). Table I presents detailed data on each circuit. See section VIII for data availability. For each circuit, we also report the total number of paths connecting $|\Psi_0\rangle$ to $|\Psi_f\rangle$ and the number of such paths that actually have non-zero amplitude ($\#NZ$), allowing quantification of path space sparsity. Evaluating $\#NZ$ is a computationally very demanding task – values are not available (n.a.) for larger circuits.

(ID, n, L)	#paths / #NZ
(IQP5,5,17)	1E24 / 32
(IQP15,15,55)	7E243 / $\approx 3E4$
(IQP25,25,95)	1E830 / n.a.
(RND8,8,9)	2E19 / 512
(RND10,10,13)	1E43 / $\approx 2E5$
(RND12,12,15)	1E61 / $\approx 1E9$

Table I: Circuits for experimental assessment of the convergence rate (n - qubits, L - layers, $\#paths$ - total number of paths, $\#NZ$ - number of non-zero paths).

B. H2 – Simulation of large circuits

$\tilde{I}_{BS}$ allows estimating transition amplitudes of very large quantum circuits (e.g., 4096 qubits) because its memory complexity is $\mathcal{O}(nL)$ and it efficiently finds non-zero paths in a very large and sparse path space.

Evaluating $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$ for circuits with hundreds or thousands of qubits is not possible using commonly available classical simulators. To verify the correctness of the proposed algorithms, experiments are performed using deterministic Hidden Shift Circuits (**HSC**), which prepare a classical state, $|\Psi_f\rangle = |j\rangle$, $j \in \{0, \dots, 2^n - 1\}$ that is known *a priori* [31]. The simulator's output can thus be verified against the correct, known output. Table II presents detailed data on each used Hidden Shift Circuit.

(ID, n, L)	#paths
(HSC256,256,43)	1E3600
(HSC1024,1024,43)	1E14400
(HSC4096,4096,43)	1E57400

Table II: Large Hidden Shift Circuits (n - qubits, L - layers, $\#paths$ - total number of paths).

C. H3 – Sparsity and numerical sign problem

$\tilde{I}_{BS}$ improvements on the convergence rate, over $\tilde{I}_{FS}$, are due to a more efficient sampling of the sparse path space, rather than to a better handling of destructive interference, i.e., MC's numerical sign problem.

To assess the impact of non-classicality on simulation performance, an additional set of 500 random circuits is used. All circuits have the same shape: 8 qubits and depth 9. Gates are randomly drawn from a pool of X , H , T , and CX gates, with probabilities $[0.28, 0.28, 0.28, 0.16]$, respectively, whilst simultaneously making sure that each qubit is acted upon by exactly 9 gates. All circuits share several properties, such as shape and total number of paths associated with any $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$ ($2^{n*(L-1)} = 2^{64}$). The different gates' lay-

outs, however, result in wide ranges of sparsity and inverse squared phase, χ – see section II E. The ratio $\frac{Var_{FS}}{Var_{BS}}$ will be used to assess the impact these properties have on the estimators' relative convergence rates. Figure 6 illustrates one such circuit.

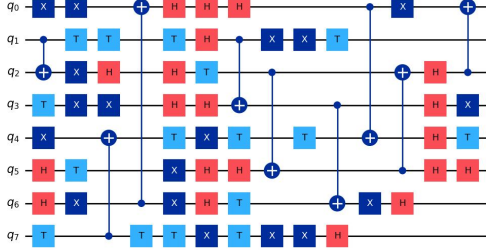


Figure 6: Example random circuit used to assess how sparsity and destructive interference impact MC estimators' convergence.

VI. RESULTS

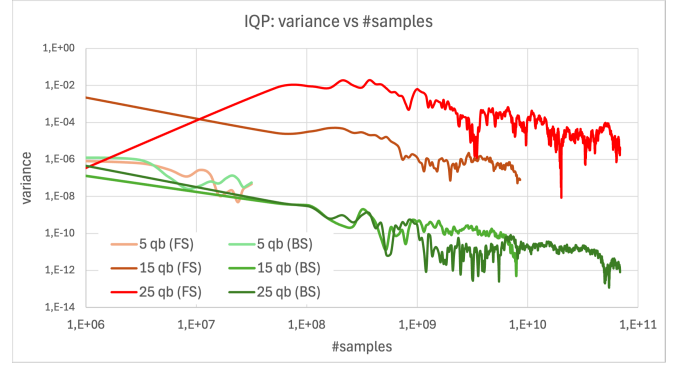
Empirical data were collected using the two algorithms for the different circuits. Results' analysis is organized into a comparison of the algorithms' convergence, handling of very large circuits, and the impact of the numerical sign problem.

To facilitate visual interpretation data obtained for the $\tilde{I}_{FS}$ estimator is presented in shades of red. In contrast, data for the $\tilde{I}_{BS}$ estimator is presented in shades of green. The saturation of the respective hue represents the size of the circuit associated with each curve: smaller circuits are depicted with less saturated colors.

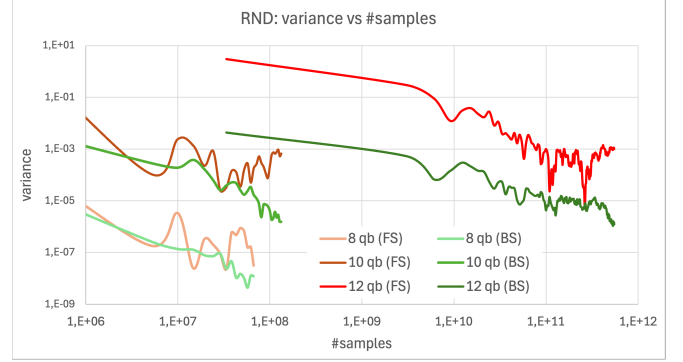
A. H1 – Convergence rate

Figure 7 illustrates how variance varies with the number of samples, N . **BS** outperforms **FS** for similar sampling counts, and this trend gets more significant as circuits get larger. For the **RND12** and **IQP25** circuits, **BS**'s variance is smaller than **FS**'s by 2 and 5 orders of magnitude, respectively. It must be kept in mind, however, that each **BS** sample gives rise to L paths.

Figure 8 shows that, for the same execution time, **BS** consistently outperforms **FS**, as initially hypothesized. For one sample, **FS** evaluates one forward path, whereas **BS** assesses a pair of (forward, backward) connections corresponding to L paths. Assessing these L paths is quite efficient, resulting in variance gains ranging from one to six orders of magnitude as the circuits' sizes increase.



(a) IQP 5, 15 and 25 qubits



(b) RND 8, 10 and 12 qubits

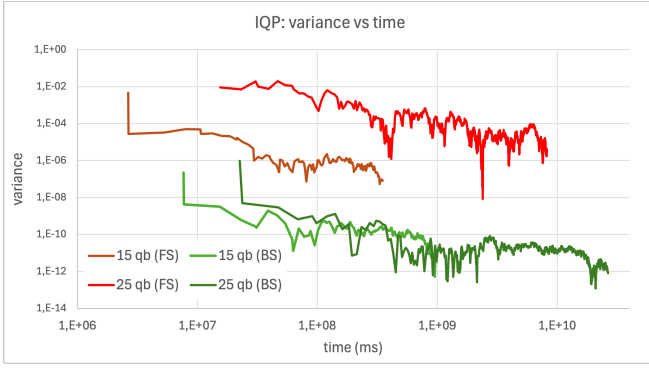
Figure 7: Variance as a function of the number of Monte Carlo samples

BS achieves superior convergence because it is more successful than **FS** in the discovery of non-zero amplitude paths, as illustrated in figure 9. For **IQP25** and the same number of samples, **BS** finds around six orders of magnitude more non-zero paths than **FS**, whereas for **RND12** the gain is about two orders of magnitude. The bidirectional approach, by connecting forward and backward generated paths, is much more successful at building viable paths. Using **MIS** to evaluate the weights ensures good convergence.

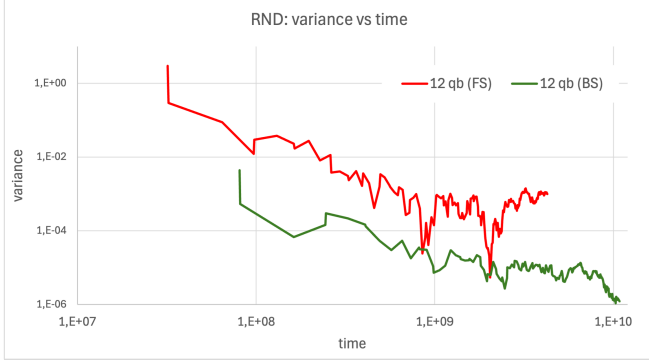
B. H2 – Simulation of large circuits

The motivation for Feynman sum of paths algorithms is their space complexity, which is $\mathcal{O}(nL)$, opening up the possibility of simulating very large circuits. Table III compares the memory footprint of the state vector (Qiskit 2.3.0) and the path summing simulators for **IQP** circuits (section V A) with varying numbers of qubits. The exponential space complexity of the former versus the linear complexity of the latter is well demonstrated.

Table IV presents the results of estimating transition amplitudes for the Hidden Shift Circuits (**HSC**) described in table II. These have 256, 1024 and 4096 qubits, which would be impossible to process with a naive



(a) IQP 15 and 25 qubits



(b) RND 12 qubits

Figure 8: Variance as a function of time

	RSS MiB						
n	5	15	20	25	27	35	45
State vector	109.0	131.0	217.0	2595.0	10267.0	–	–
Feynman	1.2	1.8	2.0	2.0	2.4	2.5	2.8

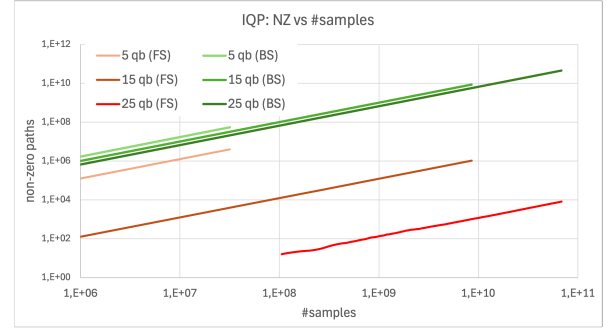
Table III: **RSS** (resident set size) in MiB for **IQP** circuits with varying number of qubits, using the state vector and the Feynman sum of paths simulators.

Schrödinger-type simulator with exponential memory requirements. 4, 8, and 16 computing nodes, with 128 physical processing cores each, have been used to simulate the three circuits, respectively. The number of samples, i.e., pairs of forward and backward path segments, is 2^{24} .

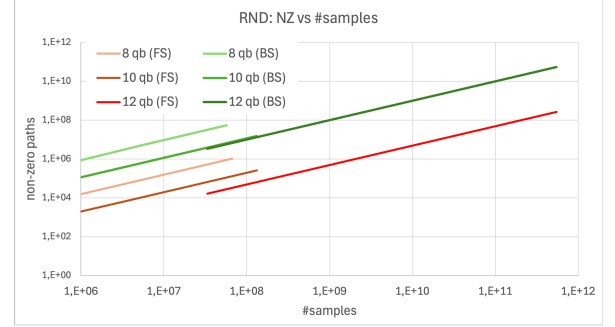
The data show that **BS** is capable of estimating the desired transition amplitude by finding some non-zero paths. These are very sparse path spaces; the **FS** algorithm did not find any non-zero path for up to 2^{26} samples.

The capability to simulate a 4096-qubit circuit is quite remarkable. A Schrödinger-based approach would require memory proportional to $\mathcal{O}(2^{4096})$, which is unfeasible.

HSC have been shown to be efficiently simulable using classical resources [5]. Our empirical results' observed execution time and resource usage, presented in Table IV,



(a) IQP 5, 15 and 25 qubits



(b) RND 8, 10 and 12 qubits

Figure 9: Number of non-zero amplitude paths as a function of the number of samples

are consistent with polynomial, rather than exponential, scaling over the tested range.

computing time				
#qb	nodes	(m:s)	#NZ	Var
256	4	6m51s	511	$4e(-6)$
1024	8	13m56s	502	$5e(-4)$
4096	16	29m31s	501	$4e(-3)$

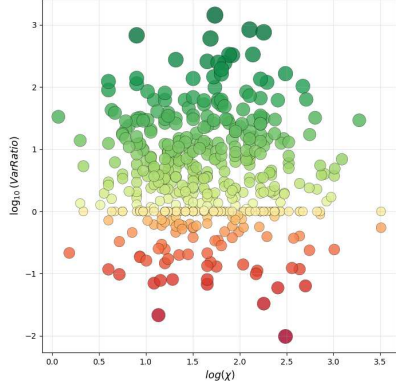
Table IV: **BS** simulation of very large **HSC** – $N = 2^{24}$

C. H3 – Sparsity and numerical sign problem

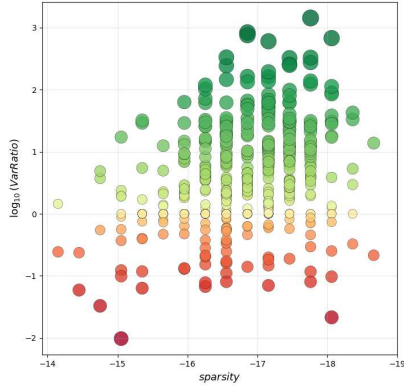
Figure 10 presents the logarithm of the ratio $\frac{Var_{FS}}{Var_{BS}}$ as a function of the logarithms of χ and sparsity, for all 500 circuits described in section VC and for constant execution time. Sparsity is defined as the logarithm of the ratio of the number of non-zero paths over the total number of paths – more negative values indicate sparser path spaces.

The vertical axis indicates how many orders of magnitude is Var_{BS} smaller than Var_{FS} – negative values in the vertical axis indicate that the **FS** algorithm achieved smaller variance than **BS**, which happens for a reduced number of circuits.

The variance ratio's logarithm for each circuit is displayed as a colored circle. Circles are yellow for unit ratios and get greener (respectively, redder) as the ratio increases (respectively, decreases). The circles' radii increase with the absolute value of the ratio logarithm.



(a) Variance ratio versus χ



(b) Variance ratio versus sparsity

Figure 10: Variance ratio as a function of sparsity and average phase - logarithmic scales

Visual inspection confirms that the variance ratio is significantly less sensitive to χ than to sparsity. Pearson's coefficients of the variance ratio relative to χ and to sparsity are ≈ -0.02516 and ≈ -0.13890 , respectively. These results are consistent with the hypothesis that the advantage of bidirectional multiple importance sampling, over forward sampling, comes from its capability to more effectively find and weight non-zero paths, rather than by handling the numerical sign problem caused by destructive interference among paths.

VII. CONCLUDING REMARKS

This paper proposes a bidirectional multiple importance sampling Monte Carlo estimator for transition amplitudes $\langle \Psi_f | \hat{U} | \Psi_0 \rangle$, where $\hat{U}$ represents a quantum circuit. This estimator, denoted by $\tilde{I}_{BS}$, is inspired by the Bidirectional Path Tracing algorithm [38] and its convergence rate is experimentally compared to that of a forward-only, unidirectional estimator, denoted by $\tilde{I}_{FS}$.

Empirical data on the convergence of the proposed estimator were collected using a wide range of quantum circuits and assuming noise-free, ideal conditions. These data confirm the superiority of the bidirectional algorithm, corroborating the three hypotheses proposed in the introduction.

- H1 – Convergence rate::** the variance of the bidirectional estimator is shown to decrease faster than that of the unidirectional estimator, as both a function of the number of samples and execution time;
- H2 – Simulation of large circuits::** the $\mathcal{O}(nL)$ memory requirements of the PIMC approach compound with the bidirectional algorithm's efficient path sampling and weighting to allow the simulation of very large quantum circuits (estimation of transition amplitudes for 4096 qubits' circuits demonstrated in this work);
- H3 – Sparsity and numerical sign problem::** the demonstrated convergence rate improvements, over the unidirectional approach, are due to a more efficient sampling of the sparse path space, rather than to a better handling of destructive interference, i.e., MC's numerical sign problem.

A. Future work

This work focused on ideal, non-noisy systems. Future work entails addressing real, noisy systems, understanding under which conditions does reversibility holds and whether bidirectional sampling maintains its superior convergence properties.

It has been shown that bidirectional multiple importance sampling of Feynman paths achieves better convergence rates than unidirectional sampling, being able to effectively sample within a very sparse path space. However, the main challenge of PIMC algorithms is the numerical sign problem. As destructive interference among paths' amplitudes becomes more significant, the sampling requirements grow exponentially. Figure 11 shows how the variance of the bidirectional estimator increases with the inverse squared average phase (χ) for a constant number of samples and the same 500 random circuits described in section V C.

Figure 12 illustrates how χ captures the strength of the numerical sign problem. Two random circuits, with exactly 128 non-zero paths in the path space associated

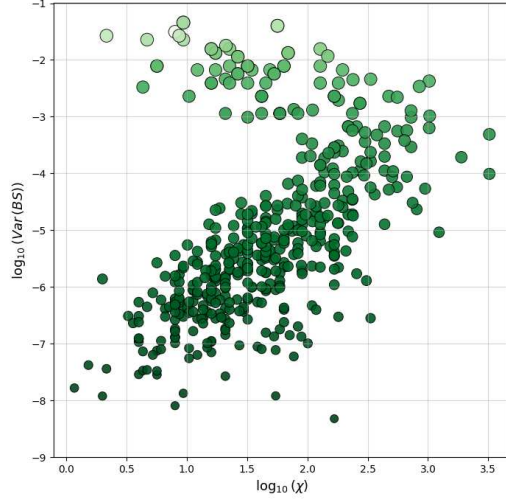
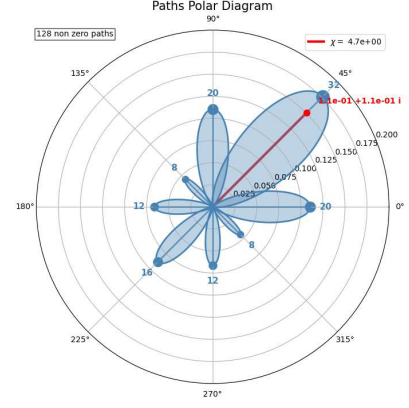


Figure 11: $\text{Var}_{\tilde{I}_{BS}}$ as a function of inverse squared average phase (χ); each data point corresponds to the execution of each of the circuits described in section VC and for a constant number of samples - logarithmic scales

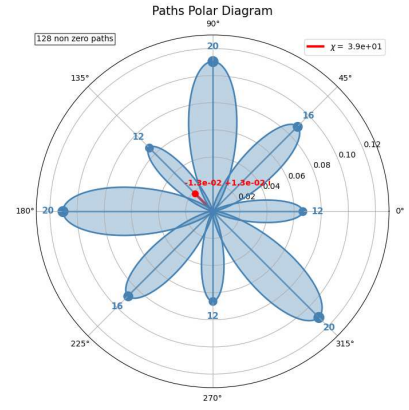
with $\langle 0|\hat{U}|0\rangle$ but exhibiting different χ , were selected among those described in section VC.

The width of each lobe is proportional to the number of non-zero paths with that particular phase (this number is displayed next to the corresponding marker). The length of the lobe is equal to the sum of the magnitudes of all paths with the same phase. The $\langle 0|\hat{U}|0\rangle$ transition amplitude, depicted in red, is given by the sum of the contributions of all paths. Figure 12a illustrates the case for $\chi = 4.7$. The lobes within the first quadrant concentrate most of the total amplitude. Destructive interference is not very strong and the Monte Carlo estimator exhibits low variance. Figure 12b, on the other hand, illustrates the case for larger χ . Paths amplitudes are now distributed across all quadrants and destructive interference among paths dominates the whole process, leading to high Monte Carlo sampling requirements.

The above analysis raises the following research question: can alternative path representations be used to alleviate the numerical sign problem? Paths in this work are sequences of intermediate states represented in the computational basis. Many alternative representations are possible, such as stabilizer states or quasiprobability decompositions, for example. Can different representations lead to different distributions of the paths amplitudes lobes in the polar phase diagram, hopefully reducing destructive interference? Studying how the path representation impacts on Monte Carlo sampling requirements can lead to a deeper understanding of the dynamics of path sum simulators. A related question is whether a metric can be found that suggests what representation



(a) $\chi = 4.7$



(b) $\chi = 38.8$

Figure 12: Polar diagrams for circuits with 128 non zero paths and different inverse squared average phases (χ)

to use for every particular problem. Computing, or estimating, this metric must be computationally efficient.

VIII. DATA AVAILABILITY

The quantum circuits analysed during the current study are available in the **Experimental-Data** git repository: <https://github.com/luisps/ExperimentalData.git>.

AUTHORS CONTRIBUTIONS

LPS and TBR both contributed to the original idea and results analysis. LPS was a major contributor to writing the manuscript and code development. All authors read and approved the final manuscript.

ACKNOWLEDGEMENTS

This study was funded by FCT - Fundação para a Ciência e Tecnologia, I.P., grant number UID/50014/2025

(<https://doi.org/10.54499/UID/50014/2025>). The funder played no role in study design, data collection, analysis and interpretation of data, or the writing of this manuscript.

-
- [1] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *Proceedings of the 32nd Computational Complexity Conference, CCC '17*, Dagstuhl, DEU, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
 - [2] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.
 - [3] Andrei Alexandru, Gökçe Başar, Paulo F. Bédague, and Neill C. Warrington. Complex paths around the sign problem. *Rev. Mod. Phys.*, 94:015006, Mar 2022.
 - [4] Andrei Alexandru, Paulo F. Bédague, Henry Lamm, and Scott Lawrence. Finite-density monte carlo calculations on sign-optimized manifolds. *Phys. Rev. D*, 97, May 2018.
 - [5] Matthew Amy and Lucas Shigeru Stinchcombe. Polynomial-time classical simulation of hidden shift circuits via confluent rewriting of symbolic sums, 2024.
 - [6] F. Arute, K. Arya, and R. Babbush. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019.
 - [7] Thomas Ayrat. Classical and quantum algorithms for many-body problems. *Comptes Rendus Physique*, 26:25–89, 2025.
 - [8] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
 - [9] Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell, 2017.
 - [10] S. Boixo, S.V. Isakov, and V.N. et al. Smelyanskiy. Characterizing quantum supremacy in near-term devices. *Nature Phys.*, 14:595–600, 2018.
 - [11] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, September 2019.
 - [12] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011.
 - [13] Lukas Burgholzer, Hartwig Bauer, and Robert Wille. Hybrid schrödinger-feynman simulation of quantum circuits with decision diagrams. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 199–206, 2021.
 - [14] Lukas Burgholzer and Robert Wille. Advanced simulation of quantum computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):3952–3965, 2020.
 - [15] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, Dec 2021.
 - [16] Christopher M. Dawson and Michael A. Nielsen. The solovay-kitaev algorithm. *Quantum Information & Computation*, 6(1):81–95, 2006.
 - [17] Andrew J. Ferris and Guifre Vidal. Perfect sampling with unitary tensor networks. *Phys. Rev. B*, 85:165146, Apr 2012.
 - [18] Hector J. Garcia and Igor L. Markov. Simulation of quantum circuits via stabilizer frames. *IEEE Transactions on Computer-Aided Design*, 31(12):1837–1849, 2012.
 - [19] D. Gottesman. The Heisenberg representation of quantum computers. In *Proc. of the XXII Int. Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1999. Long version at arXiv preprint quant-ph/9807006.
 - [20] Daniel Greenbaum. Introduction to Quantum Gate Set Tomography. *arXiv preprint*, 9 2015.
 - [21] Xin Hong, Xiangzhen Zhou, Sanjiang Li, Yuan Feng, and Mingsheng Ying. A tensor network based decision diagram for representation of quantum circuits, 2025.
 - [22] Mark Howard and Earl T. Campbell. Application of a resource theory for magic states to fault-tolerant quantum computing. *Physical Review Letters*, 118(9):090501, 2017.
 - [23] Mdivin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. WILEY-VCH Verlag GmbH, 1964.
 - [24] Eric Laforge and Yves Willems. Bi-directional path tracing. *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics'93)*, 93, 01 1998.
 - [25] Lorenzo et al. Leone. Quantifying magic: resource theory of nonstabilizerness. *PRX Quantum*, 3:020331, 2022.
 - [26] Igor L. Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.
 - [27] Simon C. Marshall, Scott Aaronson, and Vedran Dunjko. Improved separation between quantum and classical computers for sampling and functional tasks, 2024. Accessed: 2025-07-28.
 - [28] M Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, London, 2010.
 - [29] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.
 - [30] Hakop Pashayan, Joel J. Wallman, and Stephen D. Bartlett. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Physical Review Letters*, 115(7):070501, 2015.
 - [31] Filipa C. R. Peres and Ernesto F. Galvão. Quantum circuit compilation and hybrid computation using Pauli-based computation. *Quantum*, 7:1126, October 2023.
 - [32] Tomaž Prosen and Iztok Pižorn. Operator space entanglement entropy in a transverse ising chain. *Phys. Rev. A*, 76:032316, Sep 2007.
 - [33] Patrick Rall, Daniel Liang, Jeremy Cook, and William Kretschmer. Simulation of qubit quantum circuits via pauli propagation. *Phys. Rev. A*, 99:062337, Jun 2019.

- [34] Manuel S. Rudolph, Tyson Jones, Yanting Teng, Armando Angrisani, and Zoë Holmes. Pauli propagation: A computational framework for simulating quantum systems. *PRX Quantum*, 7:032001, Aug 2026.
- [35] T. Sauer. *THE FEYNMAN PATH GOES MONTE CARLO*, pages 29–42. World Scientific, 2001.
- [36] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Classical simulation of quantum circuits by half-space monte carlo. *Physical Review Letters*, 119(18):180509, 2017.
- [37] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical Review Letters*, 94, May 2005.
- [38] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997.
- [39] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 419–428, New York, NY, USA, 1995. Association for Computing Machinery.
- [40] Victor Veitch, Christopher Ferrie, David Gross, and Joseph Emerson. Negative quasi-probability as a resource for quantum computation. *New Journal of Physics*, 14:113011, 2012.
- [41] George F. Viamontes, Igor L. Markov, and John P. Hayes. Improving gate-level simulation of quantum circuits. *Quantum Information Processing*, 8(2-3):109–127, 2009.
- [42] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14):147902, 2003.
- [43] Robert Wille, Stefan Hillmich, and Lukas Burgholzer. *Decision Diagrams for Quantum Computing*, pages 1–23. Springer International Publishing, Cham, 2023.
- [44] Xu Zhang, Gaopei Pan, Xiao Yan Xu, and Zi Yang Meng. Fermion sign bounds theory in quantum monte carlo simulation. *Phys. Rev. B*, 106, Jul 2022.