

Phase-space tableau simulation for quantum computation

Selman Ipek ¹, Atak Talay Yucel ², Farzad Shahi ¹, Cagdas Ozdemir ³, and Cihan Okay ^{1,*}

¹*Department of Mathematics, Bilkent University, Ankara, Turkey*

²*Department of Computer Engineering, Bilkent University, Ankara, Turkey*

³*Department of Physics, Bilkent University, Ankara, Turkey*



(Received 28 October 2025; accepted 22 January 2026; published 5 March 2026)

We introduce a novel tableau-based classical simulation method for quantum computation, formulated within the phase-space framework of the extended stabilizer theory of closed noncontextual operators. This method enables the efficient classical simulation of a broader class of quantum circuits beyond the stabilizer formalism. We identify additional families of input states that are efficiently simulatable within this framework, going beyond stabilizer states, and implement the simulator and benchmark its performance on basic quantum algorithms, including the hidden shift and Deutsch–Jozsa algorithms.

DOI: [10.1103/11f2-nqht](https://doi.org/10.1103/11f2-nqht)

I. INTRODUCTION

Classical simulation—the task of performing quantum algorithms using classical computation—plays a central role in understanding the computational power of quantum systems relative to their classical counterparts [1]. Although simulating universal quantum circuits is widely believed to require exponential classical resources, various methods have been developed that perform efficiently in specific regimes. Our work introduces such a method: a tableau algorithm that extends stabilizer tableau techniques to handle a broader class of circuits.

A well-known tractable regime is described by stabilizer circuits, which encompasses stabilizer input states, Clifford unitaries, and Pauli measurements—optionally with classical feed-forward and randomness. The celebrated Gottesman–Knill (GK) theorem [2] provides a tableau algorithm that can efficiently simulate stabilizer circuits in the number of qubits. This tableau algorithm was later improved by Aaronson and Gottesman (AG) [3], who also, in that seminal work, extended these methods to a broader class of circuits going beyond stabilizer theory.

Among the extensions of the stabilizer formalism beyond Clifford operations, quantum computing with magic states (QCM) [4] provides an alternative framework for simulating non-Clifford circuits via the injection of magic states. Alongside the stabilizer rank methods introduced by Bravyi and collaborators [5–7], quasiprobability methods have been a key tool in systematically studying the classical simulation of QCM and its complexity. Initially based on Wigner

functions in the odd local-dimensional case [8], the extension of Wigner-like quasiprobability representations to qubits [9,10] ultimately led to the phase space method of Raussendorf *et al.* [11]. Central to this approach are the so-called closed noncontextual (CNC) operators, which form the qubit phase space and generalize stabilizer states.

For qudits of odd-local dimension, it is possible to attribute to every Pauli observable a deterministic outcome, which makes it possible to define the corresponding Wigner phase point operators [12]. This is not possible for qubits due to the emergence of state-independent contextuality, a phenomenon manifested in the Mermin square paradox [13]. The CNC phase space thus consists of operators that assign deterministic outcomes to subsets of Pauli observables such that a Mermin-type paradox is not encountered. These objects have also found applications in characterizing the quantum advantage of variational quantum eigensolvers (VQE) through the use of contextual subspace methods [14–16].

The general principle of quasiprobabilistic methods is to express the initial quantum state of QCM as a quasiprobabilistic mixture of a fixed set of reference operators. The overhead of classical simulation is then measured by a monotone called the robustness, defined with respect to the chosen reference operators, as formulated by Pashayan *et al.* [17]. One prominent example is due to Howard and Campbell [18], who used stabilizer states in their quasiprobabilistic model. In that setting, the robustness of magic—the associated magic monotone—characterizes the minimal amount of negativity needed to represent a magic state and, in turn, controls the cost of classical simulation.

Similar ideas have been applied using CNC operators as the reference set to quantify the overhead of classical simulation in the resulting quasiprobability representation. Raussendorf *et al.* [11] found that this overhead is smaller than in the stabilizer case, providing evidence for improved efficiency in Born rule estimation. Moreover, CNC operators admit positive representations for a larger set of states than probabilistic mixtures of stabilizer states, enabling efficient weak

*Contact author: cihan.okay@bilkent.edu.tr

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

simulation of a broader class of circuits than stabilizer circuits. Determining the precise boundary of this broader class remains an open question. These two advantages of phase-space simulation over stabilizer simulation motivate the continued study of phase-space methods for qubits.

In this work, we introduce a tableau method for the phase-space simulation, called the phase-space tableau or CNC tableau. Our phase-space tableau method extends the stabilizer tableau by incorporating additional structure beyond the standard stabilizer and destabilizer components. Specifically, the standard qubit phase space—defined as Pauli operators modulo scalars—is decomposed into a stabilizer part, a destabilizer part, and an additional Jordan–Wigner component, which is a special type of fermionic transformation. Under a Pauli measurement, each of these three components is updated according to rules that refine the standard stabilizer update procedure. The asymptotic complexity of our simulation scales quadratically in the number of qubits in both space and time, just as that of the AG simulation (Theorem 2).

In AG, the improved version of the GK simulator [3], there are two update cases depending on whether the measured Pauli operator lies inside or outside the stabilizer group of the state. Similarly, the original CNC update rules of [11] also involve two cases. In contrast, our update rules are divided into four cases (Case I–IV), more closely mirroring and refining the three cases considered in Ref. [3] for mixed stabilizer states. Our analysis shows that, in previous formulations [11], the update of a maximal CNC does not necessarily preserve maximality. In contrast, our update rules ensure that maximal CNCs map to maximal CNCs under Pauli measurement—a feature absent from earlier versions.

Much like standard stabilizer methods, the phase-space tableau introduced here can also be used to efficiently simulate contextual behaviors arising from the qubit stabilizer subtheory; see also Ref. [19]. Our phase-space tableau requires $O(n^2)$ bits of memory (Theorem 2) and so provides a concrete method of classical simulation that asymptotically saturates the memory lower bound given by Karanjai *et al.* in Ref. [20]. Moreover, the same framework connects the polyhedral phase-space picture to an operational model: we extend stabilizer circuits to a larger class of efficiently simulatable circuits whose input state is a variational noncontextual eigenstate [16] and whose subsequent dynamics consist of stabilizer operations (Clifford unitaries and adaptive Pauli measurements), thereby going beyond the usual stabilizer boundary of efficient classical simulation.

Our CNC tableau simulator is implemented in Ref. [21]. We compared its performance to implementations of the AG tableau algorithm and benchmarked it on basic quantum algorithms, including the hidden-shift and Deutsch–Jozsa algorithms.

II. PERFORMANCE ANALYSIS

A. CNC tableau in a nutshell

Our simulator uses CNC operators as classical keys to represent quantum states. For an n -qubit system, these keys are parametrized by an integer $1 \leq m \leq n$. As in the stabilizer tableau, the stabilizer S and destabilizer D components are

recorded; in addition, the CNC tableau contains a Jordan–Wigner component W .

A classical key in our CNC simulator is represented by a $2n \times (2n + 1)$ binary tableau of the form

$$\begin{bmatrix} D & \gamma_D \\ S & \gamma_S \\ W & \gamma_W \end{bmatrix},$$

where the D and S components are $(n - m) \times n$ matrices, and the W component is a $2m \times n$ matrix, each with separate X and Z parts. The final column, labeled γ , contains a single bit that records the phase. Although our keys take positive m values, setting $m = 0$ recovers the original stabilizer tableau of AG.

A stabilizer circuit consisting of a sequence of Pauli measurements interspersed with Clifford unitaries is simulated by updating the tableau representation above according to specific quantum-mechanical rules. For Clifford unitaries, the updates are analogous to those for stabilizer states and are implemented via the action of a binary symplectic matrix. The main novelty of our simulator lies in the treatment of Pauli measurements. These are handled by update rules divided into four distinct cases:

Case I. Similar to the deterministic case of AG.

Cases II and III. New probabilistic updates (more involved).

Case IV. Similar to the probabilistic case of AG.

This four-case structure refines the improved stabilizer tableau method of AG. It is a novel feature of our phase-space tableau simulation and can be regarded as a next-generation extended stabilizer-theory tableau algorithm refining the original stabilizer tableau approach.

B. Simulation experiments

We now perform an in-depth analysis of the performance of the CNC tableau, drawing comparisons to the CHP simulator [3]—based on the AG tableau—where relevant. We highlight two key observations for interpreting the results of our simulation experiments:

(O1) A distinguishing feature of our tableau method is that it supports the measurement of general multiqubit Pauli observables. As a result, the CNC approach has an intrinsic quadratic-time overhead required to check which update rule to apply (Lemma 7) among Cases I–IV. In contrast, CHP requires only linear time in n to determine whether a computational-basis measurement will yield a deterministic or random outcome.

(O2) A novel ingredient in our simulation is the introduction of an additional parameter m . Two extreme cases are illustrative:

(i) When $m = 0$, the tableau reduces to the stabilizer tableau, and only Cases I and IV are performed, corresponding to deterministic and random outcomes.

(ii) When $m = n$, only Cases II and III are performed. These cases are more computationally demanding than Cases I and IV, and thus their higher frequency increases simulation overhead.

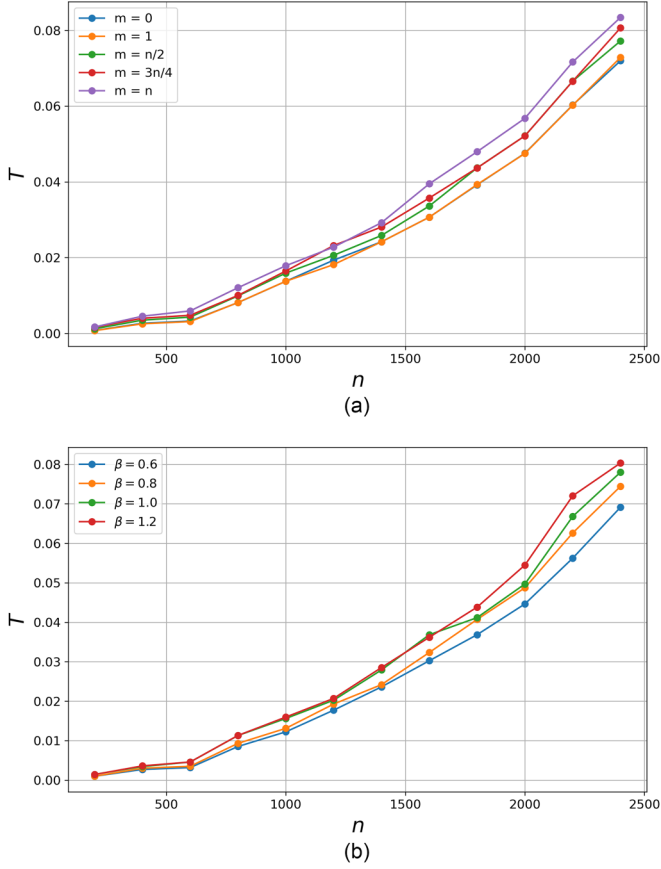


FIG. 1. Panels show the tested average measurement time T in seconds as a function of the qubit count n for random Clifford circuits, consisting of $\lfloor \beta n \log_2 n \rfloor$ gates on n qubits, simulated using our model [21]. (a) Fixed $\beta = 1$, varying m . (b) Varying β , fixed $m = n/2$. Simulations were run on an Apple M1 Pro MacBook Pro with 16 GB RAM.

To evaluate the performance of our simulator, we conduct the following computational experiment, following the setup of [3]:

- (1) Fix n , m , and $\beta > 0$.
- (2) Apply $\lfloor \beta n \log_2 n \rfloor$ Clifford gates— H_i (Hadamard), S_j (phase), and CX_{ij} (controlled-NOT)—each chosen with probability $1/3$.
- (3) Measure each qubit in the computational basis in sequential order.

In Ref. [3] the parameter β is varied, revealing a “phase transition” in which the simulation shifts from roughly linear to quadratic runtime. This behavior is due to the increased number of calls to the row product operation as β —that is, the number of Clifford unitaries—increases [22]. The CNC tableau simulation, by contrast, is less sensitive to changes in β as demonstrated in Fig. 1(b). This reduced sensitivity is largely due to the intrinsic quadratic overhead required simply to determine which update case to apply (O1). Nonetheless, the CNC simulation is not entirely insensitive to β , since in realistic scenarios, Case I and Case II updates are more efficient than Case III and Case IV updates, and the former occur less frequently as β increases (O2).

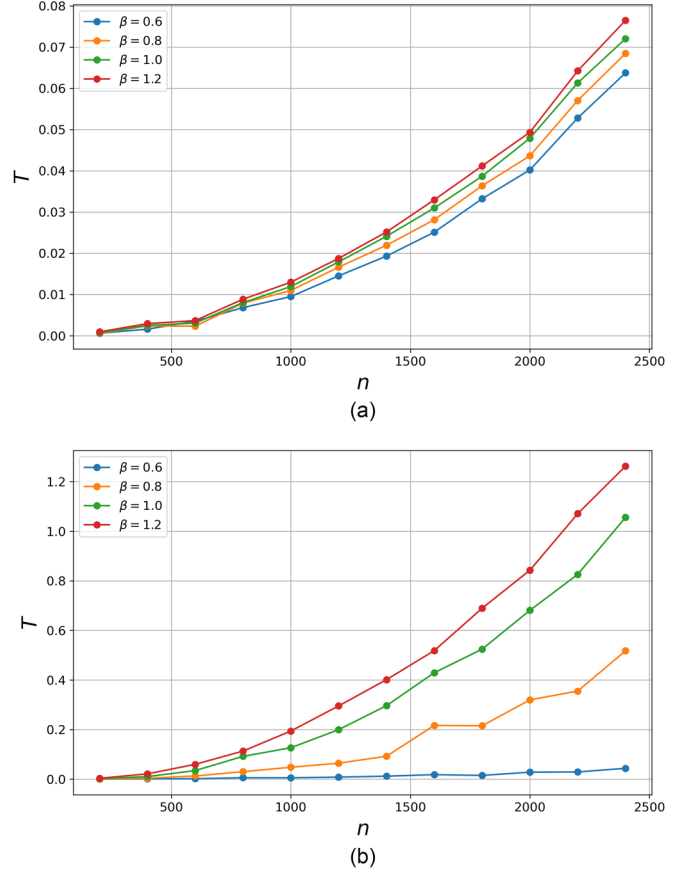


FIG. 2. Panels show the tested average measurement time T in seconds as a function of the qubit count n for random Clifford circuits, consisting of $\lfloor \beta n \log_2 n \rfloor$ gates on n qubits, simulated using our model [21] and (a) our model [21] with $m = 0$ and (b) CHP implementation of Gidney [23]. Both simulations were run on an Apple M1 Pro MacBook Pro with 16 GB RAM.

Next, we ran this experiment for a fixed value of $\beta = 1$ and with n ranging from 200 to 2400 in increments of 200. For each n we chose for m the values 0, 1, $n/4$, $n/2$, $3n/4$, and n . We find that increased m results in increased runtime, likely because with increased m there are more applications of the Case II and Case III update rules (O2), which are more complex as compared with Case I and Case IV, respectively; see Fig. 1(a).

When we fix $m = 0$, the CNC tableau has the same structure as the AG tableau, apart from an additional row that contains only zero entries in this case. We compared the performance of our CNC simulator with $m = 0$ to a Python implementation of the AG tableau due to Craig Gidney [23]. Our results are shown in Fig. 2. Numerically, we find that our implementation is considerably more performant (note the vertical-axis scaling), an interesting feature given that our Case I and Case IV parallel the deterministic and random update rules of stabilizer simulation.

III. RESULTS AND OUTLINE

The remainder of the paper is organized as follows: In Sec. IV, we review the necessary background on stabilizer

theory and CNC operators—the constituents of the phase space—and recall the robustness measure that quantifies the overhead of classical simulation.

In Sec. V, we connect the polyhedral phase-space picture to an operational model by introducing the CNC circuit class and identifying a family of variational noncontextual states that lie in the CNC polytope, and are therefore efficiently classically simulatable.

Section VI develops the symplectic theory of CNC operators, beginning with the Jordan–Wigner decomposition (Definition 4), which provides the symplectic foundation for our classical keys parametrized by (n, m) . We then give a different symplectic proof of the classification of maximal CNC operators (Theorem 1)—a result first observed in Ref. [11] and originally proved using graph-theoretic methods. Additional details of our proof are provided in Appendix A.

In Sec. VII, we first recall the original update rules of the phase-space framework [11], which divide into two cases. We then present our key observation that these rules can be refined into four distinct cases (Cases I–IV). Case III introduces an additional issue, overlooked in Ref. [11], in which the update may break the maximality property of the CNC operator, resulting not in a single phase space point but in a uniform probabilistic mixture [Eq. (35)]. This section forms the technical backbone of our tableau algorithm.

In Sec. VIII, we introduce the CNC tableau and present the full details of the four update rules, translating the technical results of the previous section into the tableau framework. These update rules are significantly more sophisticated than those of the stabilizer tableau simulation. To verify the rules and perform experiments, we implemented our tableau algorithm in Ref. [21]. A quick overview of the tableau structure and an analysis of its performance are given in Sec. II.

Section IX provides further details on our simulation method and the underlying algorithm including pseudocode presented in Appendix B.

IV. THE PHASE SPACE

In this section, we review stabilizer theory [24] and its extension based on closed noncontextual (CNC) operators, as introduced in Ref. [11].

A. Stabilizer theory

We consider a system of n qubits modeled by a Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. Given a bit string $a = (a_X, a_Z) \in E_n$, where $E_n := \mathbb{Z}_2^n$, a Pauli operator is a Hermitian operator of the form

$$T_a = i^{\phi(a)} X^{a_X} Z^{a_Z}, \quad (1)$$

where $\phi : E_n \rightarrow \mathbb{Z}_4$ is a phase function that is conventionally chosen to be $\phi(a) = a_X a_Z$. Here, we have adopted the notation $X^{a_X} = X^{(a_X)_1} \otimes \dots \otimes X^{(a_X)_n}$ with a similar notation for Z^{a_Z} .

The vector space E_n has a symplectic form defined by $[a, b] = a_X b_Z + a_Z b_X$ where the dot product is the usual dot product over $\mathbb{Z}_2$. Pauli operators T_a and T_b then satisfy $T_a T_b = (-1)^{[a, b]} T_b T_a$ and thus commute if and only if the corresponding symplectic form $[a, b] = 0$ and anticommute

otherwise. The product of commuting Pauli operators satisfies $T_a T_b = (-1)^{\beta(a, b)} T_{a+b}$; see for example Ref. [25]. Using our phase convention β has an explicit form given by

$$\beta(a, b) = \frac{\phi(a) + \phi(b) + 2(a_Z b_X) - \phi(a \oplus b)}{2}, \quad (2)$$

where $\oplus$ denotes addition in $\mathbb{Z}_2$ and other operations are over $\mathbb{Z}_4$.

The Pauli group P_n is given by the set of operators $\{\pm 1, \pm i\} \times \{T_a : a \in E_n\}$ with matrix multiplication as the group operation. Its quotient with respect to its center $\langle i\mathbb{1} \rangle$ is denoted $\bar{P}_n$. This group is isomorphic to E_n . The n -qubit Clifford group is the normalizer of the Pauli group with phases quotiented out:

$$\text{Cl}_n := \frac{\{U \in \mathcal{U}(\mathcal{H}) : U P_n U^\dagger = P_n\}}{\langle e^{i\theta} \mathbb{1} : \theta \in \mathbb{R} \rangle}. \quad (3)$$

We can further identify the quotient group $\text{Cl}_n / \bar{P}_n$ with the symplectic group $\text{Sp}_{2n}(\mathbb{Z}_2)$, the matrix group of invertible linear maps $E_n \rightarrow E_n$ that preserve the symplectic form. That is, there is a surjective group homomorphism

$$\varphi : \text{Cl}_n \rightarrow \text{Sp}_{2n}(\mathbb{Z}_2).$$

In fact, any Clifford unitary U_S acts by conjugation on Pauli operators according to

$$U_S T_a U_S^\dagger = \pm T_{S(a)},$$

where S is a symplectic map.

A subspace $I \subset E_n$ is called isotropic if for every $a, b \in I$ we have that $[a, b] = 0$. The orthogonal complement $I^\perp$ of an isotropic subspace I consists of all elements that commute with I . Note that $I \subset I^\perp$. Moreover, if $\dim(I) = n - m$ then $\dim(I^\perp) = n + m$ and $\dim(I) + \dim(I^\perp) = 2n = \dim(E_n)$. An isotropic subspace is called maximal if $I = I^\perp$, i.e., it is self-dual. A stabilizer code is a projector,

$$\Pi_I^s = \frac{1}{|I|} \sum_{a \in I} (-1)^{s(a)} T_a,$$

where I is an isotropic and $s : I \rightarrow \mathbb{Z}_2$ is a value assignment. When I is maximal, then Π_I^s is precisely a stabilizer state. At the other extreme, when $I = \langle a \rangle$, we denote the corresponding projector as Π_a^s , where s represents the value in $\mathbb{Z}_2$ assigned to a . By Witt's lemma, the symplectic group acts on the set of isotropic subspaces of fixed dimension in a transitive way. Similarly, the action of the Clifford group on the stabilizer projectors of fixed rank is transitive.

B. Closed noncontextual operators

Closed noncontextual operators provide a natural extension of the stabilizer theory.

Definition 1. A subset $\Omega \subset E_n$ is called closed if, for every pair $a, b \in \Omega$ satisfying $[a, b] = 0$, it follows that $a + b \in \Omega$. For a closed set Ω , a function $\gamma : \Omega \rightarrow \mathbb{Z}_2$ is called a value assignment if, for all commuting $a, b \in \Omega$, the relation

$$\gamma(a + b) = \gamma(a) + \gamma(b) + \beta(a, b), \quad (4)$$

holds, where $\beta(a, b)$ is defined in Eq. (2). A closed subset $\Omega \subset E_n$ that admits such a value assignment is called a closed noncontextual (CNC) set.

The structure of CNC sets was extensively studied in Ref. [11]; see also Ref. [14]. We call a CNC set maximal if it is not contained in any other CNC set. Given a CNC set Ω and a value assignment $\gamma : \Omega \rightarrow \mathbb{Z}_2$ (sometimes called a CNC pair), we define a unit trace Hermitian operator

$$A_\Omega^\gamma := \frac{1}{2^n} \sum_{a \in \Omega} (-1)^{\gamma(a)} T_a. \quad (5)$$

Operators of this form are called CNC operators. Stabilizer states are particular examples of CNC operators.

C. Classical simulation

CNC operators span the space of unit trace Hermitian operators. In particular, any quantum state ρ can be expressed by an affine mixture of maximal CNC operators. Thus, these operators form a Clifford covariant finite state space that can be used for a sampling-based classical simulation algorithm [11]. This algorithm belongs to a class of sampling algorithms on the set of vertices of an operator-theoretic polytope. Let SP_n denote the stabilizer polytope defined as the convex hull of n -qubit stabilizer states. The dual of this polytope, denoted by Λ_n , provides a probabilistic algorithm for universal quantum computation [26]. Furthermore, it was also shown for all n that CNC operators with maximal Ω are vertices of the Λ polytopes. Let CP_n denote the CNC polytope defined as the convex hull of the CNC operators. It is not difficult to show (see Corollary 2) that stabilizer states are not maximal CNC operators and thus they lie in CP_n , so that

$$\text{SP}_n \subset \text{CP}_n \subset \Lambda_n.$$

Classical simulation based on the CNC state space extends and improves the classical simulation based on stabilizer states. As demonstrated in Ref. [11], the update of CNC operators under Clifford unitaries and Pauli measurements can be efficiently simulated on a classical computer. In Ref. [11], they demonstrate this using the characterization of CNC sets (described in more detail below) as a union of isotropic subspaces, each of which can essentially be updated via a stabilizer tableau. The goal of the present work is to refine the analysis of [11], making the implementation and computational efficiency of the update rules more explicit.

The amount of negativity required to represent any quantum state ρ as an affine mixture of maximal CNC operators is less than when using stabilizer states. This is quantified by the robustness measure

$$\mathfrak{R}(\rho) := \min_{W : \rho = \sum_{\alpha} W_\alpha A_\alpha} \|W\|_1, \quad (6)$$

where $\rho = \sum_{\alpha} W_\alpha A_\alpha$ is an affine mixture and the sum is over CNC pairs labeled by α . The robustness is a monotone under Clifford unitaries and Pauli measurements [11] and is upper bounded by $\mathfrak{R}_S$, the analogous measure using stabilizer states [18]. In fact, for any n -qubit quantum state ρ , these measures are related by [11]

$$\mathfrak{R}(\rho) \leq \mathfrak{R}_S(\rho) \leq (4n+1)\mathfrak{R}(\rho).$$

By the arguments in Ref. [17], the CNC phase space therefore yields an (up to quadratic in n) improvement in sampling overhead for Born rule estimation over the stabilizer state

TABLE I. Robustness values for the CNC and stabilizer phase spaces with $n \leq 4$ qubits. CNC robustness for $n = 4$ is not exact and was obtained using a subset of CNC operators in the optimization problem. For best known upper-bounds on the robustness of magic, see Ref. [27].

	$ T\rangle$	$ T\rangle^{\otimes 2}$	$ T\rangle^{\otimes 3}$	$ T\rangle^{\otimes 4}$
$\mathfrak{R}_C$	1.000	1.000	1.283	≤ 2.172
$\mathfrak{R}_S$	1.414	1.748	2.219	2.863

phase space. Furthermore, all operators lying in CP_n can be efficiently classically simulated in the weak sense and in Sec. V we identify a class of nonstabilizer quantum states lying in CP_n for all n . These gains in efficiency motivate the implementation of our tableau algorithm.

V. EXTENDED STABILIZER CIRCUITS

A. The CNC circuit class

On the operational side, the stabilizer circuit class can be extended, reflecting the geometric picture on the polytope theoretic side. We define the CNC circuit class to consist of those quantum circuits initiated at a quantum state contained in the CNC polytope, transformed under Clifford unitaries, and operated on by (adaptive) Pauli measurements. Stabilizer circuits are contained in this class. Identifying this circuit class and comparing it to other efficiently simulatable circuit classes, such as matchgate circuits [28–30], is of fundamental importance. The polyhedral point of view provides a geometric approach to determining the boundary of this new efficiently simulatable circuit class extending the stabilizer circuits.

The description of the CNC polytope given above is a V description, since it is specified as the convex hull of finitely many operators. An H description, by contrast, describes the polytope via its facet-defining inequalities, and can offer a more direct characterization of the quantum states satisfying these constraints. Passing between these descriptions is a duality problem that is notoriously difficult in general. Since the number of CNC operators exceeds the number of stabilizer states, one expects this facet or vertex enumeration task to be harder than the corresponding enumeration for the Λ polytopes, which are dual to the stabilizer polytope.

As noted in Ref. [11], the families of quantum states currently known to lie in the CNC polytope are quite restricted. For single-qubit systems, the CNC polytope is a cube containing the Bloch sphere; in particular, every single-qubit quantum state lies in the CNC polytope. For two qubits, the magic state $|T\rangle^{\otimes 2}$ is known to lie in the CNC polytope; see Table I. More generally, if ρ_m belongs to the m -qubit CNC polytope, then

$$U(\Pi_{n-m} \otimes \rho_m)U^\dagger$$

belongs to the n -qubit CNC polytope for any n -qubit Clifford unitary U and any $(n-m)$ -qubit stabilizer projector Π_{n-m} . This follows from the fact, proved in Ref. [31], that for any maximal m -qubit CNC operator A_Ω^γ , the operator

$$U(\Pi_{n-m} \otimes A_\Omega^\gamma)U^\dagger \quad (7)$$

is a maximal n -qubit CNC operator. Beyond these specific cases, relatively little is known about which quantum states lie in this polytope.

B. Variational noncontextual states

Next, we identify a family of quantum states in the CNC polytope that generalizes the Bloch-sphere picture for single qubits. These states are constructed from the non-contextual states appearing in contextual variational eigensolvers [16,32].

We restrict attention to maximal CNC operators A_Ω^γ whose nontrivial Pauli terms in the expansion pairwise anticommute. Let $\{a_1, a_2, \dots, a_{2n+1}\}$ denote the corresponding nonzero labels in Ω . (Equivalently, $\Omega \setminus \{0\}$ is an anticommuting set in the sense of Sec. VI.) Define

$$A(f) = \frac{1}{2^n} \left(\mathbb{1} + \sum_{i=1}^{2n+1} f(a_i) T_{a_i} \right).$$

Then

$$\begin{aligned} A(f) &= \frac{1}{2^n} \left(\mathbb{1} + \sum_{i=1}^{2n+1} f(a_i) T_{a_i} \right) \\ &= \frac{1}{2^n} (\mathbb{1} + \|f\| A(\hat{f})) \\ &= \lambda_0 \Pi^0(\hat{f}) + \lambda_1 \Pi^1(\hat{f}), \end{aligned}$$

where $\|f\|$ is the Euclidean norm, $\hat{f}(a_i) = f(a_i)/\|f\|$, $\lambda_s = (1 + (-1)^s \|f\|)/2^n$, and $\Pi^s(\hat{f})$ denotes the projector onto the λ_s eigenspace of $A(\hat{f})$. Here we use that $A(\hat{f})^2 = \mathbb{1}$, so its eigenvalues are ± 1 . The operator $A(f)$ is a quantum state if and only if all its eigenvalues are non-negative; equivalently,

$$\|f\| \leq 1. \quad (8)$$

We apply this criterion to probabilistic mixtures of CNC operators A_Ω^γ as the value assignment γ ranges over all choices. Each assignment can be identified with an element of $\mathbb{Z}_2^{2n+1}$, since we may freely assign signs ± 1 to the nontrivial Pauli operators appearing in the expansion of a maximal CNC operator.

Let p be a probability distribution on the set of all value assignments on Ω ; equivalently, a distribution on $\mathbb{Z}_2^{2n+1}$. For each $i \in \{1, \dots, 2n+1\}$, let p_i denote the i th marginal of p , where $p_i(0)$ is defined by

$$\sum_{s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_{2n+1} \in \mathbb{Z}_2} p(s_1, \dots, s_{i-1}, 0, s_{i+1}, \dots, s_{2n+1}).$$

Then p_i is a probability distribution on $\mathbb{Z}_2$, with $p_i(1) = 1 - p_i(0)$.

Proposition 1. The probabilistic mixture

$$A_p = \sum_{\gamma} p(\gamma) A_\Omega^\gamma$$

is a quantum state if and only if the expectation vector $E(p) = (E_1, \dots, E_{2n+1})$, with $E_i := 2 p_i(0) - 1$, satisfies

$$\|E(p)\| \leq 1. \quad (9)$$

Proof. We write A_p in the form $A(f)$, where $f(a) := \sum_{\gamma} p(\gamma) (-1)^{\gamma(a)}$. Then

$$\begin{aligned} \|f\| &= \sqrt{\sum_{a \in \Omega \setminus \{0\}} \left(\sum_{\gamma} p(\gamma) (-1)^{\gamma(a)} \right)^2} \\ &= \sqrt{\sum_{i=1}^{2n+1} (p_i(0) - p_i(1))^2} \\ &= \sqrt{\sum_{i=1}^{2n+1} E_i^2} \\ &= \|E(p)\|. \end{aligned}$$

The claim now follows from Eq. (8). $\blacksquare$

Note that as p ranges over all probability distributions on $\mathbb{Z}_2^{2n+1}$, the expectation vectors $E(p)$ fill out the hypercube $[-1, 1]^{2n+1} \subset \mathbb{R}^{2n+1}$. Thus, the condition (9) is simply the requirement that $E(p)$ lies in the unit ball $B^{2n+1} \subset \mathbb{R}^{2n+1}$.

We now describe the boundary case $\|E(p)\| = 1$, i.e., $E(p) \in S^{2n}$. In this case, the eigenvalues of $A(f)$ are

$$\lambda_0 = \frac{1}{2^{n-1}} \text{ and } \lambda_1 = 0,$$

and hence the corresponding quantum state can be written as

$$A_p = \frac{\Pi^0(E(p))}{2^{n-1}}, \quad E(p) \in S^{2n}. \quad (10)$$

Projectors of the form $\Pi^0(E)$ with $\|E\| = 1$ coincide with the projectors onto the noncontextual subspaces appearing in contextual variational eigensolvers. For $n \geq 2$ these are not stabilizer projectors [16], while for $n = 1$ this construction reduces to the familiar Bloch-sphere picture with S^2 .

Moreover, the states in Eq. (10) do not, in general, lie in the stabilizer polytope SP_n . To see this, note that for any stabilizer state σ , the expectation vector $(\text{Tr}(\sigma T_{a_1}), \dots, \text{Tr}(\sigma T_{a_{2n+1}}))$ lies in $\{\pm e_1, \dots, \pm e_{2n+1}\}$, where e_i denotes the i th standard basis vector. Therefore, for any stabilizer mixture $\rho = \sum_k q_k \sigma_k$ we have

$$\begin{aligned} \sum_{i=1}^{2n+1} |\text{Tr}(\rho T_{a_i})| &= \sum_{i=1}^{2n+1} \left| \sum_k q_k \text{Tr}(\sigma_k T_{a_i}) \right| \\ &\leq \sum_k q_k \sum_{i=1}^{2n+1} |\text{Tr}(\sigma_k T_{a_i})| \leq 1, \end{aligned}$$

by the triangle inequality and the fact that each stabilizer state contributes at most one nonzero entry, of magnitude 1. Applying this to $\rho = A_p$ yields the necessary condition $\sum_{i=1}^{2n+1} |E_i(p)| \leq 1$ for A_p to lie in the stabilizer polytope. Under the boundary constraint $\|E(p)\| = 1$, this inequality holds if and only if $E(p) = \pm e_j$ for some j . In this special case, A_p is a Pauli eigenstate (hence a stabilizer state). By contrast, if $E(p)$ has at least two nonzero components, then $\sum_i |E_i(p)| > 1$, and so A_p cannot be expressed as a probabilistic mixture of stabilizer states; equivalently, it lies outside SP_n .

We can extend the above construction to any maximal CNC operator A_Ω^γ of type (n, m) , where $1 \leq m \leq n$, using the closure property in Eq. (7). In particular, any quantum state that arises as a probabilistic mixture of type (n, m) CNC operators can be written as

$$\rho(p) = \frac{1}{2^n} U(\Pi_{n-m} \otimes \Pi_m(p)) U^\dagger, \quad (11)$$

where $\Pi_m(p) := \Pi^0(E(p))$ for some probability distribution p on $\mathbb{Z}_2^{2m+1}$ with $E(p) \in S^{2m}$. Since the state $\rho(p)$ admits a probabilistic CNC representation, any circuit in the CNC circuit class initialized in $\rho(p)$ can be efficiently simulated by the CNC tableau algorithm.

VI. STRUCTURE OF THE PHASE SPACE

In this section, we introduce the symplectic preliminaries required for constructing the tableau algorithm. The key concept is a different decomposition of the symplectic vector space $E_n = \mathbb{Z}_2^{2n}$ into stabilizer, destabilizer, and Jordan–Wigner components, which we refer to as a Jordan–Wigner decomposition. This decomposition allows us to encode additional information in our tableau, extending beyond the stabilizer tableau.

A. Jordan–Wigner decomposition

A set of elements

$$\mathcal{S} = \{a_1, \dots, a_N\} \subset E_n$$

is called an anticommuting set if its elements pairwise anticommute, i.e., $[a_i, a_j] = 1$ for all $i \neq j$. An anticommuting set is called a maximal anticommuting set if it is not strictly contained in any other anticommuting set.

We begin with some basic observations on maximal anticommuting sets; see also Ref. [33].

Lemma 1. For an anticommuting set $\mathcal{S}$ with $N = |\mathcal{S}|$ even, the sum $b = \sum_{i=1}^N a_i$ is nonzero and adding this element to the set results in a maximal anticommuting set.

Proof. If $N = 0$, then $\mathcal{S} = \{0\}$ is clearly maximal. So, assume $N > 0$. If $b = 0$ then $a_N = \sum_{i=1}^{N-1} a_i$ and $[a_N, a_i] = \sum_{j=1}^{N-1} [a_j, a_i] = 0$ for $i = 1, \dots, N-1$. Therefore b is nonzero. Adding this element to the set gives a maximal anticommuting set since any other element that anticommutes with all the a_i 's commutes with b . ■

For an anticommuting set $\mathcal{S}$ and an element $c \in E_n$, we define

$$\begin{aligned} \mathcal{A}_c &= \{a_i \mid [a_i, c] = 1, i = 1, \dots, N\}, \\ \mathcal{C}_c &= \{a_i \mid [a_i, c] = 0, i = 1, \dots, N\}. \end{aligned} \quad (12)$$

Note that the definition of these sets also depends on $\mathcal{S}$, which is omitted from the notation for clarity.

Proposition 2. An anticommuting set $\mathcal{S}$ is maximal if and only if $\sum_{i=1}^N a_i = 0$. Furthermore, in this case, N must be odd.

Proof. First, assume that $\mathcal{S}$ is maximal and $b = \sum_{i=1}^N a_i$. Lemma 1 implies that N is odd. Let $c \in E_n$ be such that

$[b, c] = 1$. Then,

$$[b, c] = \sum_{i=1}^N [a_i, c] = \sum_{a \in \mathcal{A}_c} [a, c] = 1,$$

which implies that $|\mathcal{A}_c|$ is odd and $|\mathcal{C}_c|$ is even since N is odd. The element $d = c + \sum_{a \in \mathcal{C}_c} a$ satisfies $[d, a_i] = 1$ for all $i = 1, \dots, N$ and $d \notin \mathcal{S}$. This contradicts the maximality of $\mathcal{S}$. Hence, no such c exists, and this means $b = 0$. For the converse, assume that $b = \sum_{i=1}^N a_i = 0$. By Lemma 1, N must be odd. Since we have $a_N = \sum_{i=1}^{N-1} a_i$ and $|\mathcal{S} \setminus \{a_N\}|$ is even, Lemma 1 implies that $\mathcal{S}$ is maximal. ■

Notice that $\mathcal{S}$ is linearly dependent if and only if there exists a nonempty $I \subset \{1, \dots, N\}$ such that $\sum_{i \in I} a_i = 0$. The latter condition is equivalent to $\{a_i\}_{i \in I}$ being a maximal anticommuting set by Proposition 2. Therefore, we have the following:

(1) If $\mathcal{S}$ is maximal, then any subset of $\mathcal{S}$ with size $N - 1$ is linearly independent.

(2) If $\mathcal{S}$ is not maximal, then $\mathcal{S}$ is linearly independent.

As a consequence, we can see that the maximum size of an anticommuting set is $2n + 1$. These max-sized anticommuting sets are particularly important for our discussion.

Definition 2. An anticommuting set of $2n + 1$ elements, $\{a_1, \dots, a_{2n+1}\}$, is called a Jordan–Wigner (JW) set, and its elements are referred to as JW elements. Any $2n$ JW elements form a basis of E_n that we call a JW basis.

We typically choose the first $2n$ JW elements as the basis so that the remaining element a_{2n+1} is the sum of the JW basis elements.

Lemma 2. Any anticommuting set that is not maximal can be extended to a JW set.

Proof. Let $\mathcal{S}_N = \{a_1, \dots, a_N\}$ be a nonmaximal anticommuting set. If $N = 2n$ then the result directly follows from Lemma 1. Therefore, we are done if we can show that $\mathcal{S}_N$ can be extended to an anticommuting set $\mathcal{S}_{N+1} = \mathcal{S}_N \cup \{a_{N+1}\}$ that is not maximal when $N < 2n$. If N is odd, there exists $a_{N+1} \in E_n$ such that $[a_{N+1}, a_i] = 1$ for all $i \in \{1, \dots, N\}$ since $\mathcal{S}_N$ is not maximal. Since $N + 1$ is even, the extended set $\mathcal{S}_{N+1}$ is not maximal by Lemma 1. Now if N is even, then we can find an element $c \in E_n \setminus \langle a_1, \dots, a_N \rangle$ since $N < 2n$. We can construct

$$a_{N+1} = \begin{cases} c + \sum_{a \in \mathcal{A}_c} a & \text{if } |\mathcal{C}_c| \text{ is odd} \\ c + \sum_{a \in \mathcal{C}_c} a & \text{if } |\mathcal{C}_c| \text{ is nonzero and even} \\ c & \text{if } |\mathcal{C}_c| = 0. \end{cases}$$

It is easy to see that $[a_{N+1}, a_i] = 1$ for all $i \in \{1, \dots, N\}$. So, $\{a_1, \dots, a_N, a_{N+1}\}$ is an anticommuting set. Moreover, $\sum_{i=1}^{N+1} a_i \neq 0$ since otherwise it would imply that $c \in \langle a_1, \dots, a_N \rangle$. Hence, $\{a_1, \dots, a_N, a_{N+1}\}$ is not maximal. ■

Definition 3. The Jordan–Wigner (JW) transform of a symplectic basis $\{e_i, f_i\}_{i=1}^n$ is given by the anticommuting set:

$$a_1 = e_1, \quad a_2 = f_1,$$

and for $i = 2, \dots, n$,

$$a_{2i-1} = e_i + \sum_{k=1}^{i-1} (e_k + f_k), \quad a_{2i} = f_i + \sum_{k=1}^{i-1} (e_k + f_k). \quad (13)$$

Conversely, the inverse JW transform of an anticommuting set $\{a_1, \dots, a_{2n}\}$ yields the symplectic basis $\{e_i, f_i\}_{i=1}^n$, defined by

$$e_1 = a_1, \quad f_1 = a_2,$$

and for $i = 2, \dots, n$,

$$e_i = a_{2i-1} + \sum_{k=1}^{2i-2} a_k, \quad f_i = a_{2i} + \sum_{k=1}^{2i-2} a_k. \quad (14)$$

These transformations are inverses of each other. So, $\langle a_1, \dots, a_{2n} \rangle = \langle e_1, f_1, \dots, e_n, f_n \rangle$ in both cases. The JW elements associated with the canonical symplectic basis $\{z_i, x_i\}_{i=1}^{2n}$ are called the canonical JW elements. Moreover,

$$W = \langle a_1, \dots, a_{2k} \rangle \quad (15)$$

is a symplectic vector space of dimension $2k$.

Lemma 3. Any $b \in E_n$ can be expressed in the JW basis as

$$b = \sum_{k=1}^{2n} v_k a_k, \quad (16)$$

where $v_k = [a_{2n+1} + a_k, b]$.

Proof. We have

$$\begin{aligned} [a_{2n+1} + a_k, b] &= [a_1, b] + \dots + [a_{k-1}, b] + [a_{k+1}, b] + \dots + [a_{2n}, b] \\ &= v_1 + \dots + v_{k-1} + v_{k+1} + \dots + v_{2n} + (2n-1) \sum_{i=1}^{2n} v_i \\ &= v_1 + \dots + v_{k-1} + v_{k+1} + \dots + v_{2n} + \sum_{i=1}^{2n} v_i \\ &= v_k, \end{aligned}$$

where we used $a_{2n+1} = \sum_{k=1}^{2n} a_k$ and $[a_k, b] = v_k + \sum_{i=1}^{2n} v_i$ in the second equality. ■

It is convenient to write $v_{2n+1} = [a_{2n+1}, b]$, so that $v_k = v_{2n+1} + [a_k, b]$.

Corollary 1. For $b \in E_n$, let $\mathcal{A}_b$ and $\mathcal{C}_b$ be the sets defined in Eq. (12) with respect to a JW set. Then,

$$b = \sum_{a \in \mathcal{A}_b} a = \sum_{a' \in \mathcal{C}_b} a',$$

and $|\mathcal{A}_b|$ is even.

Proof. Using Eq. (16), we have

$$\begin{aligned} b &= \sum_{k=1}^{2n} v_k a_k \\ &= v_{2n+1} \sum_{k=1}^{2n} a_k + \sum_{k=1}^{2n} [a_k, b] a_k \\ &= v_{2n+1} a_{2n+1} + \sum_{a \in \bar{\mathcal{A}}_b} a, \end{aligned}$$

where $\bar{\mathcal{A}}_b = \mathcal{A}_b \setminus \{a_{2n+1}\}$. Now if $a_{2n+1} \notin \mathcal{A}_b$, then $v_{2n+1} = 0$ and $\mathcal{A}_b = \bar{\mathcal{A}}_b$. So, $b = \sum_{a \in \mathcal{A}_b} a$. If $a_{2n+1} \in \mathcal{A}_b$, then

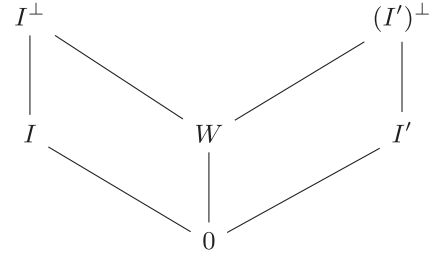


FIG. 3. The intersections and sums of the subspaces in a JW decomposition: $W \cap I = 0$, $W \oplus I = I^\perp$, $W \cap I' = 0$, and $W \oplus I' = (I')^\perp$

$v_{2n+1} = 1$ and $\mathcal{A}_b = \bar{\mathcal{A}}_b \cup \{a_{2n+1}\}$ which gives

$$b = a_{2n+1} + \sum_{a \in \bar{\mathcal{A}}_b} a = \sum_{a \in \mathcal{A}_b} a.$$

Since $\sum_{i=1}^{2n+1} a_i = 0$, we have $\sum_{a \in \mathcal{A}_b} a + \sum_{a' \in \mathcal{C}_b} a' = 0$, and this implies $b = \sum_{a' \in \mathcal{C}_b} a'$. Lastly, if $|\bar{\mathcal{A}}_b|$ is odd, then $a_{2n+1} \in \mathcal{A}_b$ since $a_{2n+1} = \sum_{i=1}^{2n} a_i$. If $|\bar{\mathcal{A}}_b|$ is even, then $a_{2n+1} \notin \mathcal{A}_b$. Hence, $|\mathcal{A}_b|$ is even.

Now, we come to the main definition of this section.

Definition 4. A Jordan-Wigner decomposition for E_n of type m consists of

- (1) an isotropic subspace $I_{n-m} \subset E_n$ of dimension $n-m$,
- (2) a subspace $W_{2m} \subset I^\perp$ of dimension $2m$ such that $W_{2m} \cap I_{n-m} = 0$ and $W_{2m} \oplus I_{n-m} = I_{n-m}^\perp$,
- (3) an isotropic subspace $I'_{n-m} \subset E_n$ of dimension $n-m$ such that $I'_{n-m} \cap I_{n-m}^\perp = 0$ and $I'_{n-m} \oplus W_{2m} = (I'_{n-m})^\perp$.

The subspaces in the pair (I_{n-m}, I'_{n-m}) will be referred to as conjugate isotropic subspaces.

We write

$$E_n = I_{n-m} \oplus W_{2m} \oplus I'_{n-m}$$

for a decomposition of type m . Subscripts indicating the dimensions will usually be omitted to simplify the notation. The relationship between these components is depicted in Fig. 3. Note that we have

$$I^\perp \cap (I')^\perp = W.$$

Lemma 4. Given conjugate isotropic subspaces I and I' , we can choose a basis $\{e_1, \dots, e_{n-m}\}$ and $\{f_1, \dots, f_{n-m}\}$, respectively, such that

$$[e_i, f_j] = \delta_{ij},$$

where $1 \leq i, j \leq n-m$.

Proof. It is a well-known fact in stabilizer theory that for $b \notin I^\perp$, we can make a change of basis so that only one of the basis vectors anticommute with b , and the rest commute [24]. Let λ be the minimum of the set $\{1 \leq i \leq n-m \mid [e_i, b] = 1\}$. Then, we can write

$$\begin{aligned} I &= \langle \bar{e}_1, \dots, \bar{e}_{n-m} \rangle \\ \bar{e}_i &= \begin{cases} e_i + e_\lambda & \text{if } [e_i, b] = 1 \text{ and } i \neq \lambda \\ e_i & \text{otherwise.} \end{cases} \end{aligned} \quad (17)$$

We apply this change of basis first with $b = f_1$, and apply another change of basis with $b = f_2$, and so on. Note that since

f_i 's pairwise commute we obtain the desired commutation relations. ■

Such a basis will be called a conjugate basis. Note that given $\{e_i\}_{i=1}^{n-m}$ the conjugate basis is unique.

The JW decomposition of type m associated with the choices $I = \langle z_1, \dots, z_{n-m} \rangle$, $I' = \langle x_1, \dots, x_{n-m} \rangle$, and $W = \langle a_1, \dots, a_{2m} \rangle$, where the JW basis is obtained from $\{z_{n-m+i}, x_{n-m+i}\}_{i=1}^m$ using Eq. (13), is called the canonical JW decomposition. Note that W can be identified by E_m once the indices of the basis vectors are shifted down by $n - m$.

Lemma 5. Let $E = I \oplus W \oplus I'$ be a JW decomposition of type m and $\{a_i\}_{i=1}^{2m}$ be a JW basis for $W \cong E_m$. For $b \in W$, write $\mathcal{A}_b = \{a_1, \dots, a_{2t}\} \subset W$ and define $\{\hat{a}_i\}_{i=1}^{2t}$ by

$$\hat{a}_1 = b \text{ and } \hat{a}_i = a_i, \quad i = 2, \dots, 2t.$$

Let $\{\hat{e}_i, \hat{f}_i\}_{i=1}^{2t}$ be the symplectic basis of W associated with $\{\hat{a}_i\}_{i=1}^{2t}$ given by Eq. (14). Then, there exists a JW decomposition for W of type $m - t$:

$$W = J \oplus \bar{W} \oplus J',$$

such that

- (1) $J = \langle \hat{e}_1, \dots, \hat{e}_t \rangle$,
 - (2) $J' = \langle \hat{f}_1, \dots, \hat{f}_t \rangle$,
 - (3) $\{b + a_{2t+1}, \dots, b + a_{2m}\}$ is a JW basis of $\bar{W} \cong E_{m-t}$.
- Moreover, $J \oplus J' = \langle a_1, \dots, a_{2t} \rangle$.

Proof. Notice that $\{\hat{a}_i\}_{i=1}^{2t}$ is an anticommuting set and $a_1 = \sum_{i=1}^{2t} \hat{a}_i$ since $\sum_{a \in \mathcal{A}_b} a = b$. Explicitly, $\hat{e}_i$ and $\hat{f}_i$ are defined as follows:

$$e_1 = \hat{a}_1 \text{ and } e_i = \hat{a}_{2i-1} + \sum_{k=1}^{2i-2} \hat{a}_k, \quad (18)$$

and

$$f_1 = \hat{a}_2 \text{ and } f_i = \hat{a}_{2i} + \sum_{k=1}^{2i-2} \hat{a}_k, \quad (19)$$

where $i = 2, \dots, t$. So, J and J' are conjugate isotropics and $S = J \oplus J' = \langle \hat{a}_1, \dots, \hat{a}_{2t} \rangle = \langle a_1, \dots, a_{2t} \rangle$ is a symplectic subspace with a symplectic basis $\{\hat{e}_i, \hat{f}_i\}_{i=1}^t$. Since $\{a_i\}_{i=1}^{2m}$ is a basis of W , we have $W = S \oplus \langle a_{2t+1}, \dots, a_{2m} \rangle$. Since $b \in S$ we have $W = S \oplus \bar{W}$ where $\bar{W} = \langle b + a_{2t+1}, \dots, b + a_{2m} \rangle$. Lastly, notice that $\{b + a_{2t+1}, \dots, b + a_{2m}\}$ is pairwise anticommuting since $\mathcal{C}_b = \{a_{2t+1}, \dots, a_{2m+1}\}$. Hence, $\{b + a_{2t+1}, \dots, b + a_{2m}\}$ is a JW basis for $\bar{W}$. ■

Given an isotropic subspace $I = \langle e_1, \dots, e_{n-m} \rangle$ a JW decomposition (Definition 4) can be obtained by the following (efficient) procedure: To begin, consider a linear map $L : \mathbb{Z}_2^{2n} \rightarrow \mathbb{Z}_2^{n-m}$ such that $\ker(L) = I^\perp$. The corresponding $(n - m) \times 2n$ matrix over $\mathbb{Z}_2$ will also be denoted by L . We now outline an efficient procedure for obtaining a JW decomposition. To construct a basis for $I^\perp = I \oplus W$:

(1) Use Gaussian elimination (GE) to solve the set of homogeneous linear equations $Lx = 0_{n-m}$. This yields a set of vectors $\{v_1, \dots, v_{n+m}\}$.

(2) Use GE once again to identify $2m$ vectors $\{v_{i_1}, \dots, v_{i_{2m}}\}$ from $\{e_1, \dots, e_{n-m}\}$ that are linearly independent.

We now have that $I^\perp = \langle e_1, \dots, e_{n-m}, v_{i_1}, \dots, v_{i_{2m}} \rangle$. Due to the rank-nullity theorem, we have that $E_n = \ker(L) \oplus$

$\text{Im}(L^T)$ wherein we identify the $(n - m)$ -dimensional subspace $I' = \text{Im}(L^T)$ as the basis conjugate to I . In practice, we can construct a basis for I' as follows:

(1) Use GE to determine the linearly independent columns $\{\tilde{f}_1, \dots, \tilde{f}_{n-m}\}$ of L .

(2) Obtain a basis $\{f_1, \dots, f_{n-m}\} \subset I'$ according to $f_i = L^T \tilde{f}_i$ since $L|_{I'}$ is injective.

As described in Eq. (14), we can obtain a JW set for W by first identifying a symplectic basis $W = \langle e'_1, \dots, e'_m, f'_1, \dots, f'_m \rangle$, then constructing a_i 's for $i = 1, \dots, 2m$ and setting $a_{2m+1} = \sum_{i=1}^{2m} a_i$ (Lemma 1). This can be achieved efficiently using symplectic Gram-Schmidt orthogonalization (SGSO) [34,35]. Moreover, the procedure outlined in Lemma 4 is a special case of SGSO in which the decomposition $W^\perp = I \oplus I'$ is given, and thus is also efficient. The protocol as a whole is, in turn, efficient as well since the most expensive subroutines are GE and SGSO, both of which are polynomial-time algorithms.

B. Classification of CNC sets

The original classification of CNC sets in Ref. [11] (see also Ref. [14]) was based on graph-theoretic methods. In this section, we provide an alternative proof using symplectic techniques, centered on our notion of JW decomposition (Definition 4).

We begin with the observation that anticommuting sets together with isotropic subspaces can be used to build CNC sets. We begin with the observation that if γ is a value assignment on an isotropic subspace J and J is contained in a larger isotropic subspace J' , then the value assignment can be extended $\tilde{\gamma} : J' \rightarrow \mathbb{Z}_2$. Let I be an isotropic subspace of E_n and $\{a_1, \dots, a_N\}$ be an anticommuting set in $I^\perp$. Then,

$$\Omega = \bigcup_{k=1}^N I_k, \quad (20)$$

where $I_k = \langle a_k, I \rangle$ is a CNC set. To see this, first observe that the closedness property under commutation is clear. To construct a value assignment, start from a value assignment $\gamma_0 : I \rightarrow \mathbb{Z}_2$ and extend it to each I_k to obtain value assignments $\gamma_k : I_k \rightarrow \mathbb{Z}_2$. Then the family $\{\gamma_k\}_{k=1}^N$ of value assignments that match on I defines a value assignment $\gamma : \Omega \rightarrow \mathbb{Z}_2$.

The classification theorem says that the converse of this observation is also true. Let us start by considering a closed subset Ω . We write

$$I(\Omega) = \{a \in \Omega \mid [a, b] = 0 \forall b \in \Omega\}, \quad (21)$$

and call this isotropic subspace the center of Ω . Assume that $\dim(I(\Omega)) = n - m$. Then we can write $I(\Omega)^\perp = I(\Omega) \oplus W(\Omega)$ where $W(\Omega)$ is a $2m$ -dimensional symplectic subspace in a JW decomposition. When there is no confusion, we simply write I and W for the subspaces associated with the closed subset. Using the closedness of Ω , we can write it as a union of isotropic subspaces of the form $\langle a, I \rangle$ where $a \in \Omega \setminus I$. Note that the difference is empty when $\Omega = I$. Also, $\langle a, I \rangle = \langle \bar{a}, I \rangle$ for every $a \in \Omega \setminus I$ where $\bar{a}$ is the projection of a onto W . Therefore, we can write Ω as in Eq. (20), where $a_k \in W \setminus \{0\}$ for all k and $\langle a_k, I \rangle \neq \langle a_l, I \rangle$ for $k \neq l$. Proving that $\{a_1, \dots, a_N\}$ forms an anticommuting set requires

additional work, which we defer to Sec. A. The proofs of the following two results can be found there.

Theorem 1. Let $\Omega \subset E_n$ be a CNC subset. Then there exists an anticommuting set $\{a_1, \dots, a_N\} \subset W(\Omega)$ such that

$$\Omega = \bigcup_{k=1}^N \langle a_k, I(\Omega) \rangle, \quad (22)$$

where $1 \leq N \leq 2m+1$ and $\dim(I(\Omega)) = n-m$ for some $1 \leq m \leq n$.

Corollary 2. The maximal CNC sets are precisely those that have a decomposition as in Eq. (22) with $N = 2m+1$.

We call a maximal CNC set Ω of type (n, m) if $I(\Omega)$ has dimension $n-m$ and $N = 2m+1$ in Eq. (22). For each such Ω there exists a symplectic isomorphism $\alpha : E_n \rightarrow E_n$ such that

$$\begin{aligned} \alpha(I) &= \langle z_1, \dots, z_{n-m} \rangle, \\ \alpha(I') &= \langle x_1, \dots, x_{n-m} \rangle, \\ \alpha(\{a_i\}_{i=1}^{2m+1}) &= \{\hat{a}_i\}_{i=1}^{2m+1}, \end{aligned} \quad (23)$$

where $\{\hat{a}_i\}_{i=1}^{2m+1}$ is the JW set associated with the symplectic basis $\{x_{n-m+i}, z_{n-m+i}\}_{i=1}^m$. Hence $\text{Sp}_{2n}(\mathbb{Z}_2)$ acts transitively on the set of maximal CNC sets of type (n, m) .

The surjective group homomorphism $\varphi : \text{Cl}_n \rightarrow \text{Sp}_{2n}(\mathbb{Z}_2)$ implies there exists a Clifford unitary $U_\alpha \in \text{Cl}_n$ such that $\varphi(U_\alpha) = \alpha$. Moreover, the phases associated with the conjugation action of U_α on the Pauli operators $\{T_{v_i}\}_{i=1}^{2n}$ where $\{v_i\}_{i=1}^{2n}$ is a (not necessarily symplectic) basis of E_n can be chosen freely [36]. Therefore, for each maximal CNC set Ω of type (n, m) , there exists a Clifford unitary U_α such that

$$A_\Omega^\gamma = U_\alpha^\dagger (\Pi_{\langle z_1, \dots, z_{n-m} \rangle}^0 \otimes A_\Omega^0) U_\alpha, \quad (24)$$

where $\tilde{\Omega}$ is a maximal CNC of type (m, m) and 0 refers to the value assignment that assigns the value 0 to each element in the corresponding set. Conversely, a $(n-m)$ -qubit stabilizer state and an m -qubit CNC operator can be composed—i.e., tensored as in Eq. (24)—to produce an n -qubit CNC operator.

This lets us compute the number of phase point operators; see [[37], Theorem 5]. The number $N_{n,m}$ of maximal CNC sets of type (n, m) is given by

$$N_{n,m} = N_{n-m} N_{m,m},$$

where

(1) N_{n-m} is the number of isotropic subspaces of dimension $n-m$:

$$N_{n-m} = \prod_{k=1}^{n-m} \frac{4^{n-k+1} - 1}{2^k - 1}.$$

(2) $N_{m,m}$ is the number of maximal CNC sets of type (m, m) , i.e., the number of distinct sets of JW elements:

$$N_{m,m} = \frac{|\text{Sp}_{2m}(\mathbb{Z}_2)|}{(2m+1)!},$$

where $|\text{Sp}_{2m}(\mathbb{Z}_2)| = 2^{m^2} \prod_{j=1}^m (4^j - 1)$.

Additionally, the number $M_{n,m}$ of maximal CNC operators of type (n, m) is given by

$$M_{n,m} = N_{n,m} V_{n,m},$$

where $V_{n,m}$ is the number of value assignments on a CNC set of type (n, m) :

$$V_{n,m} = 2^{n+m+1}.$$

C. Enlarging to maximal CNC

Given a CNC set

$$\Omega = \bigcup_{k=1}^N I_k,$$

where $I_k = \langle a_k, I \rangle$, $\dim(I) = n-m$, and $N < 2m+1$, there are two ways to enlarge to a maximal CNC set:

(1) Assume that $N = 2l+1$ for some $1 \leq l < m$. For an isotropic subspace $J \subset I^\perp$ of dimension $t = m-l$ satisfying $I \cap J = 0$, we can construct a maximal CNC set $\tilde{\Omega}$ defined by

$$\tilde{\Omega} = \bigcup_{k=1}^{2l+1} \tilde{I}_k,$$

where $\tilde{I}_k = \langle a_k, \tilde{I} \rangle$ and $\tilde{I} = I \oplus J$. After choosing a basis $J = \{e_{n-m+1}, \dots, e_{n-m+t}\}$ we can also construct the value assignments $\tilde{\gamma} : \tilde{\Omega} \rightarrow \mathbb{Z}_2$. Such value assignments are of the following form: For $s \in \mathbb{Z}_2^t$, we define the value assignment

$$\gamma * s(a) = \begin{cases} \gamma(a), & a \in \Omega \\ s_i, & a = \hat{e}_{n-m+i}. \end{cases} \quad (25)$$

Note that, on other Pauli elements, the values are automatically determined.

(2) Next, we provide an enlargement where the center is fixed but the size of the anticommuting set is increased. Complete the set $\{a_1, \dots, a_l\}$ to a JW set. It is only possible when $\{a_1, \dots, a_l\}$ is not maximal, and we can do this by following Algorithms 1 and 2 in Ref. [33]. Let $t = 2(m-l)$. Then, for $s \in \mathbb{Z}_2^t$, we define

$$\gamma * s(a) = \begin{cases} \gamma(a), & a \in \Omega \\ s_i, & a = a_{2l+1+i}. \end{cases} \quad (26)$$

VII. THE UPDATE RULES

In this section, we describe the update rules for CNC operators, as defined in Eq. (5). Under a Pauli measurement T_b , a CNC operator A_Ω^γ is updated to a probabilistic mixture of CNC operators. For the update rules of A_Ω^γ , we need the following notation:

(1) Given $b \in \Omega$, we write $\gamma + [b, \cdot] : \Omega \rightarrow \mathbb{Z}_2$ for the value assignment defined by $a \mapsto \gamma(a) + [b, a]$.

(2) Given $b \in E_n \setminus \Omega$, we write $\Omega(b)$ for the closed set defined by

$$\Omega(b) = \langle b \rangle + \langle b \rangle^\perp \cap \Omega,$$

and for $r \in \mathbb{Z}_2$ we write $\gamma * r : \Omega(b) \rightarrow \mathbb{Z}_2$ for the value assignment defined by

$$\gamma * r(a) = \begin{cases} \gamma(a), & a \in \langle b \rangle^\perp \cap \Omega \\ \gamma(b+a) + r + \beta(b, a), & a \in \langle b \rangle + \langle b \rangle^\perp \cap \Omega. \end{cases}$$

Alternatively, $\gamma * r$ can be described as the unique value assignment determined by $\gamma * r(b) = r$ and $\gamma * r(a) = \gamma(a)$ for $a \in \langle b \rangle^\perp \cap \Omega$.

If the outcome $r \in \mathbb{Z}_2$ is observed, the update of a CNC operator under measurement of a Pauli operator T_b is given by the following formula [11]:

$$\Pi_b^r A_\Omega^\gamma \Pi_b^r = \begin{cases} \delta_{r, \gamma(b)} \frac{A_\Omega^\gamma + A_\Omega^{\gamma+[b, \cdot]}}{2}, & b \in \Omega \\ \frac{1}{2} A_\Omega^{\gamma * r}, & b \notin \Omega. \end{cases} \quad (27)$$

So, there are two cases for the update rule determined by whether $b \in \Omega$.

This formula generalizes the well-known update of a stabilizer projector under a Pauli measurement. Let I be an isotropic subspace of E_n and s be a value assignment on I . Then

$$\Pi_b^r \Pi_I^s \Pi_b^r = \begin{cases} \delta_{r, s(b)} \Pi_I^s, & b \in I \\ \frac{|(b)^\perp \cap I|}{|I|} \Pi_{I(b)}^{s * r}, & b \notin I. \end{cases}$$

Next, we refine the update rule given in Eq. (27) as a preparation for the tableau representation in Sec. VIII. We have the following containment of subspaces:

$$I \subset I_k \subset \Omega \subset I^\perp \subset E_n,$$

where $I_k = \langle a_k, I \rangle$. The refined update rules will break into four cases:

- (1) *Case I.* $b \in I$;
- (2) *Case II.* $b \in I_k \setminus I$;
- (3) *Case III.* $b \in I^\perp \setminus \Omega$;
- (4) *Case IV.* $b \notin I^\perp$.

For our simulation, we consider only maximal CNCs. So, assume that Ω is a maximal CNC set of type (n, m) . Then, Theorem 1 implies that $\{a_1, \dots, a_{2m+1}\}$ is a JW set in $W(\Omega) = \langle a_1, \dots, a_{2m} \rangle$. Using this, we can distinguish the first three cases by the number of JW elements that anticommute with b . Note that $|\mathcal{A}_b|$ is even by Corollary 1.

Lemma 6. For $b \in I^\perp$, let us define the integer

$$N_b = \frac{|\mathcal{A}_b|}{2}.$$

Then, we have three cases:

- (1) $b \in I$ if and only if $N_b = 0$.
- (2) $b \in I_k \setminus I$ if and only if $N_b = m$.
- (3) $b \in I^\perp \setminus \Omega$ if and only if $1 \leq N_b \leq m-1$.

Proof. Part (1): Note that $\langle \Omega \rangle = I^\perp$. Then we can see that $b \in I$ if and only if all JW elements commute with b , i.e., $|\mathcal{A}_b| = 0$. Part (2): It is clear that if $b \in I_k \setminus I$ then $N_b = m$. Now, Let $N_b = m$ and $\bar{b}$ denote the projection of b onto W . Then $\bar{b} = a_k$ for some k by Lemma 3 which implies $b \in I_k \setminus I$. Part (3) holds in the remaining case, i.e., b does not belong to Ω if and only if $|\mathcal{A}_b| \neq 0, 2m$. ■

Cases I and II fall under the case $b \in \Omega$ of Eq. (27), and the CNC set remains the same under update, although the value assignment may not. Cases III and IV require more work. Given a maximal CNC set Ω , we consider the associated JW decomposition (I, I', W) where

$$\begin{aligned} I &= \langle e_1, \dots, e_{n-m} \rangle, \\ I' &= \langle f_1, \dots, f_{n-m} \rangle, \\ W &= \langle a_1, \dots, a_{2m} \rangle. \end{aligned}$$

The bases $\{e_i\}_{i=1}^{n-m}$ and $\{f_i\}_{i=1}^{n-m}$ are chosen to be conjugate (Lemma 4).

(a) *Case III.* Let $b \in I^\perp \setminus \Omega$ and Ω be a maximal CNC set. We begin by reindexing the JW elements so that $\mathcal{A}_b = \{a_1, \dots, a_{2t}\}$, where $1 \leq t \leq m-1$. Then, observe that $I(b) = \langle e_1, \dots, e_{n-m}, b \rangle$. When we compute $\Omega(b)$, we immediately find that

$$\Omega(b) = \bigcup_{k=2t+1}^{2m+1} I_k(b), \quad (28)$$

where $I_k(b) = \langle a_k, I(b) \rangle$. A conjugate basis can be given so that $I(b)' = \{f_1, \dots, f_{n-m}, a\}$ for some $a \in \mathcal{A}_b$. The CNC set $\Omega(b)$ is maximal if and only if $t = 1$.

Example 1. The smallest number of qubits for which maximality fails in the update rules is $n = 3$. For $m = 3$, consider the maximal CNC determined by the $2m+1 = 7$ mutually anticommuting elements

$$\begin{aligned} a_1 &= x_1, & a_2 &= z_1, \\ a_3 &= y_1 + x_2, & a_4 &= y_1 + z_2, \\ a_5 &= y_1 + y_2 + x_3, & a_6 &= y_1 + y_2 + z_3, \\ a_7 &= y_1 + y_2 + y_3. \end{aligned}$$

For $b = x_1 + x_2 + x_3$, we obtain $t = |\mathcal{A}_b|/2 = 2$, and

$$\Omega(b) = \langle I, x_1 \rangle \cup \langle I, y_1 + z_2 \rangle \cup \langle I, y_1 + y_2 + x_3 \rangle,$$

where $I = \langle x_1 + x_2 + x_3 \rangle$. The resulting CNC set $\Omega(b)$ is not maximal.

If $t > 1$, as in the example above, then we construct a maximal CNC by enlarging its center (see Sec. VIC). The maximal CNC set

$$\tilde{\Omega}(b) = \bigcup_{k=2t+1}^{2m+1} \tilde{I}_k(b) \quad (29)$$

containing $\Omega(b)$ and the corresponding conjugate bases are described as follows:

- (1) $\tilde{I}(b)' = \langle \tilde{f}_1, \dots, \tilde{f}_{n-m+t} \rangle$ where

$$\tilde{f}_i = \begin{cases} \hat{f}_{i-(n-m)}, & n-m+1 \leq i \leq n-m+t \\ \tilde{f}_i, & \text{otherwise,} \end{cases} \quad (30)$$

and

$$\hat{f}_i = \begin{cases} a_2, & i = 1 \\ a_{2i} + \sum_{k=1}^{i-1} (\hat{e}_k + \hat{f}_k), & 2 \leq i \leq t, \end{cases} \quad (31)$$

- (2) $\tilde{I}(b) = \langle \bar{e}_1, \dots, \bar{e}_{n-m+t} \rangle$ where

$$\bar{e}_i = \begin{cases} \hat{e}_{i-(n-m)}, & n-m+1 \leq i \leq n-m+t \\ e_i, & \text{otherwise,} \end{cases} \quad (32)$$

and

$$\hat{e}_i = \begin{cases} \bar{b}, & i = 1 \\ a_{2i-1} + \sum_{k=1}^{i-1} (\hat{e}_k + \hat{f}_k), & 2 \leq i \leq t, \end{cases} \quad (33)$$

where $\bar{b}$ is the projection of b onto W found using $\bar{b} = \sum_{a \in \mathcal{A}_b} a = \sum_{a' \in \mathcal{C}_b} a'$ from Corollary 1 and the fact that $[b, a_i] = 0$ if and only if $[\bar{b}, a_i]$ for all i .

- (3) $\tilde{I}_k(b) = \langle \bar{a}_k, \tilde{I}(b) \rangle$ where

$$\bar{a}_k = \bar{b} + a_k, \quad (34)$$

and $k = 2t+1, \dots, 2m+1$.

This construction comes directly from Lemma 5 and the corresponding equations Eqs. (18) and (19).

Given the description of $\tilde{\Omega}(b)$ in Eq. (29) and its subset $\Omega(b)$ in Eq. (28), a value assignment γ' on $\Omega(b)$ can be extended to $\tilde{\Omega}(b)$ by choosing values $s_i \in \mathbb{Z}_2$ for the basis vectors $\hat{e}_i$ where $2 \leq i \leq t$. This gives the value assignment $\gamma' * s$ described by the construction in Eq. (25). Then, we can write the Case III update rule as follows:

$$\Pi_b^r A_\Omega^\gamma \Pi_b^r = \frac{1}{2} \frac{\sum_{s \in \mathbb{Z}_2^{t-1}} A_{\tilde{\Omega}(b)}^{(\gamma' * r) * s}}{2^{t-1}}. \quad (35)$$

(b) *Case IV.* Now, we turn to the final update rule. Let $b \notin I^\perp$ and Ω be a maximal CNC set. The maximal CNC set

$$\Omega(b) = \bigcup_{k=1}^{2m+1} I_k(b),$$

with the corresponding conjugate bases are described as follows: Let λ be the minimum of the set $\{1 \leq i \leq n-m \mid [e_i, b] = 1\}$ of indices.

(1) $I(b)' = \langle \bar{f}_1, \dots, \bar{f}_{n-m} \rangle$ where

$$\bar{f}_i = \begin{cases} e_\lambda, & i = \lambda \\ f_i + e_\lambda, & \text{if } [f_i, b] = 1 \\ f_i, & \text{otherwise.} \end{cases} \quad (36)$$

(2) $I(b) = \langle \bar{e}_1, \dots, \bar{e}_{n-m} \rangle$ where

$$\bar{e}_i = \begin{cases} b, & i = \lambda \\ e_i + e_\lambda, & \text{if } [e_i, b] = 1 \text{ and } i \neq \lambda \\ e_i, & \text{otherwise.} \end{cases} \quad (37)$$

(3) $I_k(b) = \langle \bar{a}_k, I(b) \rangle$ where

$$\bar{a}_k = \begin{cases} a_k + e_\lambda, & \text{if } [a_k, b] = 1 \\ a_k, & \text{if } [a_k, b] = 0, \end{cases} \quad (38)$$

and $k = 1, \dots, 2m+1$.

This construction relies on the basis transformation described in Eq. (17).

We conclude this section with a remark on Case III. If the CNC set after the measurement is not maximal, we apply the first enlargement procedure described in Sec. VIC. As we demonstrate next, this is the only feasible method for achieving a maximal CNC, since the alternative approach—enlarging the JW elements—is not possible. Let $\Omega \subset E_n$ be a maximal CNC set with $\dim(I) = n-m$ and JW elements $\{a_1, \dots, a_{2m+1}\}$. Then, after measuring $b \in I^\perp \setminus \Omega$ (reindex JW elements such that $\mathcal{A}_b = \{a_1, \dots, a_{2t}\}$), we have the CNC set

$$\Omega(b) = \bigcup_{a \in \mathcal{C}_b} \langle I(b), a \rangle = \bigcup_{k=2t+1}^{2m+1} \langle I(b), a_k \rangle,$$

where $I(b) = \langle I, b \rangle$ is the center of $\Omega(b)$. Let $\tilde{W}$ be a symplectic subspace such that $I(b)^\perp = I(b) \oplus \tilde{W}$. Then, we claim that any choice of an anticommuting set $\{\hat{a}_1, \dots, \hat{a}_{2(m-t)+1}\} \subset \tilde{W} \cap \Omega$ such that

$$\Omega(b) = \bigcup_{k=1}^{2(m-t)+1} \langle I(b), \hat{a}_k \rangle$$

is maximal. To see this, first note that

$$\sum_{a \in \mathcal{C}_b} a = \sum_{k=2t+1}^{2m+1} a_k = b \in I(b).$$

Additionally, without loss of generality we have that $\hat{a}_k = c_k + a_{2t+k}$ for some $c_k \in I(b)$ where $k = 1, \dots, 2(m-t)+1$. This gives

$$\sum_{k=1}^{2(m-t)+1} \hat{a}_k \in I(b),$$

and $\tilde{W} \cap I(b) = 0$ implies

$$\sum_{k=1}^{2(m-t)+1} \hat{a}_k = 0.$$

Therefore, by Proposition 2 $\{\hat{a}_1, \dots, \hat{a}_{2(m-t)+1}\}$ is a maximal anticommuting set in $\tilde{W}$.

VIII. THE TABLEAU REPRESENTATION

In this section, we introduce our phase-space tableau, also referred to as the CNC tableau, and describe its update rules. These rules are implemented in Ref. [21]; see Fig. 1.

A. The phase-space tableau

As shown in Theorem 1, a maximal CNC set Ω has the form

$$\Omega = \bigcup_{k=1}^{2m+1} I_k,$$

where $I = I(\Omega)$ is the center, $I_k = \langle a_k, I \rangle$, and $\{a_1, \dots, a_{2m+1}\}$ is a JW set in $W = W(\Omega)$. Here we chose a JW decomposition $E_n = I \oplus W \oplus I'$ associated with the CNC set Ω where I' is a conjugate isotropic. Our goal is to construct a tableau representation for Ω and a value assignment γ on this CNC set given by a matrix whose rows encode the conjugate basis vectors and the JW elements.

For an element $a \in E_n$ expressed as $a = \sum_{i=1}^n (\alpha_i x_i + \beta_i z_i)$ we write

$$R(a) = (\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n)$$

for the row vector consisting of the coefficients. For a matrix M we write $M_{i,*}$ for its i th row vector.

Definition 5. Given a JW decomposition $E_n = I \oplus W \oplus I'$ of type m together with a basis $\{e_1, \dots, e_{n-m}\}$ for I , a conjugate basis $\{f_1, \dots, f_{n-m}\}$ for I' , and an associated JW set $\{a_1, \dots, a_{2m+1}\}$ for W , we define a $(2n) \times (2n+1)$ matrix

$$\mathcal{T}_{i,*} = \begin{cases} R(f_i), & 1 \leq i \leq n-m \\ R(e_{i-(n-m)}), & n-m+1 \leq i \leq 2(n-m) \\ R(a_{i-2(n-m)}), & 2(n-m)+1 \leq i \leq 2n+1 \end{cases}$$

called a tableau representation of type m for the JW decomposition of type m . We denote the i th row of the tableau by $R_i = \mathcal{T}_{i,*}$.

We separate the matrix $\mathcal{T} = \mathcal{T}_\Omega$ into three parts:

(1) The destabilizer part $\mathcal{T}|_D$ defined by

$$(\mathcal{T}|_D)_{i,*} = R_i,$$

where $1 \leq i \leq n-m$.

(2) The stabilizer part $\mathcal{T}|_S$ defined by

$$(\mathcal{T}|_S)_{i,*} = R_{n-m+i},$$

where $1 \leq i \leq n-m$.

(3) The Jordan-Wigner part $\mathcal{T}|_{JW}$ defined by

$$(\mathcal{T}|_{JW})_{i,*} = R_{2(n-m)+i},$$

where $1 \leq i \leq 2m+1$.

Next, we extend the tableau to include a value assignment $\gamma : \Omega \rightarrow \mathbb{Z}_2$. The values of γ on the basis of I and the JW elements are recorded in the last column. In the destabilizer part, we extend the value assignment with zero values for convenience, even though these values are irrelevant to the simulation algorithm.

Definition 6. A tableau representation for a maximal CNC pair (Ω, γ) is the matrix $\mathcal{T}_\Omega^\gamma$ obtained from a tableau representation of the associated JW decomposition by appending a column consisting of the value assignments to the basis elements; that is, $(\mathcal{T}_\Omega^\gamma)_{i,j} = \mathcal{T}_{i,j}$ for $1 \leq j \leq 2n$, $1 \leq i \leq 2n+1$, and

$$(\mathcal{T}_\Omega^\gamma)_{i,2n+1} = \begin{cases} 0, & 1 \leq i \leq n-m \\ \gamma(e_{i-(n-m)}), & n-m+1 \leq i \leq 2(n-m) \\ \gamma(a_{i-2(n-m)}), & 2(n-m)+1 \leq i \leq 2n+1. \end{cases}$$

If the maximal CNC has type (n, m) , we say that the tableau $\mathcal{T}_\Omega^\gamma$ is of type m .

A simple algorithm to construct the tableau for a given maximal CNC operator A_Ω^γ is provided in Algorithm 3. For notational simplicity, we write $\tilde{\mathcal{T}} = \mathcal{T}_\Omega^\gamma$. The rows of this matrix are denoted by $\tilde{R}_i$ and they have the form

$$\tilde{R}_i = (R_i, s_i),$$

where R_i is the i th row of $\mathcal{T}$ and s_i is the value assigned by γ to the corresponding Pauli element. Similarly, we partition $\tilde{\mathcal{T}}$ into destabilizer, stabilizer, and JW parts:

$$\tilde{\mathcal{T}}|_D, \quad \tilde{\mathcal{T}}|_S, \quad \tilde{\mathcal{T}}|_{JW}.$$

Example 2. Let $I = \langle z_1, \dots, z_{n-m} \rangle$, $I' = \langle x_1, \dots, x_{n-m} \rangle$, and $\{a_1, \dots, a_{2m+1}\}$ denote the corresponding JW set. We wish to depict the corresponding tableau. For convenience, let us group together even and odd elements a_{2k} and a_{2k-1} , respectively. We have

$$\left[\begin{array}{c|cc} I_{n-m} & & \\ & A & B \\ & B & A \\ & 1_m & 1_m \end{array} \right].$$

Here 1_m is a row vector of m ones, I_{n-m} is an $(n-m) \times (n-m)$ identity matrix, and empty spaces represent matrices of all zeros of the appropriate dimensions. A and B are $m \times m$ lower triangular and strictly lower triangular matrices, respectively, satisfying

$$A_{ij} = \begin{cases} 1, & i \leq j \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad B_{ij} = \begin{cases} 1, & i < j \\ 0, & \text{otherwise} \end{cases}.$$

Example 3. Consider CNC tableaux $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}'$ of type (n, m) and $(n', 0)$, respectively. These tableaux are of the form

$$\left[\begin{array}{cc} D & \\ S & \gamma_S \\ W & \gamma_W \end{array} \right] \quad \text{and} \quad \left[\begin{array}{cc} D' & \\ S' & \gamma_{S'} \end{array} \right],$$

where the γ indicate phase bits. Following Eq. (24), their composition, i.e., the tensor product, is given by the tableau

$$\left[\begin{array}{ccc} D & & \\ & D' & \\ S & & \gamma_S \\ & S' & \gamma_{S'} \\ W & & \gamma_W \end{array} \right].$$

B. Updating the phase-space tableau

Just as in the stabilizer tableau approach, a CNC tableau can be updated either through the deterministic action of Clifford unitaries or the probabilistic action of Pauli measurements. The rules for updating under Clifford action are exactly the same as in the stabilizer approach. The n -qubit Clifford group Cl_n is generated by the Hadamard and phase gates H_i and S_j that act nontrivially only on a single qubit, where

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},$$

together with the two-qubit cnot entangling gate $CX_{ct} = |0\rangle\langle 0|_c \otimes I_t + |1\rangle\langle 1|_c \otimes X_t$, where c and t are the control and target qubits. Recall that there is a surjective group homomorphism $\varphi : \text{Cl}_n \rightarrow \text{Sp}_{2n}(\mathbb{Z}_2)$. By surjectivity, the images of the Clifford generators by $h_i := \varphi(H_i)$, $s_j := \varphi(S_j)$, and $cx_{ct} := \varphi(CX_{ct})$ themselves generate the symplectic group $\text{Sp}_{2n}(\mathbb{Z}_2)$.

The rules for updating under the action of h_i , s_j , or cx_{ct} (together with keeping track of the phase bit) are the same as those given in Ref. [3]. Supposing a circuit consists of L Clifford gates, these updates can be performed in $O(nL)$ time.

Next, we turn to the update rules for the Pauli measurements. Given $a \in E_n$ and $s \in \mathbb{Z}_2$, consider the row vector

$$\tilde{R}(a) = (R(a), s),$$

obtained by appending to $R(a)$ the value s as the last column entry. Given another element $b \in E_n$ and value $r \in \mathbb{Z}_2$, we can compute the commutator and the β function [see Eq. (2)] as follows:

$$[a, b] = R(a)\omega R(b)^T, \quad \text{where } \omega = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix},$$

and

$$\begin{aligned} \beta(a, b) = & \frac{1}{2} \left[\sum_{i=1}^n [R(a)_i R(a)_{i+n} + R(b)_i R(b)_{i+n}] \right. \\ & + 2 \sum_{i=1}^n [R(a)_{i+n} R(b)_i] \\ & \left. - \sum_{i=1}^n [R(a) \oplus R(b)]_i [R(a) \oplus R(b)]_{i+n} \right]. \quad (39) \end{aligned}$$

We define the row product operation between two rows:

$$\tilde{R}(a) * \tilde{R}(b) = [R(a) + R(b), s + r + \beta(a, b)]. \quad (40)$$

Notice that the computation of $\beta(a, b)$ requires $O(n)$ steps so that the row product itself also has $O(n)$ complexity in total.

Our first task is to identify which of the measurement cases from Sec. VII apply.

Lemma 6. For a maximal CNC set Ω and $b \in E_n$, checking which of the cases (i) $b \in I$, (ii) $b \in I_k/I$, (iii), $b \in I^\perp/\Omega$, or (iv) $b \notin I^\perp$ occurs can be performed in $O(n^2)$ time.

Proof. To determine which update rule to apply for $b \in E_n$ we must compute the symplectic form of $R(b)$ with each row of $\mathcal{T}_\Omega$. We mainly use Lemma 6 to distinguish the cases. It suffices to perform the matrix multiplication $\mathcal{T}_\Omega(b) = \mathcal{T}_\Omega \omega R(b)^T$, which can be done in $O(n^2)$ time. Once we have computed the symplectic forms, we observe that a Pauli element $b \in \Omega$ must satisfy either (i) $b \in I$ or (ii) $b \in I_k \setminus I$ for some unique $k = 1, \dots, 2m+1$. The former occurs if and only if b commutes with all rows $n-m+1$ to $2n+1$, whereas the latter occurs if and only if b commutes with rows $n-m+1$ to $2(n-m)$ plus one additional row in $2(n-m)+1$ to $2n+1$. On the other hand, if $b \notin \Omega$ then we have either (iii) $b \in I^\perp \setminus \Omega$ or (iv) $b \notin I^\perp$. The latter happens if and only if b anticommutes with any element in rows $n-m+1$ to $2(n-m)$, whereas the former occurs if b commutes with all rows $n-m$ to $2(n-m)$ but anticommutes with $2t$ rows in $2(n-m)+1$ to $2n+1$. ■

We are now ready to describe the update rules for the tableau representation using the results of Sec. VII. Consider a maximal CNC set Ω and suppose we perform a measurement of a Pauli operator [38] T_b with outcome r_b . Let $\bar{b}$ be the projection of b onto W . Then we define $\tilde{R}(b) = (R(b), r_b)$ and $\tilde{R}(\bar{b}) = (R(\bar{b}), r_b)$.

Case I. $b \in I$: Deterministic outcome $r_b = \gamma(b)$ and deterministic tableau update $\mathcal{T}_\Omega^\gamma \mapsto \mathcal{T}_\Omega^\gamma$, i.e., tableau remains untouched.

Case II. $b \in I_k \setminus I$: Deterministic outcome $r_b = \gamma(b)$ and probabilistic tableau update $\mathcal{T}_\Omega^\gamma \mapsto \mathcal{T}_\Omega^\gamma$ or $\mathcal{T}_\Omega^\gamma \mapsto \mathcal{T}_\Omega^{\gamma+[b, \cdot]}$ with equal probability.

Case III. $b \in I^\perp \setminus \Omega$: Sample $r_b \in \mathbb{Z}_2$ uniformly as the outcome. Sample $s = (s_2, \dots, s_t) \in \mathbb{Z}_2^{t-1}$ uniformly at random to update the tableau $\mathcal{T}_\Omega^\gamma \mapsto \mathcal{T}_{\Omega(b)}^{(\gamma * r_b) * s}$ as follows: First, order the rows of $\mathcal{T}_\Omega^\gamma|_{JW}$ in such a way that the new rows $\tilde{R}'_1, \dots, \tilde{R}'_{2m+1}$ satisfy

$$R'_i \omega R(b)^T = \begin{cases} 1, & 1 \leq i \leq 2t \\ 0, & \text{otherwise.} \end{cases}$$

We make the assignment

$$R'_1 \leftarrow \sum_{i=2t+1}^{2m+1} R'_i,$$

which computes $R(\bar{b})$ using Corollary 1 that we use in place of $R(b)$. Then, the new tableau has type $m-t$ and is given by using Lemma 5 as

$$\begin{aligned} (\mathcal{T}_{\Omega(b)}^{(\gamma * r_b) * s}|_D)_{i,*} &= \begin{cases} (R'_2, 0), & i = n-m+1 \\ (R'_{2k} + \sum_{j=1}^{2k-2} R'_j, 0), & i = n-m+k, k = 2, \dots, t \\ (\mathcal{T}_\Omega^\gamma|_D)_{i,*}, & \text{otherwise,} \end{cases} \\ (\mathcal{T}_{\Omega(b)}^{(\gamma * r_b) * s}|_S)_{i,*} &= \begin{cases} (R'_1, r_b), & i = n-m+1 \\ (R'_{2k-1} + \sum_{j=1}^{2k-2} R'_j, s_k), & i = n-m+k, k = 2, \dots, t \\ (\mathcal{T}_\Omega^\gamma|_S)_{i,*}, & \text{otherwise,} \end{cases} \\ (\mathcal{T}_{\Omega(b)}^{(\gamma * r_b) * s}|_{JW})_{i,*} &= \tilde{R}'_{2t+i} * \tilde{R}(\bar{b}) \text{ where } 1 \leq i \leq 2(m-t)+1. \end{aligned}$$

Case IV. $b \notin I^\perp$: Sample $r_b \in \mathbb{Z}_2$ uniformly as the outcome and update the tableau $\mathcal{T}_\Omega^\gamma \mapsto \mathcal{T}_{\Omega(b)}^{\gamma * r_b}$ as follows: Let λ be the minimum of the indices satisfying

$$(\mathcal{T}_\Omega^\gamma|_S)_{i,*} \omega R(b)^T = 1.$$

Then,

$$(\mathcal{T}_{\Omega(b)}^{\gamma * r_b})_{i,*} = \begin{cases} \tilde{R}_{\lambda+n-m}, & i = \lambda \\ \tilde{R}(b), & i = \lambda + n - m \\ \tilde{R}_i * \tilde{R}_\lambda, & R_i \omega R(b)^T = 1, \\ & i \neq \lambda, \lambda + n - m \\ \tilde{R}_i, & \text{otherwise.} \end{cases}$$

Let us discuss how these updates are implemented in practice.

Case I. The outcome r_b is deterministic and can be computed directly from $\mathcal{T}_\Omega(b)$ using the destabilizer entries in the

first $n-m$ rows. Similar to the stabilizer tableau [3], we have that

$$R(b) = \sum_{i=1}^{n-m} [R(b), R_i] R_{i+n-m}.$$

Introduce an additional row of scratch space and compute $\tilde{R}(b) = \tilde{R}_{i_1+n-m} * \tilde{R}_{i_2+n-m} * \dots * \tilde{R}_{i_K+n-m}$, where $i_k \in \{1, \dots, n-m\}$ is such that $[R(b), R_{i_k}]$ is nonzero. The deterministic outcome $r_b = \gamma(b)$ is the phase bit of $\tilde{R}(b)$.

Case II. This follows similarly to Case I with the only difference that

$$R(b) = R_j + \sum_{i=1}^{n-m} [R(b), R_i] R_{i+n-m}$$

for some unique $j \in \{2(n-m)+1, 2n+1\}$ such that $[R(b), R_j] = 0$. From this $\tilde{R}(b)$ and $r_b = \gamma(b)$ can be computed.

Case III. This case is somewhat more involved. Detailed pseudocode can be found in Algorithm 1. Recall first that $R(b)$ anticommutes with $2t$ ($1 \leq t \leq m-1$) rows in $\mathcal{T}|_{JW}$. Letting $K = 2(n-m)$, these $2t$ rows are moved to rows $K+1, \dots, K+2t$. Next, we make the assignment

$$R_{K+1} \leftarrow \sum_{i=K+2t+1}^{2n+1} R_i,$$

which computes $R(\bar{b})$ using Corollary 1 and assigns it to the $(K+1)$ st row. Randomly choose $r_b \in \mathbb{Z}_2$ as the outcome. The new stabilizer is $\tilde{R}_{K+1} = (R_{K+1}, r_b)$ and the new destabilizer is $\tilde{R}_{K+2} = (R_{K+2}, 0)$. For $i = 2, \dots, 2t$ we do as follows. Introduce a row of scratch space $R(S)$ initialized at $R(S) = R_{K+1} + R_{K+2}$ and randomly choose $s_i \in \mathbb{Z}_2$. We make the reassignments:

$$\begin{aligned} R_{K+2i-1} &\leftarrow R_{K+2i-1} + R(S), \\ \tilde{R}_{K+2i-1} &= (R_{K+2i-1}, s_i), \\ R_{K+2i} &\leftarrow R_{K+2i} + R(S), \\ \tilde{R}_{K+2i} &= (R_{K+2i}, 0), \\ R(S) &\leftarrow R(S) + R_{K+2i-1} + R_{K+2i}, \end{aligned}$$

which are now the i th new stabilizer and destabilizer, respectively. For all $i > K+2t$ we update the rows as

$$\tilde{R}_i \leftarrow \tilde{R}_i * \tilde{R}_{K+1}.$$

We complete the update by making in-place modifications to the tableau, such that, for $i = 1, \dots, t$, the rows $\tilde{R}_{K+2i}$ are in positions $n-m+1$ to $n-m+t$ and rows $\tilde{R}_{K+2i-1}$ are in positions $K+t+1$ to $K+2t$.

Case IV. Here $b \notin I^\perp$ and thus $R(b)$ anticommutes with one or more stabilizer rows. The update proceeds in a fashion similar to stabilizer theory. We identify a stabilizer row with least index λ such that $[R(b), R_\lambda] = 1$, then for all $i \notin \{\lambda, \lambda-n+m\}$ such that $[R(b), R_i] = 1$ we make the reassignment

$$\tilde{R}_i \leftarrow \tilde{R}_i * \tilde{R}_\lambda,$$

which is similar to Eq. (17), and this guarantees that $R(b)$ anticommutes only with R_λ (and possibly $R_{\lambda-n+m}$). Make a random choice $r_b \in \mathbb{Z}_2$. We now make the updates

$$R_{\lambda-n+m} \leftarrow R_\lambda \text{ and } \tilde{R}_\lambda \leftarrow (R(b), r_b)$$

to the destabilizer and stabilizer, respectively. Pseudocode can be found in Algorithm 2.

Theorem 2. The following statements are valid for the CNC tableau:

- (1) A CNC tableau can be encoded using $O(n^2)$ bits of memory.
- (2) The update of a CNC tableau under:
 - (a) Clifford unitaries can be performed in $O(n)$ time,
 - (b) Pauli measurements can be performed in $O(n^2)$ time.

Proof. We adopt a notion of complexity in which a basic computational step consists of bitwise multiplication or addition. The proof of the first statement follows from the fact

that our CNC tableau is a $(2n+1) \times (2n+1)$ matrix over $\mathbb{Z}_2$ requiring $4n^2 + 4n + 1$ bits, which has a linear memory overhead increase over the stabilizer tableau. The update under Clifford unitaries follows in analogy to the stabilizer case, with an $O(1)$ increase due to the additional row that we keep in memory. Let us turn to the analysis of the measurements. First, by Lemma 7, it takes $O(n^2)$ rather than $O(n)$ time to check which of the measurement updates to perform. For Cases I and II, we simply require the procedure for outputting the deterministic outcome, which requires $O(n)$ row product operations and thus $O(n^2)$ operations in total. Generating the new stabilizer and destabilizer in Case III rows requires summing $O(t)$ rows, thus $O(nt)$ basic operations and $O(n^2)$ operations in the worst case. Updating the commuting JW rows in Case III requires performing $O(m-t)$ row product operations, which has $O(n^2)$ complexity in the worst case. Finally, Case IV requires us to, in the worst case, update $n+m$ rows that anticommute with $R(b)$, which requires $O(n(n+m))$ operations. In all cases, scratch space used is $O(n)$. Hence the overall memory complexity is $O(n^2)$. ■

IX. PHASE-SPACE TABLEAU SIMULATION

For classical simulation of QCM we consider an initial n -qubit quantum state expanded using CNC operators as a generally quasiprobabilistic mixture

$$\rho = \sum_{\alpha} W_{\rho}(\alpha) A_{\alpha},$$

where α runs over our phase space points which consists of all maximal (n, m) CNC pairs. This initial state may be completely general; for example, it could be a pure magic state or its approximation if one is willing to tolerate some error in the computation.

Adaptive stabilizer operations necessary for universal quantum computation can be described using instruments; see, i.e., Ref. [39]. For Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, an instrument Φ with outcome set Σ consists of completely positive linear operators $\Phi^s : L(\mathcal{H}) \rightarrow L(\mathcal{K})$ for all $s \in \Sigma$ such that $\Phi = \sum_{s \in \Sigma} \Phi^s$ is a quantum channel. Sequences of instruments can be composed and satisfy the channel condition

$$\Psi = \sum_{s \in \Sigma_1 \times \dots \times \Sigma_T} (\Phi_T \circ \dots \circ \Phi_1)^s = \Psi_N \circ \dots \circ \Psi_1.$$

We consider instruments consisting of Pauli measurements and Clifford gates, optionally with classical feedforward and adaptivity, that can be given a Kraus decomposition

$$\Phi^{s_i}(A) = V_{s_i} \Pi_b^{s_i} A \Pi_b^{s_i} V_{s_i}^\dagger,$$

where $A \in L(\mathcal{H})$ is an arbitrary linear operator and V_{s_i} is a Clifford unitary for $s_i \in \Sigma_i$. Observe that nonadaptive Clifford unitaries can always be absorbed into the definition of V_{s_i} . These instruments are referred to as completely stabilizer preserving instruments and the corresponding channels are a subset of simulable completely stabilizer-preserving channels [40]. An important class of examples includes gadgets that deterministically implement non-Clifford gates by performing adaptive stabilizer operations acting on a magic state input [41].

A. Classical simulation of QCM

There are two distinct sampling-based classical simulation tasks one can perform, depending on whether W_ρ represents a proper probability distribution or not. In the former case, we perform a weak simulation [42,43], in which the output of the simulation is samples from the true quantum-mechanical probability distribution obtained via the Born rule. When negativity is present, however, one cannot directly sample from W_ρ , and a weak simulation is not possible. One instead aims in this case to estimate Born rule probabilities using the methods of Ref. [17]; see also Ref. [18].

1. Weak simulation

Let us consider a sequence $\Phi_1, \dots, \Phi_T$ of n -qubit stabilizer instruments. Quantum states ρ for which W_ρ is a probability distribution lie in CP_n , the CNC polytope given by the convex hull of maximal n -qubit CNC operators. Weak simulation proceeds by first sampling an initial phase space element α_0 from the probability distribution W_ρ . Each stabilizer instrument Φ_i induces a stochastic map $\alpha_{i-1} \mapsto (s_i, \alpha_i)$, where the pair (s_i, α_i) is sampled from a conditional probability distribution $Q_{\alpha_{i-1}, i}$. This distribution is determined by

$$\Phi_i(A_{\alpha_{i-1}}) = V_{s_i} \Pi_{A_i} A_{\alpha_{i-1}} \Pi_{A_i} V_{s_i},$$

and Eqs. (27) and (35), up to a known Clifford unitary. Here $\alpha_{i-1} = (\Omega_{i-1}, \gamma_{i-1})$ denotes the phase-space element sampled after the application of Φ_{i-1} .

In practice, these stochastic maps are implemented using the phase-space tableau so that a stabilizer instrument Φ_i induces an efficient tableau update $\tilde{\mathcal{T}}_{\alpha_{i-1}} \mapsto \tilde{\mathcal{T}}_{\alpha_i}$. Note that the outcome can be retrieved from the tableau. Our procedure for weak simulation using the phase-space tableau is given in Algorithm 4. Correctness of our algorithm follows from the correctness of our update rules and Theorem 3 of Ref. [11]. If W_ρ can be efficiently sampled from, then the efficiency of our tableau update rules (Theorem 2) makes explicit the efficiency of Algorithm 4.

2. Born rule estimation

Suppose that ρ is an n -qubit quantum state that does not lie in CP_n and that Ψ is a stabilizer channel with decomposition into stabilizer instruments $\Phi_1, \dots, \Phi_T$ so that $\Psi = \sum_s (\Phi_T \circ \dots \circ \Phi_1)^s$, as above. In many cases of interest, the initial state is a magic state and Ψ implements a unitary coming from a universal gate set on a subset of the qubits. We wish to estimate the Born rule probability

$$p(x) := \text{Tr}[\Pi^x \Psi(\rho)] = \sum_{\alpha \in V} W_\rho(\alpha) \text{Tr}[\Pi^x \Psi(A_\alpha)], \quad (41)$$

where $\Pi^x = |x\rangle\langle x| \otimes I_{n-k}$ is a computational basis measurement (stabilizer projector) on the first $k \leq n$ qubits for a fixed outcome $x \in \mathbb{Z}_2^k$. This can be rewritten as an expectation value

$$p(x) = \sum_{\alpha \in V} \tilde{W}_\rho(\alpha) \cdot E(x|\alpha),$$

with

$$E(x|\alpha) = \text{sign}(\alpha) \|W_\rho\|_1 p(x|\alpha),$$

and where $\sigma(\alpha)$ is the sign of the weight $W_\rho(\alpha)$, $\|W_\rho\|_1$ is the negativity of the quasimixture, and $p(x|\alpha) = \text{Tr}[\Pi^x \Psi(A_\alpha)]$ is the probability of obtaining x given an initial phase-space element α . The average is taken with respect to the normalized probability distribution $\tilde{W}_\rho(\alpha) = |W_\rho(\alpha)|/\|W_\rho\|_1$.

We pursue a sampling-based approach to estimating $p(x)$ using the methods of Refs. [17,18]. Fix a number M of samples and consider the estimator

$$\hat{p}(x) = \frac{1}{M} \sum_i \hat{E}_i,$$

where $\hat{E}_i = E(x|\alpha_i)$ and α_i is the i th initial sample. In many practical cases, it is efficient to estimate $\hat{E}_i$ since we can efficiently sample from the probability $p(x|\alpha_i)$ using the phase-space tableau, as in weak simulation, and then use a $\text{poly}(n)$ subroutine to produce an estimate up to the desired precision [44].

Central to the estimation procedure (Algorithm 5) is that $\hat{p}(x)$ is an unbiased estimator of the Born rule probability $p(x)$ whose value is bounded by $\pm \|W_\rho\|_1$. Hoeffding's inequality [45] then implies that to estimate $p(x)$ within an additive error ϵ with probability $1 - \delta$ requires a number of samples given by

$$M \geq \frac{2}{\epsilon^2} \|W_\rho\|_1^2 \log(2/\delta).$$

Each run of the circuit can be performed in $\text{poly}(n)$ time, so the dominant contribution to the complexity of estimation is the number of samples M , which scales with $\|W_\rho\|_1^2$. When the decomposition is optimal, the sample complexity scales with $\Re(\rho)^2$.

B. Phase-space simulation of quantum algorithms

As a proof of concept, we tested our phase-space tableau approach [21] on two families of deterministic quantum algorithms—one solving the hidden shift problem [46], while the other solves the well-known Deutsch-Jozsa problem [47]. In particular, we used quasiprobabilistic methods to estimate the Born rule probability of a fixed deterministic outcome, as described above.

1. Hidden shift algorithm

In recent years, the hidden shift quantum algorithm [46] has become widely used to benchmark methods for classically simulating Clifford+ T circuits [5–7,48,49]. This algorithm was originally introduced by Roetteler in Ref. [46] to solve the hidden shift problem and to demonstrate an oracle separation between the complexity classes BQP and P. A circuit implementation of the hidden shift oracle was given by Bravyi and Gosset in Ref. [6] using the $\{Z, CZ, CCZ\}$ diagonal gate set. An appealing feature of the hidden shift algorithm is that one can control both the number of qubits n and the number of non-Clifford gates m so that there are instances of the problem that are solved by Clifford-dominated circuits with large n and relatively small m , a task which is impractical using state vector simulation.

For our family of oracles, we first fix positive integers κ and $v \geq 3\kappa$, with $n = 2v$ and $c = 2\kappa$. The parameter κ controls the total number of CCZ gates. Consider also a hidden shift

string $x = (x_1, \dots, x_n) \in \mathbb{Z}_2^n$ as well as a pair of Maierana-McFarland bent functions $f, f' : \mathbb{Z}_2^v \rightarrow \{\pm 1\}$ such that f' is the Walsh-Hadamard transformation of the shifted version of f . We then have that

$$|x\rangle = U|0\rangle, \text{ where } U = H^{\otimes n} O_f H^{\otimes n} O_{f'} H^{\otimes n}.$$

The oracles are n -qubit diagonal unitary operators that can be constructed explicitly according to

$$O_f = \left(\prod_{i=1}^v CZ_{i,i+v} \right) (O_g \otimes \mathbb{1}_v),$$

and

$$O_{f'} = \left(\prod_{i=1}^v CZ_{i,i+v} \right) (\mathbb{1}_v \otimes O_g) Z(x).$$

Here $Z(x) = Z^{x_1} \otimes \dots \otimes Z^{x_n}$ is an n -qubit unitary and O_g is a v -qubit diagonal unitary corresponding to an arbitrary Boolean function $g : \mathbb{Z}_2^v \rightarrow \mathbb{Z}_2$. For our purposes, we choose O_g to be of the form

$$O_g = \prod_{j=0}^{c-1} CCZ_{j_1, j_2, j_3}, \text{ where } j_k = k + jc \quad (k = 1, 2, 3).$$

If $n > 3v$ then one can perform an arbitrary unitary on the remaining qubits using diagonal Clifford gates drawn from $\{Z, CZ\}$.

2. Deutsch-Jozsa algorithm

The Deutsch-Jozsa problem [47] is among the first examples of a problem whose quantum solution shows an exponential separation from the corresponding classical solution. Here we give a brief overview of the quantum algorithm; for additional details, see Ref. [24]. We are promised that a Boolean function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ is either constant or balanced, and we are given access to an oracle O_f which is an $(n+1)$ -qubit unitary operator whose action on the computational basis is given by $O_f|y, z\rangle = |y, z + f(y)\rangle$ with $y \in \mathbb{Z}_2^n$ and where the sum is taken modulo 2. Implementing the unitary $U = H^{\otimes n} O_f H^{\otimes(n+1)}$ we have a final quantum state given by $|\psi\rangle = U|0, 1\rangle$ wherein $\langle\psi|(\Pi^0 \otimes \mathbb{1})|\psi\rangle = 0, 1$ if f is balanced, constant, respectively.

Our explicit construction of the oracle O_f is as follows: Fix some positive integer c and let $n \geq 3c$. We consider Boolean functions $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ whose polynomial degree is at most quadratic in the inputs. When f is constant, we simply take $f(y) = 0$ for all $y \in \mathbb{Z}_2^n$ so that O_f is just the identity. For the balanced case, we take

$$f(y) = \sum_{j=0}^{c-1} y_{j_1} y_{j_2} + y_{j_3} + \sum_{k=3c+1}^n y_k.$$

The function f is readily seen to be a balanced function since it is a sum of balanced functions with disjoint domains in $\mathbb{Z}_2^n$. To implement the quantum oracle, recall first that the action of CX and the Toffoli gate CCX on the computational basis is $CX_{c,t}|y_c, y_t\rangle = |y_c, y_t + y_c\rangle$ and $CCX_{c_1, c_2, t}|y_{c_1}, y_{c_2}, y_t\rangle = |y_{c_1}, y_{c_2}, y_t + y_{c_1} y_{c_2}\rangle$, respectively. We then have that the

unitary implementing the oracle O_f is given by

$$O_f = \prod_{j=0}^{c-1} CCX_{j_1, j_2, j_3} CX_{j_3, t} \prod_{k=3c+1}^n CX_{k, t}.$$

To our knowledge, the Deutsch-Jozsa algorithm has not previously been used to benchmark Clifford+ T circuits. One advantage of our Deutsch-Jozsa oracle over that used in the hidden shift algorithm is that ours is more flexible in its use of non-Clifford gates, since it can allow for any number of CCX gates, while the hidden shift oracle always requires an even number of $CCZ = (\mathbb{1} \otimes \mathbb{1} \otimes H) CCX (\mathbb{1} \otimes \mathbb{1} \otimes H)$ gates.

3. Numerical implementation and results

We used our publicly available code in Ref. [21] to obtain Born rule estimates for the hidden shift and Deutsch-Jozsa circuits. For a quantum circuit with n qubits and m T gates our initial state in QCM will be given by

$$\rho = |0\rangle\langle 0| \otimes (|T\rangle\langle T|)^{\otimes m},$$

where

$$|T\rangle = T|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle),$$

and $|0\rangle$ is n -qubit all-zero state. It suffices, in fact, to decompose the m -fold tensor product of $|T\rangle\langle T|$ as a quasimixture of CNC operators since the tensor product of a CNC operator with a stabilizer state (Example 3) is once again a CNC operator.

Suppose that a CNC decomposition $W_C^{(k)}$ is found for k -fold tensor product of $|T\rangle$ and that stabilizer decompositions $W_S^{(\ell)}$ and $W_S^{(z)}$ are found for ℓ -fold and z -fold tensor products of the $|T\rangle$ state, respectively, such that $m = k + r\ell + z$, for some non-negative integers r, z . We then have that the sample complexity scales as

$$M \sim \frac{1}{\epsilon^2} \|W_C^{(k)}\|_1^2 \|W_S^{(\ell)}\|_1^{2r} \|W_S^{(z)}\|_1^2.$$

For fixed k the asymptotic scaling follows the stabilizer part of the decomposition with the CNC part giving a constant improvement over using stabilizer states alone. For practical implementations, we were able to obtain a CNC decomposition for $k = 4$ and stabilizer decompositions with $\ell = 4$ and $z \leq 3$. Precise numerical values of robustness can be found in Table I. We used these decompositions to estimate Born rule probabilities for the hidden shift and Deutsch-Jozsa algorithms; see Table II.

X. CONCLUSION

In this paper, we introduce a novel tableau representation for the phase-space simulation algorithm based on closed noncontextual (CNC) operators, which can be viewed as a generalization of stabilizer states. The resulting simulation algorithm efficiently simulates a broader class of quantum circuits beyond stabilizer circuits. In particular, we show that any quantum computation with magic states (QCM) initialized in a variational noncontextual eigenstate and acted on by stabilizer operations lies within this class. The complexity of

TABLE II. Simulation results for (a) hidden shift algorithm with $x = (1, \dots, 1) \in \mathbb{Z}_2^n$ and (b) Deutsch-Jozsa algorithm with $x = (0, \dots, 0) \in \mathbb{Z}_2^n$. The run-time τ is given in seconds. See Ref. [21] for implementation details.

(a)							
n	m	ϵ	δ	$\ W\ _1$	M	τ (s)	$\hat{p}(x)$
6	14	0.1	0.1	31.1	579 626	1284	1.010
106	14	0.1	0.1	31.1	579 626	19 022	1.000
6	14	0.01	0.1	31.1	57 962 672	127 688	0.997

(b)							
n	m	ϵ	δ	$\ W\ _1$	M	τ (s)	$\hat{p}(x)$
4	7	0.1	0.1	4.82	13 914	13	0.000
104	7	0.1	0.1	4.82	13 914	373	0.000
4	7	0.01	0.1	4.82	1 391 418	1314	0.000
104	7	0.01	0.1	4.82	1 391 418	36 817	0.000
7	14	0.1	0.1	31.1	579 626	1162	0.000
107	14	0.1	0.1	31.1	579 626	18 199	0.000
7	14	0.01	0.1	31.1	57 962 672	36 816	0.000

our simulator scales similarly to the well-known Gottesman-Knill stabilizer tableau simulation. We have implemented our simulator in Ref. [21] and benchmarked it on basic quantum algorithms, including the hidden shift and Deutsch-Jozsa algorithms.

The phase-space simulator is part of a more general class of simulators based on polytope-theoretic methods. These methods can be utilized to provide a systematic approach to the problem of identifying quantum resources responsible for quantum advantage. The Λ polytope and the corresponding simulator introduced in Ref. [26] are the first examples of a probabilistic representation of universal quantum computation on a finite state space. Our phase space points are also vertices of the Λ polytope, specifying a region of efficiently simulatable quantum states. However, discovering more vertices of Λ can extend this region. Therefore, the systematic and refined analysis of the symplectic structure of the qubit phase space presented in our work serves as an important step in identifying additional combinatorial structures, e.g., see Ref. [50], that can be used for efficiently describing and updating more Λ vertices.

On the other hand, exploring computational models beyond QCM, such as measurement-based quantum computation [51], introduces a new family of polytopes that provide a probabilistic representation of quantum theory; see Ref. [52] for the local Pauli simulator. The tableau method introduced in this work may be extended to incorporate newly discovered types of vertices, ultimately expanding the region of efficiently simulatable quantum states and encompassing a broader class of quantum circuits. As more vertices are discovered, the simulation polytope expands, leading to a decrease in the corresponding robustness measure. This, in turn, improves efficiency when the algorithm is initialized with a quasiprobabilistic representation of the resource state.

ACKNOWLEDGMENTS

This work is supported by the Digital Horizon Europe project FoQaCiA, GA No. 101070558. The last author also acknowledges support from the US Air Force Office of Scientific Research under Award No. FA9550-24-1-0257.

DATA AVAILABILITY

The data that support the findings of this article are openly available [21], embargo periods may apply.

APPENDIX A: PROOF OF CNC CLASSIFICATION

Lemma A1. Let $p_1, p_2, p_3, p_4 \in E_n$ satisfying

$$\begin{aligned} [p_1, p_2] &= [p_1, p_3] = [p_2, p_4] = [p_3, p_4] = 0, \\ [p_1, p_4] &= [p_2, p_3] = 1. \end{aligned} \quad (\text{A1})$$

Then there is no CNC set that contains $\{p_1, p_2, p_3, p_4\}$.

Proof. Let Ω be a CNC set that contains $\{p_1, p_2, p_3, p_4\}$ and $\gamma : \Omega \rightarrow \mathbb{Z}_2$ be a value assignment on Ω . By closedness, we know that $p_1 + p_2, p_1 + p_3, p_2 + p_4, p_3 + p_4 \in \Omega$. Let $\gamma(p_i) = \alpha_i$ for all $i \in \{1, 2, 3, 4\}$ and $\alpha = \sum_{i=1}^4 \alpha_i$. Then,

$$\begin{aligned} &(-1)^\alpha T_{p_1} T_{p_2} T_{p_3} T_{p_4} \\ &= (-1)^\alpha [(-1)^{\beta(p_1, p_2)} T_{p_1+p_2}] [(-1)^{\beta(p_3, p_4)} T_{p_3+p_4}] \\ &= [(-1)^{\alpha_1+\alpha_2+\beta(p_1, p_2)} T_{p_1+p_2}] [(-1)^{\alpha_3+\alpha_4+\beta(p_3, p_4)} T_{p_3+p_4}] \\ &= (-1)^{\gamma(p_1+p_2)+\gamma(p_3+p_4)} T_{p_1+p_2} T_{p_3+p_4} \\ &= (-1)^{\gamma(p_1+p_2)+\gamma(p_3+p_4)+\beta(p_1+p_2, p_3+p_4)} T_{p_1+p_2+p_3+p_4} \\ &= (-1)^{\gamma(p_1+p_2+p_3+p_4)} T_{p_1+p_2+p_3+p_4}. \end{aligned}$$

By similar calculations, we also find that

$$(-1)^\alpha T_{p_1} T_{p_3} T_{p_2} T_{p_4} = (-1)^{\gamma(p_1+p_2+p_3+p_4)} T_{p_1+p_2+p_3+p_4}.$$

After cancellations, these imply that $T_{p_2} T_{p_3} = T_{p_3} T_{p_2}$ which is a contradiction because we assumed $[p_2, p_3] = 1$. Hence, there is no such CNC set. ■

Corollary A1. Let $q_1, q_2, q_3, q_4 \in \Omega$ for some closed subset $\Omega \subset E_n$. If any of the following conditions are satisfied, Ω is not a CNC.

$$(1) [q_1, q_2] = [q_2, q_3] = [q_1, q_4] = 0 \quad \text{and} \quad [q_1, q_3] = [q_2, q_4] = [q_3, q_4] = 1.$$

$$(2) [q_1, q_2] = [q_2, q_3] = 0 \quad \text{and} \quad [q_1, q_3] = [q_1, q_4] = [q_2, q_4] = [q_3, q_4] = 1.$$

Proof. (i) Since Ω is closed, we have $q_1 + q_4, q_2 + q_3 \in \Omega$. Taking $p_1 = q_3, p_2 = q_2 + q_3, p_3 = q_1 + q_4, p_4 = q_4$ satisfies Eq. (A1). (ii) We have $q_1 + q_2, q_2 + q_3 \in \Omega$ by closedness. Taking $p_1 = q_1 + q_2, p_2 = q_2, p_3 = q_4, p_4 = q_2 + q_3$ satisfies Eq. (A1). ■

Lemma A2. Let $\Omega \subset E_n$ be a CNC set and $a_1, a_2, a_3 \in \Omega$. If $[a_1, a_2] = [a_2, a_3] = 0$ and $a_2 \notin I(\Omega)$ then $[a_1, a_3] = 0$.

Proof. Assume that $[a_1, a_2] = [a_2, a_3] = 0$ and $a_2 \notin I(\Omega)$ but $[a_1, a_3] = 1$. Take $a_4 \in \Omega$. There are eight possibilities for the tuple $([a_1, a_4], [a_2, a_4], [a_3, a_4])$. We need to show that all of them are impossible. We show it in five cases.

$$(1) [a_2, a_4] = 0:$$

We have $[a_i, a_2] = 0$ for all $i \in \{1, 3, 4\}$. Since $a_2 \notin I(\Omega)$, there exists $b \in \Omega$ such that $[b, a_2] = 1$. Let $\eta_1 = [a_1, b]$, $\eta_3 = [a_3, b]$. Then taking $p_1 = a_1 + \eta_1 a_2$, $p_2 = a_2$, $p_3 = b$, $p_4 = \eta_3 a_2 + a_3$ satisfies Eq. (A1).

(2) $([a_1, a_4], [a_2, a_4], [a_3, a_4]) = (0, 1, 0)$:

Taking $p_i = a_i$ for all $i \in \{1, 2, 3, 4\}$ satisfies Eq. (A1).

(3) $([a_1, a_4], [a_2, a_4], [a_3, a_4]) = (0, 1, 1)$:

Taking $q_i = a_i$ for all $i \in \{1, 2, 3, 4\}$ satisfies case (i) of Corollary A1.

(4) $([a_1, a_4], [a_2, a_4], [a_3, a_4]) = (1, 1, 0)$:

By symmetry, (iii) also proves this case.

(5) $([a_1, a_4], [a_2, a_4], [a_3, a_4]) = (1, 1, 1)$:

Taking $q_i = a_i$ for all $i \in \{1, 2, 3, 4\}$ satisfies the case (ii) of Corollary A1.

Hence, all of the cases are impossible, and we have a contradiction. ■

Proof. Let us write $I = I(\Omega)$ and $W = W(\Omega)$. If $\Omega = I$ then choosing $N = 1$ and $a_1 = 0$ gives the desired decomposition. So, assume that $\Omega \neq I(\Omega)$ and $N \geq 2$. Let $\sim$ be the relation on $\{a_1, \dots, a_N\}$ defined as $x \sim y$ if and only if $[x, y] = 0$. Clearly, it is reflexive and symmetric. By Lemma A2, $\sim$ is also transitive. Therefore, $\sim$ is an equivalence relation on the set $\{a_1, \dots, a_N\}$. This set is partitioned into equivalence classes under $\sim$. Moreover, $\Omega \neq I$ implies that there are at least two equivalence classes. Let x_1, x_2 be two distinct elements in the same equivalence class, i.e., $[x_1, x_2] = 0$. Then, $x_1 + x_2 \neq 0$ belongs to Ω by closedness and $[x_1, x_1 + x_2] = [x_2, x_1 + x_2] = 0$ implies $x_1 + x_2$ must lie in the equivalence class containing x_1 and x_2 . Since there are at least two equivalence classes, there must exist $y \in \{a_1, \dots, a_N\} \setminus \{x_1, x_2, x_1 + x_2\}$ such that $[x_1, y] = [x_2, y] = [x_1 + x_2, y] = 1$. However, this is not possible since $[x_1 + x_2, y] = [x_1, y] + [x_2, y]$. So, there cannot be an equivalence class of $\sim$ with size greater than 1. Therefore, all equivalence classes of $\sim$ are singletons, which means $\{a_1, \dots, a_N\}$ is an anticommuting set. Since W is a $2m$ -dimensional symplectic vector space and $\{a_1, \dots, a_N\}$ is an anticommuting set in W we must have $N \leq 2m + 1$. ■

Proof of Corollary 2. First assume that $\Omega = I$. Let $I = \langle e_1, \dots, e_n \rangle$ and define $J = \langle e_1, \dots, e_{n-1} \rangle$. Then there exists a symplectic subspace W of dimension 2 such that $J^\perp = J \oplus W$. Clearly, $\{e_n\}$ is an anticommuting set in W that is not maximal. Therefore, we can extend it to a JW set $\{e_n, a_1, a_2\}$ in W by Lemma 2. Then $\Omega = \langle e_n, J \rangle \cup \langle a_1, J \rangle \cup \langle a_2, J \rangle$ is a CNC containing I .

Next, assume that $1 < N < 2m + 1$. If $\{a_1, \dots, a_N\}$ is not a maximal anticommuting set, then we can extend it to a JW set $\{a_1, \dots, a_N, a_{N+1}, \dots, a_{2m+1}\}$ of W by Lemma 2. Then, we see that $\tilde{\Omega} = \bigcup_{k=1}^{2m+1} \langle a_k, I \rangle$ is a CNC set that strictly contains Ω . So Ω is not maximal. If $\{a_1, \dots, a_N\}$ is a maximal anticommuting set, then $N = 2l + 1$ for some $1 \leq l < m$ and $a_{2l+1} = \sum_{i=1}^{2l} a_i$. So, we have $a_{2l+1} \in S = \langle a_1, \dots, a_{2l} \rangle$ for some symplectic subspace S of dimension $2l$. Since W and S are symplectic subspaces, we can find conjugate isotropics J, J' and write a JW decomposition $W = S \oplus J \oplus J'$. We have $J^\perp = S \oplus J$. Define $\tilde{\Omega} = \bigcup_{k=1}^{2l+1} \langle a_k, \tilde{I} \rangle$ where $\tilde{I} = I \oplus J$. We can see that $\tilde{\Omega}$ is a CNC set that strictly contains Ω . So, Ω is not maximal. Therefore, maximal CNC sets have $N = 2m + 1$.

Lastly, we show that if $N = 2m + 1$ for some $1 \leq m \leq n$ then Ω is a maximal CNC. Assume that $\{a_1, \dots, a_{2m+1}\}$ is an anticommuting set in $I^\perp$ and $\Omega = \bigcup_{k=1}^{2m+1} \langle a_k, I \rangle$. Let $\tilde{a}_i$ be the projection of a_i onto W . Then $\{\tilde{a}_1, \dots, \tilde{a}_{2m+1}\}$ is an anticommuting set in W and $\Omega = \bigcup_{k=1}^{2m+1} \langle \tilde{a}_k, I \rangle$. Let $\tilde{\Omega}$ be a maximal CNC set such that $\Omega \subset \tilde{\Omega}$. Denote $I(\tilde{\Omega})$ by $\tilde{I}$ and $W(\tilde{\Omega})$ by $\tilde{W}$. Let $\dim(\tilde{I}) = n - \tilde{m}$ for some $1 \leq \tilde{m} \leq n$. Then there exists an anticommuting set $\{b_1, \dots, b_{2\tilde{m}+1}\}$ in $\tilde{I}^\perp$ such that $\tilde{\Omega} = \bigcup_{k=1}^{2\tilde{m}+1} \langle b_k, \tilde{I} \rangle$. Since $\{a_1, \dots, a_{2m+1}\}$ is an anticommuting set, each a_i must be in a different isotropic subspace $\langle b_k, \tilde{I} \rangle$. Hence, we must have $\tilde{m} \geq m$. Also, $|\Omega| = (2m + 2)2^{n-m} \leq (2\tilde{m} + 2)2^{n-\tilde{m}} = |\tilde{\Omega}|$ implies $m \geq \tilde{m}$. Therefore, $m = \tilde{m}$, which implies $\Omega = \tilde{\Omega}$. Hence, Ω is maximal. ■

ALGORITHM 1. Algorithm for simulating measurement outcome and tableau update when $b \in I^\perp \setminus \Omega$.

Input: Current tableau $\tilde{T}_\Omega^\gamma$, Measurement basis $R(b)$, commutation information $\mathcal{T}_\Omega(b)$.

Output: Updated tableau $\tilde{T}_{\Omega(b)}^{(\gamma * r_b) * s}$ and outcome r_b .

1. **Reorganize JW Elements:**

- Identify indices of commuting and anticommuting JW elements using $\mathcal{T}_\Omega(b)$.
- Compute $t \leftarrow \frac{1}{2}|\mathcal{A}_b|$.
- Swap rows in the tableau to place anticommuting elements in the first $2t$ rows.

2. **Generate New Stabilizers and Destabilizers:**

- Let $K = 2(n - m)$ and randomly sample the measurement outcome $r_b \in \mathbb{Z}_2$.
- First Stabilizer and Destabilizer:
 - For $i = K + 2t + 1, \dots, 2n + 1$:
 - $R_{K+1} \leftarrow R_{K+1} + R_i$.
 - $\tilde{R}_{K+1} \leftarrow (R_{K+1}, r_b)$ and $\tilde{R}_{K+2} \leftarrow (R_{K+2}, 0)$.
- Generate Remaining New Stabilizers and Destabilizers:
 - Introduce extra row: $R(S) \leftarrow R_{K+1} + R_{K+2}$.
 - For $i = 2, \dots, t$:
 - Randomly choose $s_i \in \mathbb{Z}_2$.
 - Destabilizer: $R_{K+2i} \leftarrow R_{K+2i} + R(S)$, $\tilde{R}_{K+2i} \leftarrow (R_{K+2i}, 0)$.
 - Stabilizer: $R_{K+2i-1} \leftarrow R_{K+2i-1} + R(S)$, $\tilde{R}_{K+2i-1} \leftarrow (R_{K+2i-1}, s_i)$.
 - Update $R(S) \leftarrow R(S) + R_{K+2i-1} + R_{K+2i}$.

3. **Update Commuting JW Elements:**

- For $i = K + 2t + 1, \dots, 2n + 1$:
 - $\tilde{R}_i \leftarrow \tilde{R}_i * \tilde{R}_{K+1}$.

4. **Rearrange Stabilizers and Destabilizers:**

- Organize rows as follows
 - Rows $1, \dots, n - m$ are the previous destabilizer rows.
 - Rows $n - m + 1, \dots, n - m + t$ are the new destabilizer rows $K + 2i$ ($i = 1, \dots, t$).
 - Rows $n - m + t + 1, \dots, 2(n - m) + t$ are the previous stabilizer rows.
 - Rows $2(n - m) + t + 1, \dots, 2[n - (m - t)]$ are the new stabilizer rows $K + 2i - 1$ ($i = 1, \dots, t$).

• **Update type:**

- Update $m \leftarrow m - t$.

- **Return:** Updated tableau and measurement outcome r_b .
-

APPENDIX B: PSEUDOCODE FOR THE ALGORITHMS

This appendix collects pseudocode for the subroutines to implement the phase-space tableau simulation and update rules. Algorithms 1 and 2 give the measurement-step update of the extended tableau and the sampled outcome in the two nontrivial measurement regimes, namely when the measured Pauli label b lies in $I^\perp \setminus \Omega$ and when $b \notin I^\perp$, respectively. Algorithm 3 constructs an initial tableau from a given maximal CNC operator by extracting bases for the center I , its destabilizer complement, and the Jordan–Wigner (JW) elements. Finally, Algorithms 4 and 5 implement the resulting sampling procedures: weak simulation by iterated instrument updates, and additive-precision estimation of Born-rule.

ALGORITHM 2. Algorithm for simulating measurement outcome and tableau update when $b \notin I^\perp$.

Input: Current tableau $\bar{\mathcal{T}}_\Omega^\gamma$, Measurement basis $R(b)$, commutation information $\mathcal{T}_\Omega(b)$.
Output: Updated tableau $\bar{\mathcal{T}}_{\Omega(b)}^{\gamma * r_b}$ and outcome r_b .

1. **Find Minimum Anticommuting Stabilizer:**
 - $\lambda \leftarrow \min\{i + n - m\}_{i=1}^{n-m}$ such that $[R_\lambda, R(b)] = 1$.
2. **Modify Anticommuting Rows:**
 - **For** each row $i \notin \{\lambda, \lambda - n + m\}$:
 - If $[R_i, R(b)] = 1$ then $\tilde{R}_i \leftarrow \tilde{R}_i * \tilde{R}_\lambda$.
3. **Update Stabilizer and Destabilizer:**
 - New Destabilizer: $R_{\lambda-n+m} \leftarrow R_\lambda$.
 - New Stabilizer:
 - Randomly sample the measurement outcome $r_b \in \mathbb{Z}_2$.
 - $\tilde{R}_\lambda \leftarrow (R(b), r_b)$.
4. **Return:** Updated tableau and measurement outcome r_b .

ALGORITHM 3. Algorithm for creating a tableau from given maximal CNC operator A_Ω^γ .

Input: Maximal CNC operator A_Ω^γ .
Output: Tableau $\mathcal{T}_\Omega^\gamma$.

1. **Find the Stabilizer:**
 - Define I as the center of Ω , i.e., the set of all elements that commute with every element of Ω .
 - Find a basis $\{e_1, \dots, e_{n-m}\}$ for I .
2. **Find the Destabilizers:**
 - Extend $\{e_1, \dots, e_{n-m}\}$ to a symplectic basis $\{e_1, f_1, \dots, e_n, f_n\}$ of E_n using symplectic Gram-Schmidt orthogonalization.
 - Then $\{f_1, \dots, f_{n-m}\}$ is a basis for the destabilizer I' .
3. **Find JW elements:**
 - Define the symplectic subspace W as

$$W = \langle e_{n-m+1}, f_{n-m+1}, \dots, e_n, f_n \rangle.$$
 - Let P_W be the orthogonal projection onto W and set $\Omega' = \Omega \setminus I$.
 - Initialize $JW := \{\}$.
 - **For** $i = 1$ **to** $2m + 1$:
 - Choose $\hat{a}_i \in \Omega'$ and compute $a_i = P_W(\hat{a}_i)$.
 - Add a_i to JW .
 - Remove all elements that commute with a_i from Ω' .

ALGORITHM 3. (Continued).

4. **Construct the Tableau:**

- Construct tableau $\mathcal{T}_\Omega^\gamma$ as follows (note that the phases of the destabilizers are not important):

$$\mathcal{T}_\Omega^\gamma = \left(\begin{array}{c|c} \begin{matrix} f_1 \\ \vdots \\ f_{n-m} \\ e_1 \\ \vdots \\ e_{n-m} \\ a_1 \\ \vdots \\ a_{2m+1} \end{matrix} & \begin{matrix} * \\ \vdots \\ * \\ \gamma(e_1) \\ \vdots \\ \gamma(e_{n-m}) \\ \gamma(a_1) \\ \vdots \\ \gamma(a_{2m+1}) \end{matrix} \end{array} \right) \left. \begin{array}{l} \left. \begin{matrix} * \\ \vdots \\ * \end{matrix} \right\} \text{Destabilizer} \\ \left. \begin{matrix} \gamma(e_1) \\ \vdots \\ \gamma(e_{n-m}) \end{matrix} \right\} \text{Stabilizer} \\ \left. \begin{matrix} \gamma(a_1) \\ \vdots \\ \gamma(a_{2m+1}) \end{matrix} \right\} \text{JW elements} \end{array} \right\}.$$

5. **Return:** Tableau $\mathcal{T}_\Omega^\gamma$.

ALGORITHM 4. Algorithm for weak simulation using the phase space tableau.

Input: Initial distribution $W_\rho \geq 0$. Stabilizer instruments $\Phi_1, \dots, \Phi_N$.
Output: Set of outcomes $\mathcal{L} = \{s_1, \dots, s_T\}$.

1. Initialize empty outcome set $\mathcal{L}$.
2. Sample initial tableau $\tilde{\mathcal{T}}_{\alpha_0}$ from W_ρ .
3. **For** $i = 1, \dots, T$:
 - For instrument Φ_i , update tableau $\tilde{\mathcal{T}}_{\alpha_{i-1}} \mapsto \tilde{\mathcal{T}}_{\alpha_i}$.
 - Obtain outcome s_i from tableau update: $\mathcal{L} \leftarrow \mathcal{L} \cup \{s_i\}$.
4. Output $\mathcal{L}$.

ALGORITHM 5. Algorithm for producing additive-precision estimates of Born rule probabilities.

Input: Initial distribution $W_\rho \geq 0$. Stabilizer instruments $\Phi_1, \dots, \Phi_N$. Number of samples M .
Output: Estimate of $p(\vec{x})$.

1. Initialize estimator $\hat{p} \leftarrow 0$.
2. **For** $i = 1, \dots, M$:
 - Sample α_i from $\tilde{W}_\rho$.
 - Estimate $\hat{E}_i$: $\hat{E}_i \leftarrow E(x|\alpha_i)$.
 - $\hat{p} \leftarrow \hat{p} + \hat{E}_i$.
3. Output $\hat{p}/M$.

- [1] S. Aaronson and L. Chen, Complexity-theoretic foundations of quantum supremacy experiments, in *Proceedings of the 32nd Computational Complexity Conference, CCC '17* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, 2017).
- [2] D. Gottesman, The Heisenberg Representation of Quantum Computers, in *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics: Hobart, July 13/17, 1998*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis (International Press, Cambridge, MA, 1999), pp. 32–43.
- [3] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Phys. Rev. A* **70**, 052328 (2004).
- [4] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, *Phys. Rev. A* **71**, 022316 (2005).
- [5] S. Bravyi, G. Smith, and J. A. Smolin, Trading classical and quantum computational resources, *Phys. Rev. X* **6**, 021043 (2016).
- [6] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, *Phys. Rev. Lett.* **116**, 250501 (2016).
- [7] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Simulation of quantum circuits by low-rank stabilizer decompositions, *Quantum* **3**, 181 (2019).
- [8] V. Veitch, C. Ferrie, D. Gross, and J. Emerson, Negative quasiprobability as a resource for quantum computation, *New J. Phys.* **14**, 113011 (2012).
- [9] N. Delfosse, P. Allard Guerin, J. Bian, and R. Raussendorf, Wigner function negativity and contextuality in quantum computation on rebits, *Phys. Rev. X* **5**, 021003 (2015).
- [10] R. Raussendorf, D. E. Browne, N. Delfosse, C. Okay, and J. Bermejo-Vega, Contextuality and Wigner-function negativity in qubit quantum computation, *Phys. Rev. A* **95**, 052334 (2017).
- [11] R. Raussendorf, J. Bermejo-Vega, E. Tyhurst, C. Okay, and M. Zurel, Phase-space-simulation method for quantum computation with magic states on qubits, *Phys. Rev. A* **101**, 012350 (2020).
- [12] D. Gross, Hudson's theorem for finite-dimensional quantum systems, *J. Math. Phys.* **47**, 122107 (2006).
- [13] N. D. Mermin, Hidden variables and the two theorems of John Bell, *Rev. Mod. Phys.* **65**, 803 (1993).
- [14] W. M. Kirby and P. J. Love, Contextuality test of the nonclassicality of variational quantum eigensolvers, *Phys. Rev. Lett.* **123**, 200501 (2019).
- [15] W. M. Kirby and P. J. Love, Classical simulation of noncontextual Pauli Hamiltonians, *Phys. Rev. A* **102**, 032418 (2020).
- [16] W. M. Kirby, A. Tranter, and P. J. Love, Contextual subspace variational quantum eigensolver, *Quantum* **5**, 456 (2021).
- [17] H. Pashayan, J. J. Wallman, and S. D. Bartlett, Estimating outcome probabilities of quantum circuits using quasiprobabilities, *Phys. Rev. Lett.* **115**, 070501 (2015).
- [18] M. Howard and E. Campbell, Application of a resource theory for magic states to fault-tolerant quantum computing, *Phys. Rev. Lett.* **118**, 090501 (2017).
- [19] C. Hindlycke and J.-Å. Larsson, Efficient contextual ontological model of n-qubit stabilizer quantum mechanics, *Phys. Rev. Lett.* **129**, 130401 (2022).
- [20] A. Karanjai, J. J. Wallman, and S. D. Bartlett, Contextuality bounds the efficiency of classical simulation of quantum processes, [arXiv:1802.07744](https://arxiv.org/abs/1802.07744).
- [21] BilQCT, CNCSim, <https://github.com/BilQCT/CNCSim>, accessed: 2025-02-28.
- [22] Our row product operation corresponds to the **rowsum** operation of AG [3].
- [23] C. Gidney, Python CHP stabilizer simulator, <https://github.com/Strilanc/python-chp-stabilizer-simulator>, accessed: 2025-03-21.
- [24] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
- [25] C. Okay, S. Roberts, S. D. Bartlett, and R. Raussendorf, Topological proofs of contextuality in quantum mechanics, *Quantum Inf. Comput.* **17**, 1135 (2017).
- [26] M. Zurel, C. Okay, and R. Raussendorf, Hidden variable model for universal quantum computation with magic states on qubits, *Phys. Rev. Lett.* **125**, 260404 (2020).
- [27] M. Heinrich and D. Gross, Robustness of magic and symmetries of the stabiliser polytope, *Quantum* **3**, 132 (2019).
- [28] L. G. Valiant, Quantum circuits that can be simulated classically in polynomial time, *SIAM J. Comput.* **31**, 1229 (2002).
- [29] B. M. Terhal and D. P. DiVincenzo, Classical simulation of noninteracting-fermion quantum circuits, *Phys. Rev. A* **65**, 032325 (2002).
- [30] R. Jozsa and A. Miyake, Matchgates and classical simulation of quantum circuits, *Proc. R. Soc. London Ser. A* **464**, 3089 (2008).
- [31] C. Okay, M. Zurel, and R. Raussendorf, On the extremal points of the Λ -polytopes and classical simulation of quantum computation with magic states, *Quantum Inf. Comput.* **21**, 1091 (2021).
- [32] T. Weaving, A. Ralli, W. M. Kirby, A. Tranter, P. J. Love, and P. V. Coveney, A stabilizer framework for the contextual subspace variational quantum eigensolver and the noncontextual projection ansatz, *J. Chem. Theory Comput.* **19**, 808 (2023).
- [33] R. Sarkar and E. van den Berg, On sets of maximally commuting and anticommuting Pauli operators, *Res. Math. Sci.* **8**, 14 (2021).
- [34] A. Cannas da Silva, *Lectures on Symplectic Geometry*, Lecture Notes in Mathematics, Vol. 1764 (Springer, Berlin, Heidelberg, 2001).
- [35] R. Koenig and J. A. Smolin, How to efficiently select an arbitrary Clifford group element, *J. Math. Phys.* **55**, 122202 (2014).
- [36] To see this, notice that there is a local Clifford $V \in Cl_1$ such that $V(X) = X$, $V(Y) = -Y$, $V(Z) = Z$, and if we apply it to the n th qubit of the $\{T_{\hat{a}_i}\}_{i=1}^{2m+1}$, it only flips the phase of the $T_{\hat{a}_{2m+1}}$. Combining V with U_α gives the desired result.
- [37] M. Zurel, Hidden variable models and classical simulation algorithms for quantum computation with magic states on qubits, Master's thesis, University of British Columbia, 2020.
- [38] If we measure $-T_b$ we flip the outcomes $0 \leftrightarrow 1$.
- [39] J. Watrous, *The Theory of Quantum Information* (Cambridge University Press, 2018).
- [40] J. R. Seddon and E. T. Campbell, Quantifying magic for multi-qubit operations, *Proc. R. Soc. London Ser. A* **475**, 20190251 (2019).
- [41] We do not consider the most general notion of a completely stabilizer-preserving channel. An extensive study of completely stabilizer-preserving operations and their related monotones was performed in Ref. [40].

- [42] R. Jozsa and M. Van Den Nest, Classical simulation complexity of extended Clifford circuits, [Quantum Info. Comput.](#) **14**, 633 (2014).
- [43] H. Pashayan, S. D. Bartlett, and D. Gross, From estimation of quantum probabilities to simulation of quantum circuits, [Quantum](#) **4**, 223 (2020).
- [44] It was shown by Jozsa and Van den Nest [42] that computing exact Born rule probabilities (i.e., strong simulation) of *adaptive* stabilizer circuits is in the complexity class—and thus unlikely to be efficient in general.
- [45] W. Hoeffding, Probability inequalities for sums of bounded random variables, in *The Collected Works of Wassily Hoeffding*, edited by N. I. Fisher and P. K. Sen, Springer Series in Statistics (Springer, New York, 1994), pp. 409–426
- [46] M. Rötteler, Quantum algorithms for highly non-linear Boolean functions, in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, 2010), pp. 448–457.
- [47] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, [Proc. R. Soc. London Ser. A](#) **439**, 553 (1992).
- [48] H. Pashayan, O. Reardon-Smith, K. Korzekwa, and S. D. Bartlett, Fast estimation of outcome probabilities for quantum circuits, [PRX Quantum](#) **3**, 020361 (2022).
- [49] A. Kissinger and J. van de Wetering, Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions, [Quantum Sci. Technol.](#) **7**, 044001 (2022).
- [50] M. Zurel, L. Z. Cohen, and R. Raussendorf, Simulation of quantum computation with magic states via Jordan-Wigner transformations, [Phys. Rev. A](#) **112**, 042602 (2025).
- [51] R. Raussendorf and H. J. Briegel, A one-way quantum computer, [Phys. Rev. Lett.](#) **86**, 5188 (2001).
- [52] C. Okay, A. T. Yucel, and S. Ipek, Classical simulation of universal measurement-based quantum computation using multipartite Bell scenarios, [arXiv:2410.23734](#).