

# Extended Abstract: Reducing the weight of Pauli measurements in Pauli-based computation

## Submission for a talk at the 6th International Workshop on Quantum Compilation

Filipa C. R. Peres<sup>1,2</sup> and Ernesto F. Galvão<sup>1,3</sup>

<sup>1</sup>*International Iberian Nanotechnology Laboratory (INL), Braga, Portugal*

<sup>2</sup>*Departamento de Física e Astronomia, Faculdade de Ciências, Universidade do Porto, Porto, Portugal*

<sup>3</sup>*Instituto de Física, Universidade Federal Fluminense, Niterói, Rio de Janeiro, Brazil*

**1. Introduction.** Quantum computing stands at the forefront of technological innovation, promising unprecedented computational power and transformative capabilities. However, current hardware remains somewhat limited in its capabilities. Thus, the minimization of quantum resources promises to play a relevant role in current and near-term implementations. This has prompted extensive research into quantum circuit optimization, using a myriad of different techniques including, for instance, pattern matching [1, 2], ZX-calculus [3, 4], and phase polynomials [5–7], and leading to important reductions in the number of operations and/or computational depth of the circuits needed to carry out a certain task. In this work, we explore Pauli-based computation (PBC) [8] a universal model for quantum computation driven by an adaptive sequence of  $n$  independent and compatible Pauli measurements performed on a set of  $n$  qubits initialized in a separable magic state.

**2. Background.** Any (universally general) Clifford+ $T$  quantum circuit with  $n$  qubits and  $t$   $T$  gates can be transformed into and simulated by a PBC on  $t$  qubits initialized in the magic state  $|T\rangle^{\otimes t}$ . Recent work [9] demonstrated that PBC can be useful for compiling some families of quantum circuits dominated by Clifford gates; more specifically, the overall gate counts and depth of many of these quantum circuit instances can be significantly reduced by transforming each of them into a PBC which is then translated into adaptive Clifford circuits with magic-state input. The latter are called *PBC-compiled circuits* [9]. Importantly, the overall depth and gate counts of the final (compiled) circuits are intimately connected to the *weight* of the Pauli measurements in the corresponding PBC. For this reason, this is *the* measure to optimize within the PBC framework to enable not only a more practical (native) implementation of the model itself but also to enhance its performance as a circuit compilation tool. Up to now, no non-trivial upper bounds were known for the weight of each Pauli measurement in the PBC. That is, each measurement could have a weight up to  $t$ , although our prior numerical results [9] suggest that the weight is often smaller than that.

PBC is substantially different from other existing methods for reducing quantum resources. Compilation techniques often apply to non-adaptive quantum circuits and strategies for resource optimization for adaptive quantum computations are somewhat lacking in the literature. PBC addresses this on its own and the work presented here drives this even further, presenting a wide set of results that collectively improve the practical feasibility of this computational model and thus enhance its potential as a circuit compilation tool.

**3. Pre-compilation.** Our set of theoretical results stems from recognizing that, rather than directly transforming the desired Clifford+ $T$  quantum circuit with  $t$   $T$  gates into a PBC, one can start by expressing it as a one-way quantum computation [10, 11] on a  $t$ -qubit graph state. This initial step leads to novel upper bounds on the weight and depth of the corresponding PBC obtained from the latter computation.

**Theorem 1** (Improved weights). *Consider a one-way computation to be carried out on a  $t$ -qubit, computation-specific graph state  $|\mathcal{G}\rangle$  with a measurement pattern requiring only measurements along the  $\pm\pi/4$  directions on the equator of the Bloch sphere. By processing the measurements in a suitable order  $\mathcal{O}_1$  (compatible with the adaptive structure of the one-way computation), the weights of the  $2t$  Pauli operators in the (complete) PBC procedure are upper-bounded by  $\{1, 1, 2, 2, \dots, t-1, t-1, t, t\}$ .*

In this extended abstract, we avoid delving into details concerning the meaning of the processing orders  $\mathcal{O}_i$  used in the different theorems. This is something that would require an excessively lengthy explanation of the inner workings of PBC and one-way quantum computation. The reader needs only to take away that these processing orders pertain to the order in which we choose to handle the different measurements in the PBC, which is constrained by the adaptivity structure of the corresponding, underlying one-way quantum computation. Different orderings are compatible with such a structure and lead to different theorems (cf. Theorems 1, 3 and 4). For details, see the full manuscript with our work.

We know that the number of independent and pairwise commuting Pauli operators on  $t$  qubits is  $t$ . Therefore, of the list of  $2t$  Paulis mentioned in Theorem 1, at most  $t$  of those correspond to measurements that have to be performed in the quantum computer. (The PBC procedure guarantees that the remaining  $t$  measurements can be handled efficiently in a classical computer.) Assuming the worst-case scenario where the Pauli operators to be

measured are the ones with larger weight, i.e., the last  $t$  Paulis in the sequence, leads straightforwardly to the following Corollary.

**Corollary 2** (Average weight upper bound). *The average weight of the Pauli operators that need to be measured in the quantum hardware is upper bounded by  $\bar{w} \leq 3t/4 + 1/2$ .*

**Theorem 3** (Improved depth). *By taking up a suitable order  $\mathcal{O}_2$  (compatible with the adaptive structure of the problem), the depth of a PBC is upper-bounded by the logical depth of the corresponding one-way quantum computation,  $d_{1W}$ .*

Importantly, Theorems 1 and 3 do not hold simultaneously since the orderings  $\mathcal{O}_1$  and  $\mathcal{O}_2$  used to obtain them are different. A natural follow-up question is whether a result exists that provides both a weight and depth improvement with respect to the trivial upper bounds. This is answered in the affirmative by the following theorem.

**Theorem 4** (Weight-depth trade-off). *Consider a one-way computation with logical depth  $d_{1W}$  and layering so that the number of computational qubits in layer  $\ell_i$  is  $\kappa_i$ :  $\sum_{i=1}^{d_{1W}} \kappa_i = t$ . By taking up the processing order  $\mathcal{O}_3$  (compatible with the adaptive structure of the problem), the depth of the corresponding PBC is upper-bounded by  $\min\{2d_{1W} - 1, t\}$ . Moreover, the weight of the  $2\kappa_i$  Pauli measurements associated with the layer  $\ell_i$  is upper bounded by  $\sum_{j=1}^i \kappa_j$ .*

By improving the weight and/or depth of PBC, one has a direct impact on relevant parameters of the final, PBC-compiled quantum circuits such as the number of CNOT gates and the logical depth. In our work, we go beyond these theoretical results and give numerical evidence that these theorems provide not only non-trivial upper bounds to the properties of the PBC but also reduce the depth and gate counts of the compiled quantum circuits.

**4. Greedy algorithm.** Taking a different approach, we also provide a novel greedy algorithm capable of improving the weight of the Pauli measurements to be performed in a  $t$ -qubit PBC. Numerical results suggest that this heuristic algorithm achieves important reductions that, as explained above, have direct consequences on the practicality of PBC, but also on the depth and gate counts of the PBC-compiled circuits.

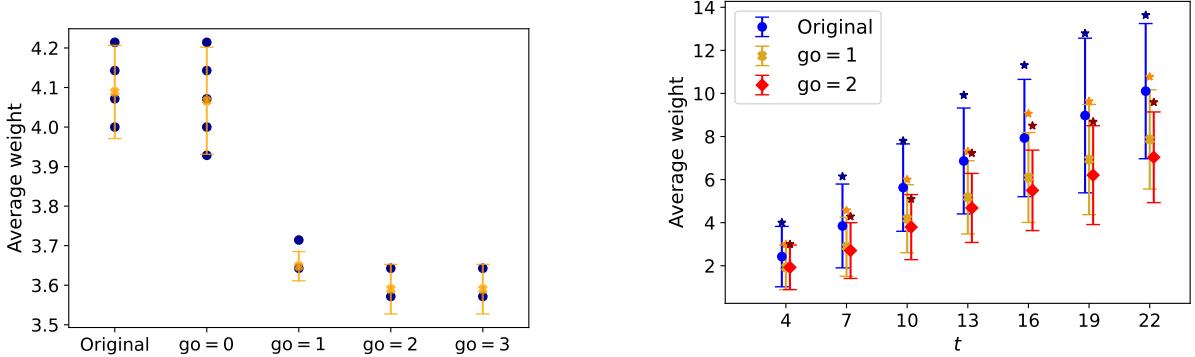
The algorithm works by recognizing that when the PBC procedure finds the new independent and pairwise-commuting Pauli operator to be measured,  $P_r$ , an equivalent evolution of the system can be performed by measuring any other Pauli operator that projects the system onto the same subspace. Put differently, any Pauli operator obtained by multiplying  $P_r$  by any combination of previously measured Pauli operators enacts an equivalent system transformation. The performance of the algorithm depends on the number of combinations used to try and achieve an improvement. For hidden-shift circuits, reductions of more than 10% and 12% to the average weight of the Pauli measurements can be achieved by searching through, respectively, a linear and quadratic number of alternative measurements at each step. On the other hand, for random quantum circuits, improvements of over 15% and 20% to the average weight are consistently achieved by the linear and quadratic orders of the algorithm for the largest instances tested. For smaller random circuits, these improvements exceed 20% and 30%, respectively for a linear and quadratic number of tested combinations. For a visual depiction of some of the numerical results see Fig. 1.

Since in noisy intermediate-scale quantum devices, two-qubit gates are prone to more errors, reducing the number of such gates is desirable as it often has a positive impact on the overall fidelity of the circuit. Considering that the mean average weight has a direct interpretation as the mean number of CNOT gates needed in the PBC-compiled quantum circuits, the improvements registered in this section are striking and should have important impacts in near-term quantum computing solutions.

**5. Concluding remarks.** We formulated three theorems that guarantee distinct specific non-trivial upper bounds for the average weight of the Pauli measurements in a PBC (a measure directly related to the number of CNOTs of the PBC-compiled quantum circuits) and/or the depth of the PBC (defined as the number of measurement layers). We complemented these formal results by providing numerical simulations of random quantum circuits. The results indicate that while the theorems promise “only” new upper bounds, the pre-compilation technique underlying them should have an impact that goes beyond that, leading to PBCs that have, in general, smaller average weights.

Our greedy algorithm further improves the natural resource savings achieved by the PBC model by providing substantial reductions to the average weight of the Pauli measurements. Importantly, this algorithm provides the option of “distributing the hardness” of the computation as one sees fit. That is, the overhead incurred by the greedy algorithm is entirely classical. Hence, one can push the classical machine by increasing the order of the greedy algorithm (i.e., the number of tested, valid combinations of Pauli operators) if one has as a priority reducing the demands on the quantum hardware. The suitable choice will depend on the (classical and quantum) resources available to the user.

The greedy algorithm inspires interesting new lines of research. The algorithm attempts to find the lowest weight Pauli operator at a given step. Alternatively, such a greedy algorithm can be coded with a different criterion in mind.



(a) Effect of the greedy algorithm on the possible average weight of the Pauli measurements in the PBCs obtained from hidden-shift circuits with  $t = 14$  and varying  $n = \{10, 14, 18, 22, 28\}$ . The results are independent of  $n$ . We register reductions of 0.7%, 10.8%, and 12.3% to the mean average weight (yellow crosses) respectively for a constant ( $go = 0$ ), linear ( $go = 1$ ), and quadratic ( $go = 2$ ) number of tested combinations. The dark blue dots represent the different possible average weights obtained for the PBCs that simulate the hidden-shift circuits under consideration and the error bars depict two standard deviations.

(b) Performance of the greedy algorithm for random quantum circuits with varying  $t = \{4, 7, 10, 13, 16, 19, 22\}$  for a linear (dark yellow crosses) and quadratic (red diamonds) number of tested combinations as compared to the results in the absence of a greedy algorithm (blue circles). The error bars depict two standard deviations from the mean average weight and the stars represent the maximum average weight. The impact of the different orders of the greedy algorithm is noticeable. In particular, for  $t > 4$ , the improvement to the mean average weight varies between 22.2% and 25.7% with  $go = 1$  and between 29.7% and 32.6% with  $go = 2$ .

FIG. 1. Performance of the greedy algorithm for (a) hidden-shift circuits and (b) the random quantum circuits.

For instance, in certain quantum hardware, a gate might exist that is noisier than the CNOT gate, so optimizing the PBC sequence to reduce, for instance, the number of Z operators might be more beneficial. One can also think of more sophisticated algorithms that, rather than trying to find the best solution at each step, try to optimize things globally. In doing this, the algorithm might avoid reducing the weight at one specific step to reap a better reward at later stages of the computation. This might be accomplished by an algorithm with a global (rather than local) reward system such as seen, for instance, in reinforcement learning algorithms.

- 
- [1] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, Quantum Circuit Simplification and Level Compaction, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **27**, 436 (2008).
- [2] R. Iten, R. Moyard, T. Metger, D. Sutter, and S. Woerner, Exact and Practical Pattern Matching for Quantum Circuit Optimization, *ACM Transactions on Quantum Computing* **3**, 1 (2022).
- [3] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus, *Quantum* **4**, 279 (2020).
- [4] A. Kissinger and J. van de Wetering, Reducing the number of non-Clifford gates in quantum circuits, *Phys. Rev. A* **102**, 022406 (2020).
- [5] M. Amy, D. Maslov, and M. Mosca, Polynomial-Time T-Depth Optimization of Clifford+T Circuits Via Matroid Partitioning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **33**, 1476 (2014).
- [6] L. E. Heyfron and E. T. Campbell, An efficient quantum compiler that reduces T count, *Quantum Science and Technology* **4**, 015004 (2018).
- [7] M. Amy and M. Mosca, T-Count Optimization and Reed-Muller Codes, *IEEE Transactions on Information Theory* **65**, 4771–4784 (2019).
- [8] S. Bravyi, G. Smith, and J. A. Smolin, Trading Classical and Quantum Computational Resources, *Phys. Rev. X* **6**, 021043 (2016).
- [9] F. C. R. Peres and E. F. Galvão, Quantum circuit compilation and hybrid computation using Pauli-based computation, *Quantum* **7**, 1126 (2023).
- [10] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [11] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, *Phys. Rev. A* **68**, 022312 (2003).

# Reducing the Weight of Pauli Measurements in Pauli-based Computation

Filipa C. R. Peres<sup>1, 2,\*</sup> and Ernesto F. Galvão<sup>1, 3</sup>

<sup>1</sup>*International Iberian Nanotechnology Laboratory (INL), Av. Mestre José Veiga, 4715-330 Braga, Portugal*

<sup>2</sup>*Departamento de Física e Astronomia, Faculdade de Ciências, Universidade do Porto, rua do Campo Alegre s/n, 4169-007 Porto, Portugal*

<sup>3</sup>*Instituto de Física, Universidade Federal Fluminense, Avenida General Milton Tavares de Souza s/n, Niterói, Rio de Janeiro 24210-340, Brazil*

(Dated: July 28, 2024)

Pauli-based computation (PBC) is a universal measurement-based quantum computation model steered by an adaptive sequence of independent and compatible Pauli measurements on separable magic-state qubits. Here, we propose several new ways of decreasing the weight of the Pauli measurements and their associated CNOT complexity; we also demonstrate how to reduce this model's computational depth. Inspired by known state-transfer methods, we introduce incPBC, a universal model for quantum computation requiring a larger number of (now incompatible) Pauli measurements of weight at most 2. For usual PBC, we prove new upper bounds on the required weights and computational depth, obtained via a pre-compilation step. We also propose a heuristic algorithm that can contribute reductions of over 30% to the average weight of Pauli measurements (and associated CNOT count) when simulating Clifford-dominated random quantum circuits with up to 22  $T$  gates and over 20% for instances with larger  $T$  counts.

## I. INTRODUCTION

Quantum computing stands at the forefront of technological innovation, promising unprecedented computational power and transformative capabilities. However, current hardware remains somewhat limited in its capabilities. Thus, the minimization of quantum resources promises to play a relevant role in current and near-term implementations. This has prompted extensive research into multiple quantum circuit optimization techniques such as pattern matching [1, 2], ZX-calculus [3, 4] and phase polynomials [5–7], and leading to important reductions in the number of operations and/or circuit depth.

Quantum circuits stand as the dominant framework underlying these works. In the quantum circuit model [8, 9], the computation requires three essential steps: (i) the preparation of an input state (typically, the  $|0\rangle^{\otimes n}$  state), (ii) its coherent unitary evolution via the sequential application of quantum gates drawn from a universal set, and (iii) the measurement of the final state in the computational basis. The fact that this is an intuitive framework with a classical analogue has likely contributed to the prevalence of quantum circuits in the field of quantum computing. Nevertheless, measurement-based models have emerged as intriguing alternatives offering unique insights and potential benefits.

In this paper, we explore Pauli-based computation (PBC) [10], a universal model for quantum computation driven by an adaptive sequence of at most  $n$  independent and compatible multi-qubit Pauli measurements performed on a set of  $n$  qubits initialized in a separable magic state. PBC remains a relatively unexplored model compared to other measurement-based schemes. Here,

we aim to gain a better understanding of this framework and improve its feasibility. Recent work [11] demonstrated that PBC can be useful for compiling some families of quantum circuits dominated by Clifford gates; more specifically, the overall gate counts and depth of many of these quantum circuit instances can be significantly reduced by transforming each of them into a PBC and then translating it into adaptive Clifford circuits with magic-state input. The latter are called PBC-compiled circuits [11]. Importantly, the overall depth and gate counts of the final circuits are intimately connected to the *weight* of the Pauli measurements in the corresponding PBC. For this reason, this is *the* measure to optimize within the PBC framework to enable not only a more practical (native) implementation of the model itself but also to enhance its efficacy as a circuit compilation tool.

By making use of a pre-compilation step, we obtain fundamental improvements to the overall complexity of the Pauli measurements in PBC. Namely, we derive new upper bounds for the weights of the Pauli operators and demonstrate that the adaptive structure of any PBC allows some of the measurements to be carried out concomitantly, establishing novel bounds for the computational depth of this model. Importantly, the best weight and depth upper bounds found do *not* hold simultaneously. To address this, we show that a weight-depth compromise is possible. We also perform numerical experiments that provide evidence that, other than establishing previously missing upper bounds, the pre-compilation step also reduces the average weight of the Pauli measurements.

Besides these theoretical contributions, we propose a novel greedy algorithm capable of further reducing the weight of the Pauli measurements. Numerical results suggest that this heuristic algorithm achieves important reductions that, as explained above, have direct consequences on the practicality of PBC, but also on the depth

---

\* filipa.peres@inl.int

and gate counts of the PBC-compiled circuits. The algorithm searches over alternative Pauli measurements that project the system onto the same desired eigenspace, selecting the one with the lowest weight. The algorithm's performance depends on the size of the search space. For hidden-shift circuits, reductions of more than 10% and 12% to the average weight of the Pauli measurements can be achieved by searching through a number of alternative measurements, respectively, linear and quadratic in the number of qubits of the PBC. On the other hand, for random quantum circuits, improvements of over 15% and 20% to the average weight are consistently achieved by the linear and quadratic orders of the algorithm for the largest instances tested. For smaller random quantum circuits, these improvements exceed 20% and 30%, respectively for a linear and quadratic number of tested combinations.

As a side result, we also ask if it is possible to perform universal computation with Pauli measurements of weight at most two. We answer this question in the affirmative by relaxing two of the defining properties of PBC – namely, that the number of measurements is at most the number of qubits and that these measurements are all pairwise commuting. We call this incompatible, constant-weight Pauli-based computation (incPBC) to distinguish it from the original PBC formulation.

This paper is organized as follows. Section II provides the necessary background for the understanding of this work. We start by recalling the Pauli and Clifford groups and presenting different notions of universality (Secs. II A and II B). In Section II C, we make a brief description of the magic-state injection model. Section II D presents a detailed review of measurement-based quantum computation (MBQC) focused on one-way quantum computing (1WQC), teleportation-based and state-transfer-based computation, and PBC. Section III introduces incPBC, demonstrates its universality, and investigates the quantum resources needed to perform a given computation within this framework. Sections IV and V contain our main results. In Sec. IV, we provide novel weight and depth upper bounds for PBC and give numerical evidence that the pre-compilation step also has practical benefits. Section V introduces our greedy algorithm which further improves the weights of the Pauli measurements. In Section VI, we summarize and discuss our main contributions, outlining interesting new lines of research.

## II. BACKGROUND

This work rests on many different concepts ranging from the stabilizer formalism and the Pauli and Clifford groups, to the magic-state injection model and several different measurement-based models. In this section, we make no pretension of giving a comprehensive description of all of these topics. Instead, we try to strike a balance between being comprehensive enough for an unfamiliar

reader to gain a sufficient understanding of all the concepts while avoiding the presentation of excessive details. With that in mind, we point the interested reader toward alternative (more in-depth) references where appropriate.

### A. Pauli and Clifford groups

Quantum circuits are the most widespread model for describing quantum computations. They work by first preparing an input state which, without loss of generality, can be the  $|0\rangle^{\otimes n}$  state. Then, a set of coherent unitary operations, known as quantum gates, are applied in an appropriate sequence,  $U$ , to the initial state, preparing the final state  $|\psi_f\rangle = U|0\rangle^{\otimes n}$ . The final readout is done via computational basis measurements on  $|\psi_f\rangle$ .

The Pauli and Clifford unitaries are operations that have a pivotal role in quantum computing in general and in our work in particular. An  $n$ -qubit Pauli operator is constructed by the  $n$ -fold tensor product of single-qubit Pauli operators ( $I$ ,  $X$ ,  $Y$ , and  $Z$ ) multiplied by one of the four possible phases  $\{\pm 1, \pm i\}$ . These operators form a group known as the  $n$ -qubit Pauli group, or simply, the Pauli group, denoted  $\mathcal{P}_n$ . We say that an  $n$ -qubit Pauli operator  $P$  has weight  $w \leq n$  if it involves  $w$  non-identity single-qubit Pauli operators. For instance, the Pauli operator  $P = X \otimes I \otimes Y \otimes Z$  is a Pauli operator of weight three. To simplify notation, we often omit the tensor product and identity from the description of multi-qubit Pauli operators and associate with each non-trivial single-qubit Pauli a subscript indicating the qubit it acts on. Using this convention, the operator in the previous example is written simply as  $P = X_1 Y_3 Z_4$ .

An  $n$ -qubit quantum state,  $|\phi\rangle$ , is said to be a stabilizer state if it is the simultaneous eigenvector of  $n$  independent and pairwise commuting Pauli operators with eigenvalue +1:  $G_i |\phi\rangle = |\phi\rangle, \forall i \in \{1, \dots, n\}$ . The operators  $G_i$  generate an abelian group called the stabilizer group:  $\mathcal{S} = \langle G_1, \dots, G_n \rangle$  which has  $|\mathcal{S}| = 2^n$  elements. Importantly, any stabilizer state can be uniquely defined by the set of generators,  $\{G_i\}_{i=1}^n$ , of its stabilizer group. Readers who are unfamiliar with the stabilizer formalism and are interested in learning more are referred to Refs. [8, Section 10.5.] and [12].

The Clifford group on  $n$  qubits,  $\mathcal{C}_n$ , is the normalizer of the Pauli group,  $\mathcal{C}_n := \{C \in U_{2^n} : C\mathcal{P}_n C^\dagger = \mathcal{P}_n\}$ , and is generated by the Hadamard ( $H$ ), phase ( $S$ ), and controlled-NOT ( $CX$ ) gates [13, 14]. The beautiful Gottesman-Knill theorem [13] states that any quantum circuit with only stabilizer state inputs, Clifford gates, and Pauli measurements is efficiently classically simulable. These circuits are known as stabilizer circuits. Since we expect quantum computers to be strictly more powerful than their classical counterparts, this result indicates that, although stabilizer circuits can be highly entangling, they are *not* universal for quantum computation.

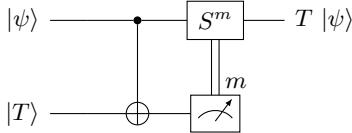


FIG. 1. Fault-tolerant implementation of the  $T$  gate via the well-known  $T$ -gadget, using only stabilizer operations and classical feedforward.

## B. Universality in quantum computation

Since we are interested in the ability to perform *any* quantum computation, it is important to discuss which operations enable universality. However, even before that, the notion of universality itself needs to be clarified.

In Ref. [15], the authors define a CQ-universal quantum computer as a device that, taking as input a classical bit string (say  $0^n$ ), is capable of preparing any quantum state  $|\psi_f\rangle$ . Put differently, given any unitary  $U$ , a CQ-universal quantum computer can prepare the state  $|\psi_f\rangle = U|0\rangle^{\otimes n}$ . This corresponds to the strong notion of universality called *strict* universality in Ref. [16]. For example, in the context of quantum computation with circuits, the Clifford unitaries supplemented by any non-stabilizer gate, i.e., any gate outside of the Clifford group, constitute a *strictly universal* gate set [17]. In this sense, alongside entanglement, non-stabilizerness (colloquially known as magic) is regarded as a necessary resource for enhanced computational power. A common choice for the additional gate is the  $T$  gate,  $T := \text{diag}(1, e^{i\pi/4})$ . Another important gate set known to be strictly universal is the set of all single-qubit rotations together with the CNOT gate [18].

A weaker notion of universality is CC-universality [15] (or *computational* universality [16]). Given any unitary operation  $U$ , a device is said to be CC-universal if it can reproduce the statistics of computational-basis measurements in any state  $U|0\rangle^{\otimes n}$ . In other words, a CC-universal quantum computer can reproduce the output probability distribution of any quantum circuit. Clearly, strict universality implies computational universality, i.e., a strictly universal gate set is also computationally universal, but the reverse is not necessarily true. An example of a computationally universal gate set that is not strictly universal is the set generated by the Toffoli and Hadamard gates [16, 19].

It is also important to comment on the difference between *exact* and *approximate* universality. Discrete gate sets (such as the Clifford+ $T$  and the Toffoli+ $H$  gate set) can only approximate arbitrary unitaries or output distributions up to a desired precision. In this sense, they are only approximately (strictly or computationally) universal. In contrast, the set comprised of the CNOT together with all single-qubit rotations is exactly (strictly) universal.

For other models of quantum computation, different sets of operations can be similarly used to enable strict or computational, and exact or approximate, universality. This work explores the transformation of Clifford+ $T$  quantum circuits into different measurement-based schemes. Hence, our use of the term *universality* always implies *approximate universality*.

## C. Magic-state injection

In the previous section, we have seen that (non-adaptive) Clifford+ $T$  quantum circuits are strictly universal for quantum computation. Here, we briefly describe another universal model of quantum computation known as the magic-state injection model [20].

In the context of fault-tolerant computation, error-correcting codes play a crucial role in protecting quantum information against errors. Stabilizer codes constitute a particular class of error-correcting codes wherein the encoding and decoding of information is accomplished using only Clifford gates. In these codes, Clifford gates can be easily implemented; specifically, to apply a Clifford gate  $C$  to an encoded qubit, one needs only apply that gate to all corresponding physical qubits. Gates that allow such a simple implementation are said to be transversal. Unfortunately, it is not possible for a universal gate set to have all its gates implemented transversally [21]. In the case of the Clifford+ $T$  gate set implemented using stabilizer codes, the  $T$  gate is non-transversal.

To circumvent the difficulty in fault-tolerantly implementing the  $T$  gate, Bravyi and Kitaev [20] proposed a way of producing low-noise copies of the magic state  $|T\rangle := (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$  from several noisier copies. Importantly, this magic-state distillation procedure uses only (fault-tolerant) Clifford operations. Once we have access to these low-noise  $|T\rangle$  states, we can implement any  $T$  gate by the so-called  $T$  gadget (Fig. 1), which uses only stabilizer operations and classical feedforward of measurement outcomes.

Using the  $T$  gadget, any universally general  $n$ -qubit Clifford+ $T$  quantum circuit with  $t$   $T$  gates can be transformed into an  $(n+t)$ -qubit adaptive Clifford circuit, that can be fault-tolerantly implemented using stabilizer codes. The price to pay for this is the need for the (offline) preparation of  $t$  copies of the  $|T\rangle$  state as well as feedforward and adaptivity.

We remark that, although magic-state injection is mostly considered in the context of fault-tolerant computing, in this paper that is not our concern. Instead, this model is briefly described here because it constitutes an important intermediate step when formulating PBC and proving its universality (see Sec. II D 3).

## D. Measurement-based quantum computation

Measurement-based quantum computation (MBQC) comes in an impressive variety of flavors. Here, we will try to center the discussion around the models that are more directly related to our work.

### 1. One-way quantum computing

In 2001, Raussendorf and Briegel proposed the one-way quantum computer [22] where the computation can be separated into two stages. First, the offline preparation of an entangled resource state known as the *cluster* state. Second, the processing stage wherein the qubits from the cluster state are measured in suitable bases to implement the desired computation. We will now focus in a bit more detail on each of these two stages.

Let  $\mathcal{G} = (V, E)$  be an undirected simple graph where  $V$  are its vertices and  $E$  its edges. For any such graph, a corresponding graph state  $|\mathcal{G}\rangle$  can be constructed as follows. First, a single qubit in state  $|+\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$  is assigned to each vertex  $i \in V$  of  $\mathcal{G}$ ; then, for each edge  $e \in E$  connecting two vertices  $i$  and  $j$ , the corresponding qubits are entangled using the (Clifford) gate  $CZ := \text{diag}(1, 1, 1, -1)$ .

Graph states are stabilizer states and therefore admit a simple representation using the stabilizer formalism. That is, an  $n$ -qubit graph state can be uniquely described by writing down the set of  $n$  generators of its stabilizer. These can be chosen to have the following form [23]: For each qubit  $i$ , a generator  $G_i$  is given by the tensor product of a Pauli operator  $X$  acting on the considered qubit and  $Z$  operators applied to each of its neighbors:

$$G_i = X_i \prod_{j \in \mathcal{N}(i)} Z_j, \quad (1)$$

where  $\mathcal{N}(i)$  denotes the set of qubits neighboring (i.e., connected to) qubit  $i$ .

The cluster state is no more than a particular kind of graph state where the underlying graph consists of a two-dimensional square grid. After preparing this state, following the prescription described above, the processing step can be carried out according to whichever algorithm we want to implement.

Raussendorf and Briegel showed that single-qubit rotations and the CNOT gate can be deterministically implemented in the cluster state by performing suitable sequences of single-qubit measurements in the equator of the Bloch sphere, together with computational basis measurements [22]. Since these gates constitute a strictly universal set, this endows the one-way computer with strictly universal capabilities.

Since the seminal work of Ref. [22], other combinations of resource states and sets of single-qubit measurements have been shown to be strictly universal [24–27]. Crucially, it is not necessary to allow a continuous range of

measurement bases and discrete sets suffice for (approximate) strict universality [25, 27]. The interested reader is pointed to Ref. [28, Table 1] for a quick overview of the different resource states and measurement bases that can be used for universal quantum computation.

One of the greatest advantages of 1WQC is that all multi-qubit operations (i.e., the entangling  $CZ$  gates) can be done offline before the processing stage. Since these unitaries are often more prone to errors than single-qubit ones, their isolation in the state preparation stage helps to mitigate the nefarious effects of those errors. Furthermore, if we have a device capable of preparing a universal resource state (of the proper size), we can perform any quantum computation by using the appropriate sequence of measurements in the processing stage. However, the production of a large quantum resource is not without its challenges. For a circuit with  $n$  qubits and logical depth  $d_L$ , the corresponding cluster state needs to have size  $O(nd_L)$ , which is extremely demanding for near- and intermediate-scale quantum hardware.

One way of circumventing this problem is to explore a peculiar feature of 1WQC: Clifford unitaries can be implemented via non-adaptive measurement patterns consisting exclusively of Pauli measurements. Because of this, all Clifford operations can be performed at once at the very beginning of the computation, regardless of their placement in the corresponding quantum circuit. Thus, 1WQC provides an intrinsically quantum way of parallelizing quantum computations, cutting across the strict temporal ordering of the quantum circuit model. If we consider an  $n$ -qubit Clifford+ $T$  quantum circuit with  $t$  non-Clifford  $T$  gates, carrying out all the Pauli measurements in the corresponding cluster state will leave us with a computation-specific graph state containing only  $t$  qubits that need to be measured in rotated angles  $\pm\pi/4$  along the equator of the Bloch sphere [25, 27] and  $n$  output qubits which will hold the final state output by the computation.

Removing all Pauli measurements and determining the computation-specific graph can be performed efficiently on a classical computer [29]. Hence, rather than needing an  $O(nd)$  (entirely general) cluster state, we require only an  $(n+t)$ -qubit computation-specific graph state. The problem with this approach is that, while we substantially save on the number of qubits needed, the graph state to be prepared can have a significantly more intricate connectivity structure which might be (more) challenging to prepare on actual quantum hardware.

**Remark 1** (Removing output qubits). Because our work focuses mostly on improving PBC and since this model of quantum computation is only computationally universal (see Sec. IID 3 and Remark 2 therein), we can content ourselves with having only a (weaker) CC-universal one-way quantum computer. That is, we are concerned only with simulating the output statistics of a certain quantum circuit. This means that further simplification to the computation-specific graph state is possible. Since the output qubits are measured in Pauli bases, their measure-

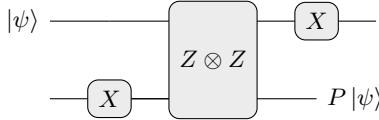


FIG. 2. Implementation of state transfer via single- and two-qubit Pauli measurements, up to a Pauli correction,  $P$ , that depends on all three measurement outcomes. Grey boxes with rounded edges are used throughout to represent projective measurements; the outcome of each of these measurements is stored in memory and accessible for future use (if needed).

ments can be classically processed together with the measurements associated with the Clifford gates leading to a computation-specific graph state that has only  $t$  qubits, which need to be measured in rotated angles  $\pm\pi/4$  along the equator of the Bloch sphere.

Recently, some results have started to arise concerning the realization of 1WQC using hypergraph states [28, 30, 31]. These are not stabilizer states; in fact, they possess both entanglement and non-stabilizerness. It has been demonstrated that hypergraph states together with single-qubit Pauli measurements are sufficient for strict universality [31].

## 2. Teleportation and state transfer

While 1WQC is considered a measurement-based model because its processing stage consists solely of single-qubit measurements, it still requires the coherent preparation of the resource state. Alternatively, in 2003, Nielsen proposed a universal scheme that requires no coherent unitary dynamics [32]; instead, it relies only on the preparation of qubits in the  $|0\rangle$  state and projective measurements of up to four qubits. Later, this approach to quantum computing was improved by Leung [33] in several ways. Notably, she removed the need for the recursive procedure to deal with the Pauli corrections while showing that a discrete set of at most two-qubit measurements suffices to guarantee universality.

These two schemes share as an underlying primitive the use of teleportation to implement universal gate sets [34]. By using state transfer (Fig. 2), Perdrix managed to propose an even simpler scheme, where a single two-qubit measurement is sufficient to realize a universal gate set and, thus, universal quantum computation [35].

Here, we will succinctly describe the scheme by Perdrix, as this will be directly useful to our work. Since the gate set generated by Hadamard,  $T$  gate, and CNOT is strictly universal for quantum computing, realizing each of these gates via state transfer is sufficient to prove the universality of the model. Figures 3 and 4 depict how one can apply these unitary transformations (up to a Pauli operator) to an arbitrary quantum state  $|\psi\rangle$ . Because the implementation is non-deterministic, in that

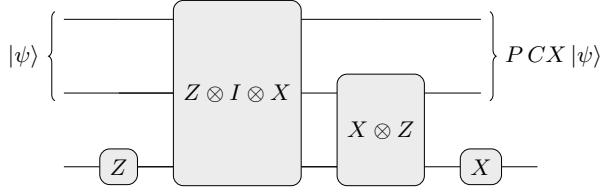


FIG. 3. Implementation of the CNOT gate using state transfer; the gate is applied to the (arbitrary) two-qubit input state  $|\psi\rangle$  up to a two-qubit Pauli operator  $P$  which depends on the outcomes of the measurements. Note that the second measurement involves only the first and last qubits as it has an identity on the second qubit, that is, it consists of the measurement  $Z_1 X_3$  (having only weight two).

the unitary is implemented up to a correction that depends on the outcomes of the measurements, we need to understand how these corrections can be dealt with. Note that standard state transfer, as depicted in Fig. 2, can be seen as an attempt to implement an identity transformation which will be done, in this scheme, up to a Pauli operation. This is the way to handle the Pauli corrections: Whenever a Pauli operator  $P$  arises from implementing a certain unitary, we perform regular state transfer until the measurement outcomes combine in such a way that any extra Pauli factors have been canceled out. This resembles the recursive procedure in Nielsen's scheme [32]. On average, four iterations suffice to correct for one single-qubit Pauli operator  $P$ ; thus, on average, this recursive procedure incurs an overhead that is linear in the total number of gates in the quantum circuit.

Finally, we remark that although at first 1WQC and these teleportation and state-transfer schemes were considered separately, several independent works [36–38] established important connections between them, thus succeeding in presenting a unified view of MBQC.

## 3. Pauli-based computation

Compared to other measurement-based models, PBC had a comparatively late appearance, being proposed in 2016 in the seminal paper by Bravyi, Smith, and Smolin [10]. Possibly owing to this, but also due to practical implementation challenges, PBC remains a relatively understudied model of quantum computation. Here, we provide a brief review of this scheme. For more detailed discussions, the interested reader is referred to Refs. [10, 11, 39]; additionally, Ref. [40, Sec. II A] provides a state-of-the-art review of recent works exploring PBC in different contexts.

In a PBC, a separable non-stabilizer input state is prepared offline; the computation is then driven by an adaptive sequence of independent and pairwise commuting multi-qubit Pauli measurements performed on the qubits of the input state. In Ref. [10], the authors showed that any (universally general) Clifford+ $T$  quantum cir-

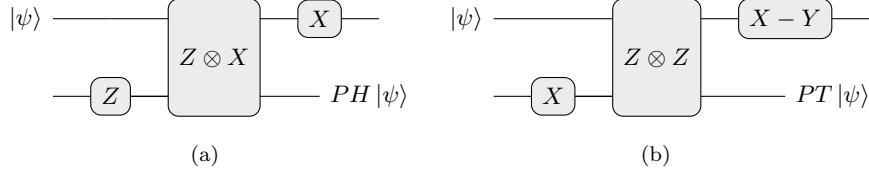


FIG. 4. Implementation of (a) the Hadamard gate and (b) the  $T$  gate using state transfer. Note that the unitary transformations are applied to the (arbitrary) input state  $|\psi\rangle$  up to a single-qubit Pauli operator  $P$  which depends on the measurement outcomes.

cuit with  $n$  qubits and  $t$   $T$  gates can be simulated by a PBC on  $t$  qubits and at most  $t$   $t$ -qubit Pauli measurements. The ability to simulate any quantum circuit makes PBC a universal model for quantum computation.

The proof of universality goes as follows. First, take the (non-adaptive)  $n$ -qubit Clifford+ $T$  circuit with  $t$   $T$  gates and transform it into the magic-state injection model (Sec. II C); this means we are left with an adaptive Clifford circuit with input  $|0\rangle^{\otimes n}|T\rangle^{\otimes t}$ . Note that the stabilizer register is stabilized by  $\mathcal{S} = \{Z_1, Z_2, \dots, Z_n\}$ . Since all operations are now Clifford unitaries, we can efficiently back-propagate every measurement to the beginning of the circuit [12]. Due to the adaptive nature of the circuit, the measurements need to be dealt with in the appropriate order. The intermediate measurements stemming from the gadgets need to be handled first (starting with the one from the first gadget and working our way successively until the last) and only afterward can the final readout measurements be pushed to the beginning of the circuit.

Once a Pauli measurement,  $P$ , arrives at the beginning of the circuit it may fall into one of three categories:

- (i)  $P$  anti-commutes with (at least) one of the operators,  $Q$ , in  $\mathcal{S}$ .—In this case, the outcome of the corresponding Pauli measurement,  $\sigma_P \in \{0, 1\}$ , can be decided classically using a coin toss;  $P$  is then dropped from the quantum circuit and replaced by the Clifford unitary:

$$V(\sigma_P, \sigma_Q) = \frac{(-1)^{\sigma_P} P + (-1)^{\sigma_Q} Q}{\sqrt{2}}, \quad (2)$$

where  $\sigma_Q$  denotes the outcome associated with the (prior) measurement of  $Q$ . The outcome  $\sigma_P$  is stored in a list containing all the outcomes.

- (ii)  $P$  commutes with all operators in  $\mathcal{S}$  and depends on a subset of them.—In this case, the Pauli measurement can again be dealt with classically, as its outcome,  $\sigma_P$ , can be efficiently inferred from the outcomes of the Pauli operators it depends on. The outcome obtained is stored in the list with all the other outcomes.
- (iii)  $P$  commutes with all operators in  $\mathcal{S}$  and is independent of them.—This is when  $P$  needs to be measured in the quantum computer. Note that case (i)

ensures that any such  $P$  must act trivially on the stabilizer register so that we can measure only its  $t$ -qubit non-stabilizer-register component. The Pauli operator  $P$  is then added to  $\mathcal{S}$  and its outcome is saved in the list with all the outcomes.

The procedure described above ensures that the original quantum circuit is simulated by an adaptive sequence of independent and compatible Pauli measurements performed only on the  $t$ -qubit non-stabilizer register, i.e., the circuit is simulated by a PBC. Since there are at most  $t$  independent and pairwise commuting Pauli operators, the number of measurements will be at most  $t$ . Potentially, these operators could all be of weight  $t$ , although our numerical results in Ref. [11] indicate that often the weight is lower than this trivial upper bound. In Section IV B, we will prove that better weights can be achieved by carrying out a pre-compilation step before proceeding to the PBC framework. More specifically, we take the input quantum circuit and transform it into a 1WQC procedure with an associated  $t$ -qubit computation-specific graph state and adaptive measurement pattern, and use the latter as the starting point for performing PBC.

**Remark 2** (Universality of PBC). The attentive reader will note that PBC simulates the original quantum circuit by producing samples drawn from the same output distribution. Therefore, Pauli-based quantum computers are devoid of strict universality and are instead computationally universal. This is in striking contrast to the other measurement-based models presented thus far which can prepare the same output state as the corresponding quantum circuit. Put differently, PBC does not concern itself with state preparation, but rather with simulating the output statistics of the corresponding quantum circuit.

### III. INCOMPATIBLE AND CONSTANT-WEIGHT PAULI-BASED COMPUTATION

In the previous section, we described PBC and noted that its computational steps are (at most)  $t$  independent and pairwise commuting Pauli measurements that can potentially involve all of the  $t$  qubits of the system. On the other hand, we have also seen that constant-weight projective measurements are sufficient for univer-

Model	Resources					
	Quantum memory		Measurement type	Number of measurements	Weight	Depth
1WQC	$t$		Compatible non-stabilizer measurements	$t$	1	$d_{1W}$
PBC	$t$		Compatible Pauli measurements	$\leq t$	$\leq t$	$\leq t$
incPBC	$2n$		Incompatible Pauli measurements	$w + 2t + 3c_2$	$\{1, 2\}$	$3d_L$

TABLE I. Comparison of the quantum resources needed across different measurement-based models of quantum computation for simulating a Clifford+ $T$  quantum circuit with  $n$  qubits,  $t$   $T$  gates,  $c_2$  CNOTs, and  $w$  readout computational basis measurements;  $d_{1W}$  denotes the number of layers in the adaptive measurement sequence of 1WQC which is often smaller than the logical depth  $d_L$  of the corresponding quantum circuit [41, 42]. The term “quantum memory” refers to the number of qubits that need to be online at any given point of the computation. As explained in the main text, qubit reinitialization and reuse are allowed in the case of incPBC.

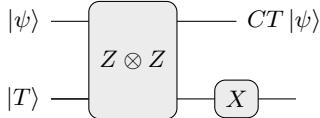


FIG. 5. Implementation of the  $T$  gate using only Pauli measurements [10]. We note that, unlike what happens in Ref. [35], the  $T$  gate is implemented up to a *Clifford* unitary assuming the form:  $I$ ,  $Z$ ,  $S$ , or  $ZS$  depending on the measurement outcomes.

sal quantum computation [32, 33, 35]. Therefore, we may ask ourselves: Can we engender a universal model for quantum computation that requires only constant-weight Pauli measurements? This question is answered here in the affirmative. We dub the resulting model incompatible, constant-weight Pauli-based computation (incPBC).

We use Perdrix’s scheme [35] as a starting point and modify it so that it requires only Pauli measurements. To that end, we relax the condition that the computation needs to be performed using measurements only and, as in PBC, allow the preparation of magic states,  $|T\rangle$ .

We note that the Hadamard and CNOT gates, implemented in Perdrix’s scheme as depicted in Fig. 4a and Fig. 3, respectively, require only Pauli measurements and are, therefore, already in the form we are interested in. Universality further requires a construction for implementing the  $T$  gate which avoids the non-stabilizer measurement  $(X - Y)/\sqrt{2}$  used in Ref. [35] (Fig. 4b) or any other such non-Pauli measurements. This can be done provided we have access to copies of the magic state  $|T\rangle$ . In that case, the  $T$  gate can be implemented deterministically, up to a *Clifford* unitary, using the construction in Fig. 5 [10].

An important detail is that half of the time, the  $T$  gate is implemented up to a unitary that involves the  $S$  gate. Thus, we also need to find a construction that deterministically carries out this unitary. It is not hard to show that the implementation in Fig. 6 performs the phase gate up to a Pauli operator  $P$  that depends on the measurement outcomes. Finally, the Pauli corrections can be dealt with similarly to Perdrix’s scheme.

Suppose we have an  $n$ -qubit Clifford+ $T$  quantum circuit with  $t$   $T$  gates,  $c_1$  phase and Hadamard gates,  $c_2$

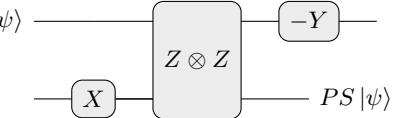


FIG. 6. Implementation of the  $S$  gate, up to a Pauli correction  $P$ , using only Pauli measurements.  $P$  depends on the measurement outcomes.

CNOTs and  $w$  final readout measurements. What are the quantum resources needed to implement the same computation using the model defined in the previous paragraphs? The answer is simple. Since the state-transfer-based implementation of each unitary requires exactly one auxiliary qubit, in each computational layer we may need up to  $n$  auxiliary qubits. Thus, we will need a quantum processor possessing at least  $2n$  qubits. The measurements involved are all Pauli measurements of weight either 1 or 2. Assuming the highly unlikely scenario wherein the measurement outcomes are such that no corrections are needed, a total of  $(w + 2t + 4c_2 + 3c_1)$  measurements need to be performed to carry out the desired computation in this model. In practice though, on average, an additional overhead linear in the total number of gates will be needed to deal with the tiresome Pauli and Clifford corrections that may arise from state transfer.

Fortunately, we can do better. Since PBC is only computationally universal, we will similarly allow this scheme to comply only with this weaker form of universality. This has immediate consequences in simplifying the framework. Notably, we do not need to worry about any Pauli corrections that may arise, as they can be pushed past subsequent Pauli measurements. We can also avoid the implementation of any single-qubit Clifford gates (i.e.,  $H$  and  $S$ ) via state-transfer constructions and similarly deal with these by propagating them until the end of the quantum circuit (past all Pauli measurements) with the consequence of changing the nature of the Pauli measurements they get pushed through, but never their weight. This is in the same spirit as PBC.

Next, we recognize that in the PBC procedure (recall Sec. IID 3), there are two mechanisms that may cause the Pauli measurements being back-propagated to the beginning of the circuit to spread (unboundedly) across the

qubits of the system. The first is the CNOT gates, which may increase (or decrease) the weight of Pauli operators by 1. The second is the Clifford unitaries  $V(\sigma_P, \sigma_Q)$  that need to be introduced into the quantum circuit whenever an anti-commuting Pauli is detected in (and removed from) the measurement sequence. The latter may increase (or decrease) the weight of Pauli measurements propagated across them by an unconstrained amount.

To ensure a model that relies only on constant-weight Pauli measurements we need to avoid both of these mechanisms. Avoiding the  $V(\sigma_P, \sigma_Q)$  unitaries means allowing measurement incompatibility in the model. Avoiding the increase of the weights promoted by propagation through CNOTs is done by implementing this gate using the state transfer construction by Perdrix [35], depicted in Fig. 3.

Therefore, the incPBC model we propose here works as follows. The CNOT gate and the  $T$  gate are implemented using the measurement-based constructions proposed, respectively, in Refs. [35] and [10] and depicted in Figs. 3 and 5. The gates are thus implemented deterministically up to Pauli or Clifford corrections that are irrelevant since they can be pushed through any Pauli measurements potentially changing their nature but without altering their weight. The Hadamard and  $S$  gates native to the circuit can similarly be pushed through any Pauli measurements that come from the implementation of the  $T$  and CNOT gates, again (possibly) changing them but never altering their weight.

To simulate the output distribution of an  $n$ -qubit Clifford+ $T$  quantum circuit with  $t$   $T$  gates,  $c_2$  CNOTs, logical depth  $d_L$ , and  $w$  readout measurements, incPBC requires (up to)  $2n$  qubits, the ability to prepare them either in the  $|0\rangle$  or the  $|T\rangle$  state, and a total of  $(w+2t+3c_2)$  Pauli measurements. We remark that not only have we managed to remove the dependence on the number of single-qubit Clifford gates in the quantum circuit but, additionally, by removing the need for the recursive (repeat-until-success) procedure in Perdrix's scheme, the number of measurements is no longer a best-case scenario requiring a convenient and highly unlikely sequence of measurement outcomes. Assuming that each layer of the circuit has at least one CNOT, the depth of this model—understood as the total number of measurement layers—is  $3d_L$ . In Table III, the resources needed in this scheme are compared to those needed to perform the same task within 1WQC and the original PBC. We note that the price to pay for having only constant-weight Pauli measurements is two-fold. First, measurement incompatibility needs to be allowed. Second, the number of measurements that need to be performed is much larger:  $(w+2t+3c_2)$  instead of  $t$ .

### A. Relation to other works

Other than the obvious tie to Perdrix's work, the astute reader may note a connection to the work by Bar-

tolucci *et al.* [43], which also makes use of measurements of constant weight. In that paper, the authors construct a scheme for fault-tolerant photonic quantum computation relying on the so-called fusions [44]; they suitably dub this model fusion-based quantum computation (FBQC).

One way to look at FBQC is as a practical architectural proposal of how to construct sufficiently large graph states to enable universal quantum computation. Ref. [43] notes that any graph state can be generated (up to local Clifford unitaries) provided one has access to 3-qubit Greenberger–Horne–Zeilinger (GHZ) states and is capable of doing Bell measurements. In the specific context of linear optics, the latter requirement corresponds to the ability to perform Bell fusions, also known as type-II fusions [44].

We will now spare some remarks on what makes incPBC and FBQC similar and what sets them apart. First of all, the description above highlights that FBQC can be regarded as a framework for universal quantum computation that makes use of few-qubit measurements. Specifically, Bell fusions consist of two-qubit measurements; additionally, since the scheme also requires access to 3-qubit GHZ states, and since multi-qubit states are not allowed as a resource in Perdrix's scheme or incPBC, a fair comparison requires understanding how such a resource could be generated by projective measurements. Since a GHZ state is stabilized by  $\mathcal{S}_{\text{GHZ}} = \langle XXX, ZZI, IZZ \rangle$ , we understand that in total FBQC requires at most measurements of weight 3. This contrasts with our scheme where measurements of at most weight 2 are needed.

FBQC is more general than incPBC in that the Bell fusions used to grow the resource state are assumed to be probabilistic. That is, the desired Bell measurement  $(XZ, ZX)$  is performed with probability  $1 - p_{\text{fail}}$  and it fails with probability  $p_{\text{fail}}$  carrying out the separable single-qubit measurements  $ZI$  and  $IZ$ . In contrast, our scheme assumes the measurements always succeed.

Finally, we note that, so far, all operations mentioned concerning FBQC are stabilizer measurements. Therefore, it remains to explain how universality can be achieved therein. The authors provide three different options: (i) applying modified fusion operations (akin to the works by Nielsen [32] and Leung [33]), (ii) making single-qubit non-stabilizer measurements (as in [35]), or (iii) replacing the resource state with a suitable magic resource state. The latter option is close in spirit to incPBC.

## IV. PRE-COMPILE AS A WAY TO IMPROVE PAULI-BASED COMPUTATION

In the previous section, we devised a computationally universal scheme that requires only Pauli measurements of weight 1 or 2. However, as we have seen, this construction steps away from that of PBC in two crucial ways: (i) it allows incompatible measurements and (ii) the number of measurements is much larger than that of qubits.

Improvement	Processing order	Formal statement
Strategy for better weights	$\mathcal{O}_1 : GM_1 \prec RO_1 \prec GM_2 \prec RO_2 \prec \dots \prec GM_t \prec RO_t$	Theorem 1
Strategy for better depth	$\mathcal{O}_2 : GMs \prec RO_{\ell_1} \prec RO_{\ell_2} \prec \dots \prec RO_{\ell_{d_{1W}}}$	Theorem 3
Weight-depth trade-off	$\mathcal{O}_3 : GM_{\ell_1} \prec RO_{\ell_1} \prec GM_{\ell_2} \prec RO_{\ell_2} \prec \dots \prec GM_{\ell_{d_{1W}}} \prec RO_{\ell_{d_{1W}}}$	Theorem 4

TABLE II. The table summarizes the main theoretical results of this section and highlights that they are obtained by choosing to back-propagate the Pauli measurements in different ways. Hence, the results in the three theorems do *not* hold simultaneously. As explained in the main text,  $GM_i$  ( $RO_i$ ) is used to denote the gadget (readout) measurement performed on the  $i$ th auxiliary (computational) qubit, while  $RO_{\ell_i}$  denotes the set of all readout measurements of computational qubits belonging to layer  $\ell_i$  and  $GM_{\ell_i}$  the set of all corresponding gadget measurements. The notation  $A \prec B$  is used to indicate that  $A$  precedes  $B$ .

The obvious follow-up questions are: Can we find alternative formulations of PBC with improved weights while retaining both measurement compatibility and a number of measurements smaller than or equal to the number of qubits? Are constant weights sufficient for PBC under those conditions? Here, we provide a partial answer to these questions. Additionally, in line with the overarching goal of improving the feasibility of PBC, we supply important new results on how to improve the depth of quantum computations carried out within this model.

### A. Overview of results and techniques

Throughout this entire section, we consider that we want to simulate a universal quantum circuit  $\mathcal{U}$  acting on  $n$  qubits and with gates drawn from the Clifford+ $T$  set. The circuit has logical depth  $d_L$ ,  $t$   $T$  gates, and  $w$  final readout computational-basis measurements. We have seen that, if we are concerned only with computational universality,  $\mathcal{U}$  can be simulated by:

- a one-way computation involving a  $t$ -qubit (computation-specific) graph-state,  $|\mathcal{G}\rangle$ , with stabilizers  $\{G_i\}_{i=1}^t$  described by Eq. (1) and single-qubit adaptive measurements along the  $\pm\pi/4$  directions in the equator of the Bloch sphere.
- a PBC requiring the separable input state  $|T\rangle^{\otimes t}$  and an adaptive sequence of at most  $t$   $t$ -qubit independent and compatible Pauli measurements.

Our approach is to use 1WQC as a stepping stone (or, put differently, as a pre-compilation step) before finding the corresponding PBC. We show that this can lead to improvements in the weights of the Pauli operators as well as in the logical depth of the computation.

Before delving into the details, we would like to punctuate a key observation underlying our three main theorems. If the starting point to the PBC procedure described in Sec. IID 3 is the quantum circuit  $\mathcal{U}$ , one processes a total of  $(t + w)$  measurements by first back-propagating the  $t$  gadget measurements and only afterward the  $w$  final measurements. On the other hand, by starting from the corresponding 1WQC procedure, we need to deal with a total of  $2t$  Pauli operators,  $t$  gadget measurements on auxiliary qubits, and  $t$  readout measurements performed on the corresponding com-

putational qubits (see Appendix A and Figs. 11 and 12 therein for details). Importantly, we have some freedom in the way in which we can process these measurements. One option is to take a processing order wherein we back-propagate first the measurement associated with the first gadget and immediately afterward the measurement of the corresponding computational qubit; we proceed in the same manner for the remaining qubits, always starting with the gadget measurement followed by the associated data qubit measurement. We can write this explicitly as

$$\mathcal{O}_1 : GM_1 \prec RO_1 \prec \dots \prec GM_t \prec RO_t , \quad (3)$$

where  $GM_i$  ( $RO_i$ ) is used to denote the gadget (readout) measurement performed on the  $i$ th auxiliary (computational) qubit and the notation  $A \prec B$  is used to indicate that  $A$  precedes  $B$ . This processing order gives rise to new upper bounds for the weight of the successive Pauli measurements, as stated in Theorem 1.

Alternatively, we can consider the measurement pattern underlying the one-way computation, where the computation is usually broken into layers with measurements in the same layer,  $\ell_i$ , being performed simultaneously and influencing the bases of measurements in subsequent layers  $\ell_j$  with  $j > i$ . This gives us the option to propagate all gadget measurements first, followed by the layered propagation of the measurements on the computational qubits:

$$\mathcal{O}_2 : GMs \prec RO_{\ell_1} \prec RO_{\ell_2} \prec \dots \prec RO_{\ell_{d_{1W}}} . \quad (4)$$

Here,  $GMs$  denotes all  $t$  gadgets measurements,  $RO_{\ell_i}$  the set of all readout measurements of computational qubits belonging to layer  $\ell_i$ , and  $d_{1W}$  the depth of the underlying one-way computation. This processing order is used to bound the overall depth of the PBC as defined in Theorem 3.

The fact that two different processing orders are used to arrive at Theorems 1 and 3 means that they do *not* hold simultaneously so we can either guarantee a reduced weight *or* a reduced depth, respectively. This led us to wonder whether a trade-off could be achieved to guarantee concurrent improvements of both of these parameters. This is the essence of Theorem 4 which is achieved through the layered propagation of both gadget and readout measurements so that:

$$\mathcal{O}_3 : GM_{\ell_1} \prec RO_{\ell_1} \prec \dots \prec GM_{\ell_{d_{1W}}} \prec RO_{\ell_{d_{1W}}} . \quad (5)$$

As before,  $\text{RO}_{\ell_i}$  denotes the set of all readout measurements of computational qubits belonging to layer  $\ell_i$ , while  $\text{GM}_{\ell_i}$  refers to the set of all corresponding gadget measurements.

Table II summarizes our main results and the associated processing orders. This is meant to provide a global view of our results and be used as a reference to guide the reader through the upcoming subsections. The fine details underlying both the discussion above and Table II are better understood by looking at the proofs of our main results (found in Appendices A and B).

## B. Improved weights

Because of the two mechanisms responsible for increasing the weight of Pauli measurements (described in Sec. III), when starting from quantum circuits it is hard to bound the weights of the  $t$  independent and pairwise commuting Pauli operators that need to be measured in the quantum computer. For this reason, the upper bound known thus far for the average weight of the measurements in a PBC,  $\bar{w}$ , consists only of the trivial one [10, 11]. Specifically, since each Pauli operator may have weight  $t$ , the average weight is upper-bounded by  $t$ :  $\bar{w} \leq t$ .

Here, we show that starting from 1WQC allows us to establish better (non-trivial) upper bounds for the weights of the Pauli measurements in the corresponding PBC.

**Theorem 1** (Improved weights). *Consider a one-way computation to be carried out on a  $t$ -qubit, computation-specific graph state  $|\mathcal{G}\rangle$  with a measurement pattern requiring only measurements along the  $\pm\pi/4$  directions on the equator of the Bloch sphere. By taking on the processing order  $\mathcal{O}_1$  defined in Eq. (3), the weights of the  $2t$  Pauli operators in the (complete) PBC procedure are upper-bounded by  $\{1, 1, 2, 2, \dots, t-1, t-1, t, t\}$ .*

We defer the proof of this result to Appendix A and use the remainder of the section to illustrate and discuss some of its practical consequences.

We know that the number of independent and pairwise commuting Pauli operators on  $t$  qubits is  $t$ . Therefore, of the list of  $2t$  Paulis, at most  $t$  of those correspond to measurements that will have to be performed in the quantum computer. Assuming the worst-case scenario where the Pauli operators to be measured are the ones with larger weight, i.e., the last  $t$  Paulis in the sequence, leads straightforwardly to the following Corollary.

**Corollary 2** (Average weight upper bound). *The average weight of the Pauli operators that need to be measured in the quantum hardware is upper bounded by  $\bar{w} \leq 3t/4 + 1/2$ .*

Corollary 2 is easy to prove, following straightforwardly from Theorem 1. It yields an improvement by

roughly 25% to the trivial upper bound that was previously known for PBC. However, the overall behavior is still linear in  $t$ . While PBC is improved by this construction, we are still far from a constant-weight scheme like the incPBC model we introduced in Sec. III. On the other hand, this improvement was possible while preserving both the compatibility condition and the total number of Pauli measurements.

We took two distinct families of random quantum circuits as testbeds for our results and their usefulness. The first family is the one we used in Ref. [11] wherein the random quantum circuits have a specific entangling structure implemented by sequences of eight different entanglement layers. (For details, see Sec. 4.2.2 of [11].) The second family is less structured in that gates were drawn from the set  $\{H, S, CX, T\}$  with probabilities  $\{(1-p)/3, (1-p)/3, (1-p)/3, p\}$ , without enforcing any specific pattern to the entangling gates. The probability,  $p$ , of drawing  $T$  was adjusted to facilitate creating quantum circuits with the desired  $T$ -count; the remaining gates were equiprobably drawn. Just like in Ref. [11], we consider circuits with 25 qubits and an adjustable number of  $T$  gates  $t = \{1, 4, 7, 10, 13, 16, 19, 22\}$ . For each family and each  $T$  count, we generated 500 different random circuits that were simulated by PBC with a total of 1024 shots. We used our code, companion to Ref. [11] and openly available at [45], selecting the dummy simulator option. This choice is strongly supported by our previous work [11].

Our results are depicted in Figs. 7a and 7b respectively for the first and second families. Given the large amount of data, we chose to use a boxplot [46]; the box extends from the first ( $q_1$ ) to the third ( $q_3$ ) quartile with the orange line inside identifying the median. The arms of the plot (extending from the box) have a length,  $l$ , of 1.5 times the interquartile range:  $l = 1.5(q_3 - q_1)$ ; any outliers are depicted as empty, black-lined circles. Red stars and yellow circles identify, respectively, the maximum and the mean average weight obtained for each  $t$ .

We note that the numerical results associated with the two distinct families are very similar. We also see that while all data points comply with the trivial upper bounds (as must be), several instances violate the worst-case average weight bound set by Corollary 2. In particular, we see this for both families at  $t$  equals 4, 7, and 10, but also at  $t = 16$  for the second family of quantum circuits. This happens because the code carries out PBC in the usual way, taking as a starting point the non-adaptive Clifford+ $T$  quantum circuits. On the other hand, the upper bound set by Corollary 2 is obtained assuming the intermediate step of transforming the circuit into a one-way computation and the subsequent suitable processing of the measurements as described by Theorem 1. Hence, the violation of this upper bound indicates that taking 1WQC as an intermediate step serves not only the theoretical purpose of finding better upper bounds but also provides practical advantages in achieving reduced weights. In this sense, 1WQC can be regarded as

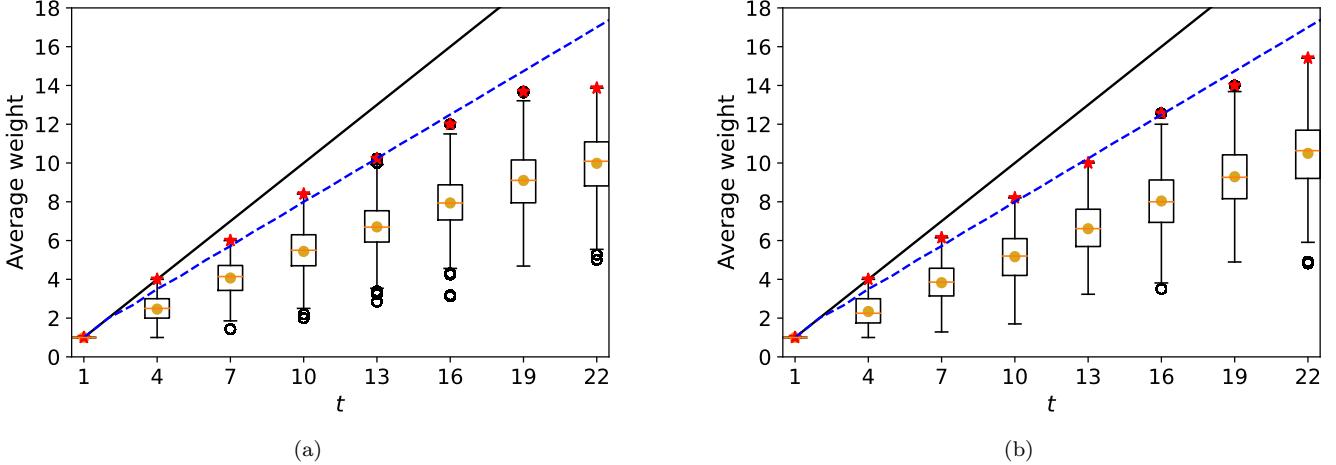


FIG. 7. Numerical data resulting from the compilation of two distinct families of random quantum circuits using our code written for Ref. [11] and openly available at [45]. (a) Circuits generated as in Ref. [11] with a well-defined and ordered entanglement structure, and (b) circuits with arbitrary entanglement structure wherein gates are drawn at random from the gate set  $\{H, S, CX, T\}$ . The numerical results are depicted using boxplots; the box extends from the first quartile ( $q_1$ ) to the third ( $q_3$ ) and its width,  $(q_3 - q_1)$ , is known as the interquartile range. The orange line inside the box represents the median; the arms extending below and above the box have a length given by 1.5 times the interquartile range and any outliers are depicted as empty black-lined circles. The red stars signal the maximal average weight obtained for a PBC for each  $t$ , while the yellow circles identify its mean. The solid black line denotes the trivial average weight upper bound of  $t$  known prior to this work and the dashed blue line gives the new upper bound for this quantity (as stated in Corollary 2).

a pre-compilation step that one might be interested in performing before running the actual PBC procedure.

### C. Improved parallelizability

Theorem 1 (or, more precisely, its proof) disregards the fact that the depth of a  $t$ -qubit one-way computation is often smaller than  $t$ . That is, some of the qubits can be measured simultaneously, constituting one computational layer; we denote by  $d_{1W}$  the depth (i.e. the number of layers) of the computation in 1WQC. One may now wonder: How does this influence the corresponding PBC procedure? Can Pauli measurements stemming from the same 1WQC computational layer be performed simultaneously as well? If we were to discover this to be the case, it could translate into important improvements to the overall computational depth.

Consider a PBC on  $t$  qubits involving the measurement of  $r \leq t$  (adaptively chosen) Pauli operators. Since the PBC procedure ensures that all of these operators are compatible, it is the adaptive structure of the PBC that determines whether some of these measurements may or may not be determined simultaneously. When starting from the quantum circuit model, it is hard to extract any information that allows us to assume a better depth than simply  $d = r$ . The adaptive structure of the 1WQC measurement pattern is determined solely by the underlying graph and its gflow (recall Sec. II D 1); in general, it ensures that the computation is broken down into layers of measurements that can be performed simultaneously.

When taking 1WQC as a starting point, we expect to mirror this property to PBC; that is, we anticipate that some of the measurements of the PBC can similarly be grouped together and performed concomitantly as they do not influence one another, constituting a single layer of the PBC computation. The fact that all measured Pauli operators need to be compatible is also a crucial factor in enabling measurement concurrency. To the total number of layers formed in this way, we call “*depth of the PBC*” and denote it  $d_{PBC}$ .

The main results of this section are the following two theorems.

**Theorem 3** (Improved depth). *By taking up the processing order  $O_2$  in Eq. (4), the depth of a PBC is upper-bounded by the logical depth of the corresponding one-way quantum computation,  $d_{1W}$ .*

This result illustrates how we can reduce the depth of PBC from  $t$  to  $d_{1W}$ . As is apparent from the proof of the theorem, this comes at the expense of losing the better weight upper bounds derived in the previous section. Hence, we wondered if an intermediate result could be provided so that the depth was still better than  $t$  while, at the same time, allowing for an upper bound on the weights better than the trivial one. Such a scenario is subsumed in Theorem 4. Recall that we denote by  $RO_{\ell_i}$  the set of all readout measurements of computational qubits belonging to layer  $\ell_i$  and  $GM_{\ell_i}$  the set of corresponding gadget measurements.

**Theorem 4** (Weight-depth trade-off). *Consider a one-way computation with logical depth  $d_{1W}$  and layering so*

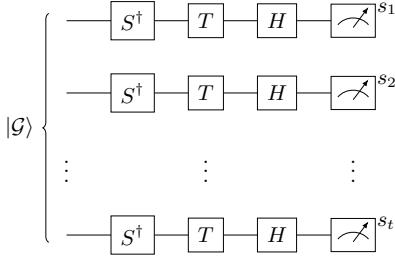


FIG. 8. A single-layer one-way computation depicted as a quantum circuit.

that the number of computational qubits in layer  $\ell_i$  is  $\kappa_i$ :  $\sum_{i=1}^{d_{1W}} \kappa_i = t$ . By back-propagating the measurements following  $\mathcal{O}_3$  in Eq. (5), the depth of the corresponding PBC is upper-bounded by  $\min\{2d_{1W}-1, t\}$ . Moreover, the weight of the  $2\kappa_i$  Pauli operators stemming from  $\text{GM}_{\ell_i}$  and  $\text{RO}_{\ell_i}$  measurements is upper bounded by  $\sum_{j=1}^i \kappa_j$ .

In this theorem, we see that there is still a successive increase of the upper bound of the weights as the computation moves along. However, while in Theorem 1 the weight upper bound would increase by one every other operator, now the upper bound evolves in steps of (variable) width  $\kappa_i$  increasing from one step to the other by an amount given by the number of qubits in the associated layer of the underlying one-way computation.

We now consider the simple case of quantum computations with depth 1. This is the only point of the main text where we go through an explicit demonstration. The reason for this is two-fold. First, it gives the reader a glimpse of the proof strategies used throughout this work via the simplest possible example. Secondly, this demonstration equips us with useful insights that are exploited in the more general proofs of Theorems 3 and 4, which are deferred to Appendix B.

**Lemma 5** (Single-layer computation). *A single-layer one-way computation on a computation-specific graph state  $|G\rangle$  with non-adaptive measurements performed along the  $\pi/4$  direction on the equator of the Bloch sphere has an associated single-layer PBC.*

*Proof.* The starting point for the proof is the quantum circuit in Fig. 8 which depicts the one-way computation in question. The first step consists of transforming the  $T$  gates into  $T$  gadgets.

Before delving into the fine details of this demonstration, we highlight the notation used both here and in the proofs of Theorems 1, 3, and 4. We choose to use different letters to denote the measurement outcomes of computational qubits ( $\{s_i\}_{i=1}^t$ ) and those of the auxiliary qubits introduced by the  $T$  gadget ( $\{m_i\}_{i=1}^t$ ). We will also differentiate the Pauli operators that stem from these two different types of measurements; Pauli operators resulting from measurements on computational qubits are denoted by  $P_i$  whilst for those originating from gadget

measurements we use  $Q_i$ . The generators of the stabilizer of the graph state,  $\mathcal{S}$ , are represented by  $G_i$  and constructed as prescribed in Eq. (1).

Since we are considering that we have  $d_{1W} = 1$ , we note that  $\mathcal{O}_2 \equiv \mathcal{O}_3 : \text{GM}_{\ell_1} \prec \text{RO}_{\ell_1}$ . Thus, we take all gadget measurements and process them first, before dealing with the readout measurements. In doing this, we note that each measurement stemming from the  $T$  gadget arrives at the beginning of the circuit  $\mathcal{C}$  as  $Q_i = Z_i Z_{t+i}$  which anti-commutes with  $G_i$ .

We know that, in general, the order in which these operators are processed is relevant. Here, we will show that, in this particular case, they can all be processed simultaneously. Take first  $Q_1 = Z_1 Z_{t+1}$ . Since it anti-commutes with  $G_1$  it is dealt with by drawing  $m_i$  uniformly at random from  $\{0, 1\}$  and then replacing  $Q_1$  with the Clifford unitary

$$V(G_1, Q_1, m_1) = \frac{G_1 + (-1)^{m_1} Q_1}{\sqrt{2}}.$$

To simplify notation, we will denote this and all upcoming  $V$  unitaries solely by  $V(m_i)$ ; the reader should keep in mind that they depend also on  $G_i$  and  $Q_i$ .

Next, comes  $Q_2$ . This operator arrives at the beginning of the quantum circuit as  $Q_2 = Z_2 Z_{t+2}$ . But, after processing  $Q_1$ , we now have the unitary  $V(m_1)$  through which  $Q_2$  also needs to be back-propagated. Crucially,  $[Q_2, G_1] = [Q_2, Q_1] = 0$  which means that it is pushed through  $V(m_1)$  without being altered. It is then processed much like  $Q_1$  by making a coin toss to decide  $m_2$  and introducing the corresponding Clifford unitary  $V(m_2)$ . In the same fashion, each subsequent  $Q_j$  is passed through previously added  $V(m_i)$ , with  $i < j$ , unitaries without being changed.

This means that all gadget measurements can be processed simultaneously in the classical machine by drawing a  $t$ -bit string  $\mathbf{m} \in \{0, 1\}^t$  uniformly at random and then adding the unitaries

$$V(m_i) = \frac{G_i + (-1)^{m_i} Q_i}{\sqrt{2}} \quad (6)$$

to the beginning of the circuit. For convenience, let us denote  $V = \prod_{i=1}^t V(m_i)$ .

Next, we need to handle the readout measurements. These can similarly all be processed simultaneously. They are back-propagated through the adaptive Clifford circuit  $\mathcal{C}$  and  $V$  in the following form:

$$Z_i \xrightarrow{\text{cv}} \begin{cases} \text{if } m_i = 0 : & P_i = R_i Y_{t+i} \\ \text{otherwise :} & P_i = G_i R_i X_{t+i} \end{cases}, \quad (7)$$

with  $R_i = (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} \left( \prod_{b \in \mathcal{N}(i)} G_b \right) \left( \prod_{c \in \mathcal{N}(i)} Z_{t+c} \right)$ . Since the calculations are somewhat extensive, we leave the details for Appendix C.

It is not hard to understand that, for any  $\mathbf{m}$ ,  $[P_i, P_j] = 0, \forall i, j$  so that all  $t$  operators are independent and compatible. Additionally, they are also compatible with the

stabilizers of the graph state. Thus, these operators are identified as Paulis that need to be measured in the quantum hardware. Importantly, they can be processed and, therefore, measured simultaneously comprising a PBC with a single layer. This concludes the proof. ■

Note that since the processing order is *not*  $\mathcal{O}_1$ , Theorem 1 is *not* guaranteed to hold. Thus, we managed to perform a single-layer computation at the expense of potentially larger Pauli weights.

Let us look more carefully into the weights of the measurements in this particular case of single-layer computation. From Eq. (7), it is easy to see that the weight of each Pauli operator  $P_i$  (in the magic register) is  $w_i = N_i + 1$ , where  $N_i$  denotes the number of neighbors of the  $i$ th qubit of the graph state. For a fully connected graph, this means that each operator has weight  $w_i = t$ . On the other hand, if the degree of each vertex of the graph is  $O(1)$ , each Pauli measurement has weight  $O(1)$  and the average weight is constant (rather than linear).

The work of Markov and Shi [47] shows that 1WQC can be classically simulated in time exponential in the treewidth of the underlying graph. This means that any graph having a treewidth logarithmic in its size will lead to a computation that can be efficiently classically simulated. The maximal degree of a graph and its treewidth are independent properties. However, if a graph has maximal degree 2 its treewidth can be at most two, leading to efficient classical simulation. Hence, no single-layer PBC with an average weight smaller or equal to 2 can lead to quantum advantage.

We conclude by noting that these observations also hold for the adaptive, multi-layered case of Theorem 3, i.e., when the processing order  $\mathcal{O}_2$  is used. For details see Appendix B.

## V. GREEDY ALGORITHM FOR SMALLER-WEIGHT MEASUREMENTS

In this section, we present a heuristic algorithm to improve the weight of the Pauli measurements to be performed in a  $t$ -qubit PBC. To understand the idea behind this algorithm consider a PBC such that a sequence of Pauli operators  $L_P = \{P_1, \dots, P_{r-1}\}$  have already been measured. The state of the system after such a sequence is given by

$$|\psi_{r-1}\rangle \propto \prod_{i=1}^{r-1} \frac{I^{\otimes t} + (-1)^{\sigma_i} P_i}{2} |T\rangle^{\otimes t},$$

where  $\sigma_i$  is the outcome of the corresponding Pauli measurement  $P_i$ ; these outcomes are stored in a list denoted  $L_\sigma$ . Continuing the PBC procedure, the ensuing Pauli measurement,  $P_r$ , is discovered. Performing this Pauli measurement would cause the state to evolve such that:

$$|\psi_r\rangle \propto \frac{I^{\otimes t} + (-1)^{\sigma_r} P_r}{2} |\psi_{r-1}\rangle.$$

---

### Algorithm 1: Greedy algorithm

---

```

Input :  $(L_P, L_\sigma, P_r, \text{go})$ 
Output:  $P'_r$ , the Pauli operator to be measured.

1  $r \leftarrow \text{Length}(L_P) + 1;$  /* Current time step */
2  $w \leftarrow \text{FindWeight}(P_r);$  /* Weight to beat */
3  $P'_r \leftarrow P_r;$ 
4 for  $a \leftarrow 0$  to go do
5   foreach  $\mathcal{W}$  in Combinations( $\{1, \dots, r-1\}$ ,  $a$ ) ∪
    Combinations( $\{1, \dots, r-1\}$ ,  $r-1-a$ ) do
6      $P_{\text{new}} \leftarrow P_r \prod_{j \in \mathcal{W}} (-1)^{L_\sigma[j]} L_P[j];$ 
7      $w_{\text{new}} \leftarrow \text{FindWeight}(P_{\text{new}});$ 
8     if  $w_{\text{new}} < w$  then
9        $P'_r \leftarrow P_{\text{new}};$ 
10       $w \leftarrow w_{\text{new}};$ 
11 return  $P'_r$ 

```

---

Since the quantum state  $|\psi_{r-1}\rangle$  is stabilized by  $\langle(-1)^{\sigma_1} P_1, \dots, (-1)^{\sigma_{r-1}} P_{r-1}\rangle$ , it is not hard to show that  $|\psi_r\rangle$  can equivalently be obtained if, rather than measuring  $P_r$ , one decides to measure any Pauli operator of the form  $P_r \prod_{j \in \mathcal{W}} (-1)^{\sigma_j} P_j$ . Here,  $\mathcal{W}$  denotes one of the  $2^{r-1} - 1$  possible subsets of  $\{1, \dots, r-1\}$  (excluding the empty set  $\emptyset$ , which corresponds to considering  $P_r$  itself). This implies that, whenever we find a Pauli operator  $P_r$  at the  $r$ th time step, we can measure exactly that Pauli or any one of the  $2^{r-1} - 1$  Pauli operators that perform an equivalent state transformation. Thus, we are free to choose whichever operator has the lowest weight.

The discussion above shows that, for the last time step  $t$ , the total number of equivalent Pauli measurements is  $2^{t-1}$ . Thus, if we try to analyze all possibilities, we will incur an exponential classical processing overhead of  $O(2^t)$ . To preserve the efficiency of the PBC procedure, the search for smaller-weight Pauli operators should be restricted to a polynomial-sized subset of equivalent Pauli operators. Put differently, the number of subsets  $\{\mathcal{W}_i\}_{i=1}^N$  considered at each time step must be such that  $N = O(\text{poly}(t))$ . An option that incurs only a constant overhead to the processing of each Pauli operator (and, therefore, an overhead of  $O(t)$  to the overall procedure) is to consider, at each time step  $r$ , only one set:  $\mathcal{W} = \{1, \dots, r-1\}$ . This means that all one needs to do is compare the weight of the Pauli operator  $P_r$  with that of  $P_r \prod_{j=1}^{r-1} (-1)^{\sigma_j} P_j$  and choose the one with the smallest weight.

Alternatively, if we restrict ourselves to subsets  $\mathcal{W}$  of size either 1 or  $r-2$ , the total number of possible subsets is  $2r, \forall r \geq 4$ . This incurs a linear overhead to the processing of each Pauli operator since we need to compare the weight of  $P_r$  to that of a total of  $O(r)$  other Pauli operators. In total, this leads to a contribution of  $O(t^2)$  to the complete PBC procedure. If we allow the subsets  $\mathcal{W}$  to have size 2 or  $r-3$ , then the total number of equivalent Pauli measurements to be tested against  $P_r$  is  $O(r^2)$ , i.e., we incur a quadratic overhead in the processing of

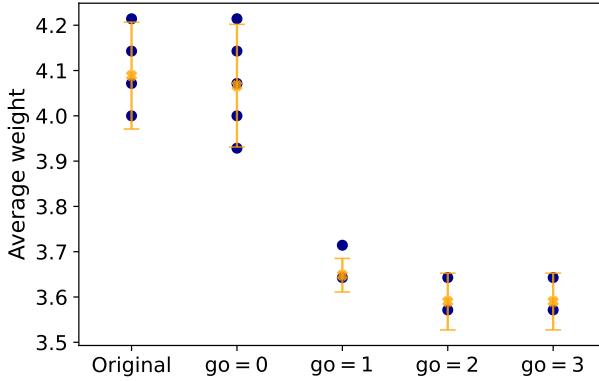


FIG. 9. Effect of the greedy algorithm on the possible average weight of the Pauli measurements in the PBCs obtained from hidden-shift circuits with  $t = 14$  and varying  $n = \{10, 14, 18, 22, 28\}$ . The results are independent of  $n$ . We register reductions of 0.7%, 10.8%, and 12.3% to the mean average weight (yellow crosses) respectively for  $\text{go} = 0, 1$ , and 2. The dark blue dots represent the different possible average weights obtained for the PBCs that simulate the hidden-shift circuits under consideration and the error bars depict two standard deviations.

each Pauli, leading to a total overhead of  $O(t^3)$  to the complete PBC procedure.

These ideas allow us to construct a greedy algorithm that, at each time step  $r$ , given the Pauli operator to be measured,  $P_r$ , searches for alternative, equivalent Pauli measurement with a smaller weight. The efficiency of the algorithm is controlled by a parameter that we call “greedy order”, denoted  $\text{go} \geq 0$ . Once we set a specific value for  $\text{go}$ , the algorithm searches for Pauli measurements with better weight among the (sub)sets  $\mathcal{W} \subseteq \{1, \dots, r-1\}$  with size  $r-1-a$  and  $a$ , with  $0 \leq a \leq \text{go} \leq (r-1)/2$ . This means that the algorithm incurs a time overhead given by:

$$\tau_{\text{greedy}} = \sum_{r=1}^t \left( 2 \sum_{a=0}^{\text{go}} \frac{(r-1)!}{(r-1-a)! a!} - 1 \right).$$

Thus,  $\text{go} = 0$  implies a linear overhead to the entire PBC procedure,  $\text{go} = 1$  a quadratic overhead,  $\text{go} = 2$  a cubic one, and so on.

Pseudocode for this greedy algorithm is outlined in Algorithm 1. We included this algorithm in the code found at [45] and used it to perform PBC compilation on the same quantum circuits studied in our previous work [11] and described in Sec. IV B.

For the hidden-shift circuits, the effect of the greedy algorithm is represented in Fig. 9. The results are independent of the number of qubits,  $n$ , in the original circuit. We see that the original circuits were compiled to a total of 4 different possible average weights (equivalently, a total of 4 possible CNOT counts). Applying the greedy algorithm with  $\text{go} = 0$  has only a small effect of adding a fifth possible average weight lower than the

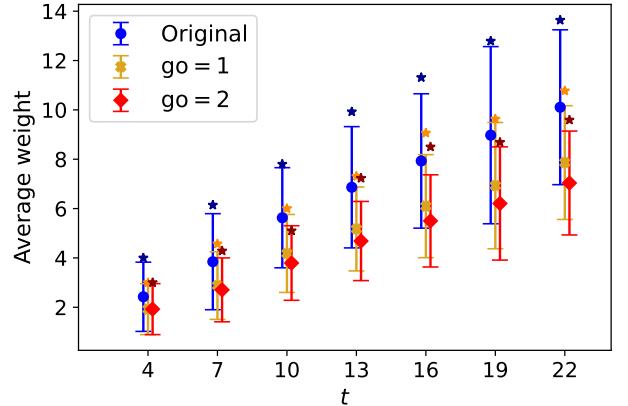


FIG. 10. Performance of the greedy algorithm for random quantum circuits with varying  $t = \{4, 7, 10, 13, 16, 19, 22\}$  for  $\text{go} = 1$  (dark yellow crosses) and  $\text{go} = 2$  (red diamonds) as compared to the results in the absence of a greedy algorithm (blue circles). The error bars depict two standard deviations from the mean average weight and the stars represent the maximum average weight. The impact of the different orders of the greedy algorithm is noticeable. In particular, for  $t = 4$ ,  $\text{go} = 1$  and  $\text{go} = 2$  yield the same performance, corresponding to an improvement of roughly 20.6% to the mean average weight. For  $t > 4$ , the improvement to the mean average weight varies between 22.2% and 25.7% with  $\text{go} = 1$  and between 29.7% and 32.6% with  $\text{go} = 2$ .

original four, reducing the mean average weight only by  $\sim 0.7\%$ . With  $\text{go} = 1$  the impact of the greedy algorithm is more expressive. In particular, the considered hidden-shift circuits lead to PBCs with one out of two possible distinct average weights, both of which are lower than the possible average weights of the PBCs obtained without the greedy algorithm or with  $\text{go} = 0$ . A similar effect is seen for  $\text{go} = 2$ , where only two possible distinct average weights occur, the largest of which corresponds to the smallest value obtained with  $\text{go} = 1$ . Focusing on the mean average weight associated with these results,  $\text{go} = 1$  and  $\text{go} = 2$  lead to improvements of 10.8% and 12.3%, respectively. Importantly, the reader should recall that any improvements registered to the average weight of PBC correspond directly to the reduction in the number of CNOT gates of the PBC-compiled circuits. For these (small) circuits, running  $\text{go} = 3$  leads to similar results as obtained with  $\text{go} = 2$  suggesting that the performance of the algorithm might stagnate after that.

One may wonder how the performance of the greedy algorithm changes for larger circuits. To understand this, we picked the same 50 larger hidden-shift circuits with  $n = t = 42$  as in our previous work [11]. In this case, the effect of  $\text{go} = 0$  is negligible, while  $\text{go} = 1$  and  $\text{go} = 2$  achieve improvements to the average weight of 13.7% and 16.7%, respectively. Interestingly, the performance of our algorithm seems to improve slightly for these larger circuits.

The results for the random quantum circuits are de-

picted in Fig. 10. To make this figure more readable, we omitted the results for  $go = 0$  even though the improvements associated with this order of the greedy algorithm are more appreciable than for the hidden-shift circuits, varying between 6.0% and 12.2%. For  $t = 4$ ,  $go = 1$  and  $go = 2$  yield the same performance, corresponding to an improvement of roughly 20.6% to the mean average weight. For  $t > 4$ , the improvement to the mean average weight varies between 22.2% and 25.7% with  $go = 1$  and between 29.7% and 32.6% with  $go = 2$ .

As before, it is interesting to understand how the algorithm performs for larger instances. To that end, we investigated 50 random quantum circuits with  $n = t = 49$ . For these circuits, the mean average weight was reduced by 5%, 18.2%, and 25.4%, respectively for  $go = 0$ ,  $go = 1$ , and  $go = 2$ . We see that contrary to what was observed in the case of larger hidden-shift circuits, the performance of the greedy algorithm seems to worsen for the larger random quantum circuits.

We leave further analyses of the performance of the greedy algorithm to Appendix D. We also discuss a variation of the algorithm and compare its performance to that of Algorithm 1.

Since in noisy intermediate-scale quantum devices, two-qubit gates are prone to more errors, reducing the number of such gates is desirable as it often has a positive impact on the overall fidelity of the circuit. Considering that the mean average weight has a direct interpretation as the mean number of CNOT gates needed in the PBC-compiled quantum circuits, the improvements registered in this section are striking and should have important impacts in near-term quantum computing solutions.

## VI. DISCUSSION AND OUTLOOK

PBC is substantially different from other methods existing in the literature for saving quantum resources. Compilation techniques often apply to non-adaptive quantum circuits and strategies for resource optimization for adaptive quantum computations are somewhat lacking. PBC addresses this on its own and the work presented here drives this even further, presenting a wide set of results that collectively improve the practical feasibility of this computational model.

In Sec. III, we defined a new version of PBC which we dubbed incPBC that uses only constant weight Pauli measurements at the expense of (i) allowing measurement incompatibility and (ii) utilizing a number of measurements greater than that used by standard PBC. We also commented on similarities and differences between this model and the interesting work done by PsiQuantum on fusion-based quantum computation.

In Sec. IV, we formulated three theorems that guarantee distinct specific non-trivial upper bounds for the average weight of the Pauli measurements in a PBC (a measure directly related to the number of CNOTs of the PBC-compiled quantum circuits) and/or the depth of the

PBC (defined as the number of measurement layers). We complemented these more formal results by providing numerical simulations of random quantum circuits. The results indicate that while the theorems promise “only” new upper bounds, the pre-compilation technique underlying them should have an impact that goes beyond that, leading to PBCs that have, in general, smaller average weights.

The greedy algorithm proposed in Sec. V further improves the natural resource savings achieved by the PBC model by providing substantial reductions to the average weight of the Pauli measurements. Importantly, this algorithm provides the option of “distributing the hardness” of the computation as one sees fit. That is, the overhead incurred by the greedy algorithm is entirely classical. Hence, one can push the classical machine by increasing the order of the greedy algorithm if one has as a priority reducing the demands on the quantum hardware. The suitable choice will depend on the (classical and quantum) resources available to the user.

Put together, the techniques and contributions presented in this work significantly improve the state of the art of the PBC model of quantum computation, making it more amenable to practical implementation. A question that is left open is whether PBC can be formulated with constant weights while retaining measurement compatibility and a maximum number of measurements equal to that of qubits.

Another interesting line of research is inspired by the greedy algorithm. As expressed in Algorithm 1, the greedy algorithm attempts to find the lowest weight Pauli operator at a given step. Alternatively, such a greedy algorithm can be coded with a different criterion in mind. For instance, in certain quantum hardware, a gate might exist that is noisier than the CNOT gate, so optimizing the PBC sequence to reduce, for instance, the number of  $Z$  operators might be more beneficial. One can think of more sophisticated algorithms that, rather than trying to find the best solution at each step (within the allowed number of tests), try to optimize things globally. In doing this, the algorithm might avoid reducing the weight at one specific step to reap a better reward at later stages of the computation. This might be accomplished by an algorithm with a global (rather than local) reward system such as seen, for instance, in reinforcement learning algorithms.

We conclude by remarking that the greedy algorithm is looking for lower-weight representations of the generators of an abelian subgroup of the Pauli group. If an algorithm exists that gives concrete performance guarantees for such a task, this could be very impactful in characterizing the experimental feasibility of different algorithms within the PBC framework. The main technical difficulty lies in finding the generators sequentially (i.e. adaptively), as the full sequence of Pauli measurements is unknown *a priori*.

## ACKNOWLEDGEMENTS

FCRP is supported by the Portuguese institution FCT – Fundação para a Ciência e a Tecnologia via the Ph.D. Research Scholarship 2020.07245.BD. E.F.G. acknowledges support from FCT – Fundação para a Ciência e a Tecnologia (Portugal) via project CECINST/00062/2018. This work was supported by the Digital Horizon Europe project FoQaCiA, GA no.101070558, funded by the European Union, NSERC (Canada), and UKRI (U.K.). Numerical simulations were made possible by INCD funded by FCT and FEDER under the project 01/SAICT/2016 n° 022153 and also by the Search-ON2: Revitalization of HPC infrastructure of UMinho (NORTE-07-0162-FEDER-000086), co-funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

## Appendix A: Proof of Theorem 1

Here, we prove Theorem 1.

**Theorem 1** (Improved weights). *Consider a one-way computation to be carried out on a  $t$ -qubit, computation-specific graph state  $|\mathcal{G}\rangle$  with a measurement pattern requiring only measurements along the  $\pm\pi/4$  directions on the equator of the Bloch sphere. By taking on the processing order  $\mathcal{O}_1$  defined in Eq. (3), the weights of the  $2t$  Pauli operators in the (complete) PBC procedure are upper-bounded by  $\{1, 1, 2, \dots, t-1, t-1, t, t\}$ .*

*Proof.* As explained in the main text, any Clifford+ $T$  quantum circuit  $\mathcal{U}$  can be simulated by a one-way computation involving a  $t$ -qubit (computation-specific) graph state,  $|\mathcal{G}\rangle$ . The computation is driven by a sequence of measurements that are broken into layers, with outcomes from one layer determining the bases of measurements in subsequent layers. We can represent this procedure in the form of a circuit as depicted in Fig. 11. Importantly, we note that we are exploring the fact that a discrete set of measurement bases is sufficient for universality [25], notably the measurement bases along the  $\pm\pi/4$  directions on the equator of the Bloch sphere suffice for universal quantum computation. We are assuming that any Pauli measurements, including the readout measurements, have already been removed. (Recall Remark 1.) Additionally, we note that the labeling of the qubits is such that no measurement  $M_j$  depends on the outcome of measurement  $M_i$  with  $i > j$ . This means there is a time-ordering to the measurement pattern so that:  $M_i \prec M_j \implies i < j$ . This ordering is assumed throughout, including in Figs. 11 and 12 where the potential classical influence of a measurement outcome on subsequent measurements is depicted by the classical wires.

As in the regular PBC procedure, the first step consists of transforming each  $T$  gate in this circuit into a  $T$  gadget; this originates the adaptive Clifford circuit,  $\mathcal{C}$ , shown in Fig. 12. We choose to use different letters to denote the measurement outcomes of computational qubits ( $\{s_i\}_{i=1}^t$ ) and those of the auxiliary qubits introduced by the  $T$  gadget ( $\{m_i\}_{i=1}^t$ ). Additionally, we also differentiate the Pauli operators that stem from these two types of measurements; Pauli operators resulting from measurements on computational qubits are denoted by  $P_i$  whilst for those originating from gadget measurements we use  $Q_i$ . The generators of the stabilizer of the graph state,  $\mathcal{S}$ , are represented by  $G_i$  and constructed as prescribed in Eq. (1).

To simulate this computation using a PBC, we need to back-propagate each measurement to the beginning of the circuit, assess into which of the three categories described in Sec. II D 3 it falls, and act accordingly. Looking at Fig. 12, it is easy to understand that the adaptive nature of the circuit implies that any gadget measurement must be performed before the measurement of the corresponding computational qubit, as we need to know whether the Clifford correction  $S^{m_i}$  is present or absent before propagating the measurement of the  $i$ th computational qubit. However, as explained in Sec. IV A, there is still some freedom in the order in which the measurements are back-propagated.

As stated in the formulation of Theorem 1, we consider the processing order  $\mathcal{O}_1$  shown in Eq. (3). We start with the measurement of the first auxiliary qubit. Back-propagating this measurement leads to:  $Z_{t+1} \longrightarrow Q_1 = Z_1 Z_{t+1}$ . It is clear that this operator anti-commutes with  $G_1$ . Thus, the standard PBC procedure informs us that its outcome  $m_1$  is determined in the classical computer by making a coin toss;  $Q_1$  is then dropped from the quantum circuit and replaced by the Clifford operator:

$$V(G_1, Q_1, m_1) = \frac{G_1 + (-1)^{m_1} Q_1}{\sqrt{2}}.$$

To alleviate notation, we will label all  $V$  unitaries by the outcome associated with the Pauli operator that originated it so that  $V(G_1, Q_1, m_1) \equiv V(m_1)$ . We note that this operator entangles the first qubit of the auxiliary register to the first computational qubit and its neighbors on the graph. Finding  $m_1$  decides the presence or absence of the gate  $S^{m_1}$  acting on the first data qubit. This means that we are capable of back-propagating the measurement on that qubit until it reaches the beginning of the quantum circuit depicted in Fig. 12. Doing so leads to:

$$Z_1 \xrightarrow{\mathcal{C}} \begin{cases} \text{if } m_1 = 0 : & P'_1 = Y_1 X_{t+1} \\ \text{otherwise :} & P'_1 = X_1 X_{t+1} \end{cases} .$$

However, we need to remember that the Clifford unitary  $V(m_1)$  is now present in the quantum circuit, and we need to propagate  $P'_1$  through it. For a generic  $V$  operator as given by Eq. (2), an arbitrary Pauli operator  $R$  is

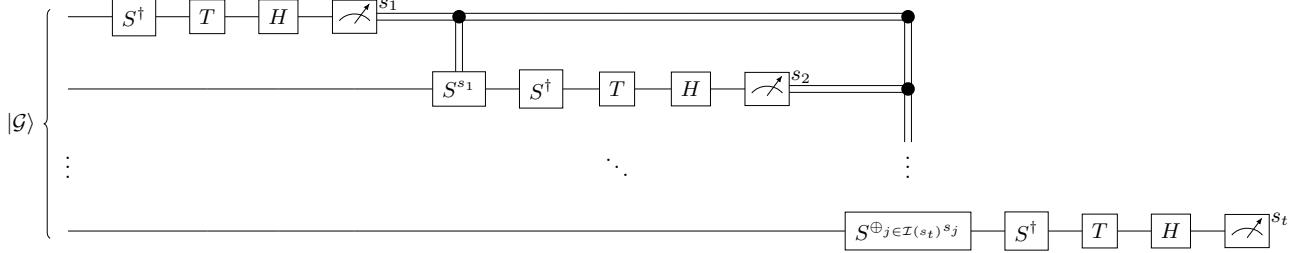


FIG. 11. Circuit corresponding to a given one-way quantum computation on a graph state  $|\mathcal{G}\rangle$ . The fact that the measurements can be restricted to the bases along the angles of  $\pm\pi/4$  on the equator of the Bloch sphere means that we do not need arbitrary  $Z$ -rotation gates, but that the  $T$  gate is sufficient. The measurement outcomes  $s_i$  in a given layer influence measurement bases on subsequent layers; this is indicated by the classical wires depicted in the picture. The notation  $\bigoplus_{j \in \mathcal{I}(s_t)} s_j$  denotes the addition modulo 2 of the outcomes  $s_j$  that influence the measurement basis of the computational qubit  $t$ .

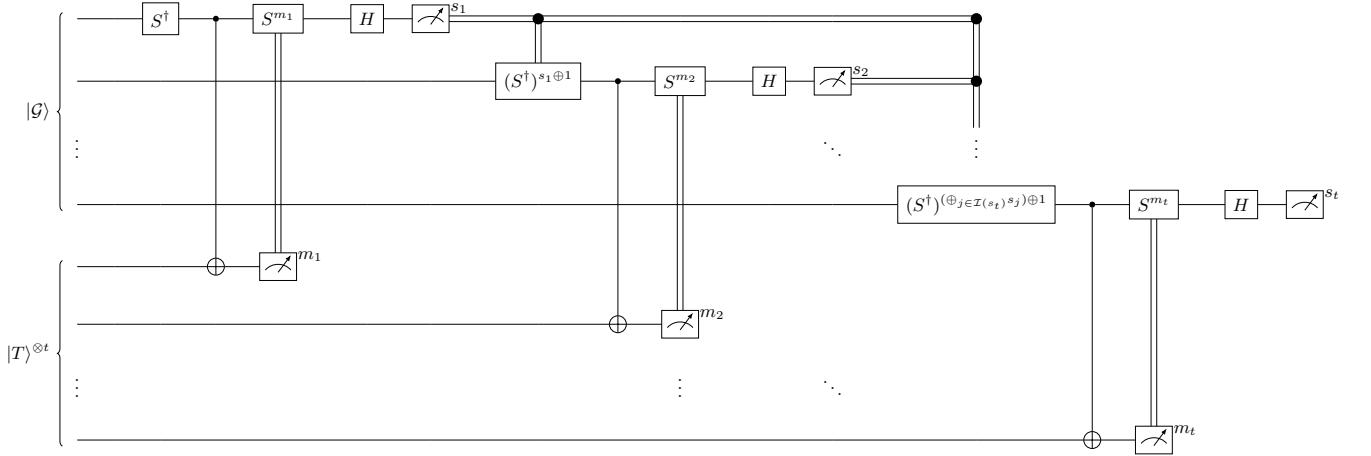


FIG. 12. Transformation of the circuit in Fig. 11 into an adaptive Clifford circuit by replacing each  $T$  gate with the  $T$  gadget depicted in Fig. 1. Note that the outcomes of measurements of computational qubits are denoted by  $s_i$ , while for the outcomes of gadget measurements,  $m_i$  is used.

back-propagated through  $V$  in the following manner:

$$R \xrightarrow{V} \begin{cases} \text{if } [R, P] = [R, Q] = 0 : & R := R \\ \text{if } [R, P] = \{R, Q\} = 0 : & R := \alpha Q P R \\ \text{if } \{R, P\} = [R, Q] = 0 : & R := \alpha P Q R \\ \text{if } \{R, P\} = \{R, Q\} = 0 : & R := -R \end{cases}, \quad (\text{A1})$$

with  $\alpha = (-1)^{\sigma_P + \sigma_Q}$ . Applying these update rules to the present case we obtain:

$$P'_1 \xrightarrow{V(m_1)} \begin{cases} \text{if } m_1 = 0 : & P_1 = \prod_{j \in \mathcal{N}(1)} Z_j Y_{t+1} \\ \text{otherwise :} & P_1 = X_1 X_{t+1} \end{cases},$$

where the product runs over all qubits neighboring the computational qubit 1. In both cases,  $P_1$  is recognized as an anti-commuting Pauli whose outcome  $s_i$  is determined via coin toss and that originates the Clifford unitary:

$$V(s_1) = \frac{G_{j \in \mathcal{N}(1)} + (-1)^{s_1} P_1}{\sqrt{2}},$$

which needs to be placed at the beginning of the quantum circuit. Once again, we observe that this unitary establishes a connection between the first auxiliary qubit and a subset of computational qubits. The attentive reader will note that  $V(s_1)$  also depends on  $m_1$ , as  $P_1 \equiv P_1(m_1)$ , although the chosen notation does not make this dependence explicit.

That neither of these operators is an actual quantum measurement that needs to be performed in the QPU is something that we can make sense of qualitatively. Specifically, we do not expect that single-qubit Pauli measurements on a product state  $|T\rangle^{\otimes t}$  lead to extra computational power beyond that of classical computation (as entanglement is lacking from this scenario). The existence of Pauli measurements of weight 1 at the beginning of the computation would mean that the corresponding (magic) qubit serves the sole purpose of being measured in the  $X$ ,  $Y$ , or  $Z$  basis (and could then be removed from the computation). Together, these two statements mean that the same computation could be performed with fewer qubits thus avoiding the need for

weight-1 measurements at the start of the computation. Note that the same reasoning does not apply for measurements in the middle of the PBC procedure because entanglement starts to arise and a single-qubit measurement performed on an entangled state has a potentially non-trivial influence (suffices to think of 1WQC).

Reasoning in a similar way as illustrated above, we will need to process the remaining  $2t - 2$  Pauli operators; it is clear that the structure of the one-way computation ensures that no  $P_i$  or  $Q_i$  measurement can have a non-trivial presence on the qubits of the (magic) auxiliary register with  $j > i$ . On the other hand, potential  $V$  unitaries added by the processing of previous Pauli operators may lead to non-trivial contributions in auxiliary qubits labeled  $j \leq i$ . In essence, we are saying that these Pauli measurements can take the following form (on the  $t$ -qubit magic register):  $P_i = R_i \otimes I^{\otimes t-i}$ , with  $R_i$  denoting a Pauli operator on  $i$  qubits with weight  $1 \leq w \leq i$ . A similar expression applies to  $Q_i$ .

This guarantees that the  $2t - 2$  Pauli operators processed in this way have maximum weights (in the magic register) given by  $\{2, 2, 3, 3, \dots, t, t\}$ , as stated in the theorem.

As a concluding remark, we remind the reader that not all of these Pauli operators will be measured. Since there are at most  $t$  independent and pairwise commuting Pauli operators on  $t$  qubits, the maximum number of quantum measurements is still  $t$ , as explained in the main text. ■

## Appendix B: Proofs of Theorems 3 and 4

In Sec. IV C, we formulated and demonstrated how the PBC associated with any one-way computation with a single computational layer also has a single measurement layer, that is, all of the  $r \leq t$  Pauli measurements can be performed simultaneously. In this Appendix, we consider the more general situation where the underlying one-way computation has  $d_{1W} > 1$ . There are two different ways to approach this scenario, which lead to Theorems 3 and 4. The proof of the former follows straightforwardly from the proof of Lemma 5.

**Theorem 3** (Improved depth). *By taking up the processing order  $\mathcal{O}_2$  in Eq. (4), the depth of a PBC is upper-bounded by the logical depth of the corresponding one-way quantum computation,  $d_{1W}$ .*

*Proof.* Assume that we take the processing order  $\mathcal{O}_2$ , given by Eq. (4). That is, all of the gadget measurements are dealt with first, followed by the layered propagation of the readout measurements.

From Fig. 12, it may look like this order cannot be realized, because, for instance, it seems that we need to know the Clifford correction  $(S^\dagger)^{s_1 \oplus 1}$ , before propagating the second gadget measurement. In reality, that is not the case, since  $Z_{t+2}$  is transformed into  $Z_2 Z_{t+2}$  via back-propagation through the CNOT gate, and  $Z_2 Z_{t+2}$  commutes with  $(S^\dagger)^{s_1 \oplus 1}$  regardless of the value of  $s_1$ . (This

is strikingly different from what happens with the back-propagation of readout measurements, which *do* require the knowledge of  $S^{m_i}$  and, therefore, the prior determination of the gadget outcome  $m_i$ .)

By choosing the order  $\mathcal{O}_2$ , we see that each Pauli  $Q_i$  stemming from a gadget measurement will take the form  $Q_i = Z_i Z_{t+i}$  and we will have the same sequence of Clifford unitaries,  $V = \prod_j V(m_j)$ , as in the single-layer case.

Similarly, each Pauli  $P_i$  stemming from a readout measurement will take the form given in Eq. (7). The only difference to the single-layer case is that now these  $P_i$  measurements are grouped into layers; operators in the same layer can be processed and measured simultaneously, but only after operators in prior layers have been measured (to fix the Clifford correction factor  $(S^\dagger)^{s_i \oplus 1}$  determined by measurements in previous layers).

This means that we end up with a PBC with  $d_{1W}$  layers with the Pauli operators to be measured given by Eq. (7). Because of the block  $V$ , these Pauli measurements can potentially have weight  $t$ . However, the same observations made at the end of Sec. IV C apply. ■

Theorem 4 provides a result wherein the depth of the PBC may be lower than  $t$  while, at the same time, we can still establish better weights for the Pauli measurements than the trivial upper bound of  $t$ . In this sense, this result supplies us with an intermediate approach between Theorems 1 and 3.

**Theorem 4** (Weight-depth trade-off). *Consider a one-way computation with logical depth  $d_{1W}$  and layering so that the number of computational qubits in layer  $\ell_i$  is  $\kappa_i$ :  $\sum_{i=1}^{d_{1W}} \kappa_i = t$ . By back-propagating the measurements following  $\mathcal{O}_3$  in Eq. (5), the depth of the corresponding PBC is upper-bounded by  $\min\{2d_{1W}-1, t\}$ . Moreover, the weight of the  $2\kappa_i$  Pauli operators stemming from  $GM_{\ell_i}$  and  $RO_{\ell_i}$  measurements is upper bounded by  $\sum_{j=1}^i \kappa_j$ .*

*Proof.* We approach the back-propagation of measurements following the order  $\mathcal{O}_3$  in Eq. (5), rather than  $\mathcal{O}_1$  or  $\mathcal{O}_2$  which were used to prove, respectively, Theorems 1 and 3.

Again, Lemma 5 and Appendix C provide us with important insights that we can use for this proof. It is clear that the  $\kappa_1$  Pauli operators stemming from  $GM_{\ell_1}$  will be back-propagated through the circuit to give:  $Q_i = Z_i Z_{t+i}$ , each of which anti-commutes with  $G_i$ . From the proof of Lemma 5, we know that these operators can all be processed simultaneously in the classical computer adding to the beginning of the circuit  $\kappa_1$  unitaries  $V(m_i)$  of the form given in Eq. (6). Here, we denote by  $V$  the unitary comprised of all of these:  $V = \prod_{i=1}^{\kappa_1} V(m_i)$ . All of the Clifford unitaries  $V(m_i)$  commute, so that they can be added in any order. So far, things completely resemble the single-layer scenario described in full in the main text.

Next, comes the back-propagation of the readout measurements associated with the computational qubits of

the first layer. After being propagated through the adaptive Clifford circuit and the  $\kappa_1 V(m_i)$  unitaries introduced in the previous layer, these will take the form:

$$Z_i \xrightarrow{CV} \begin{cases} \text{if } m_i = 0 : P_i = R_i \left( \prod_{c \in \mathcal{N}(i) \setminus \mathcal{A}} Z_c \right) Y_{t+i} \\ \text{otherwise : } P_i = R_i \left( \prod_{c \in \mathcal{N}(i) \cap \mathcal{A}} Z_c \right) X_i X_{t+i} \end{cases}, \quad (B1)$$

with  $R_i = (-1)^{\sum_{a \in \mathcal{N}(i) \cap \mathcal{A}} m_a} \left( \prod_{b \in \mathcal{N}(i) \cap \mathcal{A}} G_b Z_{t+b} \right)$ , where  $\mathcal{A}$  denotes the set of indices labeling gadget measurements that have been identified as anti-commuting (i.e., which originated  $V(m_i)$  unitaries). In the present computational layer,  $\mathcal{A} = \{1, \dots, \kappa_1\}$ ; but in future steps, this may change as some of the gadget measurements may lead to Pauli operators that need to be measured in the quantum hardware and that, therefore, do not create a unitary  $V(m_i)$ . To understand where Eq. (B1) comes from, check Appendix C and, in particular, Remark 3 therein.

The next step is to assess the  $P_i$  operators given in Eq. (B1). Since no other operators have been measured, what matters is whether they commute or anti-commute with the generators of the graph state  $\{G_j\}_{j=1}^t$ . Interestingly, for  $m_i = 0$ , if  $\mathcal{N}(i) \setminus \mathcal{A} \neq \emptyset$  it is clear that each  $P_i$  will be identified as an anti-commuting Pauli. Since in the first layer, we expect that each qubit has neighbors in the upcoming layers,  $m_i = 0$  implies that  $P_i$  will be an anti-commuting Pauli operator. On the other hand, for  $m_i = 1$ , the only way for  $P_i$  to not be anti-commuting is if  $\mathcal{N}(i) \cap \mathcal{A} = \mathcal{A}$  which we also know not to be true in general. So, *a priori*, none of these Pauli operators seem to be identified as Pauli measurements to be performed in quantum hardware.

Can the inclusion of  $V(s_i)$  unitaries stemming from this same layer change the nature of the other  $P_i$  operators within the layer? Let us suppose that  $P_1$  anti-commutes with a generator  $G_k$ . This will lead to the inclusion of the Clifford unitary:

$$V(s_1) = \frac{G_k + (-1)^{s_1} P_1}{\sqrt{2}}.$$

Two situations can happen. First, an upcoming  $P_i$  operator ( $i \neq 1$ ) may commute with  $G_k$  in which case it is pushed through  $V(s_1)$  without being altered and thus preserves its nature. Second, it may anti-commute with  $G_k$  in which case it will be modified after back-propagation through  $V(s_1)$  following the rule in the second line of Eq. (A1).

This highlights how taking ordering  $\mathcal{O}_3$  substantially complicates things. Operators in a given layer can originate  $V(m_i)$  or  $V(s_i)$  Clifford unitaries that may influence other operators in that same layer, potentially even changing their nature. This is considerably more involved than the single-layer case or the multi-layer scenario using the  $\mathcal{O}_2$  ordering.

To achieve the results stated in the theorem we take the following approach. Suppose that all gadget measurements of an arbitrary layer  $\ell_j$  have been propagated

to the beginning of the quantum circuit leading to the following sequence of operators:  $\{Q_i, Q_{i+1}, \dots, Q_{i+\kappa_j}\}$ . Importantly, all of these Pauli operators are compatible. We take them in increasing order of their indices (but any other order could be considered instead). Taking  $Q_i$ , if it is a Pauli that is recognized as independent and pairwise commuting from all previous measurements performed in the quantum hardware we store the information about that Pauli but do *not* perform the measurement immediately. If, instead, the Pauli operator anti-commutes with a generator of the graph state or some previously performed measurement,  $W$ , we add its corresponding  $V(m_i) = [(-1)^{\sigma_W} W + (-1)^{m_i} Q_i]/\sqrt{2}$  unitary to the circuit.

All the remaining Pauli operators in the layer  $\{Q_{i+1}, \dots, Q_{i+\kappa_j}\}$  are processed exactly in the same way. Importantly, upcoming operators  $Q_j$  (with  $j > i$ ) interact with  $V(m_i)$  either by being propagated without alteration (if they commute with  $W$ ) or by transforming into  $Q'_j = (-1)^{m_i} W Q_j Q_i$ . After the propagation is completed, this operator is evaluated and processed accordingly, either contributing with a new unitary  $V(m_j)$  or being saved for future measurement. After the entire procedure has been completed for a given layer, we are left with a list of operators that have been recognized as Pauli measurements to be performed in the hardware. They can then be measured simultaneously in the quantum hardware, originating a single PBC layer.

For readout measurements, the same procedure as described in the previous paragraph can be applied. This means that, in total, the PBC can have at most  $2d_{1W} - 1$  computational layers. Understanding that the weights of these measured Paulis are the ones in the theorem requires understanding that the Clifford  $V(m_i)$  and  $V(s_i)$  unitaries are the only drivers of the weight increase. In each layer  $\ell_i$ , they ensure that the Pauli operators can only act non-trivially on the first  $\sum_{j=1}^i \kappa_j$  qubits of the magic register. This concludes the proof of the theorem. ■

### Appendix C: Proof of Equation (7)

Here, we provide the explicit proof for the form of the Pauli operators  $P_i$  presented in Eq. (7). It is straightforward to see that when the  $Z_i$  measurements on the computational qubits are pushed through the Clifford circuit  $\mathcal{C}$  (after all corrections  $S^{m_i}$  have been fixed), they are transformed so that

$$Z_i \xrightarrow{C} \begin{cases} \text{if } m_i = 0 : P'_i = Y_i X_{t+i} \\ \text{otherwise : } P'_i = X_i X_{t+i} \end{cases}.$$

Next, we need to understand how each of these operators is back-propagated through the Clifford unitary  $V$  introduced by the Pauli operators  $Q_j$  stemming from the gadget measurements. Recall that  $V = \prod_{j=1}^t V(m_j)$  where each  $V(m_j)$  takes the form in Eq. (6). We note

that  $[V(m_j), V(m_k)] = 0, \forall j, k$ . This is helpful as it allows us to shuffle these unitaries around at our convenience.

Let us take the Pauli operator  $P'_i$ , and see what happens as it is back-propagated through  $V$ . We start by assuming that  $m_i = 0$  so that  $P'_i = Y_i X_{t+i}$ . In this case, the propagation of  $P'_i$  through the unitaries  $V(m_j)$  in  $V$  can be broken down into three different cases (recall Eq. (A1)).

**Case 1** ( $j \neq i \wedge j \notin \mathcal{N}(i)$ ). This means that  $[P'_i, G_j] = [P'_i, Q_j] = 0 \implies [P'_i, V(m_j)] = 0$ . Thus,  $P'_i$  passes unchanged through all such  $V(m_j)$  unitaries.

**Case 2** ( $j \neq i \wedge j \in \mathcal{N}(i)$ ). In this case,  $P'_i$  still commutes with  $Q_j$  but it anti-commutes with  $G_j$ . This means that the back-propagation of  $P'_i$  through all unitaries  $V(m_j)$

falling into this category will happen in the following manner:

$$\begin{aligned} P'_i &\xrightarrow[j_1 \in \mathcal{N}(i)]{} (-1)^{m_{j_1}} G_{j_1} Q_{j_1} P'_i \\ &\xrightarrow[j_2 \in \mathcal{N}(i)]{} (-1)^{m_{j_1} + m_{j_2}} G_{j_1} Q_{j_1} G_{j_2} Q_{j_2} P'_i \\ &\longrightarrow \dots \\ &\longrightarrow P''_i = (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} \prod_{b \in \mathcal{N}(i)} (G_b Q_b) P'_i. \end{aligned} \quad (\text{C1})$$

**Case 3** ( $j = i$ ). Finally, we can push the  $P''_i$  obtained from the previous step through  $V(m_i)$  itself, which means that we have the same commutation properties and in case 2, that is,  $P''_i$  commutes with  $Q_i$  but it anti-commutes with  $G_i$ . This leads to

$$\begin{aligned} P''_i &\xrightarrow[V(m_i)]{} P_i = (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} G_i Q_i \prod_{b \in \mathcal{N}(i)} (G_b Q_b) P'_i \\ &= (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} \left( \prod_{b \in \mathcal{N}(i)} G_b \right) G_i \left( \prod_{c \in \mathcal{N}(i)} Z_c Z_{t+c} \right) Z_i Z_{t+i} Y_i X_{t+i}. \end{aligned} \quad (\text{C2})$$

The important observation now is that  $G_i = X_i \prod_{d \in \mathcal{N}(i)} Z_d$  which leads to the final form for  $P_i$  given by

$$P_i = (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} \left( \prod_{b \in \mathcal{N}(i)} G_b \right) \left( \prod_{c \in \mathcal{N}(i)} Z_{t+c} \right) Y_{t+i}. \quad (\text{C3})$$

Next, we need to do the same for the case when  $m_i = 1$ , which means  $P'_i = X_i X_{t+1}$ . The propagation of this operator through the sequence of unitaries  $V(m_j)$  can also be split into three cases. For convenience, we will now consider them in a different order. We can easily do this because, as we have seen, the operators  $V(m_j)$  all commute and therefore we can shuffle them around at will.

**Case 1** ( $j \neq i \wedge j \notin \mathcal{N}(i)$ ). This means that  $[P'_i, G_j] = [P'_i, Q_j] = 0 \implies [P'_i, V(m_j)] = 0$ . Thus, the operator passes unchanged through all such  $V(m_j)$  unitaries.

**Case 2** ( $j = i$ ). In this case,  $P'_i$  commutes with both  $G_i$  and  $Q_i$  which means that again the operator remains unaffected.

**Case 3** ( $j \neq i \wedge j \in \mathcal{N}(i)$ ). Here, we have the same situation as case 2 of the prior scenario, meaning that  $P_i$  will assume the form given by Eq. (C1). We can re-write it in the following form:

$$P_i = (-1)^{\sum_{a \in \mathcal{N}(i)} m_a} \left( \prod_{b \in \mathcal{N}(i)} G_b \right) \left( \prod_{c \in \mathcal{N}(i)} Z_{t+c} \right) G_i X_{t+i}. \quad (\text{C4})$$

Equations (C3) and (C4) correspond to the result presented in Eq. (7) concluding the desired proof.

**Remark 3** (Scope of applicability of the results). Eqs. (C1) and (C2) are more general forms of Eqs. (C4) and (C3) respectively, and are useful when we consider a computation with multiple layers whose measurements are processed according to the ordering  $\mathcal{O}_3$  given in Eq. (5). In that case, Eqs. (C1) and (C2) remain valid with the minimal modification that the sums and products run over the elements of the neighborhood of  $i$  which have previously been identified as anti-commuting Pauli operators. This subtle new imposition has important consequences. Notably, Eqs. (C3) and (C4) are no longer valid since they were obtained by simplifications that assume that all neighbors of the computational qubit  $i$  are involved in the products. While that is verified for the multiple-layer scenario when doing the back-propagation following the ordering  $\mathcal{O}_2$  in Eq. (4), it is not the case when the ordering  $\mathcal{O}_3$  is considered. This observation was used in the proof of Theorem 4 outlined in Appendix B, leading to Eq. (B1).

#### Appendix D: Further results and comments concerning the greedy algorithm

In Sec. V, we saw how Algorithm 1 provided important improvements to the average weight of PBCs associated with both hidden-shift and random quantum circuits. The performance was analyzed both for smaller

circuits using a real, Schrödinger-type simulator and also for larger circuits using a dummy simulator. That the results obtained with the latter can be used to extract conclusions was demonstrated numerically in our prior work [11], but also verified concretely for the results with the greedy algorithm with different orders by simulating the smaller circuits with the dummy simulator and verifying that the results obtained were equivalent to those obtained with the real, Schrödinger-type simulator.

For the smaller circuits, a total of 100 circuits were transformed into PBCs using a total of 1024 shots/circuit. For the larger circuits, 50 circuits were considered, again using 1024 shots for each. We will now analyze the performance of the greedy algorithm in a slightly different manner. We consider random quantum circuits with  $n = 49$  and  $t = \{60, 70, 80, 90, 100\}$ . For each of these  $T$  counts, we generated five random quantum circuits each of which we compiled into a single PBC, corresponding to the PBC along the path where all outcomes yield 0. The motivation to fix the path is two-fold. First, it allows us to look into the impact of choosing different orders for the greedy along a fixed sequence of Pauli operators (and not just on average). Secondly, it allows us to look at circuits with larger  $T$  counts while still maintaining reasonable simulation times even for larger values of  $go$ . The results obtained are presented in Table III; a visual depiction of the evolution of the average weight of the Pauli measurements for the different values of  $go$  can be seen in Fig. 13 for the first circuit of each  $T$ -count value.

Importantly, we see that, for these larger PBCs, involving more qubits and more Pauli measurements, setting  $go = 0$  often leads to no improvement whatsoever. Contrastingly, setting  $go = 1, 2$ , or  $3$  leads, respectively, to improvements between 12.3% and 17.9%, 17.7% and 25.6%, and 21.8% and 30.5%. These results corroborate the observations made in the main text that for random quantum circuits and a fixed value of  $go$ , increasing  $t$  tends to lead to smaller improvements by Algorithm 1. Nevertheless, the improvements are still substantial, even for these larger values of  $t$ .

One may wonder whether it would be possible to reduce the overhead associated with the greedy algorithm if, rather than searching for the Pauli measurement with smallest weight among the (sub)sets  $\mathcal{W} \subseteq \{1, \dots, r-1\}$  with size  $r-1-a$  and  $a$ , with  $0 \leq a \leq go \leq (r-1)/2$ , one could stop the search after a predetermined number of attempted combinations, while still guaranteeing that, with high probability a small-weight (even if not the smallest-weight) Pauli measurement was found. The success of such an approach relies on understanding whether the fraction of combinations that reduce the weight with respect to that of the original Pauli measurement is small or large at each step. Taking one of the random quantum circuits with  $t = 60$ , we looked into the weight distribution at each step, evaluating what percentage of combinations have a weight that is smaller than the one obtained in the absence of the greedy algorithm. Figs. 14 and 15

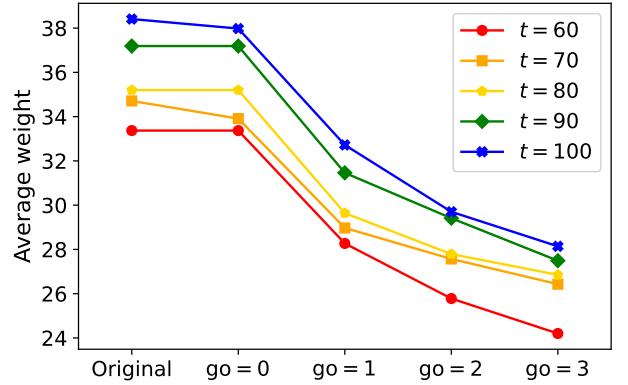


FIG. 13. The figure shows the evolution of the average weight for five randomly generated circuits with  $T$ -count 60, 70, 80, 90, and 100 as the order of the greedy algorithm is increased from 0 until 3.

suggest that the majority of combinations tested by the greedy algorithm increase the weight compared to that obtained naturally from the PBC procedure (in the absence of greedy). These results suggest that stopping the greedy ahead of time (say halfway through the full set of combinations) might significantly hinder the overall performance of the algorithm as presented in Table III.

Another relevant line of inquiry is whether alternative formulations of the greedy algorithm might exist that outperform Algorithm 1. As we have seen, the latter works by searching for Pauli measurements with better weight among all the (sub)sets  $\mathcal{W} \subseteq \{1, \dots, r-1\}$  with size  $r-1-a$  and  $a$ , with  $0 \leq a \leq go \leq (r-1)/2$ . Rather than doing so, one could search for Pauli measurements with smaller weight by picking the size of each subset and its elements uniformly at random. Evidently, for a fair comparison with the previous formulation, the same number of subsets as associated with a fixed  $go$  needs to be used. This alternative procedure randomizes the search for Pauli operators with smaller weight compared to the more structured search described in Sec. V. The goal is to understand if one reaps benefits from potentially allowing a larger number of Pauli operators to be combined to yield a new Pauli measurement.

Our intuition was that the original (more structured) formulation, should be beneficial in the context of more structured quantum circuits such as the hidden-shift circuits. This was confirmed by running this randomized version of the greedy algorithm in the smaller hidden-shift circuits ( $n = 10, t = 14$ ) allowing at each step the same number of combinations as used with the structured version set with  $go = 1$ . The improvements obtained were roughly reduced in half by using the randomized version of the algorithm instead of the more structured approach.

For the small random quantum circuits and a number of combinations corresponding to those used when setting  $go = 1$ , the differences in performance are not as striking. In those circuits, the more structured algo-

$T$ count	Original	go = 0	go = 1	go = 2	go = 3
60	33.37	33.37 (0%)	28.27 (-15.3%)	25.78 (-22.7%)	24.20 (-27.5%)
	33.05	31.68 (-4.14%)	28.17 (-14.8%)	26.07 (-21.1%)	24.60 (-25.6%)
	31.42	31.42 (0%)	25.98 (-17.3%)	24.02 (-23.6%)	22.43 (-28.6%)
	28.78	28.78 (0%)	24.42 (-15.2%)	21.42 (-25.6%)	20.02 (-30.5%)
	32.85	31.85 (-3.04%)	28.02 (-14.7%)	26.40 (-19.6%)	24.63 (-25.0%)
70	34.71	33.91 (-2.30%)	28.97 (-16.5%)	27.57 (-20.6%)	26.43 (-23.9%)
	33.66	33.66 (0%)	28.69 (-14.8%)	26.53 (-21.2%)	26.17 (-22.2%)
	33.13	33.13 (0%)	28.93 (-12.7%)	26.46 (-20.1%)	24.81 (-25.1%)
	30.81	30.81 (0%)	26.54 (-13.9%)	24.03 (-22.0%)	22.67 (-26.4%)
	32.00	32.00 (0%)	27.06 (-15.4%)	25.14 (-21.4%)	23.49 (-26.6%)
80	35.20	35.20 (0%)	29.64 (-15.8%)	27.79 (-21.1%)	26.85 (-23.7%)
	32.29	32.28 (-0.04%)	27.03 (-16.3%)	25.25 (-21.8%)	23.73 (-26.5%)
	33.19	33.19 (0%)	28.71 (-13.5%)	26.08 (-21.4%)	24.44 (-26.4%)
	33.68	33.68 (0%)	27.64 (-17.9%)	27.08 (-19.6%)	24.51 (-27.2%)
	33.40	33.40 (0%)	28.96 (-13.3%)	26.28 (-21.3%)	25.39 (-24.0%)
90	37.19	37.19 (0%)	31.46 (-15.4%)	29.41 (-20.9%)	27.49 (-26.1%)
	34.70	34.66 (-0.13%)	30.44 (-12.3%)	28.30 (-18.4%)	26.83 (-22.7%)
	35.14	35.14 (0%)	30.07 (-14.4%)	28.46 (-19.0%)	27.07 (-23.0%)
	34.24	34.24 (0%)	29.51 (-13.8%)	27.36 (-20.1%)	25.41 (-25.8%)
	33.48	33.48 (0%)	28.81 (-13.9%)	25.90 (-22.6%)	24.93 (-25.5%)
100	38.41	37.98 (-1.12%)	32.72 (-14.8%)	29.70 (-22.7%)	28.14 (-26.7%)
	34.68	34.68 (0%)	29.72 (-14.3%)	27.09 (-21.9%)	25.94 (-25.2%)
	35.31	35.31 (0%)	29.81 (-15.6%)	27.15 (-23.1%)	25.90 (-26.6%)
	36.39	36.39 (0%)	31.44 (-13.6%)	29.16 (-19.9%)	27.38 (-24.8%)
	34.91	34.91 (0%)	30.38 (-13.0%)	28.72 (-17.7%)	27.31 (-21.8%)

TABLE III. Effect of the greedy algorithm as its order is increased from 0 to 3 for five different  $T$  counts (60, 70, 80, 90, and 100) and five randomly generated circuits for each value of  $t$ . The reductions achieved by the application of the greedy algorithm are clear as soon as the order is greater or equal to 1. For go = 0 improvements may or may not occur.

rithm outperforms the randomized version by attaining reductions of the average weight that are roughly 10% larger than the reductions obtained by the randomized version of the algorithm. This difference increases as  $t$  becomes larger. For instance, for the 5 circuit instances with  $t = 100$  presented in Table III, the improvements attained by the randomized-search greedy algorithm are

between 52% and 73% lower than the ones obtained by the structured search version of the algorithm. Intriguingly, for go = 2 the difference in performance is slightly diminished and, for the latter circuits, the improvements attained by the randomized-search greedy algorithm are between 27% and 38% smaller than the ones obtained with the structured search approach.

- 
- [1] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negevvergne, Quantum Circuit Simplification and Level Compaction, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **27**, 436 (2008).
- [2] R. Iten, R. Moyard, T. Metger, D. Sutter, and S. Woerner, Exact and Practical Pattern Matching for Quantum Circuit Optimization, *ACM Transactions on Quantum Computing* **3**, 1 (2022).
- [3] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus, *Quantum* **4**, 279 (2020).
- [4] A. Kissinger and J. van de Wetering, Reducing the number of non-Clifford gates in quantum circuits, *Phys. Rev. A* **102**, 022406 (2020).
- [5] M. Amy, D. Maslov, and M. Mosca, Polynomial-Time T-Depth Optimization of Clifford+T Circuits Via Matroid Partitioning, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **33**, 1476 (2014).
- [6] L. E. Heyfron and E. T. Campbell, An efficient quantum compiler that reduces T count, *Quantum Science and Technology* **4**, 015004 (2018).
- [7] M. Amy and M. Mosca, T-Count Optimization and Reed-Muller Codes, *IEEE Transactions on Information Theory* **65**, 4771–4784 (2019).
- [8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, USA, 2011).
- [9] A. Chi-Chih Yao, Quantum circuit complexity, in *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science* (1993) pp. 352–361.

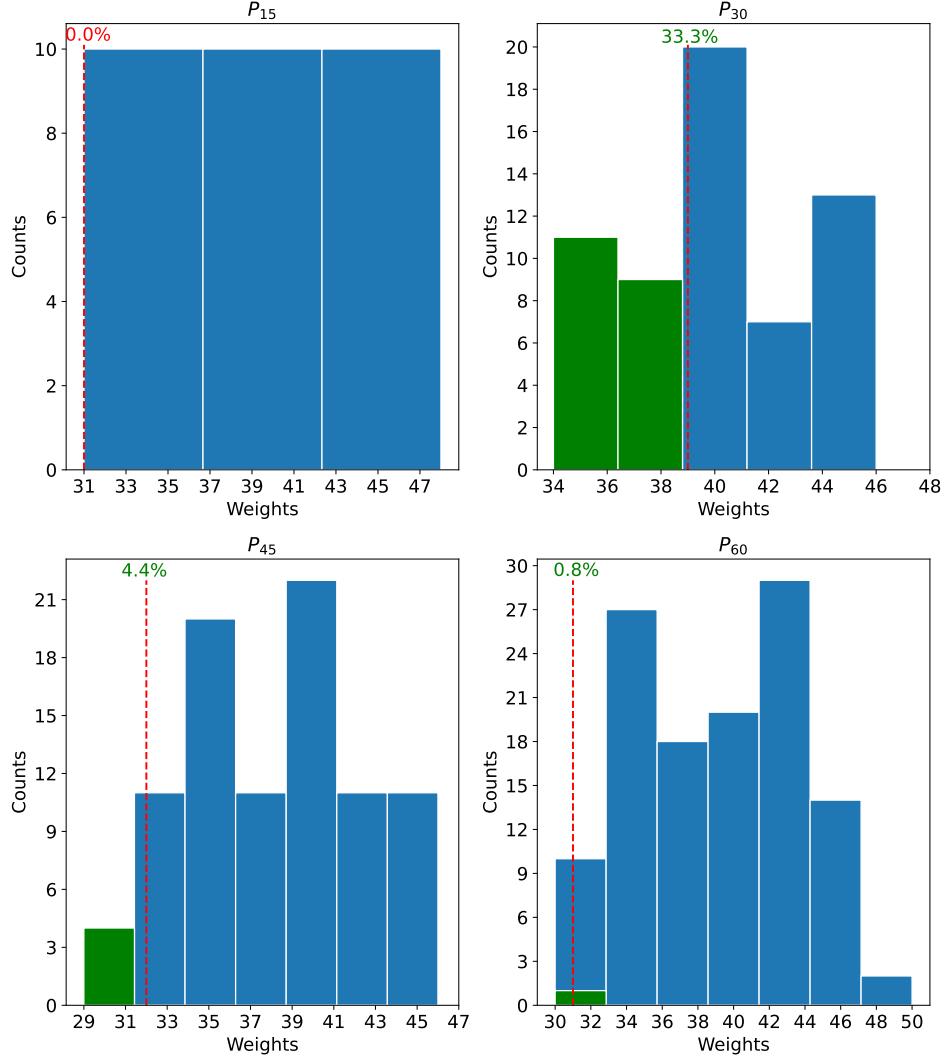


FIG. 14. Distribution of the possible weights for the Pauli measurement at time-steps  $r = 15, 30, 45$ , and  $60$  of one of the random quantum circuits in Table III with  $t = 60$  when setting  $\text{go} = 1$ . The dashed, red line represents the weight of the Pauli measurement in the absence of the greedy algorithm and the percentage on top of it indicates the proportion of combinations (represented in green) that lead to a weight smaller than that.

- [10] S. Bravyi, G. Smith, and J. A. Smolin, Trading Classical and Quantum Computational Resources, *Phys. Rev. X* **6**, 021043 (2016).
- [11] F. C. R. Peres and E. F. Galvão, Quantum circuit compilation and hybrid computation using Pauli-based computation, *Quantum* **7**, 1126 (2023).
- [12] D. Gottesman, The Heisenberg Representation of Quantum Computers, in *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics* (1998) pp. 32–43, [arXiv:quant-ph/9807006 \[quant-ph\]](#).
- [13] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. thesis, Caltech (1997), [arXiv:quant-ph/9705052 \[quant-ph\]](#).
- [14] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Quantum error correction and orthogonal geometry, *Phys. Rev. Lett.* **78**, 405 (1997).
- [15] M. V. den Nest, W. Dür, A. Miyake, and H. J. Briegel, Fundamentals of universality in one-way quantum computation, *New Journal of Physics* **9**, 204 (2007).
- [16] D. Aharonov, A Simple Proof that Toffoli and Hadamard are Quantum Universal (2003), [arXiv:quant-ph/0301040 \[quant-ph\]](#).
- [17] P. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, On universal and fault-tolerant quantum computing: a novel basis and a new constructive proof of universality for Shor’s basis, in *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)* (1999) pp. 486–494.
- [18] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [19] Y. Shi, Both toffoli and controlled-not need little help to do universal quantum computation (2002), [arXiv:quant-ph/0205115 \[quant-ph\]](#).

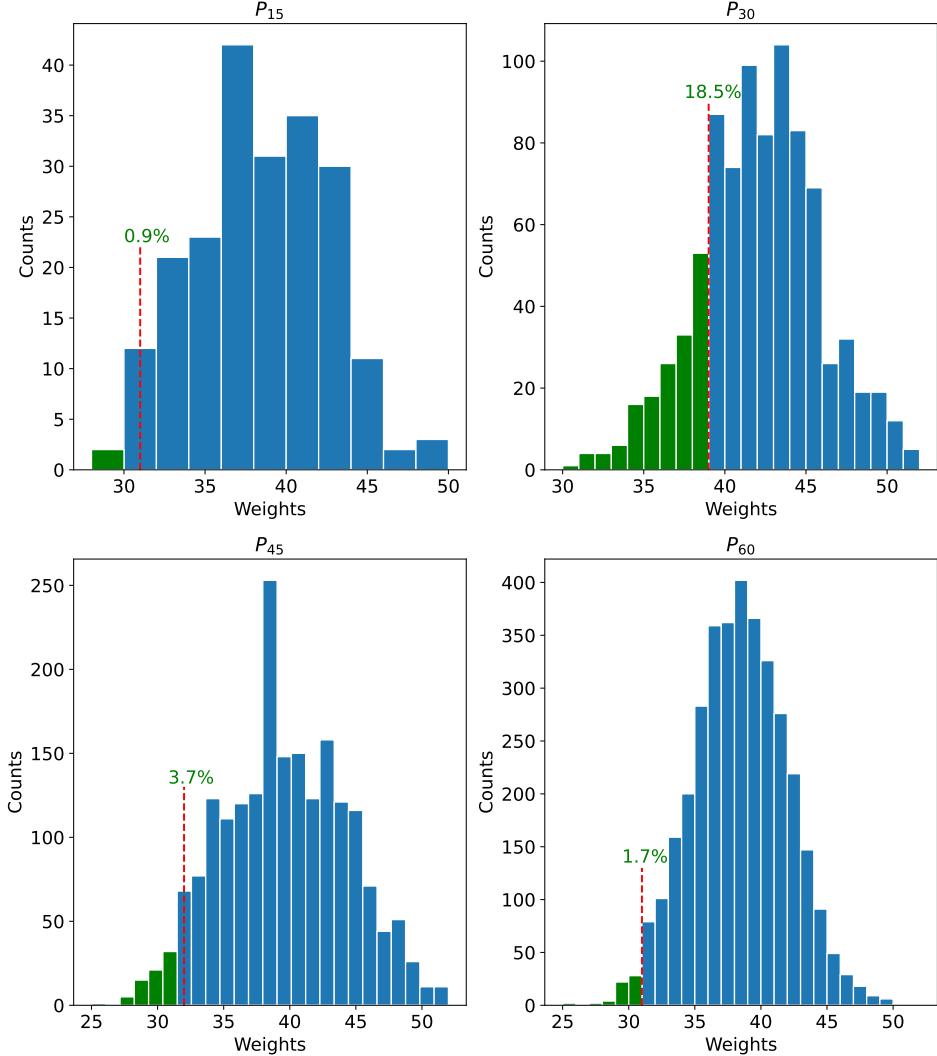


FIG. 15. Distribution of the possible weights for the Pauli measurement at time-steps  $r = 15, 30, 45$ , and  $60$  of one of the random quantum circuits in Table III with  $t = 60$  when setting  $go = 2$ . The dashed, red line represents the weight of the Pauli measurement in the absence of the greedy algorithm and the percentage on top of it indicates the proportion of combinations (represented in green) that lead to a weight smaller than that.

- [20] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, *Phys. Rev. A* **71**, 022316 (2005).
- [21] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, *Phys. Rev. Lett.* **102**, 110502 (2009).
- [22] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [23] M. Hein, J. Eisert, and H. J. Briegel, Multiparty entanglement in graph states, *Phys. Rev. A* **69**, 062311 (2004).
- [24] M. Van den Nest, A. Miyake, W. Dür, and H. J. Briegel, Universal Resources for Measurement-Based Quantum Computation, *Phys. Rev. Lett.* **97**, 150504 (2006).
- [25] A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal Blind Quantum Computation, in *2009 50th Annual IEEE Symposium on Foundations of Computer Science* (2009) pp. 517–526.
- [26] M. Mhalla and S. Perdrix, Graph States, Pivot Minor, and Universality of (X,Z)-Measurements, *Int. J. Unconv. Comput.* **9**, 153 (2012).
- [27] A. Mantri, T. F. Demarie, and J. F. Fitzsimons, Universality of quantum computation with cluster states and (x, y)-plane measurements, *Scientific Reports* **7**, 42861 (2017).
- [28] Y. Takeuchi, T. Morimae, and M. Hayashi, Quantum computational universality of hypergraph states with pauli-X and Z basis measurements, *Scientific Reports* **9**, 13585 (2019).
- [29] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, *Phys. Rev. A* **68**, 022312 (2003).
- [30] J. Miller and A. Miyake, Hierarchy of universal entanglement in 2D measurement-based quantum computation, *npj Quantum Information* **2**, 16036 (2016).
- [31] Y. Takeuchi, Catalytic Transformation from Computationally-Universal to Strictly-Universal

- Measurement-Based Quantum Computation (2023), [arXiv:2312.16433 \[quant-ph\]](https://arxiv.org/abs/2312.16433).
- [32] M. A. Nielsen, Quantum computation by measurement and quantum memory, *Physics Letters A* **308**, 96 (2003).
  - [33] D. W. Leung, Quantum computation by measurements, *International Journal of Quantum Information* **02**, 33 (2004).
  - [34] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, *Nature* **402**, 390 (1999).
  - [35] S. Perdrix, State transfer instead of teleportation in measurement-based quantum computation, *International Journal of Quantum Information* **03**, 219 (2005).
  - [36] P. Aliferis and D. W. Leung, Computation by measurements: A unifying picture, *Phys. Rev. A* **70**, 062314 (2004).
  - [37] P. Jorrand and S. Perdrix, Unifying quantum computation with projective measurements only and one-way quantum computation, in *Quantum Informatics 2004*, Vol. 5833, edited by Y. I. Ozhigov, International Society for Optics and Photonics (SPIE, 2005) pp. 44 – 51.
  - [38] A. M. Childs, D. W. Leung, and M. A. Nielsen, Unified derivations of measurement-based schemes for quantum computation, *Phys. Rev. A* **71**, 032318 (2005).
  - [39] M. Yoganathan, R. Jozsa, and S. Strelchuk, Quantum advantage of unitary Clifford circuits with magic state inputs, *Proc. R. Soc. A* **475**, 20180427 (2019).
  - [40] F. C. R. Peres, Pauli-based model of quantum computation with higher-dimensional systems, *Phys. Rev. A* **108**, 032606 (2023).
  - [41] A. Broadbent and E. Kashefi, Parallelizing quantum circuits, *Theoretical Computer Science* **410**, 2489 (2009).
  - [42] M. J. Hoban, J. J. Wallman, H. Anwar, N. Usher, R. Raussendorf, and D. E. Browne, Measurement-based classical computation, *Phys. Rev. Lett.* **112**, 140505 (2014).
  - [43] S. Bartolucci, P. Birchall, H. Bombín, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, M. Pant, F. Pastawski, T. Rudolph, and C. Sparrow, Fusion-based quantum computation, *Nature Communications* **14**, 912 (2023).
  - [44] D. E. Browne and T. Rudolph, Resource-Efficient Linear Optical Quantum Computation, *Phys. Rev. Lett.* **95**, 010501 (2005).
  - [45] F. C. R. Peres, CompHybPBC, <https://github.com/fcrperes/CompHybPBC> (2023).
  - [46] R. H. S. S. H. C. du Toit, A. G. W. Steyn, *Graphical Exploratory Data Analysis* (Springer New York, NY, 1986).
  - [47] I. L. Markov and Y. Shi, Simulating Quantum Computation by Contracting Tensor Networks, *SIAM Journal on Computing* **38**, 963–981 (2008).