arXiv:2209.14158v3 [quant-ph] 29 Nov 2024

Single-qubit gate teleportation provides a quantum advantage

Libor Caha, Xavier Coiteux-Roy and Robert König

School of Computation, Information and Technology, Technical University of Munich & Munich Center for Quantum Science and Technology, Munich, Germany.

Gate-teleportation circuits are arguably among the most basic examples of computations believed to provide a quantum computational advantage: in seminal work [1], Terhal and DiVincenzo have shown that these circuits elude simulation by efficient classical algorithms under plausible complexity-theoretic assumptions. Here we consider possibilistic simulation [2], a particularly weak form of this task where the goal is to output any string appearing with non-zero probability in the output distribution of the circuit. We show that even for single-qubit Clifford-gate-teleportation circuits this simulation problem cannot be solved by constant-depth classical circuits with bounded fan-in gates. Our results are unconditional and are obtained by a reduction to the problem of computing parity, a well-studied problem in classical circuit complexity.

Contents

1 Introduction and main result **2**

2 Locality properties of classical circuits **8**
2.1 Locality of functions . . . 8
2.2 Locality of circuits . . . 9
2.3 No-signaling features of NC^0-circuits . . . 10

3 Quantum advantage against NC^0 **13**
3.1 Relating $\mathsf{PSIM}[\mathsf{Telep}]$ to $\mathsf{PSIM}[\mathsf{CliffP}(Q)]$ . . . 13
3.1.1 Motivation . . . 14
3.1.2 An algorithm for $\mathsf{PSIM}[\mathsf{CliffP}(Q)]$ . . . 16
3.2 From $\mathsf{PSIM}[\mathsf{CliffP}(Q)_L]$ to learning stabilizers of $(I \otimes (C_L \cdots C_1))|\Phi\rangle$ . . . 23
3.2.1 On the need for possibilistic state tomography . . . 23
3.2.2 Possibilistic state tomography from the magic square . . . 24
3.3 Quantum advantage against NC^0 from $\mathsf{PSIM}[\mathsf{Telep}]$ . . . 27
3.3.1 PARITY as a word problem for the Clifford group . . . 28
3.3.2 $\mathsf{PSIM}[\mathsf{Telep}]$ cannot be in NC^0 . . . 29
3.3.3 Other word problems for Clifford modulo Pauli groups . . . 30

A $\mathsf{PSIM}[\mathsf{CliffP}(Q)]$ and possibilistic state tomography **31**

1 Introduction and main result

A quantum circuit $\mathcal{Q}$ on n qubits takes as input a classical bit string $I = (I_1, \ldots, I_m) \in \{0,1\}^m$. Starting from a product state $|0\rangle^{\otimes n}$, it applies one- and two-qubit gates that may be classically controlled by constant-size subsets $\{I_j\}_j$ of input bits, and finally measures each qubit in the computational basis to yield an outcome $O \in \{0,1\}^n$. A prominent example is the single-qubit (Clifford)-gate-teleportation circuit Telep_n (on $2n$ qubits) shown in Fig. 1. The terminology used here alludes to the fact that it is composed of several copies of the gate-teleportation protocol of Gottesman and Chuang [3]. The latter teleports an unknown quantum state while simultaneously applying a Clifford unitary to it. Here we seek to characterize the computational power of the circuit Telep_n.

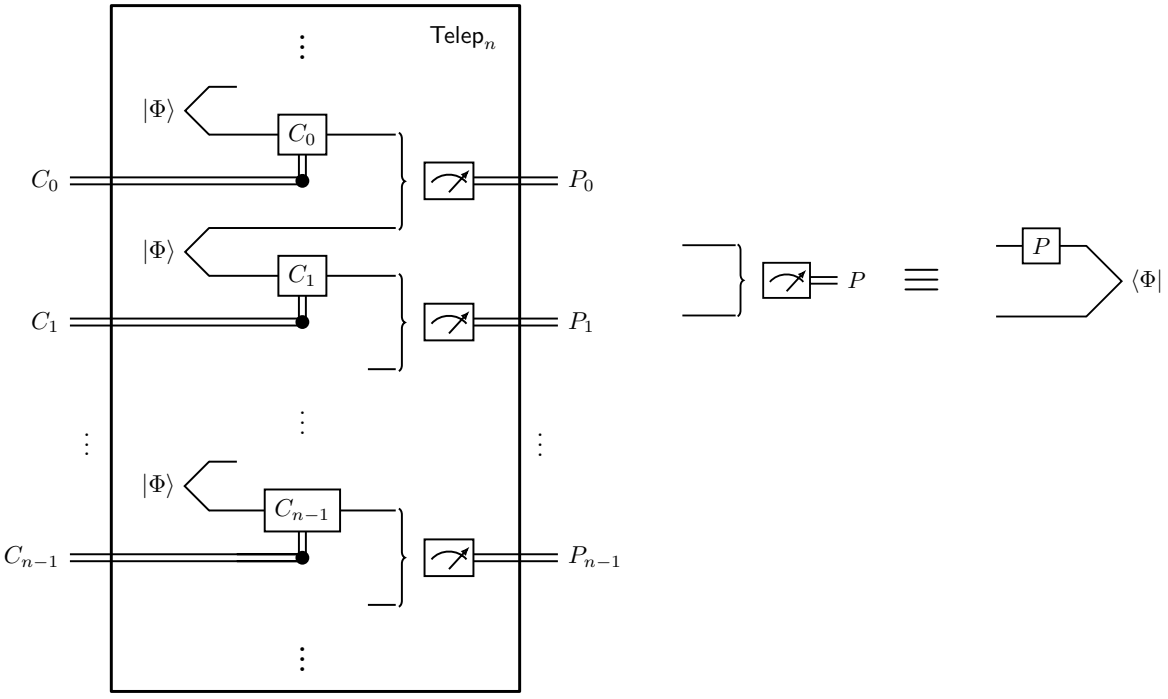
The behavior of a quantum circuit $\mathcal{Q}$ is fully specified by the conditional distribution $p^{\mathcal{Q}}(O|I)$ of the output O given a fixed input I. Its computational power is directly tied to the difficulty of sampling from $p^{\mathcal{Q}}(\cdot|I)$ for different inputs I, and, more generally, the hardness of computing or approximating properties of this collection of distributions. Corresponding notions of classically simulating $\mathcal{Q}$ give rise to various computational problems with more stringent notions of simulation associated with harder computational problems. Arguably one of the weakest notions of simulating $\mathcal{Q}$ is that of possibilistic simulation, a convenient term introduced by Wang in [2] for a concept implicitly used in earlier work [4, 5]. The corresponding computational problem, which we denote by $\mathsf{PSIM}[\mathcal{Q}]$, asks, for a given input I (an instance of $\mathsf{PSIM}[\mathcal{Q}]$), to produce $O \in \{0,1\}^n$ such that $p^{\mathcal{Q}}(O|I) > 0$, i.e., such that O occurs with non-zero probability in the output distribution of $\mathcal{Q}$.

Our main result is that, for the gate-teleportation circuit Telep, even the problem $\mathsf{PSIM}[\mathsf{Telep}]$ is hard for constant-depth classical circuits. To state this, recall that an NC^0-circuit is a constant-depth polynomial-size classical circuit with bounded fan-in gates (where, following standard terminology, we often refer to an infinite circuit family simply as a circuit).

Theorem 1.1. *The* Telep *quantum circuit trivially solves* $\mathsf{PSIM}[\mathsf{Telep}]$ *for every instance. In contrast, any (even non-uniform)* NC^0*-circuit fails to solve* $\mathsf{PSIM}[\mathsf{Telep}]$ *on certain instances.*

We note that average-case statements for suitable distributions over instances of $\mathsf{PSIM}[\mathsf{Telep}]$ can be derived in a similar manner as e.g., [4, Supplementary material]. Here we focus on the worst case for simplicity.

The study of the difficulty of classically simulating quantum gate teleportation was initiated in seminal work by Terhal and DiVincenzo [1]. It was shown in [1] (see also [6, 7]) that there is no efficient classical algorithm that can sample exactly from the output distribution of a variant of the gate-teleportation circuit unless the polynomial hierarchy collapses. The gate-teleportation circuit used in this argument acts in parallel on polynomially many qubits, and realizes any polynomial-size quantum circuit by a post-selected constant-depth quantum circuit. Furthermore, gate-teleportation circuits when combined with post-selection gives the complexity class $\mathsf{PostBQP}$ of (uniform) polynomial-size post-selected quantum circuits. This leads to the hardness of classically sampling from their output distribution (also



(a) The Telep$_n$ circuit.

(b) Pauli operator output after the Bell measurement with $|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Figure 1: The gate-teleportation circuit Telep$_n$. The circuit takes as input an n-tuple $(C_0, \ldots, C_{n-1}) \in \mathcal{C}^n$ of single-qubit Clifford group elements. Each Clifford is applied to one half of a Bell state. Subsequently, Bell-state measurements are performed on pairs of qubits. The output can be associated with an n-tuple $(P_0, \ldots, P_{n-1}) \in \mathcal{P}^n$ of Pauli elements from the set $\mathcal{P} = \{I, X, Y, Z\}$ corresponding to different outcomes of Bell measurements. The circuit has a cyclic structure: the n-th Bell measurement is applied on the first and last qubits in the figure. As the single-qubit Clifford group $\mathcal{C} = \mathcal{P} \cdot \{I, H, S, HS, SHS, (HS)^2\}$ consists of $|\mathcal{C}| = 24$ elements, its elements can be encoded into 5-bit strings. Similarly, elements of $\mathcal{P}$ can be encoded as 2-bit strings (pairs of bits). Correspondingly, we will often interpret inputs $(C_0, \ldots, C_{n-1})$ as elements of $(\{0,1\}^5)^n$ and outputs $(P_0, \ldots, P_{n-1})$ as elements of $(\{0,1\}^2)^n$.

called weak simulation in Ref. [8]) under plausible complexity-theoretic assumptions.

This line of reasoning has also been used to show hardness of simulating "commuting quantum computation" as formalized by IQP circuits [9], level-1-quantum approximate optimization (QAOA) circuits [10] and boson sampling circuits [11].

Proof outline

We establish Theorem 1.1 by connecting the gate-teleportation simulation problem PSIM [Telep] to a basic (trivial) result from classical complexity theory, namely the

fact that the parity function cannot be computed by an NC^0-circuit. Our argument proceeds through a reduction to an intermediate computational problem which relates $\mathsf{PSIM}\,[\mathsf{Telep}]$ and PARITY.

Let us briefly describe the intermediate problem and give a high-level overview of our reasoning.

Quantum advantage against NC^0

The intermediate problem we use in our analysis is another simulation problem $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$ for a two-qubit quantum circuit $\mathsf{CliffP}(Q)_L$. The circuit $\mathsf{CliffP}(Q)_L$ is defined by using an (arbitrary but fixed) function $Q : \mathcal{C}^L \to \mathcal{P}$. The circuit $\mathsf{CliffP}(Q)_L$ is illustrated in Fig. 2. Here and below we denote by $\mathcal{P}$ the set $\mathcal{P} = \{I, X, Y, Z\}$ of Pauli operators and by $\mathcal{C}$ the set $\mathcal{C} = \mathcal{P} \cdot \{I, H, S, HS, SHS, (HS)^2\}$ of single-qubit Cliffords. (These sets are representatives of different cosets of the Pauli respectively Clifford groups when taking the quotient with respect to global phases. We refer to e.g., [12] for details.) It takes as input $L+1$ single-qubit Clifford gates

$$(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$$

and outputs a Pauli $P \in \mathcal{P}$, see the caption of Fig. 1 for a definition of these sets. We call $\mathsf{CliffP}(Q)_L$ the Pauli-corrected Clifford circuit with correction function Q.

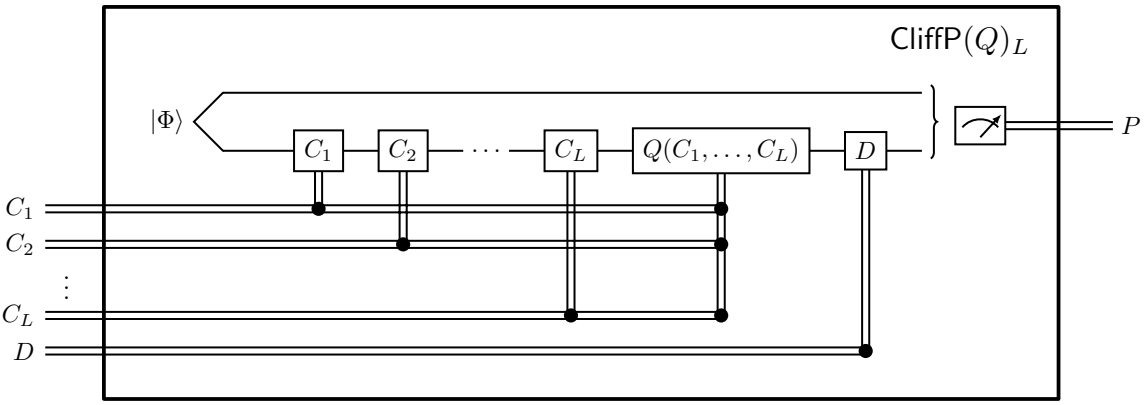


Figure 2: The Pauli-corrected Clifford circuit $\mathsf{CliffP}(Q)_L$ with correction function Q: on input $(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$, the Clifford gates $C_1, \ldots, C_L$ are successively applied to one half of a Bell state $|\Phi\rangle$. A Pauli "correction" operator $Q(C_1, \ldots, C_L)$ is applied to the resulting state. Finally, the state is measured in a basis of maximally entangled states that is obtained by rotating the Bell basis (as specified by the Clifford D). The output is a single-qubit Pauli $P \in \mathcal{P}$ obtained from a Bell measurement.

In Section 3, we show the following: assume that C is an (even non-uniform) NC^0-circuit solving $\mathsf{PSIM}\,[\mathsf{Telep}_n]$ (i.e., producing a valid output for all instances). Then, for $L = \Theta(n)$, there is a function $Q : \mathcal{C}^L \to \mathcal{P}$ (that depends on C) and a non-uniform NC^0-circuit C' solving $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$. The construction of C' makes use of C in a non-black-box manner and relies on no-signaling arguments from Section 2. We summarize this (with a slight abuse of notation) as

$$\Big(\mathsf{PSIM}\,[\mathsf{Telep}] \in \mathsf{NC}^0/\mathsf{poly}\Big) \quad \Rightarrow \quad \Big(\exists Q\colon \mathsf{PSIM}\,[\mathsf{CliffP}(Q)] \in \mathsf{NC}^0/\mathsf{poly}\Big)\,, \quad (1)$$

where $\mathsf{NC}^0/\mathsf{poly}$ is the class of problems solved by non-uniform families of NC^0-circuits.

Our second reduction heavily borrows from ideas developed in [13] in the context of interactive quantum advantage protocols. By adapting (specializing) these techniques, we show that the ability of possibilistically simulating $\mathsf{CliffP}(Q)_L$ implies the ability to compute products of single-qubit Clifford gates by a randomized NC^0-circuit with success probability at least 2/3. In particular, given a (deterministic) oracle $\mathcal{O}$ for $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)\right]$ (for any fixed function Q), this immediately yields an NC^0-circuit with additional uniformly random input bits and oracle access to $\mathcal{O}$, i.e., an $(\mathsf{NC}^0/\mathsf{rpoly})^{\mathcal{O}}$-circuit, which solves PARITY with success probability at least 2/3. Succinctly, we summarize this as

$$\left(\mathcal{O} \text{ solves } \mathsf{PSIM}\left[\mathsf{CliffP}(Q)\right]\right) \Rightarrow \left(\left(\mathsf{NC}^0/\mathsf{rpoly}\right)^{\mathcal{O}} \text{ solves } \mathsf{PARITY} \text{ with prob. } \geq \frac{2}{3}\right). \quad (2)$$

Combining (1) and (2) and using a standard derandomization argument shows that

$$\left(\mathsf{PSIM}\left[\mathsf{Telep}\right] \in \mathsf{NC}^0/\mathsf{poly}\right) \qquad \Rightarrow \qquad \left(\mathsf{PARITY} \in \mathsf{NC}^0/\mathsf{poly}\right) .$$

Thus the existence of such a circuit would contradict the folklore (trivial) result that PARITY cannot be solved by a non-uniform NC^0-circuit. Indeed, an output bit of an NC^0-circuit is influenced by at most a constant number of bits of an input instance, whereas PARITY depends on all input bits. In conclusion, this shows that there is no NC^0-circuit for the teleportation simulation problem $\mathsf{PSIM}\left[\mathsf{Telep}\right]$.

Related work

Let us briefly summarize related work. The first unconditional separation of constant-depth quantum circuits and constant-depth classical circuits with bounded fan-in gates, i.e., QNC^0 and NC^0, was achieved by Bravyi, Gosset, and König [4] for the *2D Hidden Linear Function problem*. This result was later extended to noisy quantum circuits by Bravyi, Gosset, König, and Tomamichel [5]. The latter construction uses a simpler underlying problem—the *1D Magic Square Game*. Both of these constructions are physically motivated and based on violations of Bell inequalities. In subsequent work Bene Watts, Kothari, Schaeffer, and Tal [14] achieved an even stronger separation, establishing a quantum advantage of constant-depth quantum circuits against AC^0-circuits, i.e., the family of classical constant-depth polynomial-size circuits obtained by augmenting NC^0-circuits with unbounded fan-in AND and OR gates. The problem used in [14] is the *Relaxed Parity Halving problem*. Similarly to our result, this work establishes a quantum advantage based on classical circuit complexity lower bounds as opposed to the first two [4, 5] Bell-inequality-based constructions.

In this work our aim is to give a quantum advantage proposal separating QNC^0 from NC^0. While the mentioned prior works have already established this complexity-theoretic separation, our focus is on the minimal quantum information-processing resources required to do so. We also use a novel and potentially more generally applicable proof technique. To compare our circuits to these priors works, let us discuss what geometry and gates these constructions use. The quantum circuit of [4] for

the $2D$ Hidden Linear Function problem, and the circuit of [14] for the Relaxed Parity Halving problem are geometrically $2D$-local circuits [1] composed of classically controlled Clifford gates. The $1D$ Magic Square Game problem considered in [5] is a classically controlled Clifford circuit on a chain of pairs of qubits. (This geometric locality allows for a fault-tolerant implementation by a local quantum circuit in a $3D$ geometry.) Our proposal is quite similar to that of [5], but is more directly associated with a ring of qubits rather than linear array of pairs of qubits. We note that the required resources are essentially the same despite this apparent difference. However, the use of periodic boundary conditions makes a novel argument possible, and leads to perhaps the most simple quantum circuit separating QNC^0 from NC^0.

Finally, we note that the work of Grier and Schaeffer [13] provides several quantum advantage protocols involving interaction. Unlike for typical computational problems such as those considered here (where the task is to provide a certain output given an input), corresponding advantage proposals involve a task which requires several rounds of "challenges" and "responses", respectively. These can be seen as messages passed between a prover (who aims to demonstrate quantum computational power) and a verifier. In the work [13], the quantum player (prover) can be realized by a constant-depth (classically controlled) Clifford circuit which is realized on a chain consisting of qubits, a pair of qubits, or a grid depending on the complexity-theoretic result considered: for example, a chain of qubits is used to show hardness for AC^0 with additional MOD_6 gates, and chain of pairs of qubits is used to show hardness for classical circuits with bounded fan-in gates and logarithmic depth, i.e., NC^1. We note that, while establishing several containments of classical and quantum complexity classes, this does not directly provide a separation between constant-depth quantum and constant-depth classical circuits (in the sense of separating QNC^0 from e.g., NC^0) because interactive protocols are studied (i.e., these results separate suitably defined complexity classes involving interaction.) We emphasize that, although their focus is different from ours, some of the key ideas developed in [13] are central to our derivation.

Let us briefly comment on the relation of our work to [13], where an oracle $\tilde{R}$ for the simulation of a certain interactive process is considered. The proof technique in [13] assumes that the oracle $\tilde{R}$ provides *rewind* access. It uses a reduction that does not treat $\tilde{R}$ as a black box. Instead, the classical algorithm is allowed to "rewind" the classical simulator (implementing the oracle) to an earlier state of the computation, change inputs or internal variables and continue the computation. Notice that this is a natural requirement for a classical machine (e.g. for $\mathsf{NC}^0, \mathsf{AC}^0$ or Turing machines it is implemented by the same class) but is generally impossible in a quantum computational model involving non-unitary dynamics such as measurements.

In more detail, the simulated interactive process considered in [13] is conceptually similar to a Bell experiment: in the first stage, given an L-tuple $(C_1, \ldots, C_L)$ of Clifford group elements, the stabilizer state $P(C_1, \ldots, C_L, m)(C_L \cdots C_1) |\Psi_0\rangle$ is prepared by a constant-depth quantum circuit using measurements, where the Pauli

[1] A circuit is said to be geometrically $2D$-local (resp. $1D$-local) if its underlying interaction graph is a $2D$ lattice (resp. $1D$ lattice).

"correction" $P(C_1, \ldots, C_L, m)$ is determined by the Cliffords $(C_1, \ldots, C_L)$ and the measurement results m. This first stage corresponds to preparing an initial entangled state for a Bell experiment. In the second stage of the process, an input is provided according to which a suitable Pauli measurement is applied to the prepared state. Without loss of generality, the input takes the form of a Clifford unitary D, and the executed measurement is the von Neumann (i.e., projective) measurement using the basis $\{D|z\rangle\}_z$. This second step corresponds to the execution of a Bell measurement on the prepared state. The work [13] considers simulation problems associated with different interactive protocols of this kind for k-qubit Clifford circuits, embedded into so-called graph state simulation problems.

Rewind access to an oracle $\tilde{R}$ simulating such an interactive process means that oracle queries to $\tilde{R}$ can be repeated in such a way that it yields an identical output m for the first step. In other words, the classical machine can be rewound to the beginning of the second stage. In contrast, reproducing this behavior using a quantum protocol would require post-selecting on fixed measurement outcomes occurring with exponentially small probability. The ability of a classical simulator to rewind is at the root of the hardness results of [13] concerning interactive quantum advantage.

In contrast to [13], we do not need to assume rewind access to the classical oracle $\tilde{R}$. This is achieved by exploiting the internal structure of the classical simulator: we consider oracles realized by circuits and provide a non-black-box reduction. As we discuss below (see Section 3.2), however, our arguments heavily rely on some of the techniques of [13] developed in the interactive scenario.

A more significant distinction between our work and that of [13] is the fact that the computational problem we consider does not require interaction, i.e., two distinct stages. To illustrate this difference, consider the following simple example of a two-stage (interactive) quantum advantage in a problem involving two players Alice and Bob. The problem is commonly known as the CHSH game [15]. Let us assume that Alice and Bob start out with no classical or quantum correlations. In stage 1, Alice and Bob are allowed to communicate arbitrarily. In stage 2, Alice and Bob are not allowed to communicate anymore, and they receive random bits x and y, respectively. They are asked to output bits a respectively b such that $a \cdot b = x + y \pmod 2$. It is well-known that Alice and Bob can win with higher success probability than two classical players, if they can exchange quantum messages at stage 1. It is easy to see that a similarly simple quantum advantage cannot be observed when there is only a single stage (i.e., where Alice and Bob obtain x and y at the very beginning). This example indicates that the challenge of establishing a quantum advantage heavily depends on how many (interactive) rounds are considered. While our work removes the need for interaction, an even more extreme example is the problem of sampling: Here a quantum advantage has recently been established for problems "without input" [16] (see also [17]).

Outlook

Due to the simplicity of the circuit Telep, our work provides a starting point for a fault-tolerant $3D$-local quantum advantage scheme based on the techniques of [5].

A natural open question is whether the Telep circuit also provides a quantum advantage over AC^0, with a similar separation as the (up to the best of our knowledge) strongest known unconditional result [14]. More generally, it is unclear whether or not quantum advantage schemes against AC^0 are feasible based on geometrically $1D$-local quantum circuits only.

2 Locality properties of classical circuits

In this section we identify certain no-signaling properties every NC^0-circuit necessarily has. These are crucial in our non-black-box reduction from the simulation problem $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)]$ to the problem $\mathsf{PSIM}\,[\mathsf{Telep}]$ in Section 3.

In Sections 2.1 and 2.2 we define lightcones and locality of functions and NC^0-circuits. In Section 2.3 we use combinatorial arguments to prove no-signaling properties of NC^0.

2.1 Locality of functions

In the following, we consider functions

$$\begin{array}{rccc} f: & (\{0,1\}^k)^n & \to & (\{0,1\}^r)^s \\ & I & \mapsto & f(I) = (f_1(I), \ldots, f_s(I)) \end{array} \tag{3}$$

taking an "input"-n-tuple I of k-bit strings to an "output"-s-tuple $f(I)$ of r-bit strings. We often write $I = (I_1, \ldots, I_n)$ for the input variables and $O = (O_1, \ldots, O_s)$ for the output variables, where $O_m = f_m(I)$ for $m \in \{1, \ldots, s\}$. We call each of the k-bit-strings making up an input I a block. That is, the j-th block of the input consists of the k binary variables $I_j = (I_j^{(1)}, \ldots, I_j^{(k)})$ of the argument, where $j \in \{1, \ldots, n\}$. Similarly, the m-th block of the output is associated with $O_m = (O_m^{(1)}, \ldots, O_m^{(r)})$ for $m \in \{1, \ldots, s\}$.

While a function of the form (3) can be regarded as a function

$$f : \{0,1\}^N \to \{0,1\}^M \tag{4}$$

with $N = k \cdot n$ and $M = r \cdot s$, i.e., as a function taking N-bit strings to M-bit strings, the block-structure (3) will also be important in the following. To distinguish between the two interpretations of a function f of the form (3), we refer to (4) as the function f with the block-structure ignored. Note also that for $k = r = 1$, there is no difference between these two notions. In particular, the following definitions apply to any function $f : \{0,1\}^N \to \{0,1\}^M$ taking N-bit strings to M-bit strings, for any $N, M \in \mathbb{N}$.

For a function f of the form (3) we say that the k-th output block O_k is influenced by the j-th input I_j if and only if the output block O_k depends non-trivially on the value of the j-th input block, i.e., if there is a choice of $(I_1, \ldots, I_{j-1}, I_{j+1}, \ldots, I_n) \in (\{0,1\}^k)^{n-1}$ and $J, J' \in \{0,1\}^k$ such that

$$f_k(I_1, \ldots, I_{j-1}, J, I_{j+1}, \ldots, I_n) \neq f_k(I_1, \ldots, I_{j-1}, J', I_{j+1}, \ldots, I_n)\ .$$

In the following, we will often identify input- and output-blocks with their indices. Correspondingly, we define the forward lightcone $\mathcal{L}^{\rightarrow}(j)$ of the j-th input block I_j as the set of all output blocks that are influenced by it, i.e.,

$$\mathcal{L}^{\rightarrow}(j) := \{k \in [s] \mid O_k \text{ is influenced by } I_j\} \ .$$

Similarly, the backward lightcone $\mathcal{L}^{\leftarrow}(k)$ of the k-th output block O_k is the set of all input blocks that influence it,

$$\mathcal{L}^{\leftarrow}(k) := \{j \in [n] \mid O_k \text{ is influenced by } I_j\} \ .$$

A function f is said to be ℓ-local if the size of all backward lightcones is bounded by ℓ, i.e., if

$$|\mathcal{L}^{\leftarrow}(k)| \leq \ell \qquad \text{for all} \qquad k \in [s] \ .$$

Lightcones naturally extend to a subset $\mathcal{J}$ of either input and output blocks: for $\mathcal{J}_{\text{in}} \subseteq [n]$ and $\mathcal{J}_{\text{out}} \subseteq [s]$, we set

$$\mathcal{L}^{\rightarrow}(\mathcal{J}_{\text{in}}) = \bigcup_{j \in \mathcal{J}_{\text{in}}} \mathcal{L}^{\rightarrow}(j), \qquad \text{and} \qquad \mathcal{L}^{\leftarrow}(\mathcal{J}_{\text{out}}) = \bigcup_{j \in \mathcal{J}_{\text{out}}} \mathcal{L}^{\leftarrow}(j) \ .$$

For a subset $\{I_j\}_{j \in \mathcal{J}}$ of input blocks (specified by a subset $\mathcal{J} = \{j_1 < \cdots < j_\ell\} \subseteq [n]$), we say that an output block O_k is determined by $\{I_j\}_{j \in \mathcal{J}}$ if it is only a function of the corresponding input variables, i.e., if there is a function $g : (\{0,1\}^k)^\ell \to \{0,1\}^r$ such that

$$f_k(I) = g(I_{j_1}, \ldots, I_{j_\ell}) \qquad \text{for all} \qquad I = (I_1, \ldots, I_n) \in (\{0,1\}^k)^n \ .$$

2.2 Locality of circuits

Note that for circuit $\mathsf{C}\colon \{0,1\}^N \to \{0,1\}^M$ of depth d, the maximal fan-in ν of the gates in the circuit sets an upper bound on the size of all backward lightcones: we have

$$|\mathcal{L}^{\leftarrow}(j)| \leq \nu^d \qquad \text{for all} \qquad j \in [M] \ .$$

In particular, a depth-d circuit with gates of fan-in upper bounded by ν is ℓ-local with

$$\ell \leq \nu^d \ .$$

Similar considerations apply to every "block" function $\mathsf{C}\colon (\{0,1\}^k)^n \to (\{0,1\}^r)^s$ that is realized by a circuit C. If each gate has fan-in upper bounded by ν (i.e., has at most ν bits as input), then each output bit of the circuit can be influenced by at most ν^d input bits. Thus each output bit can be influenced by at most ν^d blocks. Since every input block consists of k bits, it follows that

$$|\mathcal{L}^{\leftarrow}(j)| \leq k\nu^d \qquad \text{for all} \qquad j \in [s] \ , \tag{5}$$

i.e., the circuit is ℓ-(block-)local with $\ell \leq k\nu^d$.

In the following, we often consider families of functions

$$\{f_n : (\{0,1\}^k)^n \to (\{0,1\}^r)^{s(n)}\}_{n \in \mathbb{N}}$$

where k, r are constant and $s(n) = \mathsf{poly}(n)$ is polynomial in n. Eq. (5) implies that if such a function family is represented by an NC^0-circuit, then it has constant locality, i.e., it is $O(1)$-local.

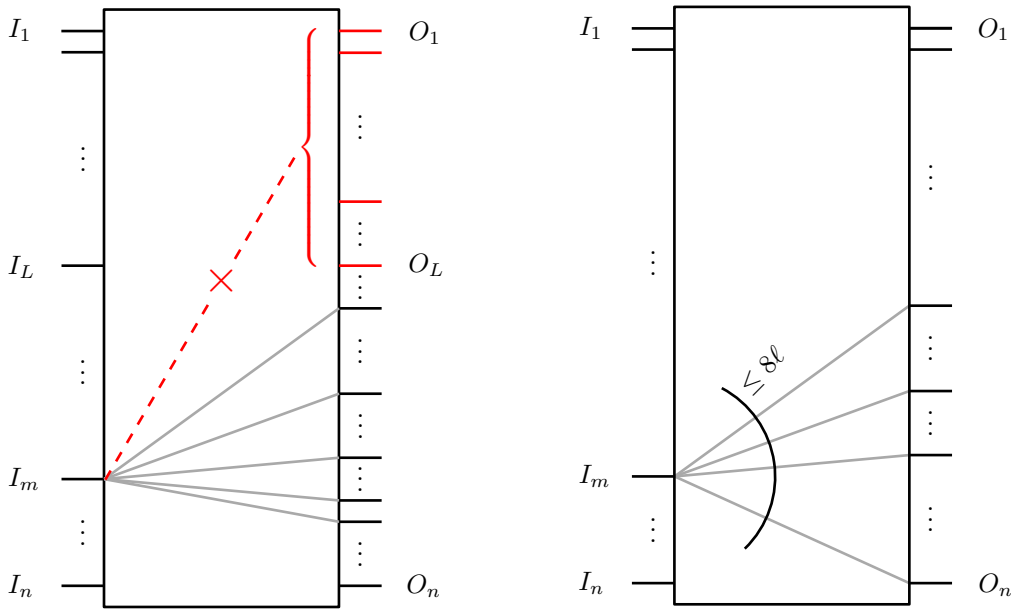
2.3 No-signaling features of NC^0-circuits

We establish no-signaling properties for a specific class of NC^0-circuits that have the same number of input and output blocks. These are exactly the type of circuits we want to analyze for the $\mathsf{PSIM}\,[\mathsf{Telep}]$ problem. Let $\mathsf{C}\colon (\{0,1\}^k)^n \to (\{0,1\}^r)^n$ be a function. That is, C takes as input $(I_1,\ldots,I_n) \in (\{0,1\}^k)^n$ and outputs $(O_1,\ldots,O_n) \in (\{0,1\}^r)^n$. We assume that the circuit C has (block-)locality ℓ.

Let $L \in [n]$. We say that $m \in [n]$ is L-good if and only if

$$\mathcal{L}^{\rightarrow}(m) \cap [L] = \emptyset \ .$$

That is, an input index $m \in [n]$ is L-good if and only if the input block I_m does not influence the first L output blocks $(O_1,\ldots,O_L)$. We illustrate the L-good property in Figure 3a. Let us denote by $\mathsf{Good}(L) \subseteq [n]$ the subset of $m \in [n]$ such that m is L-good. We also set $\mathsf{Bad}(L) := [n]\backslash\mathsf{Good}(L)$.



(a) An illustration of an L-good $m \in [n]$, i.e., the forward lightcone of the input block I_m has no intersection with the outputs $O_1,\ldots,O_L$.

(b) An illustration of an element $m \in [n]$ with limited signaling, i.e., the size of forward lightcone of the input block I_m is upper-bounded by 8ℓ.

Figure 3: Structural properties of local functions.

In a similar fashion, let us say that the m-th input block I_m has limited signaling if and only if

$$|\mathcal{L}^{\rightarrow}(m)| \leq 8\ell \ ,$$

i.e., if and only if I_m influences at most 8ℓ output blocks. (The choice of factor 8 is for convenience only.) We depict the limited signaling property in Figure 3b. Let us denote the set of limited-signaling input indices by $\mathsf{LimitedSignaling} \subseteq [n]$.

Our goal in this section is to show that for a suitably chosen value of L, there are indices $m \in [n]$ such that

(i) m is L-good,

(ii) m has the limited signaling property, and

(iii) $m > n/2$ (i.e., m belongs to the "second half" of input indices).

To establish this statement, we first show a lower bound on the number of indices $m \in [n]$ satisfying (i) and (iii). the following notation will be convenient: consider a subset $\mathcal{G} \subseteq [n]$ of input block indices. We will often partition such a set as

$$\mathcal{G} = \mathcal{G}_{\leq} \cup \mathcal{G}_{>} \qquad \text{where} \qquad \begin{array}{rcl} \mathcal{G}_{\leq} & = & \{m \in \mathcal{G} \mid m \leq n/2\} \\ \mathcal{G}_{>} & = & \{m \in \mathcal{G} \mid m > n/2\} \ . \end{array}$$

Lemma 2.1. *Set*

$$L := \left\lfloor \frac{n}{4\ell} \right\rfloor \ .$$

Then

$$|\mathsf{Good}(L)_{>}| \geq \frac{n}{4} \ .$$

Proof. By the definition of the set $\mathsf{Bad}(L)_{>}$, we have

$$\mathcal{L}^{\rightarrow}(m) \cap [L] \neq \emptyset \qquad \text{for every} \qquad m \in \mathsf{Bad}(L)_{>} \ . \tag{6}$$

By the definitions of the forward and backward lightcones, condition (6) is equivalent to

$$m \in \mathcal{L}^{\leftarrow}([L]) \qquad \text{for every} \qquad m \in \mathsf{Bad}(L)_{>} \ ,$$

that is,

$$\mathsf{Bad}(L)_{>} \subseteq \mathcal{L}^{\leftarrow}([L]) \ .$$

It follows that

$$|\mathsf{Bad}(L)_{>}| \leq |\mathcal{L}^{\leftarrow}([L])| \ . \tag{7}$$

By the locality-ℓ assumption on C, we have

$$|\mathcal{L}^{\leftarrow}(j)| \leq \ell \qquad \text{for all} \qquad j \in [n]$$

and thus

$$|\mathcal{L}^{\leftarrow}([L])| \leq L\ell \ . \tag{8}$$

Combining (8) with (7) yields

$$|\mathsf{Bad}(L)_{>}| \leq L\ell \ .$$

or equivalently

$$|\mathsf{Good}(L)_{>}| \geq \lceil n/2 \rceil - L\ell \geq n/4$$

by the definition of L. □

We next derive a lower bound on the number of indices $m \in [n]$ satisfying (ii) and (iii). The statement and the proof of the following result are analogous to those of [4, Claim 5].

Lemma 2.2. *We have*

$$\left|\mathsf{LimitedSignaling}_{>}\right| \geq \frac{3n}{8} \ .$$

Proof. Consider a bipartite graph $G = (V_I \cup V_O, E)$ with two sets of vertices

$$V_I = \{v_i \mid i \in [n]_{>}\}$$
$$V_O = \{w_o \mid o \in [n]\} \ ,$$

where $(v_i, w_o) \in V_I \times V_O$ are connected by an edge if and only if

$$i \in \mathcal{L}^{\leftarrow}(o) \ . \tag{9}$$

Observe that (9) is equivalent to

$$o \in \mathcal{L}^{\rightarrow}(i) \ . \tag{10}$$

Using (9), the set of edges of this graph can be upper bounded by[2]

$$|E| \leq \sum_{o=1}^{n} |\mathcal{L}^{\leftarrow}(o)| \leq n\ell \ . \tag{11}$$

On the other hand, using the definition of the forward lightcone and (10), we have that for any $m \in \{\lfloor n/2+1 \rfloor, \ldots, n\} \backslash \mathsf{LimitedSignaling}$, the subset $\{w_o \mid o \in \mathcal{L}^{\rightarrow}(m)\} \subseteq V_O$ has more than 8ℓ neighbors in V_I. This gives the lower bound

$$\begin{aligned} |E| &\geq |\{\lfloor n/2+1 \rfloor, \ldots, n\} \setminus \mathsf{LimitedSignaling}_{>}| \cdot 8\ell \\ &= (\lceil n/2 \rceil - |\mathsf{LimitedSignaling}_{>}|) \cdot 8\ell \ . \end{aligned} \tag{12}$$

Combining (12) with (11) gives the claim. □

Finally, we can combine both properties to show that there is a large number of indices $m \in [n]$ with $m > n/2$ (property (iii)) that are L-good (cf. (i)) and have the limited-signaling property (cf. (ii)).

Theorem 2.3. *Let* C *be a circuit with locality upper bounded by* ℓ*. Set*

$$L := \left\lfloor \frac{n}{4\ell} \right\rfloor \ .$$

Then for sufficiently large n *there is some input index* $m \in \{\lfloor n/2+1 \rfloor, \ldots, n\}$ *that is both* L*-good and has limited signaling, i.e.,*

$$\mathcal{L}^{\rightarrow}(m) \cap [L] = \emptyset$$

and

$$|\mathcal{L}^{\rightarrow}(m)| \leq 8\ell \ .$$

[2]Here and below we write $\prod_{j=0}^{k} x_j = x_k \cdots x_0$ for non-commutative variables $x_0, \ldots, x_k$.

Proof. This follows immediately by combining Lemma 2.1 with Lemma 2.2. Indeed, by Lemma 2.1 we have

$$|\mathsf{Good}(L)_{>}| \geq n/4 \ ,$$

and by Lemma 2.2 we have

$$|\mathsf{LimitedSignaling}_{>}| \geq 3n/8 \ .$$

Since both $\mathsf{Good}(L)_{>}$ and $\mathsf{LimitedSignaling}_{>}$ are subsets of $[n]_{>}$, we also have

$$|\mathsf{Good}(L)_{>} \cup \mathsf{LimitedSignaling}_{>}| \leq |\,[n]_{>}| = \lceil n/2 \rceil \ .$$

Consequently, we have

$$\begin{aligned} &|\mathsf{Good}(L)_{>} \cap \mathsf{LimitedSignaling}_{>}| \\ &\qquad = |\mathsf{Good}(L)_{>}| + |\mathsf{LimitedSignaling}_{>}| - |\mathsf{Good}(L)_{>} \cup \mathsf{LimitedSignaling}_{>}| \\ &\qquad \geq n/4 + 3n/8 - \lceil n/2 \rceil \geq n/8 - 1 \ . \end{aligned}$$

This implies the claim. □

3 Quantum advantage against NC^0

Our goal is to show that $\mathsf{PSIM}\,[\mathsf{Telep}] \notin \mathsf{NC}^0/\mathsf{poly}$, i.e., constant-depth quantum circuits have an advantage over (even non-uniform) NC^0 for the possibilistic simulation of the gate-teleportation circuit. Our proof is by contradiction: we show the implication

$$\Big(\mathsf{PSIM}\,[\mathsf{Telep}] \in \mathsf{NC}^0/\mathsf{poly}\Big) \qquad \Rightarrow \qquad \Big(\mathsf{PARITY} \in \mathsf{NC}^0/\mathsf{poly}\Big) \ . \tag{13}$$

This implication shows that $\mathsf{PSIM}\,[\mathsf{Telep}] \notin \mathsf{NC}^0/\mathsf{poly}$ since we would otherwise get a contradiction to the trivial result $\mathsf{PARITY} \notin \mathsf{NC}^0/\mathsf{poly}$.

In Section 3.1 we prove that any NC^0 algorithm for the $\mathsf{PSIM}\,[\mathsf{Telep}_n]$ problem can be used to construct a non-uniform NC^0 algorithm for the $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$ problem for $L = \Theta(n)$ and a fixed function Q. In Section 3.2 we follow and specialize the techniques of [13] to obtain a randomized NC^0 algorithm that uses an oracle for $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$ to learn some information about the stabilizer group of states of the form $(I \otimes C_L \cdots C_1)\,|\Phi\rangle$, where $|\Phi\rangle$ is a Bell state. Finally, in Section 3.3 we use this learning problem to solve PARITY. This establishes (13).

Since the teleportation circuit Telep trivially solves the $\mathsf{PSIM}\,[\mathsf{Telep}]$ problem, this completes the proof of Theorem 1.1, i.e., establishes a quantum advantage over NC^0.

3.1 Relating $\mathsf{PSIM}\,[\mathsf{Telep}]$ to $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)]$

Here we argue that an NC^0-circuit C which possibilistically simulates the gate-teleportation circuit Telep_n gives rise to a non-uniform NC^0-circuit (using oracle access to C) which simulates $\mathsf{CliffP}(Q)_L$ for $L = \Theta(n)$ and a suitably chosen function Q. That is, we show the implication

$$\Big(\mathsf{PSIM}\,[\mathsf{Telep}] \in \mathsf{NC}^0/\mathsf{poly}\Big) \qquad \Rightarrow \qquad \Big(\exists Q\colon \mathsf{PSIM}\,[\mathsf{CliffP}(Q)] \in \mathsf{NC}^0/\mathsf{poly}\Big) \ .$$

3.1.1 Motivation

To motivate the construction, recall what the effect of the (concurrent) Bell measurements in a gate-teleportation circuit is: it essentially amounts to the application of a product $C_n \cdots C_1$ of Clifford group elements to one half of a maximally entangled state, but the resulting state $(I \otimes C_n \cdots C_1) |\Phi\rangle$ is corrupted by a Pauli error $\widetilde{Q}$ determined by the measurement outcomes. In standard uses of gate teleportation (such as in error correction), this Pauli error is computed and applied in a correction step. In contrast, the circuit Telep_n does not incorporate such a correction step. We note that the computation of $\widetilde{Q}$ from the measurement outcomes is not feasible using, e.g., an NC^0 or an AC^0-circuit.

Our construction avoids the necessity of computing Pauli corrections such as $\widetilde{Q}$ by exploiting the fact that the correction function Q in the definition of the circuit $\mathsf{CliffP}(Q)_L$ can be chosen arbitrarily (without computational complexity considerations). Very roughly, the circuit Telep_n emulates the circuit $\mathsf{CliffP}(\widetilde{Q})_L$ for suitably chosen L and $\widetilde{Q}$.

In more detail, let us examine the relationship between the quantum circuits for Telep_n and $\mathsf{CliffP}(Q)_L$. The output distribution of Telep_n is given by

$$p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C_0, \ldots, C_{n-1}) = \frac{1}{4^n} \left|\mathrm{tr}\left(P_{n-1}C_{n-1} \cdots P_1 C_1 P_0 C_0\right)\right|^2 \ , \quad (14)$$

which we show at the end of this section, see Eq. (21) and bellow.

Eq. (14) can be written as[3]

$$\begin{aligned} p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C_0, \ldots, C_{n-1}) &= \frac{1}{4^n} \left|\mathrm{tr}\left((P_{n-1}C_{n-1})(P_{n-2}C_{n-2} \cdots P_1 C_1 P_0 C_0)\right)\right|^2 \\ &= \frac{1}{4^n} \left|\mathrm{tr}\left((P_{n-1}C_{n-1})\widetilde{Q}C_{n-2} \cdots C_0\right)\right|^2 \ , \end{aligned} \quad (15)$$

where the Pauli operator

$$\widetilde{Q} = \widetilde{Q}(C_0, \ldots, C_{n-2}, P_0, P_1, \ldots, P_{n-2})$$

is obtained by commuting all the Pauli operators in the product $P_{n-2}C_{n-2} \cdots P_0 C_0$ to the left, i.e.,

$$\widetilde{Q} \prod_{i=0}^{n-2} C_i = P_{n-2}C_{n-2} \cdots P_0 C_0 \ .$$

On the other hand, the output distribution of $\mathsf{CliffP}(Q)_L$ is

$$p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D) = \left|\langle\Phi| \left(P \otimes I\right)\left(I \otimes \left(DQ(C_1, \ldots, C_L)(C_L \cdots C_1)\right)\right) |\Phi\rangle\right|^2 \quad (16)$$

which can be written as

$$p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D) = \frac{1}{4} \left|\mathrm{tr}\left(PDQC_L \cdots C_1\right)\right|^2 \ , \text{ where } Q = Q(C_1, \ldots, C_L) \ . \quad (17)$$

[3]The derivation is similar to the one described for (14), see (21) and bellow.

Note that we use $Q = Q(C_1, \ldots, C_L)$ to indicate that Q is a function of $(C_1, \ldots, C_L)$ and of no other arguments. Comparing (15) and (17), setting $n := L + 1$ and

$$C'_j := \begin{cases} C_{j+1} & \text{for} \quad j = 0, \ldots, n-2 \\ D & \text{for} \quad j = n-1 \end{cases} \tag{18}$$

we have the implication

$$p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C'_0, \ldots, C'_{n-1}) > 0 \quad \Rightarrow \quad \tilde{p}(P \mid C_1, \ldots, C_L, D) > 0 \ , \tag{19}$$

where we have written P for P_{n-1}, and where the distribution $\tilde{p}$ is defined as in (17) but with the Pauli operator $Q(C_1, \ldots, C_L)$ replaced by $\widetilde{Q}(C_1, \ldots, C_L, P_0, \ldots, P_{L-1})$, that is,

$$\tilde{p}(P \mid C_1, \ldots, C_L, D) := \frac{1}{4} \left| \mathrm{tr}\left(PD\widetilde{Q}C_L \cdots C_1 \right) \right|^2 , \tag{20}$$

where we used a shorthand notation $\widetilde{Q} = \widetilde{Q}(C_1, \ldots, C_L, P_0, \ldots, P_{L-1})$. Eq. (19) and comparing (20) with (17) suggests that a possibilistic simulation of Telep_n can be used to possibilistically simulate $\mathsf{CliffP}(Q)_L$ for a suitably chosen function Q. However, the fact that $\widetilde{Q}$ also depends on $P_0, \ldots, P_{L-1}$ needs to be addressed—the expression (20) differs from the output distribution (17) of $\mathsf{CliffP}(Q)_L$. This is where no-signaling considerations enter the argument.

It remains to show the technical claim of Eq. (14) that we restate and prove here.

$$p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C_0, \ldots, C_{n-1}) = \frac{1}{4^n} \left| \mathrm{tr}\left(P_{n-1}C_{n-1} \cdots P_1 C_1 P_0 C_0 \right) \right|^2 \tag{21}$$

The correctness can be easily verified using the tensor-network formalism. Alternatively, the identity can be obtained as follows. Let $\pi_N : (\mathbb{C}^2)^{\otimes N} \to (\mathbb{C}^2)^{\otimes N}$ be the unitary which cyclically "left" permutes tensor factors, i.e., π_N is defined as $\pi_N(|\Psi_0\rangle \otimes \cdots \otimes |\Psi_{N-1}\rangle) = |\Psi_1\rangle \otimes |\Psi_2\rangle \otimes \cdots \otimes |\Psi_{N-1}\rangle \otimes |\Psi_0\rangle$ for $|\Psi_j\rangle \in \mathbb{C}^2$, $j \in \mathbb{Z}_N$, and linearly extended to all of $(\mathbb{C}^2)^{\otimes N}$. This satisfies

$$\mathrm{tr}(\pi_N(A_0 \otimes \cdots \otimes A_{N-1})) = \mathrm{tr}(A_0 \cdots A_{N-1}) \tag{22}$$

for any N-tuple of operators $(A_0, \ldots, A_{N-1})$ on $\mathbb{C}^2$. Also observe that

$$\mathcal{E}(A) := \frac{1}{4} \sum_{k,\ell \in \{0,1\}} X^k Z^\ell A X^k Z^\ell = \frac{1}{2} A^T \qquad \text{for any operator } A \text{ on } \mathbb{C}^2 \ . \tag{23}$$

We use that the projection onto the maximally entangled state $|\Phi\rangle$ can be written as

$$|\Phi\rangle\langle\Phi| = \frac{1}{4}(I \otimes I + X \otimes X - Y \otimes Y + Z \otimes Z) = \frac{1}{4} \sum_{a,b \in \{0,1\}} (X^a Z^b) \otimes (X^a Z^b) \tag{24}$$

The probability $p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C_0, \ldots, C_{n-1})$ of interest is the squared overlap of the state $((I \otimes C_0) \otimes \cdots \otimes (I \otimes C_{n-1}))\, |\Phi\rangle^{\otimes n}$ and the state

$$\pi_{2n}^\dagger \left((P_0 \otimes I) \otimes \cdots \otimes (P_{n-1} \otimes I) \right) |\Phi\rangle^{\otimes n} \ ,$$

see Fig. 1. The latter is equivalent to

$$\pi_{2n}^{\dagger}\left((P_0\otimes I)\otimes\cdots\otimes(P_{n-1}\otimes I)\right)\pi_{2n}\pi_{2n}^{\dagger}\left|\Phi\right\rangle^{\otimes n} = \left((I\otimes P_0)\otimes\cdots\otimes(I\otimes P_{n-1})\right)\pi_{2n}^{\dagger}\left|\Phi\right\rangle^{\otimes n} \ .$$

Thus the overlap of interest can be expressed as

$$\begin{aligned}
&\left|\mathrm{tr}\left(\left((I\otimes P_0C_0)\otimes\cdots\otimes(I\otimes P_{n-1}C_{n-1})\right)\left|\Phi\right\rangle\!\left\langle\Phi\right|^{\otimes n}\pi_{2n}\right)\right|^2 \\
&= \left|\frac{1}{4^n}\sum_{\vec{a},\vec{b}\in\{0,1\}^n}\mathrm{tr}\left(\pi_{2n}\Big((X^{a_0}Z^{b_0}\otimes P_0C_0X^{a_0}Z^{b_0})\otimes\ldots\right.\right. \\
&\qquad\left.\left.\cdots\otimes(X^{a_{n-1}}Z^{b_{n-1}}\otimes P_{n-1}C_{n-1}X^{a_{n-1}}Z^{b_{n-1}})\Big)\right)\right|^2 \quad \text{by (24)} \\
&= \left|\mathrm{tr}\left(\mathcal{E}(P_0C_0)\cdots\mathcal{E}(P_{n-1}C_{n-1})\right)\right|^2 \qquad \text{by (22)} \\
&= \frac{1}{2^{2n}}\left|\mathrm{tr}\left((P_0C_0)^T\cdots(P_{n-1}C_{n-1})^T\right)\right|^2 \qquad \text{by (23)} \\
&= \frac{1}{2^{2n}}\left|\mathrm{tr}\left(((P_{n-1}C_{n-1})\cdots(P_0C_0))^T\right)\right|^2 \ .
\end{aligned}$$

The claim (21) (cf. (14)) now follows because the transpose is trace-preserving, i.e., $\mathrm{tr}(A^T)=\mathrm{tr}(A)$.

3.1.2 An algorithm for $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)\right]$

Suppose $\mathsf{C}\colon\mathcal{C}^n\to\mathcal{P}^n$ is a deterministic oracle (i.e., a function) that solves the problem $\mathsf{PSIM}\left[\mathsf{Telep}_n\right]$. In this section, we argue that query access to C allows an NC^0-circuit to solve the problem $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ for some fixed function Q, assuming that C has certain locality properties we detail below. Moreover, this property matches exactly what we established in Section 2.3 for NC^0-circuits. The main idea behind the reduction is identical to that explained in the previous section: we embed instances of $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ into $\mathsf{PSIM}\left[\mathsf{Telep}_n\right]$. However, we need to deal with some amount of signaling since the $\widetilde{Q}$ in Eq. (20) could be indirectly influenced by D through the Pauli operators $(P_0,\ldots,P_{L-1})$. Therefore, contrary to the choice of embedding (18), our embeddings use longer sequences (i.e., large values of n), essentially "padding" with identity operators.

To define these embeddings, suppose that there are L and m such that $L\le m<n$, which satisfy properties illustrated in Figure 4.

We embed an instance $(C_1,\ldots,C_L,D)\in\mathcal{C}^{L+1}$ of the problem $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ (where $Q:\mathcal{C}^L\to\mathcal{P}$ can be any fixed function) into an instance $(C'_0,\ldots,C'_{n-1})\in\mathcal{C}^n$ of $\mathsf{PSIM}\left[\mathsf{Telep}_n\right]$ as follows: we set

$$C'_j=\begin{cases}C_{j+1} & \text{for} \quad j=0,\ldots,L-1\ ,\\ I & \text{for} \quad j\ge L,\ j\ne m\ ,\\ D & \text{for} \quad j=m\ .\end{cases} \tag{25}$$

To examine the result of this embedding, we need to consider the output probabilities for an embedded instance: we have

$$p^{\mathsf{Telep}_n}(P_0,\ldots,P_{n-1}\,|\,C'_0,\ldots,C'_{n-1})=\frac{1}{4^n}\left|\mathrm{tr}\left(\left(\prod_{r=m}^{n-1}P_r\right)D\left(\prod_{s=L}^{m-1}P_s\right)\left(\prod_{t=1}^{L}(P_{t-1}C_t)\right)\right)\right|^2. \tag{26}$$

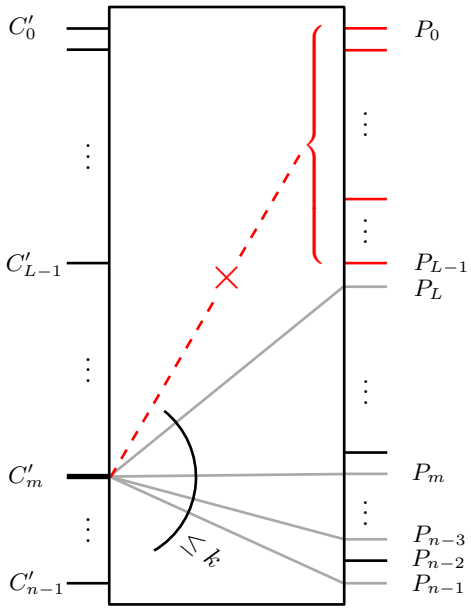


Figure 4: An illustration of the no-signaling assumption on the oracle for the PSIM $[\mathsf{Telep}_n]$ problem. We assume the input $C'_m := D$ cannot signal to outputs $P_0, \ldots, P_{L-1}$ and can only signal to a constant k number of outputs with indices from the set $\mathcal{J}$. We depict the outputs influenced by the input C'_m in grey.

We display the trace defining the probability $p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1}|C'_0, \ldots, C'_{n-1})$ as well as the same trace after we inserted variables from the embedding (i.e. the right-hand side of Eq. (26)) in tensor-network diagrammatic representation in Figure 5.

We need to relate this to the output probability $p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D)$ of the circuit $\mathsf{CliffP}(Q)_L$, see Eq. (17). It is easy to see (using commutation relations) that expression (26) can indeed be related to $p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D)$ for suitable choices of Q and P. The less obvious a priori is that, when going from $(P_0, \ldots, P_{n-1})$ to (Q, P), the dependence on the Paulis $(P_{j_1}, \ldots, P_{j_k})$ for a fixed subset $\mathcal{J} := \{j_1 < \cdots < j_k\}$ of positions can be limited to P (whereas Q can be chosen not to depend on these Paulis). This is a consequence of the cyclic structure of the circuit Telep_n. We formalize it in the following Lemma 3.1.

Lemma 3.1. *Let $L \leq m < n$ be integers. Let*

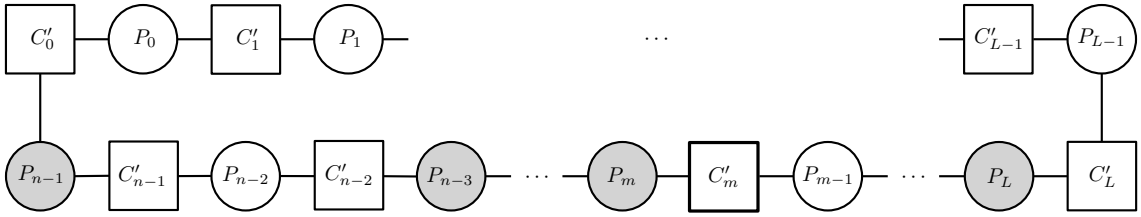
$$\mathcal{J} = \{j_1 < \cdots < j_k\}$$

be a subset of $\{L, L+1, \ldots, n-1\}$. Then there are two functions

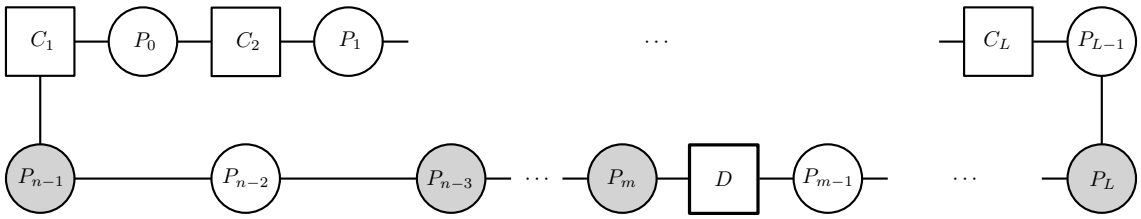
$$\begin{aligned} P: &\quad \mathcal{P}^k \times \mathcal{C} &\to &\quad \mathcal{P} \\ Q: &\quad \mathcal{P}^{n-k} \times \mathcal{C}^L &\to &\quad \mathcal{P} \end{aligned}$$

such that

$$\left|\mathrm{tr}\left(\left(\prod_{r=m}^{n-1} P_r\right) D \left(\prod_{s=L}^{m-1} P_s\right) \left(\prod_{t=1}^{L}(P_{t-1}C_t)\right)\right)\right| = \left|\mathrm{tr}\left(PDQ \prod_{j=1}^{L} C_j\right)\right| \qquad (27)$$



(a) Tensor-network representation of the trace defining $p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1}|C'_0, \ldots, C'_{n-1})$, from Eq. (14). We depict Clifford unitaries by squares and Pauli operators by circles. The output Pauli operators influenced by the input C'_m are illustrated in grey.



(b) The trace defining the output probability distribution of Telep_n after we insert the inputs from Eq. (25) (also later in Algorithm 2 after the lines (2)–(8)).

Figure 5: Tensor-network diagrammatic representation of (a) the trace that defines $p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1}|C'_0, \ldots, C'_{n-1})$, from Eq. (14), and (b) the same trace after we inserted variables from the embedding (25). Note that this figure illustrates a case where $m > L$, however, our arguments also apply for $m = L$ without further modifications.

for all $(P_0, \ldots, P_{n-1}) \in \mathcal{P}^n$ *and* $(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$, *where*

$$P = P\left((P_j)_{j\in\mathcal{J}}, D\right) \tag{28}$$

$$Q = Q\left((P_j)_{j\notin\mathcal{J}}, C_1, \ldots, C_L\right) \ . \tag{29}$$

Here we write $(P_j)_{j\in\mathcal{J}}$ *for* $(P_{j_1}, \ldots, P_{j_k})$ *(and similarly for* $(P_j)_{j\notin\mathcal{J}}$*).*

We illustrate each step in the proof of Lemma 3.1 in the tensor-network diagrammatic representation for clarity. The left-hand side of Eq. (27) is shown in Figure 5b.

Proof. Let $(P_0, \ldots, P_{n-1}) \in \mathcal{P}^n$ and $(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$ be arbitrary. Define

$$R := \left(\prod_{r=m}^{n-1} P_r\right) \qquad \text{and} \qquad S := \left(\prod_{s=L}^{m-1} P_s\right) \ .$$

Then the quantity of interest is equal to

$$\left|\mathrm{tr}\left(\left(\prod_{r=m}^{n-1} P_r\right) D \left(\prod_{s=L}^{m-1} P_s\right)\left(\prod_{t=1}^{L}(P_{t-1}C_t)\right)\right)\right| = \left|\mathrm{tr}\left(RDS\left(\prod_{t=1}^{L}(P_{t-1}C_t)\right)\right)\right| \ . \tag{30}$$

Since Pauli operators either commute or anticommute, we can factor the products of Pauli operators R and S into factors P_j with $j \in \mathcal{J}$, and factors P_j with $j \in \mathbb{Z}_n \backslash \mathcal{J}$. That is, we have

$$R = \pm Q'P' \qquad \text{and} \qquad S = \pm P''Q'' \ , \tag{31}$$

where

$$\begin{array}{rcll} P' & := & \prod_{r=m}^{n-1} & P_r^{\delta_r} \\ Q' & := & \prod_{r=m}^{n-1} & P_r^{1-\delta_r} \end{array} \qquad \text{and} \qquad \begin{array}{rcll} P'' & := & \prod_{s=L}^{m-1} & P_s^{\delta_s} \\ Q'' & := & \prod_{s=L}^{m-1} & P_s^{1-\delta_s} \end{array} \tag{32}$$

where $\delta_r \in \{0,1\}$ indicates whether or not $r \in \mathcal{J}$, that is,

$$\delta_r := \begin{cases} 1 & \text{if } r \in \mathcal{J} \\ 0 & \text{otherwise .} \end{cases}$$

Inserting the factorization (31) into expression (30) we have

$$\begin{aligned} \pm \operatorname{tr}\left(RDS\left(\prod_{t=1}^{L}(P_{t-1}C_t)\right)\right) &= \operatorname{tr}\left((Q'P')D(P''Q'')\left(\prod_{k=1}^{L}(P_{k-1}C_k)\right)\right) \\ &= \operatorname{tr}\left(Q'P'DP''Q''Q'''\left(\prod_{j=1}^{L}C_j\right)\right) . \end{aligned}$$

Here we commuted the Pauli operators $P_0, \ldots, P_{L-1}$ in the product $\prod_{t=1}^{L}(P_{t-1}C_t)$ to the left, combining them into a Pauli operator $Q''' \in \mathcal{P}$ defined by

$$Q''' := \left(\prod_{r=1}^{L}(P_{r-1}C_r)\right)\left(\prod_{r=1}^{L}C_r\right)^{-1} .$$

At this point, we use the cyclicity of the trace to move Q' to the right of the product in the argument, obtaining

$$\begin{aligned} \operatorname{tr}\left(Q'P'DP''Q''Q'''\left(\prod_{j=1}^{L}C_j\right)\right) &= \operatorname{tr}\left(P'DP''Q''Q'''\left(\prod_{j=1}^{L}C_j\right)Q'\right) \\ &= \operatorname{tr}\left(P'DP''Q\left(\prod_{j=1}^{L}C_j\right)\right) , \end{aligned} \tag{33}$$

where we defined

$$Q := Q''Q'''\left(\prod_{j=1}^{L}C_j\right)Q'\left(\prod_{j=1}^{L}C_j\right)^{-1} . \tag{34}$$

Using the cyclicity of the trace to move Q' to the right (instead of commuting it past $P'DP''$) here ensures that the resulting operator Q has no dependence on D. Indeed, it is clear from the definition of Q', Q'', Q''' and (34) that Q is a function of $(P_j)_{j\in\mathcal{J}}$ and $(C_1, \ldots, C_L)$ only as claimed, see Eq. (29). We illustrate these steps (commuting Paulis) in Figure 6.

We can rewrite (33) as

$$\operatorname{tr}\left(P'DP''Q\left(\prod_{j=1}^{L}C_j\right)\right) = \operatorname{tr}\left(PDQ\left(\prod_{j=1}^{L}C_j\right)\right) \tag{35}$$

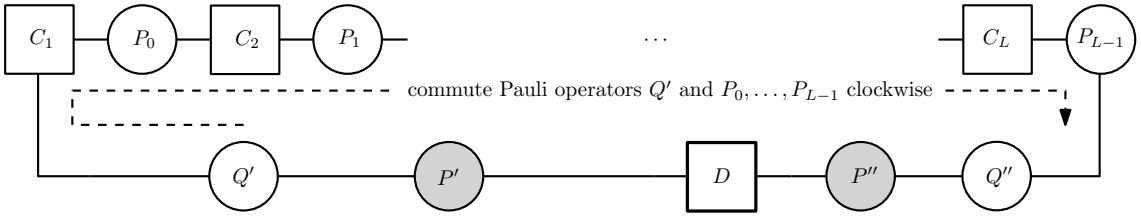


Figure 6: The use of the cyclic property of trace to commute Pauli operators not influenced by the input D in a way that does not create additional dependence of the resulting Pauli operator on D. This figure illustrates the steps from Eq. (30) to Eq. (33), after the Pauli operators composing Q', P', P'', Q'' have been respectively regrouped.

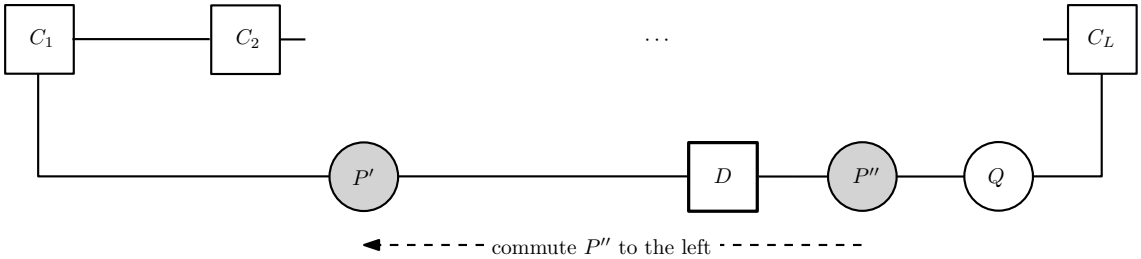


Figure 7: In this step we commute all Pauli operators depending on D to the left of it. This illustrates the Eq. (35) and the step (4) of Algorithm 1.

by defining

$$P := P'DP''D^{-1} \ . \tag{36}$$

It is easy to check using the definition of P' and P'' and (36) that P is a function of only $(P_j)_{j\in\mathcal{J}}$ and D as claimed (see Eq. (28)). We illustrate the computation of P in Figure 7.

This proves the lemma since we have shown that (27) is satisfied with these definitions, and we can assume without loss of generality that $P, Q \in \mathcal{P}$ (as (27) includes absolute values). The resulting trace is depicted in Figure 8. □

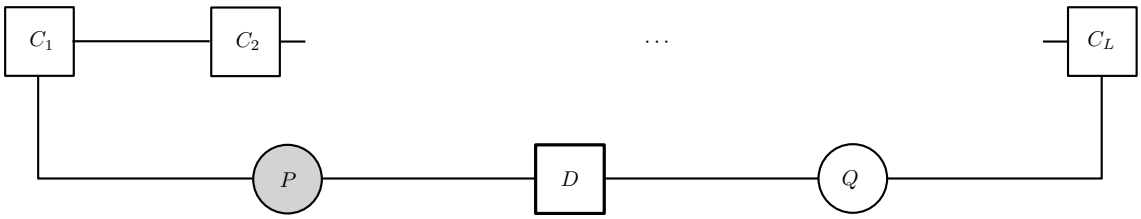


Figure 8: The resulting trace defining the probability $p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D)$ and therefore the solution to PSIM [CliffP(Q)] problem. This illustrates the trace from Eq. (17) to which we arrive at Eq. (35).

In the following, we will use that P only depends on the subsets of Paulis $(P_j)_{j\in\mathcal{J}}$ with indices belonging to $\mathcal{J}$ and the Clifford D as expressed by Eq. (28). In fact, the Pauli P can easily be computed: the proof of Lemma 3.1 also shows that the

Algorithm 1 Algorithm defining the function $\mathcal{A}_{\mathcal{J},m} : \mathcal{P}^n \times \mathcal{C} \to \mathcal{P}$. Here $\mathcal{J} \subseteq \mathbb{Z}_n$ is a subset and $m \in \mathbb{Z}_n$.

function $\mathcal{A}_{\mathcal{J},m}(P_0, \dots, P_{n-1}, D)$
  $P' \leftarrow \prod_{p=m}^{n-1} P_p^{\delta_p}$ where $\delta_p = 1$ if $p \in \mathcal{J}$ and $\delta_p = 0$ otherwise
  $P'' \leftarrow \prod_{q=L}^{m-1} P_q^{\delta_q}$
  $P \leftarrow P' D P'' D^{-1}$.
  Ignore (remove) the global phase to ensure that $P \in \mathcal{P}$.
  return P

algorithm $\mathcal{A}_{\mathcal{J},m}$ (Algorithm 1) computes this function, i.e., we have

$$\mathcal{A}_{\mathcal{J},m}(P_0, \dots, P_{n-1}, D) = P\left((P_j)_{j\in\mathcal{J}}, D\right) \quad \text{for all} \quad (P_0, \dots, P_{n-1}) \in \mathcal{P}^n \text{ and } D \in \mathcal{C}\,. \tag{37}$$

Importantly, the algorithm $\mathcal{A}_{\mathcal{J},m}$ can be implemented efficiently.

Lemma 3.2. *Let k be a constant, let $L = \mathsf{poly}(n) < n$ and m be such that $L \leq m < n$. Let $\mathcal{J} \subseteq \mathbb{Z}_n$ be of size $|\mathcal{J}| = k$. Then Algorithm 1 for computing the function $\mathcal{A}_{\mathcal{J},m} : \mathcal{P}^n \times \mathcal{C} \to \mathcal{P}$ can be implemented by an NC^0-circuit.*

Proof. This follows immediately from equations (32) and (36). Since k is constant, the corresponding products can be computed by an NC^0-circuit, see Lemma 3.1. □

Algorithm 2 The algorithm $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$, which takes as input an $L+1$-tuple $(C_1, \dots, C_L, D) \in \mathcal{C}^{L+1}$ of Clifford group elements. It uses an oracle C for the $\mathsf{PSIM}\,[\mathsf{Telep}_n]$ problem. For a suitable choice of $(L, m, \mathcal{J})$, the output P of $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$ is a valid solution to $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$ for a certain function $Q : \mathcal{C}^L \to \mathcal{P}$, see Lemma 3.3.

function $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}(C_1, \dots, C_L, D)$
  for $j = 0, \dots, L-1$ **do**
    $C'_j \leftarrow C_{j+1}$
  for $j = L, \dots, n-1$ **do**
    if $j \neq m$ **then**
      $C'_j \leftarrow I$
    else
      $C'_j \leftarrow D$
  $(P_0, \dots, P_{n-1}) \leftarrow \mathsf{C}(C'_0, \dots, C'_{n-1})$
  $P \leftarrow \mathcal{A}_{\mathcal{J},m}(P_0, \dots, P_{n-1}, D)$
  return P

Our main result of this section combines Lemma 3.1 and Lemma 3.2 to show that an NC^0 circuit C for $\mathsf{PSIM}\,[\mathsf{Telep}_n]$ can be used to solve $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$. To arrive at this result, we make a suitable choice of L, m and the subset $\mathcal{J} \subseteq \mathbb{Z}_n$ depending on the oracle C in a non-black-box manner. More precisely, we use the no-signaling properties of C established in Section 2.3. The resulting reduction is

therefore non-uniform due to three parameters $(L, m, \mathcal{J})$ that must be given as advice. The following theorem formalizes this and shows that a suitable choice always exists.

Lemma 3.3. *Let $\mathsf{C}\colon \mathcal{C}^n \to \mathcal{P}^n$ be a (possibly non-uniform) NC^0-circuit for the $\mathsf{PSIM}\,[\mathsf{Telep}_n]$ problem. Then there are $(L, m, \mathcal{J})$ with $L = \Theta(n)$, $L \leq m < n$ and $\mathcal{J} \subseteq \mathbb{Z}_n$, as well as a function $Q : \mathcal{C}^L \to \mathcal{P}$ such that the algorithm $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$ (Algorithm 2) has the following properties:*

(i) Given an arbitrary input $(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$, the output produced by the algorithm $P = \mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}(C_1, \ldots, C_L, D)$ is a valid solution to $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$ associated with the instance $(C_1, \ldots, C_L, D)$.

(ii) The algorithm $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$ can be implemented by an NC^0-circuit making one oracle query to C.

Proof. Suppose that $\mathsf{C}\colon \mathcal{C}^n \to \mathcal{P}^n$ is an NC^0-circuit. Let us denote input- and output-variables of C by $(C'_0, \ldots, C'_{n-1})$ and $(P_0, \ldots, P_{n-1})$, respectively, i.e., we write

$$(P_0, \ldots, P_{n-1}) = \mathsf{C}(C'_0, \ldots, C'_{n-1}) \ .$$

We note that by definition, the algorithm $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$ uses the embedding (25), i.e., for an instance $(C_1, \ldots, C_L, D)$ of $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$, it calls the oracle C with $(C'_0, \ldots, C'_{n-1})$ defined by (25).

As a constant-depth circuit with bounded fan-in gates, the circuit C is ℓ-local with locality bounded by $\ell = O(1)$, see Eq. 2.2. Let $L := \lfloor n/(4\ell) \rfloor$. Then $L = \Theta(n)$ by definition, and, as shown in Theorem 2.3 there is an integer $m \in \{\lfloor n/2+1 \rfloor, \ldots, n-1\}$ and a subset $\mathcal{J} \subseteq \{L, \ldots, n-1\}$ such that

(a) the collection of output variables $(P_j)_{j=0,\ldots,L-1}$ does not depend on the input variable C'_m, i.e., $\mathcal{L}^{\rightarrow}(m) \cap \{0, \ldots, L-1\} = \emptyset$, and

(b) the input variable C'_m influences at most 8ℓ output variables $\{P_j\}_{j\in\mathcal{J}}$, i.e., we have $|\mathcal{J}| \leq 8\ell = O(1)$ for $\mathcal{J} = \mathcal{L}^{\rightarrow}(m)$.

Statement (b) implies that the function $\mathcal{A}_{\mathcal{J},m}$ defined by Algorithm (1) can be computed by an NC^0-circuit from $(P_0, \ldots, P_{n-1})$, see Lemma 3.2. Claim (ii) follows from this.

It remains to prove claim (i). Note that $\mathcal{A}_{\mathcal{J},m}$ computes the function

$$P\left((P_j)_{j\in\mathcal{J}}, D\right)$$

appearing in Lemma 3.1, see Eq. (37). Furthermore, the Pauli

$$Q = Q\left((P_j)_{j\notin\mathcal{J}}, C_1, \ldots, C_L\right) \tag{38}$$

in Lemma 3.1 does not depend on C'_m by (a).

Consider an output variable P_j with $j \notin \mathcal{J}$. By definition, P_j is a function of $\left(C'_j\right)_{j\in\mathbb{Z}_n\setminus\{m\}}$. Because the embedding (25) fixes to the identity all but the first L

Cliffords $(C'_0, \ldots, C'_{L-1}) = (C_1, \ldots, C_L)$ and the m-th Clifford $C'_m = D$, we conclude that P_j (for $j \notin \mathcal{J}$) is a function of $(C_1, \ldots, C_L)$ only. Combining this with (38), we conclude that

$$Q = Q(C_1, \ldots, C_L)$$

is a function of $(C_1, \ldots, C_L)$ only as required in the definition of $\mathsf{CliffP}(Q)_L$.

With Lemma 3.1, as well as expressions (26) and (17) for the probabilities $p^{\mathsf{Telep}_n}(P_0, \ldots, P_{n-1} \mid C'_0, \ldots, C'_{n-1})$ and $p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D)$, we conclude that $\mathcal{B}^{\mathsf{C}}_{L,m,\mathcal{J}}$ indeed produces a valid solution for the problem $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$, as claimed. □

In the next section, we will use a deterministic oracle for $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ to learn stabilizers of states $I \otimes (C_L \cdots C_1) |\Phi\rangle$.

3.2 From $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ to learning stabilizers of $(I \otimes (C_L \cdots C_1)) |\Phi\rangle$

By definition, the output P of the circuit $\mathsf{CliffP}(Q)_L$ (see Fig. 2) can be interpreted as outputting the result of measuring the state

$$(I \otimes (C_L \cdots C_1)) |\Phi\rangle \tag{39}$$

in a basis of maximally entangled states obtained by rotating the Bell basis by $I \otimes Q^\dagger D^\dagger$. Here the effect of the Pauli $Q = Q(C_1, \ldots, C_L)$ is simply to possibly flip the measurement result (i.e., it can equivalently be achieved by classical post-processing). By choosing different inputs $(C_1, \ldots, C_L, D)$ with $(C_1, \ldots, C_L)$ fixed and varying over $D \in \mathcal{C}$, access to a deterministic oracle $\mathcal{O}$ for the problem $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ can thus be used by an algorithm to simulate the effect of measuring the state (39) in different bases of stabilizer states. A protocol using such measurement results to determine the state (39) (or learn some information about it) therefore allows to determine (or learn some information about) the product $C_L \cdots C_1$. As we will discus in Section 3.3, such information can in turn be used to solve the word problem for the Clifford group on a single qubit.

In this section, we give a protocol for learning the stabilizers of the state (39) using a deterministic oracle $\mathcal{O}$ for $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$. More precisely, the corresponding algorithm learns two distinct stabilizer generators of the state (39) up to signs, see Theorem 3.4 below. In other words, it finds two distinct (non-identity) Pauli operators that have expectation 1 or -1 on this state. Allowing for this remaining sign ambiguity has the key advantage that the function $Q(C_1, \ldots, C_L)$ does not need to be evaluated in the protocol.

The protocol borrows heavily from the pioneering work [13]. Indeed, Theorem 3.4 below is obtained by specializing the results of [13] to the problem at hand.

3.2.1 On the need for possibilistic state tomography

We emphasize that the problem of determining stabilizers of the state (39) using an oracle $\mathcal{O}$ for $\mathsf{PSIM}\left[\mathsf{CliffP}(Q)_L\right]$ cannot directly be solved using standard techniques

for (stabilizer) quantum state tomography. This is because of the possibilistic nature of the simulation provided by $\mathcal{O}$. Accordingly, we refer to the problem under consideration as possibilistic tomography.

To clarify the special difficulties associated with possibilistic tomography, consider a setting where several copies of an unknown stabilizer state $|\Psi\rangle \in (\mathbb{C}^2)^{\otimes 2}$ can be generated/are available. This is the setting of standard tomography. Here a set $\{S_1, S_2\}$ of generators of the stabilizer group (up to a phase) of $|\Psi\rangle$ can efficiently be determined by a sequence of measurements with respect to different Pauli bases, see e.g., [18, 19].

One of the key achievements of [13] is to extend this stabilizer state tomography to the possibilistic setting, where the process of measuring a state $|\Psi\rangle$ in a Pauli basis is replaced by an oracle $\mathcal{O}$ simulating the measurement. Remarkably, possibilistic simulation is sufficient, i.e., the simulated measurement results produced by $\mathcal{O}$ only need to lie in the support of the output distribution defined by Born's rule for measurements. While this notion of simulation has been referred to as possibilistic in [2], the authors of [13] express this by saying that $\mathcal{O}$ can be adversarial.

3.2.2 Possibilistic state tomography from the magic square

The procedure of [13] reduces possibilistic stabilizer state tomography to possibilistic circuit simulation. Here we restate their result in a simplified and specialized form suitable for our purposes:

Theorem 3.4 (Adapted from [13])**.** *Let $Q : \mathcal{C}^L \to \mathcal{P}$ be arbitrary. Suppose $\mathcal{O}$ is a deterministic oracle (i.e., a function) solving $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$. There is an algorithm $\mathcal{A}^{\mathcal{O}}$ with access to $\mathcal{O}$ with the following properties:*

(i) $\mathcal{A}^{\mathcal{O}}$ takes as input a sequence $(C_1, \ldots, C_L) \in \mathcal{C}^L$.

(ii) The algorithm $\mathcal{A}^{\mathcal{O}}$ either reports failure or outputs a pair (S_1, S_2) of two distinct two-qubit Pauli operators of the form $S_j = P_A^j \otimes P_B^j$ with $P_A^j, P_B^j \in \mathcal{P} \setminus \{I\}$, for $j = 1, 2$. The latter (successful) event happens with probability at least $2/3$.

(iii) When the algorithm succeeds, each operator S_j for $j = 1, 2$ has expectation value $\{\pm 1\}$ in the state $(I \otimes (C_L \cdots C_1))\,|\Phi\rangle$.

(iv) The algorithm $\mathcal{A}^{\mathcal{O}}$ makes $O(1)$ queries to $\mathcal{O}$.

(v) The algorithm $\mathcal{A}^{\mathcal{O}}$ is realizable by an NC^0-circuit which uses uniformly random bits and has oracle access to $\mathcal{O}$.

In the remainder of this section, we outline the proof of Theorem 3.4. We first show that the oracle $\mathcal{O}$ can be used to learn a non-stabilizer M of the state $(I \otimes (C_L \cdots C_1))\,|\Phi\rangle$. Here a non-stabilizer of a stabilizer state $|\Psi\rangle$ is a Pauli operator M such that $\langle\Psi|\,M\,|\Psi\rangle = 0$.

Lemma 3.5. *Let $\mathcal{O}$ be a deterministic oracle for $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$. Then there is an NC^0-algorithm $\mathcal{N}^{\mathcal{O}}$ which takes as input an L-tuple $(C_1, \ldots, C_L) \in \mathcal{C}^L$,*

(i) makes 6 queries to $\mathcal{O}$, and

$$\begin{array}{ccc} X \otimes X & Y \otimes Y & Z \otimes Z \\ Y \otimes Z & Z \otimes X & X \otimes Y \\ Z \otimes Y & X \otimes Z & Y \otimes X \end{array}$$

Figure 9: The matrix $(P_{j,k})_{j,k=1}^{3}$ of Pauli observables forming the magic square.

(ii) outputs a non-stabilizer M of the state $(I \otimes (C_L \cdots C_1))\,|\Phi\rangle$.

The proof of Lemma 3.5 exploits the fact that measurements of the observables of the magic square illustrated in Fig. 9 cannot be explained by a non-contextual model.

Proof. Each row $j \in [3]$ of the magic square in Fig. 9 consists of three (dependent) commuting Pauli observables $(P_{j,1}, P_{j,2}, P_{j,3})$ which can be jointly measured, i.e., give rise to a measurement with outcomes $(m_{j,1}, m_{j,2}, m_{j,3}) \in \{\pm 1\}^3$ corresponding to the eigenvalues of the corresponding operators. Similarly, each column $k \in [3]$ of the magic square defines a measurement with three (dependent) measurement outcomes $(m'_{1,k}, m'_{2,k}, m'_{3,k})$ associated with observables $(P_{1,k}, P_{2,k}, P_{3,k})$. In total, the magic square thus defines six measurements, one for each row and column.

Suppose we have six copies of a two-qubit stabilizer state $|\Psi\rangle$. Using three copies to perform the measurements associated with each row gives a matrix $m = (m_{j,k}) \in \mathsf{Mat}_{3\times 3}(\{\pm 1\})$ of measurement outcomes. Using the remaining three copies to perform the measurements associated which each column similarly gives a matrix $m' = (m'_{j,k}) \in \mathsf{Mat}_{3\times 3}(\{\pm 1\})$. Because of the "magic square relations"

$$\begin{array}{lll} P_{j,1}P_{j,2}P_{j,3} = -I & \text{for every} & j \in [3] \\ P_{1,k}P_{2,k}P_{3,k} = I & \text{for every} & k \in [3] \end{array}$$

satisfied by the observables $\{P_{j,k}\}_{j,k}$ in the magic square, we have

$$m_{j,1}m_{j,2}m_{j,3} = -1 \qquad \text{for every} \qquad j \in [3] \tag{40}$$

$$m'_{1,k}m'_{2,k}m'_{3,k} = 1 \qquad \text{for every} \qquad k \in [3]\ . \tag{41}$$

Because a magic square, i.e., a $\{\pm 1\}$-valued matrix satisfying both (40) and (41) does not exist, there must be a pair $(j,k) \in [3]^2$ such that $m_{j,k} \neq m'_{j,k}$. The corresponding Pauli operator $P_{j,k}$ thus leads to a measurement result $+1$ and a measurement result -1, depending on the context. This means that $P_{j,k}$ must be a non-stabilizer of $|\Psi\rangle$. In summary, a non-stabilizer M of a two-qubit stabilizer state $|\Psi\rangle$ can be obtained with certainty by performing Pauli measurements on six copies of $|\Psi\rangle$.

We can apply this idea to construct an algorithm $\mathcal{N}^{\mathcal{O}}$ as specified in the lemma using an oracle $\mathcal{O}$ for $\mathsf{PSIM}\,[\mathsf{CliffP}(Q)_L]$. Recall that on input $(C_1, \ldots, C_L, D) \in \mathcal{C}^{L+1}$, the oracle $\mathcal{O}$ produces a Pauli operator P occurring with non-zero probability $p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D)$ in the output distribution of $\mathsf{CliffP}(Q)_L$. We note that this output probability can be written as

$$p^{\mathsf{CliffP}(Q)_L}(P \mid C_1, \ldots, C_L, D) = \left|\langle \Phi_P(D)| \left(I \otimes (QC_L \cdots C_1)\right)|\Phi\rangle\right|^2\ ,$$

where $Q = Q(C_1, \ldots, C_L)$ is a function of $C_1, \ldots, C_L$ and of no other arguments, and where

$$|\Phi_P(D)\rangle := (I \otimes D^\dagger)(P \otimes I)\,|\Phi\rangle \qquad \text{for} \qquad P \in \mathcal{P}$$

defines a rotated Bell basis $(I \otimes D^\dagger)\mathcal{B} := \{|\Phi_P(D)\rangle\}_{P\in\mathcal{P}}$ (with $\mathcal{B} := \{|\Phi_P(I)\rangle\}_{P\in\mathcal{P}}$ being the standard Bell basis), see expression (16). In other words, this output distribution is identical to that obtained when measuring the state

$$I \otimes (QC_L \cdots C_1)\,|\Phi\rangle \tag{42}$$

using the von Neumann measurement with the orthonormal basis $(I \otimes D^\dagger)\mathcal{B}$.

We argue below that by suitably choosing D and applying the von Neumann measurement with basis $(I \otimes D^\dagger)\mathcal{B}$, we can realize any row- and every column-measurement associated with the magic square. This means that query access to $\mathcal{O}$ with $(C_1, \ldots, C_L)$ fixed and $D \in \mathcal{C}$ arbitrary immediately translates to the (possibilistic) simulation of magic square-measurements in the state (42). In particular, six queries to $\mathcal{O}$ of the form $(C_1, \ldots, C_L, D)$ with $D \in \mathcal{C}$ reveal a non-stabilizer M of the state (42). The claim (ii) follows from this using the following simple fact: if M is a non-stabilizer of a stabilizer state $(I \otimes Q)\,|\Psi\rangle$ and $Q \in \mathcal{P}$ is arbitrary, then M is also a non-stabilizer of the state $|\Psi\rangle$.

It remains to argue that the measurement of any row or column of operators in the magic square is equivalent to a von Neumann measurement in a rotated Bell basis $(I \otimes D^\dagger)\mathcal{B}$ for a suitably chosen $D \in \mathcal{C}$. To this end, observe that for any two single-qubit Cliffords $D_A, D_B \in \mathcal{C}$ and any Pauli $P' \in \mathcal{P}$, we have the identity

$$\begin{aligned}
(D_A \otimes D_B)(P' \otimes I)\,|\Phi\rangle &= (P \otimes I)(D_A \otimes D_B)\,|\Phi\rangle \qquad \text{where} \quad P := D_A P' D_A^\dagger \quad (43)\\
&= (P \otimes I)(I \otimes D_B D_A^T)\,|\Phi\rangle \\
&= (I \otimes D^\dagger)(P \otimes I)\,|\Phi\rangle \qquad \text{where} \quad D := (D_B D_A^T)^\dagger \quad (44)\\
&= |\Phi_P(D)\rangle \ .
\end{aligned}$$

This is because the Bell state $|\Phi\rangle$ satisfies $(I \otimes L)\,|\Phi\rangle = (L^T \otimes I)\,|\Phi\rangle$ for any operator L on $\mathbb{C}^2$. Definitions of P, D from Eq. (43) and (44) imply that—up to a relabeling of the measurement results as expressed by (43)—the von Neumann measurement in the rotated Bell basis $(I \otimes D^\dagger)\mathcal{B}$ is equivalent to a von Neumann measurement in the basis $(D_A \otimes D_B)\mathcal{B}$ consisting of the rotated Bell states

$$(D_A \otimes D_B)(P \otimes I)\,|\Phi\rangle \qquad \text{for} \qquad P \in \mathcal{P} \ .$$

Elements of the Bell basis $\mathcal{B}$ are stabilizer states having stabilizer generators of the form

$$(\pm X_A \otimes X_B, \pm Z_A \otimes Z_B) \ .$$

It follows that elements of the rotated Bell basis $(D_A \otimes D_B)\mathcal{B}$ have stabilizer generators of the form

$$(\pm X'_A \otimes X'_B, \pm Z'_A \otimes Z'_B) \qquad \text{where} \qquad \begin{cases} X'_A &= D_A X_A D_A^\dagger \\ Z'_A &= D_A Z_A D_A^\dagger \\ X'_B &= D_B X_B D_B^\dagger \\ Z'_B &= D_B Z_B D_B^\dagger \ . \end{cases} \tag{45}$$

A von Neumann measurement in the rotated Bell basis $(D_A \otimes D_B)\mathcal{B}$—which, as shown above, is equivalent to a measurement in the basis $(I \otimes D^\dagger)\mathcal{B}$—reveals the basis state, that is, the pair of eigenvalues of $X'_A \otimes X'_B$ and $Z'_A \otimes Z'_B$, respectively.

We note that any pair $(P_1 \otimes P_2, Q_1 \otimes Q_2)$ of Pauli observables with $P_1, P_2, Q_1, Q_2 \in \{X, Y, Z\}$ and $P_j \neq Q_j$ for $j = 1, 2$ can be written in the form (45) for suitably chosen Clifford operators D_A, D_B. It is straightforward to check that every row and column of the magic square in Fig. 9 contains a pair $(X'_A \otimes X'_B, Z'_A \otimes Z'_B)$ of two-qubit operators satisfying this condition (the third operator is equal to $\pm(X'_A \otimes X'_B)(Z'_A \otimes Z'_B)$ and its eigenvalue can be obtained from the measurement results of the other two operators). Thus each row or column of the magic square can be measured by using a von Neumann measurement in a rotated Bell basis of the form $(I \otimes D^\dagger)\mathcal{B}$, as claimed.

The fact that the algorithm $\mathcal{N}^{\mathcal{O}}$ is realizable by an NC^0-circuit is obvious since it only manipulates a constant number of bits. □

The algorithm $\mathcal{N}^{\mathcal{O}}$ of Lemma 3.5 only provides a single non-stabilizer. However, using a randomization procedure based on Kilian randomization [13, 20] and using $\mathcal{N}^{\mathcal{O}}$ as a subroutine, it is possible to construct an algorithm which produces a uniformly random non-stabilizer of $(I \otimes C_L \cdots C_1)\,|\Phi\rangle$. Let us denote $\mathcal{N}^{\mathcal{O}}$ by $\mathcal{R}$.

Theorem 3.6 (Kilian-randomization-based algorithm)**.** *There is a randomized* NC^0*-algorithm* $\mathcal{N}^{\mathcal{R}}_{uniform}$*, which uses polynomially many uniformly random bits. Given an input* $(C_1, \ldots, C_L) \in \mathcal{C}^L$ *the algorithm* $\mathcal{N}^{\mathcal{R}}_{uniform}$

(i) makes 1 *query to* $\mathcal{R}$*, and*

(ii) outputs a uniformly random non-stabilizer of $(I \otimes C_L \cdots C_1)\,|\Phi\rangle$*.*

We defer the description of $\mathcal{N}^{\mathcal{R}}_{\text{uniform}}$ and the proof of Theorem 3.6 to Appendix A.

An algorithm such as the one described in Theorem 3.6 which provides a uniformly chosen non-stabilizer of a stabilizer state $|\Psi\rangle$ can be run repeatedly to (eventually) learn the stabilizer group of $|\Psi\rangle$. For a sufficiently large but constant number of calls to such an algorithm, a pair (S_1, S_2) of Pauli operators with the desired property can be computed with probability at least 2/3.

Theorem 3.4 follows from these arguments, see Theorem A.4 in Appendix A for details.

3.3 Quantum advantage against NC^0 from PSIM [Telep]

In this section, we finish the argument for our result, establishing a quantum advantage for the PSIM [Telep] problem against (non-uniform) NC^0.

In Section 3.1, we showed that a possibilistic (non-uniform) NC^0 simulator for Telep circuits implies a possibilistic $\mathsf{NC}^0/\mathsf{poly}$ simulator for CliffP(Q) for some fixed function Q:

$$\Big(\mathsf{PSIM}\,[\mathsf{Telep}] \in \mathsf{NC}^0/\mathsf{poly}\Big) \quad \Rightarrow \quad \Big(\exists Q\colon \mathsf{PSIM}\,[\mathsf{CliffP}(Q)] \in \mathsf{NC}^0/\mathsf{poly}\Big)\ . \tag{46}$$

We will use the possibilistic simulation for the CliffP(Q) problem together with possibilistic state tomography, Section 3.2 (specialized from [13]), to solve a certain

word problem for a fixed group with success probability at least 2/3. The word problem we consider is as hard as PARITY.

This will lead to the following implication:

$$(\mathcal{O} \text{ solves } \mathsf{PSIM}\,[\mathsf{CliffP}(Q)]) \;\Rightarrow\; \left((\mathsf{NC}^0/\mathsf{rpoly})^{\mathcal{O}} \text{ solves } \mathsf{PARITY} \text{ with prob. } \geq 2/3\right) .$$

Combining it with Eq. (46) and using a standard derandomization argument, we obtain

$$\left(\mathsf{PSIM}\,[\mathsf{Telep}] \in \mathsf{NC}^0/\mathsf{poly}\right) \qquad \Rightarrow \qquad \left(\mathsf{PARITY} \in \mathsf{NC}^0/\mathsf{poly}\right) .$$

This contradicts that $\mathsf{PARITY} \notin \mathsf{NC}^0/\mathsf{poly}$, since the output of an $\mathsf{NC}^0/\mathsf{poly}$ circuit depends on a constant number of bits of an input instance. Therefore, we obtain $\mathsf{PSIM}\,[\mathsf{Telep}] \notin \mathsf{NC}^0/\mathsf{poly}$.

We relate PARITY and the word problem for a fixed group in Section 3.3.1 and use it to complete the proof of quantum advantage from PSIM [Telep] in Section 3.3.2.

3.3.1 PARITY as a word problem for the Clifford group

The word problem for groups is defined as follows. Let G be a group with identity element $\mathsf{e} \in G$. The word problem for the group G is to decide for an input sequence of group elements $(g_1, \ldots, g_L) \in G^L$ whether the product $g_L \cdots g_1$ equals the identity element e. In our scenario we use the promise variant which promises that the product is either identity or some other chosen element g. We denote this problem $\mathsf{WP}(G, g)$. A sequence of group elements $(g_1, \ldots, g_L)$ is a yes-instance if $g_L \cdots g_1 = \mathsf{e}$ and a no-instance if $g_L \cdots g_1 = g$.

Consider the word problem $\mathsf{WP}(\mathcal{C}, H)$, where $\mathcal{C}$ is the group of one-qubit Clifford unitaries with the standard product operation and H the Hadamard unitary. We show that PARITY reduces to this problem. Let $x \in \{0,1\}^L$ be a bit string. We will show how to use $\mathsf{WP}(\mathcal{C}, H)$ with the input length L to compute PARITY of the input x. Set

$$g_i := H^{x_i} \qquad \text{for all} \qquad i \in [n] \; . \tag{47}$$

Since $H^2 = I$, the product

$$g_L \cdots g_1 = H^{\mathsf{PARITY}(x)} = \begin{cases} I & \text{if } x \text{ has even parity, and} \\ H & \text{if } x \text{ has odd parity} \; . \end{cases}$$

Hence we can use $\mathsf{WP}(\mathcal{C}, H)$ to solve PARITY.

We note that this discussion may appear overly detailed to simply compute the parity. However, one may use other (Clifford) groups and corresponding word problems to address further computational tasks, along the lines of [13], which we expand upon in Section 3.3.3. In particular, using for example two qubits and corresponding gate-teleportation circuits, one can construct circuits solving word problems associated with S_6, related to Barrington's theorem [21]. However, because of the use of lightcone-type arguments, naive applications of our reasoning cannot provide quantum advantage beyond NC^0.

3.3.2 PSIM [Telep] cannot be in NC^0

In the following lemma we use a deterministic oracle for the PSIM [CliffP(Q)] problem together with the possibilistic state tomography from Section 3.2 to solve PARITY.

Lemma 3.7. *Let $\mathcal{O}$ be a deterministic oracle for the* PSIM [CliffP(Q)] *problem. There is an* NC^0*-circuit with oracle access to $\mathcal{O}$ which uses uniformly random bits, makes $O(1)$ oracle queries and solves the* PARITY *problem with success probability at least $2/3$.*

Proof. Let $x \in \{0,1\}^L$ be an input instance of PARITY_L. Set, as in Eq. (47), $(C_1, \ldots, C_L)$ where $C_i := H^{x_i}$. Again $C_L \cdots C_1 = H^{\mathsf{PARITY}(x)}$.

We will use the learning procedure with the oracle for the PSIM [CliffP$(Q)_L$] problem established in Theorem 3.4 to learn the stabilizer group (up to the sign factor) of the state

$$(I \otimes C_L \cdots C_1) |\Phi\rangle = \left(I \otimes H^{\mathsf{PARITY}(x)}\right) |\Phi\rangle \ .$$

This theorem tells us that there is a randomized NC^0 algorithm that makes $O(1)$ queries to the oracle $\mathcal{O}$ and either returns a pair $(S_1, S_2) \in (\mathcal{P} \otimes \mathcal{P})^2$ of two distinct (non-identity) Pauli operators or reports failure. If the algorithm returns (S_1, S_2), which happens with probability at least $2/3$, each S_i has expectation value ± 1 in the state $\left(I \otimes H^{\mathsf{PARITY}(x)}\right) |\Phi\rangle$. Let us list the possible outcomes for the two options, i.e., when x has even respectively odd parity:

(i) If x has even parity the algorithm learns the stabilizer group of the state $\left(I \otimes H^{\mathsf{PARITY}(x)}\right) |\Phi\rangle = |\Phi\rangle$. It is straightforward to check that the valid outputs (S_1, S_2) of the learning algorithm are two distinct elements of the set $S_{\text{even}} := \{X \otimes X, Z \otimes Z, Y \otimes Y\}$.

(ii) If x has odd parity the algorithm learns the stabilizers of the state $(I \otimes H) |\Phi\rangle$. The output (S_1, S_2) are two distinct elements of the set $S_{\text{odd}} := \{X \otimes Z, Z \otimes X, Y \otimes Y\}$.

Since $|S_{\text{even}} \cap S_{\text{odd}}| = 1$, we can determine the parity of x from the output (S_1, S_2). Therefore we can compute $\mathsf{PARITY}(x)$ with probability at least $2/3$. This completes the proof. □

We use this result to complete our argument and show that there cannot be a (non-uniform) NC^0-circuit for PSIM [Telep]. Since the Telep_n quantum circuit trivially solves PSIM [Telep_n] this completes the proof of the main theorem, i.e., Theorem 1.1.

Theorem 3.8 (PSIM [Telep] $\notin \mathsf{NC}^0/\mathsf{poly}$)**.** *The possibilistic simulation of the* Telep *circuit, i.e., the* PSIM [Telep] *problem, is not in (non-uniform)* NC^0*.*

Proof. For the sake of contradiction, assume that there is a (non-uniform) NC^0-circuit C_n for the PSIM [Telep_n] problem.

We will use C_n in a non-black-box way. From Lemma 3.3 we have that there is an $L = \Theta(n)$ and a non-uniform NC^0 algorithm $\mathcal{A}$ that solves PSIM [CliffP$(Q)_L$]

for some fixed function $Q\colon \mathcal{C}^L \to \mathcal{P}$. This algorithm can be constructed from the algorithm $\mathcal{B}^{\mathsf{C}_n}_{L,m,\mathcal{J}}$ (Algorithm 2) by fixing the parameters $m, \mathcal{J}$ that are specific for the C_n in the non-uniform way, i.e., $\mathcal{A}$ is in $\mathsf{NC}^0/\mathsf{poly}$.

Lemma 3.7 tells us that there is a randomized (uniform) NC^0-circuit $\mathcal{D}$ with oracle access to $\mathcal{A}$ which solves PARITY. Since $\mathcal{A} \in \mathsf{NC}^0/\mathsf{poly}$ and $\mathcal{D}$ makes only $O(1)$ queries to $\mathcal{A}$, the algorithm $\mathcal{D}^{\mathcal{A}}$ can be realized by a non-uniform randomized NC^0-circuit $\mathcal{E}$.

Using standard procedures, we derandomize the randomized circuit $\mathcal{E}$ as follows. Because $\mathcal{E}$ is a randomized NC^0-circuit with a single output bit, its output is determined by a constant number of bits of the input instance x, as well as a constant number of uniformly random bits. Enumerating all possible assignments of those random bits, computing the result of the circuit $\mathcal{E}$ for each assignment, and taking the majority of the results gives the parity of x with certainty. This procedure is realized by an $\mathsf{NC}^0/\mathsf{poly}$ circuit. This shows that $\mathsf{NC}^0/\mathsf{poly}$ can solve PARITY, contradicting the fact that $\mathsf{PARITY} \notin \mathsf{NC}^0/\mathsf{poly}$. In conclusion, there is no (not even a non-uniform) NC^0-circuit for the $\mathsf{PSIM}\,[\mathsf{Telep}]$ problem. □

3.3.3 Other word problems for Clifford modulo Pauli groups

Here we argue that the class of problems we could solve if we had an NC^0 simulator for the $\mathsf{PSIM}\,[\mathsf{Telep}]$ problem is much broader than just computing PARITY. To address more general problems, we follow the lines of [13].

The main idea relies on the observation that the technique used in Section 3.2 to distinguish the state $|\Phi\rangle$ from $(I \otimes H)\,|\Phi\rangle$ via classical simulation can be used to distinguish $|\Phi\rangle$ from any state $(I \otimes C)\,|\Phi\rangle$, where $C \in \mathcal{C}/\mathcal{P} \setminus \{I\}$ is a non-Pauli Clifford gate. This is because by definition of the Clifford group, any non-Pauli Clifford takes a non-stabilizer state to another, different non-stabilizer state. This allows us to solve the word problem $\mathsf{WP}(\mathcal{C}/\mathcal{P}, C)$ for any $C \in \mathcal{C}/\mathcal{P} \setminus \{I\}$, i.e., the problem of deciding whether a series of single-qubit Clifford modulo Pauli gates multiplies to I or C.

For example, the Clifford gate SH generates a cyclic group of order 3. Thus replacing the Hadamard gate H by SH in our construction allows us to compute unbounded MOD_3 gates, i.e., to compute whether $|x| \bmod 3$ of the Hamming weight of the input string x is 0. Note that MOD_3 cannot be computed by constant-depth circuits with unbounded PARITY-gates [22,23] and vice versa, hence these problems are incomparable.

Combining the ability of computing MOD_3 with that of computing PARITY, we conclude that access to an NC^0 simulator for the $\mathsf{PSIM}\,[\mathsf{Telep}]$ problem elevates NC^0-circuits to $\mathsf{NC}^0[6]$, i.e., NC^0-circuit with MOD_6 gates.

Alternatively, we can arrive at similar conclusions using Barrington's result [24]. The latter states that the promise word problem for the solvable group S_3 characterizes width-3 permutation branching programs. The promise word problem for S_3 can be treated in our setting because S_3 is isomorphic to $\mathcal{C}/\mathcal{P}$. The isomorphism is given by

$$\begin{aligned} &I \mapsto 1\ , && H \mapsto (13)\ , && S \mapsto (12)\ , \\ &HS \mapsto (123)\ , && SHS \mapsto (23)\ , && (HS)^2 \mapsto (132)\ . \end{aligned}$$

Two-qubit Clifford gate teleportation circuits A similar construction can be obtained with two-qubit (instead of one-qubit) gate teleportation. Notice that this circuit can be implemented on a ring of pairs of qubits. Assuming an NC^0 possibilistic simulator for this problem and following the lightcone argument, we can use it to solve a possibilistic simulation of the intermediate problem depicted in Figure 10. We can ask what problems such a simulator allows us to solve.

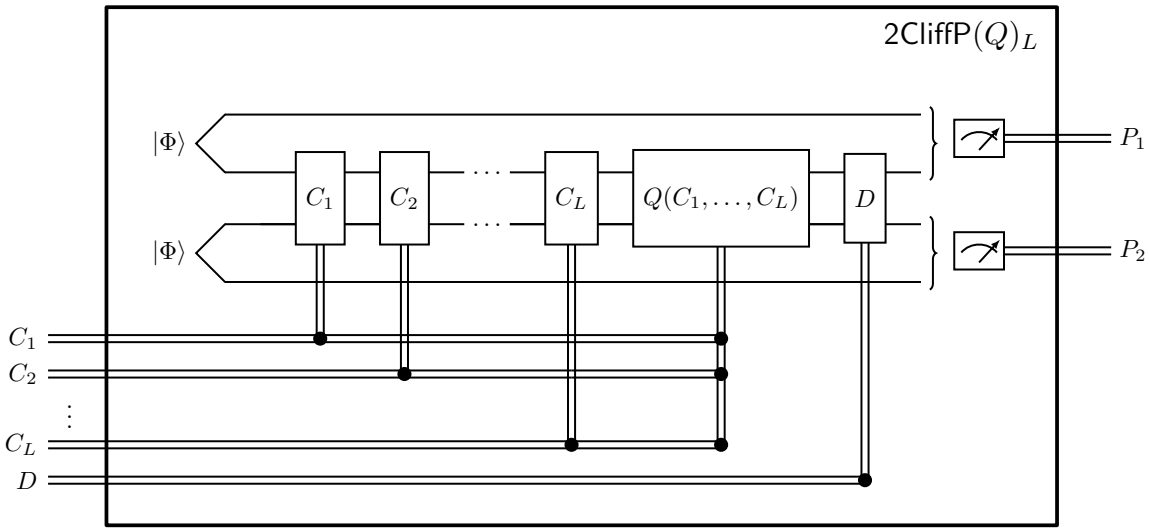


Figure 10: The Pauli-corrected two-qubit Clifford circuit 2CliffP$(Q)_L$ with correction function $Q : \mathcal{C}_2^L \to \mathcal{P}_2$. It takes an input $(C_1, \ldots, C_L, D) \in \mathcal{C}_2^{L+1}$ and outputs two single-qubit Pauli operators $P_1, P_2 \in \mathcal{P}$ obtained from the two Bell measurements.

Let us consider the word problem $\mathsf{WP}(\mathcal{C}_2/\mathcal{P}_2, H \otimes H)$, i.e., deciding whether a series of two-qubit Clifford modulo Pauli gates multiplies to $I \otimes I$ or $H \otimes H$, as used also in Ref. [13]. If we have an NC^0 simulator for $\mathsf{PSIM}\left[2\mathsf{CliffP}(Q)\right]$ we can use it to do possibilistic tomography in a similar way to what we did in Section 3.2.2 on each of the two Bell states.

Notice that the two-qubit Clifford group modulo Pauli operators is isomorphic to S_6 (see Ref. [13]). Since S_6 is a non-solvable group, the associated word problem is complete for the complexity class NC^1, i.e., for the complexity class of bounded fan-in logarithmic-depth polynomial-size circuits, according to Barrington's theorem [21]. A constant-depth classical simulator for the two-qubit gate-teleportation problem could therefore be used to solve any problem in NC^1.

Note, however, that none of these more general computational problems can be used to obtain a quantum advantage beyond NC^0 when only our techniques are applied. The lightcone argument we rely on constitutes a bottleneck for such a separation. The existence of a quantum advantage beyond NC^0 for $1D$-local quantum circuits therefore remains an open problem.

A PSIM $\left[\mathsf{CliffP}(Q)\right]$ and possibilistic state tomography

In this appendix we finish the proof of Theorem 3.4 from Section 3.2.2. It follows by specializing the arguments given in [13].

For convenience, let us define the non-stabilizer problem. Given a sequence of Cliffords $(C_1, \ldots, C_L) \in \mathcal{C}^L$ as the input the non-stabilizer problem is to output $P_A \otimes P_B \in \mathcal{P}^2$ such that $\langle \Phi | P_A \otimes P_B | \Phi \rangle = 0$, i.e, such that $P_A \otimes P_B$ is a non-stabilizer (up to $\pm$ sign) of the stabilizer state $(I \otimes C_L \cdots C_1) |\Phi\rangle$.

The remainder of the arguments of [13] shows that an oracle $\mathcal{O}$ for the non-stabilizer problem can be used to produce a uniformly random non-stabilizer of $(I \otimes C_L \cdots C_1) |\Phi\rangle$. We can then use this algorithm to produce generators of the stabilizer group of this state.

The algorithm for a uniformly random non-stabilizer relies on the following:

Lemma A.1. *Let $|\Psi\rangle$ be a stabilizer state and M a non-stabilizer of $|\Psi\rangle$. If V is a Clifford unitary chosen uniformly at random among the set of Cliffords satisfying $V|\Psi\rangle = |\Psi\rangle$, then $VMV^\dagger$ is a uniformly chosen non-stabilizer of $|\Psi\rangle$.*

For a proof see [13, Proof of Theorem 24].

Consider the Algorithm 3 below. It builds on Kilian's randomization [20] and satisfies the properties proven in Lemma A.2, see [13, Theorem 24].

Algorithm 3 Kilian-randomized function $\mathcal{M}^{\mathcal{O}}$. Here $\mathcal{O}$ is an oracle outputting a non-stabilizer of $(I \otimes C_L \cdots C_1) |\Phi\rangle$ on input $(C_1, \ldots, C_L) \in \mathcal{C}^L$.

function $\mathcal{M}^{\mathcal{O}}(C_1, \ldots, C_L)$
  $D_1, \ldots, D_L \leftarrow$ uniformly random Clifford unitaries
  $C'_1 \leftarrow D_1 C_1$
  for $j = 2, \ldots, L$ **do**
    $C'_j \leftarrow D_j C_j D_{j-1}^{-1}$
  $M \leftarrow \mathcal{O}(C'_1, \ldots, C'_L)$
  return $(I \otimes D_L)^\dagger M (I \otimes D_L)$

Lemma A.2. *Let $\mathcal{O}$ be a deterministic oracle for the non-stabilizer problem. The Algorithm 3 satisfies:*

(i) On input $(C_1, \ldots, C_L)$, the algorithm $\mathcal{M}^{\mathcal{O}}$ makes one query to the oracle $\mathcal{O}$ and returns a non-stabilizer of $(I \otimes \prod_{i=1}^L C_i) |\Phi\rangle$.

(ii) There is a random variable $\widetilde{M}$ on $\mathcal{P}$ such that for any sequence $C_1, \ldots, C_L$ of Cliffords, the output $\mathcal{M}^{\mathcal{O}}(C_1, \ldots, C_L)$ satisfies

$$\mathcal{M}^{\mathcal{O}}(C_1, \ldots, C_L) = \left(I \otimes \prod_{i=1}^L C_i\right) \widetilde{M} \left(I \otimes \prod_{i=1}^L C_i\right)^\dagger$$

with certainty. In particular, the dependence of the output of $\mathcal{M}^{\mathcal{O}}$ on the input $(C_1, \ldots, C_L)$ is only via the product $\prod_{i=1}^L C_i$.

(iii) The support of the distribution $P_{\widetilde{M}}$ (i.e., the distribution of $\widetilde{M}$) is contained in the set of non-stabilizers of the state $|\Phi\rangle$.

Proof. Claim (i) follows immediately from the definition of $\mathcal{M}^{\mathcal{O}}$: the operator M obtained in Step (6) of the Algorithm 3 is a non-stabilizer of $I\otimes(C'_L\cdots C'_1)\,|\Phi\rangle = (I\otimes D_L(\prod_{i=1}^L C_i))\,|\Phi\rangle$, hence $(I\otimes D_L)^\dagger M(I\otimes D_L)$ is a non-stabilizer of $(I\otimes\prod_{i=1}^L C_i)\,|\Phi\rangle$.

For the proof of (ii), consider the following procedure, given $(C_1,\ldots,C_L)$: pick D_L uniformly at random from the Clifford group, then pick $(C'_1,\ldots,C'_L)$ uniformly from the set of sequences satisfying

$$C'_L\cdots C'_1 = D_L\prod_{i=1}^L C_i\ . \tag{48}$$

Then set

$$M = \mathcal{O}(C'_1,\ldots,C'_L)$$

and

$$\widetilde{M} = (I\otimes(C'_L\cdots C'_1))^\dagger M(I\otimes(C'_L\cdots C'_1))\ . \tag{49}$$

Observe first that the distribution $P_{\widetilde{M}}$ is a function of $\prod_{i=1}^L C_i$ only and does not depend on the choice of $(C_1,\ldots,C_L)$ otherwise. Indeed, the sequence $(C_1,\ldots,C_L)$ only determines $\prod_{i=1}^L C_i$, implying that the product $D_L\prod_{i=1}^L C_i$ in (48) has a distribution which is a function of the product $\prod_{i=1}^L C_i$ only. Because the distribution of $(C'_1,\ldots,C'_L)$ is fixed by $D_L\prod_{i=1}^L C_i$ by definition, it follows that the distribution of M is determined by the product $\prod_{i=1}^L C_i$ (with no additional dependence on $(C_1,\ldots,C_L)$). It follows similarly from (49) that the distribution of $\widetilde{M}$ has this property.

By definition, we have

$$\begin{aligned}
\left(I\otimes\prod_{i=1}^L C_i\right)\widetilde{M}\left(I\otimes\prod_{i=1}^L C_i\right)^\dagger &= \left(I\otimes\left(\prod_{i=1}^L C_i\right)(C'_L\cdots C'_1)^\dagger\right)M\left((I\otimes(C'_L\cdots C'_1)\prod_{i=1}^L C_i\right)^\dagger\\
&= \left(I\otimes\prod_{i=1}^L C_i\left(D_L\prod_{i=1}^L C_i\right)^\dagger\right)M\left(I\otimes\left(D_L\prod_{i=1}^L C_i\right)\prod_{i=1}^L C_i\right)^\dagger\\
&= (I\otimes D_L)^\dagger M(I\otimes D_L)\\
&= (I\otimes D_L)^\dagger\mathcal{O}(C'_1,\ldots,C'_L)(I\otimes D_L)
\end{aligned}\tag{50}$$

where we used (48) in the second step. Observe that the marginal distributions $P_{C'_1\cdots C'_L D_L}$ are identical for both algorithms $\mathcal{M}$ and the procedure used in this proof: this is because steps (4) and (5) amount to Kilian randomization [20], producing a uniformly random sequence $(C'_1,\ldots,C'_L)$ of group elements with a fixed product as in (48). It follows that (50) is indeed identical to the output of $\mathcal{M}$ as claimed.

Finally, the claim (iii) is an immediate consequence of (i) and (ii). □

Let $\mathcal{M}$ be as specified in Algorithm 3 with an oracle access to the oracle $\mathcal{O}$ that solves the non-stabilizer problem. Consider the algorithm $\mathcal{M}'$ specified in Algorithm 4. Then we have the following (see [13, Theorem 24]):

Algorithm 4 Randomization function $\mathcal{M}'$.

function $\mathcal{M}'(C_1, \ldots, C_L)$
  $V \leftarrow$ a random Clifford satisfying $V|\Phi\rangle = |\Phi\rangle$.
  $C'_1 \leftarrow C_1 V$
  for $j = 2, \ldots, L$ **do**
    $C'_j \leftarrow C_j$
  $M \leftarrow \mathcal{M}^{\mathcal{O}}(C'_1, \ldots, C'_L)$
  return M

Lemma A.3. *On input $(C_1, \ldots, C_L)$, the algorithm $\mathcal{M}'$ returns a uniformly distributed non-stabilizer of $\left(I \otimes \prod_{i=1}^{L} C_i\right)|\Phi\rangle$.*

Proof. Let $\widetilde{M}$ denote the Pauli-valued random variable from Lemma A.2. Let V be a uniformly distributed one-qubit Clifford unitary subject to the constraint $V|\Phi\rangle = |\Phi\rangle$. Then we have

$$\begin{aligned} P_{\mathcal{M}'(C_1,\ldots,C_L)} &= P_{\mathcal{M}^{\mathcal{O}}(C_1V,C_2,\ldots,C_L)} \\ &= P_{\prod_{i=1}^{L} C_i V \widetilde{M} V^\dagger \prod_{i=1}^{L} {C_i}^\dagger} \end{aligned}$$

by definition of the algorithm $\mathcal{M}'$ and Lemma A.2 (ii). Since $\widetilde{M}$ is supported on the set of non-stabilizers of the state $|\Phi\rangle$ (see Lemma A.2 (iii)), it follows from Lemma A.1 that $V\widetilde{M}V^\dagger$ is a random non-stabilizer of $|\Phi\rangle$ with uniform distribution. The claim follows from this. □

Finally, we combine a deterministic oracle for the non-stabilizer problem and the randomization procedure that we specified in Algorithms 3 and 4 to an algorithm learning the stabilizer group of the state $(I \otimes C_L \cdots C_1)|\Phi\rangle$ (up to a sign factor).

Theorem A.4. *Let $\mathcal{O}$ be a deterministic oracle for the non-stabilizer problem. There is an NC^0 algorithm with an oracle access to $\mathcal{O}$ that uses uniformly random bits, and on input $(C_1, \ldots, C_L)$*

(i) makes $O(1)$ oracle calls to $\mathcal{O}$,

(ii) either outputs two distinct non-identity Pauli operators S_1, S_2 such that

$$\langle\Phi|\left(I \otimes (C_L \cdots C_1)\right)^\dagger S_j \left(I \otimes (C_L \cdots C_1)\right)|\Phi\rangle \in \{\pm 1\} \qquad \text{for} \qquad j = 1, 2\ ,$$

or reports a failure, and

(iii) succeeds with probability at least $2/3$.

Proof. Let $|\Psi\rangle$ be a stabilizer state. A Pauli operator M has expectation value $\langle\Psi|M|\Psi\rangle \in \{\pm 1\}$ if and only if M commutes with each stabilizer of $|\Psi\rangle$. If this is not the case, then M is a non-stabilizer. It follows that a two-qubit stabilizer state has exactly 12 non-stabilizers of the form $P_A \otimes P_B$ with $(P_A, P_B) \in \mathcal{P}^2$. The remaining 4 operators constitute (up to $\pm$-signs) the stabilizer group of $|\Psi\rangle$. In particular, given

the 12 non-stabilizers of $|\Psi\rangle$, it is straightforward to output two distinct non-identity Pauli operators $S_1 = P_A \otimes P_B$, $S_2 = P'_A \otimes P'_B$ such that $\langle\Psi| S_j |\Psi\rangle \in \{\pm 1\}$ for $j = 1, 2$.

We apply this to the state $|\Psi\rangle = (I \otimes C_L \cdots C_1)|\Phi\rangle$. Let $\mathcal{M}'$ be the algorithm defined in Algorithm 4, using the function $\mathcal{M}^{\mathcal{O}}$ given in Algorithm 3. Notice that this algorithm returns a uniformly random non-stabilizer of $|\Psi\rangle$ given the input $C_L \cdots C_1$. Then, there is an algorithm $\mathcal{M}''$ that uses $O(1)$ queries to $\mathcal{M}'$ and determines, with probability at least $2/3$, all 12 non-stabilizers of $|\Psi\rangle$ of the form $P_A \otimes P_B$ with $(P_A, P_B) \in \mathcal{P}^2$ or it reports failure. The claim follows. □

Acknowledgments

LC and RK gratefully acknowledge support by the European Research Council under grant agreement no. 101001976 (project EQUIPTNT). XCR thanks the Swiss National Science Foundation (SNSF) for their support.

References

[1] Barbara M. Terhal and David P. DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004. doi:10.26421/QIC4.2-5.

[2] Daochen Wang. Possibilistic simulation of quantum circuits by classical circuits. *Phys. Rev. A*, 106:062430, Dec 2022. doi:10.1103/PhysRevA.106.062430.

[3] Daniel Gottesman and Isaac Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390 – 393, 1999. doi:10.1038/46503.

[4] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, oct 2018. doi:10.1126/science.aar3106.

[5] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020. doi:10.1038/s41567-020-0948-z.

[6] Scott Aaronson. Quantum Computing, Postselection, and Probabilistic Polynomial-Time, Dec 2004. doi:10.48550/arXiv.quant-ph/0412187.

[7] Stephen Fenner, Frederic Green, Steven Homer, and Yong Zhang. Bounds on the Power of Constant-Depth Quantum Circuits. In *Fundamentals of Computation Theory*, pages 44–55, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11537311_5.

[8] Maarten Van Den Nes. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. *Quantum Information & Computation*, 10(3):258–271, 2010. doi:10.26421/QIC10.3-4-6.

[9] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial

hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011. doi:10.1098/rspa.2010.0301.

[10] Edward Farhi and Aram Wettroth Harrow. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. Technical report: MIT/CTP-4771, 2016. doi:10.48550/arXiv.1602.07674.

[11] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993682.

[12] Daniel Grier and Luke Schaeffer. The classification of Clifford gates over qubits. *Quantum*, 6:734, Jun 2022. doi:10.22331/q-2022-06-13-734.

[13] Daniel Grier and Luke Schaeffer. Interactive shallow Clifford circuits: Quantum advantage against NC^1 and beyond. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 875–888, 2020. doi:10.1145/3357713.3384332.

[14] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 515–526, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316404.

[15] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880, 1969. doi:10.1103/PhysRevLett.23.880.

[16] Adam Bene Watts and Natalie Parham. Unconditional Quantum Advantage for Sampling with Shallow Circuits, 2023. doi:10.48550/arXiv.2301.00995.

[17] Marc-Olivier Renou, Elisa Bäumer, Sadra Boreiri, Nicolas Brunner, Nicolas Gisin, and Salman Beigi. Genuine quantum nonlocality in the triangle network. *Physical Review Letters*, 123(14):140401, 2019. doi:10.1103/PhysRevLett.123.140401.

[18] Scott Aaronson and Daniel Gottesman. Identifying Stabilizer States, 2008. See http://pirsa.org/08080052/.

[19] Ashley Montanaro. Learning stabilizer states by Bell sampling, Jul 2017. doi:10.48550/arXiv.1707.04012.

[20] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 20–31, New York, NY, USA, 1988. Association for Computing Machinery. doi:10.1145/62212.62215.

[21] David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1):150–164, Feb 1989. doi:10.1016/0022-0000(89)90037-8.

[22] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41:333–338, 1987. doi:10.1007/BF01137685.

[23] Roman Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 77–82, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28404.

[24] David A. Barrington. Width-3 permutation branching programs, technical memorandum tm-293. Technical report, M.I.T. Laboratory for Computer Science, Dec 1985. URL: https://hdl.handle.net/1721.1/149102.