Learning efficient decoders for quasichaotic quantum scramblers

Lorenzo Leone,[1,2,3,*,†] Salvatore F. E. Oliviero,[1,2,3,*,‡] Seth Lloyd,[4,5,§] and Alioscia Hamma[6,7,‖]

[1]*Physics Department, University of Massachusetts Boston, Massachusetts 02125, USA*
[2]*Theoretical Division (T-4), Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[3]*Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[4]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*
[5]*Turing Inc., Brooklyn, New York, USA*
[6]*Dipartimento di Fisica 'Ettore Pancini', Università degli Studi di Napoli Federico II, Via Cintia 80126, Napoli, Italy*
[7]*INFN, Sezione di Napoli, Italy*

(Received 8 January 2023; accepted 20 December 2023; published 22 February 2024)

Scrambling of quantum information is an important feature at the root of randomization and benchmarking protocols, the onset of quantum chaos, and black-hole physics. Unscrambling this information is possible given perfect knowledge of the scrambler [arXiv:1710.03363]. We show that one can retrieve the scrambled information even without any previous knowledge of the scrambler, by a *learning* algorithm that allows the building of an efficient decoder. Remarkably, the decoder is classical in the sense that it can be efficiently represented on a classical computer as a Clifford operator. It is striking that a classical decoder can retrieve with fidelity one all the information scrambled by a random unitary that *cannot* be efficiently simulated on a classical computer, as long as there is no full-fledged quantum chaos. This result shows that one can learn the salient properties of quantum unitaries in a classical form and sheds a new light on the meaning of quantum chaos. Furthermore, we obtain results concerning the algebraic structure of t-doped Clifford circuits, i.e., Clifford circuits containing t non-Clifford gates, their gate complexity, and learnability that are of independent interest. In particular, we show that a t-doped Clifford circuit U_t can be decomposed into two Clifford circuits U_0, U_0' that sandwich a local unitary operator u_t, i.e., $U_t = U_0 u_t U_0'$. The local unitary operator u_t contains t non-Clifford gates and acts nontrivially on at most t qubits. As simple corollaries, the gate complexity of the t-doped Clifford circuit U_t is $O(n^2 + t^3)$, and it admits a efficient process tomography using $\mathrm{poly}(n, 2^t)$ resources.

DOI: 10.1103/PhysRevA.109.022429

I. INTRODUCTION

In quantum mechanics, learning an unknown quantum state or process is a crucial problem. The applications of this task range from quantum information and benchmarking protocols [1–3], the understanding of quantum chaos [4–6], quantum chemistry [7–11], quantum cryptography [12–22], and black-hole physics [23,24].

If the quantum process to be investigated is modeled by a random unitary, its learning may prove an extremely daunting task. If one were able to learn a random unitary, one could do wonders: for example, to decode the information emitted in Hawking radiation without any previous knowledge of the black hole. In this paper, by learning, we mean learning enough features of the process so that scrambled information can be retrieved from it.

There is a special class of unitary operators, the Clifford group, that has been proven to be efficiently learnable with a polynomial effort [25,26]. It is not a coincidence that this class of unitary operators is the same that can be efficiently simulated by a classical computer [27]. From a quantum advantage point of view, the ability to learn only those unitaries that can be classically simulated is unsatisfactory. One wants to learn those quantum processes that *cannot* be efficiently simulated classically. Is it possible? How costly is it?

In this paper, as well in the companion Letter [28], we show that it is possible to learn a random unitary operator that cannot be efficiently simulated, as long as this process is not *fully* chaotic. Moreover, the learned features can be efficiently represented on a classical computer. The fact that something that cannot be efficiently simulated can then be learned in a form that is efficiently represented is so surprising that is almost sounds contradictory. It seems that then, after all, the random unitary *could* be efficiently simulated. Our result must be understood in terms of what we are learning. We are learning just the features that are enough to unscramble the information. The result is still very surprising, but at least it starts sounding more believable: the complex features are useless, and what is useful is efficiently representable. Moreover, the efficient representation is not *always* possible: when quantum chaos kicks in, such a representation breaks down. At that point, in order to unscramble information, one also has to learn all the complex features of the unitary, and that requires an exponentially complex representation. In some sense, our result clarifies what quantum chaos is: that feature

[*] These authors contributed equally to this work.
[†] lorenzo.leone001@umb.edu
[‡] s.oliviero001@umb.edu
[§] slloyd@mit.edu
[‖] alioscia.hamma@unina.it

2469-9926/2024/109(2)/022429(29)

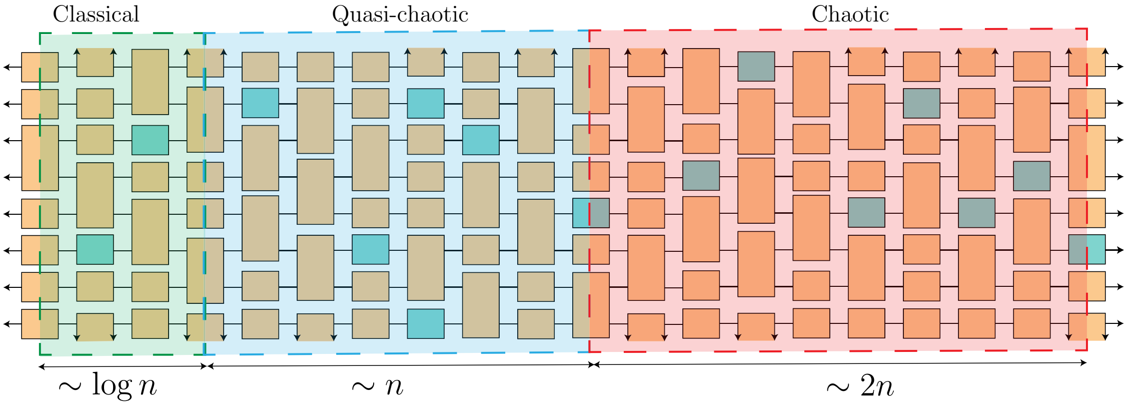


FIG. 1. Sketch of a t-doped Clifford circuit with variable depth proportional to the doping t counting the number of injected non-Clifford gates (blue). The first part of the circuit (green-dashed box) has depth within $\log n$ and corresponds to a *classical* regime, where the circuit can be efficiently simulated by a classical computer. As the depth reaches $t \sim n$ (blue-dashed box), the circuit reaches a quasichaotic regime, where classical simulation is exponentially hard but, at the same time, it cannot reproduce the universal properties of Haar-random circuits. However, quasichaotic circuits are learnable with an exponential effort in t. Quantum chaos is reached for $t \sim 2n$ (red-dashed box): then universal properties are reached and no learning is possible.

of quantum evolutions that does not allow for any kind of classical representation [29].

This work is organized as follows: in Sec. II we present a more detailed overview of the problem and of our results, eschewing the heavy technical details; in Sec. III, we briefly review previous results concerning the task of learning quantum processes; in Secs. IV A and IV B, we review the information scrambling setup introduced in Ref. [23] and the meaning of learning quantum information scrambled after a complex quantum dynamics; in Sec. IV C we present the main results of the paper in a nontechnical fashion, while in Sec. V A, after having introduced the technical tools needed for the proof of the main theorems, we present the quantum algorithm able to learn scrambled information.

II. OVERVIEW OF THE PROBLEM AND RESULTS

The notion of learning a quantum evolution is intimately connected with the notion of irreversibility. Quantum mechanics is unitary; one can, in principle, undo any quantum evolution by running it backward. Unfortunately, without any prior knowledge of the unitary U, the ability to undo a quantum evolution is almost never guaranteed. If one could learn U by query accesses then one would be able to revert quantum evolutions. However, for the overwhelming majority of unitaries, the task of learning is exponentially hard [30–34] and effectively reversibility is lost. This fact is related to the exponential growth of the Hilbert-space dimension with the number of degrees of freedom, which in turn would require exponentially small precision per exponentially many experiments.

As we mentioned above, there are some special quantum processes that do not feature complex behavior: Clifford circuits can be efficiently learned and simulated. On the opposite end of Clifford circuits, there are chaotic quantum circuits. These circuits can be obtained by random Clifford circuits on n qubits with the addition of cn ($c \geqslant 2$) non-Clifford resources, e.g., T gates. They feature universal frame potentials [35], fluctuations of entanglement and of the higher-order out-of-time-order correlation functions (OTOCs) [29,36,37]. In view of the Gottesman-Knill theorem [27], they also require an exponential number of resources to be simulated on a classical computer.

In this paper, we discuss the learnability for a wide class of unitary evolutions; that is, unitary operators obtained from a random Clifford circuit enriched with t non-Clifford gates—the so-called t-doped Clifford circuits U_t, see Fig. 1. As shown in Ref. [29], there is a gradual transition from Clifford circuits to quantum chaos. In the middle of the transition; that is, for a number of Clifford resources cn with density $c < 1$, one has not yet attained quantum chaos, although these circuits do require an exponential number of resources to be simulated. We call these circuits quasichaotic.

As we see in Sec. V B, we constructively show that every t-doped Clifford circuit can be decomposed as

$$U_t = U_0[\mathbb{1}_{[n-t]} \otimes u_{[t]}]U_0', \tag{1}$$

i.e., as a product of two Clifford circuits U_0, U_0' and a local unitary $[\mathbb{1}_{[n-t]} \otimes u_{[t]}]$ acting on at most t qubits and containing t non-Clifford gates, see Theorem 2 (*Compression Theorem*). The decomposition in Eq. (1) is valid as long as $c < 1$, i.e., for quasichaotic quantum circuits. It states that all the non-Cliffordness in U_t can be *compressed* in t qubits only. Moreover, all the Clifford parts of U_t can be learned by having query access to U_t. Here and throughout the work, we refer to *query access* as the ability to perform the unitary transformation U_t followed by a measurement on a quantum register consisting of n qubits. We present an algorithm that learns the Clifford operations U_0 and U_0' by poly$(n, 2^t)$ query accesses to U_t, see Theorem 3. As a corollary, we show that with time complexity and query complexity both scaling as poly$(n, 2^t)$, it is possible to learn a full tomographic description of a general t-doped Clifford circuit, see Corollary 3. Additionally, a straightforward consequence of Eq. (1), a t-doped stabilizer state $|\psi_t\rangle \equiv U_t\,|0\rangle^{\otimes n}$ can be *compressed* as

$$|\psi_t\rangle = U_0[|0\rangle^{\otimes(n-t)} \otimes |\phi\rangle_t], \tag{2}$$

i.e., to the computational basis state $|0\rangle^{\otimes(n-t)}$ and a nonstabilizer state $|\phi\rangle_t$ living on a t-qubit subsystem (see Corollary 2). Again, a decomposition as in Eq. (2) is valid for $c < 1$. What is more, the decomposition in Eq. (1) shows that the gate complexity $\#(U_t)$—i.e., the minimum number of gates necessary to build U_t from the identity [38]—of Clifford+T circuits obeys $\#(U_t) = O(n^2 + t^3)$, which interpolates between $O(n^2)$ for $t = o(n)$ to $O(n^3)$ for $t = \Omega(n)$, see Corollary 1. Remarkably, Clifford circuits doped with a sublinear number of T gates have the same gate complexity as Clifford circuits. In this paper, we discuss the learnability of quantum evolutions in the context of information scrambling [23]: we present a quantum algorithm based on a constrained random Clifford Completion (CC) that aims at learning a perfect and efficient decoder without any previous knowledge of the scrambler. The decoder is efficient in the sense that it can be efficiently represented on a classical computer, more precisely, it is a Clifford operator [27]. We show that such learning is efficient in terms of resources as long as the scrambler can be efficiently simulated, it is exponentially expensive in a quasichaotic regime and becomes impossible in the presence of full-fledged quantum chaos.

The main result of the paper is the following: if U_t is a unitary obtained by a t-doped Clifford circuit, one can learn a *Clifford* decoder V by means of a probabilistic quantum algorithm based on a constrained random Clifford completion algorithm, which employs $\text{poly}(n, 2^t)$ query accesses to U_t and a time $\text{poly}(n, 2^t)$.

The scrambler is a unitary U_t^{AB} acting on the $|A|$ qubits of information plus additional $|B|$ qubits with $n = |A| + |B|$. The output of the scrambler consists of $|C| + |D| = n$ qubits of which only $|D|$ can be accessed by the decoder. The decoder V takes in the qubits in D and returns an output that should contain the information initially present in A with a fidelity

$$\mathcal{F}_{\rm V}(U_t) \geqslant \frac{1}{1 + 2^{2|A|+t-2|D|}}, \tag{3}$$

while the probability of learning the decoder V is

$$\mathscr{P}(V) \geqslant 1 - 2^{t-2(n-|D|)}. \tag{4}$$

Equations (3) and (4) are the content of Theorem 1. It is important to highlight that the proposed CC algorithm searches for and implements a decoder V belonging to the Clifford group. This means that the decoder, once (and if) found, can be efficiently represented in a classical computer: it is, in this sense, a *classical* decoder. In the companion Letter [28], we present the result of Theorem 1 in a simplified fashion.

When can such a decoder be found? If the scrambler U_t is a Clifford circuit (or Clifford circuits with doping up to $t \sim \log n$), the decoder exists and it can be learned by $\text{poly}(n, 2^t)$ resources. If the doping scales like $t = cn$ with $c < 1$, learning is still possible, but it requires exponential resources, as for simulability. We call this regime quasichaotic. As the density of non-Clifford resources increases, a transition to full quantum chaos is approached [29,39] and, for $c \geqslant 2$, no learning is any longer possible, no matter the resources employed.

As one can see from Eq. (3), as the number t of T gates in the circuit U_t increases, the size of the subsystem D the decoder should access in order to unscramble the information and reconstruct the state $|\psi_A\rangle$ must increase as well. Notably, the decoding via a Clifford operator breaks down only when the number t of T gates approaches n, $t \sim n$, i.e., in the quasichaotic regime (cf. Fig. 1). In this case, to obtain perfect fidelity, one has to acquire a number of qubits $|D|$ larger than half of the system, $|D| \geqslant n/2$, which in turn makes the probability of decoding scale as $\approx 1 - 2^{n-2|D|}$. As one increases t beyond this threshold, the probability of decoding decays exponentially. In other words, Clifford operations can unscramble information as long as the dynamic is quasichaotic ($c < 1$) while unscrambling becomes impossible for fully chaotic dynamics ($c \geqslant 1$).

III. REVIEW OF PREVIOUS RESULTS

In this section, we provide a review of some known results on the problem of learning unitary dynamics and explain the advances of the present paper in the current literature. The task of learning a unitary operator U—defined on n qubits—can be generally defined in two ways: either (i) by the ability to learn and synthesize U on a quantum device or (ii) by the ability to learn some problem-depending properties of U and use them to solve a particular task. The latter includes tasks like disentangling a given quantum state [40,41], unscrambling quantum information [42], or learning expectation values. Intuitively, (ii) is a weaker form of learning, and, as a matter of fact, (i) implies (ii) (up to computational challenges). Both approaches have been widely explored in the literature [40–49].

Naively, one can define the task of learning a unitary operator U by knowing all the matrix elements $\langle x|U|x'\rangle$ of U, where $|x\rangle$, $|x'\rangle$ are basis vectors. The above task is immediately found to be inefficient because $O(2^{2n})$ matrix elements need to be determined and stored in a classical memory. The weaker requirement of applying an unknown unitary on a quantum state is more suitable: in Ref. [47], the authors show that unitaries acting on $O(\text{poly}\log(n))$ qubits can be emulated by having access to unknown samples of input-output states. While storing matrix elements of the unknown unitary would scale superpolynomially in the number of qubits, both the runtime and the query access of their algorithm scale polynomially. A more efficient way to define the learning task is knowing an efficient decomposition in elementary gates of a given unitary operator U. It is well known that a universal set of gates is given by a CNOT, a Hadamard gate that allows superposition in the computational basis, an S gate (a $\hat{z}$ rotation of $\pi/2$), and finally a T gate (a $\hat{z}$ rotation of $\pi/4$). The Solovay-Kitaev theorem ensures that any unitary operator can be approximated by an ordered product of elementary gates within any desired accuracy ϵ [50,51]. While learning the right order of gates to approximate a target unitary operator, in general a very hard combinatorial problem [51], the decomposition in elementary gates allows an efficient classical representation of a quantum unitary operator. A third approach is to use the operator basis of Pauli operators. Thanks to the unitarity of U, only the action on $O(n)$ Pauli operators needs to be determined, but at the same time (a) exponentially many measurements are typically required to learn each element of the map $P \mapsto U^\dagger P U$ for P being an element of operator basis of the Pauli group, and (b) classical postprocessing requires, in general, exponentially many memory bits.

Clifford unitary operators constitute a particular example in which the latter approach is suitable. Indeed, Clifford operations map elements of the Pauli group to elements of the Pauli group. This means that it is just sufficient to learn the Pauli operators resulting from the adjoint action of U, $P \mapsto U^{\dagger}PU \in \mathbb{P}$. The learning of Clifford circuits and states created by Clifford circuits has been studied in Refs. [25,26,52,53]. In Ref. [53] it has been shown that, with $O(n)$ queries to two copies of a given stabilizer state $|\psi\rangle$, it is possible to learn its tomographic decomposition. While in Ref. [26], generalizing the results of Refs. [25,53], it has been shown how $O(n)$ queries to a Clifford circuit U are sufficient to learn and synthesize U on a quantum computer. However, the algorithms developed in Refs. [25,26,53] are specifically designed to learn a Clifford unitary, and therefore cannot be employed in the task of learning of a general unitary operator.

The task of learning quantum unitary dynamics is intimately connected to the problem of classical simulability of quantum computation. Clifford circuits admit an efficient classical representation. This means that computation made by states created by the action of Clifford unitary operators—the so-called stabilizer states—can be efficiently reproduced by a classical computer [27]. No quantum advantage can be achieved. At the same time, stabilizer states are typically highly entangled, a condition that has been believed to be the key property to unlock a quantum computational advantage. Nevertheless, fine-grained properties in entanglement structure reveal the profound difference between the entanglement produced by Clifford circuits from that produced by universal unitary operators. Previous works [40,41,48] probed the operational difference between these two types of entanglement. It has been shown that, by employing a Monte Carlo metropolis kind of algorithm, it is possible to completely disentangle a state evolved by Clifford gates. Conversely, the metropolis algorithm fails at disentangling a state evolved by a universal circuit. At this level, the task of disentangling can be thought as that of finding a unitary operator V that, when applied on the evolved state $U|\psi_0\rangle$, makes the evolved state $VU|\psi_0\rangle$ nonentangled in a given bipartition. The success or failure of such a disentangling algorithm reveals the different entanglement structures produced by Clifford gates and universal gates. Besides connecting classical simulability and learnability of quantum dynamics, the above result naturally defines two complexity classes: (I) states that can be efficiently disentangled and (II) states that are not disentangleable.

It is noteworthy that, while it is possible to learn the classical representation of Clifford unitaries in terms of Pauli operators and to learn how to disentangle states entangled by Clifford circuits, there are other approaches to learning Clifford circuits that fail. One example is provided in the context of quantum machine learning by variational quantum algorithms (VQAs). Compilation of a given unitary operator U aims to find the minimum set of elementary gates that approximate the target U (e.g., see Refs. [30–34,44,46,54]). In this context, the strategy of VQAs is to classically train a parametrized (fix depth) quantum circuit to minimize a problem-dependent expectation value $\mathrm{tr}[OV^{\dagger}(\vec{\theta})U^{\dagger}\psi_0 UV(\vec{\theta})]$. Unfortunately, the above strategy fails in general: no-go theorems have been established that prevent the compilation of a unitary U drawn from a unitary k design with $k \geqslant 2$ (set of unitaries that reproduces up to the second moment of the full unitary group) [55]. It is noteworthy that the above VQA task fails even for the Clifford group, being a unitary 3-design. At the same time, if in a VQA task, the learner accepts to spend exponentially many resources, in Ref. [56] it has been shown that gradient descent can learn an arbitrary random unitary using an exponentially large parameter landscape. Another example is provided by probably approximately correct (PAC) learning [57–61]. The goal of PAC learning is to learn a function relative to a certain distribution of inputs. In the context of unitary circuits, the goal is to learn the output distribution of a set of observables P_x (that can be reasonably be thought as being Pauli operators labeled by x) through U given a set of input states ρ_y, i.e., $f(x,y) := \mathrm{tr}(P_x U\rho_y U^{\dagger})$. Then, a PAC learner aims to design a function $\widetilde{f}(x,y)$ such that with probability at least $1-\delta$, obeys $\mathbb{E}_{x,y}[f(x,y)-\widetilde{f}(x,y)]^2 < \epsilon$. Interestingly, Clifford unitary operators cannot be PAC learned without a collapse of polynomial hierarchies [62].

After describing various results within the domain of learning Clifford circuits, let us now move beyond the confines of the Clifford group. Any Clifford circuit can be built out of three elementary gates: CNOT, Hadamard, and phase gate S. The addition of one non-Clifford gate makes the above set universal for quantum computation. In other words, Clifford circuits fail to be universal because of the lack of just one element, which is traditionally chosen to be the T gate. The injection of non-Clifford gates into Clifford circuits gradually drives the circuits to feature universal properties [63]. This is reflected by the fact that the best-known classical simulation algorithm scales exponentially in the number of non-Clifford gates [64]. While for Clifford circuits the road map of what can or cannot be learned and relative strategies is sufficiently complete and, for universal circuits, the task of learning is believed to be unfeasible, regarding the gray area between these two "complexity classes," there are still many open questions. For example, how does the success of a learning task change for t-doped Clifford circuits, i.e., Clifford circuits polluted by t non-Clifford gates? Does the learnability encounter a sharp transition or a continuous crossover driven by the amount t of non-Cliffordness in the circuit?

The doping of Clifford circuits is intimately connected with the concept of quantum pseudorandomness: a set of unitary operators is a unitary k design if it reproduces up to the kth moment of the Haar (uniform) distribution over the unitary group [65,66]. The Clifford group has been proven to be a unitary 3-design, and to fail to be a unitary 4-design [67]. In Ref. [68], one can see that a vanishing density t/n of non-Clifford resources is sufficient to break the 4-design barrier and reproduce any k design (for $k < \log n$) with an error ϵ. In other words, in the framework of unitary k design, it is possible to *homeopathically* dope Clifford circuits to obtain approximate k designs within the desired accuracy ϵ. Unfortunately, reproducing up to the kth moment of the distribution over the full unitary group within an error ϵ is not always sufficient to reproduce the complex universal behavior. In Refs. [29,39], it has been shown that to truly address the transition between the noncomplex behavior of Clifford circuits and the complex Haar random behavior, an exponentially small error $\epsilon \sim 2^{-n}$ is required. Indeed in Ref. [35], it has been shown that a necessary and sufficient condition to form a 4-design is to reproduce the universal value of the 8-point OTOC:

$$\mathrm{OTOC}_8(U) := \frac{1}{d}\mathrm{tr}[P_1 P_2(U) P_3 P_4(U) P_1 P_4(U) P_3 P_2(U)], \quad (5)$$

where P_1, P_2, P_3, P_4 are nonidentity Pauli operators, and $P_2(U) \equiv U P_2 U^{\dagger}$ and similarly for $P_4(U)$. The average $\mathrm{OTOC}_8(U)$ for t-doped Clifford circuits $U \in \mathcal{C}_t$ is (proved in Ref. [29]):

$$\langle \mathrm{OTOC}_8(U)\rangle_{\mathcal{C}_t} = \Theta\left[\frac{1}{d^2}\left(\frac{3}{4}\right)^t + \frac{1}{d^4}\right], \quad (6)$$

that interpolates between the Clifford value $\langle \mathrm{OTOC}_8(U)\rangle_{\mathcal{C}_0} = \Theta(d^{-2})$ and the Haar value $\langle \mathrm{OTOC}_8(U)\rangle_{\mathcal{C}_\infty} = \Theta(d^{-4})$. As a result, the injection of cn (with $c \geqslant 2$) non-Clifford gates in a Clifford circuit is both necessary and sufficient to drive the transition towards the universal behavior $\Theta(d^{-4})$. The value of the eight-point OTOC discriminates between various regimes of interest of doped Clifford circuits. In particular, the injection of $t = \Theta(1)$ non-Clifford gates does not change at all the value of OTOC_8. The doping with $\Theta(\log n)$ non-Clifford resources—being part of the class of circuits that can be efficiently classically simulable—do not change the value of the OTOC up to a polynomial overhead, i.e., $\Theta(d^{-2}\,\mathrm{poly}^{-1}(n))$. Instead, the injection of a number $\Omega(\log n) < t < n$ of non-Clifford resources lies in the quasichaotic quantum circuit regime, i.e., a class of circuits that is transient between two universality classes (Clifford and Haar) that are nonuniversal but, at the same time, cannot be simulated by classical means. This transient regime is reflected by a value of the eight-point OTOC of $\Theta(d^{-3})$ [cf. Eq. (6)].

The above results thus suggest that the task of learning could, in principle, become unfeasible for universal (chaotic) circuits only, thus when the number of non-Clifford gates is $\approx 2n$.

The question about the learning of Clifford circuits polluted with t non-Clifford gates has been explored in several ways. First, from the point of view of the disentangling algorithm, in Ref. [48], it has been shown that the success of the disentangling algorithm is exponentially suppressed in the number t of non-Clifford gates. In Ref. [26], using techniques similar to those introduced by in Ref. [25,53], is proposed an efficient way to encode, learn and synthesize a particular class of t-doped Clifford circuit, i.e., circuit made as $U_0^{(1)} T_k U_0^{(2)}$, where $U_0^{(1)}$, $U_0^{(2)}$ are Clifford operations and T_k are k parallel single qubit T gates. In the paper, it is also proven that the task of learning and synthesis is possible as long as the number of non-Clifford gates $t = O(\log n)$. Remarkably, this is the same threshold for a t-doped Clifford circuit to be efficiently simulated classically [64]. Conversely, in Ref. [69] the authors claim that, while the output distribution $P(x) = |\langle x|U|0^{\otimes n}\rangle|^2$ of a Clifford circuit can be efficiently learned, the injection of even a single T gate in a Clifford circuit makes the task of learning the output distribution $P(x)$ hard [assuming the learning parties with noise (LPN) assumption [70]]. Their result provides a sharp separation between Clifford circuits and doped Clifford circuits, in contrast with the result previously discussed. Note that, the injection of a single T gate in a Clifford circuit falls inevitably in the class of circuits that can be written as $U_0^{(1)} T U_0^{(2)}$, that can be efficiently encoded classically and learned, as shown in Ref. [26]. After all, it is well known the difference in performances of learning tasks with or without the possibility of having access to two copies of the target, being a unitary or a quantum state. At the same time, the question of whether the output distribution of a 1-doped Clifford circuit can be learned when measurements in arbitrary single-qubit bases are available remains an open question.

Along these lines, in this work, the problem of learning doped Clifford circuits has been studied in the context of unscrambling quantum information. While the technicalities of the protocol will be discussed in the following sections, the concept of unscrambling is cognate to the one of disentangling: the task is to find a decoder unitary V that, mocking the action of a unitary operator U_t, undoes the action of U_t only on a subspace (say A) $VU_t\,|\psi_A\rangle\,|\psi_B\rangle$, retrieving quantum information $|\psi_A\rangle$ scrambled by U_t. In a previous work [42], a metropolis algorithm has been employed—similar to the one for disentangling—with the task of searching for the decoder V. By modeling the unitary U_t as a t-doped Clifford circuit, it is numerically shown that the success rate of the algorithm, quantified by recovery fidelity $|\langle\psi_A|VU|\psi_A \otimes \psi_B\rangle|^2$, is exponentially decaying in t. In other words, the recovery fidelity is smaller than ϵ just after $t = \Omega(\log \epsilon^{-1})$ non-Clifford gates.

Is it possible to do better? The answer is yes, as the present paper shows. We show that the proposed CC algorithm is able to learn—with $\mathrm{poly}(n, 2^t)$ resources—a perfect decoder V for a t-doped Clifford circuit, up to $t < n$ non-Clifford gates, as we set out to show starting from the next section. The main technical contribution, as mentioned earlier in the above section, is the development of the *compression theorem* (Theorem 1) which reveals the existence of a compression method for t-doped Clifford circuits [as seen in Eq. (1) above]. This compression effectively concentrates all the non-Clifford elements into a subsystem of t qubits (thus independent from n), enabling the use of a brute-force tomographic algorithm to learn the non-Clifford components. Therefore, this task is feasible only up to $t = O(\log n)$. We then put this consideration in rigorous grounds in Corollary 3, where we show that it is possible to learn an efficient classical description of a t-doped Clifford circuit using $\mathrm{poly}(n, 2^t)$ resources, which includes both sample and computational complexity. In fact, this represents an advancement over the state-of-the-art algorithms for learning t-doped Clifford circuits, which had previously been constrained to specific circuit structures.

IV. LEARNING QUANTUM INFORMATION FROM AN UNKNOWN SCRAMBLER

A. Information scrambling and decoupling theorem

In this section, we make a brief review of the decoupling theorem introduced by Hayden and Preskill in Ref. [23], in the context of black-hole evaporation. Consider the Hilbert space of $n = |A| + |B| = |C| + |D|$ qubits partitioned as

$$\mathcal{H} = A \otimes B = C \otimes D \quad (7)$$

and a unitary map

$$U_{AB} : A \otimes B \mapsto C \otimes D. \tag{8}$$

Denote as $\mathbb{P}(\Lambda)$ the Pauli group (modulo phases) on the subsystem Λ composed of $|\Lambda|$ qubits with $\Lambda \in \{A, B, C, D\}$, $d_\Lambda \equiv 2^{|\Lambda|}$ and define the average four-point out-of-time-order correlation function $\Omega(U_{AB})$ as

$$\begin{aligned}\Omega(U_{AB}) &:= \langle \mathrm{OTOC}_4(U_{AB})\rangle_{P_A,P_D}\\ &\equiv \frac{1}{d}\, \langle \mathrm{tr}[P_A P_D(U_{AB}) P_A P_D(U_{AB})]\rangle_{P_A,P_D}\,,\end{aligned} \tag{9}$$

where $P_D(U_{AB}) \equiv U_{AB}^\dagger P_D U_{AB}$ and $\langle\cdot\rangle_{P_A} \equiv \frac{1}{d_A^2}\sum_{P_A\in\mathbb{P}(A)}(\cdot)$ is the average over the Pauli group on A and similarly for P_D. The OTOC operationally quantifies how information initially encoded in A is scrambled by U through the output system $\mathcal{H}_C \otimes \mathcal{H}_D$, see Ref. [4]. The function $\Omega(U_{AB})$ is a quantity related to the group commutator between the local Pauli group on A and D: it attains the value one if the (average) support of $P_D(U_{AB})$ commutes with Pauli operators in A while it decreases as the support of P_D grows in space, the so-called operator growth, which in turn defines scrambling behavior [71–73]: a unitary operator U_{AB} is said to be a *scrambler* if and only if [4]

$$\Omega(U_{AB}) \simeq \frac{1}{d_A^2} + \frac{1}{d_D^2} - \frac{1}{d_A^2 d_D^2}, \tag{10}$$

where $\simeq$ means *up to an order* d^{-2}.

Scrambling of quantum information is connected to that of the information retrieval [23]: imagine Alice decides to encode some quantum information in A. As this is quantum information, we need to possess a reference state on R that is perfectly entangled with A. By denoting the EPR pair between two spaces of the same dimension d_X by $|XX'\rangle = d_X^{-1/2}\sum_{i_X}|i_X\rangle\otimes|i_{X'}\rangle$, the quantum information possessed by Alice is encoded in the EPR pair $|RA\rangle$. At this point, Alice tosses her half of such an EPR pair (A) in the scrambler. On the other hand, Bob wants to retrieve the information encoded by Alice and tossed into the scrambler by Alice by having access to part of the output state, namely, D. If Bob initially possesses one half of an EPR pair $|BB'\rangle$, the initial state of the system is $|RA\rangle\,|BB'\rangle$, while after scrambling the total state on $RB'CD$ is

$$|\Psi\rangle_{RB'CD} = U_{AB}\otimes I_{RB'}\,|RA\rangle\,|BB'\rangle\,. \tag{11}$$

In the context of black-hole evaporation, an old black hole B is maximally entangled with the Hawking radiation B' possessed by Bob, while D is the Hawking radiation emitted by the black hole after U_{AB} has scrambled the quantum information tossed in it by Alice and C represents the shrinking black-hole interior and is inaccessible for any observer, being beyond the event horizon. At this point, the question is how much information, initially possessed by Alice, is, after the scrambling unitary, in Bob's possession? One quantifies the information shared by two parties, e.g., R and C, by the quantum mutual information between R and C, defined through von Neumann entropies:

$$I(R|C) := S(\rho_R) + S(\rho_C) - S(\rho_{RC}), \tag{12}$$

where $S(\rho) := -\mathrm{tr}(\rho\log\rho)$ and $\rho_\Lambda := \mathrm{tr}_{\bar\Lambda}(|\Psi\rangle\langle\Psi|)$ with $\bar\Lambda$ being the complement of Λ. Simple calculations [4] show that for the state $|\Psi\rangle_{RB'CD}$ [in Eq. (11)] one obtains $S(\rho_R) = |A|$ and $S(\rho_C) = |C|$. One can also show that the two Rényi entropy $S_2(\rho) := -\log\mathrm{tr}\rho^2$ obeys [4]

$$S_2(\rho_{RC}) = -\log\frac{d_A}{d_C}\Omega(U_{AB}). \tag{13}$$

From the hierarchy of Rényi entropies one then finds that, if U_{AB} is a scrambler, the *decoupling theorem* applies:

$$I(R|C) = O(2^{2|A|-2|D|}), \tag{14}$$

that is, only an ϵ amount of information is shared between Alice (R) and the output of the scrambler C provided that $|D| = |A| + \log\epsilon^{-1}$. Thanks to the unitarity of the evolution, all the information is in Bob's possession, i.e., DB': the mutual information between R and $B'D$ is maximal,

$$I(R|B'D) = |A| - O(2^{2|A|-2|D|}). \tag{15}$$

Let us make some remarks concerning why the model can be applied in the context of black-hole evaporation. Let us take a step back and review the Page calculations on the entropy production from a black hole, see Ref. [74]. Indeed, modeling a black hole as a complex random unitary U, Page finds that the entanglement entropy between the black-hole interior I and the emitted Hawking radiation E is

$$S_I = -\log\frac{d_I + d_E}{d_I d_E + 1} = -\log\left[d_I^{-1} + d_E^{-1} + O(d^{-1})\right].$$

Thus, if $|I| = fn$ for $f < 1/2$ one has $S_I = |I| + O(2^{(1-2f)n})$, i.e., maximal entropy up to a exponentially small error. Thus, following the Page reasoning, one has (i) as long as $|I| \gg |E|$ the Hawking radiation E does not contain any information about the black-hole interior I, but rather it is the black-hole interior that knows all about E; (ii) as soon as $|E| \gg |I|$ the Hawking radiation E contains all the information about the black hole interior I, being maximally entangled with it; while (iii) between the two regimes there is a gray area where the entanglement is not maximal. In the context of the Hayden-Preskill thought experiment, the hypothesis that Bob B' shares an EPR pair with the black-hole initial interior B relies exactly upon the Page reasoning: sharing an EPR pair means being maximally entangled with the initial black-hole interior, which is possible only if the black hole has emitted much more than half of the initial qubits, i.e., $|I| \ll |E|$. Thus, among all the radiation E emitted in the history of this black hole, the qubits in Bob's possession are only a subset $B' \subset E$.

That said, the decoupling theorem, being a pure information-theoretic result, finds its own applications as a tool for, *exempli gratia*, quantum communication and quantum teleportation [75]. Thus, there is no need to specialize the discussion on black-hole physics.

B. Recovery algorithm after a scrambling dynamics

The decoupling theorem says that the quantum information initially encoded in the input state in A is completely transferred through the scrambling unitary dynamics to Bob, i.e., the system D and B'. The very scrambling behavior of U_{AB} has destroyed any correlation between the reference state in R and the inaccessible part of the information in C. Since now the state in R must be perfectly correlated with the state in the hands of Bob, there should exist a unitary operator V on $B'D$ able to recover all the information encoded in A. In other words, there should exist a unitary V which enables Bob to distill an EPR pair between R and a reference system of the same dimension of R, say R'. One calls such operator a *decoder*. In Ref. [24], it is shown how Bob can operate such distillation by picking as decoder the transpose of the scrambler U_{AB}: Bob needs a further EPR pair $|A'R'\rangle$ on auxiliary spaces A' and R' and appends it to the output of the scrambler, obtaining $|\Psi\rangle_{RCDB'}\,|A'R'\rangle$. The dimension of A' is chosen such that $A'\otimes B'$ is isomorphic to $A\otimes B$. Then Bob applies the operator $V^*_{B'A'}$ and finally projects onto an EPR pair on $D\otimes D'$ by $\Pi_{DD'}\equiv|DD'\rangle\langle DD'|$. The final state after the algorithm performed by Bob is thus

$$|\Psi_{\mathrm{out}}(V)\rangle\equiv\frac{1}{\sqrt{P_{\mathrm{out}}}}\Pi_{DD'}V^*_{B'A'}\,|\Psi\rangle_{RCDB'}\,|A'R'\rangle\,,\tag{16}$$

where P_{out} is a normalization. The success of the algorithm; that is, $V^*_{B'A'}$ being a decoder, is guaranteed if the state (16) looks like $|\Psi_{\mathrm{out}}(V)\rangle\simeq|RR'\rangle\otimes|rest\rangle_{CC'}\otimes|DD'\rangle$, i.e., a factorized state with an EPR pair between Alice qubits R and Bob qubits R'. The factorization is possible only because no information is shared between R and CC' thanks to the decoupling theorem. To check whether the algorithm is successful, one computes the fidelity between $|\Psi_{\mathrm{out}}(V)\rangle$ and the *target EPR pair* one wants to distill, i.e., $|RR'\rangle$. The fidelity between the state in Eq. (16) and $|RR'\rangle$, $\mathcal{F}_V(U)\equiv\mathrm{tr}(\Pi_{RR'}|\Psi_{\mathrm{out}}(V)\rangle\langle\Psi_{\mathrm{out}}(V)|)$, being a function of the scrambler U_{AB} and the decoder V, can be recast as [42]

$$\mathcal{F}_{\mathrm{V}}(U)=\frac{1}{d_A^2}\frac{\langle\mathrm{tr}(P_D(U)P_D(V))\rangle_{P_D}}{\langle\mathrm{tr}(P_D(U)P_AP_D(V)P_A)\rangle_{P_A,P_D}},\tag{17}$$

where we dropped the subscript for both U and V. Then one can see that, if $V=U$ and the unitary U is a *scrambler*, i.e., $\Omega(U)\simeq d_A^{-2}+d_D^{-2}-d_A^{-2}d_D^{-2}$ one obtains a fidelity

$$\mathcal{F}_V(U)=1-O(4^{|A|-|D|}),\tag{18}$$

i.e., to have a fidelity $1-\epsilon$, one must have $|D|=|A|+\log\epsilon^{-1/2}$. In the context of black-hole physics, the radiation emitted by the black hole, after Alice tosses their qubits in its interior, must contain $|D|=|A|+\log\epsilon^{-1/2}$ qubits to ensure a successful recovery by Bob [23].

As we have seen, the decoder V can be easily found if one knows perfectly U_{AB}. The main goal of this paper is to present a way of *learning* the decoder V without any previous knowledge of U_{AB}.

In the following, we drop the subscript AB and denote the scrambler as U_t because we will always be concerned with a t-doped Clifford circuit; that is, a Clifford circuit in which a number t of single-qubit non-Clifford gates has been injected, see Fig. 1.

In the following sections, we present a learning quantum algorithm that aims at finding a decoder V that maximizes the fidelity $\mathcal{F}_V(U_t)$. The main question of this paper is can one learn the behavior of U_t by limited access to it and limited resources? The answer is yes, provided that the scrambler is not too chaotic [29]. The learning quantum algorithm is a CC algorithm.

C. Main result

In this section, we present the main result of the paper, avoiding technical details of the CC algorithm, later presented in Sec. VC. We first present the main result as a main claim, and then make a rigorous statement in the form of Theorem 1.

Main claim. If U_t is a t-doped Clifford circuit it is possible to build a perfect Clifford decoder V using a quantum algorithm requiring poly$(n,2^t)$ resources, provided that $t<n$, that is if U_t is at most quasichaotic.

Remark M1. The Clifford decoder V still satisfies the decoupling theorem (see Sec. IV A), as random Clifford unitaries are good scramblers [29,76]. Surprisingly, a Clifford operator can decode a unitary U_t that makes extensive use of non-Clifford resources. As stated above, a Clifford decoder exists as long as U_t is quasichaotic. Beyond that threshold, U_t finally becomes too complex to be decoded by a Clifford decoder V.

Remark M2. For nonchaotic t-doped Clifford circuits; that is, for $t=O(\log n)$, the Clifford decoder exists and can be found with resources (time and sample complexity) both polynomial in n. The learning of the decoder is thus efficient. For quasichaotic circuits, i.e., for $t\lesssim n$, the efficient Clifford decoder can be found, but with a $\exp(n)$ amount of resources.

Remark M3. From the fidelity formula, Eq. (17), we can see that a perfect decoder [i.e., with fidelity $\mathcal{F}_V(U)\simeq1$] must reproduce the action of $U_t^\dagger P_DU_t$ for any $P_D\in\mathbb{P}(D)$. For $t=0$, this requirement can obviously be fulfilled. However, for every $t>0$, it is not possible to *exactly* reproduce the action of a non-Clifford unitary U_t on a Pauli operator P_D. How is it then possible that a decoder even exists? To understand this, let us explore the consequences of the fact that we only need to reproduce the behavior of U_t on the Pauli operators in D. First, it might happen that for some Pauli operators, U_t would send them again in Pauli operators, effectively behaving on them like a Clifford operator. Define the subgroup of the Pauli group on D

$$G_D(U):=\{P\in\mathbb{P}(D)|U^\dagger P_DU\in\mathcal{P}\},\tag{19}$$

where $\mathcal{P}$ is the Pauli group on n qubits. If U_t is a Clifford operator, then $G_D(U_t)\equiv\mathbb{P}(D)$. Similarly to Eq. (9), we can define a *truncated* OTOC by averaging over the group $G_D(U_t)$ instead of $\mathbb{P}(D)$:

$$\Omega_{G_D}(U_t):=\frac{1}{d}\,\langle\mathrm{tr}(P_AP_D(U_t)P_AP_D(U_t))\rangle_{\mathbb{P}(A),G_D(U_t)}\,.\tag{20}$$

If U_t is a scrambler, one can easily see that [75], if $|G_D(U_t)|>1$,

$$\Omega_{G_D}(U_t)\simeq\frac{1}{d_A^2}+\frac{1}{|G_D(U_t)|^2}-\frac{1}{d_A^2|G_D(U_t)|^2}.\tag{21}$$

As far as the operators in $G_D(U_t)$ are concerned, a Clifford operator would still be a perfect decoder. In building the decoder V then, we choose a Clifford operator with the constraints

$$\forall\,P\in G_D(U_t),\quad V^\dagger PV=U_t^\dagger PU_t;\tag{22}$$

that is, V equals the action of U_t on the subgroup $G_D(U_t)$. The above requirement can be fulfilled because U_t acts as a Clifford operator on $G_D(U_t)$ and because of the unitarity of both U_t and V or, equivalently, thanks to the group structure of $G_D(U_t)$. While a unitary operator U_t is uniquely defined by its adjoint action on every Pauli operator (or, to be rigorous, on all the generators of $\mathbb{P}$), Eq. (22) constraints the unitary V only on the generators of the group $G_D(U_t)$, leaving the other degrees of freedom free. This will be the key insight for the success of the randomized algorithm presented in Sec. V C. The randomized algorithm builds a decoder V by first imposing the constraints Eq. (22) and then completing the Clifford operator in a *random* way. We name this algorithm the constrained random Clifford completion (CC) algorithm. We show that the fidelity attained by the decoder V is

$$\mathcal{F}_{\mathrm{V}}(U_t) = \frac{1+R}{d_A^2\Omega_{G_D}(U_t)+R'}, \tag{23}$$

where

$$\begin{aligned} R &:= [d|G_D(U_t)|]^{-1} \sum_{P_D\in\mathbb{P}(D)\backslash G_D(U_t)} \operatorname{tr}(P_D(U_t)P_D(V)),\\ R' &:= [d|G_D(U_t)|]^{-1} \sum_{P_D\in\mathbb{P}(D)\backslash G_D(U_t),P_A} \operatorname{tr}(P_A P_D(U_t)P_A P_D(V)) \end{aligned} \tag{24}$$

(see proof in Appendix C). Is not surprising that, if $\operatorname{tr}(P_D(U_t)P_D(V)) = 0$ for every $P_D \notin G_D(U_t)$, then $R = R' = 0$. Whether $R, R' = 0$ depends on both U_t and V. Since V is partially random, we can consider the probability of $R = R' = 0$. Remarkably, the unconstrained degrees of freedom in choosing the decoder V allow finding, with an overwhelming probability, a decoder for which $R = R' = 0$. As we shall see, the size of the set of constrained degrees of freedom is of crucial importance.

Remark M4. The CC algorithm searches for and implements a decoder V belonging to the Clifford group. There are two important consequences of this result: First, Clifford circuits admit an efficient classical representation and can be stored easily in a classical memory. Second, synthesis of Clifford circuits is also efficient [77]: starting from the classical representation of a Clifford unitary V, one needs $O(n^2)$ moves in terms of CNOT, Hadamard, and phase gates. Lastly, the implementation of Clifford circuits can be easily done fault-tolerantly, making the above algorithm not too expensive in terms of quantum resources. The following theorem is the main result of the paper:

Theorem 1. Let U_t be a t-doped Clifford scrambler. Let $\mathcal{V}_{U_t}^D := \{V \in \mathcal{C}(n) | V^\dagger P V = U_t^\dagger P U_t,\ \forall\, P \in G_D(U_t)\}$, the set of Clifford circuits obeying Eq. (22). The CC algorithm builds a Clifford decoder $V \in \mathcal{V}_{U_t}^D$ with time complexity and a number of query accesses scaling as $\mathrm{poly}(n, 2^t)$ such that, with probability

$$\Pr_{V\in\mathcal{V}_{U_t}}(R=0, R'=0) \geqslant 1 - 2^{-(2|C|-t)}, \tag{25}$$

it yields a fidelity obeying $R = R' = 0$. The decoder V thus retrieves the information with a fidelity given by

$$\mathcal{F}_{\mathrm{V}}(U_t) = \frac{1}{d_A^2\Omega_{G_D}(U_t)}. \tag{26}$$

If U_t is a scrambler, then the fidelity reads

$$\mathcal{F}_V(U_t) \simeq \frac{1}{1+\frac{d_A^2-1}{|G_D(U_t)|}} \geqslant \frac{1}{1+2^{2|A|+t-2|D|}}, \tag{27}$$

cf. Eq. (21) and Lemma 4 in Appendix C.

The above theorem says that a randomized decoder built according to the CC algorithm presented in Sec. V C recovers the information scrambled by U_t with probability $\Pr(R = 0, R' = 0)$ that converges to one exponentially fast with $2|C| - t$, and success fidelity converging to one exponentially fast with $2|D| - 2|A| - t$. In Sec. V E, we provide numerical evidence of the success of the CC algorithm in finding a perfect decoder for quasichaotic scramblers.

Remark T1. As later shown in Sec. V C, a query access to the unitary U_t corresponds to the ability to apply the unitary U_t on an n-qubit quantum register. We remark that querying the unitary U_t twice enables the application of $U_t^{\otimes 2}$ on a $2n$-qubit quantum register.

Remark T2. The key insight for the success of the algorithm is that the randomization over the unconstrained degrees of freedom in V [which dictate the behavior of the decoder V on the elements of $\mathbb{P}(C) \cup \mathbb{P}(D) \setminus G_D(U_t)$], yields, with high probability, a value $R = R' = 0$. First of all, this condition is not necessary to achieve perfect fidelity. What is needed is that $R, R' \ll 1$. However, there is an intuitive explanation as to why the stronger condition $R = R' = 0$ is likely, given the assumptions. Both the quantities R, R' are proportional to the sum over Hilbert-Schmidt inner products. This sum depends on at most 2^t terms. Since $P_D(V)$ is still a Pauli operator (because V is Clifford), it is the tensor product over n qubits of single qubit Pauli matrices and can be represented by a $2n$-bit string. The Hilbert-Schmidt inner product then becomes the bit-string inner product. Of these $2n$ bits, though, the constraints in Eq. (22) fix at least $2|D| - t$ bits leaving $2n - 2|D| + t = 2|C| + t$ bits free. The probability that this string is orthogonal to another $2n$ bit string is thus lower-bounded by $1 - 2^{-(2|C|+t)}$. However, the operator $P_D(U_t)$ is the linear combination of 2^t strings, because every T gate evolved by a Clifford circuit produces two strings. In other words, a t-doped Clifford circuit produces string entropy [63,78]. Finally, we can conclude that the probability that 2^t strings of type $P_D(V)$ are orthogonal to the corresponding $P_D(U)$ is then lower-bounded by $1 - 2^{-(2|C|+t)}2^{2t} = 1 - 2^{-(2|C|-t)}$.

Remark T3. From the above formulas, it can be easily checked that the number of T gates increases the size of the subsystem D that must be processed for a successful decoding. Indeed, the size of the subsystem D that must be read by the decoder scales as $|D| = |A| + t/2 + \log \epsilon^{-1/2}$ to ensure a decoding fidelity ϵ close to one. Notably, the decoding is still possible when the number of T gates scales as $t \sim n$, while it becomes no longer possible as $t > n$ because the success probability becomes exponentially suppressed.

Remark T4. One of the reasons why the result of Theorem 1 is surprising is that, if we read out too many bits $|D|$, for example, capturing too many bits of the Hawking radiation, then the algorithm fails. After all, one might think that the more one learns, the better it is. However, the fidelity crucially depends on the fact that we can imitate the unitary U_t, which is not Clifford, with a Clifford operator V. This is only possible if V encodes away in C all the differences between the two. If $|D|$ grows to become the full number of qubits n, the fidelity (23) becomes the unitary fidelity $d^{-2}|\operatorname{tr}(U_t^\dagger V)|^2$ which is obviously less than one even for a vanishing density of non-Clifford gates (see Lemma 5 in Appendix C).

Remark T5. The fact that, in order to achieve fidelity $\mathcal{F}_V(U_t) = 1$, the density c of non-Clifford gates cannot exceed the unity is also important and it is connected to the transition in quantum complexity and crossover to quantum chaos driven by the doping by non-Clifford resources, see Refs. [29,39]. To obtain universal purity fluctuations and universal behavior for the 8-OTOC [Eq. (5)], the amount of non-Clifford gates must be greater than $2n$, cf. Eq. (6). The same result is obtained in Ref. [79] with the tool of the unitary stabilizer nullity. Similar conclusions can be reached by looking at the stabilizer Rényi entropy $M(|U_t\rangle)$ of the Choi state $|U_t\rangle$ associated with U_t. A necessary condition to obtain the universal (and maximal) value $M(|U_t\rangle) \simeq 2n$ is that the number of non-Clifford gates $t \geqslant 2n$ [78,80].

Remark T6. The use of a Clifford circuit that learns a t-doped Clifford circuit allows efficient classical memory storage; indeed, Clifford operators can be efficiently encoded in classical memory using $O(n^2)$ parameters. Thus, although beyond $t = O(\log n)$ the algorithm becomes exponentially hard in t, the fact that Clifford operators are suitable decoders for quasichaotic quantum circuits implies that, at least from a memory-storage point of view, the algorithm remains efficient in terms of classical resources. Conversely, in the regime when the density c of non-Clifford gates is $c \geqslant 2$, a chaotic circuit maps all the Pauli operators to a superposition of exponentially many Pauli strings that, preventing the possibility of finding a suitable Clifford decoder, leads to an exponential needs of classical memory and the consequent impossibility of the learning process, even provided an infinite measurement precision.

All the above considerations show why we call quasichaotic, doped Clifford circuits having finite densities less than one. As the density of non-Clifford resources overcomes $c = t/n > 1$, a transition quantum complexity happens and eventually the dynamic reaches the Haar random behavior for $t/n \geqslant 2$ after which nothing can be reliably learned.

V. THE CLIFFORD COMPLETION ALGORITHM

A. Technical preliminaries

In this section, we review well-known notions on the stabilizer formalism because they are instrumental in proving the main result of the paper. We refer to Appendix A for the list of notations used throughout the paper. Consider the Hilbert space of n qubit $\mathcal{H}$ and let $d = 2^n$ its dimension. Let us introduce the Pauli matrices $\mathbb{1}_{[1]}, \sigma^x, \sigma^y$, and σ^z:

$$\mathbb{1}_{[1]} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$\sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{28}$$

where $\mathbb{1}_{[1]}$ is the identity on the space of one qubit. Throughout the paper, we denote operators $O_{[m]}$ acting on a subsystem $[m]$ containing m qubits with subscript $[m]$. The Pauli group $\mathcal{P}$ on n qubits is defined as the n-fold tensor product of the single-qubit group $\mathcal{P}([1])$ obtained by $\{\mathbb{1}_{[1]}, \sigma^x, \sigma^y, \sigma^z\}$ times a multiplicative factor of $\pm 1, \pm i$. Note that choosing two Pauli operators $P, Q \in \mathcal{P}$, they either commute $[P, Q] = 0$ or anticommute $\{P, Q\} = 0$. In what follows, we consider the quotient group of the Pauli group (that is, we ignore the global phases $\{\pm 1, \pm i\}$),

$$\mathbb{P} := \mathcal{P}/\{\pm 1, \pm i\}. \tag{29}$$

Note that $\mathbb{P}$, the group of Pauli strings, is an Abelian group with respect to the matrix multiplication modulo phases. In the following, we take the license to refer to both $\mathbb{P}$ and $\mathcal{P}$ as the Pauli group, but mind that the two different notations mean slightly different things. A set of generators for the Pauli group is given by the set $\mathfrak{l} = \{\sigma_i^x, \sigma_i^z\}_{i=1}^n$, where $\sigma_i^{x,z}$ is the operator acting as σ^x, σ^z on the ith qubit and identically elsewhere. Otherwise, we denote as $\mathfrak{g}$ any other generators of the Pauli group. Thanks to the unitarity of $U \in \mathcal{U}(n)$, one can compute the adjoint action $U^\dagger P U$ on every $P \in \mathbb{P}$ by knowing all the (adjoint) actions of U on the set of generators $\mathfrak{g}$, i.e., $U^\dagger \sigma U$ for any $\sigma \in \mathfrak{g}$. Thus, the knowledge of $U^\dagger P U$ for every $P \in \mathfrak{g}$ completely determines U up to a global phase. This property is particularly useful because $|\mathfrak{g}| = 2n$, i.e., the size of the set of generators scales linearly with n. Although, only $O(n)$ chunks of information are required to completely determine a $2^n \times 2^n$ matrix, for a general unitary operator, the knowledge of $U^\dagger P U$ requires 4^n complex numbers. However, there exists a special class of unitary operators for which the knowledge of $U^\dagger P U$ requires just $O(n)$ bits of information: the Clifford group.

Denote as $\mathcal{C}(n)$ the Clifford group on n qubit, i.e., a subgroup of the unitary group with the following property:

$$\mathcal{C}(n) := \{U_0 \in \mathcal{U}(n) | U_0^\dagger P U_0 \in \mathcal{P}, \ \forall\, P \in \mathcal{P}\}. \tag{30}$$

In other words, the Clifford group is the normalizer of the Pauli group $\mathcal{P}$. Thanks to the aforementioned property, quantum computation employing Clifford unitary operators can be simulated classically in a time scaling as $O(n^3)$ [27,77]. As we shall see, any Clifford operator U_0 can be encoded in a tableau [77] T_{U_0}, which efficiently encodes the action of U_0 on a set of generators $\mathfrak{g}$, which is conventionally chosen to be the local set of generators $\mathfrak{l}$. Let us first introduce some technical notions. Let us first recall that $\mathfrak{l} = \{\sigma_i^x, \sigma_i^z\}_{i=1}^n$ represents the set of generators for the Pauli operators. Hence, any Pauli operator P in $\mathbb{P}$ can be expressed as

$$P = (-i)^{\sum_i^n x_i z_i} \left(\sigma_1^x\right)^{x_1} \left(\sigma_1^z\right)^{z_2} \otimes \left(\sigma_2^x\right)^{x_2} \left(\sigma_2^z\right)^{z_2}$$
$$\otimes \cdots \otimes \left(\sigma_n^x\right)^{x_n} \left(\sigma_n^z\right)^{z_n}, \tag{31}$$

where $\sum$ denotes the sum modulo 2, and where $(x_1, z_1, \ldots, x_n, z_n)$ belongs to $\mathbb{F}_2^{2n}$, with $\mathbb{F}_2$ being the finite field of integers with arithmetic modulo 2. For sake of clarity, let us introduce the following notation: $(P)_{\mathbf{xz}} \equiv (x_1, z_1, \ldots, x_n, z_n)$. Thanks to Eq. (31), $(P)_{\mathbf{xz}}$ completely characterizes the Pauli operator P, and so there exists an isomorphism between $\mathbb{P}$ and the field $\mathbb{F}_2^{2n}$.

Example 1. The single qubit Pauli group $\mathbb{P}([1])$ is isomorphic to $\mathbb{F}_2^2$:

$$(\mathbb{1}_{[1]})_{xz} = (00), \quad (\sigma^x)_{xz} = (10),$$
$$(\sigma^z)_{xz} = (01), \quad (\sigma^y)_{xz} = (11).$$

The above example clearly shows how one can associate a pair of integers modulo 2 with each Pauli matrix. In this isomorphism, the product of two Pauli operators is given by the XOR operation performed on the corresponding binary strings.

Example 2. Consider $P_1 = \sigma_x$ and $P_2 = \sigma_y$; their product is equal to

$$\sigma^x \sigma^y \mapsto (10) \oplus (11) = (01) \mapsto \sigma^z.$$

The final element to characterize the isomorphism between $\mathbb{P}$ and the field $\mathbb{F}_2^{2n}$ is given by the commutation relations of two Pauli operators. Given $P_1, P_2 \in \mathbb{P}$, then [67]

$$P_1 P_2 = (-1)^{\omega[(P_1)_{\mathbf{xz}},(P_2)_{\mathbf{xz}}]} P_2 P_1, \tag{32}$$

where $\omega[(P_1)_{\mathbf{xz}}, (P_2)_{\mathbf{xz}}] \equiv (P_1)^T_{\mathbf{xz}} \Omega (P_2)_{\mathbf{xz}}$ is the symplectic form, with Ω a $2n \times 2n$ block-diagonal matrix with each block equal to $(0 \quad 1 \; 1 \quad 0)$.[1] In formulas,

$$\Omega := \bigoplus_{i=1}^{n} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{33}$$

As a consequence $\omega[(P_1)_{\mathbf{xz}}, (P_2)_{\mathbf{xz}}]$ is able to tell us if two Pauli operators P_1 and P_2 commute.

Example 3. Consider $n = 1$ and $P = \sigma^x$, $P' = \sigma^y$. Then $(P)_{xz} = (10)$ and $(P')_{xz} = (11)$. Computing the symplectic form $\omega[\cdot, \cdot]$, we have

$$\omega[(P)_{xz}, (P')_{xz}] = [(1 \cdot 1) + (1 \cdot 0)] = 1, \tag{34}$$

and therefore σ_x anticommutes with σ_y, as expected. Now consider $n = 2$ and $P \equiv \sigma^x \otimes \sigma^x$ and $P' \equiv \sigma^z \otimes \sigma^y$. One has $(P)_{\mathbf{xz}} = (1010)$ and $(P')_{\mathbf{xz}} = (0111)$, thus

$$\omega[(P)_{\mathbf{xz}}, (P')_{\mathbf{xz}}] = 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 = 0, \tag{35}$$

and therefore $\sigma^x \otimes \sigma^x$ commutes with $\sigma^z \otimes \sigma^y$ as expected.

The isomorphism between $\mathbb{P}$ and the field $\mathbb{F}_2^{2n}$ sets the first building block in the implementation of a $2n \times 2n$ tableau that encodes all the Clifford information. Notably, with the efficient description of a Pauli operator P in terms of a $2n$-dimensional vector $(P)_{\mathbf{xz}}$, the possibility of implementing a classical representation of a Clifford operator becomes less surprising. Consider the symplectic group $\mathrm{Sp}(2n, \mathbb{F}_2^{2n})$, a group of $2n \times 2n$ matrices M satisfying the following equation:

$$M \Omega M^T = \Omega. \tag{36}$$

It has been shown [81] that, for every Clifford operator $U_0 \in \mathcal{C}(n)$, there exists a unique symplectic matrix $\tilde{T}_{U_0} \in \mathrm{Sp}(2n, \mathbb{F}_2)$ such that, for $U_0 P U_0^\dagger =\propto P'$, one has

$$\tilde{T}_{U_0} (P)_{\mathbf{xz}} = (P')_{\mathbf{xz}}. \tag{37}$$

Conversely, the opposite is also true, so every symplectic matrix is associated with a Clifford unitary $U_0 \in \mathcal{C}(n)$. The aforementioned facts highlight that a Clifford unitary can be represented by a $2n \times 2n$ matrix, providing an efficient encoding scheme. The action of a Clifford operator on $\sigma \in \mathfrak{l}$ can be efficiently encoded—being $U_0^\dagger \sigma U_0$ a Pauli operator—in a $(2n+1)$-bit string, where the first $2n$ bits encode $(U_0^\dagger \sigma U_0)_{\mathbf{xz}}$, while the last bit encodes the phase of $U_0^\dagger \sigma U_0$, that can be either $+1$ or -1. One can implement the tableau T_{U_0} through a $(2n) \times (2n+1)$ Boolean matrix, where each row stores the action of U_0 on one of the $2n$ generators of $\mathbb{P}$ and, by convention, the set of generators is chosen to be $\mathfrak{l}$. A generic tableau T_{U_0} can be written in the following way:

$$T_{U_0} \equiv (\tilde{T}_{U_0} | \boldsymbol{\phi}) = \left(\begin{array}{cccc|c} \left(U_0^\dagger \sigma_1^x U_0\right)_{x_1 z_1} & \left(U_0^\dagger \sigma_1^x U_0\right)_{x_2 z_2} & \cdots & \left(U_0^\dagger \sigma_1^x U_0\right)_{x_n z_n} & \phi_1 \\ \left(U_0^\dagger \sigma_1^z U_0\right)_{x_1 z_1} & \left(U_0^\dagger \sigma_1^z U_0\right)_{x_2 z_2} & \cdots & \left(U_0^\dagger \sigma_1^z U_0\right)_{x_n z_n} & \phi_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \left(U_0^\dagger \sigma_n^z U_0\right)_{x_1 z_1} & \left(U_0^\dagger \sigma_n^z U_0\right)_{x_2 z_2} & \cdots & \left(U_0^\dagger \sigma_n^z U_0\right)_{x_n z_n} & \phi_{2n} \end{array}\right), \tag{38}$$

where $\tilde{T}_{U_0} \in \mathrm{Sp}(2n, \mathbb{F}_2^{2n})$ is the *partial tableau* [81,82], a $2n \times 2n$ symplectic matrix that encodes the action $\sigma \mapsto U_0^\dagger \sigma U_0 \in \mathbb{P}$; while $\boldsymbol{\phi}$ is a $2n \times 1$ matrix (vector) that encodes the phases of the adjoint action of U_0 on every σ. In the right-hand side (r.h.s.) of Eq. (38), the notation $(U_0^\dagger \sigma U_0)_{x_i z_i}$ stands for the two bits corresponding to the ith component of the Pauli matrix on the ith qubit, while ϕ_i stands for the phase of $U_0^\dagger \sigma U_0$. For example, let $U_0^\dagger \sigma U_0 = \sigma^x \otimes \sigma^y \otimes \sigma^z \otimes \cdots \otimes \mathbb{1}$, then $(U_0^\dagger \sigma U_0)_{x_1 z_1} = (10)$, while $(U_0^\dagger \sigma U_0)_{x_3 z_3} = (01)$, etc. According to the lighter notation for the $2n$ bit string introduced above, $(P)_{\mathbf{xz}} \equiv (x_1, z_1, x_2, z_2, \ldots, x_n, z_n)$, the partial tableau $\tilde{T}_{U_0}$ in Eq. (38) can be written as

$$\tilde{T}_{U_0} = \begin{pmatrix} \left(U_0^\dagger \sigma_1^x U_0\right)_{\mathbf{xz}} \\ \left(U_0^\dagger \sigma_1^z U_0\right)_{\mathbf{xz}} \\ \vdots \\ \left(U_0^\dagger \sigma_n^x U_0\right)_{\mathbf{xz}} \\ \left(U_0^\dagger \sigma_n^z U_0\right)_{\mathbf{xz}} \end{pmatrix}. \tag{39}$$

Note that the generators $\sigma_i^{x,z} \in \mathfrak{l}$ in the tableau T_{U_0} are arranged so that $\{\sigma_i^x, \sigma_i^z\} = 0$ for any $i = 1, \ldots, n$ and that $[\sigma_i^{x,z}, \sigma_j^{x,z}] = 0$ for any $i, j = 1, \ldots, n$ and $j \neq i$. Let us set up the following notation for the rest of the paper: let A be a square matrix, then $[A]_\alpha$ denotes the αth row vector of A, for example, $[\tilde{T}_{U_0}]_1 = (U_0^\dagger \sigma_1^x U_0)_{\mathbf{xz}}$. Let us make an example.

Example 4. Consider the phase gate S, the partial tableau $\tilde{T}_S \equiv \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ is symplectic

$$T_S \Omega T_S^T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \Omega, \tag{40}$$

[1] Note that the matrix Ω presented in our work deviates from the usual representation, which is commonly expressed as a block off-diagonal matrix $\left(\begin{smallmatrix} 0 & I \\ I & 0 \end{smallmatrix}\right)$, consequence of a different choice for the basis of $\mathbb{F}_2^{2n}$. The usual representation can be recovered by setting $(P)_{\mathbf{xz}} \equiv (x_1, x_2, \ldots, x_n, z_1, z_2 \ldots, z_n)$.

where we used that the arithmetic is modulo 2. This simple example illustrates that the partial tableau $\tilde{T}_S$ corresponds to the unique symplectic matrix associated with the phase gate S.

The generating set of the Clifford group is given by the controlled NOT gate (CNOT), the Hadamard gate H, and the phase gate S. The action of these native gates is mapped to a matrix operation on the tableau T_{U_0} by looking at their action on a Pauli string $P \in \mathbb{P}$. The Hadamard gate H(i) acting on the qubit i results in swapping the x_ith and the z_ith component on the entire column, namely:

$$(P)_{x_i z_i} \xrightarrow{H(i)} (P)_{z_i x_i} \tag{41}$$

for all $i = 1, \dots, n$, $\phi_i = \phi_i \oplus x_i z_i$, while the phase gate $S(i)$ acting on the qubit i results in an XOR operation between the x_i and z_i:

$$(P)_{x_i z_i} \xrightarrow{S(i)} (P)_{x_i z_i \oplus x_i}, \tag{42}$$

for all $i = 1, \dots, n$, $\phi_i = \phi_i \oplus x_i z_i$. Finally, the CNOT$(k, i)$ having control qubit k and acting on the qubit i reads

$$(P)_{x_k z_k}(P)_{x_i z_i} \xrightarrow{\text{CNOT}(k,i)} (P)_{x_k z_i \oplus z_k}(P)_{x_i \oplus x_k z_i}. \tag{43}$$

for all $i = 1, \dots, n$, $\phi_i = \phi_i \oplus x_k z_i(x_i \oplus z_k \oplus 1)$. Let us conclude this paragraph by introducing the concept of a symplectic transformation as a map denoted by $\mathcal{S} : \mathrm{Sp}(2n, \mathbb{F}_2^{2n}) \to \mathrm{Sp}(2n, \mathbb{F}_2^{2n})$. This map transforms one tableau $\tilde{T}U_0$ into another tableau $\tilde{T}U_0'$. Despite the general nature of this transformation, since it is a mapping between two symplectic matrices, there always exists an element $T_{U_0''} \in \mathrm{Sp}(2n, \mathbb{F}_2^{2n})$ within the group such that $T_{U_0''} T_{U_0} = T_{U_0'}$. In the paper, while we introduce other symplectic transformations, our focus will be on determining the specific symplectic matrix (Clifford unitary) that accomplishes the desired task. We will not pursue a general mapping approach but rather identify the particular symplectic matrix that achieves the desired transformation.

Let us define the diagonalizer transformation $\mathcal{D}_{\mathfrak{h}}(\cdot)$, a symplectic transformation, that will be the core of the CC algorithm in Sec. VC. We first need to define the following encoding on a subset of generators.

Definition 1. $[\tau_{\mathfrak{h}}]$ Consider a set $\mathfrak{g}$ of generators of the Pauli group, and a subset $\mathfrak{h} \equiv \{g_1, g_2, \dots g_h\} \subset \mathfrak{g}$ with h elements. From a subset of generators $\mathfrak{h}$, we define the matrix $\tau_{\mathfrak{h}}$ as follows:

$$\tau_{\mathfrak{h}} \equiv \begin{pmatrix} (g_1)_{\mathbf{xz}} \\ (g_2)_{\mathbf{xz}} \\ \vdots \\ (g_i)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ (g_h)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \tag{44}$$

where $(g_i)_{\mathbf{xz}}$ corresponds to the $2n$-bit string $(x_1 z_1 \dots x_n z_n) \in \mathbb{F}_2^{2n}$ encoding the generator g_i, and $\mathbf{0}$ is a $2n$-bit string of zeros. The matrix $\tau_{\mathfrak{h}}$ is build from the subset of generators $\mathfrak{h}$ in the following way: if for a given $g_i \in \mathfrak{h}$ there is no $g_j \in \mathfrak{h}$ such that $\{g_i, g_j\} = 0$, the generator g_i is just followed by a null vector $\mathbf{0}$, otherwise g_i and g_j (such that $\{g_i, g_j\} = 0$) are placed in two consecutive rows. There can be many ways to build the matrix $\tau_{\mathfrak{h}}$; however, in this paper, we adopt the following convention: *paired* generators, which are couples of anticommuting generators, occupy the initial rows of the matrix $\tau_{\mathfrak{h}}$. The subsequent rows are filled with *unpaired* generators, each followed by a null vector $\mathbf{0}$. The remaining part of the matrix is just filled by null vectors $\mathbf{0}$ (see *Matrix initializer* for a step-by-step algorithm detailing how to construct this matrix given a subset of generators denoted as $\mathfrak{h}$). Let us provide a concrete example:

Example 5. Consider $n = 2$ and a set of generators $\mathfrak{g} = \{\sigma^x \otimes \sigma^x, \sigma^y \otimes \sigma^y, \sigma^z \otimes \sigma^x, \sigma^y \otimes \sigma^z\}$. Let $\mathfrak{g} \supset \mathfrak{h} = \{\sigma^x \otimes \sigma^x, \sigma^y \otimes \sigma^y, \sigma^z \otimes \sigma^x\}$. To construct $\tau_{\mathfrak{h}}$, note that $[\sigma^x \otimes \sigma^x, \sigma^y \otimes \sigma^y] = [\sigma^z \otimes \sigma^x, \sigma^y \otimes \sigma^y] = 0$, while $\{\sigma^x \otimes \sigma^x, \sigma^z \otimes \sigma^x\} = 0$. Therefore we can assign $[\tau_{\mathfrak{h}}]_1 = (\sigma^x \otimes \sigma^x) \equiv (1010)$, $[\tau_{\mathfrak{h}}]_2 = (\sigma^z \otimes \sigma^x) \equiv (0110)$, $[\tau_{\mathfrak{h}}]_3 = (\sigma^y \otimes \sigma^y) \equiv (1111)$, and $[\tau_{\mathfrak{h}}]_4 = \mathbf{0} \equiv (0000)$. Therefore, we can assign to $\mathfrak{h}$ the following matrix $\tau_{\mathfrak{h}}$:

$$\tau_{\mathfrak{h}} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{45}$$

Note that, the way the matrix $\tau_{\mathfrak{h}}$ is filled is the same as the partial tableau $\widetilde{T}_{U_0}$ with the only difference that the partial tableau encodes all the $2n$ generators $\in \mathfrak{g}$ of $\mathbb{P}$, while $\tau_{\mathfrak{h}}$ encodes only a subset $\mathfrak{h} \subset \mathfrak{g}$. This fact motivates us to define the following set of Boolean matrices:

Definition 2. Let $\mathcal{B}_{2n}$ the set of $2n \times 2n$ Boolean matrices. Define $\mathcal{T}_{2n} \subset \mathcal{B}_{2n}$ the set of matrices with the following properties:

(1) $\forall\ \tau \in \mathcal{T}_{2n}$, $\tilde{\omega}([\tau]_{2i+1}, [\tau]_j) = 0$ for $i = 1, \dots, n$ and $j = 1, \dots, 2n$, $j \neq 2i + 2$.

(2) $\forall\ \ \tau \in \mathcal{T}_{2n} \tilde{\omega}([\tau]_{2i+1}, [\tau]_{2i+2}) = \{^{1,\ \ \text{if}\ \ [\tau]_{2i+2} \neq \mathbf{0}}_{0,\ \ \text{else}}$ for $i = 1, \dots, n$.

where $\tilde{\omega}$ is an extension of the symplectic form ω applied to the rows of $\tau_{\mathfrak{h}}$.

Remark 1. Given a subset $\mathfrak{h} \subset \mathfrak{g}$ of generators $\mathfrak{g}$, then $\tau_{\mathfrak{h}} \in \mathcal{T}_{2n}$. The partial tableau $\tilde{T}_{U_0}$ corresponding to a Clifford circuit U_0 belongs to $\mathcal{T}_{U_0}$. Moreover, the $2n \times 2n$ identity matrix $I_{2n} \in \mathcal{T}_{2n}$.

Let us define the following $2n$-bit strings $e_\alpha \equiv (\delta_{\alpha 1}, \dots, \delta_{\alpha 2n})$, where $\delta_{\alpha\beta}$ is the Kronecker delta.

Definition 3 (Diagonalizer). Let $\mathfrak{h}$ be a subset of generators and $\tau_{\mathfrak{h}} \in \mathcal{T}_{2n}$ the corresponding matrix defined in Definition 1. The diagonalizer $\mathcal{D}_{\mathfrak{h}}(\cdot)$ is a map $\mathcal{D}_{\mathfrak{h}} : \mathcal{T}_{2n} \mapsto \mathcal{T}_{2n}$, whose action is defined as

$$[\mathcal{D}_{\mathfrak{h}}(\tau_{\mathfrak{h}})]_\alpha = \begin{cases} e_\alpha, & \text{if } [\tau_{\mathfrak{h}}]_\alpha = (g_i)_{\mathbf{xz}} \quad \text{for some } g_i \in \mathfrak{h} \\ \mathbf{0}, & \text{if } [\tau_{\mathfrak{h}}]_\alpha = \mathbf{0}. \end{cases} \tag{46}$$

In other words, the diagonalizer $\mathcal{D}_{\mathfrak{h}}(\cdot)$ maps $\tau_{\mathfrak{h}}$ to a partial identity matrix belonging to $\mathcal{T}_{2n} \subset \mathcal{B}_{2n}$:

$$\mathcal{D}_{\mathfrak{h}}(\tau_{\mathfrak{h}}) := \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \end{pmatrix}. \qquad (47)$$

In Appendix B, we introduce the algorithm that, given any matrix $\tau_{\mathfrak{h}} \in \mathcal{T}_{2n}$, performs the diagonalizer in time $O(n^2)$ in terms of symplectic transformations. As a consequence, the diagonalizer $\mathcal{D}_{\mathfrak{h}}$ itself is a symplectic transformation and therefore equivalent to a Clifford operator, denoted $\hat{\mathcal{D}}_{\mathfrak{h}}$, that maps the set $\mathfrak{h}$ in a subset of the set $\mathfrak{l}$ (the local generators of the Pauli group), i.e., $\hat{\mathcal{D}}_{\mathfrak{h}}^{\dagger}\mathfrak{h}\hat{\mathcal{D}}_{\mathfrak{h}} \equiv \{\hat{\mathcal{D}}_{\mathfrak{h}}^{\dagger}g_1\hat{\mathcal{D}}_{\mathfrak{h}}, \hat{\mathcal{D}}_{\mathfrak{h}}^{\dagger}g_2\hat{\mathcal{D}}_{\mathfrak{h}}, \dots, \hat{\mathcal{D}}_{\mathfrak{h}}^{\dagger}g_h\hat{\mathcal{D}}_{\mathfrak{h}}\} \subset \mathfrak{l}$. For example, consider the partial tableau $\widetilde{T}_{U_0}$ defined in Eq. (38), with set of generators $\mathfrak{g} := \{\sigma_1, \sigma_2, \dots, \sigma_{2n}\}$ such that $\{\sigma_{2i}, \sigma_{2i+1}\} = 0$ for all i, then the diagonalizer $\mathcal{D}_{\mathfrak{g}}$ on the partial tableau $\widetilde{T}_{U_0}$ acts as follows:

$$\mathcal{D}_{\mathfrak{g}}(\widetilde{T}_{U_0}) = \begin{pmatrix} (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_1U_0\hat{\mathcal{D}})_{x_1z_1} & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_1U_0\hat{\mathcal{D}})_{x_2z_2} & \dots & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_1U_0\hat{\mathcal{D}})_{x_nz_n} \\ (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_2U_0\hat{\mathcal{D}})_{x_1z_1} & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_2U_0\hat{\mathcal{D}})_{x_2z_2} & \dots & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_2U_0\hat{\mathcal{D}})_{x_nz_n} \\ \vdots & \vdots & \ddots & \vdots \\ (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_{2n}U_0\hat{\mathcal{D}})_{x_1z_1} & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_{2n}U_0\hat{\mathcal{D}})_{x_2z_2} & \dots & (\hat{\mathcal{D}}^{\dagger}U_0^{\dagger}\sigma_{2n}U_0\hat{\mathcal{D}})_{x_nz_n} \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{2n} \end{pmatrix} = I_{2n}, \qquad (48)$$

where I_{2n} is the $2n \times 2n$ identity matrix. We remark here that, in the above case, one has $\hat{\mathcal{D}}_{\mathfrak{g}} \equiv U_0^{\dagger}$ and the diagonalizer is unique up to a phase. Before concluding the section, let us give a basic example of diagonalizer.

Example 6. Let $n = 2$. Consider the subset of generators $\mathfrak{h} = \{\sigma^x \otimes \sigma^x, \sigma^y \otimes \sigma^y, \sigma^z \otimes \sigma^x\}$. In Eq. (45), we computed the matrix $\tau_{\mathfrak{h}} \in \mathcal{T}_4$. The diagonalizer $\mathcal{D}_{\mathfrak{h}}$ acting on $\tau_{\mathfrak{h}}$ results in

$$\mathcal{D}_{\mathfrak{h}}\left[\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad (49)$$

and corresponds to the Clifford operator $\hat{\mathcal{D}}_{\mathfrak{h}} = H(0)H(1)\textsc{cnot}(1,2)H(1)\textsc{cnot}(1,2)$.

B. The structure of doped Clifford circuits: Compression theorems, gate complexity, and learnability

Equipped with the notions introduced in the previous section, here we discuss the structure of t-doped Clifford circuits, i.e., Clifford circuits doped with a finite number t of (single-qubit) non-Clifford gates, which for simplicity are considered T gates.[2] Denote as U_t a t-doped Clifford circuit. Let $P \in \mathbb{P}$ be a Pauli operator. In general, $U_t^{\dagger}PU_t \notin \mathcal{P}$. However, there exists a subgroup $G(U_t) \subset \mathbb{P}$ of the Pauli group such that $U_t^{\dagger}PU_t \in \mathcal{P}$ for every $P \in G(U_t)$. The group $G(U_t)$ is defined as

$$G(U_t) = \{P \in \mathbb{P} | U_t^{\dagger}PU_t \in \mathcal{P}\}, \qquad (50)$$

whose cardinality is lower-bounded by $|G(U_t)| \geqslant 2^{2n-t}$ [79], i.e., at most a fraction of 2^t Pauli operators is not preserved by the action of U_t. Note that this notion is tied to the one introduced in Sec. IV C, where we discussed the number of preserved Pauli operators with support on a subspace D. Let $G_D(U_t)$ be the group defined in Eq. (19), clearly we have $G_D(U_t) \subset G(U_t)$. Since $G(U_t)$ is a subgroup of the Abelian group $\mathbb{P}$, it is finally generated by a subset $g(U_t) \subset G(U_t)$ of generators, whose cardinality is lower-bounded by $|g(U_t)| \geqslant 2n - t$. We denote with brackets $\langle \cdot \rangle$ the generating operation, e.g., $G(U_t) = \langle g(U_t) \rangle$. Thanks to the unitarity of U_t, there exists a set of Clifford operations $\mathcal{V}_{U_t}$ defined as[3]

$$\mathcal{V}_{U_t} := \{V \in \mathcal{C}(n) | V^{\dagger}PV = U_t^{\dagger}PU_t, \ \forall\, P \in G(U_t)\}. \qquad (51)$$

After all, the action of U_t on the group $G(U_t)$ is Clifford-like and can be replicated by some Clifford operations. Since $G(U_t)$ is a subgroup of the Pauli group with cardinality $|G(U_t)|$, there exists an integer s and a Clifford operation $\hat{\mathcal{D}}_{g(U_t)}$ such that $\mathbb{P}([s]) \subset \hat{\mathcal{D}}_{g(U_t)}^{\dagger}G(U_t)\hat{\mathcal{D}}_{g(U_t)}$, where $\mathbb{P}([s])$ denote the local Pauli group on a system $[s]$ containing s qubits. Note that the Clifford operation $\hat{\mathcal{D}}_{g(U_t)}$ introduced above is exactly the Clifford operation corresponding to the symplectic diagonalizer operation $\mathcal{D}_{g(U_t)}$ defined in Definition 3. In particular, let $\tau_{g(U_t)}$ be the matrix corresponding to $g(U_t)$, then $[\mathcal{D}_{g(U_t)}(\tau_{g(U_t)})]_{\alpha} = e_{\alpha}$ if $[\tau_{g(U_t)}]_{\alpha} \neq \mathbf{0}$. Let $\hat{\mathcal{D}}_{g(U_t)}$ be the Clifford operator corresponding to the diagonalizer, then $\hat{\mathcal{D}}_{g(U_t)}$ acts on the subset $g(U_t) \subset \mathfrak{g}$ (for some set of generators $\mathfrak{g}$)

[2] The extension of this discussion to arbitrary single-qubit non-Clifford gates is straightforward.

[3] The reason why there are Clifford unitaries capable of emulating a general unitary U_t on $G(U_t)$ can be explained as follows: as shown in Sec. V A, Clifford unitaries (as well as all the unitaries) are solely characterized by their action on Pauli operators. In particular, Clifford circuits map Pauli operators to Pauli operators with the only condition of preserving commutation relations between them. Since the conditions imposed by the unitary U_t on $G(U_t)$ pertain to Pauli operators and, due to the unitarity of U_t, preserve their commutation relation.

and transforms it to a subset of the local generating set $\mathfrak{l}$ of the Pauli group. The integer s obeys the following lower bound:

$$s \geqslant n - t, \tag{52}$$

i.e., in the worst case, only a local Pauli group on t qubits is not preserved by the adjoint action of U_t. Equation (52) easily descends from the fact that $|g(U_t)| \geqslant 2n - t$: the set $g(U_t)$ is a subset of $\mathfrak{g}$, which consists of a set of generators containing n pairs of anticommuting generators. As the cardinality of $g(U_t)$ is bounded from below by $2n - t$, it implies that $g(U_t)$ contains at least $n - t$ pairs of generators from $\mathfrak{g}$. These pairs, when diagonalized by $\mathcal{D}_{g(U_t)}$, correspond to the local generating set of the Pauli group $P([s])$ of a subsystem consisting of s qubits, where s is greater than or equal to $n - t$. The above considerations allow us to decompose any unitary U_t in Clifford blocks plus a non-Clifford operation acting on (at most) t qubits. To see this, let us show that for every $V \in \mathcal{V}_{U_t}$, the unitary operator given by the product $\hat{\mathcal{D}}^\dagger_{g(U_t)} U_t V^\dagger \hat{\mathcal{D}}_{g(U_t)}$ acts identically on s qubits. Let $P_s \in \mathbb{P}([s]) \subset \hat{\mathcal{D}}^\dagger_{g(U_t)} G(U_t) \hat{\mathcal{D}}_{g(U_t)}$ a local Pauli operator on s qubits, then

$$\hat{\mathcal{D}}^\dagger_{g(U_t)} V U_t^\dagger \hat{\mathcal{D}}_{g(U_t)} P_s \hat{\mathcal{D}}^\dagger_{g(U_t)} U_t V^\dagger \hat{\mathcal{D}}_{g(U_t)} = P_s; \tag{53}$$

indeed, $\hat{\mathcal{D}}_{g(U_t)} P_s \hat{\mathcal{D}}^\dagger_{g(U_t)} \in G(U_t)$ by definition, $V U_t^\dagger P U_t V^\dagger = P$ for every $P \in G(U_t)$ thanks to Eq. (51), and $\hat{\mathcal{D}}_{g(U_t)}$ sends P back to P_s. This remarkable fact means that $\hat{\mathcal{D}}^\dagger_{g(U_t)} U_t V^\dagger \hat{\mathcal{D}}_{g(U_t)} = \mathbb{1}_{[s]} \otimes u_{[n-s]}$ for some local unitary $u_{[n-s]}$ acting on $n - s$ qubits where $n - s \leqslant t$ [see Eq. (52)]. Moreover, note that the unitary $u_{[n-s]}$ contains all the T gates, since $\hat{\mathcal{D}}_{g(U_t)}$, $V \in \mathcal{C}(n)$, and thus is a t-doped Clifford circuit on $(n - s)$ qubits. We denoted $\mathbb{1}_{[s]}$ the identity matrix acting on the subsystem $[s]$ containing s qubits. All the above considerations are summarized in Eq. (1), as well as in the following theorem, which is one of the main results of the paper.

Theorem 2 (Compression theorem). Let U_t a t-doped Clifford circuit and $V \in \mathcal{V}_{U_t}$. There exists a integer $s \geqslant n - t$, a subset $[s]$ of s qubits and a Clifford operation $\hat{\mathcal{D}}_{g(U_t)}$—where $\mathcal{D}_{g(U_t)}$ is the diagonalizer defined in Definition 3—that allows the following decomposition for U_t:

$$U_t = \hat{\mathcal{D}}_{g(U_t)} (\mathbb{1}_{[s]} \otimes u_{[n-s]}) \hat{\mathcal{D}}^\dagger_{g(U_t)} V \tag{54}$$

for some unitary operator $u_{[n-s]}$ acting on $(n - s)$ qubit and containing at most t non-Clifford (T) gates.

The above result, besides giving strong insights on the structure of t-doped Clifford circuit, as discussed in Sec. II, allows us to bound gate complexity of t-doped Clifford circuits. The gate complexity $\#(U)$ of a unitary operator U is defined as the minimum number of elementary gates, chosen from a certain universal subset (such as, e.g., {H, CNOT, T}), necessary to build U from the identity. Given that any Clifford unitary operator acting on n qubits can be distilled using $O(n^2)$ gates [83], a simplistic upper bound for the gate complexity would be $O(tn^2)$. This is because the structure of a t-doped Clifford circuit can always be seen as Clifford circuits interleaved by t non-Clifford (T) gates, leading to the crude bound provided. If $t \leqslant n$, the bound can be improved by employing the techniques of *Clifford compression* derived in Theorem 2. We know that $\#(V) = O(n^2)$, $\#(\hat{\mathcal{D}}_{g(U_t)}) = \mathcal{O}(n^2)$, $\#(u_{[n-s]}) = \mathcal{O}(t^3)$, because $n - s \leqslant t$. From this fact, the following corollary of Theorem 2 readily descends:

Corollary 1. Let U_t be a t-doped Clifford circuit. The gate complexity $\#(U_t) = O(n^2 + t^3)$.

Moreover, as expected, the compression theorem above gives an analogous compression result for t-doped stabilizer states, i.e., states $|\psi_t\rangle$ obtained from a stabilizer initial state $|\sigma\rangle$ by the action of U_t.

Corollary 2. Consider a t-doped Clifford circuit U_t and the t-doped stabilizer state obtained $|\psi_t\rangle = U_t\,|0\rangle_{[n]}$. Then, there exists a choice of the diagonalizer of Eq. (54) of Theorem 2, denoted $\tilde{\mathcal{D}}_{g(U_t)}$, such that the state $|\psi_t\rangle$ can be compressed as

$$|\psi_t\rangle = \tilde{\mathcal{D}}_{g(U_t)} (|0\rangle_{[s]} \otimes |\phi\rangle_{[n-s]}), \tag{55}$$

where $|\phi\rangle_{[n-s]}$ is a quantum state defined on the system $[n - s]$ of $n - s \leqslant t$ qubits.

Proof. Call $G_{V|0\rangle}$ the stabilizer group of the stabilizer state $V\,|0\rangle$. The only thing to note is that, exploiting the freedom in defining the diagonalizer $\mathcal{D}_{g(U_t)}$, it is possible to define a diagonalizer, denoted $\tilde{\mathcal{D}}_{g(U_t)}$, such that the group $G(U_t) \cap G_{V|0\rangle} \subseteq G(U_t)$ is mapped in a local Pauli group $\mathbb{Z}[s]$, where $s \geqslant n - t$. In formulas, $\tilde{\mathcal{D}}^\dagger_{g(U_t)} G(U_t) \cap G_{V|0\rangle} \tilde{\mathcal{D}}_{g(U_t)} \subseteq \mathbb{Z}[s]$. This is because $G(U_t) \cap G_{V|0\rangle}$ is a commuting subgroup of $G(U_t)$ with cardinality lower-bounded by 2^{n-t} and thus it is always possible to design a diagonalizer with this desired property (see Sec. V A). Therefore, choosing $\tilde{\mathcal{D}}_{g(U_t)}$ we have the following chain of identities:

$$\begin{aligned} U_t\,|0\rangle &= \tilde{\mathcal{D}}_{g(U_t)} \mathbb{1}_{[s]} \otimes u_{[n-s]} \tilde{\mathcal{D}}^\dagger_{g(U_t)} V\,|0\rangle \\ &= \tilde{\mathcal{D}}_{g(U_t)} \mathbb{1}_{[s]} \otimes u_{[n-s]} (|0\rangle_{[s]} \otimes |\omega\rangle_{[n-s]}) \\ &= \tilde{\mathcal{D}}_{g(U_t)} (|0\rangle_{[s]} \otimes |\phi\rangle_{[n-s]}). \end{aligned} \tag{56}$$

To conclude the proof, we remark that the choice of the diagonalizer $\tilde{\mathcal{D}}_{g(U_t)}$ that annihilates the action of V on $|0\rangle_{[n]}$ strictly depends on the (stabilizer) input state. ■

The following theorem and its subsequent corollary establish the methodology for learning a t-doped Clifford circuit U_t.

Theorem 3. Let U_t be a t-doped Clifford circuit and let $G(U_t)$ the associated group, $g(U_t)$ its generating set, and $\mathcal{V}_{U_t} = \{V \in \mathcal{C}(n) | V^\dagger P V = U_t^\dagger P U_t,\ \forall\, P \in G(U_t)\}$. Then, with $\mathrm{poly}(n, 2^t)$ query accesses to U_t, the CC algorithm finds and efficiently encodes $g(U_t)$, $V \in \mathcal{V}_{U_t}$, and $\hat{\mathcal{D}}_{g(U_t)}$ in a time $\mathrm{poly}(n, 2^t)$. In particular, the CC algorithm finds the Clifford operations U_0 and U_0' in Eq. (1).

Proof. Call the CC algorithm in Sec. V C with $m = 0$. Thanks to Eq. (54), then $U_0 = \hat{\mathcal{D}}_{g(U_t)}$ and $U_0' = \hat{\mathcal{D}}^\dagger_{g(U_t)} V$.

The above theorems say that we can always decompose a t-doped Clifford circuit in a product of Clifford operations and a local unitary acting on (at most) t qubits. Surprisingly, by employing a finite number of query accesses to U_t, one is able to isolate the non-Clifford gates and concentrate them into a local unitary acting on at most t qubits, and learn Clifford blocks of the decomposition in Eq. (1). Once again, this fact discriminates circuits where the number of T gates is less than or exceeds the number of qubits n: while for $t < n$ a *Clifford compression* is possible, for $t > n$ (in general) the circuit cannot be compressed as in Eq. (54). Let us conclude the section with the following corollary:

Corollary 3. Let U_t be a t-doped Clifford circuit, then using $\mathrm{poly}(n, 2^t)$ total resources including time complexity and query complexity to the unitary U_t, is possible to learn a full tomographic description of U_t.

Proof. From Theorem 2, we know that $U_t = U_0(\mathbb{1}_{[n-t]} \otimes u_{[t]})U_0'$ for $U_0 = \hat{\mathcal{D}}_{g(U_t)}$ and $U_0' = \hat{\mathcal{D}}_{g(U_t)}V$. From Theorem 3, the CC algorithm learns and synthesizes U_0 and U_0' with $\mathrm{poly}(n, 2^t)$ query accesses to U_t. As a consequence, $U_0^{\dagger}U_tU_0^{\dagger\prime}$ acts nontrivially on at most t qubits. This fact allows us to run a unitary process tomography that requires $\exp(t)$ resources [56]. ■

C. The learning Clifford completion algorithm

In this section, we present the Clifford completion (CC) algorithm. Let U_t be a t-doped Clifford circuit. Let m be a integer, $0 \leqslant m \leqslant n$, then one can define the following quantities for a t-doped Clifford circuit U_t:

$$G_{[n-m]}(U_t) = \{P \in \mathbb{P}([n-m]) | U_t^{\dagger}PU_t \in \mathcal{P}\},$$
$$g_{[n-m]}(U_t) \subset G_{[n-m]} |\, \langle g_{[n-m]}\rangle = G_{[n-m]}(U_t),$$
$$\mathcal{V}_{U_t}^{[n-m]} \equiv \{V \in \mathcal{C}(n) | V^{\dagger}PV = U_t^{\dagger}PU_t,\ \forall\, P \in G_{[n-m]}(U_t)\}.$$

Note that, for $m = n$, one has $G_{[n]}(U_t) \equiv G(U_t)$ defined in Eq. (50), $g_{[n]}(U_t) \equiv g(U_t)$ and $\mathcal{V}_{U_t}^{[n]} \equiv \mathcal{V}_{U_t}$. From Lemma 2, we have that the following facts hold:

(1) $|G_{[n-m]}(U_t)| \geqslant 2^{2(n-m)-t}$.

(2) $|g_{[n-m]}(U_t)| \geqslant 2(n-m) - t$.

(3) $G_{[n-m]}(U_t) \subset G_{[n-m']}(U_t)$ for $m > m'$.

(4) $\mathcal{V}_{U_t}^{[n-m]} \subset \mathcal{V}_{U_t}^{[n-m']}$for $m < m'$.

The CC algorithm is capable of learning $g_{[n-m]}(U_t)$, $\hat{\mathcal{D}}_{g_{[n-m]}}$, and $V \in \mathcal{V}_{U_t}^{[n-m]}$ corresponding to a t-doped Clifford circuit U_t by allowing query accesses to U_t (which correspond to apply multiple times the unitary U_t on a quantum register). In particular, the CC algorithm can learn (i) for $m = 0$ the decomposition of U_t as $U_t = U_0[\mathbb{1}_{n-t} \otimes u_t]U_0'$ (see Theorem 2), and (ii), as later discussed in Sec. V D, for $m = |C|$ the Clifford decoder for the information recovery protocol.

Tools. The algorithm will extensively utilize the tools presented in Sec. V A. In particular, it will make use of the matrix $\tau_{\mathfrak{h}} \in \mathcal{T}_{2n}$, as defined in Definition 1. This matrix can be systematically constructed from a subset $\mathfrak{h}$ of a generating set of the Pauli group, along with the diagonalizer transformation, which can be built out from any matrix in $\mathcal{T}_{2n}$. Therefore, we recommend that interested readers first familiarize themselves with the formalism presented in Sec. V A.

Main idea. Let us briefly explain the underlying idea behind the CC algorithm in a more technical fashion. First, define $\bar{g}_{[n-m]}(U_t)$ as the set of generators such that, together with $g_{[n-m]}(U_t)$, is able to generate all the Pauli group $\mathbb{P}([n-m])$ on $n-m$ qubits, i.e., $\mathbb{P}([n-m]) = \langle g_{[n-m]}(U_t) \cup \bar{g}_{[n-m]}(U_t)\rangle$. Every operator $V \in \mathcal{V}_{U_t}^{[n-m]}$ mocks the action of U_t on all $\sigma \in g_{[n-m]}(U_t)$, i.e., $V^{\dagger}\sigma V = U_t^{\dagger}\sigma U_t$; the action of V on every other $\sigma \notin g_{[n-m]}(U_t)$ is free and it is constrained only by the commutation relations with $\sigma \in g_{[n-m]}(U_t)$. This shows that the set $\mathcal{V}_{U_t}^{[n-m]}$ contains more than one element. Thus, the algorithm needs first to search the generating set $g_{[n-m]}(U_t) \cup \bar{g}_{[n-m]}(U_t)$, and then write the tableau corresponding to $V \in \mathcal{V}_{U_t}^{[n-m]}$ in such a generating basis, cf. Sec. V A. Naively, the search for this generating set is exponentially hard in $n-m$ because, in general, one should pick every Pauli operator P in $\mathbb{P}([n-m])$ and check whether $P \in G_{[n-m]}(U_t)$ or not, i.e., one should check whether P is preserved by the action of U_t. In what follows, we describe how to sample Pauli operators in a way that allows us to find the generating set $g_{[n-m]}(U_t) \cup \bar{g}_{[n-m]}(U_t)$ in $\mathrm{poly}(n, 2^t)$ steps. Once that $g_{[n-m]}(U_t) \cup \bar{g}_{[n-m]}(U_t)$ is found, the algorithm learns the Clifford-like action of U_t on every $\sigma \in g_{[n-m]}(U_t)$. The algorithm thus generates one instance of $V \in \mathcal{V}_{U_t}^{[n-m]}$ uniformly at random. Implementing the diagonalizer $\mathcal{D}_{g_{[n-m]}}$, defined in Definition 3, on $\tau_{g_{[n-m]}}$ [where $g_{[n-m]}$ is a short notation for $g_{[n-m]}(U_t)$], defined in Definition 1, the algorithm builds the Clifford operation $\hat{\mathcal{D}}_{g_{n-m}}$. In the case $m = 0$, it thus finds $U_0 \equiv \hat{\mathcal{D}}_{g(U_t)}$ and $U_0' \equiv \hat{\mathcal{D}}_{g(U_t)}^{\dagger}V$.

Sampling Pauli operators. Let us describe the sampling method that allows us to find a generating set $g_{[n-m]}(U-t)$. Let $\mathfrak{g}_{k-1} \equiv \{g_1, \dots, g_{N_{k-1}}\} \subset g_{[n-m]}(U_t)$ be the set of N_{k-1} generators of $G_{[n-m]}(U_t)$ already found by the algorithm after $k-1$ steps. Note that, in general, $N_{k-1} \leqslant 2(k-1)$. Consider the $2n \times 2n$ matrix $\tau_{\mathfrak{g}_{k-1}}$ corresponding to the subset of generators $\mathfrak{g}_{k-1}$. Let $\mathcal{D}_{\mathfrak{g}_{k-1}}$ be the diagonalizer acting on $\tau_{\mathfrak{g}_{k-1}}$ as $[\mathcal{D}_{\mathfrak{g}_{k-1}}(\mathfrak{g}_{k-1})]_{\alpha} = e_{\alpha}$ if $[\tau_{\mathfrak{g}_{k-1}}]_{\alpha} \in \mathfrak{g}_{k-1}$, otherwise $[\mathcal{D}_{\mathfrak{g}_{k-1}}(\tau_{\mathfrak{g}_{k-1}})]_{\alpha} = \mathbf{0}$ (see Definition 3). Let $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$ be the Clifford unitary operator associated with the diagonalizer $\mathcal{D}_{\mathfrak{g}_{k-1}}$. Define $\widetilde{G}_{k-1} := G_{[n-m]}(U_t)/\langle \mathfrak{g}_{k-1}\rangle$ the quotient group of $G_{[n-m]}(U_t)$ respect to the normal subgroup $\langle \mathfrak{g}_{k-1}\rangle$. The operator $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$ maps the group $\widetilde{G}_{k-1}$ to a subgroup of the Pauli group $\mathbb{P}([n-m-k+1])$ on $n-m-k+1$ qubits, i.e., $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}^{\dagger}\widetilde{G}_{k-1}\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}} \subset \mathbb{P}([n-m-k+1])$; this is because, by construction [see Eq. (47)], the diagonalizer maps the generators $\mathfrak{g}_{k-1}$ in a subset of the local generators $\mathfrak{l}$ of the Pauli group on the first $k-1$ qubits. Therefore, to find the generators of the Pauli group $\mathbb{P}([n-m])$ containing $g_{[n-m]}(U_t)$, we sample a random Pauli operator on $n-m-k+1$ qubits, say P^x, and then check whether $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}P^x\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}^{\dagger}$ belongs to $G_{[n-m]}(U_t)$. In this way, we are sure that $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}P^x\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}^{\dagger} \notin \langle \mathfrak{g}_{k-1}\rangle$.

The diagonalizer thus allows us to extract independent Pauli operators at every step, making the effort to find the generators $g_{[n-m]}(U_t)$ exponentially hard in t, rather than in $n-m$. To see this, let us compute the probability of finding a Pauli operator belonging to $G_{[n-m]}(U_t)$ at the kth step. The cardinality of the quotient group is

$$\begin{aligned} |\widetilde{G}_{k-1}(U_t)| &= \frac{|G_{[n-m]}(U_t)|}{|\langle \mathfrak{g}_{k-1}\rangle|} \\ &\geqslant \frac{2^{2n-2m-t}}{2^{N_{k-1}}} \\ &\geqslant 2^{2(n-m-k+1)-t}. \end{aligned} \tag{57}$$

The first inequality follows from Lemma 4, and the second inequality follows from the fact that $|\langle \mathfrak{g}_{k-1}\rangle| \leqslant 4^{k-1}$. The probability that the extracted Pauli operator belongs to $\widetilde{G}_{k-1}$ is therefore

$$\begin{aligned} \Pr[\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}P^x\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}^{\dagger} \in \widetilde{G}_{k-1}] &= \frac{|\widetilde{G}_{k-1}| - 1}{4^{n-k+1}} \\ &\geqslant 2^{-t} - 4^{-n+m+k-1}, \end{aligned} \tag{58}$$

TABLE I. Sketch of the subroutines used in the CC algorithm, see Appendix B for details.

Subroutine	Input	Output	Time Complexity	Description
Diagonalizer	$\tau_{\mathfrak{h}}$	$\mathcal{D}_{\mathfrak{h}}, \hat{\mathcal{D}}_{\mathfrak{h}}$	$O(n^2)$	It transforms the matrix $\tau_{\mathfrak{h}}$ to the partial identity, see Eq. (46), saving the Clifford operator $\hat{\mathcal{D}}_{\mathfrak{h}}$.
Constrained random Clifford	$\tau_{\mathfrak{h}}, \phi_{\mathfrak{h}}$	T_{U_0}	$O(n^3)$	It generate a tableau T_{U_0} corresponding to the Clifford U_0 constrained by mapping some local generators to the generators $\mathfrak{h}$. The remaining local generators are mapped into random ones.
Learning a Pauli string	$U, P \in \mathbb{P}$	$Q \in \mathbb{P}$	$O(n^2M)$	It aims at learning $U^\dagger PU$ using $O(nM)$ queries to U. It learns a Pauli string Q regardless that P is preserved by U. The algorithm fails with probability $O(n2^{-M})$.
Verification and removal	P, Q	Yes or no	$O(2^{5t})$	It checks whether the learned Pauli operator Q via the *Learning a Pauli string* subroutine is accurate and, consequently, whether P is preserved.
Phase check	$P, Q \equiv \pm U^\dagger PU$	±1	$O(n^3)$	It learns the phase of the adjoint action of U on a given Pauli operator P using one single query to U.

i.e., the total dimension of the set we want to pick an element from (i.e., $\widetilde{G}_{k-1} \setminus \mathbb{1}$) divided by the dimension of the set we are sampling Pauli operators from [i.e., $\mathbb{P}([n-m-k+1])$]. Note that, the -1 is neglecting the identity. The (bound on the) above probability becomes zero for $2k = 2n - 2m - t + 2$; that is because the algorithm already found a maximum number of generators [recall indeed that $g_{[n-m]}(U_t) \geqslant 2(n-m) - t$, see Lemma 4].

In summary using the above sampling method, the probability of extracting a valid Pauli operator is lower-bounded by $3/2^{-(t+2)}$ for every step and, sampling $2^{t+2}/3 \times n$ number of Pauli operators in $\mathbb{P}([n-m-k+1])$ (given that $k \leqslant n - m - t/2$), one has an overwhelming probability, i.e., $\geqslant 1 - \exp(-n)$, to extract a P^x such that $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}} P^x \hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} \in G_{[n-m]}(U_t)$.

Subroutines. The upcoming algorithm will utilize several subroutines, namely, diagonalizer, Constrained Random Clifford, Learning a Pauli string, Verification and Removal and Phase Check, which are described in the Appendix B. The decision to present these subroutines in the Appendix is to enhance the readability of the algorithm. Nonetheless, the input, output, and time complexity are summarized in Table I for clarity. The algorithm is shown below.

```
Input: $n,\ m < n,\ U_t$.
Output: $g_{[n-m]}(U_t),\ V \in \mathcal{V}^{[n-m]}(U_t),\ \hat{\mathcal{D}}_{g_{[n-m]}}$.
  Let $\widetilde{T}_V$ a $2n \times 2n$ Boolean matrix such that $[\widetilde{T}_V]_\alpha = \mathbf{0}$ for any $\alpha = 1, \ldots, 2n$, $\boldsymbol{\phi} = \mathbf{0}$, $\mathfrak{g}_0 \equiv \emptyset$, and $\hat{\mathcal{D}}_{\mathfrak{g}_0} \equiv \mathbb{1}$ and $k = 1$.
      (I) While $k \leqslant n - m$, do:
          (i) For $M''$[4] times do:
              (a) Extract a random Pauli operators on $(n-m) - k + 1$ qubits $p^x \in \mathbb{P}([n-m-k+1])$ and define $P^x := \mathbb{1}_{[m+k-1]} \otimes p^x$.
              (b) Use Learning a Pauli string subroutine to learn the adjoint action of $U_t$ on $\hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} P^x \hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$. Denote the learned Pauli string as $P^x(V)$, and the corresponding encoding string as $(P^x(V))_{\mathbf{xz}}$.
              (c) Use Verification and Removal subroutine on $P_x$, $P^x(V)$ to check whether $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}} P^x_k \hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} \in G_{[n-m]}(U_t)$.
              (d) If $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}} P^x_k \hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} \in G_{[n-m]}(U_t)$ quit the while-loop and go to step (iii).
          (ii) Go to step (II).
          (iii) Use the Phase Check subroutine to read the phase $\phi_{2k-1}$ of $\hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} P^x_k \hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$.
          (iv) Update:
                   $[\widetilde{T}_V]_{2k-1} \mapsto [\widetilde{T}_V]_{2k-1} \oplus (P^x_k(V))_{\mathbf{xz}}$,
                   $\mathfrak{g}_{k-1} \mapsto \mathfrak{g}_{k-1} \cup \hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} P^x_k \hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$,
                   $(\boldsymbol{\phi})_{2k-1} \mapsto (\boldsymbol{\phi})_{2k-1} \oplus \phi_{2k-1}$.
          (v) While $m < M''$, do:
              (a) Extract a random Pauli operator $p^z \in \mathbb{P}([n-m-k+1])$ such that $\{p^x, p^z\} = 0$ denote $P^z \equiv \mathbb{1}_{[m+k-1]} \otimes p^z$, and do steps $(b)$ and $(c)$ of the algorithm with $P^z$ instead of $P^x$.[5]
              (b) if $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}} P^z \hat{\mathcal{D}}^\dagger_{\mathfrak{g}_{k-1}} \in G_{[n-m]}(U_t)$, quit the for-cycle and go to step (vii).[6]
```

[4] Note that, if the algorithm does not find a valid P^x_D such that $\hat{\mathcal{D}}_{k-1} P^x_D \hat{\mathcal{D}}^\dagger_{k-1} \in G_D(U_t)$ after $2^{t+2}/3\,n$, we can say that the algorithm found all the generators of $G_D(U_t)$ with probability $\geqslant 1 - \exp(-n)$.

[5] The probability of extraction an anticommuting Pauli operator is $1/2$.

[6] Let us compute the probability of success. There are two cases to discuss: (i) either $\nexists\ Q \in G_D(U_t)$ such that $\{Q, \hat{\mathcal{D}}_{k-1} P^x_{D_k} \hat{\mathcal{D}}^\dagger_{k-1}\} = 0$, or (ii) it needs to be found. It is easy to be convinced that, in the second

(vi) Go to step (ix).

(vii) Use the *Phase Check* subroutine to learn the phase ϕ_{2k} of $P_k^z(U_t)$.

(viii) Update:

$$[T_V]_{2k} \mapsto [T_V]_{2k} \oplus (P_k^z(V))_{\mathbf{xz}},$$
$$\mathfrak{g}_{k-1} \mapsto \mathfrak{g}_{k-1} \cup \hat{D}^{\dagger}_{\mathfrak{g}_{k-1}} P^z \hat{D}_{\mathfrak{g}_{k-1}},$$
$$(\boldsymbol{\phi})_{2k} \mapsto (\boldsymbol{\phi})_{2k} + \phi_{2k}.$$

(ix) Use the *diagonalizer* subroutine on $\mathfrak{g}_{k-1}$, and denote $\hat{\mathcal{D}}_{\mathfrak{g}_{k-1}}$ the Clifford circuit corresponding to the diagonalizer $\mathcal{D}_{\mathfrak{g}_{k-1}}$ [cf. Eq. (46)].

(x) $k \to k+1$.

(II) Use the *Constrained Random Clifford* subroutine on the tableau $T_V \equiv (\widetilde{T}_V \,|\, \boldsymbol{\phi})$ to extract a random constrained Clifford V. Define

$$g_{[n-m]}(U_t) \equiv \mathfrak{g}_k,$$
$$\hat{\mathcal{D}}_{g_{[n-m]}(U_t)} \equiv \hat{\mathcal{D}}_{\mathfrak{g}_k},$$
$$\mathcal{V}_{U_t}^{[n-m]} \ni V \mapsto \hat{\mathcal{D}}^{\dagger}_{g_{[n-m]}} V. \tag{59}$$

The above algorithm builds a random instance in $\mathcal{V}_{U_t}^{[n-m]}$ and finds the generating set $g_{[n-m]}(U_t)$ and the diagonalizer $\mathcal{D}_{g_{n-m}}$. The algorithm runs in time $O(n^5 2^{6t})$. Specifically, the time complexity is $O(nM''(n^2M + 2^{5t}))$, while the query complexity is $O(nM''(nM + 2^{5t}))$, where M is the number of shot measurements for *Learning a Pauli string* subroutine which fails with probability $O(n2^{-M})$. Thus, choosing $M'' = \frac{1}{3}2^{t+2}n$, and $M = n$, one has time complexity $O(n^5 2^{6t})$, query complexity scaling as $O(n^4 2^{6t})$ and exponentially small probability of failure $O(\text{poly}(n)2^{-n})$, obtained by the union bound.

D. Random Clifford decoder generation based on Clifford completion algorithm

In this section, we show the quantum algorithm based on CC capable of learning a Clifford decoder V for the information recovery protocol by means of poly$(n, 2^t)$ queries to U_t.

We want to remark once again that the CC algorithm finds a Clifford decoder: we first construct the tableau T_V corresponding to the decoder V, and subsequently, we employ the distillation algorithm described in Ref. [77] to distill the circuit in terms of CNOT, S, H. Recall the definition of the group $G_D(U_t)$, given in Sec. IV C, $G_D(U_t) := \{P_D \in \mathbb{P}(D) \,|\, U_t^{\dagger} P_D U_t \in \mathcal{P}\}$, i.e., the set of all the Pauli operators (defined on D) sent by the adjoint action of U_t in Pauli operators $\mathcal{P}$. We want the decoder V to mock the action of U_t on all $\sigma \in g_D(U_t)$, i.e., $V^{\dagger}\sigma V = U_t^{\dagger}\sigma U_t$ for any $\sigma \in g_D(U_t)$; for any other $\sigma \notin g_D(U_t)$ we choose the action of V at random. As in Eq. (V C), we define $\mathcal{V}_{U_t}^{D}$ (see Theorem 1) as

$$\mathcal{V}_{U_t}^{D} = \left\{V \in \mathcal{C}(n) | V^{\dagger}PV = U_t^{\dagger}PU_t,\ \forall\, P \in G_D(U_t)\right\}. \tag{60}$$

Since $G_D(U_t) \subset G(U_t)$, we clearly have $\mathcal{V}_{U_t} \subset \mathcal{V}_{U_t}^{D}$. We can run the CC-algorithm, presented in Sec. VC, and learn a decoder $V \in \mathcal{V}_{U_t}^{D}$ for $m = |C|$. Once again, the reasons why we look for a decoder in $\mathcal{V}_{U_t}^{D}$, instead of one in $\mathcal{V}_{U_t}$, is twofold:

(1) The entire CC algorithm can be run by an observer that has access to the subsystem D only, thus making the whole CC algorithm suitable for the information unscrambling problem.

(2) The set $\mathcal{V}_{U_t}^{D}$ is way larger than $\mathcal{V}_{U_t}$, having $2|C| + t$ unconstrained rows instead of t. The CC algorithm draws a decoder $V \in \mathcal{V}_{U_t}^{D}$ at random, making the probability of learning $\mathscr{P}(V)$ in Eq. (4) exponentially close to one in the size of C.

E. Numerics

In this section, we perform numerical tests on the CC algorithm to probe the accuracy of the scalings we derived in Sec. V C. Due to the computational cost of the CC algorithm on a classical computer, we do not specifically test the scaling with the number of qubits. Instead, we focus on providing the scaling in terms of t, that is the number of non-Clifford gates used within a Clifford circuit. For the scrambler U_t, we utilize a T-depth 1 doped Clifford circuit model, which consists of a non-Clifford unitary acting on $t/2$ qubits, sandwiched between two deep Clifford circuits. We chose this specific architecture for the sake of numerical simulation convenience. However, it is important to note that, according to Theorem 2, such architecture is applicable in an (almost) general sense for t-doped Clifford circuits. The non-Clifford unitary circuit consists of t T gates and is constructed as $\prod_{i=1}^{t/2} T_i H_i T_i$ for even t while $(\prod_{i=1}^{(t-1)/2} T_i H_i T_i) T_{\frac{t+1}{2}}$ for odd t, where T_i, H_i are Hadamard and T gates acting on the ith qubit. On the other hand, the deep Clifford circuits are randomly generated. The simulations were executed on a standard laptop, thus we set the values of $n = 8$ and $|D| = |C| = 4$, while $|A| = 1$ and $|B| = 7$. For each value of $t = 0, 1, \ldots, 6$, we collect $N_{\text{sample}} = 100$ distinct samples of scramblers U_t.

In Fig. 2(a), we plot the average time taken by the CC algorithm to find a Clifford decoder V for the scrambler U_t. As shown in the inset, the estimated average time complexity is much more favorable with respect to the algorithm's worst case prediction (see Sec. VC), which is $O(2^{6t})$. In Fig. 2(b), we plot the averaged fidelity $\mathcal{F}_V(U_t)$ over multiple realizations of the scrambler U_t. The inset showcases the failure probability of finding perfect decoders that align perfectly with the lower bound stated in Eq. (4), i.e., $1 - \mathscr{P}(V) \leqslant 2^{t-8}$.

VI. CONCLUSIONS

The possibility of learning relevant features of complex quantum dynamics from its observable behavior is of crucial importance for the understanding of quantum many-body systems away from equilibrium, loss of coherence and control in quantum devices, quantum chaos, criticality, and black-hole physics, and the general understanding of what quantum complexity is [38]. In particular, scrambling and information retrieval in quantum circuits pose a set of challenging questions in this context [4,6,35,84–94], the most relevant of which is to what extent one can learn how to unscramble information having no previous knowledge of the scrambling dynamics

case, if such a Pauli operator exists then the probability to be found is again lower-bounded by 2^{-t}, and choosing $M'' = 2^{t+2}/3\,n$ one has $1 - \exp(-4/3n)$ probability to success after M'' steps.

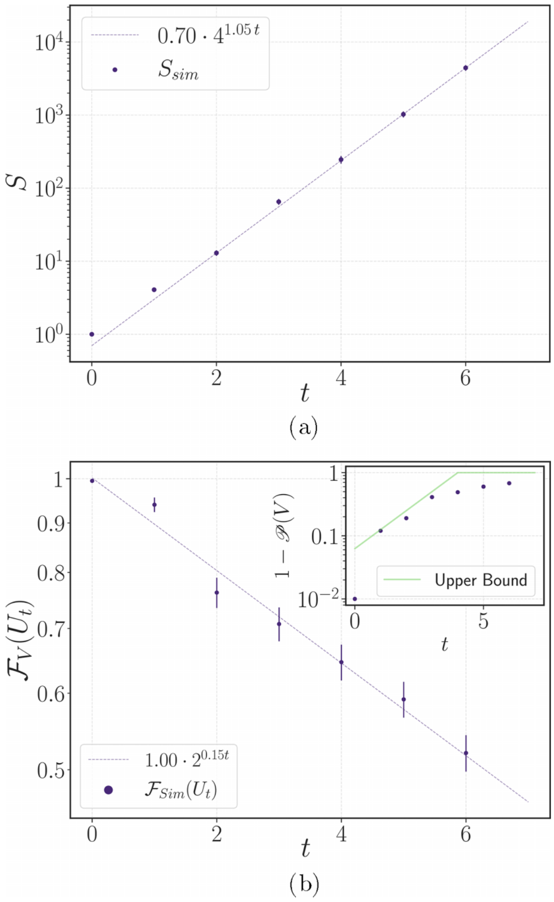


FIG. 2. The results of the CC algorithm for decoding scramblers U_t are depicted in this plot. The parameters used are $n = 8$, $|D| = |C| = 4$, $|A| = 1$, and $|B| = 7$. For each value of $t = 0, 1, \ldots, 6$, we collected $N_{\text{sample}} = 100$ different samples of scramblers U_t. Panel (a) illustrates the average number of steps S over N_{sample} taken by the CC algorithm to find a Clifford decoder V for each scrambler U_t. The legend reveals that the average number of steps $S = O(4^t)$ is much more favorable compared with the worst case time complexity estimated as $O(2^{6t})$, see Sec. VC. Panel (b) displays the averaged fidelity $\mathcal{F}_V(U_t)$ over N_{sample} realizations of the scrambler U_t. The inset highlights the failure probability, $1 - \mathscr{P}(V)$, of finding perfect decoders that perfectly match the lower bound stated in Eq. (4), where $1 - \mathscr{P}(V) \leqslant 2^{t-8}$.

and limited access (e.g., Hawking radiation) to the system. Unscrambling is achieved by means of a unitary operator called the *decoder*. One asks: is there always a decoder? Can it be found, and under which conditions? Is the decoder in itself efficient? The general answer to the existence of a decoder was given in Ref. [23]. The decoder exists provided the information is properly scrambled and one has sufficient access to the system. A suitable decoder requires complete knowledge of the scrambling dynamics. It is important to highlight that, even with this knowledge, this decoder is a complex quantum unitary operator and cannot in itself be efficiently simulated classically. This is hardly surprising: complex quantum dynamics must not be unscrambled classically, after all.

In this work, we explore the very ambitious problem of *learning an efficient decoder*. This means that, at the same time, we want to build the decoder by limited access to the system, without any previous knowledge of the internal dynamics, *and* have a decoder that is efficiently represented on a classical computer. One may think that this might be only possible if the scrambling dynamics is, in itself, classically simulable, for instance, the dynamics described by a random Clifford circuit: in this case, we indeed show an algorithm (CC) that can learn a perfect and efficient decoder with only polynomial resources. As the Clifford circuit is polluted by t non-Clifford gates, the cost of simulation grows exponentially in t. As t reaches a scaling with the number of qubits n, there is no efficient simulation. At this point, one would think that the system is chaotic, and one needs an exponential number of resources on a classical computer to perform the decoding even if the decoder is given [24]. The surprising result presented here is that the CC algorithm, in spite of requiring exponential resources to build the decoder, does build a decoder that is itself classical: it is a Clifford unitary that can be efficiently encoded in a classical computer. We see then that we are in a gray area where simulation and finding the decoder is hard, while the decoding itself is efficiently represented even on a classical computer and still achieves perfect recovery. Only when the doping crosses over to $t \simeq 2n$, the decoder loses its capability of retrieving information. At that point, there is real quantum chaos [29,39]: learning is hard, and what one has learned is hard to keep in a classical memory.

Why is this possible? How can a classical decoder be so good at retrieving quantum information that has been scrambled by a unitary dynamic that cannot be represented classically? Is this a contradiction in terms? First of all, this is not a contradiction. We do not learn the full scrambling dynamics. The gate fidelity between the decoder and the scrambler is strictly less than one. Indeed, one cannot turn a complex quantum unitary into a classical one. However, what we learn are the relevant features of the dynamics, defined in terms of being able to decode the scrambled information. And we find that, to some extent, these can be represented in a classical operator even for very complex (but not fully chaotic) quantum dynamics.

Again, how this is possible requires an explanation. The decoder might be acting like a quantum correction code, encoding away the non-Cliffordness in the part of the system that is inaccessible and this process is found to be possible until the onset of full-fledged quantum chaos. If this is true, then the amount of non-Cliffordness (i.e., *magic* [80,95–103]) shoved in the inaccessible part by the decoder must increase after the decoding.

Finally, one can ask: can one improve access to resources needed to find the decoder in the low doping case, by looking for decoders that are not Clifford? These operators must have, after all, better global gate fidelity. We believe the answer is no. Looking outside the Clifford group will pollute the search and make the search more complex, as it has been shown in the case of disentangling algorithms [40–42,48,49]. This kind of effect is also at play in the appearance of barren plateaus in VQAs even in the case of Clifford circuits as one tries to learn Clifford operations by using non-Clifford resources [55].

Quantum complexity is thus not necessarily featured in the number of elementary gates needed to decompose a unitary [38], but in the hardness of search problems, i.e., in the size of neighborhoods of target quantum states and processes, a point of view that is more reminiscent of Boltzmann's entropy [104]. All the above questions beg for an answer, and we believe they will be the source of very exciting future works.

ACKNOWLEDGMENTS

The authors thank Claudio Chamon for enlightening conversations that have been at the inception of this work and Bin Yan for a very useful remark about how to compute the success probability. We are also grateful to Dylan Lewis for helpful and long discussions in examining some technical details of the algorithm. Special thanks go to Lennart Bittel for fundamental inceptions for the proof of Lemma 7. L.L and S.F.E.O acknowledge support from NSF Award No. 2014000. A.H. acknowledges financial support from PNRR MUR Project No. PE0000023-NQSTI and PNRR MUR project CN 00000013-ICSC. The work of L.L. and S.F.E.O. was supported in part by the U.S. Department of Energy (DOE) through a quantum computing program sponsored by the Los Alamos National Laboratory Information Science & Technology Institute, and by the Center for Nonlinear Studies at Los Alamos National Laboratory (LANL).

APPENDIX A: NOTATION

In the following, we list the notations used throughout the paper:

(1) n: total number of qubits.

(2) $\mathcal{H}$: Hilbert space of n qubits.

(3) $\mathcal{P}$: Pauli group on n qubits.

(4) $\mathbb{P} := \mathcal{P}/\{\pm 1, \pm i\}$, is the group of Pauli strings, also referred to, with license, as just Pauli group. Note that $\mathbb{P} \neq \mathcal{P}$.

(5) $\mathcal{C}(n)$: Clifford group on n qubits.

(6) U_t: Clifford unitary operator doped with a number t of single qubit non-Clifford gates.

(7) $\mathcal{C}_t$ set of Clifford circuits doped with t single qubit non-Clifford gates.

(8) $O(U) \equiv U^\dagger O U$, where O is a operator on n qubits and U a unitary operator on n qubits.

(9) $|A|$: number of qubits in the subsystem A.

(10) A, B, C, D: subsystem of qubits such that $|A| + |B| = |C| + |D|$.

(11) $[m]$: subsystem of m qubits.

(12) $u_{[m]}$: local unitary acting on a subsystem $[m]$ of m qubits.

(13) $\mathbb{P}([m])$: a local Pauli group on a region $[m]$ of m qubits.

(14) $\langle f(P_A)\rangle_{P_A}$: average over the local Pauli group $P(A)$.

(15) Let G be a group. Then if $\mathfrak{G} \subset G$ is a generating set, we write $G = \langle \mathfrak{G} \rangle$.

(16) $\mathfrak{g} \subset \mathbb{P}$ denotes a generating set of the Pauli group $\mathbb{P}$.

(17) $\mathfrak{l} \equiv \{\sigma_i^x, \sigma_i^z\}_{i=1}^n$ is the local generating set of $\mathbb{P}$.

(18) Let $\mathfrak{h} \subset \mathfrak{g}$. Then $\tau_{\mathfrak{h}}$ is the $2n \times 2n$ Boolean matrix defined in Definition 1.

(19) Let $\mathfrak{h} \subset \mathfrak{g}$. Then $\mathcal{D}_{\mathfrak{h}}$ is the diagonalizer acting on $\mathfrak{h}$ defined in Definition 3.

(20) $\hat{\mathcal{D}}_{\mathfrak{h}}$: Clifford operator corresponding to the diagonalizer $\mathcal{D}_{\mathfrak{h}}$.

APPENDIX B: SUBROUTINES

In this section, we present the subroutines used throughout the paper.

(1) *Matrix initializer*
(2) *Sweeping*
(3) *Random Clifford Sampling*
(4) *Diagonalizer*
(5) *Constrained random Clifford*
(6) *Learning a Pauli string*
(7) *Verification and Removal*
(8) *Phase check*

Note that the algorithms for the subroutines *Sweeping* and *Random Clifford Sampling* have been introduced in Ref. [105].

1. Matrix initializer

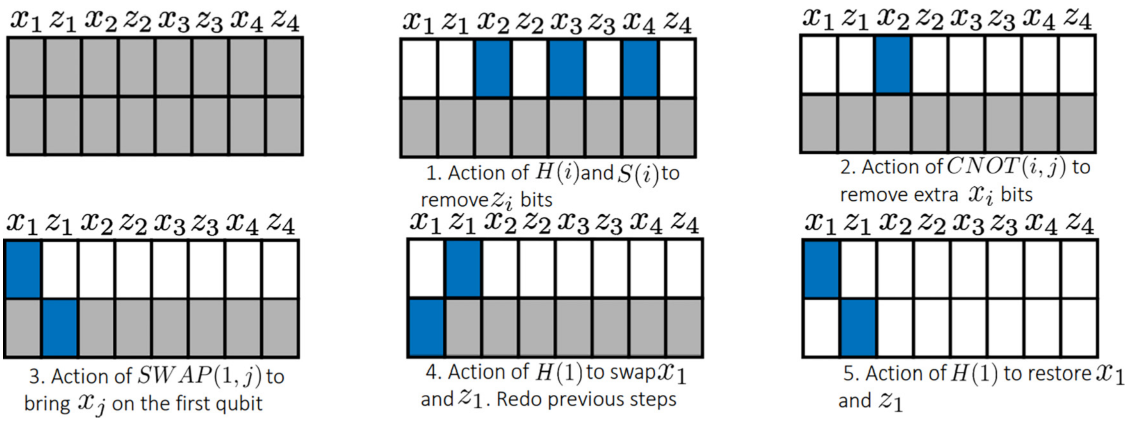
This section is devoted to the subroutine required to initialize the matrix $\tau_{\mathfrak{h}}$ given a subset of generators $\mathfrak{h}$ Below, we give an algorithm that builds the matrix $\tau_{\mathfrak{h}}$ from the subset $\mathfrak{h}$.

```
Input: $\mathfrak{h}$
Output: $\tau_{\mathfrak{h}}$
(1) Initialize $\tau'_{\mathfrak{h}}, \tau_{\mathfrak{h}}$ as two $2n \times 2n$ matrices filled by zeros.
(2) let $h = \text{card}(\mathfrak{h})$.
(3) for $i \in (0, h-1)$ do:
    (a) $[\tau'_{\mathfrak{h}}]_{2i+1} \mapsto [\tau'_{\mathfrak{h}}]_{2i+1} \oplus (\mathfrak{h}[1])_{\mathbf{xz}}$.
    (b) $\mathfrak{h} \mapsto \mathfrak{h} \setminus \{\mathfrak{h}[1]\}$.
    (c) for $j \in (0, \text{card}(\mathfrak{h}) - 1)$ do:
        (i) if $\tilde{\omega}([\tau_{\mathfrak{h}}]_{2i+1}, (\mathfrak{h}[j+1])_{\mathbf{xz}}) = 1$:
            (A) $[\tau'_{\mathfrak{h}}]_{2i+2} \mapsto [\tau'_{\mathfrak{h}}]_{2i+2} \oplus (\mathfrak{h}[j+1])_{\mathbf{xz}}$.
            (B) $\mathfrak{h} \mapsto \mathfrak{h} \setminus \{\mathfrak{h}[j+1]\}$.
(4) $k = 0$.
(5) for $i \in (0, h-1)$ do:
    (a) if $[\tau'_{\mathfrak{h}}]_{2i+2} \neq \mathbf{0}$:
        (i) $[\tau_{\mathfrak{h}}]_{2k+1} \mapsto [\tau_{\mathfrak{h}}]_{2k+1} \oplus [\tau'_{\mathfrak{h}}]_{2i+1}$.
        (ii) $[\tau_{\mathfrak{h}}]_{2k+2} \mapsto [\tau_{\mathfrak{h}}]_{2k+2} \oplus [\tau'_{\mathfrak{h}}]_{2i+2}$.
        (iii) $k \mapsto k+1$.
(6) for $i \in (0, h-1)$ do:
    (a) if $[\tau'_{\mathfrak{h}}]_{2i+1} \neq \mathbf{0}$ and $[\tau'_{\mathfrak{h}}]_{2i+2} = \mathbf{0}$:
        (i) $[\tau_{\mathfrak{h}}]_{2k+1} \mapsto [\tau_{\mathfrak{h}}]_{2k+1} \oplus [\tau'_{\mathfrak{h}}]_{2i+1}$.
        (ii) $k \mapsto k+1$.
```

Let us briefly explain the content of the algorithm. The first **for** cycle fills the matrix with generators. Each generator is followed either by an anticommuting generator or by **0** if there is no anticommuting generator in $\mathfrak{h}$. The other two **for** cycles are used to order the matrix according to the convention outlined in Sec. V A: paired generators are placed first, then followed by unpaired generators, and then by zeros. The time complexity of the matrix initializer is $O(nh)$

2. Sweeping

The following algorithm is called as a subroutine for both the *Diagonalizer* and *Random Clifford Sampling* subroutines. The sweeping is a symplectic transformation, denoted Sw, whose task is to manipulate two rows of $2k$ bits (odd and even) at a time and map them to the $2k$-bit string encoding σ_1^x/σ_1^z,


$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$
$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$
$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$
$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$
$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$
$x_1\ z_1\ x_2\ z_2\ x_3\ z_3\ x_4\ z_4$

1. Action of $H(i)$and $S(i)$to remove z_i bits

2. Action of $CNOT(i,j)$ to remove extra x_i bits

3. Action of $SWAP(1,j)$ to bring x_j on the first qubit

4. Action of $H(1)$ to swap x_1 and z_1. Redo previous steps

5. Action of $H(1)$ to restore x_1 and z_1

FIG. 3. Sketch of the Sweeping subroutine summarized in five steps.

i.e.,

$$\begin{pmatrix}1&0&0&1&\dots&0&1\\0&1&1&0&\dots&1&1\end{pmatrix} \xrightarrow{Sw} \begin{pmatrix}1&0&0&0&\dots&0&0\\0&1&0&0&\dots&0&0\end{pmatrix}, \tag{B1}$$

through symplectic transformations (equivalent to elementary Clifford gates). A sketch of the algorithm as follows: Denote by P_1, P_2 the Pauli operators corresponding to the odd and even rows, respectively. First, it maps P_1 in a Pauli string of σ^x. Such mapping can be realized by the repetitive application of Hadamard and phase gates. Then, it cancels the redundant σ^x, leaving just the σ^x on the most significant qubit by using CNOTs and results in $P_1 \mapsto \tilde{P}_1 = \sigma^x_1$. After that, the task of the algorithm is to map P_2 to σ^z. By applying the Hadamard gate on the first qubit, it transforms $\tilde{P}_1$ to σ^z_1. This operation allows us to map P_2 to σ^x_1 following the same procedure described above for P_1. The subsequent application of a Hadamard gate on the first qubit restores the right order; in this way, one has the encoding of σ^x_1 on the odd row and the one of σ^z_1 on the even row (see Fig. 3 for a sketch of the Sweeping subroutine). The algorithm proceeds as follows:

Input: $k, P_1,\ P_2 \in \mathbb{P} \mid \{P_1, P_2\} = 0$

Output: σ^x_1, σ^z_1

(1) Look at the odd row: if $z_i = 0$, do nothing; if instead $z_i = 1$ and $x_i = 0$ apply a Hadamard gate H to the circuit on ith qubit, otherwise if $z_i = 1$ and $x_i = 1$ apply a phase gate S to the circuit on the ith qubit.

(2) Search for the $x_i = 1$ and build a sorted list of indices $\mathcal{J} = \{i|x_i = 1\}$ then apply CNOT gates on the different indices of $\mathcal{J}$ as follows:

(a) Pair the indices of $\mathcal{J}$ from the less significant qubit to the most significant qubits, then apply in parallel CNOT$(\mathcal{J}_i, \mathcal{J}_{i+1})$ on the different pairs.

(b) Update the set $\mathcal{J}$ removing the last element of the pairs from it.

(c) Repeat step 2b ntil only the most significant qubit index is contained in the set $\mathcal{J}$.

(3) Apply a SWAP between the ith location and the first location.

(4) Apply a Hadamard gate on the first qubit.

(5) Repeat the steps from (i) to (iii) on the even row.

(6) Apply a Hadamard gate on the first qubit.

The time complexity of the sweeping is $O(n^2)$ [105].

3. Random Clifford sampling

The task of the algorithm is to build a random Clifford operator (following Refs. [77,105]). As shown in Sec. V A, a tableau T_{U_0} can efficiently encode a Clifford operator, and thus sampling T_{U_0} is equivalent to sampling a Clifford operator. The algorithm to sample a tableau T_{U_0} works on two rows per time, namely, the ith and the $i + 1$st row, and proceeds as follows:

Input: n

Output: T_{U_0}

Let $\widetilde{T}_{U_0}$ be a two $2n \times 2n$ Boolean matrix such that $[\widetilde{T}_V]_\alpha = \mathbf{0}$ for any $\alpha = 1, \dots, 2n$, $\boldsymbol{\phi}_{U_0} = \mathbf{0}$. While $j \leqslant n$:

(i) Extract a random Pauli operators acting on $n - j + 1$ qubits $P^x_{n-j+1} \in \mathbb{P}([n - j + 1])$.

(ii) Extract a random Pauli operators acting $n - j + 1$ qubits $P^z_{n-j+1} \in \mathbb{P}([n - j + 1])$, such that $\{P^x_{n-j+1}, P^z_{n-j+1}\} = 0$.

(iii) Update:

$$[\widetilde{T}_{U_0}]_{2j-1} \mapsto [\widetilde{T}_{U_0}]_{2j-1} \oplus \left(\mathbb{1}_{j-1} \otimes P^x_{n-j+1}\right)_{\mathbf{xz}},$$
$$[\widetilde{T}_{U_0}]_{2j} \mapsto [\widetilde{T}_{U_0}]_{2j} \oplus \left(\mathbb{1}_{j-1} \otimes P^z_{n-j+1}\right)_{\mathbf{xz}}. \tag{B2}$$

(iv) Perform a Sweeping Sw_j on the rows $[\widetilde{T}_{U_0}]_{2j-1}$ and $[\widetilde{T}_{U_0}]_{2j-1}$.

(v) Sample $[\boldsymbol{\phi}_{U_0}]_{2j-1} \in \{0, 1\}$.

(vi) Sample $[\boldsymbol{\phi}_{U_0}]_{2j} \in \{0, 1\}$.

The Tableau T_{U_0} is the $2n \times 2n + 1$ matrix obtained as

$$T_{U_0} = ((Sw_n Sw_{n-1} \cdots Sw_1)^{-1}(I_{2n})|\boldsymbol{\phi}_{U_0}), \tag{B3}$$

where I_{2n} is the identity $2n \times 2n$ matrix. $Sw(A)$ denotes the action of the symplectic transformations Sw_j for $j = 1, \dots, n$ on a symplectic matrix A. The algorithm has a time complexity of $O(n^2)$. To synthesize a Clifford circuit from the random tableau built out from the algorithm, one can employ the algorithm described in Ref. [77] with time complexity of $O(n^3)$.

Let us comment on the above algorithm. In steps (i) and (ii), we perform the sampling of an anticommuting pair of Pauli operators. In step (i), the task is to sample a nonidentity Pauli operator whose sampling probability is equal to $(1-4^{-k})$. In step (ii), the task is to sample a Pauli operator anticommuting with the one obtained from step (i). The probability of such sampling is equal to $1/2$ since a k-qubit Pauli operator anticommutes with $4^k/2$ Pauli operators. Thus, the probability to sample a pair of anticommuting Pauli operators $P_1, P_2 \in \mathbb{P}([k])$, having nontrivial support on k qubits reads

$$\Pr(P_1, P_2 \,|\, \{P_1, P_2\} = 0) = \frac{(1-4^{-k})}{2} \geqslant \frac{3}{8}. \tag{B4}$$

The time complexity of a single iteration of this subroutine is $O(n)$. The reason why the algorithm works on $2n-2j$ bits each jth step of the *while* loop is due to the constraints given by the commutation relations between the Pauli operators of a tableau (cf. Sec. V A). In more detail, at the $j+1$st step of the while loop, the algorithm has already sampled j pairs of generators. Via the sweeping transformation the algorithm transforms the j pairs of generators to $\{\sigma_i^x, \sigma_i^z\}_{i=1}^{j}$. A new pair of anticommuting generators must commute with the already found j pairs of generators. The only way to sample a pair of generators, that commutes with $\{\sigma_i^x, \sigma_i^z\}_{i=1}^{j}$, is that it acts identically on the first j qubits, which in turn implies that the first $2j$ bits are zeros.

4. Diagonalizer

This section describes the subroutine that maps an incomplete tableau to a partial identity, i.e., the diagonalizer $\mathcal{D}_{\mathfrak{h}}(\cdot)$, where $\mathfrak{h} \subset \mathfrak{g}$ a generating set of $\mathbb{P}$, introduced in Sec. V A. Let $\tau_{\mathfrak{h}}$ be the matrix corresponding to the subset $\mathfrak{h}$, as the one introduced in Eq. (44):

$$\tau_{\mathfrak{h}} \equiv \begin{pmatrix} (g_1)_{\mathbf{xz}} \\ (g_2)_{\mathbf{xz}} \\ \vdots \\ (g_i)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ (g_h)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}. \tag{B5}$$

The algorithm can then be written in the following way:

```
Input: $n$, $\tau_{\mathfrak{h}}$
Output: $\mathcal{D}_{\mathfrak{h}}$, $\hat{\mathcal{D}}_{\mathfrak{h}}$
(i) For each $j = 1, \dots, n$, do:
    (a) Perform the Sweeping of the pair $([\tau_{\mathfrak{h}}]_{2j-1}, [\tau_{\mathfrak{h}}]_{2j})$.
(ii) For each $j = 1, \dots, n$, do:
    (a) Check all the elements corresponding to the columns $2j-1, 2j$, and the rows $2j < r < 2n$.
    (b) Locate the extra 1 bits, and store the row index $r$ and the column index $c$. If $c$ is even, apply a Hadamard gate on the $j$th qubit, while if $r$ is even, apply a Hadamard gate on the $r/2$th qubit.
    (c) If $r$ is even, apply CNOT$(r/2, j)$, else apply CNOT$((r-1)/2, j)$.
    (d) In the end, if $c$ is even, apply a Hadamard gate on the $j$th qubit, while if $r$ is even, apply a Hadamard gate on the $r/2$th qubit.
```

The algorithm after step (i) maps the matrix $\tau_{\mathfrak{h}}$ in a new matrix $\widetilde{\tau}_{\mathfrak{h}}$, that reads

$$\widetilde{\tau}_{\mathfrak{h}} := \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & z_1 & \dots & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \end{pmatrix}. \tag{B6}$$

As also shown in Eq. (B6), the action of the sweeping on two anticommuting rows cancel all the 1 bits in the most significant columns. Instead, when the sweeping is performed on unpaired rows, the cancellation of 1 bits is partial since it is not constrained by the commutation or anticommutation relations. Step (ii) addresses this issue: it locates all the spurious σ^zs in the string and converts them to σ^x through the action of Hadamard gates $[\mathrm{H}(\sigma^z)\mathrm{H} = \sigma^x]$. Then it removes the redundant σ^x through the action of CNOT gates $[\mathrm{CNOT}(\sigma^x \otimes \sigma^x)\mathrm{CNOT} = \sigma^x \otimes \mathbb{1}_{[1]}]$. In the end, a layer of Hadamard gates (equal to the one applied before) is applied to restore the most significant σ^z. After the steps (ii) the matrix $\widetilde{\tau}_{\mathfrak{h}}$ is then mapped in

$$[\mathcal{D}_{\mathfrak{h}}(\tau_{\mathfrak{h}})]_\alpha = \begin{cases} e_\alpha, & \text{if } [\tau_{\mathfrak{h}}]_\alpha = (g_i)_{\mathbf{xz}} \quad \text{for some } g_i \in \mathfrak{h} \\ \mathbf{0}, & \text{if } [\tau_{\mathfrak{h}}]_\alpha = \mathbf{0}. \end{cases} \tag{B7}$$

Note that, at the end of the algorithm, both the diagonalizer, as a $2n \times 2n$ symplectic matrix, and the corresponding Clifford circuit is revealed. The time complexity of the diagonalizer subroutine is $O(n^2)$ due to the time complexities of the two main steps: step (i) and (ii). Both steps can be shown to possess time complexity $O(n^2)$.

5. Constrained random Clifford completion

This section describes the novel algorithm to generate a random Clifford when some rows of the tableau are already fixed. Such random Clifford is *constrained* because its adjoint action on some Pauli operators is fixed by the given rows of the tableau. As explained in Sec. V A, the tableau is organized as a list of $2n$-bit strings (ignoring the last phase-bit) describing the map $\sigma \mapsto U_0^\dagger \sigma U_0$ for $\sigma \in \{\sigma_1^x, \sigma_1^z, \dots, \sigma_n^x, \sigma_n^z\}$. The fixed rows of the given incomplete tableau can be of two types: (i) paired, i.e., the tableau contains the information about the map $\sigma_j^x \mapsto U_0^\dagger \sigma_j^x U_0$ and $\sigma_j^z \mapsto U_0^\dagger \sigma_j^z U_0$ for some j; or, unpaired, i.e., the tableau just contains the information about either $\sigma_j^x \mapsto U_0^\dagger \sigma_j^x U_0$ or $\sigma_j^z \mapsto U_0^\dagger \sigma_j^z U_0$ for some j. Therefore, without loss of generality, one can consider an incomplete tableau $\tau_{\mathfrak{h}}$ as the one introduced in Definition 3, and a phase vector $\boldsymbol{\phi}_{\mathfrak{h}} = (\phi_1, \phi_2, \dots, \phi_i, 0, \dots, \phi_h, 0 \dots 0)$. The matrix $\tau_{\mathfrak{h}}$ will be written as

$$\tau_{\mathfrak{h}} = \begin{pmatrix} (g_1)_{\mathbf{xz}} \\ (g_2)_{\mathbf{xz}} \\ \vdots \\ (g_i)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ (g_h)_{\mathbf{xz}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} P = \text{Paired rows} \\ N = \text{Unpaired rows} \\ \mathbf{0}_{2(n-n_P-n_N)} \end{pmatrix}, \tag{B8}$$

where P is a matrix of dimension $2n_P \times 2n$ that encodes the set of anticommuting pairs of Pauli operators, N is a matrix of dimension $2n_N \times 2n$ that encodes the set of unpaired Pauli operators; $\mathbf{0}_{2(n-n_P-n_N)}$ is a matrix of dimension $2(n - n_P - n_N) \times 2n$ that encodes the set of unconstrained rows. Note that the total number of constrained rows is $2n_P + n_N$. The task is to fill the missing rows uniformly at random and implement one of the corresponding Clifford operators. The algorithm proceeds as follows:

```
Input: $n$, $\tau_{\mathfrak{h}}$, $\boldsymbol{\phi}_{\mathfrak{h}}$
Output: $T_{U_0}$

(i) For $i = 1, \dots, n_p$ do:
    (a) Perform the Sweeping on the pair $([\tau_{\mathfrak{h}}]_{2i-1}, [\tau_{\mathfrak{h}}]_{2i})$.
(ii) Apply a random symplectic transformations Sym on the last $n - n_P$ qubits. Denote with $\widetilde{\tau_{\mathfrak{h}}}$ the tableau after the action of Sym.
(iii) For $i = n_p + 1, \dots, n_N + n_P$ do:
    (a) Perform the Sweeping on the pair $([\widetilde{\tau}_{\mathfrak{h}}]_{2i-1}, [\widetilde{\tau}_{\mathfrak{h}}]_{2i})$.
(iv) For $j = n_P + 1, \dots, n_N + n_P$:
    (a) Check all the elements corresponding to the columns $2j - 1, 2j$, and the rows $2j < r < 2(n_P + n_N)$.
    (b) Locate the extra 1 bits, and store the row index $r$ and the column index $c$. If $c$ is even, apply a Hadamard gate on the $j$th qubit, while if $r$ is even, apply a Hadamard gate on the $r/2$th qubit.
    (c) If $r$ is even, apply CNOT$(r/2, j)$, else apply CNOT$((r - 1)/2, j)$.
    (d) In the end, if $c$ is even, apply a Hadamard gate on the $j$th qubit, while if $r$ is even, apply a Hadamard gate on the $r/2$th qubit.
    (e) Sample $[\boldsymbol{\phi}]_{2j} \in \{0, 1\}$.
(v) Fill the incomplete rows by completing the identity.
(vi) Fill the last $2(n - n_P - n_N)$ columns of $\mathbf{0}_{2(n-n_P-n_N)}$ with a random tableau (see Random Clifford Sampling).
(vii) for $j = n_N + n_P, \dots, n$:
    (a) Sample $[\boldsymbol{\phi}_{\mathfrak{h}}]_{2j-1} \in \{0, 1\}$.
    (b) Sample $[\boldsymbol{\phi}_{\mathfrak{h}}]_{2j} \in \{0, 1\}$.
```

The final tableau is then given by

$$T_{U_0} = (\tau_{\mathfrak{h}} | \boldsymbol{\phi}_{\mathfrak{h}}) = \big((\widetilde{Sw}_2 Sw_2 Sym Sw_1)^{-1} T_{U_0^{(1)}}(I_{2n}) | \boldsymbol{\phi}_{\mathfrak{h}} \big), \tag{B9}$$

where Sw_1 denotes the action of step (i), Sym the action of step (ii), Sw_2 and $\widetilde{Sw}_2$ denote respectively the action of step (iii) and (iv), and $T_{U_0^{(1)}}$ labels instead the random tableau generated in step (vi). At the end of the subroutine, a constrained random Clifford has been generated, where the randomness comes from two elements: the application of the random Clifford U_{Sym} in step $(ii)(a)$ and the generation of a random Clifford $U_0^{(1)}$ on $n - (n_P + n_N)$ qubits.

The algorithm runs in a time $O(n^3)$. The time complexity of step (i) is $O(n_P^2)$, due to the time complexity of the *Sweeping* [105]. Step (ii) has time complexity $O((n - n_P)^3)$. Step (iii) has time complexity $O((n_N)^2)$. Step (v), corresponding to the generation or a random tableau, has time complexity $O((n - n_P - n_N)^2)$. While the last steps have time complexity $O(n)$. Then, one can synthesize the Clifford operator from the tableau in terms of CNOT, H, S with an overhead of $O(n^3)$ steps (see Ref. [77]). In the following, it will be discussed in more detail the action of every step on the incomplete tableau. Step (i) being the iterated action of the *Sweeping* subroutine on the first n_P pair of rows, maps P to I_{2n_P} (the $2n_P \times 2n_P$ identity matrix), so step (i) acts on the tableau as

$$\tau_{\mathfrak{h}} \xrightarrow{Sw_1} Sw_1(\tau_{\mathfrak{h}}) \equiv \begin{pmatrix} I_{2n_P} & 0 \\ 0 & N_{Sw_1} \\ 0 & W \end{pmatrix}, \tag{B10}$$

where N_{Sw_1} denotes the image of the matrix N through the action of the sweeping Sw_1. The unknown part $\mathbf{0}_{2(n-n_P-n_N)}$ is untouched by any operation, being filled by zeros. The iterated action of the sweeping, being symplectic, preserves the commutation relations between the rows, and thus erases all the 1 bits in the most significant qubit columns, which explains the zeros below I_{2n_P}. In step (ii) to avoid the introduction of bias it is first applied a symplectic transformation Sym on the last $2(n - n_P)$ bits of the tableau $\tau_{\mathfrak{h}}$. The resulting action is

$$Sw_1(R) \xrightarrow{Sym} SymSw_1(R) \equiv \begin{pmatrix} I_{n_P} & 0 \\ 0 & N_{SymSw_1} \\ 0 & \mathbf{0}_{2(n-n_P-n_N)} \end{pmatrix}, \tag{B11}$$

where N_{SymSw_1} denotes the action of Sym on N_{Sw_1}. Step (iii), denoted with Sw_2, corresponds to the action of an iterated sweeping on N_{SymSw_1}. Looking only at the matrix N_{SymSw_1}, one obtains

$$N_{SymSw_1} \xrightarrow{Sw_2} N_{Sw_2SymSw_1} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ - & - & - & - & \dots & - \\ 0 & z_1 & 1 & 0 & \dots & 0 \\ - & - & - & - & \dots & - \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{B12}$$

Note that the iterated sweeping Sw_2, contrary to the previous cases, is unable to remove all the 1 bits in the most significant columns. This degree of freedom is manifest in Eq. (B12) with the free z_1 bit: indeed, both $\mathbb{1}_{[1]} \otimes \sigma^x$, and $\sigma^z \otimes \sigma^x$ do commute with $\sigma^x \otimes \mathbb{1}_{[1]}$. The action of step (iv), labeled with $\widetilde{Sw}_2$, is necessary to address this issue. $\widetilde{Sw}_2$ is a symplectic transformation that can be built from a Clifford circuit. The circuit, described in step (iv), is made by a layer of Hadamard gates, a layer of CNOT gates, and another layer of Hadamard gates. The first layer is necessary to convert σ^z into a σ^x, the second layer is required to remove the redundant σ^x, and the last one to restore the σ^z. As a result the matrix $N_{Sw_2SymS_1}$ is then mapped in

$$N_{Sw_2SymS_1}\xrightarrow{\widetilde{Sw}_2}N_{\widetilde{Sw}_2Sw_2SymSw_1}$$
$$\equiv\begin{pmatrix}1&0&0&0&\dots&0\\ -&-&-&-&\dots&-\\ 0&0&1&0&\dots&0\\ -&-&-&-&\dots&-\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots\end{pmatrix}. \tag{B13}$$

In step (v) we complete $N_{Sw_2SymS_1}$ with 1 bits to be the identity matrix. In step (vi), the tableau is completed by the addition of a random tableau $T_{U_0^{(1)}}$. The resulting incomplete tableau is equal to

$$\tau_{\mathfrak{h}}\ {}_{\widetilde{Sw}_2Sw_2SymSw_1}\xrightarrow{\text{Random Clifford sampling}}\begin{pmatrix}I_{n_P}&0&0\\ 0&I_{n_N}&0\\ 0&0&T_{U_0^{(1)}}\end{pmatrix}. \tag{B14}$$

The final steps sample the phases of the constrained Clifford operator. At the end we obtain

$$T_{U_0}=(\tau_{\mathfrak{h}}|\boldsymbol{\phi})=\big((\widetilde{Sw}_2Sw_2SymSw_1)^{-1}T_{U_0^{(1)}}(I_{2n})|\boldsymbol{\phi}\big). \tag{B15}$$

6. Learning a Pauli string

In this section, we describe the subroutine able to learn the adjoint action of a unitary operator U on a given Pauli string, say P. We assume that $U^\dagger PU\in\mathcal{P}$. Note that the following subroutine works also in the case $U^\dagger PU\notin\mathcal{P}$ and returns still a Pauli string by construction that, of course, will not correspond to the correct operator resulting from the adjoint action of U on P. We want to stress that the strategy of the algorithm presented in Sec. VC is to pretend that $U^\dagger PU\in\mathcal{P}$ for all $P\in\mathbb{P}[n-m]$ and then check the correct learning by *Verification and Removal* subroutine.

Input: $n, U, P\in\mathbb{P}$

Output:

$$U^\dagger PU\in\mathbb{P},\quad \text{if } U^\dagger PU\in\mathcal{P}$$
$$\mathbb{P}\ni Q\neq U^\dagger PU,\quad \text{else.}$$

For each $j=1,\dots,n$, do:

(i) Prepare the following states, defined on two copies of $\mathcal{H}$:

$$|\psi_j^{(0)}\rangle:=|\text{EPR}_j\rangle\otimes|0\rangle^{\otimes 2},$$
$$|\psi_j^{(+)}\rangle:=|\text{EPR}_j\rangle\otimes|+\rangle^{\otimes 2}, \tag{B16}$$

where $|\text{EPR}_j\rangle=2^{(n-1)/2}\sum_{k\neq j}|kk\rangle$ is a EPR pair but on the jth qubit.

(ii) Evolve each branch pair with the scrambler U, i.e., $|\psi_j^{(0,+)}(U)\rangle=U\otimes U\,|\psi_j^{(0,+)}\rangle$.

(iii) Measure M times the operator $P^{\otimes 2}$ on both $|\psi_j^{(0,+)}(U)\rangle$, collecting a string of $2M$ bits $\{\sigma_1^{(0)},\dots,\sigma_M^{(0)},\sigma_1^{(+)},\dots,\sigma_M^{(+)}\}$. If $\sigma_1^{(0)}=\dots=\sigma_M^{(0)}$ assign $s_j^{(0)}=1$ to the binary variable $s_j^{(0)}$, otherwise assign $s_j^{(0)}=0$. Analogously for $s_j^{(+)}$.

(a) if $\{s_j^{(0)},s_j^{(+)}\}=\{1,1\}$, then the jth component of $U^\dagger\sigma U$ is the identity.

(b) if $\{s_j^{(0)},s_j^{(+)}\}=\{1,0\}$ is the Pauli matrix σ^z.

(c) if $\{s_j^{(0)},s_j^{(+)}\}=\{0,1\}$ is the Pauli matrix σ^x.

(d) if $\{s_j^{(0)},s_j^{(+)}\}=\{0,0\}$ is the Pauli matrix σ^y.

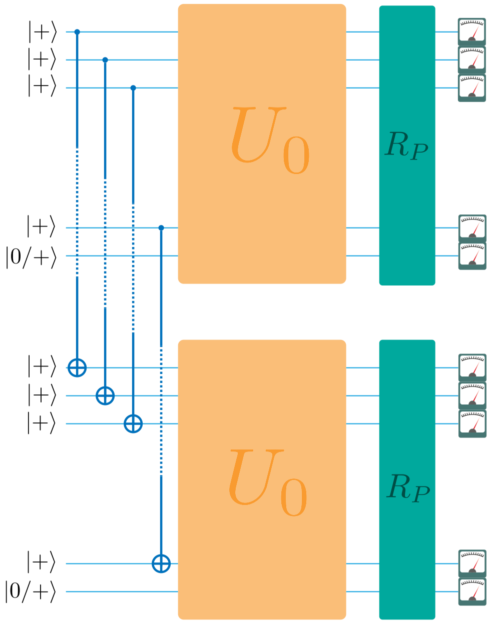


FIG. 4. Pictorial representation of *Learning a Pauli string* subroutine. The algorithm works with the quantum circuit sketched above consisting in 4 main steps: preparation of the Bell state via $|+\rangle$ states and CNOTs; application of two copies of the scrambler U_0; a rotation R_P conditioned on the measurement of the expectation value of a given Pauli operator P, e.g., $P=X\otimes Z\otimes Y\otimes\mathbb{1}\otimes\cdots\otimes Z$ then $R_P=H\otimes\mathbb{1}\otimes HS\otimes\mathbb{1}\otimes\cdots\otimes\mathbb{1}$. Finally, a measurement in the computational basis.

The above algorithm requires $O(nM)$ queries to the unitary U, runs in a time $O(n^2M)$, and reveals the Pauli string image of the adjoint action of U on P. We are interested in estimating the failure probability only for $P\in G(U_t)$. Thus, if $P\in G(U_t)$, then $U_t^\dagger PU_t\in\mathbb{P}$ and we can easily estimate the probability of failure of the above algorithm. Indeed, the probability to fail the learning of P is $2n\times 2^{-M}$, indeed the failure probability is given by the probability that an unbiased coin gives tail for M tosses in a row, i.e., 2^{-M}. See Fig. 4 for a pictorial representation of the above algorithm.

7. Verification and removal

In this section, we describe the algorithm to verify whether the *Learning a Pauli string* subroutine worked correctly. Denote with P the input and Q the output of *Learning a Pauli string* respectively, and proceed as follows:

Input: $n, P, Q \in \mathbb{P}, U_t$

Output: yes or no

(i) Construct a Bell pair $|EPR\rangle$ between two copies of $\mathcal{H}$.

(ii) Evolve one branch with U_t and obtain $|U_t\rangle \equiv \mathbb{1} \otimes U_t\,|EPR\rangle$.

(iii) Measure the expectation value $\langle U_t|Q \otimes P|U_t\rangle$ up to an error $\epsilon \leqslant 2^{-2t}$.

(iv) If $\langle U_t|Q \otimes P|U_t\rangle = \pm 1$, then output yes; otherwise output no.

The above discrimination works because

$$\langle U_t|Q \otimes P|U_t\rangle = \pm 1 \text{ iff } U_t^\dagger P U_t = \pm Q. \tag{B17}$$

The above algorithm requires $O(\epsilon^{-2})$ queries to U_t, and runs in a time $O(\epsilon^{-2})$. The key insight here is that since U_t is a t-doped Clifford circuit, the expectation value $\langle U_t|Q \otimes P|U_t\rangle$ takes discrete values. In other words, there exists a minimal resolution δ_t, defined as the minimum difference between two consecutive values of $\langle U_t|Q \otimes P|U_t\rangle$, denoted as $\delta_t \equiv \min|\langle U_t|Q \otimes P|U_t\rangle - \langle U_t|Q' \otimes P'|U_t\rangle|$. Thus, given ϵ such that $\epsilon \leqslant \delta_t$, the learner can determine the expectation value $\langle U_t|Q \otimes P|U_t\rangle$ exactly. In Appendix D, we present the finite resolution Lemma 7, for which we bound $\delta_t \geqslant 2^{-bt}$, where $b \simeq 2.27$. Additionally, we expect that the average case scenario will yield even more favorable results as our numerics in Sec. V E demonstrate. We therefore arrive at the conclusion that, in the worst case scenario, by selecting $b < 2.5$, it is possible to determine whether $P \in G(U_t)$, meaning whether P is a preserved Pauli string, with $O(2^{5t})$ queries to the t-doped Clifford circuit U_t.

8. Phase check

In this section, we introduce the subroutine to learn the phase of the Pauli string, associated with the adjoint action $U^\dagger P U$. Let be $|epr\rangle_+ \equiv \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $|epr\rangle_- \equiv \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$. The action of $\mathbb{1}_2$, $\sigma^{x\otimes 2}$ or $\sigma^{z\otimes 2}$ on $|epr\rangle_+$ returns $|epr\rangle_+$; similarly the action of $\sigma^{y\otimes 2}$ on $|epr\rangle_-$ returns $|epr\rangle_-$. Let P be the Pauli string of which one wants to learn the phase $s \in \{0, 1\}$ of through the action of a unitary U. In the algorithm, it is assumed that $P(U) \equiv U^\dagger P U \in \mathcal{P}$.

Input: $n, P(U) \equiv U^\dagger P U$

Output: $s \in \{0, 1\}$

(i) Use the *Constrained random Clifford* subroutine to build a Clifford U_0 such that $U_0^\dagger P U_0 = \pm P(U)$.

(ii) Read each element of the Pauli string $U_0 P U_0^\dagger$ and build the n-fold state $|s\rangle$, constructed as a tensor product of $|epr\rangle_+$ and $|epr\rangle_-$ in the following way: if the ith element of the Pauli string is $\mathbb{1}_{[1]}$, σ^x or σ^z, then the ith element of $|s\rangle$ is $|epr\rangle_+$, otherwise the ith element of $|s\rangle$ is $|epr\rangle_-$.

(iii) Evolve $|s\rangle$ with $U \otimes V$, and let s be the result of a one-shot measurement of the operator $P^{\otimes 2}$. If $s = 1$, then the phase is $+1$; conversely, if $s = -1$ the phase is -1.

The algorithm needs one query to U, and runs in a time $O(n^3)$.

APPENDIX C: PROOF OF THEOREM 1

To prove Theorem 1, we first enunciate and prove a series of lemmas. Afterward, the proof of the theorem will descend in a straightforward fashion.

Lemma 1. [Eq. (23)] Let U_t be a t-doped Clifford circuit, the fidelity in Eq. (17) can be written as

$$\mathcal{F}_{\rm V}(U_t) = \frac{1 + R}{d_A^2 \Omega_{G_D}(U_t) + R'}, \tag{C1}$$

where

$$R := \frac{1}{d|G_D(U_t)|} \sum_{P_D \in \overline{G_D}} \operatorname{tr}(P_D(U_t) P_D(V)),$$

$$R' := \frac{1}{d|G_D(U_t)|} \sum_{P_D \in \overline{G_D}, P_A} \operatorname{tr}(P_A P_D(U_t) P_A P_D(V)).$$

Lemma 2. Let U_t be a t-doped Clifford circuit, the cardinality of $G_{[n-m]}(U_t)$, defined in Eq. (19), is lower-bounded as $|G_{[n-m]}(U_t)| \geqslant 2^{2n-2m-t}$.

Corollary 4. Let U_t be a t-doped Clifford circuit, the cardinality of $G_D(U_t)$, defined in Eq. (19), is lower-bounded as $|G_D(U_t)| \geqslant 2^{2|D|-t}$.

Lemma 3. If $\operatorname{tr}(P_D(U_t) P_D(V)) = 0$ for all $P_D \in \mathbb{P}(D) \setminus G_D(U_t)$, then $R = 0$ and $R' = 0$. Thus

$$\Pr_{V \in \mathcal{V}_{U_t}} [R = 0, R' = 0] \geqslant \Pr_{V \in \mathcal{V}_{U_t}} [\operatorname{tr}(P_D(U_t) P_D(V)) = 0,$$
$$\forall\, P_D \in \mathbb{P}(D) \setminus G_D(U_t)]. \tag{C2}$$

Lemma 4. For a random Clifford decoder $V \in \mathcal{V}_{U_t}^D$, the probability that $\operatorname{tr}(P_D(U_t) P_D(V)) = 0$ for all $P_D \in \mathbb{P}(D) \setminus G_D(U_t)$ is lower-bounded by

$$\Pr_{V \in \mathcal{V}_{U_t}^D} [\operatorname{tr}(P_D(U_t) P_D(V)) = 0,$$
$$\forall\, P_D \in \mathbb{P}(D) \setminus G_D(U_t)] \geqslant 1 - \frac{2^t}{d_C^2}.$$

Lemma 5. Let $\mathcal{F}_V(U_t)$ the fidelity defined in Eq. (17). If $V \in \mathcal{V}_{U_t}$ and $|D| = n$, $|C| = 0$ then the fidelity becomes the gate fidelity,

$$\mathcal{F}_{\rm V}(U_t) = \frac{|\operatorname{tr}(V^\dagger U_t)|}{d^2}, \tag{C3}$$

and $\mathcal{F}_V(U_t) < 1$ is strictly less than one if and only if U_t is a non-Clifford unitary operator.

Proof of Lemma 1. Rewrite the fidelity $\mathcal{F}_V$ in Eq. (17) as

$$\mathcal{F}_{\rm V}(U_t) = \frac{d^{-1} \sum_{P_D} \operatorname{tr}(P_D(U_t) P_D(V))}{d^{-1} \sum_{P_A, P_D} \operatorname{tr}(P_A P_D(U_t) P_A P_D(V))}. \tag{C4}$$

Then, let us evaluate the numerator and denominator separately, the numerator can be rewritten as

$$d^{-1} \sum_{P_D} \operatorname{tr}(P_D(U_t) P_D(V))$$
$$= |G_D(U_t)| + d^{-1} \sum_{P_D \in \overline{G_D}(U_t)} \operatorname{tr}(P_D(U_t) P_D(V)), \tag{C5}$$

where we used the fact that $U_t^{\dagger}P_D U_t = V^{\dagger}P_D V$ for any $P_D \in G_D(U_t)$, and defined $\overline{G_D}(U_t)$ as the complement set of G_D. While for the denominator:

$$\begin{aligned} d^{-1}\sum_{P_A,P_D}\operatorname{tr}(P_A P_D(U_t)P_A P_D(V)) &= d^{-1}\sum_{P_D\in G_D,P_A}\operatorname{tr}(P_A P_D(U_t)P_A P_D(U_t)) \\ &\quad + d^{-1}\sum_{P_D\in\overline{G_D},P_A}\operatorname{tr}(P_A P_D(U_t)P_A P_D(V)). \end{aligned}$$

Define the following two quantities:

$$\begin{aligned} R &:= \frac{1}{d|G_D(U_t)|}\sum_{P_D\in\overline{G_D}}\operatorname{tr}(P_D(U_t)P_D(V)), \\ R' &:= \frac{1}{d|G_D(U_t)|}\sum_{P_D\in\overline{G_D}(U_t),P_A}\operatorname{tr}(P_A P_D(U_t)P_A P_D(V)). \end{aligned}$$

Then define the truncated OTOC similarly to Eq. (9) as

$$\mathrm{OTOC}_{G_D}(U_t) = \frac{1}{d}\,\langle\operatorname{tr}(P_A P_D(U_t)P_A P_D(U_t))\rangle_{\mathbb{P}(A),G_{(U_t)}}\,, \tag{C6}$$

where we defined $\langle\cdot\rangle_{G_D} := |G_D(U_t)|^{-1}\sum_{P_D\in G_D(U_t)}(\cdot)$. Note that we can write

$$\begin{aligned} \mathcal{F}_{\mathrm{V}}(U_t) &= \frac{|G_D(U_t)|(1+R)}{|G_D(U_t)|\left[d_A^2\mathrm{OTOC}_{G_D}(U_t)+R'\right]} \\ &= \frac{1+R}{d_A^2\mathrm{OTOC}_{G_D}(U_t)+R'}. \end{aligned} \tag{C7}$$

■

Proof of Lemma 2. First of all, let us recall the definition

$$G_{[n-m]}(U_t) := \{P\in\mathbb{P}_D \,|\, P(U_t)\equiv U_t^{\dagger}PU_t\in\mathcal{P}\}. \tag{C8}$$

Let us prove that it is a subgroup of the full Pauli group. It is trivial to say that $\mathbb{1}\in G_{[n-m]}(U_t)$. Than, since $P^{-1}=P$ we have that for any $P\in G_{[n-m]}(U_t)$, then $P^{-1}\in G_{[n-m]}(U_t)$. Finally, thanks to the unitarity of U_t we have that if $P,Q\in G_{[n-m]}(U_t)$ then $PQ\in G_{[n-m]}(U_t)$. Now, let us prove that $|G_{[n-m]}(U_t)|\geqslant 2^{2n-2m-t}$. First of all define $G(U_t)$ as

$$G(U_t) := \{P\in\mathbb{P} \,|\, P(U_t)\in\mathcal{P}\}. \tag{C9}$$

It is known that [79] $|G(U_t)|\geqslant 2^{2n-t}$, and it is clear that $G(U_t)\leqslant\mathbb{P}$. Let us use the following group theory result: let $A,B\leqslant H$ two subgroup of H, then:

$$|A\cap B|\geqslant\frac{|A||B|}{|H|}. \tag{C10}$$

First, note that we can write

$$\begin{aligned} G_{[n-m]}(U_t) &= [\mathbb{P}([n-m])\cap U_t\mathbb{P}U_t^{\dagger}] \\ &= [\mathbb{P}([n-m])\cap\mathbb{P}\cap U_t\mathbb{P}U_t^{\dagger}] \\ &= [\mathbb{P}([n-m])\cap G(U_t)]. \end{aligned}$$

We find

$$|G_{[n-m]}(U_t)|\geqslant\frac{|\mathbb{P}([n-m])||G(U_t)|}{|\mathbb{P}|} = 2^{2n-2m-2t}. \tag{C11}$$

■

Proof of Lemma 3. Let us recall the lemma: The proof for $R=0$ is trivial. Let us proceed to the proof for $R'=0$. Since V is a Clifford operator, then $P_D(V)\in\mathbb{P}$, and thus $P_D(V)P_A = \phi(P_A,P_D(V))P_A P_D(V)$, where the phase is defined as

$$\phi(P_A,P_D(V)) := \frac{1}{d}\operatorname{tr}(P_A P_D(V)P_A P_D(V)). \tag{C12}$$

We can thus rewrite R' as

$$R' = \sum_{P_D\in\mathbb{P}(D)\setminus G_D(U_t),P_A}\phi(P_A,P_D(V))\operatorname{tr}(P_D(U_t)P_D(V)). \tag{C13}$$

Thus, from the last equality if $\operatorname{tr}(P_D(U_t)P_D(V))=0$ for all $P_D\in\mathbb{P}(D)\setminus G_D(U_t)$, then $R'=0$. ■

Proof of Lemma 4. Let $\mathcal{V}_{U_t}^D = \{V\in\mathcal{C}(n)\,|\,V^{\dagger}PV = U_t^{\dagger}PU_t,\ \forall\,P\in G_D(U_t)\}$ be the set of all Clifford decoders that can be found by the algorithm in Sec. VC. We are indeed only interested in random Clifford decoders modulo phases. ■

First of all, defined $\overline{\mathfrak{g}_D}(U_t)$ as the set of Pauli operators such that $\langle\mathfrak{g}_D\cup\overline{\mathfrak{g}_D}(U_t)\rangle = \mathbb{P}(D)$. After the injection of t non-Clifford gates, we have that $|\overline{\mathfrak{g}_D}(U_t)|\leqslant t$. Each $P_D\in\mathbb{P}(D)\setminus G_D(U_t)$ can be rewritten as $P_D = P_D'\widetilde{P}_D$, where $P_D'\in G_D(U_t)$, while $\widetilde{P}_D$ belongs to the set $\overline{G}_D(U_t)$ generated by $\overline{G}_D(U_t) = \langle\overline{\mathfrak{g}_D}(U_t)\rangle\setminus\mathbb{1}$. Since $|\mathbb{P}(D)\setminus G_D(U_t)| = d_D^2-|G_D(U_t)|$, we have that $|\overline{G}_D(U_t)| = d_D^2/|G_D(U_t)|-1$, where the -1 comes from the fact that $\overline{G}_D(U_t)$ does not contain the identity. For any $P_D\in\mathbb{P}(D)\setminus G_D(U_t)$, we thus write

$$\operatorname{tr}(P_D(U_t)P_D(V)) = \pm\operatorname{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V)), \tag{C14}$$

where we used the unitarity of U_t and V to write $P_D(U_t) = P_D'(U_t)\widetilde{P}_D(U_t)$, and $P_D(V) = \pm\widetilde{P}_D(V)P_D'(V)$. Thus,

$$\begin{aligned} &\Pr_{V\in\mathcal{V}_{U_t}^D}[\operatorname{tr}(P_D(U_t)P_D(V))=0,\ \forall\,P_D\in\mathbb{P}(D)\setminus G_D(U_t)] \\ &= \Pr_{V\in\mathcal{V}_{U_t}^D}[\operatorname{tr}\big(\widetilde{P}_D(U_t)\widetilde{P}_D(V)\big)=0,\ \forall\,\widetilde{P}_D\in\overline{G}_D(U_t)]. \end{aligned} \tag{C15}$$

$\widetilde{P}_D(U_t)$ is, in general, a combination of l Pauli operators, where $2\leqslant l\leqslant 2^t$, i.e., $\widetilde{P}_D(U_t) = \sum_{i=1}^{l}\alpha_i p_i$, where $p_i\in\mathbb{P}$ and $\alpha_i = d^{-1}\operatorname{tr}(p_i\widetilde{P}_D(U_t))$. Thus, for a single $\widetilde{P}_D\in\overline{G}_D(U_t)$, the probability $\Pr[\operatorname{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V))=0] = 1-\Pr[\operatorname{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V))\neq 0]$ and

$$\Pr_{V\in\mathcal{V}_{U_t}^D}[\operatorname{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V))\neq 0] = \Pr_{V\in\mathcal{V}_{U_t}^D}\left[\bigcup_{i=1}^{l}\operatorname{tr}(p_i\widetilde{P}_D(V))\neq 0\right]. \tag{C16}$$

The above is true because if $\operatorname{tr}(p_i P_D(V))\neq 0$ for some $i=1,\dots,l$, then $\operatorname{tr}(p_j P_D(V))=0$ for any $j\neq i$. By using Fréchet inequality we can upper bound the above probability by

$$\Pr_{V\in\mathcal{V}_{U_t}^D}[\operatorname{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V))\neq 0]\leqslant\sum_{i=1}^{l}\Pr_{V\in\mathcal{V}_{U_t}}[\operatorname{tr}(p_i\widetilde{P}_D(V))\neq 0]. \tag{C17}$$

Looking at the above equation, for any given p_i, there are two occurring cases: either there is no $V\in\mathcal{V}_{U_t}^D$ such that $p_l\propto\widetilde{P}_D(V)$ for every $\widetilde{P}_D(V)$, or there exists $V\in\mathcal{V}_{U_t}^D$ such that $p_l\propto\widetilde{P}_D(V)$. We thus only consider the latter, being the worst case scenario for $\operatorname{tr}(p_l\widetilde{P}_D(V))\neq 0$. Denote as $\widetilde{\mathcal{V}}_{U_t}^D(P) :=$

$\{V^\dagger PV \in \mathbb{P} \,|\, V \in \mathcal{V}_{U_t}^D\}$ the set of images of the Pauli operator P through the action of random decoders belonging to $\mathcal{V}_{U_t}^D$. For example, if $P \in G_D(U_t)$, then $\widetilde{\mathcal{V}}_{U_t}^D(P) = \{U_t^\dagger PU_t\}$, i.e., $\widetilde{\mathcal{V}}_{U_t}^D(P)$ is just the singleton of the image of P through U_t by construction. Since the algorithm in Sec. VC is generating a decoder V uniformly at random from the set $\mathcal{V}_{U_t}^D$, we conclude that a single p_i with $i = 1, \ldots, l$:

$$\Pr_{V\in\mathcal{V}_{U_t}^D}[\mathrm{tr}(p_i\widetilde{P}_D(V)) \neq 0] \leqslant \frac{1}{|\widetilde{\mathcal{V}}_{U_t}^D(\widetilde{P}_D)|}, \tag{C18}$$

where $|\widetilde{\mathcal{V}}_{U_t}^D(\widetilde{P}_D)|$ is the cardinality of the set $\widetilde{\mathcal{V}}_{U_t}^D(\widetilde{P}_D)$, i.e., the number of all possible Pauli operator [not belonging to $G_D(U_t)$] resulting from the adjoint action of $V \in \mathcal{V}_{U_t}^D$. We have the following lemma:

Lemma 6. Let U_t be a t-doped Clifford circuit, then

$$\min_{\widetilde{P}_D\in\overline{G}_D(U_t)} \left|\widetilde{\mathcal{V}}_{U_t}^D\left(\widetilde{P}_D\right)\right| \geqslant 2^t d_C^2. \tag{C19}$$

Proof. First of all note that

$$\min_{\widetilde{P}_D\in\overline{G}_D(U_t)} \left|\widetilde{\mathcal{V}}_{U_t}^D\left(\widetilde{P}_D\right)\right| = \min_{\widetilde{\sigma}\in\overline{\mathfrak{g}_D}(U_t)} \left|\widetilde{\mathcal{V}}_{U_t}^D(\widetilde{\sigma})\right|, \tag{C20}$$

i.e., the minimum number of possible images of a generator $\widetilde{\sigma} \in \overline{\mathfrak{g}_D}(U_t)$ of the group $\overline{G}_D(U_t)$ is exactly the minimum number of possible images of an element of the group $\overline{G}_D(U_t)$, being $\widetilde{\sigma} \in \overline{G}_D(U_t)$. Therefore, we need to compute in how many ways we can write a compatible row corresponding to the map $V : \widetilde{\sigma} \in \overline{\mathfrak{g}_D}(U_t) \mapsto \widetilde{\sigma}(V)$ in the incomplete tableau T_V. To take into account this, we need to look at the *Constrained random Clifford* subroutine [in particular Eq. (B8)]. There are two types of rows to fill: the (unpaired) ones belonging to N and those belonging to $\mathbf{0}$. To fill a row belonging to N, one needs to consider the anticommutation relation with the already known row (e.g., the consecutive one). There are

$$2^{n_N} \times 2^{2(n-n_P-n_N)} = 2^{2n-2n_P-n_N} \tag{C21}$$

ways to write the resulting Pauli string. Let us explain the above counting: in the submatrix of unpaired rows N, there are $2n_N$ rows, and only n_N of them are fixed. Thus, to fix just one of the empty rows, one has 2^{n_N} possibilities to write a $2n$ bit string corresponding to a valid Pauli operator. Conversely, to fill rows in the empty part of the incomplete tableau, i.e., $\mathbf{0}$, one has $2^{2n-2n_P-2n_N}$ degrees of freedom, which correspond to the number of rows contained in $\mathbf{0}$. Using the fact that $2^{2n_P+n_N} = |G_D(U_t)|$ by construction, one has that

$$\min_{\widetilde{\sigma}\in\overline{\mathfrak{g}_D}(U_t)} \left|\widetilde{\mathcal{V}}_{U_t}^D(\widetilde{\sigma})\right| = \frac{d^2}{|G_D(U_t)|}. \tag{C22}$$

As an immediate corollary of the above lemma, from Eq. (C18), we have

$$\forall\, p_i \in \{p_1, \ldots, p_l\}, \ \Pr_{V\in\mathcal{V}_{U_t}}[\mathrm{tr}(p_j\widetilde{P}_D(V)) \neq 0] \leqslant \frac{|G_D(U_t)|}{d^2}. \tag{C23}$$

For a single $\widetilde{P}_D \in \overline{G}_D(U_t)$ from Eq. (C17) we can write

$$\Pr[\mathrm{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V)) = 0] \geqslant 1 - \frac{l|G_D(U_t)|}{d^2}. \tag{C24}$$

To have the probability for every Pauli $\widetilde{P}_D \in \overline{G}_D(U_t)$, we use the Fréchet bound on intersection of events:

$$\Pr\left[\bigcap_{\forall\widetilde{P}_D\in\overline{G}_D(U_t)} (U_t)\mathrm{tr}(\widetilde{P}_D(U_t)\widetilde{P}_D(V)) = 0\right] \geqslant 1 - \sum_{\alpha=1}^{|\overline{G}_D(U_t)|} \frac{l_\alpha|G_D(U_t)|}{d^2}. \tag{C25}$$

Using the fact that $2 \leqslant l_\alpha \leqslant 2^t$ for any α and that $|\overline{G}_D(U_t)||G_D(U_t)| = d_D^2 - |G_D(U_t)| \leqslant d_D^2$, we finally proved the statement:

$$\begin{aligned}&\Pr[\mathrm{tr}(P_D(U_t)P_D(V)) = 0,\ \forall\, P_D \in \mathbb{P}(D) \setminus G_D(U_t)]\\ &\geqslant 1 - \frac{2^t d_D^2}{d^2} = 1 - \frac{2^t}{d_C^2}.\end{aligned} \tag{C26}$$

■

Proof of Lemma 5. Recall Eq. (17) for $\mathbb{P}(D) = \mathbb{P}$:

$$\mathcal{F}_{\mathrm{V}}(U_t) = \frac{\langle \mathrm{tr}(P(U_t)P(V))\rangle_{P\in\mathbb{P}}}{d_A^2\,\langle \mathrm{tr}(P_AP(U_t)P_AP(V))\rangle_{P_A,P\in\mathbb{P}}}. \tag{C27}$$

Computing both average over $\mathbb{P}$ and using $\sum_P PAP = d\,\mathrm{tr}(A)$, we have

$$\mathcal{F}_{\mathrm{V}}(U_t) = \frac{d^{-2}|\mathrm{tr}(U_tV^\dagger)|^2}{d_A^2 d^{-2}\sum_{P_A}\mathrm{tr}(P_A)^2} = \frac{|\mathrm{tr}(U_tV^\dagger)|^2}{d^2}. \tag{C28}$$

Let $V \in \mathcal{V}_{U_t}$, i.e., $V^\dagger PV = U_t^\dagger PU_t$ for every $P \in \mathbb{P}$. Then, the unitary fidelity can be written as

$$\frac{|\mathrm{tr}(U_tV^\dagger)|^2}{d^2} = \frac{1}{d^2}|G(U_t)| + \frac{1}{d^3}\sum_{P\notin G(U_t)} \mathrm{tr}(P(U_t)P(V)). \tag{C29}$$

Now $P(U_t)$ is at least a summation over two Pauli strings and therefore $d^{-1}\mathrm{tr}(P(U_t)P(V)) < 1$. We thus obtain the following bound:

$$\frac{|\mathrm{tr}(U_tV^\dagger)|^2}{d^2} < \frac{1}{d^2}[|G(U_t)| + |P \setminus G(U_t)|] = 1. \tag{C30}$$

■

APPENDIX D: FINITE-RESOLUTION LEMMA

In this section, we present the proof of the finite resolution of the expectation values of Pauli operators for the Choi state of a t-doped Clifford circuit U_t. Let us state it formally. Let us define the following set:

$$\mathcal{S}_{U_t} = \{P \in \mathbb{P} \,|\, \langle U_t|P|U_t\rangle \neq 0\}, \tag{D1}$$

as the set of Pauli operators having nonzero expectation value on the Choi state $|U_t\rangle$ associated with the t-doped Clifford circuit U_t. Define

$$\delta_t = \min_{\substack{P,Q\in\mathcal{S}_{U_t}\\ \langle P\rangle\neq\langle P'\rangle}} |\,\langle U_t|\big(P - P'\big)|U_t\rangle\,|. \tag{D2}$$

Then, the following lemma holds:

Lemma 7 (Finite-resolution lemma). Let U_t be a t-doped Clifford circuit. Let $|U_t\rangle$ be the Choi state associated with U_t. Then, the following bounds hold:

$$\min_{P\in S_{U_t}} |\langle U_t|P|U_t\rangle| \geqslant \frac{1}{3\sqrt{2}^{t-1}}\left(1-\frac{1}{\sqrt{2}}\right)^t, \tag{D3}$$

while

$$\delta_t \geqslant \frac{1}{6\sqrt{2}^{t-1}}\left(1-\frac{1}{\sqrt{2}}\right)^t. \tag{D4}$$

We proceed as follows: we first bound $\min_{P\in S_{U_t}} |\mathrm{tr}(P|U_t\rangle\langle U_t|)|$ and then we bound the gap δ_t, as the second will be just a trivial generalization of the first one. Before proving the statement, let us recall that the action of a T gate, defined as $T = \mathrm{diag}(1, e^{-i\pi/4})$ applied on the ith qubit on a Pauli operator P results in

$$T_i P T_i^\dagger = \begin{cases} P, & [P, Z] = 0 \\ \frac{1}{\sqrt{2}}(P - iZ_i P), & \{P, Z_i\} = 0. \end{cases} \tag{D5}$$

Note that, to bound $\min_{P\in S_{U_t}} |\langle U_t|P|U_t\rangle|$, we can alternatively bound $\min_P |\mathrm{tr}(U_t P U_t^\dagger \sigma)|$ for σ being an arbitrary stabilizer state. Let us look at the action of U_t on a Pauli operator. First decompose $U_t = \prod_{i=1}^t U_1^{(i)}$, where $U_1^{(i)}$ is a $(t=1)$-doped Clifford circuit. Let us set up the following notation:

$$U_1^{(1)} P U_1^{(1)\dagger} = x_{(0)} P_{(0)} + \frac{x_{(1)}}{\sqrt{2}} P_{(1)} + \frac{x_{(2)}}{\sqrt{2}} P_{(2)}, \tag{D6}$$

where $x_{(0)}, x_{(1)}, x_{(2)} \in \{-1, 0, +1\}$. Equation (D6) must be understood as there is a choice of $x_{(0)}, x_{(1)}, x_{(2)}$ and the respective Pauli operators $P_{(0)}, P_{(1)}, P_{(2)}$ such that the left-hand side (l.h.s.) is equal to the r.h.s. of Eq. (D6). Before generalizing to the generic t, it is useful to act again on $U_1^{(1)} P U_1^{(1)\dagger}$ with $U_1^{(2)}$:

$$\begin{aligned} &U_1^{(2)} U_1^{(1)} P U_1^{(1)\dagger} U_1^{(2)\dagger} \\ &\quad = x_{(00)} P_{(00)} + \frac{1}{\sqrt{2}}(x_{(10)} P_{(10)} \\ &\qquad + x_{(01)} P_{(01)} + x_{(20)} P_{(20)} + x_{(02)} P_{(02)}) \\ &\qquad + \frac{1}{2}(x_{(11)} P_{(11)} + x_{(12)} P_{(12)} + x_{(21)} P_{(21)} + x_{(22)} P_{(22)}), \end{aligned}$$

where each variable $x_{(ij)}$ for $i = 0, 1, 2$ can take values in $x_{(ij)} \in \{-1, 0, +1\}$. As one can see, the subscript string (ij) attached to each variable $x_{(ij)}$ reveals how many times a T-gate splits the Pauli operator P in 2 Pauli operator with the corresponding $1/\sqrt{2}$ factor. Again, Eq. (D7) must be understood as there exists a choice of the variables $x_{(ij)}$ and the respective Pauli operators $P_{(ij)}$ for which the l.h.s. and the r.h.s. of Eq. (D7) agrees. Now, that we set up the above general and powerful notation, we can easily generalize the action to U_t. We have the following:

$$U_t P U_t^\dagger = \sum_{k=0}^{t} \sum_{\pi\in S_k} \frac{x_{\pi(\mathbf{y}_k)}}{\sqrt{2}^k} P_{\pi(\mathbf{y}_k)}. \tag{D7}$$

In Eq. (D7) above, we have defined a few elements. First of all, we defined the t-bit string $\mathbf{y}_k$ with Hamming weight k as

$$\mathbf{y}_k = \left(\underbrace{1, 1, \ldots, 1}_{k}, 0, \ldots, 0\right). \tag{D8}$$

Next, we defined a set S_k of operations π that act on $\mathbf{y}_k$. S_k is the set containing all the permutations of the $k1$ in $\mathbf{y}_k$ into t spots, combined with the operation that transforms $1 \leftrightarrow 2$, in accordance with Eq. (D7). Let us illustrate this with an example. Set $t = 2$ and $k = 1$, so $\mathbf{y}_1 = (10)$. All the possible permutations of (10), combined with the operation $1 \leftrightarrow 2$, result in the strings (10),(01),(20),(02). Similarly, for $t = 2$ and $k = 2$, $\mathbf{y}_2 = (11)$ and the set of operations in S_2 returns (11),(22),(21),(12).

It is useful to count the number of operations within S_k for fixed k. The set S_k is the combination of $\binom{t}{k}$ many ways to permute $\mathbf{y}_k = (1, 1, \ldots, 1, 0, \ldots, 0)$ times the 2^k different choices of either 1 or 2 at any site. The above simple counting thus returns

$$\sum_{\pi\in S_k} = 2^k \binom{t}{k}. \tag{D9}$$

From Eq. (D7), we can formally compute the expectation value of $U_t P U_t^\dagger$ with a generic stabilizer state σ and get

$$\mathrm{tr}(U_t P U_t^\dagger \sigma) = \sum_{k=0}^{t} \sum_{\pi\in S_k} \frac{\tilde{x}_{\pi(\mathbf{y}_k)}}{\sqrt{2}^k}, \tag{D10}$$

where we defined the variables $\tilde{x}_{\pi(\mathbf{y}_k)} := x_{\pi(\mathbf{y}_k)} \mathrm{tr}(P_{\pi(\mathbf{y}_k)}\sigma) \in \{-1, 0, +1\}$ because $\mathrm{tr}(P_{\pi(\mathbf{y}_k)}\sigma) \in \{-1, 0, +1\}$. Now, we set up all the necessary notation to finally prove Eq. (D3).

We are interested in computing the minimum achievable value for $\mathrm{tr}(U_t P U_t^\dagger \sigma)$. Let us first multiply both sides for $\sqrt{2}^t$. We thus get

$$\begin{aligned} \sqrt{2}^t \mathrm{tr}(U_t P U_t^\dagger \sigma) &= \sum_{k=0}^{t} \sum_{\pi\in S_k} \sqrt{2}^{t-k} \tilde{x}_{\pi(\mathbf{y}_k)} \\ &= \sum_{l=0}^{t} \sum_{\pi\in S_{t-l}} \sqrt{2}^l \tilde{x}_{\pi(\mathbf{y}_{t-l})}, \end{aligned}$$

where in the second equality, we defined $l = t - k$. Let us set t to be even and split odd and even terms in the sum

$$\begin{aligned} \sum_{l=0}^{t} \sum_{\pi\in S_{t-l}} \sqrt{2}^l \tilde{x}_{\pi(\mathbf{y}_{t-l})} &= \sum_{l=0}^{t/2} 2^l \sum_{\pi\in S_{t-2l}} \tilde{x}_{\pi(\mathbf{y}_{t-2l})} \\ &\quad + \sqrt{2} \sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} \tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})}. \end{aligned}$$

Define the following function of t:

$$\begin{aligned} A(t) &:= \sum_{l=0}^{t/2} 2^l \sum_{\pi\in S_{t-2l}} \tilde{x}_{\pi(\mathbf{y}_{t-2l})}, \\ B(t) &:= \sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} \tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})}. \end{aligned} \tag{D11}$$

Note that $A(t), B(t) \in \mathbb{Z}$, i.e., they are positive and negative natural numbers, for any $t = 2t'$ for $t \in \mathbb{N}$. We can thus write

$$\sqrt{2}^t |\mathrm{tr}(U_t P U_t^\dagger \sigma)| = |A(t) + \sqrt{2}B(t)| = |B(t)| \left|\sqrt{2} + \frac{A(t)}{B(t)}\right|. \tag{D12}$$

Therefore, the lower bound deals with the approximation of the algebraic number $\sqrt{2}$ by a rational number $A(t)/B(t)$. To make it explicit we can lower bound the r.h.s. of Eq. (D12) as

$$\sqrt{2}^t|\mathrm{tr}(U_tPU_t^\dagger\sigma)| \geqslant |B(t)|\left|\sqrt{2}-\frac{|A(t)|}{|B(t)|}\right|, \tag{D13}$$

and we can invoke the Liouville Theorem (see Ref. [106]) of approximating a algebraic number α with two rational numbers $p, q \in \mathbb{Q}$ that reads as follows: There exist a constant $c(\alpha)$ independent from p, q such that

$$\left|\sqrt{2}-\frac{p}{q}\right| \geqslant \frac{c(\alpha)}{q^D}, \tag{D14}$$

where D is the degree of the algebraic number α. In the case of $\alpha = \sqrt{2}$ we have $D = 2$ because $\sqrt{2}$ corresponds to the solution to the irreducible polynomial $z^2 - 2 = 0$ which has degree 2, and $c(\sqrt{2}) = \frac{1}{6}$ [106]. Applying Eq. (D14) to Eq. (D13), we thus get

$$|\mathrm{tr}(U_tPU_t^\dagger\sigma)| \geqslant \frac{1}{6|B(t)|\sqrt{2}^t}. \tag{D15}$$

We now are just left to find an upper bound to $B(t)$. We proceed with the following equality:

$$\begin{aligned}
|B(t)| &= \left|\sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} \tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})}\right| \\
&\leqslant \sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} |\tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})}| \\
&= \sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} = \sum_{l=0}^{t/2-1} 2^l 2^{t-(2l+1)}\binom{t}{2l+1} \\
&= \frac{1}{\sqrt{8}}2^t\left[\left(1+\frac{1}{\sqrt{2}}\right)^t - \left(1-\frac{1}{\sqrt{2}}\right)^t\right] \\
&\leqslant \frac{1}{\sqrt{8}}\left(1-\frac{1}{\sqrt{2}}\right)^{-t},
\end{aligned}$$

where in the second equality, we used the fact that $\tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})} \in \{-1, 0, +1\}$ and in the last inequality, we used the fact that

$$\begin{aligned}
&\frac{1}{\sqrt{8}}2^t\left[\left(1+\frac{1}{\sqrt{2}}\right)^t - \left(1-\frac{1}{\sqrt{2}}\right)^t\right] \\
&\leqslant \frac{1}{\sqrt{8}}2^t\left(1+\frac{1}{\sqrt{2}}\right)^t = \frac{1}{\sqrt{8}}\left(1-\frac{1}{\sqrt{2}}\right)^{-t}.
\end{aligned} \tag{D16}$$

An analogous procedure with $t = 2t' + 1$ with t' leads to the same exact bound. Therefore for any $t \in \mathbb{N}$, we find

$$|\mathrm{tr}(C_tPC_t^\dagger\sigma)| \geqslant \frac{\sqrt{8}}{6}\left(\frac{1}{\sqrt{2}}-\frac{1}{2}\right)^t. \tag{D17}$$

Now, let us turn to analyze the gap $\delta_t \equiv \mathrm{tr}[(P - P')\psi_t]$. Using the same notation as before, we can write the adjoint action of C_t on $P - P'$ as follows:

$$U_t(P-P')U_t^\dagger = \sum_{k=0}^{t}\sum_{\pi\in S_k}\left(\frac{x_{\pi(\mathbf{y}_k)}}{\sqrt{2}^k}P_{\pi(\mathbf{y}_k)} + \frac{x'_{\pi(\mathbf{y}_k)}}{\sqrt{2}^k}P'_{\pi(\mathbf{y}_k)}\right), \tag{D18}$$

and, therefore, by repeating the same procedure, we can define

$$\begin{aligned}
A'(t) &:= \sum_{l=0}^{t/2} 2^l \sum_{\pi\in S_{t-2l}} \tilde{x}_{\pi(\mathbf{y}_{t-2l})} + \tilde{x}'_{\pi(\mathbf{y}_{t-2l})}, \\
B'(t) &:= \sum_{l=0}^{t/2-1} 2^l \sum_{\pi\in S_{t-(2l+1)}} \tilde{x}_{\pi(\mathbf{y}_{t-(2l+1)})} + \tilde{x}'_{\pi(\mathbf{y}_{t-(2l+1)})},
\end{aligned} \tag{D19}$$

and write the gap as

$$\begin{aligned}
\delta_t &= \frac{1}{\sqrt{2}^t}|A'(t)+\sqrt{2}B'(t)| \\
&\geqslant \frac{|B'(t)|}{\sqrt{2}^t}\left|\sqrt{2}-\frac{|A'(t)|}{|B'(t)|}\right| \geqslant \frac{1}{6\sqrt{2}^t|B'(t)|}.
\end{aligned} \tag{D20}$$

Following the inequalities in Eq. (D16), one can find

$$|B'(t)| \leqslant \frac{1}{\sqrt{2}}\left(1-\frac{1}{\sqrt{2}}\right)^{-t}, \tag{D21}$$

which recovers the desired result in Eq. (D4):

$$\delta_t \geqslant \frac{\sqrt{2}}{6}\left(\frac{1}{\sqrt{2}}-\frac{1}{2}\right)^t. \tag{D22}$$

[1] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Phys. Rev. A **77**, 012307 (2008).

[2] J. J. Wallman and S. T. Flammia, New J. Phys. **16**, 103032 (2014).

[3] I. Roth, R. Kueng, S. Kimmel, Y.-K. Liu, D. Gross, J. Eisert, and M. Kliesch, Phys. Rev. Lett. **121**, 170502 (2018).

[4] P. Hosur, X.-L. Qi, D. A. Roberts, and B. Yoshida, J. High Energy Phys. (online) 02 (2016) 004.

[5] J. S. Cotler, D. Ding, and G. R. Penington, Ann. Phys. (NY) **396**, 318 (2018).

[6] T. Xu, T. Scaffidi, and X. Cao, Phys. Rev. Lett. **124**, 140602 (2020).

[7] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, Nat. Commun. **5**, 4213 (2014).

[8] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New J. Phys. **18**, 023023 (2016).

[9] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya *et al.*, Chem. Rev. (Washington, DC, U. S.) **119**, 10856 (2019).

[10] W. J. Huggins, J. R. McClean, N. C. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush, npj Quantum Inf. **7**, 23 (2021).

[11] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Chem. Rev. (Washington, DC, U. S.) **120**, 12685 (2020).

[12] C. A. Fuchs and J. Van De Graaf, IEEE Trans. Inf. Theory **45**, 1216 (1999).

[13] B. W. Reichardt, F. Unger, and U. Vazirani, Nature (London) **496**, 456 (2013).

[14] D. Mills, A. Pappa, T. Kapourniotis, and E. Kashefi, in *Proceedings 14th International Conference on Quantum Physics and Logic, QPL 2017, Nijmegen, The Netherlands, 3-7 July 2017* (Open Publishing Association, 2017), pp. 209–221.

[15] J. F. Fitzsimons, npj Quantum Inf. **3**, 23 (2017).

[16] J. F. Fitzsimons and E. Kashefi, Phys. Rev. A **96**, 012303 (2017).

[17] A. Coladangelo, A. B. Grilo, S. Jeffery, and T. Vidick, in *Advances in Cryptology – EUROCRYPT 2019*, Lecture Notes in Computer Science, edited by Y. Ishai and V. Rijmen (Springer International Publishing, Cham, 2019), pp. 247–277.

[18] U. Mahadev, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Paris, France, 2018), pp. 259–267.

[19] A. Gheorghiu, T. Kapourniotis, and E. Kashefi, Theory Comput. Syst. **63**, 715 (2019).

[20] A. Gheorghiu and T. Vidick, in *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Baltimore, Maryland, 2019), pp. 1024–1033.

[21] I. Šupić and J. Bowles, Quantum **4**, 337 (2020).

[22] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. **74**, 145 (2002).

[23] P. Hayden and J. Preskill, J. High Energy Phys. (online) 09 (2007) 120.

[24] B. Yoshida and A. Kitaev, arXiv:1710.03363.

[25] R. A. Low, Phys. Rev. A **80**, 052314 (2009).

[26] C.-Y. Lai and H.-C. Cheng, IEEE Trans. Inf. Theory **68**, 3951 (2022).

[27] D. Gottesman, *The Heisenberg representation of quantum computers*, Group22: *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis (International Press, Cambridge, 1999), pp. 32–43.

[28] S. F. E. Oliviero, L. Leone, S. Lloyd, and A. Hamma, companion paper, Unscrambling quantum information with Clifford decoders, Phys. Rev. Lett. **132**, 080402 (2024).

[29] L. Leone, S. F. E. Oliviero, Y. Zhou, and A. Hamma, Quantum **5**, 453 (2021).

[30] I. L. Chuang and M. A. Nielsen, J. Mod. Opt. **44**, 2455 (1997).

[31] A. M. Childs, I. L. Chuang, and D. W. Leung, Phys. Rev. A **64**, 012314 (2001).

[32] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White, Phys. Rev. Lett. **90**, 193601 (2003).

[33] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, Phys. Rev. A **77**, 032322 (2008).

[34] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, Phys. Rev. A **87**, 062119 (2013).

[35] D. A. Roberts and B. Yoshida, J. High Energy Phys. (online) 04 (2017) 121.

[36] L. Leone, S. F. E. Oliviero, and A. Hamma, Entropy **23**, 1073 (2021).

[37] S. F. E. Oliviero, L. Leone, F. Caravelli, and A. Hamma, SciPost Phys. **10**, 076 (2021).

[38] A. R. Brown and L. Susskind, Phys. Rev. D **97**, 086015 (2018).

[39] S. F. E. Oliviero, L. Leone, and A. Hamma, Phys. Lett. A **418**, 127721 (2021).

[40] Z.-C. Yang, A. Hamma, S. M. Giampaolo, E. R. Mucciolo, and C. Chamon, Phys. Rev. B **96**, 020408(R) (2017).

[41] C. Chamon, A. Hamma, and E. R. Mucciolo, Phys. Rev. Lett. **112**, 240501 (2014).

[42] L. Leone, S. F. E. Oliviero, S. Piemontese, S. True, and A. Hamma, Phys. Rev. A **106**, 062434 (2022).

[43] M. Holzäpfel, T. Baumgratz, M. Cramer, and M. B. Plenio, Phys. Rev. A **91**, 042129 (2015).

[44] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum **3**, 140 (2019).

[45] K. W. Moore Tibbetts, C. Brif, M. D. Grace, A. Donovan, D. L. Hocker, T.-S. Ho, R.-B. Wu, and H. Rabitz, Phys. Rev. A **86**, 062309 (2012).

[46] T. Volkoff, Z. Holmes, and A. Sornborger, PRX Quantum **2**, 040327 (2021).

[47] I. Marvian and S. Lloyd, Universal quantum emulator, arXiv:1606.02734.

[48] S. True and A. Hamma, Quantum **6**, 818 (2022).

[49] S. Piemontese, T. Roscilde, and A. Hamma, Phys. Rev. B **107**, 134202 (2023).

[50] A. Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997).

[51] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[52] A. Montanaro and T. J. Osborne, Chic. J. Theor. Comput. Sci. **16**, 1 (2010).

[53] A. Montanaro, arXiv:1707.04012.

[54] R. Levy, D. Luo, and B. K. Clark, Phys. Rev. Res. **6**, 013029 (2024).

[55] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, Phys. Rev. Lett. **126**, 190501 (2021).

[56] B. T. Kiani, S. Lloyd, and R. Maity, arXiv:2001.11897.

[57] S. Aaronson, Proc. R. Soc. London, Ser. A **463**, 3089 (2007).

[58] S. Aaronson, in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, 2018), pp. 325–338.

[59] A. Rocchetto, Quantum Inf. Comput. **18**, 541 (2018).

[60] A. Rocchetto, S. Aaronson, S. Severini, G. Carvacho, D. Poderini, I. Agresti, M. Bentivegna, and F. Sciarrino, Sci. Adv. **5**, eaau1946 (2019).

[61] A. Gollakota and D. Liang, Quantum **6**, 640 (2022).

[62] D. Liang, Quantum **7**, 1036 (2023).

[63] S. Zhou, Z.-C. Yang, A. Hamma, and C. Chamon, SciPost Phys. **9**, 087 (2020).

[64] S. Bravyi and D. Gosset, Phys. Rev. Lett. **116**, 250501 (2016).

[65] D. P. DiVincenzo, D. W. Leung, and B. M. Terhal, IEEE Trans. Inf. Theory **48**, 580 (2002).

[66] J. Emerson, Y. S. Weinstein, M. Saraceno, S. Lloyd, and D. G. Cory, Science **302**, 2098 (2003).

[67] H. Zhu, R. Kueng, M. Grassl, and D. Gross, arXiv:1609.08172.

[68] J. Haferkamp, F. Montealegre-Mora, M. Heinrich, J. Eisert, D. Gross, and I. Roth, Commun. Math. Phys. **397**, 995 (2023).

[69] M. Hinsche, M. Ioannou, A. Nietner, J. Haferkamp, Y. Quek, D. Hangleiter, J.-P. Seifert, J. Eisert, and R. Sweke, Phys. Rev. Lett. **130**, 240602 (2023).

[70] K. Pietrzak, in *SOFSEM 2012: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, edited by M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, and G. Turán (Springer, Berlin, Heidelberg, 2012), pp. 99–114.

[71] A. Nahum, S. Vijay, and J. Haah, Phys. Rev. X **8**, 021014 (2018).

[72] V. Khemani, A. Vishwanath, and D. A. Huse, Phys. Rev. X **8**, 031057 (2018).

[73] C. Chamon, E. R. Mucciolo, and A. E. Ruckenstein, Ann. Phys. (NY) **446**, 169086 (2022).

[74] D. N. Page, Phys. Rev. Lett. **71**, 1291 (1993).

[75] B. Yoshida and N. Y. Yao, Phys. Rev. X **9**, 011006 (2019).

[76] B. Yoshida, arXiv:2109.08691.

[77] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004).

[78] L. Leone, S. F. E. Oliviero, and A. Hamma, Phys. Rev. A **107**, 022429 (2023).

[79] J. Jiang and X. Wang, Phys. Rev. Appl. **19**, 034052 (2023).

[80] L. Leone, S. F. E. Oliviero, and A. Hamma, Phys. Rev. Lett. **128**, 050402 (2022).

[81] J. Dehaene and B. De Moor, Phys. Rev. A **68**, 042318 (2003).

[82] D. Gosset, D. Grier, A. Kerzner, and L. Schaeffer, arXiv:2009.03218.

[83] S. Bravyi and D. Maslov, IEEE Trans. Inf. Theory **67**, 4546 (2021).

[84] D. Ding, P. Hayden, and M. Walter, J. High Energy Phys. (online) 12 (2016) 145.

[85] J. S. Cotler, G. Gur-Ari, M. Hanada, J. Polchinski, P. Saad, S. H. Shenker, D. Stanford, A. Streicher, and M. Tezuka, J. High Energy Phys. (online) 05 (2017) 118.

[86] J. Cotler, N. Hunter-Jones, J. Liu, and B. Yoshida, J. High Energy Phys. (online) 11 (2017) 048.

[87] X. Chen and T. Zhou, arXiv:1804.08655.

[88] Y.-L. Zhang, Y. Huang, X. Chen *et al.*, Phys. Rev. B **99**, 014303 (2019).

[89] B. Yan, L. Cincio, and W. H. Zurek, Phys. Rev. Lett. **124**, 160603 (2020).

[90] X. Mi, P. Roushan, C. Quintana, S. Mandra, J. Marshall, C. Neill, F. Arute, K. Arya, J. Atalaya, R. Babbush *et al.*, Science **374**, 1479 (2021).

[91] Q. Zhuang, T. Schuster, B. Yoshida, and N. Y. Yao, Phys. Rev. A **99**, 062334 (2019).

[92] G. Styliaris, N. Anand, and P. Zanardi, Phys. Rev. Lett. **126**, 030601 (2021).

[93] A. Touil and S. Deffner, PRX Quantum **2**, 010306 (2021).

[94] R. J. Garcia, Y. Zhou, and A. Jaffe, Phys. Rev. Res. **3**, 033155 (2021).

[95] E. T. Campbell and D. E. Browne, Phys. Rev. Lett. **104**, 030503 (2010).

[96] E. T. Campbell, H. Anwar, and D. E. Browne, Phys. Rev. X **2**, 041021 (2012).

[97] V. Veitch, S. A. H. Mousavian, D. Gottesman, and J. Emerson, New J. Phys. **16**, 013009 (2014).

[98] N. Koukoulekidis and D. Jennings, npj Quantum Inf. **8**, 42 (2022).

[99] G. Saxena and G. Gour, Phys. Rev. A **106**, 042422 (2022).

[100] S. F. E. Oliviero, L. Leone, A. Hamma, and S. Lloyd, npj Quantum Inf. **8**, 148 (2022).

[101] S. F. E. Oliviero, L. Leone, and A. Hamma, Phys. Rev. A **106**, 042426 (2022).

[102] O. Hahn, A. Ferraro, L. Hultquist, G. Ferrini, and L. Garcia-Alvarez, Phys. Rev. Lett. **128**, 210502 (2022).

[103] T. Haug and M. S. Kim, PRX Quantum **4**, 010301 (2023).

[104] S. Goldstein, J. L. Lebowitz, R. Tumulka, and N. Zanghì, in *Statistical Mechanics and Scientific Explanation* (World Scientific, 2019), pp. 519–581.

[105] E. Van Den Berg, *2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA* (IEEE, 2021), pp. 54–59.

[106] M. R. Murty and P. Rath, in *Transcendental Numbers*, edited by M. R. Murty and P. Rath (Springer, New York, 2014), pp. 1–6.