Neural-Network Decoders for Measurement Induced Phase Transitions

Hossein Dehghani,[1, 2, *] Ali Lavasani,[1, 3] Mohammad Hafezi,[1, 2] and Michael J. Gullans[2]

[1] *Joint Quantum Institute, NIST/University of Maryland, College Park, MD 20742, USA*
[2] *Joint Center for Quantum Information and Computer Science,*
NIST/University of Maryland, College Park, MD 20742, USA
[3] *Condensed Matter Theory Center, University of Maryland, College Park, MD 20742, USA*

Open quantum systems have been shown to host a plethora of exotic dynamical phases. Measurement-induced entanglement phase transitions in monitored quantum systems are a striking example of this phenomena. However, naive realizations of such phase transitions requires an exponential number of repetitions of the experiment which is practically unfeasible on large systems. Recently, it has been proposed that these phase transitions can be probed locally via entangling reference qubits and studying their purification dynamics. In this work, we leverage modern machine learning tools to devise a neural network decoder to determine the state of the reference qubits conditioned on the measurement outcomes. We show that the entanglement phase transition manifests itself as a stark change in the learnability of the decoder function. We study the complexity and scalability of this approach and discuss how it can be utilized to detect entanglement phase transitions in generic experiments.

INTRODUCTION

Entanglement entropy in closed quantum systems that thermalize generically tends to increase until reaching a volume-law behavior with entanglement spread throughout the system [1, 2]. Coupling to a bath profoundly changes the internal evolution of the system [3], which in turn can suppress the growth of entanglement and correlations within the system to an area-law behavior [4, 5]. A prominent example of such systems are random quantum circuits with intermediate measurements [6–10]. In these circuits, where the unitary time evolution of the system is interspersed by quantum measurements, the competition between unitary and non-unitary elements leads to a measurement-induced phase transition between a pure phase with an area-law and a mixed phase with a volume-law entanglement behavior [11–31]. Such entanglement phase transitions are only accessible when the density matrix is conditioned on the measurement outcomes while they are hidden from any observable which can be expressed as a linear function of the density matrix. On the other hand, to experimentally probe observables which are non-linear functions of the density matrix, one naively needs to reproduce multiple copies of the same state. However, due to intrinsic randomness in measurement outcomes, this naive approach requires repeating the experiment exponentially many times (in system size) [10, 26].

Building on the close connection between measurement-induced entanglement phase transitions and quantum error correction [11, 12, 32–35], a possible workaround to this obstacle was found in Ref. [36], where it was proposed to probe these phase transitions through purification dynamics of an ancilla reference qubit that is initially entangled to local system degrees of freedom. Subsequently, the time dependence of the entanglement entropy of the reference qubits signifies the phase transition properties [11, 15, 36]. To employ this method, one needs to find the density matrix of reference qubits conditioned on the measurement outcomes of the circuit. Hence, the final objective in this approach is to obtain a "decoder" that maps the measurement outcomes to the density matrix of the reference qubit. However, such decoders are only known and implemented for special classes of circuits such as stabilizer circuits [9]. For more generic circuits like Haar-random circuits, finding an analytical solution to this problem is likely unfeasible.

Here, motivated by the the recent successful applications of machine learning algorithms in quantum sciences [37] and especially optimizing quantum error correction codes and quantum decoders [38–45], we provide a generic neural network approach that can efficiently find the aforementioned decoders. First, we sketch our physically motivated neural network architecture. Although we use numerical simulations of Clifford circuits to show the efficacy of our neural network (NN) decoder, we argue that the same neural network decoder with slight modifications should in principle work for any generic circuit. We investigate the complexity of our learning task by studying the number of circuit runs required for training the neural network decoder. Importantly, we show that the learning task only needs measurement outcomes inside a rectangle encompassing the statistical light-cone [19, 36] of the reference qubit. Furthermore, we demonstrate that by studying the temporal behavior of the learnability of the quantum trajectories, one can estimate the critical properties of the phase transition. Finally, we verify that for large circuits one can train the neural networks over smaller circuits which proves the scalablity of our method.

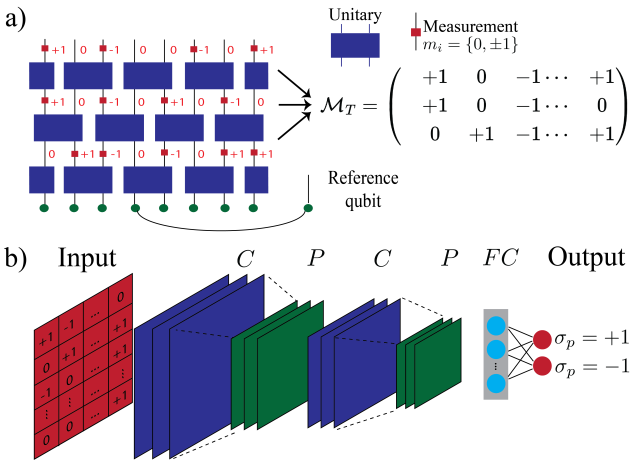
a)
Unitary
Measurement
$m_i = \{0, \pm 1\}$
$\mathcal{M}_T = \begin{pmatrix} +1 & 0 & -1\cdots & +1 \\ +1 & 0 & -1\cdots & 0 \\ 0 & +1 & -1\cdots & +1 \end{pmatrix}$
Reference qubit
b)
Input
C
P
C
P
FC
Output
$\sigma_p = +1$
$\sigma_p = -1$

Figure 1. (a) Brickwall structure of hybrid circuit with random two-qubit Clifford gates interspersed with projective Z measurements. $\mathcal{M}_T$ denotes the measurement outcome matrix with matrix elements $m_i = 0, \pm 1$, where $T = 3$ for this example. (b) Neural network architechure: We use convolutional neural networks composed of (C:convolutional), (P:pooling) and (F:fully connected) layers, used to learn the decoder function which is trained by the measurement outcomes of quantum trajectories in order to predict the measurement result of the reference qubit σ_p.

MODEL

The circuits that we study have a brickwork structure as in Fig. 1, with L qubits. We consider time evolution with T time steps with repetitive layers of two-qubit random unitary gates, followed by a round of single-site measurements of the Pauli Z operators at each site with probability p. As one tunes p past some critical value p_c, there is a phase transition from a volume-law entanglement behavior ($p < p_c$) to an area-law behavior ($p > p_c$) and a logarithmic scaling at the critical point ($p = p_c$). Crucially for this work, this phase transition is also manifested in the time dependence of the entanglement entropy of a reference qubit entangled with the system $S_Q(t)$ [36]. $S_Q(T)$, averaged over many circuit runs, is known as the coherent quantum information and plays a crucial role in the fundamental theory of quantum error correction [46]. For polynomial in system size circuit depths, $S_Q(T)$ maintains a finite value in the volume-law phase and vanishes in the area law phase. The protocol we use to probe $S_Q(t)$ is illustrated in Fig. 1a. Starting from a pure product state, we make a Bell pair out of the qubit in the middle and an ancilla reference qubit. Throughout the paper, we use periodic boundary conditions for the circuit.

Decoder. To find $S_Q(T)$ in experiment, we need to find the density matrix of the reference qubit at time T, which is a vector inside the Bloch sphere and can be specified by its three components $\langle\sigma_X\rangle$, $\langle\sigma_Y\rangle$ and $\langle\sigma_Z\rangle$. Therefore, probing the phase transition can be viewed as the task of finding a decoder function $F_\mathcal{C}$ for a given circuit $\mathcal{C}$, such that

$$F_\mathcal{C}(\mathcal{M}_T) = (\langle\sigma_X\rangle, \langle\sigma_Y\rangle, \langle\sigma_Z\rangle) \qquad (1)$$

where $\mathcal{M}_T$ is the set of circuit measurement outcomes. Let $p_P(m|\mathcal{M}_T)$ for $P \in \{X, Y, Z\}$ denote the probability of getting reference qubit outcome $m = \pm 1$ when measuring σ_P of the reference qubit after time $t = T$, conditioned on the measurement outcomes $\mathcal{M}_T$. Since $\langle\sigma_P\rangle = \sum_{m=\pm 1} m\ p_P(m|\mathcal{M}_T)$, the problem of finding the decoder $F_\mathcal{C}$ is equivalent to finding the probability distributions $p_P(m|\mathcal{M}_T)$ for $P \in \{X, Y, Z\}$.

DEEP LEARNING ALGORITHM

Instead of finding $p_P(m|\mathcal{M}_T)$ analytically for a given circuit $\mathcal{C}$, we plan to use ML methods to learn these functions from a set of sampled data points which in principle could be obtained from experiments. The task of learning conditional probability distributions is known as the probabilistic classification task in ML literature [47, 48]. Let us fix the circuit $\mathcal{C}$ and the Pauli P. A sample data point is a pair of $(\mathcal{M}_T, m)$ for a single run of the circuit where $\mathcal{M}_T$ is the circuit measurement outcomes and m is the of outcome of measuring the reference qubit in the σ_P basis at the end of the circuit. By repeating the experiment N_t times, we can generate a training set of N_t data points. By training a neural network using this data set, we obtain a neural network representation of the function $p_P(m|\mathcal{M}_T)$.

Framing the problem as a probabilistic classification task does not necessarily mean that the learning task would be efficient. Indeed, given that the number of different possible $\mathcal{M}_T$ outcomes scales exponentially with the system size, one would naively expect that the minimum required N_t should also scale exponentially for the learning task to succeed, i.e., we need to run the circuit exponential number of times to generate the required training data set. However, the crucial point made in Ref. [36] is that, when the reference qubit is initially entangled locally to the system, its density matrix at the end of the circuit only depends on the measurement outcomes that lie inside a statistical light cone, and up to a depth bounded by the correlation time that is finite away from the critical point. Hence, for a typical circuit away from critically, the function $p_P(m|\mathcal{M}_T)$ depends only on a finite number of elements in $\mathcal{M}_T$ and that makes the learning task feasible.

To show the effectiveness of this method, we test our decoder using data points gathered from numerical simulation of Clifford circuits with $p_c = 0.160(1)$ [8], which enables us to study circuits of large enough sizes. Due to Clifford dynamics, the reference qubit either remains completely mixed at $t = T$ or it is purified along one of the Pauli axis. This means the measurement outcome of σ_P at the end of the circuit is either deterministic or completely random. Therefore, it is more natural to view the problem as a hard classification task (rather than probabilistic) where we train the neural network to determine the measurement outcome of σ_P (See the methods section). Note, if the reference qubit is purified at the end of the circuit, then the decoder can in principle learn the decoding function while, if it is not, then the measurement outcomes are completely random, leading to an inevitable failure of the hard classification. Thus, the purification phase transition shows itself as a learnability phase transition. It is worth noting that we are only changing how we interpret the output of the NN, i.e. we pick the label with highest probability, so the same NN architecture can be used for more generic gate sets. For simplicity, we also only look at the data points corresponding to the basis P in which the reference qubit is purified. In an experiment, the purification axis is not known, so one needs to train the NN for each of the three choices of P; if the learning task fails for all of them, it means the qubit is totally mixed. Otherwise, the learning task will succeed for one axis and fail for the other two [49], which means the reference qubit is purified.

Since locality and causality structure play important roles in purification dynamics, we employ a particular deep learning [50–52] architecture called convolutional neural networks (CNN) that are efficient in detecting local features in image recognition applications [53]. In utilizing these networks the input data is treated as a snapshot as in Fig. 1(b) with each pixel treated as a feature of the neural network and the label of each image is the measurement outcome of σ_P.

LEARNING COMPLEXITY

For a fixed circuit $\mathcal{C}$, we start the training procedure by training the neural network with a given number of labeled quantum trajectory measurements, and then evaluate its performance on predicting the labels of new randomly generated trajectories produced by the same circuit $\mathcal{C}$. The learning accuracy $1-\epsilon_l$ is the probability that the NN predicts the right label. The minimum number of training samples denoted by $M(\epsilon_l)$ to reach a specified learning error ϵ_l can provide an empirical measure of the learning complexity of the decoder function $F_\mathcal{C}$ [54]. In what follows, we fix the learning error of each circuit to be $\epsilon_l = 0.02$.

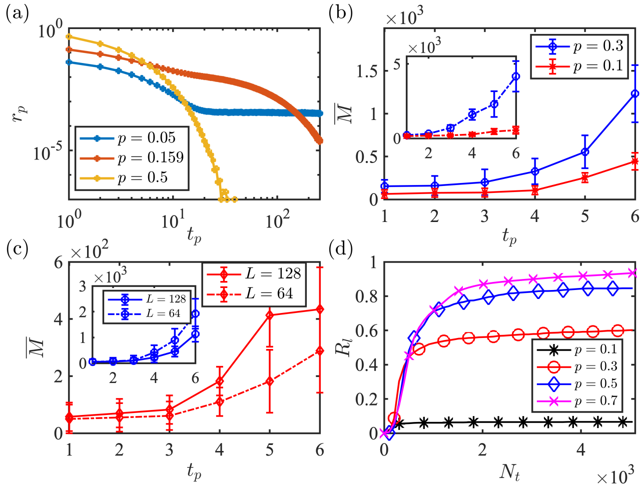

Figure 2. (a) Distribution of purified circuits as a function of the purification time for different measurement rates $p = 0.05$(mixed phase), $p \simeq p_c \simeq 0.16$ (critical value), $p = 0.5$ (pure phase) with $L = 16$ qubits and $N_c = 10^7$ random circuits. (b) and (c) Averaged number of quantum trajectories required for learning the reference qubit after postselection as a function of the purification time t_p, for $p = 0.1$ (mixed phase) and $p = 0.3$ (pure phase). Averaging is performed over $N_c = 20$ circuits for each t_p and error bars are set according to the standard deviation. In (b) we have circuits with $L = 128$ qubits. In the main plot measurement outcomes from inside the fixed light-cone box are used for training while for the inset we use the measurement outcomes from the whole circuit. In (c) we have circuits with $L = 128$ qubits (solid-line) and $L = 64$ qubits (dashed-line) with $p = 0.1$ in the main plot and $p = 0.3$ in the inset. (d) Ratio of learned circuits as a function of number of quantum trajectories with $L = 64$ and for different p without postselection on the purification time with $N_c = 10^3$ circuits for each p.

In performing this analysis, different learning settings can be considered. Intuitively, for a fixed circuit, we expect the purification time of the reference qubit, t_p, after which the reference qubit's state does not alter any further, to play an important role in determining M. Therefore, in our first learning setup, we consider a postselected learning scheme where for a given measurement rate, we postselect quantum trajectories based on their purification time t_p, which allows us to study the effect of system size on the learning complexity. Moreover, we discard measurement outcomes corresponding to measurements performed after t_p. Here, we note that given N_c circuits with the same purification time, in addition to the learning efficiency of each circuit, we need to fix the learning inaccuracy averaged over N_c circuits, δ_l, which we fix to be $\delta_l = 20\%$. We remark that this number is larger than ϵ_l since some of the postselected circuits have not been learned.

In the second setting, we remove the postselection constraint and only consider the overall complexity of the learning task when we randomly generate circuits for a given p. The two schemes can be related using the probability distribution r_p of the purification time as shown in Fig. 2(a) and explained more concretely in the methods section. Note that since the reference qubit is entangled locally at the beginning, there is always a finite probability that it will be purified in early times. In the mixed phase, the distribution has a exponentially small tail until exponentially long times (both in system size) whereas in the pure phase, the ancilla purifies in a constant time independent of system size. Inspired by the approximate locality structure of hybrid circuits [36], we also consider a light-cone learning scheme, where we train the NN using only the measurement outcomes inside a box centered at the middle (see Fig. 4(b)). In Fig. 2(b), we compare the postselected complexity of the learning task in the pure and mixed phases both by using the light-cone box (main) and whole circuit (inset) measurement data. For each purification time and p, we consider $N_c = 20$ different circuits and we average over their minimum required training numbers to calculate $\bar{M}(\epsilon_l, \delta_l)$, and show the standard deviation as the error bar. Here, for all the curves, we observe an approximate exponential growth of $\bar{M}(\epsilon_l, \delta_l)$ as a function of the purification time t_p. By comparing the mixed and pure phases, we notice that the postselected learning task is more complicated in the pure phase than the mixed phase, which is expected since, all else being equal, there are more measurements in the pure phase. Additionally, as shown in the inset, we find that learning with light-cone data is less complicated than using all the measurement outcomes. These behaviors can be understood by recognizing that to learn the decoder we need to explore the domain of the mapping in Eq.(1) whose size scales exponentially with 2^{pTL}.

In Fig. 2(c) we compare the system size dependence of the complexity in the two phases with $L = 64, 128$ where we train our networks with the light-cone data. We note that since the size of the light-cone box for a fixed t_p is independent of the system-size, the complexity should be roughly independent of the system size. Taking the error bars into consideration, our numerical observation is partially in agreement with this theoretical expectation. We remark that, as shown in the methods section, similar complexity results are obtained for circuits with initial states scrambled by a high-depth random Clifford circuit.

In the final step, we consider the learning task without postselection. Fig. 2(d) shows the ratio of circuits that can be learned, denoted by R_l, as a function of N_t, with the circuit depth fixed at $T = 10$.

After an initial fast growth in R_l, the learning procedure slows down. This can be understood by noting that exponentially more samples are required to learn the decoder for circuits with longer purification time. Moreover, the saturation value for each p is bounded by the ratio of circuits that are purified by time T, which can be expressed as

$$R_p(T) = \int_0^T r_p \mathrm{d}t \qquad (2)$$

where r_p is the purification rate plotted in Fig. 2(a).

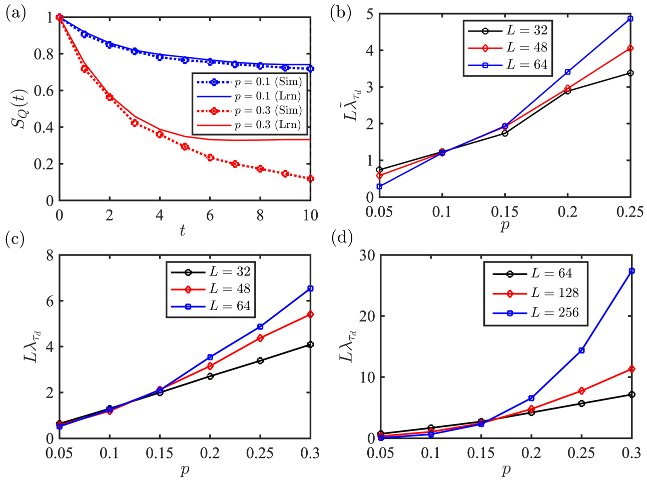

Figure 3. Temporal behavior of the reference qubit's entanglement entropy averaged over different circuit configurations for a given p. (a) Comparing the temporal behavior for a circuit with $L = 64$ qubits in the mixed ($p = 0.1$) and pure ($p = 0.3$) phases. The dashed and solid lines are achieved from learning quantum trajectories, and exact simulation of the circuits, respectively. (b) Temporal derivative of the learned entanglement entropy S_Q^l as a function of the measurement rate at a fixed scaled time $\tau_d = t_d/L = 1/16$. (c) Rate of the simulated entanglement entropy S_Q as a function of the measurement rate at the same $\tau_d = t_d/L = 1/16$ and L. (d) Rate of the simulated entanglement entropy for larger L. The crossing point matches the critical point, $p_c \simeq 0.16$.

DYNAMICS OF COHERENT INFORMATION

We can utilize the NN decoder to study the critical properties of the phase transition. For a fixed p, let $S_Q(t)$ denote the average entropy of the reference qubit after time t, i.e., the coherent quantum information of the system with 1 encoded qubit. We may assume on general grounds that $S_Q(t)$ follows an early time exponential decay $e^{-\lambda t}$ with λ following the scaling form

$$\lambda = L^{-z} f[(p - p_c)L^{z/\nu}], \qquad (3)$$

where z and ν are the dynamical and correlation length critical exponents respectively [9]. On the other hand, since $S_Q(t)$ and the ratio of purified circuits $R_p(t)$ are related by $S_Q = 1 - R_p$, we can estimate $S_Q(t)$ by the ratio of learnable circuits of depth t in the postselection free scheme described above, which we denote it by $\tilde{S}_Q$.

In Fig. 3(a) we compare the temporal behavior of the coherent information obtained from an ideal decoder and the NN decoder introduced here. As demonstrated in Fig. 3(a), in the mixed phase the learned entanglement entropy closely follows the simulated entanglement entropy, while in the pure phase the two curves start to deviate from each other after a few time steps. This behavior is consistent with previous observations in Fig. 2 where we demonstrated that the learning task is easier in the mixed phase. Since at the critical point this phase transition can be described by a $1+1$-D conformal field theory [6, 8], the dynamical critical exponent can be fixed in advance $z = 1$. Defining the scaled time $\tau = t/L$, based on the scaling relation in Eq.(3), we expect to see a crossing in

$$L\lambda_{\tau_d} \approx |\frac{d\ln S_Q}{d\tau}|_{\tau_d}. \tag{4}$$

when its plotted for different system sizes. Here, $\tau_d = t_d/L$ is the differentiation time which should be sufficiently large. In Fig. 3(b), we evaluate the decay rate obtained by learning, $\tilde{\lambda}_{\tau_d}$, for three different system sizes, $L = \{32, 48, 64\}$, at $\tau_d = 1/16$ using $\tilde{S}_Q$. The corresponding times are $t_d = \{2, 3, 4\}$ for which the deviation of the learned and simulated coherent information is negligible. Here, we notice an approximate crossing in the region $0.1 \lesssim p_c \lesssim 0.15$ which is consistent with the simulation results, $p_c \simeq 0.16$. To further verify this result, we compare our learning results, with similar results obtained from simulations of $S_Q(t)$ at the same τ_d and for the same L in Fig. 3(c). This plot shows a similar crossing behavior in the region $0.1 \lesssim p_c \lesssim 0.15$. Achieving better results for the p_c is possible by increasing τ_d or considering larger system sizes as displayed in Fig. 3(d) where we see a crossing at $p = 0.16 \pm 0.005$. Analogously, by applying our NN decoder to larger system sizes, we expect more accurate estimates of p_c.

SCALABILITY OF LEARNING

An important feature of a practical decoder is the possibility of training it on small circuits and then utilizing it for decoding larger circuits. Here, due to the approximate locality of the temporal evolution of the random hybrid circuits, one can examine the scalability of the decoders in a concrete manner. For a given circuit with L qubits, we generate smaller circuits with $L_B < L$ number of qubits which have identical gates as the original circuit in a rectangular narrow strip around the middle qubit which is entangled to the reference qubit. The geometry of the two sets of circuits is displayed in Fig. 4(a) where the depth of the two sets of circuits are chosen to be equal. Here, for each p we generate $N_c = 10^2$ large circuits with $L = 32$ with 10 time steps. We also postselect over circuits that are learnable using measurement outcomes from the original circuit. Next, for each of these circuits, for $L_B = \{4, 8, \cdots, 20\}$ we generate their corresponding smaller circuits and we run them to generate $N_t = 5 \times 10^3$ quantum trajectories. In the training step, we use the quantum trajectories produced from the smaller circuits to train our neural networks. In the testing step, however, we use these neural networks to make prediction for the quantum trajectories obtained from the larger circuits. As we observe in Fig. 4(b), by increasing L_B the ratio of the circuits that can be learned by the smaller circuits' measurements increase. Also, consistent with the effective light-cone picture, we see that by allowing longer purification times, larger L_B is required.

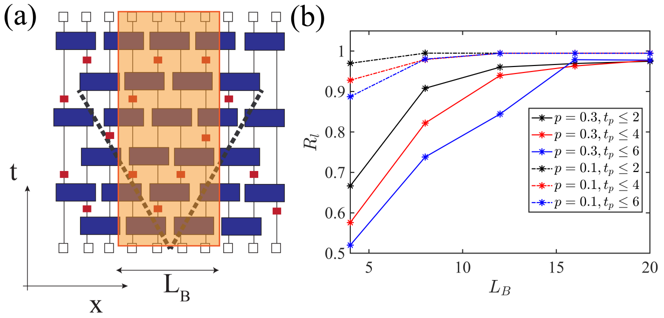

Figure 4. (a) Predicting the decoder function of a circuit with $L = 32$ qubits using the neural network trained by the measurement outcomes inside the small circuit in the orange box. Ratio of learned circuits with neural networks trained on the measurement outcomes from the smaller box with length L_B. (b) Ratio of learned circuits as a function of L_B and for different purification times. Circuits with longer purification times t_p need quantum trajectories from larger L_B.

OUTLOOK

There are a number of directions to extend these results. We first note that Clifford circuits are a special type of circuits, and it is important to consider more general circuits such as Haar-random circuits, where the purification axis can be along any radius in the Bloch sphere. From an experimental perspective, it is possible incorporate different errors, which are common in the realization of the two-qubit gates and/or measurement processes, in our machine learning framework. An intriguing possibility is to find neural network decoders that are successful in learning deep circuits with local data [35]. Similarly, implementing neural network decoders for $2D$ and higher dimensional measurement-induced phase transitions is an immediate extension of this work. In the context of quantum error correction and fault-tolerance, the purification dynamics in measurement-induced phase transitions leads to a rich set of examples of dynamically generated quantum error correcting codes [11, 32, 55, 56]. Designing similar decoders as considered here for other types of dynamically generated logical qubits is a rich avenue of investigation. Finally, we highlight that our empirical complexity results raise interesting questions about the complexity of learning an effective Hamiltonian description [57, 58] of the measurement outcome distributions for monitored quantum systems.

Acknowledgments.—We acknowledge stimulating discussions with Alireza Seif, David Huse, Pradeep Niroula, Crystal Noel, Grace Sommers, and Christopher

White. We acknowledge support from the National Science Foundation (JQI-PFC-UMD and QLCI grant OMA-2120757). H.D. and M.H. acknowledge support from ARO W911NF2010232, AFOSR FA9550-19-1-0399, NSF OMA-2120757, QSA-DOE and Simons and Minta Martin foundations. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), supported by the grant number PHY210049, at the Pittsburgh Supercomputing Center (PSC) [59]. M.H. thanks ETH Zurich for their hospitality during the conclusion of this work.

* hdehghan@umd.edu

* hdehghan@umd.edu

[1] H. Kim and D. A. Huse, Phys. Rev. Lett. **111**, 127205 (2013).

[2] R. Nandkishore and D. A. Huse, Annual Review of Condensed Matter Physics **6**, 15 (2015).

[3] H.-P. Breuer, F. Petruccione, *et al.*, *The theory of open quantum systems* (Oxford University Press on Demand, 2002).

[4] B. Bauer and C. Nayak, Journal of Statistical Mechanics: Theory and Experiment **2013**, P09005 (2013).

[5] M. Serbyn, Z. Papić, and D. A. Abanin, Phys. Rev. Lett. **111**, 127201 (2013).

[6] B. Skinner, J. Ruhman, and A. Nahum, Phys. Rev. X **9**, 031009 (2019).

[7] Y. Li, X. Chen, and M. P. A. Fisher, Phys. Rev. B **98**, 205136 (2018).

[8] Y. Li, X. Chen, and M. P. A. Fisher, Phys. Rev. B **100**, 134306 (2019).

[9] C. Noel, P. Niroula, D. Zhu, A. Risinger, L. Egan, D. Biswas, M. Cetina, A. V. Gorshkov, M. J. Gullans, D. A. Huse, and C. Monroe, "Measurement-induced quantum phases realized in a trapped-ion quantum computer," (2021), arXiv:2106.05881 [quant-ph].

[10] J. M. Koh, S.-N. Sun, M. Motta, and A. J. Minnich, arXiv:2203.04338 (2022).

[11] M. J. Gullans and D. A. Huse, Phys. Rev. X **10**, 041020 (2020).

[12] S. Choi, Y. Bao, X.-L. Qi, and E. Altman, Physical Review Letters **125**, 030505 (2020).

[13] C.-M. Jian, Y.-Z. You, R. Vasseur, and A. W. Ludwig, Physical Review B **101**, 104302 (2020).

[14] Y. Bao, S. Choi, and E. Altman, Physical Review B **101**, 104301 (2020).

[15] A. Zabalo, M. J. Gullans, J. H. Wilson, S. Gopalakrishnan, D. A. Huse, and J. Pixley, Physical Review B **101**, 060301 (2020).

[16] Q. Tang and W. Zhu, Phys. Rev. Research **2**, 013022 (2020).

[17] Y. Fuji and Y. Ashida, Phys. Rev. B **102**, 054302 (2020).

[18] X. Turkeshi, R. Fazio, and M. Dalmonte, Phys. Rev. B **102**, 014315 (2020).

[19] M. Ippoliti, M. J. Gullans, S. Gopalakrishnan, D. A. Huse, and V. Khemani, Phys. Rev. X **11**, 011030 (2021).

[20] A. Lavasani, Y. Alavirad, and M. Barkeshli, Nature Physics **17**, 342 (2021).

[21] S. Sang and T. H. Hsieh, Phys. Rev. Research **3**, 023200 (2021).

[22] M. Van Regemortel, Z.-P. Cian, A. Seif, H. Dehghani, and M. Hafezi, Phys. Rev. Lett. **126**, 123604 (2021).

[23] M. Buchhold, Y. Minoguchi, A. Altland, and S. Diehl, Phys. Rev. X **11**, 041004 (2021).

[24] Y. Bao, S. Choi, and E. Altman, Annals of Physics **435**, 168618 (2021), special issue on Philip W. Anderson.

[25] S.-K. Jian, C. Liu, X. Chen, B. Swingle, and P. Zhang, Phys. Rev. Lett. **127**, 140601 (2021).

[26] S. Czischek, G. Torlai, S. Ray, R. Islam, and R. G. Melko, Phys. Rev. A **104**, 062405 (2021).

[27] A. C. Potter and R. Vasseur, arXiv preprint arXiv:2111.08018 (2021).

[28] M. Block, Y. Bao, S. Choi, E. Altman, and N. Y. Yao, Phys. Rev. Lett. **128**, 010604 (2022).

[29] T. Minato, K. Sugimoto, T. Kuwahara, and K. Saito, Phys. Rev. Lett. **128**, 010603 (2022).

[30] T. Müller, S. Diehl, and M. Buchhold, Phys. Rev. Lett. **128**, 010605 (2022).

[31] M. Van Regemortel, O. Shtanko, L. P. Garcia-Pintos, A. Deshpande, H. Dehghani, A. V. Gorshkov, and M. Hafezi, arXiv preprint arXiv:2201.12672 (2022).

[32] M. J. Gullans, S. Krastanov, D. A. Huse, L. Jiang, and S. T. Flammia, Phys. Rev. X **11**, 031066 (2021).

[33] R. Fan, S. Vijay, A. Vishwanath, and Y.-Z. You, Physical Review B **103**, 174309 (2021).

[34] Y. Li and M. P. A. Fisher, Phys. Rev. B **103**, 104306 (2021).

[35] B. Yoshida, arXiv preprint arXiv:2109.08691 (2021).

[36] M. J. Gullans and D. A. Huse, Phys. Rev. Lett. **125**, 070606 (2020).

[37] J. Carrasquilla, Advances in Physics: X **5**, 1797528 (2020).

[38] G. Torlai and R. G. Melko, Phys. Rev. Lett. **119**, 030501 (2017).

[39] S. Krastanov and L. Jiang, Scientific reports **7**, 1 (2017).

[40] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. Beenakker, Quantum **2**, 48 (2018).

[41] P. Andreasson, J. Johansson, S. Liljestrand, and M. Granath, Quantum **3**, 183 (2019).

[42] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Quantum **3**, 215 (2019).

[43] Y.-H. Liu and D. Poulin, Phys. Rev. Lett. **122**, 200501 (2019).

[44] E. Flurin, L. S. Martin, S. Hacohen-Gourgy, and I. Siddiqi, Phys. Rev. X **10**, 011006 (2020).

[45] R. Sweke, M. S. Kesselring, E. P. van Nieuwenburg, and J. Eisert, Machine Learning: Science and Technology **2**, 025005 (2020).

[46] B. Schumacher and M. A. Nielsen, Phys. Rev. A **54**, 2629 (1996).

[47] A. Niculescu-Mizil and R. Caruana, in *Proceedings*

of the 22nd International Conference on Machine Learning, ICML '05 (Association for Computing Machinery, New York, NY, USA, 2005) p. 625–632.

[48] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 70, edited by D. Precup and Y. W. Teh (PMLR, 2017) pp. 1321–1330.

[49] Note, for a fixed Clifford circuit, the purification axis does not depend on $\mathcal{M}_T$.

[50] G. E. Hinton and R. R. Salakhutdinov, science **313**, 504 (2006).

[51] Y. LeCun, Y. Bengio, and G. Hinton, nature **521**, 436 (2015).

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT press, 2016).

[53] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, IEEE transactions on neural networks **8**, 98 (1997).

[54] E. Bairey, C. Guo, D. Poletti, N. H. Lindner, and I. Arad, New Journal of Physics **22**, 032001 (2020).

[55] W. Brown and O. Fawzi, 2013 IEEE International Symposium on Information Theory (ISIT) , 346 (2013).

[56] M. B. Hastings and J. Haah, Quantum **5**, 564 (2021).

[57] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar, Nature Phys. **17**, 931 (2021).

[58] J. Haah, R. Kothari, and E. Tang, arXiv:2108.04842 (2021).

[59] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, Computing in Science & Engineering **16**, 62 (2014).

[60] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," (2002).

[61] D. Gottesman, Phys. Rev. A **57**, 127 (1998).

[62] D. Gottesman, arXiv preprint quant-ph/9807006 (1998).

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

METHODS

Quantum Dynamics. The dynamics of hybrid circuits considered in this work in general can be described using the quantum channel formalism. The wave function of the circuit, denoted by $|\psi_S\rangle$ at the beginning of time evolution is entangled to a reference qubit. Formally, the time evolution of the system under this setting can be modeled using Kraus operators [60],

$$K_{\vec{m}} = U_t P_t^{m_t} \cdots U_1 P_1^{m_1} \tag{5}$$

where m_t, U_t and $P_t^{m_t}$, denote the measurement outcomes, unitary gates, and projective measurements at the t-th layer of the circuit, respectively. We also denote the set of all measurement outcomes in different layers via $\vec{m}$. The corresponding evolution of the density matrix, ρ, can be described via the following quantum channel,

$$\mathcal{N}_t(\rho) = \sum_{\vec{m}} K_m \rho K_m^\dagger \otimes |\vec{m}\rangle\langle\vec{m}|. \tag{6}$$

For our purpose, to generate the quantum trajectories we need to consider the time evolution of the system at the level of the wave functions. Under an arbitrary unitary operator U, the wave function evolves as

$$|\psi\rangle \to U|\psi\rangle. \tag{7}$$

For projective measurements, we consider a complete set of orthogonal projectors with eigenvalues labeled by m satisfying $\sum_m P_t^m = 1$ and $P_t^m P_t^{m'} = \delta_{mm'} P_t^m$ under which the wave function evolves as,

$$|\psi\rangle \to \frac{P_t^m|\psi\rangle}{||P_t^m|\psi\rangle||}. \tag{8}$$

In simulating the time evolution of the wave functions, we use random unitaries sampled from the Clifford group where, under any conjugation operation, the Pauli group is mapped to itself [61]. Such circuits, according to the Gottesman-Knill theorem, can be classically simulated in polynomial times in the system size [62, 63].

Implementation of Deep Learning Algorithms. In this work we mainly used convolutional neural networks for learning the decoder function. These network are composed of several interconnected convolutional and pooling layers. The convolutional layer uses the locality of the input data to create new features from a linear combination of adjacent features through a convolution process. These layers are followed by pooling layers which reduce the number of features. Finally, a fully connected layer is used to associate a label to the newly generated features, thus classifying the data. These layers can be repeated a number of times for more complicated input data. Our neural network architecture symbolically displayed in Fig. 1(b) consists of eight layers whose hyperparameters are chosen by an empirical parametric search to optimize the learning accuracy when the number of samples are smaller than 5×10^4. From left to right these layers include: (1) a convolutional layer with a $L_q/2$ filters where L_q is the number of qubits with a kernel size of 4×4, and a stride size of 1×1 with a rectified linear unit (ReLu) activation function, (2) a convolutional layer with a $L_q/2$ filters where L_q is the number of qubits with a kernel size of 3×3, and a stride size of 1×1 with a Relu activation function, (3) a maximum pooling layer with a window size of 2×2 to decrease the dimension of the input data, (4) a dropout layer with a dropping rate of $r_d = 0.2$ to prevent overfitting, (5) a flattening layer to convert the data into a one-dimensional vector, (6) a dense fully connected layer with a Relu activation function whose number of output neurons is variable and is determined according to the number of training samples, $N_n = 512 * (1 + 2\lfloor N_t/2000 \rfloor)$ where $\lfloor x \rfloor$ denotes the floor function of x, (7) a dropout layer with a dropping rate of $r_d = 0.2$, (8) a dense fully connected layer with a sigmoid activation function which generates the prediction for the spin of the reference qubit. Finally, since we have a classification problem, the loss function for comparing the predicted labels and the actual labels is a binary cross entropy function. Using this loss function, for training our neural network model, we use the Adam optimization algorithm with a learning rate $l = 0.001$. The implementation of our neural network layers and their optimization was done by the Python deep-learning packages TensorFlow and Keras.

Key measurements in clifford circuits

Consider a hybrid Clifford circuit $\mathcal{C}$ which has M Pauli measurements. Imagine applying this circuit on an initial stabilizer state which is entangled to a reference qubit. Assume that as a result of this, the reference qubit disentangles and purifies into the $|P; p_R\rangle$ state, where P is one of the Paulis and $p_R = \pm 1$ determines which eigenvector of P the reference qubit has been purified into. Let $s_1, \cdots, s_M = \pm 1$ denote the measurement outcomes for a single run of the circuit. If we run the same circuit again, the ancilla will purify in the same basis P, but we may get different p_R as well as different s_i. The goal is to understand the relation between the value of p_R and the measurement outcomes $\{s_i\}_{i=1}^{M}$.

When a Pauli string is measured on a stabilizer state, the result is either predetermined (in case the Pauli string is already a member of the stabilizer group up to a phase) or it is ± 1 with equal probability. We call the former determined measurements and the latter undetermined measurements. Note that in a stabilizer circuit, whether a measurement is determined or undetermined is independent of previous measurement *outcomes*. Therefore, for a given circuit $\mathcal{C}$ and a fixed ordering of performing measurements, it is well defined to label measurements as either determined or undetermined without referring to a specific circuit run.

The following is a straightforward result of the Gottesmann-Knill theorem:

Corollary 1. *There exists a unique subset of undetermined measurement results $\{s_{j_1}, \cdots, s_{j_m}\}$ (which we call key measurements) such that,*

$$p_R \times s_{j_1} \times s_{j_2} \times \cdots \times s_{j_m} = c \tag{9}$$

where $c = \pm 1$ is the same for all circuit runs. We call this set the key measurements set.

Note that since key measurements are undetermined measurements, their value are independent of each other. Hence, to predict p_R from undetermined measurement outcomes with any accuracy better than 1/2, one needs to have access to all key measurement results.

Each determined measurement can be seen as a constraint between previous undetermined measurement outcomes. Specifically, if s_i is a determined measurement result for some i it means that there is some fixed $c' = \pm 1$ (independent of circuit run) and a subset of undetermined measurements $\{s_{j'_1}, \cdots, s_{j'_m}\}$ such that

$$s_i \times s_{j'_1} \times \cdots \times s_{j'_m} = c' \tag{10}$$

The similarity to the Corollary 1 is not accidental: if the reference qubit is purified in the P Pauli basis, it means that measuring it in the P basis would be a determined measurement.

Existence of these constraints then means that if we relax the condition of the measurements being undetermined in Corollary 1, then the set of key measurements is no longer unique; we may be able to replace some measurement outcomes in Eq.(9) with a product of others using the constraints between measurement outcomes.

Relation between postselected and postselection-free learning schemes

Here, under certain conditions, we argue that the results of the two learning schemes as displayed in Fig. 2 are related to each other. In particular, using the purification-time distribution of the circuits in Fig. 2(a), learnability $R_l(N_t)$, is related to the purification ratio $r_p(t_p)$. In what follows to make our analysis more intelligible, we assume that the learning error is nearly vanishing, $\epsilon_l \simeq 0$. Next, we need to study the averaged learning efficiency of our decoder which for a given t_p and N_t we denote by $\eta_l(t_p, N_t)$. For a given t_p and N_t, this quantity is related to the averaged inaccuracy introduced in the text by $\eta_l = 1 - \delta_l$. To proceed, we employ a simplifying assumption which is approximately consistent with our numerical results. More concretely, we imagine a decoder with a sharp step-like behavior for $\eta_l(t_p, N_t)$ as a function of N_t. Using the Heaviside theta function $\theta_H(x)$, we suppose $\eta_l(t_p, N_t) = \theta_H(N_t - M(t_p))$ where $M(t_p)$ is the minimum number of training samples to reach full efficiency for $t \leq t_p$. From the definitions, if follows straightforwardly that

$$R_l(N_t) = \sum_{t_p=1}^{t_p^{\text{Max}}(N_t)} r_p(t_p), \tag{11}$$

where $t_p^{\text{Max}}(N_t)$ is the maximum purification time that can be learned for a given N_t. However, this quantity can be evaluated by inverting the function $M(t)$ according to $t_p^{\text{Max}}(N_t) = M^{-1}(N_t)$ where $M^{-1}(N_t)$ is the inverse function of $M(t_p)$. Now, we notice that $M(t_p)$ after averaging over different circuits, can be read from the averaged minimum number of training samples in Fig. 2(b). Therefore, by integrating the information in

Fig. 2(a) and Fig. 2(b) plus $\eta_l(t_p, N_t)$, one can explain the behavior of $R_l(N_t)$ in Fig. 2(d). Here, although we do not have the explicit form of $\eta_l(t_p, N_t)$, we use the step-like behavior as an approximation which is justifiable due to the exponential behaviors of the complexity as a function of the purification time. Thus, using Eq.11 as a plausible approximation for the learnability of our decoder, we expect that during the initial fast growth of the curves in Fig. 2(a), learned circuits mostly belong to the circuits with short purification times. However, since for longer purification times an exponentially large number of training samples is required, the initial exponential growth is followed by a slow learning curve. Therefore, in Fig. 2(d), we observe that deep in the pure phase where the majority of circuits have a short purification time, R_l asymptotically approaches one.

Complexity results for scrambled initial states

Here, we present our results for the circuits scrambled by a high-depth random Clifford circuit. Concretely, to obtain such states, we first run our circuits with initial product states only with two-qubit random Clifford gates in the absence of any measurements. This unitary time evolution creates a highly-entangled state after $T \sim L$ time steps with an entanglement entropy proportional to the system size. Next, we entangle the reference qubit to one of the circuit's qubits and run the same circuit in the presence of two-qubit gates and random measurements. As shown in Ref. [11], there is a purification phase transition such that for $p < p_c$ the subsystem entanglement entropy of the circuit after $T \sim L$ still has a volume-law behavior while for $p > p_c$, its entanglement entropy is negligible. Using such initially mixed states, the complexity results are displayed in Fig. 5. Here, as in Fig. 4, we observe a nearly exponential behavior with the purification time. Furthermore, we notice that the postselection learning scheme is more difficult in the pure phase compared to the mixed phase. By comparing the inset and main plots, we also observe that learning with the light-cone data requires less training samples. Finally, by comparing Fig. 2(b) and Fig. 5 we observe that learning the circuits with scrambled initial conditions requires more training samples than the circuits with product state initial conditions.

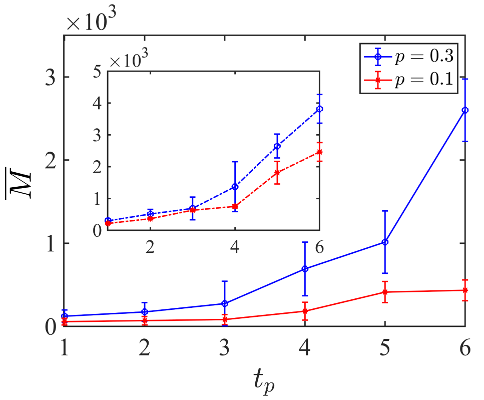

Figure 5. Averaged number of training samples required for learning the reference qubit after postselection as a function of the purification time t_p, for $p = 0.1$ (mixed phase) and $p = 0.3$ (pure phase) with a scrambled initial state. Averaging is performed over $N_c = 20$ circuits for each t_p and error bars are set according to the standard deviation. We have circuits with $L = 128$ qubits. In the main plot measurement outcomes from inside the fixed light-cone box are used for training while for the inset we use the measurement outcomes from the whole circuit.