Reinforced Disentanglers on Random Unitary Circuits

Ning Bao,[1,2,*] Keiichiro Furuya,[1,†] and Gün Süer[1,‡]

[1] *Department of Physics, Northeastern University, Boston, MA, 02115, USA*
[2] *Computational Science Initiative, Brookhaven National Laboratory, Upton, NY, 11973, USA*

(Dated: November 14, 2024)

We search for efficient disentanglers on random Clifford circuits of two-qubit gates arranged in a brick-wall pattern using the proximal policy optimization (PPO) algorithm [1]. Disentanglers are a set of projective measurements inserted between consecutive entangling layers. An efficient disentangler is a set of projective measurements that minimize the averaged von Neumann entropy of the final state with the least number of total projections possible. The problem is naturally amenable to reinforcement learning techniques by taking the binary matrix representing the projective measurements along the circuit as our state and actions as bit-flipping operations on this binary matrix that add or delete measurements at specified locations. We give rewards to our agent dependent on the averaged von Neumann entropy of the final state and the configuration of measurements, such that the agent learns the optimal policy that will take him from the initial state of no measurements to the optimal measurement state that minimizes the von Neumann entropy. Our results indicate that the number of measurements required to disentangle a random quantum circuit is drastically less than the numerical results of measurement-induced phase transition papers. Additionally, the reinforcement learning procedure enables us to characterize the pattern of optimal disentanglers, which is not possible in the works of measurement-induced phase transitions.

CONTENTS

I. INTRODUCTION

Measurement-induced phase transition (MIPT) [2–6] is a relatively new class of phase transition observed in random quantum circuits when measurements are incorporated alongside unitary operations. In these circuits, random unitary operations generate entanglement across the system, while measurements tend to collapse quantum states, reducing the overall entanglement. The competition between these processes—entanglement generation and destruction—results in a critical transition. Specifically, as the measurement rate increases, the system undergoes a phase transition from a volume-law phase, where entanglement entropy or von Neumann entropy of a subsystem scales with system size, to an area-law phase, where entanglement entropy scales with the boundary of a subsystem.

The volume-law phase represents a highly entangled quantum state, while in the area-law phase, frequent measurements collapse the quantum state, leading to low entanglement. The transition is non-thermal, emerging from the nature of quantum dynamics rather than thermal fluctuations, making it an example of a non-equilibrium phase transition. This critical point depends on the rate of measurements, and numerical techniques such as tensor networks and Monte Carlo simulations are often used to explore the phase diagram. MIPTs have been explored in various contexts, including random Clifford circuits, stabilizer circuits, and hybrid quantum systems.

These transitions are significant for understanding the limitations of quantum error correction and how information scrambling behaves in quantum systems [7]. Li et al. (2018) [8] discussed how the quantum Zeno effect plays a role in this transition from volume to area law in random circuits. Skinner et al. (2019) [3] presented the numerical evidence for the existence of these transitions in entanglement dynamics. Chan et al. (2019) [9] extended the understanding of the entanglement dynamics in unitary-projective hybrid circuits. Together, these works lay the theoretical and numerical foundation for studying MIPTs.

Even though great insight has been gained regarding entanglement properties of one-dimensional

* ningbao75@gmail.com
† k.furuya@northeastern.edu
‡ suer.g@northeastern.edu

qubit systems through numerical simulations studying measurement-induced phase transitions, the probabilistic nature of the projections blur the essential properties of the disentanglers. To address that issue, we consider a bottom-up approach for constructing efficient disentanglers in random quantum circuits. In the MIPT literature, brick-wall circuits are composed of consecutive layers of projections and random two-qubit unitaries. In that sense, constructing efficient disentanglers can be framed as a game, in which the disentangler plays against the brick-wall structure, learning how to pick the positions of projections aiming to minimize the reward function consisting of averaged von Neumann entropy of the final state and a configuration of measurements on the circuit.

To solve the disentangling problem, we can take inspiration from reinforcement learning (RL) frameworks [10]. In RL, an agent interacts with an environment and learns a policy to maximize a cumulative reward. For our case, the agent (the disentangler) would act within the quantum circuit (the environment), learning optimal strategies for placing projections to minimize the von Neumann entropy. This setup naturally fits the RL paradigm, where actions correspond to the placement of projections, and the reward could be inversely related to the final entropy of the state.

In particular, policy-based methods such as Proximal Policy Optimization (PPO) have shown strong performance in complex, high-dimensional spaces, making them a suitable candidate for this task [1]. PPO operates by iteratively improving the policy based on the expected return while ensuring updates remain within a trust region to avoid overly large policy shifts, which can destabilize learning. The disentangler's task can be framed as learning a policy $\pi_\theta(a|s)$, where a are the actions (placement of projections) and s is the current state of the system, aiming to minimize the entropy after several layers of the quantum circuit.

PPO's key advantage is its balance between exploration and exploitation, allowing the disentangler to efficiently search for better placements without getting stuck in sub-optimal policies. Moreover, PPO's clipped objective function ensures stable learning by penalizing drastic changes in the policy, which is particularly important in stochastic environments like random quantum circuits. Over time, the disentangler would learn a robust policy that can generalize across different configurations of the brick-wall circuit, leading to low-entropy final states with high efficiency.

We organize this letter as follows. In section II, we review stabilizer states and random Clifford structures. Most importantly, the entanglement entropy can be calculated much more efficiently compared to direct diagonalization methods. We also introduce key reinforcement learning concepts and provide the essential features of the PPO algorithm while explaining the reward profile that we used in training efficient disentanglers. In section III, we present the result of our numerical simulations. Our results enable the understanding of the pattern for efficient disentanglers, which is currently unattainable with the current random projection models studied in the MIPT literature. We conclude the letter with section IV, where we discuss modifications to the reinforcement learning algorithms and reward functions and comment on possible applications.

II. METHODS

We study the dynamics of a 1/2-spin chain with periodic boundary conditions. The time evolution is discrete in the sense that each forward time step corresponds to unitary evolution by a layer of random two-qubit Clifford unitaries. In between each unitary layer, we insert single spin projections along the z-component of spin S_z. Complementary to the standard MIPT literatures, these measurements are not probabilistic, rather our goal is to learn to optimal set of measurements such that the final state has no entanglement.

A. Random Clifford circuits

The main tool for characterizing the dynamics is the entanglement entropy. For a pure state $|\psi\rangle$, and a bipartition $(A, \bar{A})$, the n-th Renyi entropy is given by

$$S_n(A) = \frac{1}{1-n} \log \mathrm{Tr}(\rho_A)^n, \tag{1}$$

where the reduced density matrix for the subsystem A is given by $\rho_A = \mathrm{Tr}_{\bar{A}} |\psi\rangle \langle\psi|$. For general circuits, the computational complexity of the entanglement entropy calculation scales exponentially with the circuit size. However, as Gottesmann and Knill [11] showed, Clifford circuits (or stabilizer circuits) can still be simulated efficiently even when the entanglement grows exponentially, by encoding the quantum state in terms of stabilizers.

Stabilizer states form the foundations of error correcting codes. For an n-qubit system with state $|\psi\rangle$, its stabilizer group is defined by a complete set of tensored Pauli operators $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$ such that

$$g_i |\psi\rangle = + |\psi\rangle . \tag{2}$$

The trivial example is the state with all spin-up qubits satisfying

$$Z_i |\psi\rangle = |\psi\rangle . \tag{3}$$

Under the action of a unitary U, the state evolves as $|\psi\rangle \to U|\psi\rangle$, while the stabilizer group evolves in the Heisenberg representation

$$\mathcal{G} \to \mathcal{G}^U = \left\{ U g_1 U^\dagger, \ldots, U g_{|\mathcal{G}|} U^\dagger \right\} . \tag{4}$$

To keep the stabilizer group properties invariant under time evolution, the unitary U must be an element of the

Clifford group, such that $g_i^U = U g_i U^\dagger$ is still a tensor product of Pauli operators. Therefore, to simulate a stabilizer state under Clifford circuits, one only needs to keep track of the Pauli strings in the stabilizer group $\mathcal{G}$, which only takes polynomial time in the number of qubits.

When $|\psi\rangle$ is a code-word, the Renyi entropies are independent of the index n, and it's completely determined by the size of the stabilizer group [2, 12, 13]

$$S(A) = |A| - \log_2(|\mathcal{G}_A|), \tag{5}$$

where $\mathcal{G}_A$ is a subgroup of $\mathcal{G}$ that only acts with identity I on qubits that are in $\bar{A}$.

B. Reinforcement learning

Reinforcement Learning (RL) is a framework in which an agent interacts with an environment, learning a policy $\pi(a|s)$ that maximizes cumulative reward through trial and error. Formulated as a Markov Decision Process (MDP) [14], the problem is defined by a tuple (S, A, P, R, γ), where S is the set of states, A the set of actions, $P(s'|s,a)$ the transition probability between states, $R(s,a)$ the reward function, and $\gamma \in [0,1]$ the discount factor that balances immediate and future rewards. The agent aims to optimize the expected cumulative reward, expressed as the return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$, where R_t is the reward at time t. RL methods seek to find an optimal policy π^* that maximizes the value function $V^\pi(s) = \mathbb{E}[G_t|s]$, which represents the expected return from state s, or its action-value counterpart $Q^\pi(s,a) = \mathbb{E}[G_t|s,a]$.

There are two main categories of RL methods: model-free and model-based. In model-free methods like Q-learning [15], the agent learns directly from the environment without needing a model of state transitions, updating its Q-function iteratively using the Bellman equation

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R + \gamma \max_{a'} Q(s',a') - Q(s,a) \right). \tag{6}$$

On the other hand, model-based methods involve learning a model of the environment, often through supervised learning, and using it for planning future actions. Both methods must balance exploration (selecting unfamiliar actions to gather more information) and exploitation (choosing actions that maximize reward based on the current knowledge). Recent advances in deep reinforcement learning (deep RL) combine RL with neural networks, allowing for scalable learning in high-dimensional spaces [16].

1. *Proximal policy optimization*

Proximal Policy Optimization (PPO) [1] is a reinforcement learning algorithm designed to improve the stability and performance of policy gradient methods while simplifying the computational complexity seen in approaches like Trust Region Policy Optimization (TRPO) [17]. PPO achieves this by limiting the extent of policy updates to prevent overly large updates that could degrade performance. The key idea is to optimize a surrogate objective function that includes a clipped probability ratio, $\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$, which ensures that the policy update remains within a specified range, thus avoiding excessively large changes. The objective function for PPO is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{7}$$

where $r_t(\theta)$ is the probability ratio, $\hat{A}_t$ is the estimated advantage function, and ϵ is a small hyperparameter controlling the threshold for clipping. This clipping mechanism stabilizes training by preventing large, destructive updates, resulting in improving the performance across a wide range of tasks in continuous and discrete action spaces [1]. PPO has become a popular method in deep reinforcement learning due to its simplicity, ease of implementation, and competitive performance in tasks such as those in the OpenAI Gym [18].

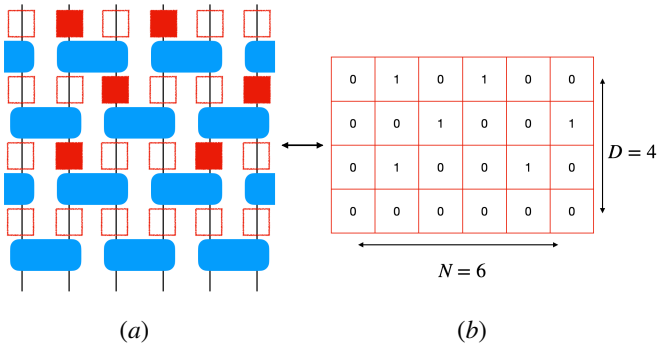

FIG. 1. (a) Depiction of the random quantum circuits with the brick-wall structure. Random two-qubit Clifford gates are given in blue, and the projections are given in red (b) The binary matrix that corresponds to the positions of projection operators.

2. *Rules of the disentangling game*

An environment is prepared by drawing a random Clifford circuit with circuit size $N \times D$. We define the state space and the action space as the set of all possible $N \times D/2$ binary matrices P by translating the length N disentangling circuit layers as length N bit-strings, see Figure 1. Thus, the binary matrix P is constructed by

$$P_{ij} = \begin{cases} 1 & \text{if a measurement is on } (i,j) \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

P starts as a zero matrix at the beginning. At each time step, a single measurement is placed on the circuit

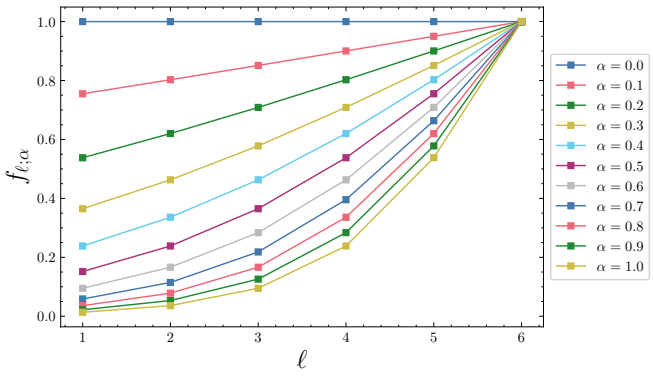

FIG. 2. Measurement weights $f_{l;\alpha}$ as a function of layers l for increasing values of penalty slope α.

corresponding to a single-bit flip in P. A single episode is completed when the von Neumann entropy S_{avg} averaged over all the bipartions of a N-qubit pure state consisting of a set of subsystems $\{A_1, \cdots, A_N\}$ vanishes, i.e.,

$$S_{avg} = \frac{1}{N-1} \sum_{i=1}^{N-1} S(A_1, \cdots, A_i) = 0. \tag{9}$$

This guarantees that the final state is a product state.

3. *Rewards and penalties*

We design a sparse reward function. The sparse reward function is evaluated at the end of every episode.

Placing the measurement on the last layer at least $N/2$ is the trivial solution. To avoid such trivial solutions, both reward functions consider the number of measurements needed to obtain non-trivial efficient disentanglers. Hence, we define a measurement cost as the weighted sum of measurements per layer as

$$C = \sum_{l=1}^{D/2} f_l m_l \tag{10}$$

where $0 \leq m_l \leq N$ is the number of measurements at the l-th layer. The weights $0 \leq f_l \leq 1$ is a monotonically increasing penalty rate as a layer gets deeper, i.e.,

$$f_l > f_l' \text{ for } l > l' \tag{11}$$

where $l, l' \in D/2$ are the labels of l-th and l'-th layers. We adopt the following weight function,

$$f_{l;\alpha} = \frac{2e^{-\alpha l}}{1 + e^{-\alpha l}} \tag{12}$$

where a penalty slope $\alpha \in \mathbb{R}_+$ controls the slope, see figure 2.

Then, the sparse reward function is defined using (10) as

$$R = 1 - \frac{C}{FN} \tag{13}$$

where $F = \sum_l^{D/2} f_{l;\alpha}$, and $0 \leq R \leq 1$.

III. RESULTS

In this section, we present the results of numerical simulations of reinforced disentanglers. We study the performance of the PPO agent as a function of N, D, and α. The quality and behavior of the learning process are discussed in Appendix A alongside the plots for the reinforcement learning metrics as a function of time.

We define the averaged number of measurements minimized with respect to a reward function R over the final configurations P_f of measurements as

$$\langle M \rangle_{eps} := \min_{P_f}|_R \langle \tilde{M} \rangle_{eps}\,. \tag{14}$$

where $\tilde{M}$ is the number of measurements of the non-optimal configuration. For the analysis below, we further simplify it as

$$\mathbb{M} := \langle M \rangle_{eps}\,. \tag{15}$$

In the following subsections, we denote any multilinear function $\eta(x_1, x_2, \cdots, x_n)$ on variables $x_1, \cdots, x_n$ for some n as

$$\eta^{x_i, x_j}(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_{j-1}, x_{j+1}, \cdots, x_n) \tag{16}$$

when evaluating η for a fixed x_i and x_j, for instance.

A. Sparse rewards

When the sparse reward function (13) is adopted, the number of disentanglers averaged over episodes $\mathbb{M}(N, D, \alpha)$ is a function of the number of qubits N, the circuit depth D, and the penalty slope α in (12).

For a fixed depth D and a penalty slope α, $\mathbb{M}^{D,\alpha}(N)$ depends linearly on the number of qubits N as in figure 3, i.e., fitted as

$$\mathbb{M}^{D,\alpha}(N) = \gamma_1^{N,\alpha}(D)N + \gamma_2^{N,\alpha}(D) \tag{17}$$

where $\gamma_1^N(D, \alpha)$ and $\gamma_2^N(D, \alpha)$ are the fitting parameters as the functions of D and α for a given N. Physically, one needs one more measurement on average when two qubits are added[19].

For a fixed N and α, $\mathbb{M}^{N,\alpha}(D)$ behaves as a hyperbolic tangent function on the circuit depth D, figure 3, i.e., fitted as

$$\mathbb{M}^{N,\alpha}(D) = \gamma_1^D(N, \alpha) \tanh(\gamma_2^D(N, \alpha)D) + \gamma_3^D(N, \alpha) \tag{18}$$

Its monotonically increasing behavior, i.e.,

$$\mathbb{M}^{N,\alpha}(D) \leq \mathbb{M}^{N,\alpha}(D') \tag{19}$$

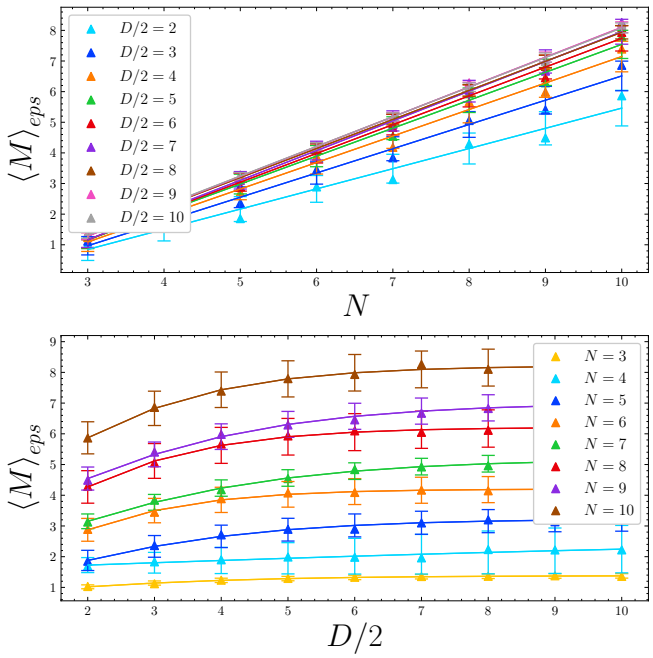

FIG. 3. Rewards and number of projections averaged over episodes vs the number of qubits the trained PPO models, with $\alpha = 0.1$, $t_s = 250,000$, $\ell_r = 0.1$, $e_c = 0.01$, and positive sparse reward $p_r = 50.0$. The best fit of the form (Top) $y = \gamma_1 \tanh(\gamma_2 x) + \gamma_3$ and (Bottom) $y = \gamma_1 x + \gamma_2$ are displayed with error bars. (Bottom) Fit parameters can be found in table I.

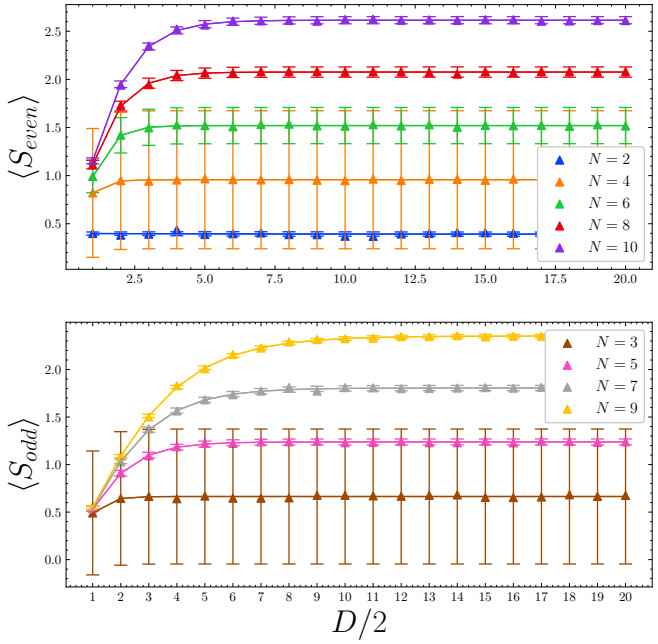

FIG. 4. Entanglement growth as a function of depth for brick-wall random Clifford circuits with no projections, with the increasing number of qubits.

for $D < D'$, can be argued based on the numerical fact that

$$\mathbb{S}^{N,\alpha}(D) \leq \mathbb{S}^{N,\alpha}(D') \tag{20}$$

where $\mathbb{S} := \langle S_{avg} \rangle$ is the averaged von Neumann entropy in (9) averaged over 1000 random circuits, as in Figure 4. When the circuit gets deeper, and the averaged von Neumann entropy $\mathbb{S}$ in (9) stays the same or increases, one needs to increase the number of measurements at least one to keep the final state to be a product state. $\mathbb{M}^{N,\alpha}(D)$ saturates to a constant because there need no more additional measurements after the averaged von Neumann entropy $\mathbb{S}^{N,\alpha}(D)$ saturates to a constant[20].

For a fixed number of qubits N and the circuit depth D, $\mathbb{M}^{N,D}(\alpha)$ increases as in figure 5. The measurements on the last layer can contribute to the disentanglement much more than the ones on the earlier layers. However, the larger α is, the lesser the number of measurements placed on the later layers is. This results in the increment of $\mathbb{M}^{N,D}(\alpha)$ as a function of α. In other words, most measurements are on the last layer when $\alpha = 0$. As α increases, the measurements tend to be placed away from the later layers. We visualize the phenomenon by defining the weighted average of the layers,

$$L = \frac{1}{M} \sum_{l}^{D/2} l m_l \tag{21}$$

for a fixed N and D, where each l-th layer is weighted by the number of measurements on its layer. We plot L averaged over 1000 episodes $\langle L \rangle_{eps}$ for $N \times D/2 = 6 \times 6$ as a function of α in figure 5.

IV. DISCUSSIONS

Using proximal policy optimization, we have provided a bottom-up approach to understanding the pattern of disentanglers in brick-wall random Clifford circuits. Complementary to the usual direction in the MIPT literature, we try to learn the optimal set of measurements that disentangle random Clifford circuits rather than probabilistically placing them between entangling layers with a fixed rate. This enables a deeper understanding of entanglement structure in one-dimensional spin chains. In future work, we hope to report on modified reinforcement learning schemes and reward functions, as well as implications for multipartite entanglement classification. We would also investigate how reinforcement learning techniques similar to the ones used in this work can be extended to studying quantum resource theories.

ACKNOWLEDGEMENTS

We would like to thank Fabian Ruehle, James Halverson, and Yuki Kagaya for their valuable discussions. N.B. is funded by the Spin Chain Bootstrap for Quantum Computation project from the DOE Office of Science-

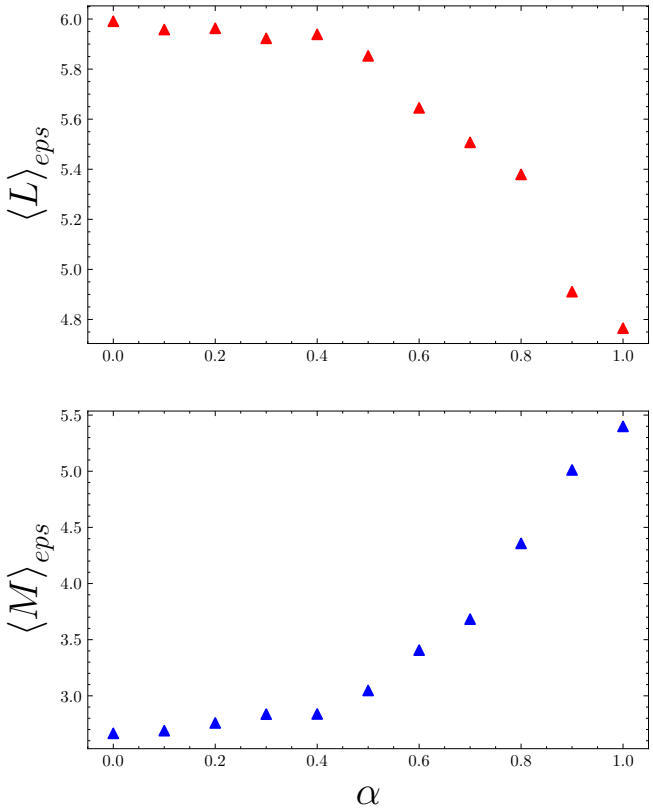

FIG. 5. Layers averaged over measurements, and the total number of measurements averaged over 1000 episodes as a function of the penalty slope α, for circuits of size $N \times D/2 = 6 \times 6$.

Basic Energy Sciences, project number PM602. K. F. and G. S. are supported by N.B.'s startup grant at Northeastern University. This work was completed in part using the Discovery cluster, supported by Northeastern University's Research Computing team.

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms (2017), arXiv:1707.06347 [cs.LG].

[2] A. Nahum, J. Ruhman, S. Vijay, and J. Haah, Phys. Rev. X **7**, 031016 (2017).

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

[4] D. Aharonov, Phys. Rev. A **62**, 062311 (2000).

[5] A. Chan, R. M. Nandkishore, M. Pretko, and G. Smith, Phys. Rev. B **99**, 224307 (2019).

[6] S. Choi, Y. Bao, X.-L. Qi, and E. Altman, Phys. Rev. Lett. **125**, 030505 (2020).

[7] H.-P. Breuer and F. Petruccione, *The theory of open quantum systems* (Oxford University Press, USA, 2002).

[8] Y. Li, X. Chen, and M. P. A. Fisher, Physical Review B **98**, 10.1103/physrevb.98.205136 (2018).

[9] A. Chan, R. M. Nandkishore, M. Pretko, and G. Smith, Physical Review B **99**, 10.1103/physrevb.99.224307 (2019).

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (The MIT Press, 2018).

[11] D. Gottesman, The heisenberg representation of quantum computers (1998), arXiv:quant-ph/9807006 [quant-ph].

[12] A. Hamma, R. Ionicioiu, and P. Zanardi, Physical Review A **71**, 10.1103/physreva.71.022315 (2005).

[13] A. Hamma, R. Ionicioiu, and P. Zanardi, Physics Letters A **337**, 22–28 (2005).

[14] M. L. Puterman, Handbooks in operations research and management science **2**, 331 (1990).

[15] C. Watkins and P. Dayan, Machine Learning **8**, 279 (1992).

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, Nature **518**, 529 (2015).

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 37, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015) pp. 1889–1897.

[18] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, Openai gym (2016), arXiv:1606.01540 [cs.LG].

[19] Odd and even, basically the same. On average, one needs at least 0.5 more measurement when the number of qubits increased by one.

[20] We leave detailed analytical studies of the behavior from the aspect of entanglement membranes and the percolation theory for future work.
[21] H.-Y. Hu, C. Zhao, T. L. Patti, A. Gu, A. M. Gomez, F. Abney-McPeek, Y.-Z. You, and S. F. Yelin, Manuscript in preparation (2024), unpublished manuscript, available upon request mailto:hongyehu@g.harvard.edu.
[22] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Journal of Machine Learning Research **22**, 1 (2021).

Appendix A: Numerical Methods

In this section, we provide details on the numerical methods used to simulate efficient disentanglers. Our methods heavily rely on Clifford circuits and stabilizer states, for that end the sampling of random 2-qubit Clifford gates and von Neumann entropy calculations using stabilizer states were performed using the PyClifford package [21]. As discussed in the main text, we have used proximal policy optimization methods to find the reinforced disentanglers. For that end, the discrete observation and action spaces were constructed using the Open AI Gym package [18], while the proximal policy optimization model was initiated, trained, and saved using Stable-Baselines3 [22].

Depending on the circuit size, we have trained our models using $t_s = 2 \times 10^5, 2.5 \times 10^5, 5 \times 10^5$ maximum time steps, an entropy coefficient of $e_c = 10^{-2}$, and a learning rate of $l_r = 10^{-3}$.

For circuit sizes $N = [3, 4, 5, 6, 7, 8, 9, 10, 11]$, $D/2 = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$ we have fixed $\alpha = 0.1$ and trained for $t_s = 200,000$. Metrics for the learning over time can be found in figure 6.For these simulations, we have resorted to a sparse reward function, such that the agent is rewarded only when it succeeds, and the cost of measurements is subtracted from the reward. At intermediate steps where the entropy is non-zero, the reward is fixed to zero.

Rewards	N	γ_1	γ_2	γ_3	Measurements	N	γ_1	γ_2	γ_3
	3	0.199631	0.234615	0.744301		3	0.69849	0.2617	0.683677
	4	-0.284870	-0.2755	0.642525		4	1.089911	0.072598	1.568935
	5	0.176500	0.160431	0.755519		5	2.608932	0.261157	0.626993
	6	0.255514	0.181856	0.666857		6	3.578105	0.369607	0.624915
	7	0.221658	0.188698	0.693190		7	3.639860	0.236387	1.548429
	8	0.285976	0.189250	0.625591		8	4.741684	0.338947	1.472527
	9	0.260782	0.199553	0.647152		9	4.560935	0.247167	2.456257
	10	0.327017	0.206863	0.575002		10	5.345993	0.316333	2.876648
Rewards	$D/2$	γ_1	γ_2		Measurements	$D/2$	γ_1	γ_2	
	2	-0.01584903	0.86759581			2	0.65911905	-1.12977381	
	3	-0.0142489	0.90259737			3	0.7920119	-1.40795235	
	4	-0.01204827	0.91725471			4	0.8650119	-1.50420238	
	5	-0.01059102	0.92935558			5	0.9120000	-1.57749999	
	6	-0.00951192	0.93983818			6	0.93965476	-1.65563095	
	7	-0.00867545	0.94648441			7	0.97002381	-1.74515476	
	8	-0.00715495	0.9458137			8	0.94880952	-1.55026190	
	9	-0.0069116	0.9516103			9	0.97405952	-1.64026191	
	10	-0.00634318	0.95471422			10	0.96753572	-1.59435715	

TABLE I. Best-fit coefficients for tanh and linear fit for depth and number of qubit dependence of rewards and number of projections averaged over episodes for the trained PPO model.

For fixed circuit sizes $N, D/2 = [5 \times 5, 10 \times 10, 15 \times 15]$ we have also simulated efficient disentanglers for varying penalty slopes $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$. Metrics for the learning over time can be found in figure 7.

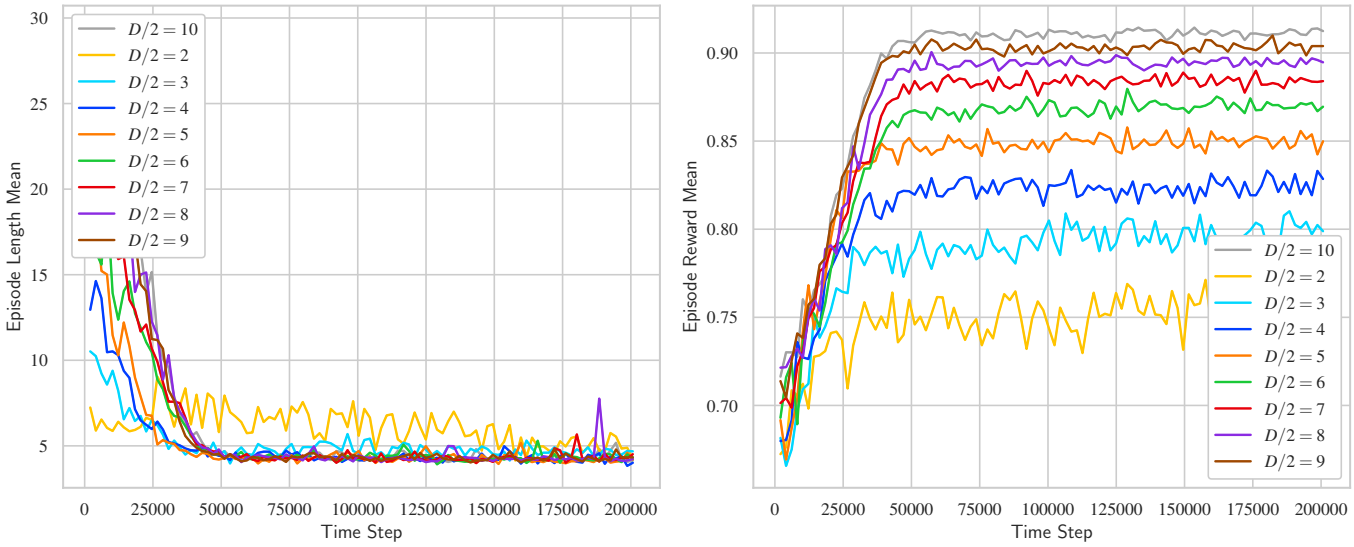

FIG. 6. Mean episode length and mean episode reward as a function of time for increasing depth. $N = 6$ and $\alpha = 0.1$.

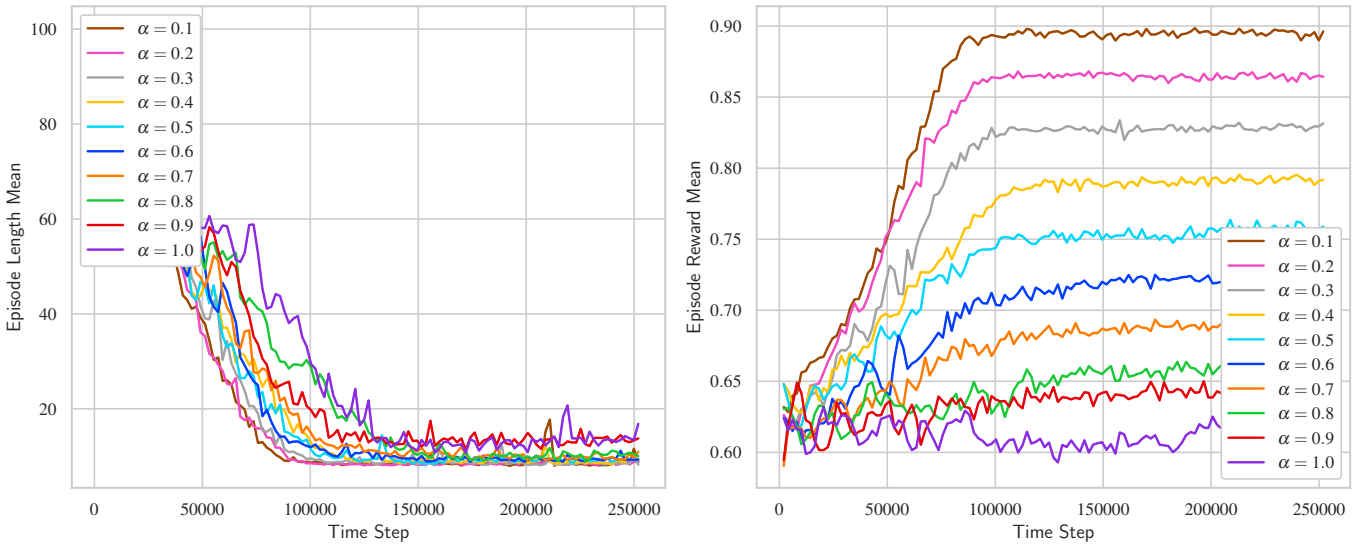

FIG. 7. Mean episode length and mean episode reward as a function of time for increasing values of penalty slope. Circuit size $N \times D = 10 \times 20$.