RL-Synthesised Quantum Circuits: A Novel Lens for Phase Transitions in Many-Body Systems

Anonymous Authors[1]

Abstract

Quantum computing utilises the fundamental properties of quantum mechanics to carry out computations. The quantum circuit complexity of a computation has embedded information about important questions in many-body physics. In this paper, we train a reinforcement learning agent to synthesise quantum circuits that retrieve the time evolution operator of the transverse field Ising Hamiltonian from a simple starting state. We formalise the problem as three Markov Decision Processes and show that the tensor network implementation outperforms other implementations and accurately encodes information about the phase transition boundary of the Hamiltonian by showing a stark decrease in circuit complexity at the transition point.

1. Introduction

Quantum computing exploits the properties and peculiarities of quantum mechanics to undertake some computational tasks faster and more efficiently than classical computers (Shor, 1997). Similar to classical computing, quantum computing algorithms rely on circuits[1] to carry out computation. Handcrafted heuristic methods of circuit discovery are often inefficient and not scalable, paving way for research in automated circuit discovery using machine learning methods (Zen et al., 2025). Wang et al. (2025) have shown that a tensor network representation of quantum computation proves to be more accurate than other representations when training a reinforcement learning model on circuit generation.

The complexity of a quantum circuit is just the number of gates implemented in the circuit. Although a simple counting problem, quantum circuit complexity provides insight on important problems in physics. Susskind (2016) proposes that the circuit complexity of a boundary quantum state is dual to the volume of the interior of the corresponding black-hole[2]. In many-body physics, some phase transitions are invisible to local order parameters (Hastings & Wen, 2005) and require information about global properties: information encoded in quantum circuits.

In this work, we train a reinforcement learning agent on quantum circuit discovery for estimating the time-evolution operator for the transverse field Ising Hamiltonian with a known phase transition. Theoretically, quantum circuit complexity remains low before and after the phase transition[3], with a spike during the phase transition. We approach this problem in a similar fashion to Wang et al. (2025), by training on three different discovery modes, namely direct, reverse, and tensor network. We observe that while direct and inverse methods fail to capture information about the phase transition, the tensor network representation shows a peak at exactly the phase transitions, empirically showing that the unsupervised framework has learnt circuit complexity as a perfect order parameter carrying information about the phase transition.

2. Problem Formulation

2.1. Task : Time Evolution Operator Synthesis

Given n qubits $|q_0, q_1, g_2, ...q_{(n-1)}\rangle = |0, 0, 0, ...0\rangle$ and a universal gate set $G = \{\text{H, T, CNOT}, R_z(\theta)\}$, find a quantum circuit which implements the time evolution operator:

$$U(J, t) = e^{i \cdot H(J) \cdot t}$$

where $H(J)$ is the transverse field Ising Hamiltonian:

$$H(J) = -J \sum_i Z_i Z_{i+1} - \sum_i X_i$$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]Paradigms which stray from the circuit model of quantum computation exist, like Measurement Based Quantum Computation (MBQC), where computation is implemented as a series of measurements on a cluster of prepared qubits.

[2]Therefore, circuit complexity is a metric encoding geometric properties in bulk space-time.

[3]Circuit complexity is proportional to correlations and symmetry. Before the phase transition, the system has very small correlations and no symmetry, and becomes highly symmetric and correlated after.

where Z_i and X_i are the Pauli matrices. For $n = 2$ qubits, $U(J)$ is a unitary matrix, computed by expanding $H(J) = -J(Z_0 \otimes Z_1) - (X_0 \otimes I_1) - (I_0 \otimes X_1)$, giving us:

$$H(J) = \begin{pmatrix} -J & -1 & -1 & 0 \\ -1 & J & 0 & -1 \\ -1 & 0 & J & -1 \\ 0 & -1 & -1 & -J \end{pmatrix}$$

The time evolution operator can be obtained by diagonalising H(J):

$$H(J) = V \cdot diag(\lambda_0, \lambda_1, \lambda_2, \lambda_3)V^\dagger$$

$$U(J,t) = Vdiag(e^{-i\lambda_0 t}, e^{-i\lambda_1 t}, e^{-i\lambda_2 t}, e^{-i\lambda_3 t})V^\dagger$$

Where λ_i is the ith eigenvalue of H(J). Since $H(J)$ is hermitian, it admits a unitary eigendecomposition. For $n = 3$, we arrive at the expression for the Hamiltonian:

$$\begin{aligned} H(J) = &- J(Z \otimes Z \otimes I + I \otimes Z \otimes Z) \\ &- (X \otimes I \otimes I + I \otimes X \otimes I + I \otimes I \otimes X) \end{aligned}$$

Which can be be diagonalised and used to find $U(J)$ in a similar fashion.

2.2. Circuit Synthesis Models

We model the problem as three variants of a Markov Decision Process (MDP), closely following Wang et al. (2025).

2.2.1. Forward Discovery

We define an action space $\mathcal{A} = \{H, T, CNOT, R_z(\theta)\}$, where $H = \{H_i\} = \{H_0, H_1, H_2..., H_{n-1}\}$ is the Hadamard gate applied to the ith qubit, $T = \{T_i\} = \{T_0, T_1, T_2..., T_{n-1}\}$ is the phase shift gate on the ith qubit, $CNOT = \{CNOT_{ij}\}$ is the controlled-NOT gate applied to the jth qubit with the ith qubit as control, and $R_z(\theta) = \{R_z^{(i)}(\theta)\} = \{R_z^{(0)}(\theta), R_z^{(1)}(\theta), R_z^{(2)}(\theta)..., R_z^{(n-1)}(\theta)\}$ is the rotation gate applied to the ith gate with angle θ. Together, this corresponds to a universal gate set[4].

We also define state space $\mathcal{S}$ with the initial state $U_0 = I$, where I is the identity matrix. For an action $a \in \mathcal{A}$, the state S changes as $S' = AS$, therefore, the state space is a tree. The connecting lines from one node to another in $\mathcal{S}$ is an action chosen from $\mathcal{A}$.

For example, in a setup with $n = 2$ qubits, if the target matrix is the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, the optimal trajectory to reach target matrix $S \in \mathcal{S}$ is $\{H, CNOT_{01}\}$, representing the final state $(CNOT_{01}(H \otimes I))\,|00\rangle$.

[4]These actions are enough to reach any target state.

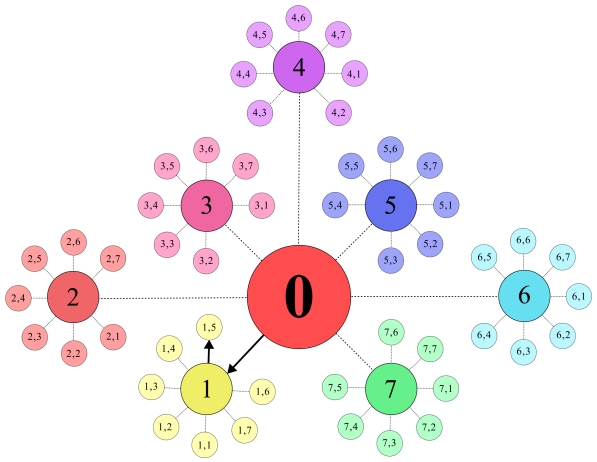

Figure 1. State tree for the Markov Decision Process defined from initial state to Bell State for $n = 2$ qubits.

$$\begin{aligned} S_1 &= (H_0 \otimes I)S_0 \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \\ S_{(1,5)} &= CNOT_{01}S_1 \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \end{aligned}$$

Where $S_{(1,5)} = |\Phi^+\rangle$. In the action set $\mathcal{A}_\in = \{H_0, H_1, T_1, T_2, CNOT_{01}, R_z^{(0)}(\theta), R_z^{(1)}(\theta)\}$, H_0 corresponds to first action in the set, and $CNOT$ corresponds to the fifth. Therefore, (1,5) are the required actions, as shown in Figure 1.

The reward function for the task of finding the Bell state from identity at state S_1 is

$$R(s,a) = \begin{cases} 100 & \text{if } s = S_1 \text{ and } a = \text{CNOT}_{01} \\ 0 & \text{otherwise} \end{cases}$$

2.2.2. Reverse Discovery

The previous formalisation has the disadvantage of requiring retraining for different target states. This problem can be solved training on the inverse problem: given target state U, which inverse gates need to be applied to reach I?

We define an action space $\mathcal{A}^{-1} = \{H^{-1}, T^{-1}, CNOT, R_z(-\theta)\}$, where $H^{-1} = \{H_i^{-1}\} = \{H_0^{-1}, H_1^{-1}, H_2^{-1}..., H_{n-1}^{-1}\}$ is the inverse of the

Table 1. Comparison of the three discovery modes on simple tasks

TARGET	DIRECT	REVERSE	TENSOR NETWORK	ACTION SET
BELL STATE $\|\Phi^+\rangle$	86%	85%	100%	$\{H_0, H_1, T_0, T_1, \text{CNOT}_{01}, \text{CNOT}_{10}\}$
BELL STATE $\|\Phi^-\rangle$	41%	25%	94%	$\{H_0, H_1, T_0, X_0, X_1, \text{CNOT}_{01}\}$
BELL STATE $\|\Psi^+\rangle$	55%	53%	95%	$\{H_0, H_1, T_0, X_0, X_1, \text{CNOT}_{01}\}$
BELL STATE $\|\Psi^-\rangle$	5%	4%	15%	$\{H_0, H_1, T_0, X_0, X_1, Z_0, Z_1, \text{CNOT}_{01}\}$

Hadamard gate applied to the ith qubit, $T^{-1} = \{T_i^{-1}\} = \{T_0^{-1}, T_1^{-1}, T_2^{-1} ..., T_{n-1^{-1}}\}$ is the inverse phase shift gate on the ith qubit, $CNOT = \{CNOT_{ij}\}$ is the controlled-NOT[5] gate applied to the jth qubit with the ith qubit as control, and $R_z(-\theta) = \{R_z^{(i)}(-\theta)\} = \{R_z^{(0)}(-\theta), R_z^{(1)}(-\theta), R_z^{(2)}(-\theta) ..., R_z^{(n-1)}(-\theta)\}$ is the inverse rotation gate applied to the ith gate with angle $-\theta$. Together, this corresponds to a universal gate set[6].

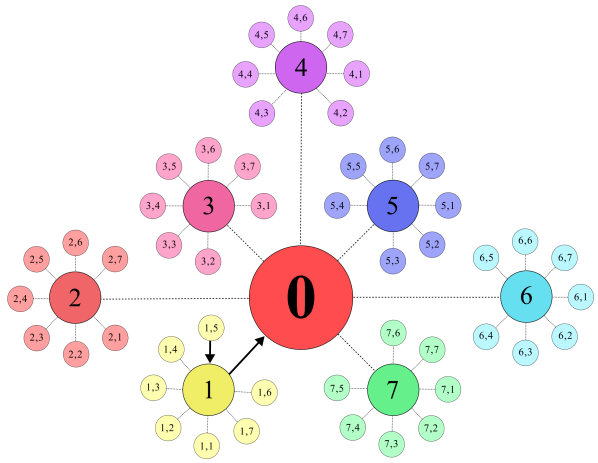

Figure 2. State tree for the Markov Decision Process defined from Bell state to initial matrix for $n = 2$ qubits.

We also define state space $\mathcal{S}^{-1}$ with the initial state $U_0 = I$, where I is the identity matrix. For an action $a^{-1} \in \mathcal{A}^{-1}$, the state S^{-1} changes as $S'^{-1} = A^{-1}S^{-1}$, therefore, the state space is a tree. The connecting lines from one node to another in $\mathcal{S}$ is an action chosen from $\mathcal{A}$.

For example, in a setup with $n = 2$ qubits, if the initial matrix is the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, the optimal trajectory to reach identity matrix I is $\{CNOT_{01}^{-1}, H^{-1}\}$, representing the final state $((H^{-1} \otimes I)CNOT_{01}^{-1})|\Phi^+\rangle$. This avoids retraining for different target matrices, since the task is to reach the identity matrix.

[5]The inverse of CNOT is CNOT itself.

[6]These actions are enough to reach the initial matrix from any target state.

The reward function for the task of finding the identity from the Bell state when the current state is $S_1^{-1} = CNOT_{01}^{-1})|\Phi^+\rangle$ is

$$R(s,a) = \begin{cases} 100 & \text{if } s = S_1^{-1} \text{ and } a = H_0^{-1} \\ 0 & \text{otherwise} \end{cases}$$

2.2.3. TENSOR NETWORK

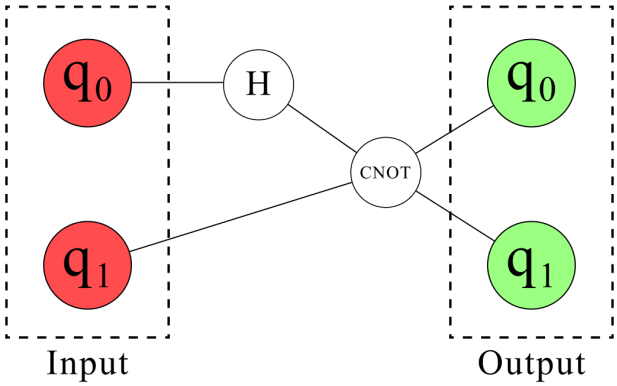

Figure 3. Tensor network representation for a circuit that generates Bell states $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ from $|00\rangle$.

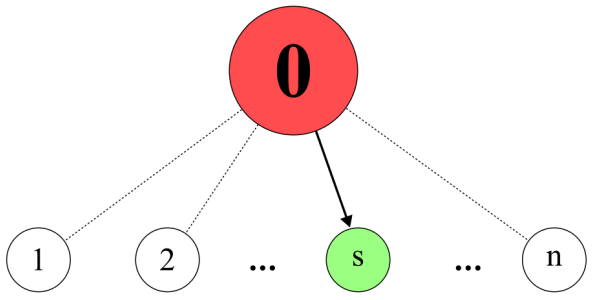

Figure 4. State tree for the tensor network circuit discovery model. There are n elements in the action space, with s being the correct action.

The tensor network representation of quantum circuits is a powerful paradigm, where single qubit gates are tensors of

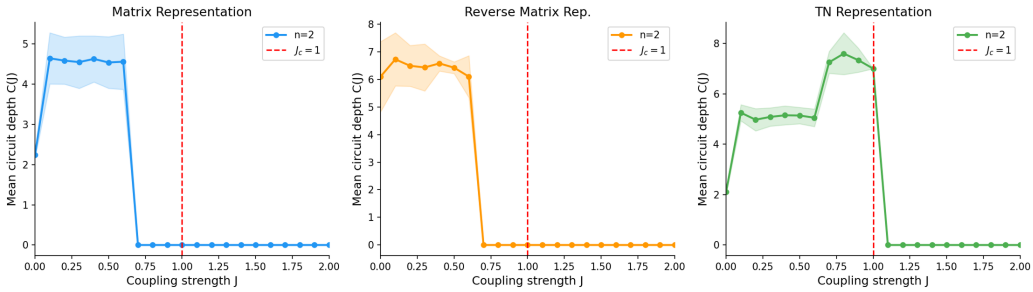

Figure 5. Circuit depth $C(J)$ vs. coupling strength J for matrix representation (Direct), reverse matrix representation (Inverse), and Tensor Network Representation.

order 2, and double qubit gates are tensors of order 4, and a tensor network is a network of interconnected tensors.

For demonstration, we consider a two qubit circuit with two gates, with gates $G = \{H, T\}$. The action space is defined as $\mathcal{A} = \{H_0, H_1, T_0, T_1, (H_0, H_0), (H_0, H_1), (H_0, T_0), (H_0, T_1), (H_1, H_0), (H_1, H_1), (H_1, T_0), (H_1, T_1), (T_0, H_0), (T_0, H_1), (T_0, T_0), (T_1, H_0), (T_1, H_1), (T_1, T_0), (T_1, T_1)\}$. The state updates as $S' = AS$. For the Bell state task, the TN representation and search tree can be found in Figure 3 and Figure 4 respectively.

The reward function for the task of finding the Bell state from identity is

$$R(s,a) = \begin{cases} 100 & \text{if } s = S_0 \text{ and } a = (H_0, \text{CNOT}_{01}) \\ 0 & \text{otherwise} \end{cases}$$

2.3. Reinforcement Learning Models

We use the Q-learning algorithm by Watkins & Dayan (1992) which updates a Q-table according to the expression:

$$Q_{new}(S_t, a_t) \leftarrow (1-\alpha) \cdot Q(S_t, a_t) + \alpha \left(R_{t+1} + \gamma \cdot \max_a Q(S_{t+1}, a) \right)$$

Where α is the learning rate, γ is the discount factor and $Q(S_t, a_t)$ is the Q-table value. We run the experiment over similar conditions as (Wang et al., 2025) over $\alpha = 0.5; \gamma = 0.9$. We train over 500 iterations for 100 rounds, and measure success ratio over 100 rounds. We run the experiment over $J = \{0, 0.1, 0.2, 0.3...1.9, 2\}$. We report the accuracies of these three discovery modes as for the simpler task of finding four Bell States in Table 1.

3. Results

We run the reinforcement learning circuit discovery model over 1,050,000 episodes for $n = 2$ qubits for all three formulations (direct, inverse, and tensor network). We measure the mean circuit depth $C(J)$, which is the number of gates required to retrieve the time evolution operator $U(J, t)$ from $|00\rangle$, where $t = 1$. Theoretically, the phase transition occurs at $J = 1$, where the number of gates required to retrieve J increases, and then decreases for both $J > 1$ and $J < 1$. This relation arises from gapless systems exhibiting a phase change from a "disordered" state to an "ordered" state. Disordered states require few gates because of low correlation length, and ordered states require few gates due to global symmetry. During the phase transitions, correlation length increases, while global symmetry is still low, requiring more gates.

The reinforcement learning model learns a perfect order parameter (circuit depth) which carries information about the phase transition occurring at $J = 1$, without any supervision.

Future Work

In this experiment, only 2-qubit systems are analysed. A natural extension is to increase fidelity and implement this formulation to account for higher amounts of qubits. Also of note is the search space blow-up for this problem, and implementing methods like Monte-Carlo Tree Search (MCTS) might prove fruitful. It is theorised that peak-to-trough ratio in $C(J)$ should increase as a power law of n (Roca-Jerat et al., 2023), motivating the search of experimental evidence of the same through TN based RL for circuit synthesis. Topological phases like the quantum Hall states, topological insulators, etc. have phase transitions invisible to local order parameters, requiring global information accumulation, making similar frameworks well suited to study them.

References

Hastings, M. B. and Wen, X.-G. Quasiadiabatic continuation of quantum states: The stability of topological ground-state degeneracy and emergent gauge invariance. *Phys. Rev. B*, 72:045141, Jul 2005. doi: 10.1103/PhysRevB.72.045141. URL https://link.aps.org/doi/10.1103/PhysRevB.72.045141.

Roca-Jerat, S., Sancho-Lorente, T., Román-Roche, J., and Zueco, D. Circuit complexity through phase transitions: Consequences in quantum state preparation. *SciPost Physics*, 15(5), November 2023. ISSN 2542-4653. doi: 10.21468/scipostphys.15.5.186. URL http://dx.doi.org/10.21468/SciPostPhys.15.5.186.

Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. ISSN 1095-7111. doi: 10.1137/s0097539795293172. URL http://dx.doi.org/10.1137/S0097539795293172.

Susskind, L. Entanglement is not enough. *Fortschritte der Physik*, 64(1):49–71, 2016. doi: https://doi.org/10.1002/prop.201500095. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prop.201500095.

Wang, Z., Feng, C., Poon, C., Huang, L., Zhao, X., Ma, Y., Fu, T., and Liu, X.-Y. Reinforcement Learning for Quantum Circuit Design: Using Matrix Representations. 1 2025.

Watkins, C. J. C. H. and Dayan, P. Q-learning. *Machine Learning*, 8(3):279–292, May 1992. ISSN 1573-0565.

Zen, R., Olle, J., Colmenarez, L., Puviani, M., Müller, M., and Marquardt, F. Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning. *Physical Review X*, 15(4), October 2025. ISSN 2160-3308. doi: 10.1103/gqpr-dgz7. URL http://dx.doi.org/10.1103/gqpr-dgz7.

A. Appendix

A.1. Quantum Circuits and Gates

You can find exposition about how we use the matrix representations of quantum circuits and gates in this section.

A quantum state $|\Psi\rangle \in \{|0\rangle, |1\rangle\}$ can be represented as a matrix.

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}; |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

For multi-qubit systems,

$$|\Psi\rangle = |q_0\rangle \otimes |q_1\rangle \otimes |q_2\rangle ... = \bigotimes |q_i\rangle = |q_0, q_1, q_2 ...\rangle$$

where the computational formula for the tensor product $\otimes$ is defined as:

$$\mathbf{A}_{2\times 2} \otimes \mathbf{B}_{m\times n} = \begin{pmatrix}A_{00} & A_{01}\\ A_{10} & A_{11}\end{pmatrix} \otimes \mathbf{B} = \begin{pmatrix}A_{00}\mathbf{B}_{m\times n} & A_{01}\mathbf{B}_{m\times n}\\ A_{10}\mathbf{B}_{m\times n} & A_{11}\mathbf{B}_{m\times n}\end{pmatrix}$$

Therefore, a state $|\Psi\rangle = |001\rangle$ is:

$$|0\rangle \otimes |0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^\top$$

A gate $\mathbf{G}$ operates on a state $|\Psi\rangle$ by $\mathbf{G}|\Psi\rangle$, where both $\mathbf{G}$ and $|\Psi\rangle$ are in matrix form. The matrix form of the gates used in this paper are given below.

Hadamard Gate (H) : $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$

$\pi/8$ **Gate (T)** : $\begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$

Controlled-NOT Gate[7] **(CNOT)** : $\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$

Rotation Gate ($R_z(\theta)$) : $\begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}$

And the Pauli matrices $\mathbf{X} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ and $\mathbf{Z} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$.

[7] This is a two qubit gate, and only operates on input states of the form $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle$