

Quantum Simulator Base on Matrix Product State

Kai-Hsin Wu*

Department of Physics, Boston University, 590 Commonwealth Avenue, Boston, Massachusetts 02215, USA

The MPS formalism have been widely used to represent the state in the tensor network algorithm for solving such as DMRG, iTEBD etc in the context of simulating the large scale condensed matter system. And has been shown to be a powerful and efficient way to approximately represent a many-body quantum state that scales linearly with system size n . In this final project, we will do a review study of the quantum simulator base on the matrix product state (MPS) approach, and use that to implement and study Grover algorithm.

I. INTRODUCTION

How big can classical computer simulate quantum computer is always a challenging problem. In order to exactly hold the information of a n -qubit state, the naive approach is to store the state with 2^n complex elements, which scales exponentially with system size thus limited the size of problem one can solve.

One way to tackle this problem is to represent the state using the Matrix product state (MPS) where the first idea can be trace back to 2004 [1, 2]. Since then, the MPS approach has been widely used in lots of quantum information studies [3–5] and even begin adapted into well known quantum simulator such as Qiskit [6].

In this final project, we will implement the MPS base on my tensor network library [7]. Firstly, the matrix product state is introduced as a way to represent the state. Some common operations and forms of MPS that is useful for implementing quantum circuit will also be introduced. We will start by implementing one of the important building block—the multi-control gate base on the toffoli gate decomposition. We will show a specific implementation that do not require swap gate to make the operation nearest-neighboring for the MPS. Finally, we will implement the Grover algorithm, and study how the entanglement changed through each gate application, as well as how one can benefit from the MPS representation by performing truncation.

II. MATRIX PRODUCT STATE

Consider a n qubits state

$$\psi = \sum_{i,j,k,l,\dots \in \{0,1\}} C_{ijkl\dots} |ijkl\dots\rangle, , \quad (1)$$

generally one need 2^n variables $C_{ijkl\dots}$ to precisely represent the wave function. However, do we really need that many variables? In some of the states, most of the variables can be made zero by basis transformation, thus it is a waste to store them using 2^n variables as most of them are just zeros. A simple example is the product state. Taking 4-qubit state $\psi = |0\rangle|1\rangle|0\rangle|0\rangle$ as example. Here, only $C_{0100} = 1$ and the rest are just zero.

The matrix product state (MPS) is a clever way to represent the manybody wavefunction, so that one can numerically "adjust" the number of parameters when the underlying state does not need that much variables to represent.

* khwu@bu.edu



FIG. 1.

So how do we know if we don't need that many variables? Recall that in the class, we introduce the schmidt decomposition, where the schmidt values provides how parts of a manybody state "entangled". For the product state, as we learned, only one schmidt values $\lambda = 1$ and the rest are just zeros.

The MPS make use of this property. We decompose the manybody state into it's local part by utilizing the SVD decomposition, and obtain the schmidt values between sites. If the entanglement is not large, it is possible to "throw out" some schmidt values to reduce the variables, but still has a good approximation of the state. To show how things are actually done, let's start with a bipartite system:

$$\psi = \sum_{ab} C_{ab} |a\rangle |b\rangle, \quad (2)$$

where $|a\rangle$ and $|b\rangle$ can be multi-qubit state. Using SVD to do schmidt decomposition, we have

$$C_{ab} = \sum_i U_{ai} \lambda_i V_{ib}^\dagger \quad (3)$$

where λ_i are the schmidt values and $\sum_i \lambda_i^2 = 1$. It is more intuitive to represent this graphically by tensor notation, as shown in Fig. 1 where a n-rank tensor is represented by a body with n -legs. Connected legs indicates the trace of the indices. Further more, one can show that the schmidt value characterize the entanglement between parts A and B by consider the density matrix

$$\rho_a = \text{Tr}_b \rho = \sum_i \lambda_i^2 |\Phi_a\rangle \langle \Phi_a| \quad (4)$$

and thus the entanglement entropy relates to schmidt values as $S = -2 \sum_i \lambda_i^2 \ln \lambda_i$.

If two parts are slightly entangled, then most of the singular values are zeros or small, then it is possible to drop those values but still have a good approximation. This leads to the truncation scheme of the MPS. For example, consider the decomposition in Fig. 1. If a and b have same dimension D , then we need D^2 variables to exactly represent a general state. The MPS decomposition leads to total variables $2D\chi$ (if we absorb λ into either U or V^\dagger .) where χ is the number of singular values we keep, and sometimes referring to "virtual" bond dimension. In an extreme example if a and b are not entangled at all, then only one singular values $\lambda_0 = 1$, the truncation leads to $\chi = 1$ and thus reduce the number of variables from $D^2 \rightarrow 2D$.

One can generalize this decomposition to n qubits state eq.(1) by perform successive SVD decomposition



FIG. 2.

to the coefficient-tensor $C_{ijkl\dots}$

$$C_{ijkl\dots} = \sum_{\alpha} U_{i\alpha} s_{\alpha}^{[1]} V_{\alpha jkl\dots}^{\dagger} \quad (5)$$

$$= \sum_{\alpha, \beta} U_{i\alpha}^{[1]} s_{\alpha}^{[1]} U_{\alpha j\beta}^{[2]} s_{\beta}^{[2]} V_{\beta k l\dots}^{\dagger} \quad (6)$$

$$= \sum_{\alpha\beta\gamma} U_{i\alpha}^{[1]} s_{\alpha}^{[1]} U_{\alpha j\beta}^{[2]} s_{\beta}^{[2]} U_{\beta k\gamma}^{[3]} s_{\gamma}^{[3]} V_{\gamma l\dots}^{\dagger} \quad (7)$$

$$= \text{Tr}\{U^{[1]}(s^{[1]}U^{[2]})(s^{[2]}U^{[3]})(s^{[3]}U^{[4]})\dots(s^{[n]}V^{\dagger})\} \quad (8)$$

$$= \text{Tr}(M^{[1]}M^{[2]}\dots M^{[n]}) \quad (9)$$

where each M is a 3-rank tensor (3-leg) as shown in Fig. 2. Note that the singular value matrices $s^{[i]}$ we represent here does not equal to the schmidt values of their corresponding bipartition. In order to get the correct schmidt values, one has to put the MPS into co called "canonical forms", as we will show in the following section.

A. Canonical forms

One of the important thing in the MPS representation is that there is an degree of freedom in the choice of M tensors, which is sometimes called a 'gauge' degree of freedom. One can perform arbitrary unitary transformation on all M with unitary matrix U_g

$$M^{[i]} \rightarrow U M^{[i]} U^{\dagger} \quad (10)$$

such that the MPS is invariant. There are some choice of gauge that bring the M tensors into it's "canonical form", and will be very helpful for evaluating measurement, and performing truncation.

Starting with eq. (8), one can shift all the singular values to a single site, by successively perform SVD such that

$$C_{ijkl\dots} = U^{[1]}(s^{[1]}U^{[2]}s^{[2]})U^{[3]}s^{[3]}U^{[4]}\dots s^{[n]}V^{\dagger} \quad (11)$$

$$= U^{[1]}(\tilde{U}^{[2]}\tilde{s}^{[1]}\tilde{V}^{[2]})U^{[3]}s^{[3]}U^{[4]}\dots s^{[n]}V^{\dagger} \quad (12)$$

$$= U^{[1]}\tilde{U}^{[2]}(\tilde{s}^{[1]}\tilde{V}^{[2]}U^{[3]}s^{[3]})U^{[4]}\dots s^{[n]}V^{\dagger} \quad (13)$$

$$= U^{[1]}\tilde{U}^{[2]}(\tilde{U}^{[3]}\tilde{s}^{[2]}\tilde{V}^{[3]})U^{[4]}\dots s^{[n]}V^{\dagger} \quad (14)$$

$$= U^{[1]}\tilde{U}^{[2]}\tilde{U}^{[3]}\tilde{U}^{[4]}\dots\tilde{U}^{[n]} \quad (15)$$

In this form, all the M tensors are left unitary. The MPS in this form is called left-canonical. One can also perform the same procedure, but starting from the right, then one arrives

$$C_{ijkl\dots} = \tilde{V}^{\dagger[1]} \tilde{V}^{\dagger[2]} \dots \tilde{V}^{\dagger[n]} \quad (16)$$

which is called right-canonical form.

Another important form is the "mixed"-canonical form. Starting from left-canonical form, one can perform the SVD procedure from the right, and "rotate" the singular values to any site, i.e.

$$C_{ijkl\dots} = \tilde{U}^{[1]} \tilde{U}^{[2]} \dots U^{[m]} s_m V^{\dagger[m+1]} \dots V^{\dagger[n-1]} V^{\dagger[n]} \quad (17)$$

This form is called "mixed"-canonical form, and plays an important role when performing truncation because the singular matrix s_m (which sometimes also called "orthogonal center") is the actual schmidt values between bipartition $A = 1 \sim m$ and $B = m + 1 \sim n$. Thus s_m will be used to determine the truncation of virtual bonds between site m and $m + 1$, as well as calculating the entanglement entropy.

B. Applying local quantum gate and truncation

Now that we have introduced MPS as a representation of n -qubits state, it's time to consider applying gates on it. As we learned from the class, all the n -qubit quantum gate can be decomposed into universal gate sets, which is local.

Consider an initial state to be MPS with virtual bond dimension χ , and we are going to apply a two-site unitary gate on two of the intermediate, nearest-neighboring sites m and $m + 1$. In Fig. 3(a), we show the circuit diagram and Fig. 3(b) is the corresponding operation in tensor notation.

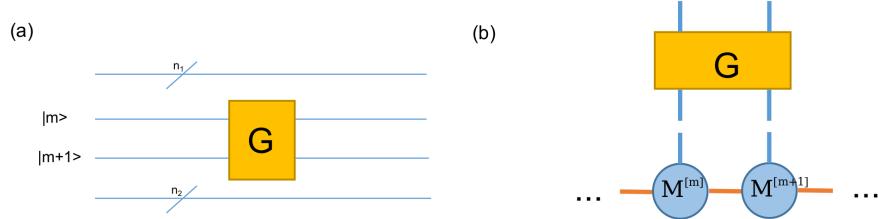


FIG. 3.

The procedure is as following. First, we put the MPS into mixed canonical form by rotating orthogonal center in between site m and $m + 1$. We then contract local tensors $M^{[m]}$, s and $M^{[m+1]}$ forming the local wave function ψ . The gate is then apply onto ψ , giving a new local wave function $\tilde{\psi}$. Finally, to make it back to the MPS form, we next perform SVD on $\tilde{\psi}$.

Notice that now the intermediate virtual bond dimension is now growth to 2χ . At this point, if we want our MPS bond dimension staying in χ , then one need to perform truncation by dropping the smallest χ singular values, and renormalize λ . This truncation generally leads to approximation unless the singular values we dropped are zero. The whole procedure are shown in Fig. 4. In practice, on top of setting a maximum bond dimension, we usually set also a cutoff to singular values (for example 1e-12) so that when the underlying state does not require so large bond dimension, it can automatically reduce.

It is also worth noting that the gate operation in this way only involves local MPS sites m and $m + 1$. The other $n - 2$ MPS sites are not involved in the operation, which makes it very efficient. However, there

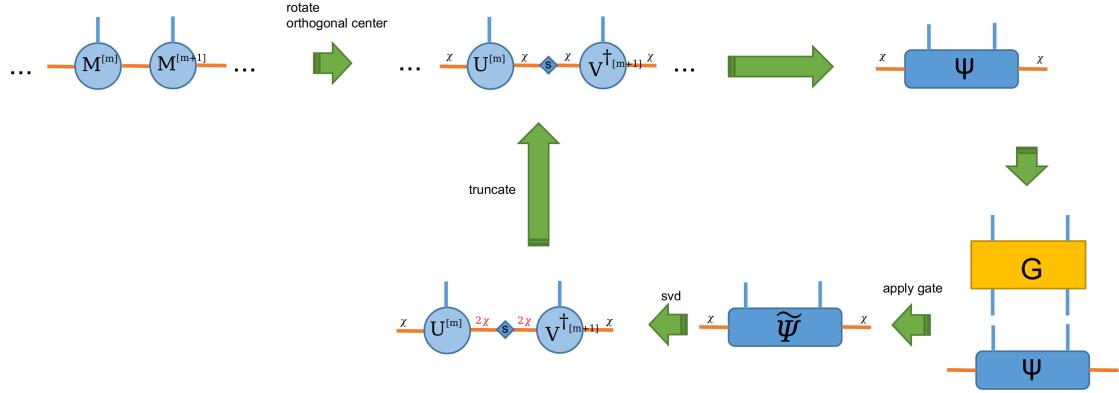


FIG. 4.

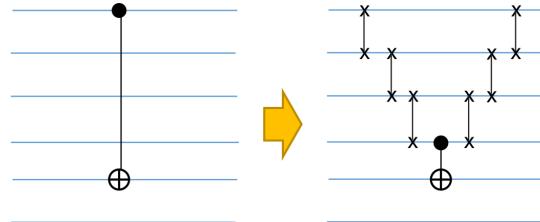


FIG. 5.

is an obstacle for this kind of gate application to MPS. In quantum computation, we say that a gate is local in the sense that it only involves finite qubits. However, in MPS base simulation in order to reduce the computational load, we want the site to be involved in applying gate to be as less as possible, i.e. we also want a gate to be "nearest neighbor"(NN). Thus practically, if a local gate applies to qubits that are separated by other qubits, then one need to first using swap gates to make those qubits NN, and depends on the scenario, swap back to its original position. In Fig. 5, we show an example of such decomposition for a CNOT gate acting on a long range qubit pair.

C. Preparing product state

The product state is sometimes used as the initial state in tensor network algorithm. In quantum computing, most of the algorithm start with preparing states as equal superposition of classical states, which is the product state that can be constructed starting from the $|0\rangle$ state and then apply single-qubit Hadamard gates on all sites.

The product state can be easily constructed in MPS representation with virtual bond dimension one. Consider a product state $|x\rangle = |x_0\rangle |x_1\rangle \dots |x_n\rangle$ where $x_i = \{0, 1\}$ is binary variable. Then it is straightforward to see that one can write each qubit state as a two dimensional vector $|0\rangle = \langle 1, 0 \rangle$ and $|1\rangle = \langle 0, 1 \rangle$. Since a vector can be extend to arbitrary rank tensor by adding redundant legs with dimension one (as shown in

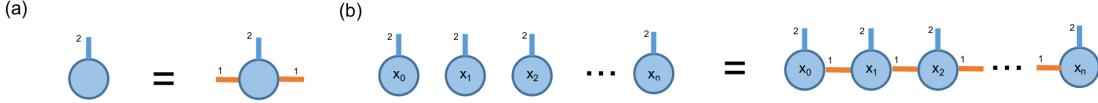


FIG. 6. (a) A 1-rank tensor (vector) can be extended to 3-rank tensor by adding two redundant legs with dimension one. (b) the MPS for the product state can be constructed by this extension with adding redundant virtual bonds.

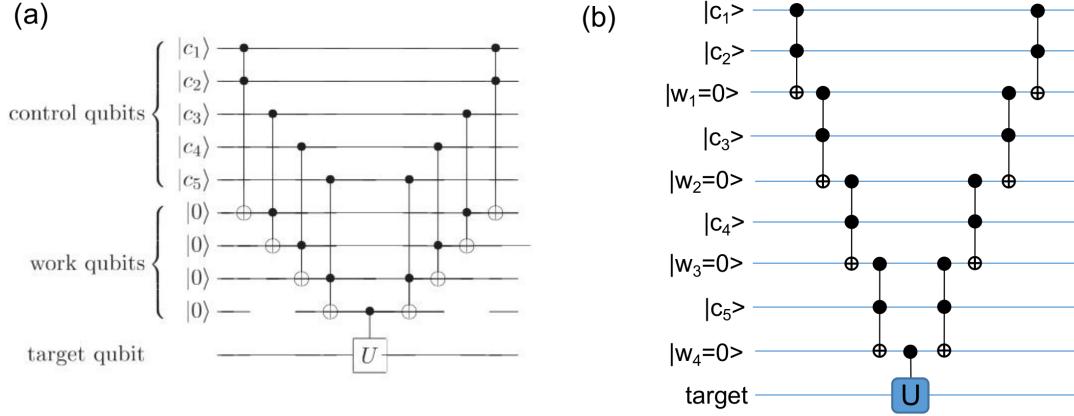


FIG. 7.

Fig. 6(a)), it follows that the MPS for product state is simply just adding each site vector with two redundant legs (with dimension=1) as illustrated in Fig. 6(b)

III. IMPLEMENTATION OF MULTI-CONTROL GATE BASE ON TOFFOLI

Before going into the Grover algorithm, let's first discuss the implementation of multi-control gate, as it will heavily be used in Grover algorithm. Generally, one can use any universal gate set to construct multi-control gate. Here, we consider the implementation using Toffoli gate with scratch qubits as we have done in one of the homework. The decomposition are shown in Fig. 7(a) from NC[8] Figure 4.10.

One might think the Toffoli gate is a three site gate, and maybe it would be better for simulation if we further decompose Toffoli into two-qubit and one-qubit gates. However, it has been shown that in the MPS calculation, it is more computational efficient to just use three-site Toffoli gate directly than further decompose Toffoli into more fundamental two site and single site gates [9].

Another thing worth to mention here is that at first glance in Fig. 7, the toffoli gates seems to be not NN and one would need SWAP gates when implementing with MPS approach. However, notice that if we re-arrange the order of control, target and work (ancilla) qubits, it is possible to avoid the use of SWAP gate as shown in Fig. 7(b). Thus in the following discussion, our implementation of MPS will base on this arrangement.

In the above construction of multi-control gate from Toffoli, one has to be careful about how entanglement of the state varies when go through intermediate Toffoli gates. Since the application to local gate always

increase the corresponding virtual bond dimension of MPS, inevitably one would need to perform truncation at intermediate Toffoli gates when the bond dimension reached the threshold. One of the question is whether or not this decomposition leads to a possibility that the entanglement at some intermediate toffoli gate steps becoming larger than the actual entanglement of final state.

To study this, we first consider a "multi-control identity gate" $C^n(I)$ which is simply just n -bit identity operation, and we initialize our state as a equal superposition of control state, which can be prepared by starting with $|000\dots 0\rangle$ state and apply Hadamard gate on clause qubits $|c_i\rangle$:

$$|\psi(t=0)\rangle = |\vec{c}\rangle \left| \vec{w} = \vec{0} \right\rangle = \left(\sum_x |x\rangle \right) \left| \vec{w} = \vec{0} \right\rangle \quad (18)$$

Since this initial state is a product state, the entanglement at each location of bipartition should be exactly zero. Notice also that the entanglements of our final state should also be exactly zero since the multi-control identity is just identity operation. The whole circuit is shown in Fig. 8(a). Now, our goal is to see how the entanglement at each bipartition location l changed as we go through the decomposition circuit. In Fig. 8(b), we plot the entanglement of bipartition at site r as a function of gate step t , and consider number of clause-qubits $n = 10$ (so the total number of qubits considered is $n_{tot} = 2n - 2$ when including the ancilla qubits).

From the result, we observe that the first application of Toffoli entangle the first three qubits, for which the entanglement is the largest of all t . The successive toffoli gates "diffuse" the entanglement to higher r location, until we reach the mid-point, and the control identity is operate on the last two qubits (which here is simply just operate nothing). Then, the rest of successive Toffoli gates gradually remove the entanglement between states until the last step we recover the initial product state.

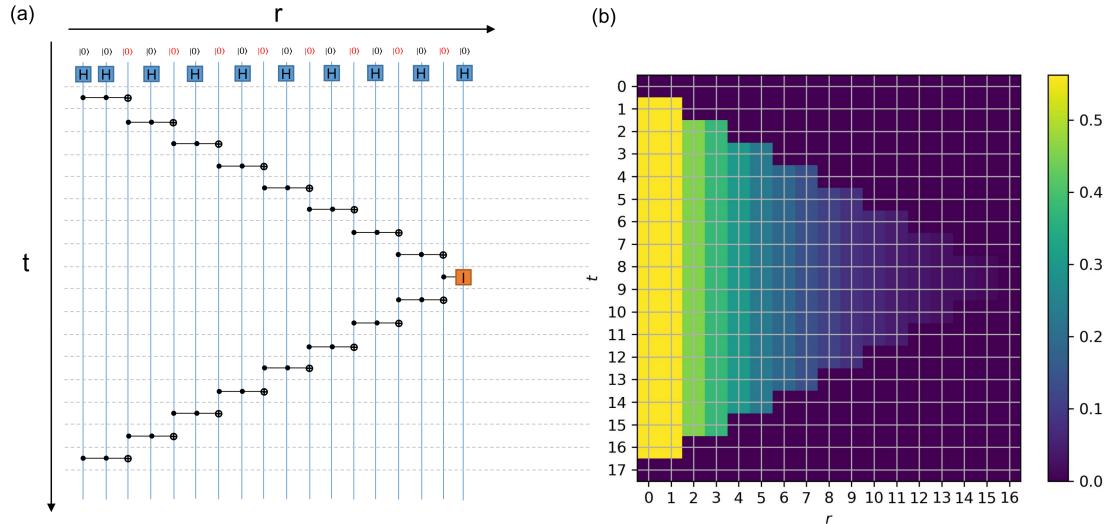


FIG. 8. (a) The circuit diagram for a decomposed multi-control identity gate. Here, the total clause-qubit $n = 10$ (black) and the ancilla qubits are marked by red. Initial clause state is prepared to be the equal superposition state by applying Hadamard gates. (b) The entanglement entropy at different bipartition location r as a function of gate step t for a decomposed multi-control identity gate. Here, r is defined as the location of rotation center (bipartition location) between site r and $r + 1$ qubits.

Next, let's consider the multi-control Z gate, which is legit to study as the gate is actually used in Grover algorithm (in fact, the multi-control Z gate is the only gate that involves multiple bits in the Grover search with one target, as we will see later). Again, we use the same set-up as in Fig. 8(a) but replace the identity operator with pauli-Z. As shown in Fig. 9(a), the entanglement plot shows a similar structure as the multi-control I. In fact, the first $n - 1$ steps which is the Toffoli ladder part is exactly the same for $C^n(Z)$ and $C^n(I)$. To further compare them in detail, in Fig. 9(b), we plot the entanglement entropy at different time t after the mid-point. We find that at the mid-point, the control-Z gate entangled the last two qubits, which then after the mid point, the successive Toffoli ladders removing the "background" entanglement, and eventually get the final state.

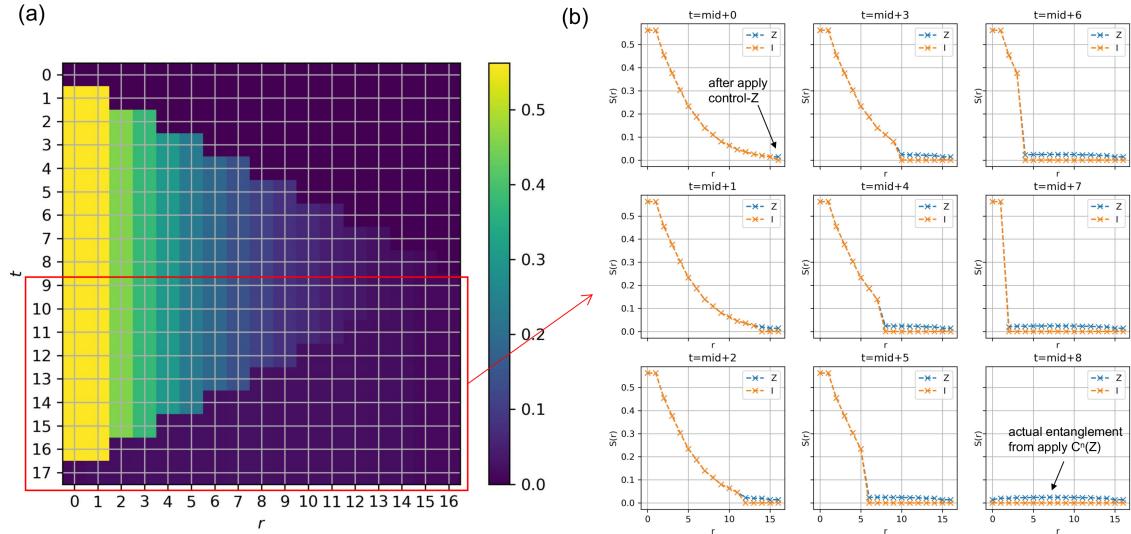


FIG. 9. (a) The entanglement entropy at different bipartition location r as a function of gate step t for a decomposed multi-control Z gate. Here, r is defined as the location of rotation center (bipartition location) between site r and $r + 1$ qubits. (b) Comparison of $C^n(I)$ and $C^n(Z)$ at different time after apply the control gate at mid point.

One of the interesting thing that we observe here is the decomposition seems to give large entanglement in the intermediate steps. The first half Toffoli-ladder construct an intermediate "background" entanglement, and the application of control-gate at mid point entangled the last two qubits. Then the inverse Toffoli-ladder remove this background entanglement. What we observe here is that this background entanglement is actually larger than the actual entanglement of the final output.

IV. GROVER ALGORITHM

After implementing and studying the multi-control gate, we now move to the study of Grover algorithm. In Grover algorithm, the goal is to perform search of the target (classical) states. The algorithm starts with the initial state that is prepared as the equal superposition of all classical states. Then the Grover rotation G rotate the state in two dimension space spanned by target state $|\omega\rangle$ and it's orthogonal state $|s'\rangle \sim \sum_{x \neq \omega} |x\rangle$.

Each Grover rotation G consist of two parts: First, the oracles

$$U_\omega = I - 2 |\omega\rangle\langle\omega| \quad (19)$$

reflect the state along $|s'\rangle$. Then the amplifier

$$U_s = 2 |s\rangle\langle s| - I \quad (20)$$

reflect the state along the initial state $|s\rangle = \sum_x |x\rangle$ as shown in Fig. 10 from wikipedia [10]. The probability for measuring the target state after T Grover rotation is then

$$P_\omega = \sin^2 \left((T + \frac{1}{2})\theta \right) \quad (21)$$

where $\sin \frac{\theta}{2} = \frac{r}{\sqrt{2^n}}$ with r being the number of targets and n is the number of clause qubits. In the following, we will study the case of single target first, and then move to multiple targets.

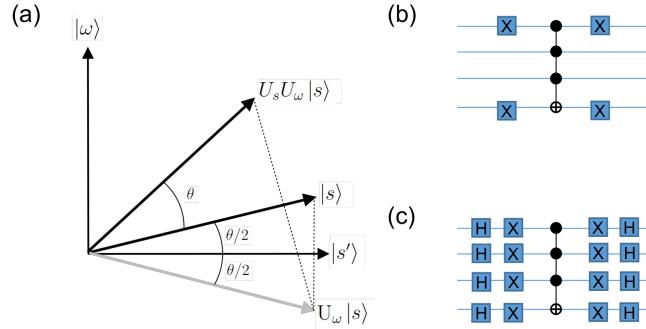


FIG. 10. (a) The Grover rotation rotate the state in the two dimension space spanned by target state $|\omega\rangle$ and it's orthogonal state $|s'\rangle$. (b) The example circuit diagram of oracle with $n = 4$ and target $|\omega\rangle = |0110\rangle$. (c) The example circuit diagram of amplifier with $n = 4$.

A. Single target

1. Single target oracle and implementation

For a single target $|\omega\rangle = \prod_i |\omega_i\rangle$ where $\omega_i = \{0, 1\}$ is classical bit, the oracle can be constructed with multi-control Z gate and single qubit pauli-X gates [11] as following:

$$U_\omega = \left(\prod_i (X_i)^{(\omega_i+1)\bmod 2} \right) C^n(Z) \left(\prod_i (X_i)^{(\omega_i+1)\bmod 2} \right) \quad (22)$$

In Fig. 10(b), we show the circuit diagram with $n = 4$ and $|\omega\rangle = |0110\rangle$. The amplification can be constructed using multi-control Z gate, single qubit pauli-X and Hadamard gates as

$$U_s = \left(\prod_i H_i X_i \right) C^n(Z) \left(\prod_i X_i H_i \right) \quad (23)$$

example of the circuit diagram with $n = 4$ is shown in Fig. 10(c).

In our simulation, we start by first preparing the initial state as a product state of $|000\dots00\rangle$ including the ancilla qubits that are used to implement the multi-control gate. The order of clause qubits and ancilla qubits in MPS follows the same arrangement as we mentioned in Sec. III. We then acting on the Hadamard gates on clause qubits to put the state into the equal superposition state $|s\rangle = \sum_x |x\rangle$. The Grover rotation $G = U_s U_\omega$ then apply to the MPS.

To check our implementation, we first do the simulation without any truncation made to MPS, and compare with the exact result for the probability. In Fig. 11(a), we show the simulation result for system with $n = 5$ (total number of qubits including ancilla is $2n - 2$). From the exact result, we know that the exact probability for getting the target state should be

$$P(\omega) = \sin^2((T + 1/2)\theta). \quad (24)$$

In our simulation here, we consider $|\omega\rangle = |01110\rangle$, and we measure the probability of $|\omega\rangle$ after each Grover iteration T . We find that our results match with the exact probability $\sin^2((T + 1/2)\theta)$, which confirms the correctness of our MPS implementation.

Further more, we can also look at other classical states' probability. In Fig. 11(b), we plot the probability not only for the target state, but the probability of all the classical states x (represented as integer value here) as a function of T . We observe all the non-target classical states have same probability at each iteration T and oscillating conjugate to the oscillation of target state. This result can be easily understood as following. The other non-target states are orthogonal to the target state $|\omega\rangle$, which is packed in the orthogonal state $|s'\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{x \neq \omega} |x\rangle$ defined as equal superposition of non-target state. The probability amplitude for getting $|s'\rangle$ is then

$$A(s') = \cos((T + 1/2)\theta) \quad (25)$$

Since $|s'\rangle$ is equal superposition of non-target states, then it follows that all the non-target states will have same probability, with probability any given non-target state $|x\rangle$ is

$$P(x) = \frac{1}{2^n - 1} \cos^2((T + 1/2)\theta). \quad (26)$$

We also provide the comparison of this exact probability with the probability measured in MPS simulation in Fig. 11(a)

2. Entanglement along iteration

One of the interesting thing worth study is to see how the entanglement entropy evolves with Grover iteration. To study this, we consider a system with $n = 8$ (for which total number of sites/qubits is $2n - 2$), and single target state $|\omega\rangle = |01110011\rangle$. Again, we use the same alternative orientation of clause and ancilla qubits as in Sec. III in order to have benefit in multi-control gate implementation. The circuit diagram is shown in Fig. 12 where each Grover rotation G is represented by the red square, and each G consist of a oracle U_ω (light orange) and a amplifier U_s (light green).

The entanglement results at different bipartition location r as a function of gate step t are shown in the upper part of Fig. 12. We observe that the entanglement entropy has the similar oscillating feature as the probability for getting the target state.

The result is actually not surprising. One can interpret the result by thinking about how the Grover algorithm works. As shown in Fig. 13(a), the Grover algorithm reflect and effectively rotate the state in a

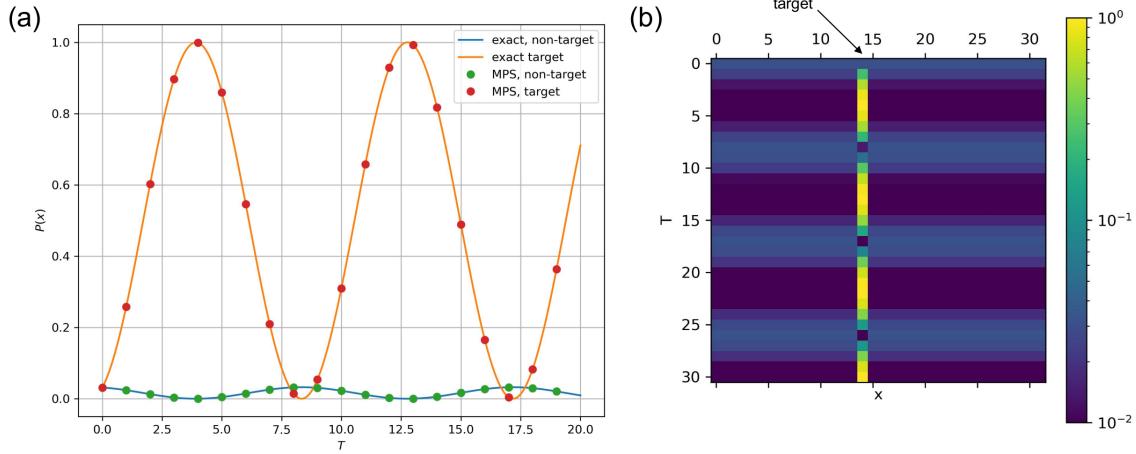


FIG. 11. (a) The probability for target state and non-target state measured in simulation, and compare with exact analytical result. (b) The probability for all the states as a function of Grover iteration T . The system is $n = 5$ (total qubits including ancilla is $2n - 2$), and the target state we considered here is $|\omega\rangle = |01110\rangle$

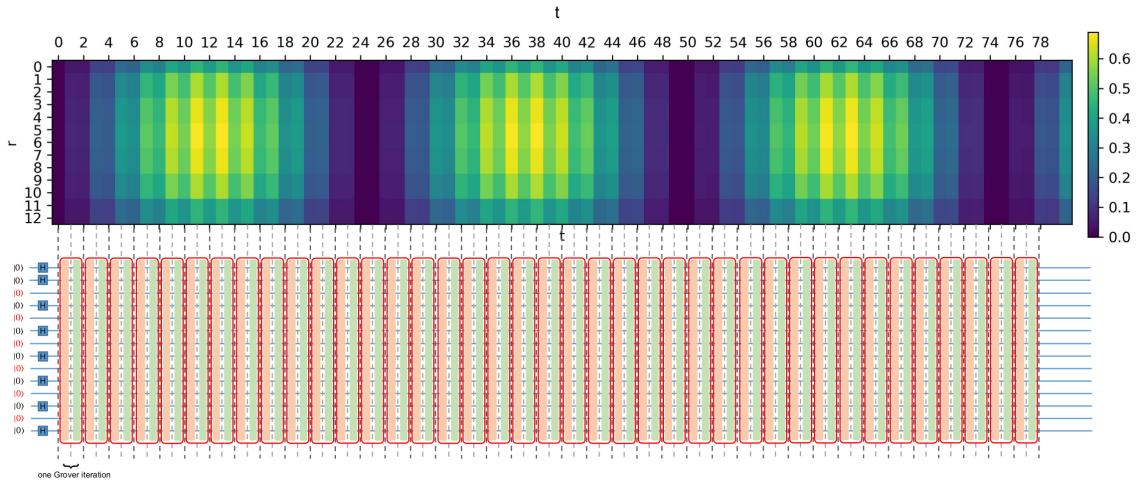


FIG. 12. The entanglement map at different bipartition location r as a function of gate step t (upper figure). The lower figure shows the corresponding qubits orientation and circuit diagram. The system we consider is $n = 8$ (total qubits is $2n - 2$). The qubits marked by red is the ancilla qubits. Here, t is defined as applying a single U_ω or U_s so that a complete Grover interaction has two step. r is defined as the bipartition location where the orthogonal center is in between site r and $r + 1$.

two dimension space spanned by $|\omega\rangle$ and $|s'\rangle = \sum_{x \neq \omega} |x\rangle$. For single solution $|\omega\rangle$ is a product state, thus it should have zero entanglement entropy. Further more, the initial state $|s\rangle$ is also a product state since it's just a basis transformation of $|0\rangle$ product state. Then the Grover iteration rotate the state starting from a product state, rotate counterclockwise in this two dimension space. One can expected in the intermediate

angle between $|\omega\rangle$ and $|s\rangle$, there will be a region where the state has higher entanglement.

In Fig. 13(b) we plot the probability of target state as well as the entanglement entropy for the equal bipartition. We find that whenever the state is close to $|s\rangle$ or $|\omega\rangle$, the entanglement entropy drops to almost zero, and the entanglement shows maximum when the state is rotated in between.

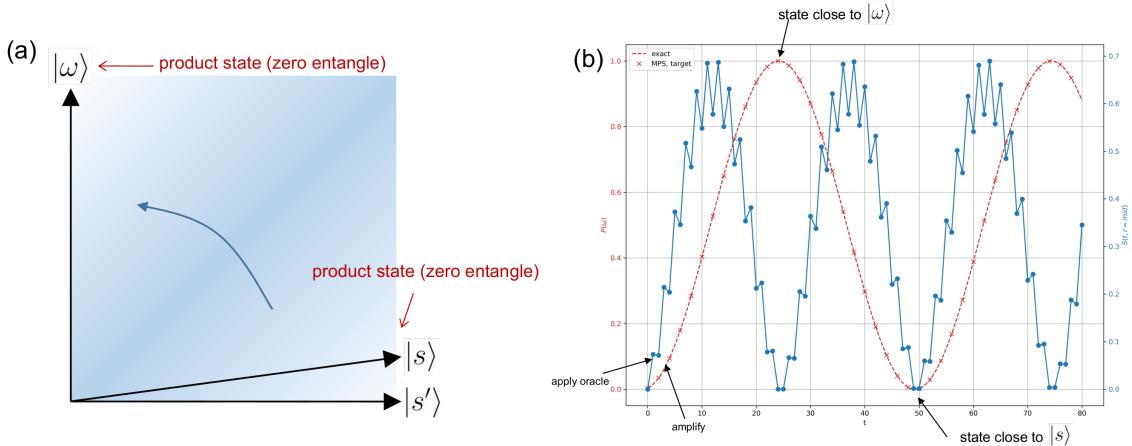


FIG. 13. The probability for target state $|\omega\rangle$ and the entanglement entropy with equal bipartition (orthogonal center at middle of MPS.) as a function of gate step t defines as in Fig. 12.

3. Truncation effect

In the previous study, we look into the entanglement in the state at each Grover iteration, and we see that the entanglement oscillate when state rotate between $|s\rangle$ and $|\omega\rangle$. So far, we haven't apply any truncation to our virtual bond dimension. We just let it growth until the number of parameter reach 2^n .

The next thing we would like to do is to perform truncation, and see how it affect the result. In the following, we perform truncation after each local gate application on MPS using the procedure introduced in Sec. II B. We set a maximum bond dimension χ , so that whenever a certain virtual bond dimension growth exceed χ after applying the gate, we truncate the bond dimension down to χ .

In Fig. 14(a), we plot the probability of target state for $n = 6$ (total qubits $2n - 2$) with different cut-off χ . We find that even with $\chi = 2$ gives good match to the exact analytic result. Other system size n shows the same results. We can actually do better by changing the truncation criterion. Instead of setting a maximum bond-dimension χ , we set a truncation error ϵ and truncate all the singular values less than ϵ after applying each local quantum gate. In Fig. 14(b), we plot the maximum bond-dimension of the whole MPS as a function of gate step t by setting $\epsilon = 1.0e - 12$. We find that the bond dimension saturate to 2, indicates that for Grover algorithm with single solution, MPS with only dimension 2 should be enough to give exact result.

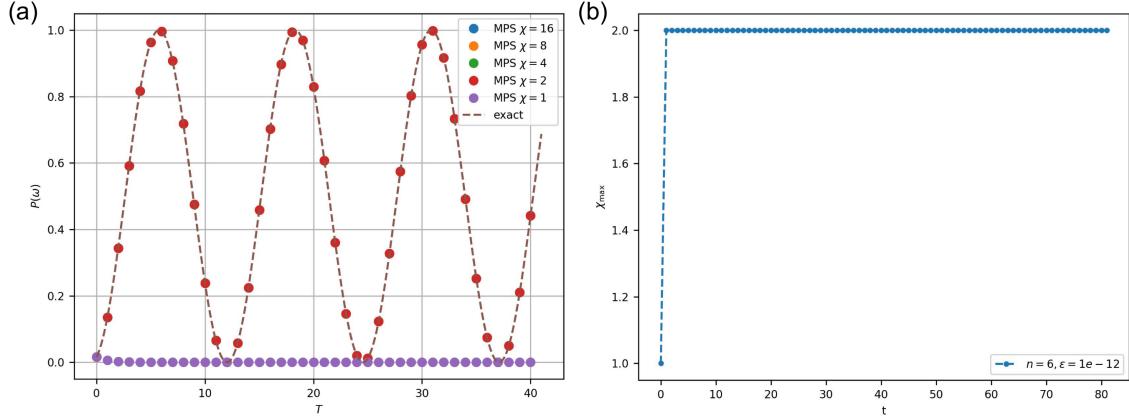


FIG. 14. (a) The probability for target state $|\omega\rangle$ with different MPS virtual bond dimension cutoff χ for a system size $n = 6$ (total qubits including ancillas $2n - 2$). (b) The maximum virtual bond dimension as a function of gate step t in the same $n = 6$ system with setting truncation cutoff $\epsilon = 1.0e - 12$ instead of a fixed maximum cutoff bond dimension. Here, gate step t is in unit of either U_ω or U_s as previously defined, while T is the number of grover iteration.

B. Multiple targets

From the previous study, we show that for a single target, a MPS with bond-dimension 2 is sufficient to represent all the intermediate state. We will now study the case with multiple targets.

1. Multiple-target oracle implementation

For the simplicity of implementing oracle, here we consider the number of targets be $r = 2^m$ where $0 \leq m < n$ is any integer less than the system size n , and consider target states to be $|\omega_i\rangle = (\prod_0^{n-r} |1\rangle) |r_i\rangle$. For example, in system with $n = 6$, $r = 4$ the target states will be $|111111\rangle$, $|111110\rangle$, $|111101\rangle$ and $|111100\rangle$. The oracle for these target states can be easily constructed via only multi-control Z as

$$U_{\vec{\omega}} = C^{n-m}(Z). \quad (27)$$

The implementation of amplification U_s is the same as previously described in Sec. IV A 1. In Fig. 15, we show the example implementation of the oracles for system size $n = 4$ with different $r = 2^m$.

The probability for getting a target state can be easily calculated. Since the target state is now $|\omega\rangle = \frac{1}{\sqrt{r}} \sum_i |\omega_i\rangle$ then we have

$$\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{2^n} \sqrt{r}} \times r = \sqrt{\frac{r}{2^n}} \quad (28)$$

the probability for getting a certain $|\omega_i\rangle$ is then

$$P(\omega_i) = \frac{1}{r} \sin^2\left((T + \frac{1}{2})\theta\right) \quad (29)$$

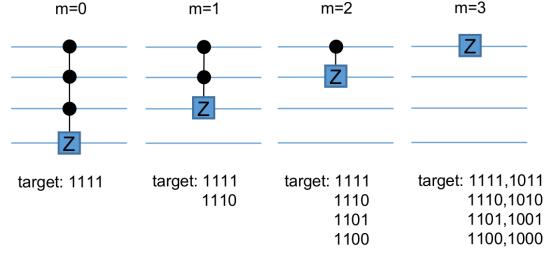


FIG. 15. Implementation of multi-target oracle using multi-control-Z gate. Here $n = 4$ and the number of solution $r = 2^m$ where m is integer.

where T is the number of Grover iteration. Again, to check if our MPS simulation works as we expected, we start with simulation without truncation and compare with the exact results. In Fig. 16(a), we plot the MPS simulation results with size $n = 5$ with different r with this exact probability, the consistent of results confirms the correctness of our MPS implementation. Furthermore, for other non-target states, with the same argument as in the case of single target, all the non-target state $|x\rangle$ should have the same probability as

$$P(x) = \frac{1}{2^n - r} \cos^2((T + 1/2)\theta). \quad (30)$$

Fig. 16(b) shows the probability for all the classical states.

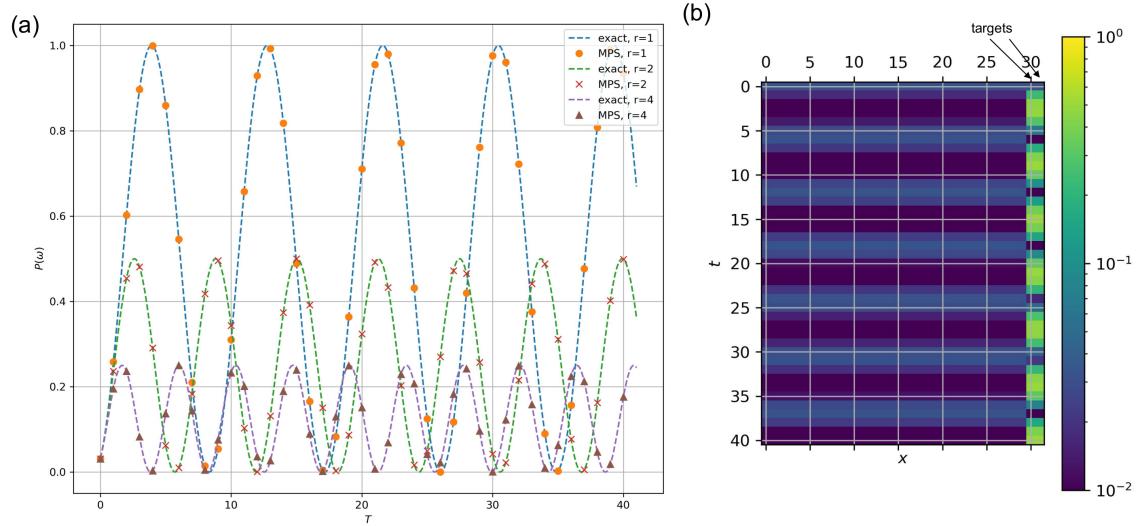


FIG. 16. (a) Probability of a given target with number of targets r in the Grover search. (b) The probability calculated via MPS simulation for all the classical states $|x\rangle$ (packed in integer values) as a function of gate time t , with number of targets $r = 2$.

2. Entanglement along iteration

Different from the case of single solution, where $|\omega\rangle$ is product state with zero entanglement entropy. With multiple solution, generally $|\omega\rangle = \sum_i |\omega_i\rangle$ is not a product state, and should have finite entanglement. The initial state $|s\rangle$ is of course, always a product state. Then the question we want to address here, is how the entanglement in the state changed as the state rotate in the two dimension space when there are more than one solution.

In Fig. 17, we plot the entanglement map for system size $n = 8$ (total qubits $2n - 2$ including ancilla qubits) with different number of solutions r . Interestingly, we find that for these special sets of targets, the higher qubits seems not entangled at all. From the construction of our multi-target oracle U_ω considered here, we know that for a given r , the multi-control Z does not act on the highest r qubits, so it's straightforward to see that it keeps the highest qubits invariant. However, the amplifier U_s does have a multi-control Z acting on all the qubits. To explain this result, let's slightly re-written the initial state as

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum |x\rangle = |\tilde{s}\rangle \otimes |k\rangle \quad (31)$$

where $|\tilde{s}\rangle = (\sum |y\rangle)$ is the equal superposition of $(n - r)$ -bits classical state $|y\rangle$, and similarly $|k\rangle$ is the equal superposition of the last r -bit classical state. Now, consider after applying the oracle, only the first $(n - r)$ qubits state are changed, and the $|k\rangle$ is invariant. The state after applying an oracle is then

$$|\psi\rangle = |\tilde{\psi}\rangle |k\rangle \quad (32)$$

where $|\tilde{\psi}\rangle$ is the first $(n - r)$ -qubits state. The Amplifier $U_s = 2|s\rangle\langle s| - I$ acting on the state gives:

$$U_s |\psi\rangle = (2|\tilde{s}\rangle |k\rangle \langle k| \langle \tilde{s}| - I) |\psi\rangle \quad (33)$$

$$= 2 \langle \tilde{s} | \tilde{\psi} \rangle |\tilde{s}\rangle |k\rangle - |\tilde{\psi}\rangle |k\rangle \quad (34)$$

$$= (2 \langle \tilde{s} | \tilde{\psi} \rangle - |\tilde{\psi}\rangle) |k\rangle \quad (35)$$

$$= |\tilde{\psi}_{\text{new}}\rangle |k\rangle \quad (36)$$

which keeps the $|k\rangle$ invariant as the product state of two bipartition. Thus we see that there should be no entanglement between the first $n - r$ -qubits and the rest r -qubits, and the last r -qubits state $|k\rangle$ stays as the product state. This is exactly what we observed in our MPS simulation.

3. Truncation effect

Finally, let's discuss the truncation of MPS when there are multiple solution. For the case we considered above with multiple targets with $n = 8$, we find that if we set $\epsilon = 1.0e - 12$, the maximum bond dimension at any given gate time t never exceed $\chi = 2$, exactly as we seen in the single solution case in Fig. 15(b).

The multiple solutions that we considered here have a structure, where the target state can always be written as product state of two bipartition $|\omega\rangle = |\tilde{\omega}\rangle \otimes |k\rangle$, where $|k\rangle$ is the equal superposition of classical bits. In this case the multiple solutions reduce to single solution finding in the $|\tilde{\omega}\rangle$ subspace, and hence the conclusion from the single solution finding applies. The MPS with bond dimension 2 is sufficient to represent the state.

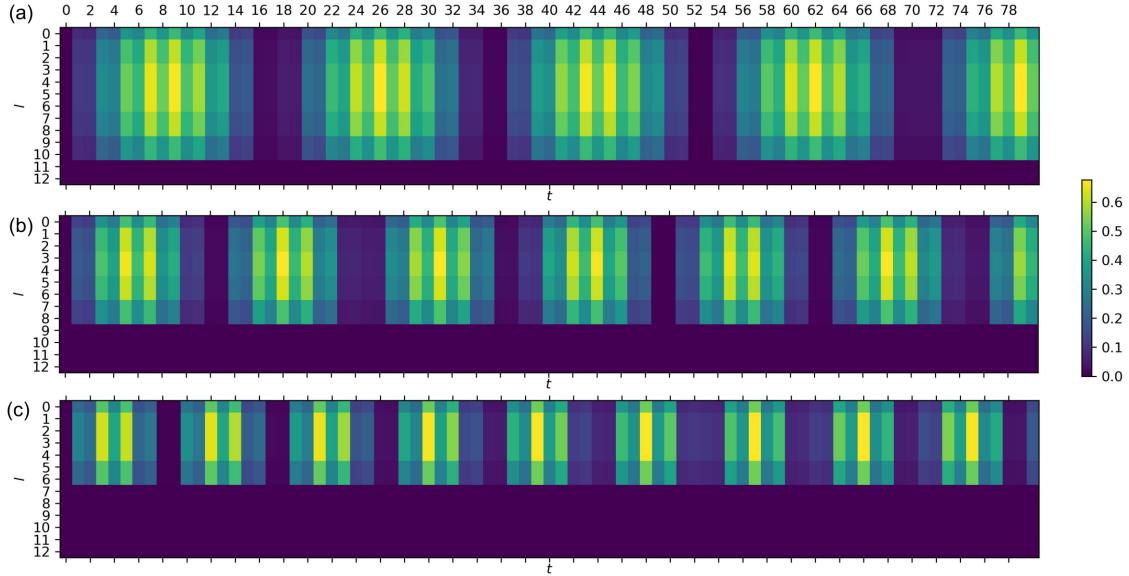


FIG. 17. Entanglement map at different bipartition location l calculated from the MPS simulation, as a function of gate step t . Here, the system size $n = 8$ (total qubits $2n - 2$ including ancilla). (a) With $r = 2$ targets ($m = 1$) (b) $r = 4$ targets ($m = 2$) (c) $r = 8$ targets ($m = 3$)

V. CONCLUSION & OUTLOOK

We introduce the matrix product state as an efficient representation for the quantum state. Using this representation, we discuss how one can benefit from this MPS structure which allow us to efficiently acting local quantum gates on the state. We discuss in detail how the multi-control gate can be construct base on Toffoli decomposition, and study the entanglement entropy in the intermediate steps of this decomposition. We find that the first half intermediate Toffoli ladder raise a background entanglement in the system that is even larger than the final state, and the second half Toffoli ladder remove this background entanglement.

We then use this MPS approach to study the Grover algorithm with single and multiple solutions. Specifically we study how the entanglement entropy evolves at each Grover iteration, and shows that for Grover algorithm, a maximum bond-dimension 2 is enough to capture the state even with the Toffoli decomposition for multi-control gate.

As we mentioned in the content, the multiple targets we considered here can always be constructed as product state of two bipartition $|\omega\rangle = |\tilde{\omega}\rangle \otimes |k\rangle$, where $|k\rangle$ itself is product state. In this case the multiple solution reduce to single solution finding in the $|\tilde{\omega}\rangle$ subspace and bond dimension 2 is sufficient to represent the state. It will also be interesting to study the case where the multiple solution cannot be written in this form in the future, and also study other algorithm with this approach.

VI. SOURCE CODE

The source codes for all the simulation studies in this report can be found on github [12], and are implemented using Cytnx v0.7.5+.

-
- [1] A. Kawaguchi, K. Shimizu, Y. Tokura, and N. Imoto, arXiv e-prints , quant-ph/0411205 (2004), arXiv:quant-ph/0411205 [quant-ph].
- [2] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
- [3] A. SaiToh and M. Kitagawa, Phys. Rev. A **73**, 062332 (2006).
- [4] C. Chamon and E. R. Mucciolo, Phys. Rev. Lett. **109**, 030503 (2012).
- [5] Y. Zhou, E. M. Stoudenmire, and X. Waintal, Phys. Rev. X **10**, 041038 (2020).
- [6] M. S. A. et.al., “Qiskit: An open-source framework for quantum computing,” (2021).
- [7] K.-H. Wu, “Project cytnx, a cross-section of python and c++ tensor network library,” .
- [8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
- [9] A. SaiToh, Computer Physics Communications **184**, 2005 (2013).
- [10] Wikipedia, “Grover’s algorithm — Wikipedia, the free encyclopedia,” http://en.wikipedia.org/w/index.php?title=Grover%27s_algorithm&oldid=1055896229 (2021), [Online; accessed 05-December-2021].
- [11] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe, Nature Communications **8**, 1918 (2017).
- [12] “Source code of this project on Github,” https://github.com/kaihsin/cytnx_grover.