Learning architectures based on quantum entanglement: a simple matrix product state algorithm for image recognition

Yuhan Liu,[1] Xiao Zhang,[1] Maciej Lewenstein,[2,3] and Shi-Ju Ran[2,*]

[1] *Department of Physics, Sun Yat-sen University, Guangzhou 510275, China*
[2] *ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain*
[3] *ICREA, Passeig Lluis Companys 23, 08010 Barcelona, Spain*

(Dated: May 5, 2019)

It is a fundamental, but still elusive question whether methods based on quantum mechanics, in particular on quantum entanglement, can be used for classical information processing and machine learning. Even partial answer to this question would bring important insights to both fields of both machine learning and quantum mechanics. In this work, we implement simple numerical experiments, related to pattern/images classification, in which we represent the classifiers by quantum matrix product states (MPS). Classical machine learning algorithm is then applied to these quantum states. We explicitly show how quantum features (i.e., single-site and bipartite entanglement) can emerge in such represented images; entanglement characterizes here the importance of data, and this information can be practically used to improve the learning procedures. Thanks to the low demands on the dimensions and number of the unitary matrices, necessary to construct the MPS, we expect such numerical experiments could open new paths in classical machine learning, and shed at same time lights on generic quantum simulations/computations.

I. INTRODUCTION

Classical information processing deals with pattern recognition and classification. The classical patterns in question may correspond to images, temporal sound sequences, finance data, and so on. During the last thirty years of developments of the quantum information science, there were many attempts to generalize classical information processing to the quantum world, for instance by proposing quantum perceptrons and quantum neural networks (e.g., see some early works [1–3] and a review [4]), quantum finance (e.g., [5]), quantum game theory [6–8], to name a few. More recently, there were successful proposals to use quantum mechanics to enhance learning processes by introducing quantum gates, or quantum computers [9–13].

Conversely, there were various attempts to apply methods of quantum information theory to classical information processing task, for instance by mapping classical images to quantum mechanical states. In 2000, Hao et al. [14] developed a different representation technique for long DNA sequences, obtaining mathematical objects similar to many-body wave-function. In 2005 Latorre [15] developed independently a mapping between bitmap images and many-body wavefunctions which has a similar philosophy, and applied quantum information techniques in order to develop an image compression algorithm. Although the compression rate was not competitive with standard jpeg, the insight provided by the mapping was of high value [16]. A crucial insight for this work was the idea that Latorre's mapping might be inverted, in order to obtain bitmap images out of many-body wave-functions. In fact, in Ref. [17] developed a reverse idea, and mapped quantum many body states to images.

Recently, there was a considerable progress in the interdisciplinary field merging quantum many-body physics and machine learning [18]. From one side, machine learning techniques are introduced to solve challenging physical problems. For example, it has been proposed to use neural networks to learn quantum phases of matter, and detect quantum phase transitions [19–33]. Different schemes of machine learning, including supervised, unsupervised, and reinforcement learning, are applied to systems of spins, bosons and fermions, combined with gradient methods, Monte Carlo, and so on.

On the other side, methods of quantum many-body physics, particularly tensor network (TN) [34–37], are used to understand and tackle machine learning problems [38–43]. TN is a mathematical model that is defined by a number of tensors contracted together in a specific way. TN originates from quantum many-body physics. A many-body system usually cannot be well described analytically by single-particle approximations due to the strong correlations. Numerically, the vector space (usually called Hilbert space) to describe such systems suffers an exponential growth with the size of the system. TN provides an efficient mathematical structure, with which the memory cost using classical computations grows only polynomially with the size of the system. Many TN ansatz's have been proposed, such as matrix product states (MPS) [35], projected entangled pair states [35, 44], tree TN states [45], or multi-scale entanglement renormalization ansatz [46].

Recently, TN proved their great potential in the field of machine learning, providing a natural way to build the mathematical connections between quantum physics and classical information. The data gathered from the classical world (images, language, etc.) could be char-

* Corresponding author. Email: shi-ju.ran@icfo.eu

acterized and processed not only by classical statistics and computations, but also by quantum approaches and simulations. Among others, MPS has been utilized to supervised image recognition [38] and generative modeling to learn joint probability distribution [39]. Tree TN that has a hierarchical structure is also used to natural language modeling [42] and image recognition [40, 41], which is proven to be of high efficiency. The relations between the mathematical models of machine learning, e.g., Boltzmann machine and TN states, MPS and string-bond state, and deep convolutional arithmetic circuits and quantum many-body wave functions, have been investigated [43, 47–49]. However, both the relations between the quantum features and classical data, as well as ways of utilizing quantum features to improve machine learning such as image recognition, are still elusive.

In this work, we implement simple numerical experiments and show how quantum entanglement can emerge from images and be used for the learning architecture. We map sets of images consisting of pixels of a certain shade of grey, onto vectors (states) in a Hilbert space of high dimensions. The classifiers of the encoded images are represented as MPS's. A training algorithm based on Multiscale Entanglement Renormalization Ansatz (MERA) is then used to optimize the MPS. We show, considering both the images before and after the discrete cosine transformation (DCT), that the efficiency of such classical computation is directly related to the bipartite entanglement entropy (BEE). The MPS for classifying the images after DCT possesses much smaller BEE, meaning higher efficiency, than the MPS for classifying the images before DCT. We demonstrate also that the single-site entanglement entropy (SEE) of the trained state characterizes the local importance of the data. This permits to discard the less important data, so that the necessary length of the MPS can be largely reduced. Our simulations show that to reach the same accuracy, the length of the MPS for classifying the images after DCT is about ten times smaller than the MPS for classifying before DCT. Furthermore, we propose to reorder the data according to SEE, and achieve in this way higher computational efficiency without harming the accuracy.

II. REVIEW OF MATRIX PRODUCT STATE AND TRAINING ALGORITHM

The basic idea is after mapping the classical data into a vector (quantum Hilbert) space, quantum states (or the quantum operator formed by these states) are trained to capture different classes of the images, in order to solve specific tasks such as classifications. Since the Hilbert space is usually exponentially large when the size of the images increases, TN (MPS in this work) are to implement the calculations efficiently by classical computers.

Such machine leaning contains two central ingredients. One is the feature map [41] that transform each input image to a product state. Each pixel (say, the l-th pixel $\theta_{n,l}$ of the n-th image) is transformed to a d-dimensional normalized vector as

$$v_s^{[n,l]} = \sqrt{\binom{d-1}{s-1}}(\cos\Big(\frac{\pi}{2}\theta_{n,l}\Big))^{d-s}(\sin\Big(\frac{\pi}{2}\theta_{n,l}\Big))^{s-1}, \tag{1}$$

where s runs from 1 to d. Then, the n-th image is mapped to a d^L-dimensional tensor product state $\prod_{l=1}^{L}\mathbf{v}^{[n,l]}$ (L is the number of pixels of the image). Note that in the paper we use the bold face to represent vectors without explicitly writing the indices.

The output of the n-th image is obtained by contracting the corresponding vectors with a linear projector denoted by $\hat{\mathbf{\Psi}}$ as

$$u_b^{[n]} = \sum_{s_1\cdots s_L}\hat{\Psi}_{b,s_1\cdots s_L}\prod_{l=1}^{L}v_{s_l}^{[n,l]}. \tag{2}$$

$\hat{\mathbf{\Psi}}$ is actually a map from a d^L-dimensional to a D-dimensional vector space. Here, we take $\hat{\mathbf{\Psi}}$ as a unitary MPS (Fig. 1) which is written as

$$\hat{\Psi}_{b,s_1\cdots s_L} = \sum_{a_1\cdots a_{L-1}}A^{[1]}_{bs_1a_1}A^{[2]}_{s_2a_1a_2}\cdots A^{[l]}_{s_la_{l-1}a_l}\cdots A^{[L]}_{s_La_{L-1}}. \tag{3}$$

Note the indexes $\{a\}$, which are often called virtual bonds, will be all summed over. The dimensions of the virtual bonds (denoted by χ) determines the upper bound of the entanglement that can be carried by the MPS. The total number of parameters in the MPS increases only linearly with L, i.e. $O(d\chi^2L)$.

To train the MPS, we optimize the tensors $\{\mathbf{A}^{[l]}\}$ in the MPS to minimize the error of the classification. To this end, the cost function to be minimized is chosen to be the cross entropy $f^{CE} = -\sum_n \ln\Big(\sum_b B_b^{[n]}u_b^{[n]}\Big)$, with $\mathbf{B}^{[n]}$ a D-dimensional vector (D is the number of classes) that satisfies

$$B_b^{[n]} = \begin{cases} 1, & \text{if the n-th image} \in \text{the } b\text{-th class} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We use the MERA-inspired algorithm to optimize the MPS [40], where all tensors are taken as isometries that satisfy the right orthogonal condition $\sum_{s_la_l}A^{[l]}_{s_l,a_{l-1}a_l}A^{[l]}_{s_l,a'_{l-1}a_l} = I_{a_{l-1}a'_{l-1}}$ (for the rightmost tensor, it still satisfies this condition by considering it as a $\chi\times d\times 1$ tensor). Then the MPS in Eq. (3) gives a unitary projector from a d^L-dimension to a D-dimensional vector space. The tensors in the MPS can be initialized randomly, and then are optimized one by one (from right to left, for example). The key step is to calculate the environment tensor $\mathbf{E}^{[l]}$, which is defined by contracting everything after taking out the tensor $\mathbf{A}^{[l]}$ that is to be updated (Fig. 1). By implementing SVD as $\mathbf{E}^{[l]} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, the tensor is updated as $\mathbf{A}^{[l]} \leftarrow \mathbf{V}\mathbf{U}^{\mathrm{T}}$. One can see that the new tensor still satisfies the orthogonal condition.

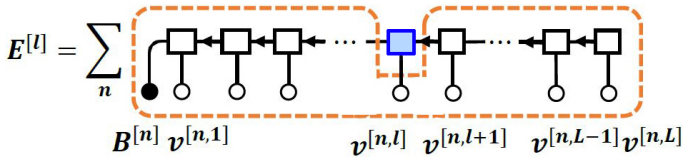

Figure 1. Illustration of MPS $\hat{\Psi}$ and the environment tensor $\mathbf{E}^{[l]}$. $\hat{\Psi}$ satisfies the orthogonal condition, indicated by the arrows. $\mathbf{E}^{[l]}$ is defined by contracting everything after taking out the tensor (blue) that is to be updated.

III. LEARNING ARCHITECTURE BASED ON QUANTUM ENTANGLEMENT

In this section, we show that by learning the images from the frequency space (reached by DCT), the computational cost can be largely reduced without lowing the accuracy. This is revealed by a lower bipartite entanglement entropy of the MPS, which means that smaller virtual bond dimensions are needed. More interestingly, we show that by calculating the single-site entanglement entropy of each site and accordingly rearranging the image data, the computational cost including the length of MPS can be further reduced without harming the accuracy. Our work demonstrates how (bipartite and single-site) quantum entanglement can be utilized to design machine learning algorithms for classical data. It exhibits an explicit example of using quantum features for learning architecture.

A. Discrete cosine transform and motivation

We use standard discrete cosine transformation (DCT) to transform the images in frequency space as

$$\eta_{u,v} = \frac{2}{M}\alpha(u)\alpha(v)\sum_{x=0}^{M-1}\sum_{y=0}^{M-1} \theta_{x,y}\cos[\frac{(2x+1)u\pi}{2M}]\cos[\frac{(2y+1)v\pi}{2M}], \tag{5}$$

with M the width/height of the images, (x, y) the position of a pixel, and $\alpha(u) = 1/\sqrt{2}$ if $u = 0$, or $\alpha(u) = 1$ otherwise. In our case, we have $M = 28$ for the images in the MNIST dataset. Note $L = M^2$.

We propose that while using MPS to deal with images, DCT is very helpful. Since MPS is essentially a 1D mathematical object, a 1D path that covers the 2D image has to be chosen to define the MPS. In the frequency space, there exists a natural 1D path for this. It is the zig-zag path shown in Fig. 2 (a) that is used in many standard image algorithms (e.g., JPEG). The frequency is non-increasing along the path. Note that in previous works using MPS, the 2D images are directly reshaped into 1D (i.e., $(1 \times M^2)$) images to define the MPS.

Moreover, it is known from the existing image algorithms that the most important information is normally stored in the low-frequency data. We will show in the following that the entanglement of the trained MPS is consistent with this property, and the computational complexity can be largely reduced when defining the MPS on the zig-zag path.

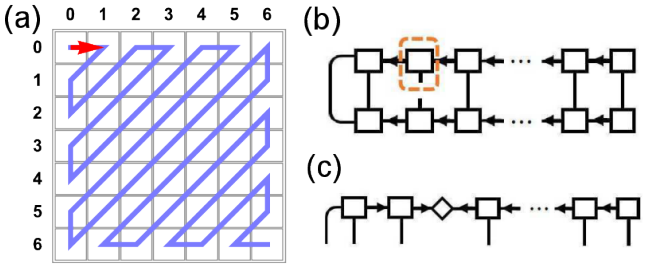

Figure 2. (a) Zigzag ordering. We use a 7×7 image as an example. Note the images in MNIST dataset are of 28×28 pixels. (b) Computation of the single-site entanglement entropy (SEE), and (c) bipartition entanglement entropy (BEE).

B. Single-site and bipartite entanglement entropy of the trained MPS

Before presenting our results, let us define the single-site entanglement entropy (SEE) (say, of the l-th site) and bipartite entanglement entropy (BEE). The reduced density matrix $\hat{\rho}^{[l]}$ of the l-th site is defined as

$$\hat{\rho}^{[l]}_{s_l s'_l} = \sum_{bs_1\cdots s_{l-1}s_{l+1}\cdots s_L} \hat{\Psi}_{bs_1\cdots s_l\cdots s_L}\hat{\Psi}_{bs_1\cdots s'_l\cdots s_L}. \tag{6}$$

Note $\hat{\rho}^{[l]}$ is non-negative. The computation of $\hat{\rho}^{[l]}$ with MPS is shown in Fig. 2 (b), where one contracts everything except s_l and s'_l. The leading computational complexity is about $O(ld\chi^3)$ after using the orthogonal condition. After normalizing $\hat{\rho}^{[l]}$ by $\hat{\rho}^{[l]} \leftarrow \hat{\rho}^{[l]}/\mathrm{Tr}\hat{\rho}^{[l]}$, the SEE is defined as

$$S^{[l]}_{\mathrm{SEE}} = -\mathrm{Tr}\hat{\rho}^{[l]}\ln\hat{\rho}^{[l]}. \tag{7}$$

The BEE measured between the l-th and $(l+1)$-th sites is similarly defined by the reduced density matrix obtained after tracing over either half of MPS. In our context, there is an easier way to obtain BEE by singular value decomposition (SVD), where BEE is obtained by the singular values (or called Schmidt numbers). The SVD is formally written as

$$\hat{\Psi}_{bs_1\cdots s_l s_{l+1}\cdots s_L} = \sum_{aa'} X_{bs_1\cdots s_l,a}\lambda^{[l]}_{aa'}Y_{a',s_{l+1}\cdots s_L}, \tag{8}$$

where the singular values are given by the non-negative diagonal matrix $\lambda^{[l]}$, and X and Y satisfy the orthogonal conditions $\mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \mathbf{I}$. Normalizing $\lambda^{[l]}$ by $\lambda^{[l]} \leftarrow \lambda^{[l]}/|\lambda^{[l]}|$, BEE is defined as

$$S^{[l]}_{\mathrm{BEE}} = -\sum_a \lambda^{[l]2}_{aa}\ln\lambda^{[l]2}_{aa}. \tag{9}$$

The computation of BEE in our context is illustrated in Fig. 2 (c). One only needs to transform the first $(l-1)$ tensors to the left orthogonal form (indicated by the arrows), then $\lambda^{[l]}$ is obtained by the SVD of $\mathbf{A}^{[l]}$ as $A^{[l]}_{s_l a_{l-1} a_l} = \sum_{aa'} X_{s_l a_{l-1}, a} \lambda^{[l]}_{aa'} Y_{a', a_l}$. The leading computational cost is about $O(ld\chi^3)$.

In Fig. 3 (and most of the paper), we take the MPS trained for classifying images "0" and "2" as an example, and show its SEE and BEE with and without the DCT. Without DCT, the data are in the real space, i.e., simply the pixels of the 2D images. The relatively large values of SEE are distributed almost all over the 2D plane. The BEE grows non-decreasingly as the position goes towards the label bond located at the left end of the zig-zag path. With DCT, the data are in fact weights of different frequencies. The large values of the SEE only appears in the positions that are close to the label bond. The BEE also changes in an non-decreasing way when the position approaches the label bond.

SEE actually characterizes the amount of non-trivial information carried by the data. Without DCT, the important information is distributed almost all over the 2D plane, while with DCT, the important information to the classification problem are mainly of low frequencies. This is consistent with what is know from the well-established image algorithms, that the low-frequency data are more important. With our work, such a phenomenon is naturally justified mathematically by the values of SEE of the trained MPS. Besides, we notice that with the real-space data, SEE is zero along the edges of about 4-pixel-width, corresponding to the blank edges of the MNIST images. This serves as another proof that SEE characterizes the importance of the data provided on different sites.

Meanwhile, the BEE with DCT increases in a much slower way than that without DCT. Due to the orthogonal conditions of the MPS, the information flows from the right end of the MPS to the left (label bond). Each time when the non-trivial information (indicated by a relatively large SEE) is passed through, BEE increases and finally saturates to a finite value $\ln D$. In the MPS schemes, it is well-known that the BEE determines the needed dimensions of the corresponding virtual bond. Particularly, when the entanglement entropy vanishes to zero, it means the corresponding data is uncorrelated to others and need not be fed to the MPS. In the following, we will show that to reach the same accuracy, smaller length of MPS, meaning smaller computational costs, are needed with DCT than without DCT. This provides an efficient scheme to discard the sites with small SEE.

C. Learning architecture based on single-site entanglement entropy

By minimizing the BEE, we propose to reorder the data so that the SEE is in a non-ascending order. The steps are listed in Box-I below. After reordering, the BEE will be lowered, meaning the computational cost will be lowered, while the accuracy remains unchanged.

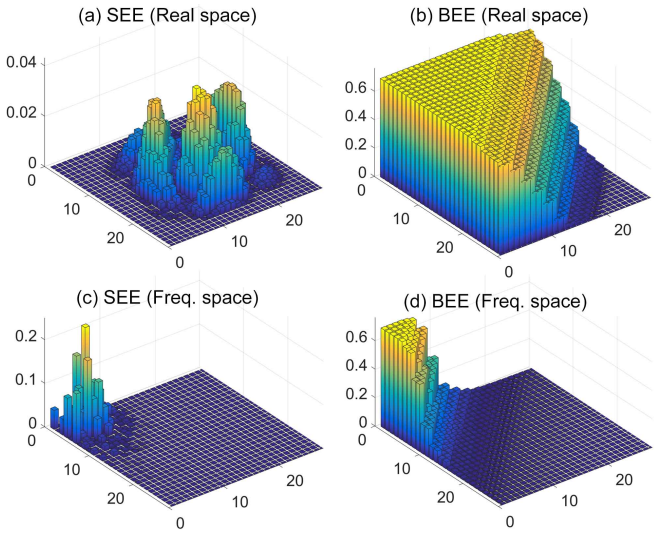

Figure 3. (a) Single-site entanglement entropy (SEE) and (b) bipartite entanglement entropy (BEE) of MPS without DCT. (c) and (d) show the SEE and BEE with DCT. We take the classification between the images "0" and "2" as an example. The virtual bond dimension is $\chi = 16$, with $D = 2, d = 2$.

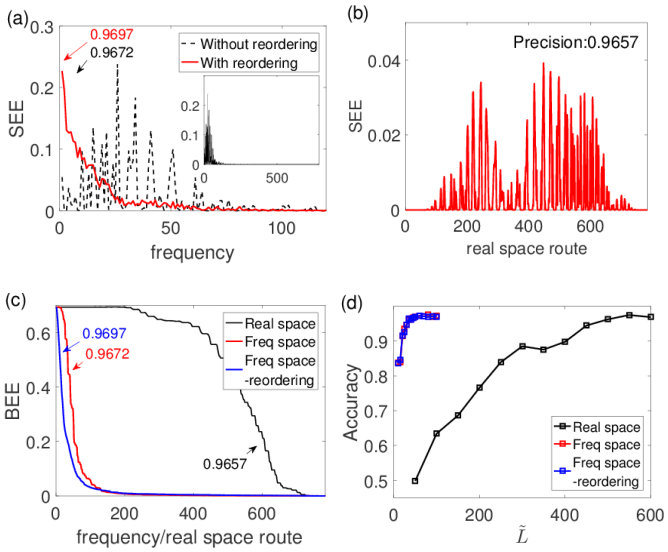

Figure 4. (a) SEE in frequency space without and with reordering according to SEE, (b) SEE in real space, (c) BEE in real space and frequency space, without and with reordering, and (d) accuracy on the test dataset for different MPS length $\tilde{L}$. The virtual bond dimension is $\chi = 16$, with $D = 2$ and $d = 2$.

Box. Steps of the training algorithm

1. Randomly initialize the MPS and train it by the standard algorithm; calculate the SEE of the trained MPS.
2. Reorder the data according to the values of SEE at different sites.
3. Randomly initialize the MPS and train it using the reordered data.
4. Calculate the SSE. If the SSE is in an acceptable descending order, end the training; if not, go back to Step 2.
5. Calculate the BEE and find the $\tilde{L}$-th site where BEE equals to $0.1 \ln D$. Cut the data at this site and train the new MPS with length of $\tilde{L}$.

To explain how the reordering works, let us give a simple example with a three-spin quantum state. The wave function reads $|\psi\rangle = |\uparrow\uparrow\downarrow\rangle + |\downarrow\uparrow\uparrow\rangle$, where $|\uparrow\rangle$ and $|\downarrow\rangle$ stand for the spin-up and spin-down states, respectively. By writing the wave function into a three-site MPS, one can easily check that the two virtual bonds are both two-dimensional. The total number of parameters of this MPS is $2^2 + 2^3 + 2^2 = 16$. However, if we swap the second spin to either end of the three-spin chain, say swapping it with the third spin, the wave function becomes $|\psi\rangle = |\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\rangle = (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle) \otimes |\uparrow\rangle$. Obviously, the virtual bonds of the MPS are two- and one-dimensional, respectively, and the total number of parameters is only $2^2 + 2^2 + 2 = 10$. Importantly, the SEE of the second site (before swapping) is zero. Normally, the SSE will be in a good descending order after reordering only once.

Fig. 4 (a) shows the SEE in the frequency space with and without reordering. Without reordering, the important data where the values of SSE are relatively large are distributed on the first 100 sites (see the inset of Fig. 4 (a)). By zooming in such a range, one can see that the SSE are in a good descending order by using the reordering trick. For comparison, by training with the real-space data, the sites with large SSE are distributed in almost the whole MPS, as shown in Fig.4 (b).

Fig. 4 (c) shows the BEE, which indicates the computational cost of using the MPS to solve the classification task. It is obvious that the BEE of the MPS trained by the frequency data is much smaller than that of the MPS trained by the real-space data. By reordering, the BEE is further reduced, indicating that smaller bond dimensions are needed.

Fig. 4 (d) shows the accuracy when discarding certain less important data. We only use the first $\tilde{L}$ data of each image to train the $\tilde{L}$-site MPS. We observe that as $\tilde{L}$ increases, the accuracy trained with the frequency data rises quickly and reach the value around 0.96 only with $\tilde{L} \simeq 50$. For comparison, the accuracy trained by the real-space data is obviously much worse. The difference between the accuracies with and without reordering is relatively small. This is because we take $\chi = 16$, where the maximal capacity of the entanglement entropy ($\ln \chi$) is much larger than its reduction after reordering.

To characterize the improvement of efficiency that can be gained by discarding the less important data, we define the *complexity ratio* $\xi = \frac{\tilde{L}}{L}$. $\tilde{L}$ is defined so that the BEE equals $c \ln D$ when measured at the $\tilde{L}$-th site. Note that the maximum of BEE is $O(\ln D)$. c is a small number determined by the requirement of accuracy. We take $c = 0.1$. When $\xi \ll 1$, it means the data on the last $(1-\xi)L$ sites can be ignored without harming the accuracy too much. Our results show that $\xi = 0.81$ when trained with real-space data, and $\xi = 0.11$ and 0.07 using frequency data with and without reordering, respectively. More results are given in the supplementary material, where we show that the trainings by the data with and without DCT lead to similar accuracy, but the efficiencies (characterized by the complexity ratios) are largely different.

IV. SUMMARY AND PROSPECTS

In this work, we explicitly show that quantum entanglement can be used for guiding the learning of data for image recognition. Specifically speaking, by training the unitary MPS, our numerical experiments demonstrate that the bipartite entanglement entropy indicates the efficiency of the training by classical computations. Meanwhile, the single-site entanglement entropy characterizes the importance of the data, with which a reordering technique is proposed to further improve the efficiency of the training. Our proposal can be readily applied to improve the efficiency of other schemes, such as tree TN. It can also be easily combined with other computational techniques besides DCT for preprocessing data, such as neural networks, to develop efficient training algorithms.

Furthermore, there are two advantages of our proposal from the viewpoint of quantum computations: (1) the MPS we train is formed by unitaries, which has good accuracy with relatively small bond dimensions; (2) our proposal permits to largely reduce the size of the MPS without harming the accuracy. Note that in principle, any local unitary mappings or gates can be realized by quantum simulators or computers. However, the complexity strongly depends on the dimensions and number of the gates. With our proposal, the low demands on the bond dimensions and, particularly, on the size, permit to simulate machine learning tasks by quantum simulations or quantum computations in the near future.

ACKNOWLEDGEMENT

Y.H.L thanks Naichao Hu for reading the manuscript. This work was supported by ERC AdG OSYRIS (ERC-

2013-AdG Grant No. 339106), the Spanish MINECO grants FISICATEAMO (FIS2016-79508-P) and "Severo Ochoa" Programme (SEV-2015-0522), Catalan AGAUR SGR 1341, Fundació Privada Cellex, and EU FETPRO QUIC (H2020-FETPROACT-2014 No. 641122). S.J.R. was supported by Fundació Catalunya - La Pedrera · Ignacio Cirac Program Chair. Y.H.L is supported by National Program For Top-notch Undergraduate in Basic Science under Grant No. 03100-31911002 from the Ministry of Education of P.R. China. X.Z. is supported by the National Natural Science Foundation of China (No. 11404413), the Natural Science Foundation of Guangdong Province (No. 2015A030313188), and the Guangdong Science and Technology Innovation Youth Talent Program (Grant No. 2016TQ03X688).

[1] M. Lewenstein, Journal of Modern Optics **41**, 2491 (1994).
[2] R. Chrisley, In New directions in cognitive science: Proceedings of the international symposium, Saariselka 4-9, 1995.
[3] S. Kak, Advances in Imaging and Electron Physics **94**, 259 (1995).
[4] M. Schuld, I. Sinayskiy, and F. Petruccione, Quantum Inf Process **13**, 2567 (2014).
[5] B. E. Baaquie, *Quantum finance: Path integrals and Hamiltonians for options and interest rates* (Cambridge University Press, 2007).
[6] J. Eisert, M. Wilkens, and M. Lewenstein, Phys. Rev. Lett. **83**, 30773080 (1999).
[7] N. F. Johnson, Phys. Rev. A **63**, 020302(R) (2001).
[8] J. Du, H. Li, X. Xu, M. Shi, J. Wu, X. Zhou, and R. Han, Physical Review Letters **88**, 137902 (2002).
[9] V. Dunjko, J. M. Taylor, and H. J. Briegel, Phys. Rev. Lett. **117**, 130501 (2016),
[10] V. Dunjko, Y.-K. Liu, X. Wu, and J. M. Taylor, arXiv preprint arXiv:1710.11160 (2017).
[11] L. Lucas, Scientific Reports **7** (2017).
[12] A. Monràs, G. Sent'ıs, and P. Wittek, Phys. Rev. Lett. **118**, 190503 (2017),
[13] A. Hallam, E. Grant, V. Stojevic, S. Severini, and A. G. Green, arXiv preprint arXiv:1711.03357 (2017).
[14] B.-L. Hao, H. C. Lee, and S,-Y, Zhang, Chaos Solitons Fractals **11**, 825 (2000).
[15] J. I. Latorre, Image compression and entanglement arXiv: quant-ph/0510031 (2005).
[16] P. Le, F. Dong and K. Hirota, Quantum Inf. Process **10** 6384(2011).
[17] J. Rodríguez-Laguna, P. Migdał, Miguels Ibáñez Berganza, M. Lewenstein, and G. Sierra, New Journal of Physics **14**, 053028 (2012).
[18] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).
[19] J. Carrasquilla and R. G. Melko, Nature Physics (2017).
[20] P. Broecker, F. F. Assaad, and S. Trebst, arXiv preprint arXiv:1707.00663 (2017).
[21] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Scientific reports **7**, 8823 (2017).
[22] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Physical Review X **7**, 031038 (2017).
[23] K. Ch'ng, N. Vazquez, and E. Khatami, arXiv preprint arXiv:1708.03350 (2017).
[24] Y. Zhang and E.-A. Kim, Phys. Rev. Lett. **118**, 216401 (2017),
[25] W. Hu, R. R. P. Singh, and R. T. Scalettar, arXiv preprint arXiv:1704.00080 (2017).
[26] L. Wang, Physical Review B **94**, 195105 (2016).
[27] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, arXiv preprint arXiv:1705.00565 (2017).
[28] T. Ohtsuki and T. Ohtsuki, Journal of the Physical Society of Japan **85**, 123706 (2016), https://doi.org/10.7566/JPSJ.85.123706,
[29] G. Carleo and M. Troyer, Science **355**, 602 (2017), http://science.sciencemag.org/content/355/6325/602.full.pdf,
[30] P. Huembeli, A. Dauphin, and P. Wittek, arXiv preprint arXiv:1710.08382 (2017).
[31] P. Palittapongarnpim, P. Wittek, and B. C. Sanders, in *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2016), pp. 327–332.
[32] Z. Cai and J. Liu, arXiv preprint arXiv:1704.05148 (2017).
[33] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Physical Review B **96**, 205152 (2017).
[34] J. I. Cirac and F. Verstraete, J. Phys. A: Math. Theor. **42**, 504004 (2009).
[35] R. Orús, Ann. Phys. **349**, 117 (2014).
[36] R. Orús, The European Physical Journal B **87**, 280 (2014),
[37] S.-J. Ran, E. Tirrito, C. Peng, X. Chen, G. Su, and M. Lewenstein, arXiv:1708.09213 (2017).
[38] E. Stoudenmire and D. J. Schwab, in *Advances in Neural Information Processing Systems* (2016), pp. 4799–4807.
[39] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, arXiv:1709.01662 (2017).
[40] D. Liu, S.-J. Ran, P. Wittek, C. Peng, R. B. Garc'ia, G. Su, and M. Lewenstein, arXiv preprint arXiv:1710.04833 (2017).
[41] E. M. Stoudenmire, arXiv preprint arXiv:1801.00315 (2017).
[42] V. Pestun and Y. Vlassopoulos, arXiv preprint arXiv:1710.10248 (2017).
[43] Y. Levine, D. Yakira, N. Cohen, and A. Shashua, CoRR, abs/1704.01552 (2017).
[44] F. Verstraete and J. I. Cirac, arXiv:0407066.
[45] Y. Y. Shi, L. M. Duan, and G. Vidal, Phys. Rev. A **74**, 022320 (2006).
[46] G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007).
[47] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, arXiv:1701.04831 (2017).
[48] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, arXiv preprint arXiv:1710.04045 (2017).
[49] Y. Huang and J. E. Moore, arXiv:1701.06246 (2017).

SUPPLEMENTARY MATERIAL A: SPEED-UP TRICKS IN THE TRAINING ALGORITHM

We introduce several tricks to speed up the training procedure. Firstly, we evolve the environment tensors $\mathbf{E}^{[l]}$ to avoid putting too many training samples in one single iteration. Specifically speaking, we only randomly select a small number of samples (say $O(10^2)$) and compute the corresponding environment tensor $\tilde{\mathbf{E}}$. Then we update $\mathbf{E}^{[l]} \leftarrow \mathbf{E}^{[l]} + \delta\tilde{\mathbf{E}}$ with δ a small constant. $\mathbf{E}^{[l]}$ is the total environment tensor and can be initialized as the $\tilde{\mathbf{E}}$ obtained in the first iteration. Then we use SVD of the total environment tensor $\mathbf{E}^{[l]} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$ to update the tensor as $\mathbf{A}^{[l]} \leftarrow \mathbf{V}\mathbf{U}^{\mathrm{T}}$. We found this harms little the accuracy but can largely save the computational time and memory.

Secondly, we restore all the intermediate vectors during the contraction process to avoid repetitive computations. This trades the computational time by memory, and do no harm to the accuracy.

Thirdly, we normalize the output vector after contracting with the MPS, i.e.,

$$u_b^{[n]} \leftarrow \frac{\sum_{s_1\cdots s_L} \hat{\Psi}_{b,s_1\cdots s_L} \prod_{l=1}^{L} v_{s_l}^{[n,l]}}{\sqrt{\sum_{b'} u_{b'}^{[n]2}}}. \tag{10}$$

The convergence is largely accelerated by keeping all intermediate vectors normalized. The normalization factors can connect to the cross entropy that is defined as

$$f^{CE} = \ln \mathrm{Tr}(\hat{\mathbf{\Psi}}\hat{\mathbf{\Psi}}^\dagger) - \frac{\sum_n \ln\Big(\sum_{bs_1\cdots s_L} B_b^{[n]} \hat{\Psi}_{b,s_1\cdots s_L} \prod_{l=1}^{L} v_{s_l}^{[n,l]}\Big)}{\mathrm{Tr}(\hat{\mathbf{\Psi}}\hat{\mathbf{\Psi}}^\dagger)}, \tag{11}$$

Considering $\mathrm{Tr}(\hat{\Psi}\hat{\Psi}^\dagger)$ as a constant according to the orthogonal condition, one has

$$\frac{\partial f^{CE}}{\partial \mathbf{A}^{[l]}} = \sum_n \frac{\mathbf{E}^{[l,n]}}{\sum_{bs_1\cdots s_L} B_b^{[n]} \hat{\Psi}_{b,s_1\cdots s_L} \prod_{l=1}^{L} v_{s_l}^{[n,l]}}, \tag{12}$$

with $\mathbf{E}^{[l,n]}$ the environment tensor for the n-th sample without normalization. The normalization factors naturally appear on the right-hand-side of the above equation.

More investigations are to be done to further understand the techniques explained above [Zheng-Zhi Sun *et al*, in preparation]. We shall stress that our proposal of entanglement-based architecture is independent on the tricks of optimizing the MPS or other TN's. Once the algorithm is chosen, our proposal can be utilized to reveal the "quantum" features of the machine learning tasks and improve the efficiency of the training.

SUPPLEMENTARY MATERIAL B: PRECISION OF THE TWO-CLASS CLASSIFIERS ON THE TEST DATASET

In Table I, we show the accuracy on the test dataset for all the two-class classifiers trained by the frequency data. We take physical bond dimension $d = 2$ and the virtual bond dimension $\chi = 16$. For comparison, the accuracy obtained from the real-space data is shown in Table II. In general, the accuracy from the frequency data is generally at the save level with that from the real-space data. This is expected since the DCT gives a unitary transformation on the data.

Although DCT does not change the data essentially, the efficiency of the training after DCT is largely improved, shown by the huge differences between the complex ratios ξ with and without DCT (see Table III). Note that the complex ratio indicates the number of data (ξL, with L the total number of, e.g., pixels), so that accuracy is kept unharmed. See the main text for more details.

–	1	2	3	4	5	6	7	8	9
0	0.9953	0.9627	0.9734	0.9842	0.9455	0.9510	0.9851	0.9632	0.9693
1	-	0.9377	0.9716	0.9844	0.9605	0.9775	0.9630	0.9597	0.9827
2	-	-	0.9324	0.9687	0.9506	0.9518	0.9534	0.9372	0.9667
3	-	-	-	0.9829	0.9106	0.9787	0.9671	0.8896	0.9663
4	-	-	-	-	0.9621	0.9691	0.9682	0.9642	0.8749
5	-	-	-	-	-	0.9514	0.9729	0.9228	0.9563
6	-	-	-	-	-	-	0.9884	0.9715	0.9853
7	-	-	-	-	-	-	-	0.9640	0.9146
8	-	-	-	-	-	-	-	-	0.9435

Table I. Precision of the two-class classifiers trained by frequency data.The virtual bond dimension is $\chi = 16$, with $D = 2$ and $d = 2$.

–	1	2	3	4	5	6	7	8	9
0	0.9943	0.9702	0.9769	0.9862	0.9407	0.9448	0.9846	0.9601	0.9703
1	-	0.9335	0.9758	0.9816	0.9556	0.9785	0.9653	0.9597	0.9813
2	-	-	0.9334	0.9697	0.9595	0.9553	0.9544	0.9392	0.9637
3	-	-	-	0.9859	0.8738	0.9817	0.9666	0.9012	0.9564
4	-	-	-	-	0.9589	0.9510	0.9677	0.9668	0.8709
5	-	-	-	-	-	0.9432	0.9750	0.9116	0.9642
6	-	-	-	-	-	-	0.9854	0.9643	0.9822
7	-	-	-	-	-	-	-	0.9610	0.9141
8	-	-	-	-	-	-	-	-	0.9425

Table II. Precision of the two-class classifiers trained by real-space data.The virtual bond dimension is $\chi = 16$, with $D = 2$ and $d = 2$.

–	1	2	3	4	5	6	7	8	9
0	0.07(0.83)	0.07(0.81)	0.08(0.79)	0.08(0.82)	0.08(0.82)	0.10(0.80)	0.10(0.84)	0.10(0.82)	0.09(0.84)
1	-	0.11(0.86)	0.11(0.84)	0.09(0.77)	0.10(0.82)	0.11(0.82)	0.12(0.80)	0.14(0.84)	0.12(0.81)
2	-	-	0.10(0.84)	0.08(0.86)	0.08(0.86)	0.12(0.88)	0.08(0.86)	0.11(0.86)	0.11(0.88)
3	-	-	-	0.11(0.82)	0.14(0.82)	0.10(0.81)	0.09(0.84)	0.12(0.82)	0.14(0.84)
4	-	-	-	-	0.10(0.81)	0.12(0.84)	0.10(0.81)	0.12(0.80)	0.16(0.82)
5	-	-	-	-	-	0.12(0.83)	0.10(0.84)	0.16(0.79)	0.15(0.85)
6	-	-	-	-	-	-	0.10(0.82)	0.12(0.84)	0.13(0.84)
7	-	-	-	-	-	-	-	0.11(0.84)	0.16(0.81)
8	-	-	-	-	-	-	-	-	0.15(0.81)

Table III. Complexity ratios ξ of classifiers trained by frequency data after reordering, and by the real-space data (shown in the bracket).