Letter

Scrambling ability of quantum neural network architectures

Yadong Wu,[1] Pengfei Zhang,[2,3,*] and Hui Zhai[1,†]

[1]*Institute for Advanced Study, Tsinghua University, Beijing 100084, China*

[2]*Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA*

[3]*Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, California 91125, USA*

(Received 30 November 2020; revised 1 March 2021; accepted 23 August 2021; published 3 September 2021)

In this Letter, we propose a guiding principle for how to design the architecture of a quantum neural network in order to achieve a high learning efficiency. This principle is inspired by the equivalence between extracting information from the input state to the readout qubit and scrambling information from the readout qubit to input qubits. We characterize the quantum information scrambling by operator size growth. By Haar random averaging over operator sizes, we propose an averaged operator size to describe the information scrambling ability of a given quantum neural network architecture. The key conjecture of this Letter is that this quantity is positively correlated with the learning efficiency of this architecture. To support this conjecture, we consider several different architectures, and we also consider two typical learning tasks. One is a regression task of a quantum problem, and the other is a classification task on classical images. In both cases, we find that, for the architecture with a larger averaged operator size, the loss function decreases faster or the prediction accuracy increases faster as the training epoch increases, which means higher learning efficiency. Our results can be generalized to more complicated quantum versions of machine learning algorithms.

DOI: 10.1103/PhysRevResearch.3.L032057

Classical neural networks can extract information from the input, usually a high-dimensional vector, and encode the information into a number or a low-dimensional vector as output. Classical neural networks have found broad applications in both technology developments and scientific research. For these applications, there are studies on how to design proper architectures of neural networks, such as the number of layers, the number of neurons in each layer, and the activation functions such that extracting information can be made most efficiently [1]. Quantum machine learning algorithms are considered one of the most promising applications in the near-term noisy intermediate-scale quantum technology and have attracted considerable attention recently [2–5], which include unsupervised learning, such as classification tasks [6–8], generative models [9,10], information extraction [11], and quantum generalization of neural networks, which include quantum state preparation [12–15], combination of quantum neural network and tensor network [16,17], learning optimization [18–20], and generalized quantum circuit from classical neural network [21–31]. Quantum neural networks (QNNs) also extract information from the input, usually a high-dimensional quantum wave function, and encode the information into one or a few read-out qubits. Usually, QNNs are made of local unitary quantum gates, and, in practice, we should face the same problem of how we design the architectures of the QNN properly.

To be concrete, we consider the QNN as shown in Fig. 1(a). The dataset is denoted by $\{(|\psi^d\rangle, y^d)\}$ (d labels data), where $|\psi^d\rangle$ is a quantum wave function and y^d is its label. The quantum circuit denoted by a unitary transformation $\hat{\mathcal{U}}$ is made of a number of local (say, two-qubit) quantum gates. There are various ways to construct $\hat{\mathcal{U}}$ with two-qubit gates, and different constructions correspond to different architectures. In the end, one measures the readout qubit r, say, by measuring $\hat{\sigma}_x^r$, and one can introduce the measurement operator $\hat{M}$ as

$$\hat{M} = \hat{\sigma}_0^1 \otimes \cdots \otimes \hat{\sigma}_x^r \otimes \cdots \hat{\sigma}_0^N, \tag{1}$$

where the superscript $i = 1, \ldots, N$ labels the qubits. Aside from the readout qubit r, no measurement is performed at other qubits, which are described by the identity matrix denoted by $\hat{\sigma}_0^i$. The measurement yields a readout

$$\tilde{y}^d = \langle \psi^d | \hat{\mathcal{U}}^\dagger \hat{M} \hat{\mathcal{U}} | \psi^d \rangle. \tag{2}$$

A loss function is designed to measure how close $\tilde{y}^d$ is to y^d, and one trains the parameters in the two-qubit gates to minimize the loss function. During training, the QNN can also make predictions on the dataset. Therefore, for a given task and dataset, and by averaging over different initializations, the loss or the accuracy as a function of training epoch mostly depends on the architecture of the QNN. The issue addressed in this Letter is whether there is a guiding principle for designing the most efficient architecture in learning, that is, as the training epoch increases, the decreasing of the loss or the increasing of the accuracy is the fastest.

*pengfeizhang.physics@gmail.com

†hzhai@tsinghua.edu.cn

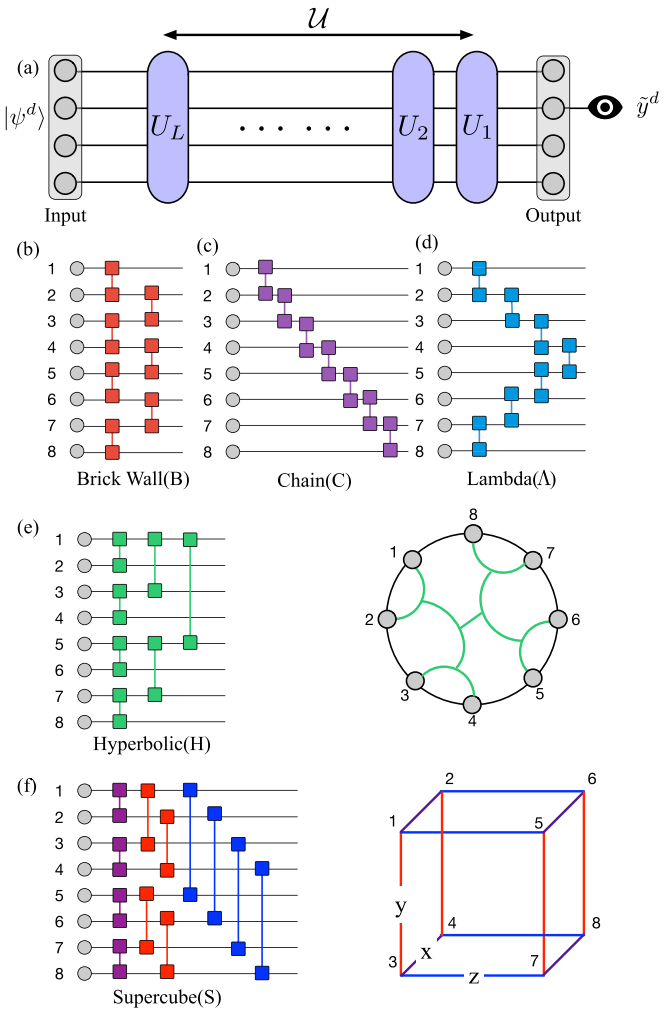

FIG. 1. (a) The global structure of the QNN. Each $\hat{U}_l$ is called a unit in this Letter, which is chosen among one of the building blocks shown in (b)–(f). (b)–(f) Various typical building blocks for constructing the QNN, which are respectively called the "brick wall"(B), "chain"(C), "lambda" (Λ), "hyperbolic"(H) and "supercube"(S) in this Letter.

In the cases of classical neural networks, there is always information loss from the input layer to the output layer. However, for the QNN, the total information is conserved during unitary transformation through the quantum circuit. Note that a unitary transformation is reversible, when we say a QNN encodes the information from the input wave function to the readout qubit, it is equivalent to say that the QNN scrambles the information from the readout qubit to all input qubits. Thus, the efficiency of extracting information is equivalent to the efficiency of scrambling information. Lots of studies in the past few years have established several related quantities to characterize quantum information scrambling, such as the out-of-time-ordered correlator [32–37], the tripartite information [38–41], and the operator size growth [42–52]. Recently, the tripartite information has also been used to reveal universal features in the training dynamics of the QNN [53]. This Letter focuses on the architecture and the main results are two folds:

(i) We propose a quantity based on the operator size to characterize the information scrambling ability of a QNN architecture.

(ii) We show that the scrambling ability quantified in this way is positively correlated with the learning efficiency of the QNN architecture.

Architectures. We demonstrate our results using several different architectures shown in Figs. 1(b)–1(f) as examples. The entire quantum circuit $\hat{\mathcal{U}}$ is made of a number of units, i.e., $\hat{\mathcal{U}} = \hat{U}_1\hat{U}_2\cdots\hat{U}_L$, as shown in Fig. 1(a). Each unit $\hat{U}_l$ contains a number of two-qubit gates $\hat{u}_{ij}$, and we require that each qubit is operated, at least, once. $\hat{u}_{ij}$ denotes a two-qubit gate acting on qubits i and j. For universal quantum computing, each $\hat{u}_{ij}$ is parametrized as

$$\hat{u}_{ij} = \exp\left(\sum_k \alpha_{ij}^k \hat{g}_k\right), \tag{3}$$

where $\hat{g}_k$ are SU(4) generators and α_{ij}^k are parameters. In a QNN, these parameters need to be determined by training. How to arrange these $\hat{u}_{ij}$ to form $\hat{U}_l$, and then to form $\hat{\mathcal{U}}$, is referred to as the architecture here.

Figures 1(b)–1(f) show architectures considered in this Letter. For cases shown in Figs. 1(b)–1(d), all qubits are aligned along a one-dimensional line and all gates operator on two neighboring qubits. They differ by the ordering of these gates, and they are called brick wall (B), chain (C), and lambda (Λ) as what they look like. For the case shown in Fig. 1(e), all qubits sit in a one-dimensional circle, and the way they interact is reminiscent of the hyperbolic geometry for which it is called hyperbolic (H). Finally, for the case shown in Fig. 1(f), qubits sit at the corners of a three-dimensional cube. The two-qubit gates first act on four pairs of neighboring gates along x, and then four pairs of neighboring gates along y and finally four pairs of neighboring gates along z. Below we explicitly show the scrambling ability and its correlation with learning ability using these architectures, however, we emphasize that we have tried more generic architectures and our conclusions below hold for general architectures [54]. We note in certain systems, all-to-all interactions are realized. However, this is because the intermediate degree of freedoms mediating the interactions are integrated out, which can be viewed as a specific architecture with local gates.

Operator size. Now we briefly introduce the operator size [42–52]. Let us consider a system with N qubit and an operator $\hat{O}$ in this system. Generally, we can expand the operator as

$$\hat{O} = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \hat{\sigma}_{\alpha_1}^1 \otimes \hat{\sigma}_{\alpha_2}^2 \cdots \otimes \hat{\sigma}_{\alpha_N}^N, \tag{4}$$

where $\hat{\sigma}_{\alpha_i}^i$ with subscript $\alpha_i = 0$–3, respectively, denotes identity ($\alpha_i = 0$) and three Pauli matrices $\hat{\sigma}_{x,y,z}$. Here $\boldsymbol{\alpha}$ denotes a set $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, and we use $l(\boldsymbol{\alpha})$ to denote the number of nonzero elements in the set $\boldsymbol{\alpha}$, i.e., the number of operators in $\hat{\sigma}_{\alpha_1}^1 \otimes \hat{\sigma}_{\alpha_2}^2 \cdots \otimes \hat{\sigma}_{\alpha_N}^N$ that are not identity. Then, the size of an operator is defined as

$$\text{size}(\hat{O}) = \sum_{\boldsymbol{\alpha}} |c_{\boldsymbol{\alpha}}|^2 l(\boldsymbol{\alpha}). \tag{5}$$

In the most general case, there are totally 4^N terms in the expansion Eq. (4). Here we give two examples. If we consider the measurement operator $\hat{M}$ defined in Eq. (1), we have $\text{size}(\hat{O}) = 1$. If we consider a uniform distribution among all $4^N - 1$ traceless operators with $|c_{\boldsymbol{\alpha}}|^2 = 1/(4^N - 1)$, then

$$\text{size}(\hat{O}) = \frac{1}{4^N - 1} \sum_{n=1}^{N} \frac{N!}{n!(N-n)!} 3^n n \approx \frac{3N}{4}. \tag{6}$$

Furthermore, if we consider the situation that, among N qubits, operators on a fraction of αN qubits ($\alpha < 1$) are uniformly distributed among $\hat{\sigma}_{0,\cdots,4}$ and operators on the rest of the $(1-\alpha)N$ qubits are always identity. Then, the operator size is reduced to $3\alpha N/4$.

Now we present an argument to bring out the connection between the operator size and the learning ability of a QNN. Let us consider the operator $\hat{M}' = \hat{\mathcal{U}}^\dagger \hat{M} \hat{\mathcal{U}}$ in Eq. (2). Initially, the $\hat{M}$ operator is not identity only at the measurement qubit r, however, because $\hat{\mathcal{U}}$ does not commute with $\hat{M}$, $\hat{M}'$ can also take one of the three Pauli matrices on other qubits and the operator size increases. When the operator $\hat{\mathcal{U}}$ becomes more and more complicated as the depth of the QNN increases, the operator size of $\hat{M}'$ increases. However, if $\text{size}(\hat{M}')$ is not sufficiently large, there is still a large probability that $\hat{M}'$ takes the identity operator on some qubits. Since $\hat{M}'$ acts on the input state, and if the operator $\hat{M}'$ is nearly identity on some qubits, the QNN can hardly extract information from the input wave function at those qubits. Therefore, a necessary condition for accurate learning is that $\text{size}(\hat{M}')$ reaches a sufficiently large value.

$\text{Size}(\hat{M}')$ depends on both the architecture and the parameters of the unitary $\hat{\mathcal{U}}$. Since for a QNN, the parameters keep updating during training but the architecture is fixed as a prior, we would like to have a quantity that only depends on the architecture. To this end, we propose to consider an averaged operator size,

$$\overline{\text{size}} = \int d\hat{\mathcal{U}}\, \text{size}(\hat{\mathcal{U}}^\dagger \hat{M} \hat{\mathcal{U}}). \tag{7}$$

Here $\int d\hat{\mathcal{U}}$ means the Haar random average over all two-qubit gates in $\hat{\mathcal{U}}$. Since the parameters in $\hat{\mathcal{U}}$ have been averaged over, $\overline{\text{size}}$ defined by Eq. (7) only depends on the architecture. This quantity characterizes that for generic parameters, how fast the operator size grows in a given QNN architecture. We propose to use this parameter to quantify the ability of scrambling quantum information for a given architecture. We argue that for an architecture with larger $\overline{\text{size}}$, it is easier to reach a suitable parameter such that $\text{size}(\hat{\mathcal{U}}^\dagger \hat{M} \hat{\mathcal{U}})$ is large enough that ensures efficient information extraction from the input wave functions.

The Haar random average can also simplify the calculation of the operator size $\overline{\text{size}}$. For instance, let us consider a two-qubit system and an operator $\hat{\sigma}_x \otimes \hat{\sigma}_0$. Expanding $\hat{\mathcal{U}}^\dagger \hat{\sigma}_x \otimes \hat{\sigma}_0 \hat{\mathcal{U}}$ as Eq. (4), and after averaging over the Haar random unitary, the weight $c_{\boldsymbol{\alpha}}$ [$\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$] reads [42]

$$\overline{|c_{\boldsymbol{\alpha}}|^2} = \frac{1 - \delta_{\alpha_1 0}\delta_{\alpha_2 0}}{15}. \tag{8}$$

Consequently, the probability of having a nonidentity operator only on the first or only on the second site is 1/5, and the probability of having nonidentity operators on both sites is 3/5. Based on Eq. (8), for any QNN with $\hat{\mathcal{U}}$ composited by two-qubit gates, the operator size growth can be explicitly deduced as the depth of the QNN increases.

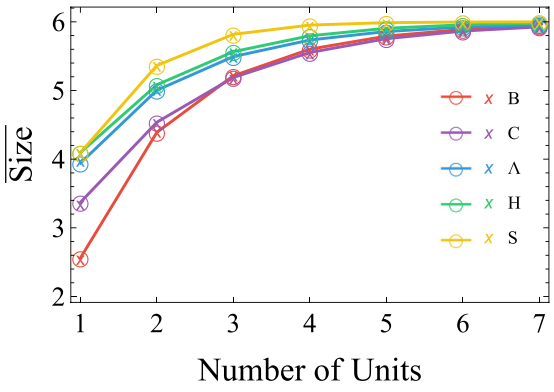

FIG. 2. The Haar-random-averaged operator size $\overline{\text{size}}$ defined in Eq. (7) for different architectures. For each architecture, all units share the same structure chosen as one of the cases shown in Figs. 1(b)–1(f) and labeled by the same label introduced in Figs. 1(b)–1(f). The horizontal axis is the number of units. Given the number of units, the numbers of two-qubit gates are the same for different cases under comparison [55]. Cross markers with different labels are obtained by numerical simulations of sampling 10^3 different parameters, and the solid lines with empty circles are obtained by the analytical formula assuming infinite sampling. Here we have taken the number of qubits $N = 8$.

We compute $\overline{\text{size}}$ defined in Eq. (7) for different architectures shown in Fig. 1 and the results are shown in Fig. 2. The results show the ordering of $\overline{\text{size}}$ as $(S) > (H) \approx (\Lambda) > (C) \approx (B)$. Especially, it is clear that the supercube (S) performs obviously better than others. And the difference between different architectures is the most significant for an intermediate QNN depth. When the number of units is too small (e.g., 2) and the QNN is too shallow, the unitary simple enough that a local operator is not sufficiently scrambled for all architectures. On the other hand, when the number of units is large enough (e.g., ~7) and the QNN is deep enough, the unitary is sufficiently complicated for all architectures and $\overline{\text{size}}$ for all cases approach $3N/4$ (=6 for $N = 8$ considered here) [54], and their differences also become insignificant. What is more, we also consider the QNN made of three-qubit gates, and the order of averaged operator size of different architectures is similar to the results of the two-qubit case [54].

Learning efficiency. To confirm the relation between the learning efficiency and the scrambling ability defined above, we consider two typical training tasks. The first is a regression task of information recovering in a quantum system. Let us consider an unknown initial product state $|\phi^d\rangle$, its total magnetization is given by

$$M_z^d = \frac{1}{N} \langle \phi^d | \sum_{i=1}^{N} \hat{\sigma}_z^i | \phi^d \rangle. \tag{9}$$

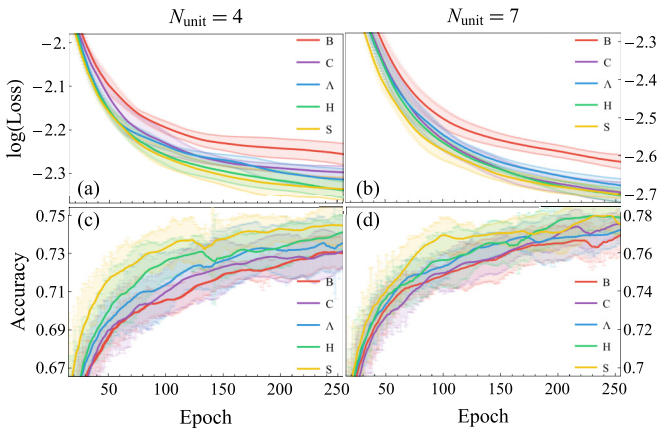

FIG. 3. Performance of different architectures for two different tasks. (a) and (b) Loss as a function of a training epoch for the information recovering task on the quantum spin problem. (c) and (d) Prediction accuracy as a function of training epoch for the second classification problem of red-green-blue (RGB) images of numbers. The number of units $N_{\text{unit}} = 4$ for (a) and (c) and $N_{\text{unit}} = 7$ for (b) and (d). In both cases, the results have been averaged over ten different initializations. The shaded area indicates the standard deviation of averaging over ten different initializations.

Now let us consider a chaotic Hamiltonian,

$$\hat{H} = \sum_{\langle ij \rangle} \sum_{\alpha=x,y,z} J_\alpha^{ij} \hat{\sigma}_\alpha^i \hat{\sigma}_\alpha^j + \sum_i \sum_{\alpha=x,y,z} h_\alpha^i \hat{\sigma}_\alpha^i, \tag{10}$$

where J_α^{ij} and h_α^i are a set of randomly chosen parameters. We evolve $|\phi^d\rangle$ with this Hamiltonian for a sufficiently long time to ensure a chaotic unitary dynamics, which yields $|\psi^d\rangle = e^{-i\hat{H}t}|\phi^d\rangle$. For the QNN, the training dataset is taken as $\{\{|\psi^d\rangle, y^d\}, d = 1, \ldots, N_{\text{D}}\}$, where y^d is taken as the total magnetization $y^d = M_z^d$ and N_{D} is the number of datasets. The loss function is taken as

$$\mathcal{L} = \frac{1}{N_{\text{D}}} \sum_{d=1}^{N_{\text{D}}} |\tilde{y}^d - y^d|, \tag{11}$$

where $\tilde{y}^d$ is the readout of the QNN given by Eq. (2) with input $|\psi^d\rangle$. In Figs. 3(a) and 3(b) we show how the loss function decreases as the training epoch increases. The trained QNN supposedly can recover the magnetization information of the initial state from the final state after a chaotic evolution.

The second task is a classification task of recognizing classical images. We take large numbers of RGB images with either a number 6 or a number 9 embedded in the background. Each image contains $16 \times 16 = 256$ pixels. Considering a system with $N = 8$ qubits, there are totally $2^8 = 256$ bases in the Hilbert space. A general wave function can be expanded in terms of these 256 bases. Each pixel corresponds to a base, and the information of each pixel is encoded into the coefficient of its corresponding base [54]. In this way, for each image, we generate a wave function $|\psi^d\rangle$ as input. The label is taken as $y^d = 0$ if the image contains the number 6 and $y^d = 1$ if the image contains the number 9. The readout of the QNN $\tilde{y}^d$ is also given by Eq. (2) with the input $|\psi^d\rangle$. In this case, the loss function is taken as the cross entropy between y^d and p^d, and since $\tilde{y}^d$ lies between $[-1, 1]$, we define p^d as $(1 + \tilde{y}^d)/2$ such that it lies in the range of [0,1]. Then the loss function is given by

$$\mathcal{L} = \frac{1}{N_{\text{D}}} \sum_{d=1}^{N_{\text{D}}} [-y^d \ln \; p^d - (1 - y^d) \ln(1 - p^d)]. \tag{12}$$

After learning, we let the QNN to make predictions on the dataset $\{\{|\psi^d\rangle, y^d\}, d = 1, \ldots, N_D\}$. For each input $|\psi^d\rangle$, a trained QNN returns a prediction $\tilde{y}^d$ given by Eq. (2). Now we interpret the prediction as the number 9 with $p^d = 1$ if $\tilde{y}^d > 0$, and as the number 6 with $p^d = 0$ if $\tilde{y}^d < 0$. Then, we can obtain an accuracy as

$$\frac{1}{N_{\text{D}}} \sum_{d=1}^{N_{\text{D}}} |p^d - y^d|. \tag{13}$$

In Figs. 3(c) and 3(d) we also show how the accuracy increases as the training epoch increases.

The results shown in Fig. 3 have been averaged over a few runs with different initializations, and, therefore, their differences mainly reflect the differences in learning efficiency between different architectures. In Figs. 3(a) and 3(b), we show that in the first task for most training epochs the loss function is ordered as $(S) < (H) \lesssim (\Lambda) \lesssim (C) < (B)$. In Figs. 3(c) and 3(d), we show that in the second task for most training epochs the accuracy is ordered as $(S) > (H) \gtrsim (\Lambda) > (C) \gtrsim (B)$. Both orders are consistent with the order of size defined for different architectures. This means that for a fixed target loss value or prediction accuracy, the architecture with the largest size can reach this target with the smallest training epoch. In this sense, we consider this architecture as the most efficient one. Therefore, these examples support our argument of the positive correlation between scrambling ability and learning efficiency. Such a positive correlation also holds when noises are introduced into the quantum circuits [54].

We also note that this correlation is most pronounced for intermediate training epochs and for intermediate depths of the QNN. This is because size quantifies the scrambling ability of architectures with generic parameters, but for sufficiently long training, the QNN can always reach the optimal parameters. Also, for the sufficiently deep QNN, all architectures with generic parameters can always lead to the most scrambled operators, whose size reaches the saturation value, as one can see from Fig. 2. Therefore, their differences in learning efficiency also become less significant as one can see by comparing Figs. 3(b) and 3(d) with 3(a) and 3(c).

Outlook. To the best of our knowledge, this Letter is an attempt to understand how to design the most efficient architectures in the QNN. Our design principle is based on quantum information scrambling in a quantum circuit, described by the operator size growth. We propose a quantity to quantify the scrambling ability of a QNN architecture, which is based on how fast the size of a local operator grows under generic unitary transformations generated by the quantum circuit. We conjecture the positive correlation between this quantity and the learning ability of the QNN, and the conjecture is confirmed by two typical learning tasks. Our discussion is so far limited to the quantum version of fully connected neural networks, and in the future, it can be generalized to other

quantum versions of neural networks, such as quantum convolutional neural networks [24–26], quantum recurrent neural networks [27,28], and quantum autoencoders [29–31].

Acknowledgment. This work was supported by the Beijing Outstanding Young Scientist Program, NSFC Grant No. 11734010, MOST under Grant No. 2016YFA0301600.

[1] I. Goodfellow, Y. Bengio, and A. Couville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).
[2] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).
[4] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa, E. Peters, O. Lockwood, A. Skolik, S. Jerbi, V. Dunjko, M. Leib, M. Streif, D. V. Dollen, H. Chen, S. Cao, R. Wiersema, H.-Y. Huang, J. R. McClean, R. Babbush, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, TensorFlow quantum: A software framework for quantum machine learning, arXiv:2003.02989.
[5] L. Lamata, Quantum machine learning and quantum biomimetics: A perspective, Mach. Learn.: Sci. Technol. **1**, 033002 (2020)
[6] E. Farhi, and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.
[7] M. Schuld, A. Bocharov, Krysta M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).
[8] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, Hierarchical quantum classifiers, npj Quantum Inf. **4**, 65 (2018).
[9] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit Born machines, Phys. Rev. A **98**, 062324 (2018).
[10] J. Zeng, Y. Wu, Jin-G. Liu, L. Wang, and J. Hu, Learning and inference on generative adversarial quantum circuits, Phys. Rev. A **99**, 052306 (2019).
[11] X. Liang, Y. Wu and H. Zhai, The quantum cocktail party problem, Sci. China: Phys., Mech. Astron., **63**, 250362 (2020).
[12] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. PerdomoOrtiz, A generative modeling approach for benchmarking and training shallow quantum circuits, npj Quantum Inf. **5**, 45 (2019).
[13] M. J. S. Beach, R. G. Melko, T. Grover, and T. H. Hsieh, Making trotters sprint: A variational imaginary time ansatz for quantum many-body systems, Phys. Rev. B **100**, 094434 (2019).
[14] S. Ghosh, T. Paterek, and Timothy C. H. Liew, Quantum Neuromorphic Platform for Quantum State Preparation, Phys. Rev. Lett **123**, 260404 (2019).
[15] T. Krisnanda, S. Ghosh, T. Peterek, and Timothy C. H. Liew, Creating and concentrating quantum resource states in noisy environments using a quantum neural network, Neural Networks **136**, 141-151 (2021).
[16] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Phys. Rev. Research **2**, 033125 (2020).
[17] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, Quantum Sci. Technol. **4**, 024001 (2019).
[18] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, and R. Wolf, Efficient Learning for Deep Quantum Neural Networks, Nat. Commun. **11**, 808 (2020).
[19] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models, Phys. Rev. X **7**, 041052 (2017).
[20] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).
[21] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).
[22] E. Torrontegui, and J. J. García-Ripoll, Unitary quantum perceptron as efficient universal approximator, Europhys. Lett. **125**, 30004 (2019).
[23] S. Ghosh, T. Krisnanda, T. Peterek, and T. C. H. Liew, Realising and compressing quantum circuits with quantum reservoir computing, Commun. Phys. **4**, 105 (2021).
[24] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).
[25] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, Quanvolutional neural networks: powering image recognition with quantum circuits, arXiv:1904.04767.
[26] S. Oh, J. Choi, and J. Kim, A tutorial on quantum convolutional neural networks (QCNN), arXiv:2009.09423.
[27] J. Bausch, Recurrent quantum neural networks, *Advances in Neural Information Processing Systems 33* (NeurIPS 2020).
[28] L. Behera, I. Kar, and A. C. Elitzur, Recurrent quantum neural network and its applications, *The Emerging Physics of Consciousness*, edited by J. A. Tuszynski (The Frontiers Collection, Springer, Berlin, Heidelberg, 2006).
[29] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. **2**, 045001 (2017).
[30] D. Bondarenko, and P. Feldmann, Quantum Autoencoders to Denoise Quantum Data, Phys. Rev. Lett. **124**, 130502 (2020).
[31] A. Pepper, N. Tischler, and G. J. Pryde, Experimental Realization of a Quantum Autoencoder: The Compression of Qutrits via Machine Learning, Phys. Rev. Lett. **122**, 060501 (2019).
[32] S. H. Shenker, and D. Stanford, Black holes and the butterfly effect, J. High Energy Phys. 03 (2014) 067.
[33] D. A. Roberts, D. Stanford, and L. Susskind, Localized shocks, J. High Energy Phys. 03 (2015) 051.
[34] D. A. Roberts, and D. Stanford, Diagnosing Chaos Using Four-Point Functions in Two-Dimensional Conformal Field Theory, Phys. Rev. Lett. **115**, 131603 (2015).
[35] J. Maldacena, S. H. Shenker, and D. Stanford, A bound chaos, J. High Energy Phys. 08 (2016) 106.
[36] P. Hosur, X.-L. Qi, and D. Roberts, Chaos in quantum channels, J. High Energy Phys. 02 (2016) 004.
[37] R. Fan, P. Zhang, Huitao Shen and H. Zhai, Out-of-Time-Order correlation for many-body localization, Sci. Bulletin **62**, 707 (2017).

[38] M. Rangamani, and M. Rota, Entanglement structures in qubit systems, J. Phys. A: Math. Theor. **48**, 385301 (2015).

[39] A. Kitaev, and J. Preskill, Topological Entanglement Entropy, Phys. Rev. Lett. **96**, 110404 (2006).

[40] M. Levin, and X.-G. Wen, Detecting Topological Order in a Ground State Wave Function, Phys. Rev. Lett. **96**, 110405 (2006).

[41] C. Sünderhauf, L. Piroli, X.-L. Qi, N. Schuch, and J. I. Cirac, Quantum chaos in the Brownian SYK model with large finite N : OTOCs and tripartite information, J. High Energy Phys. 11 (2019) 038.

[42] D. A. Roberts, D. Stanford, and A. Streicher, Operator growth in the SYK model, J. High Energy Phys. 06 (2018) 122.

[43] A. Nahum, S. Vijay, and J. Haah, Operator Spreading in Random Unitary Circuits, Phys. Rev. X **8**, 021014 (2018).

[44] C. W. von Keyserlingk, T. Rakovszky, F. Pollmann, and S. L. Sondhi, Operator Hydrodynamics, OTOCs, and Entanglement Growth in Systems without Conservation Laws, Phys. Rev. X **8**, 021013 (2018).

[45] V. Khemani, A. Vishwanath, and D. A. Huse, Operator Spreading and the Emergence of Dissipative Hydrodynamics under Unitary Evolution with Conservation Laws, Phys. Rev. X **8**, 031057 (2018).

[46] T. Rakovszky, F. Pollmann, and C. W. V. Keyserlingk, Diffusive Hydrodynamics of Out-of-Time-Ordered Correlators with Charge Conservation, Phys. Rev. X **8**, 031058 (2018).

[47] X. Chen, and T. Zhou, Operator scrambling and quantum chaos, arXiv:1804.08655.

[48] A. Lucas, Operator Size at Finite Temperature and Planckian Bounds on Quantum Dynamics, Phys. Rev. Lett. **122**, 216601 (2019).

[49] S. Xu, and B. Swingle, Locality, Quantum Fluctuations, and Scrambling, Phys. Rev. X **9**, 031048 (2019).

[50] X.-L. Qi, and A. Streicher, Quantum epidemiology: operator growth, thermal effects and SYK, J. High Energy Phys. 08 (2019) 012.

[51] S. Xu, and B. Swingle, Accessing scrambling using matrix product operators, Nat. Phys. **16**, 199 (2020).

[52] Y. D. Lensky, X.-L. Qi, and P. Zhang, Size of bulk fermions in the SYK model, J. High Energy Phys. 10 (2020) 053.

[53] H. Shen, P. Zhang, Y.-Z. You, and H. Zhai, Information Scrambling in Quantum Neural Networks, Phys. Rev. Lett. **124**, 200504 (2020).

[54] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.3.L032057 for details of training data encoding, general architectures, saturation of operator size, noise against, and training process.

[55] For supercube circuits (S), we set the last unit $\hat{U}_1$ the same as the hyperbolic unit Fig. 1(e). Other units $\hat{U}_{l\geqslant 2}$ are chosen such that the circuit $\hat{U}_L \cdots \hat{U}_2$ repeats the structure of Fig. 1(f) and each unit contains seven two-qubit gates. Such a structure eliminates redundant gates in U_1 and is found to be optimal for the operator size growth.