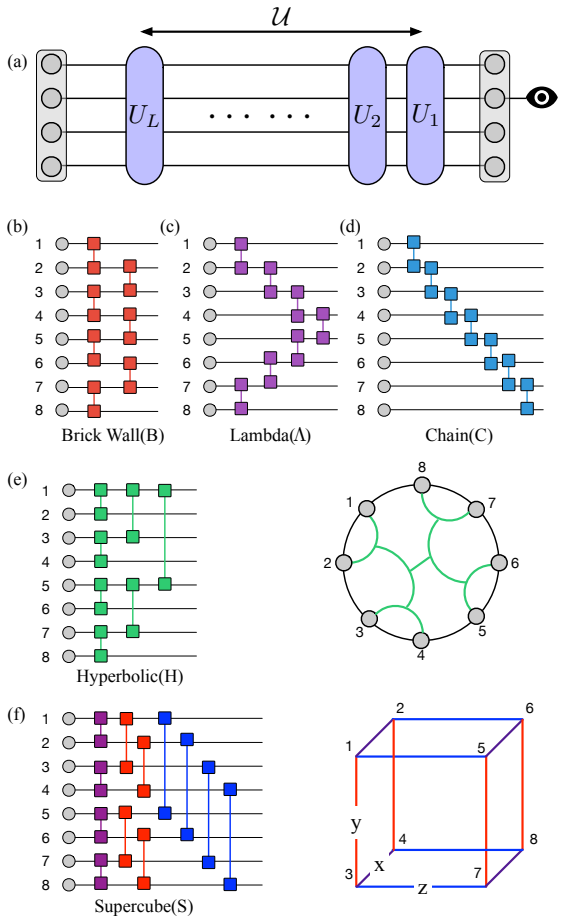

FIG. 1: (a) The global structure of the QNN. Each $\hat{U}_l$ is called a unit in this work, which is chosen among one of the building blocks shown in (b-f). (b-f) Various typical building blocks for constructing the QNN, which are respectively called the "brick wall"(B), "Lambda" (Λ), "Chain"(C), "Hyperbolic"(H) and "Supercube"(S) in this work.

tains a number of two-qubit gates $\hat{u}_{ij}$, and it ensures that every qubit is operated at least once. $\hat{u}_{ij}$ denotes a two-qubit gate acting on qubit-i and -j. Each $\hat{u}_{ij}$ is parameterized as

$$\hat{u}_{ij} = e^{\sum_k \alpha_{ij}^k \hat{g}_k}, \tag{3}$$

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.

Fig. 1(b-f) show architectures considered in this work. For cases shown in Fig. 1(b-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), Lambda (Λ), and Chain (C) 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.

Operator Size. Now we briefly introduce the operator size [29–39]. 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 $\alpha_i = 0, 1, 2, 3$ 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 non-zero 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-qubit, operators on a fraction of αN qubits ($\alpha < 1$) are uniformly distributed among $\hat{\sigma}_{0,\ldots,4}$ and operators on the rest $(1-\alpha)N$ qubits are always identity. Then, the operator size is reduced to $3\alpha N/4$.

We present an argument to bring out the connection between 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, $\hat{M}$ 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 be one of the three Pauli matrices on other qubits and the operator size increases. Generally, when the operator $\hat{\mathcal{U}}$ becomes more and more complicated as the depth of 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 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 sufficient 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

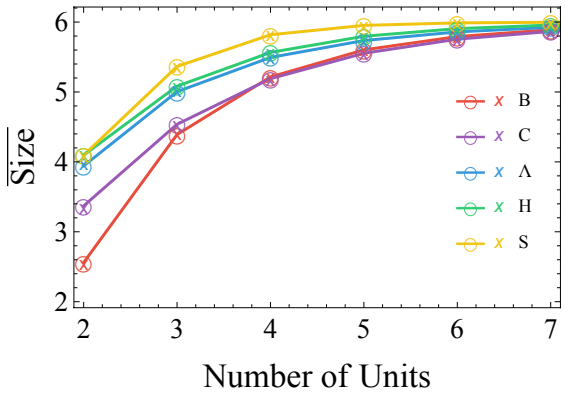

FIG. 2: The Haar-random-averaged operator size $\overline{\text{Size}}$ defined in Eq. 7 for different architectures. For each architecture, all units share the same structure chosen as one of the cases shown in Fig. 1(b-e) with (f) a little different [50] and labeled by the same label introduced in Fig. 1(b-f). The horizontal axis is the number of units. Cross markers with different labels are obtained by numerical simulations and the solid lines with empty circles are obtained with analytical formula. Here we have taken the number of qubits $N = 8$.

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 Haar random average overall 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 quantifies 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 to scramble 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. For instance, let us consider a two-qubits 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 [29]

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

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

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) ≈ (Λ) > (C) ≈ (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 intermediate QNN depth. When the number of units is too small (e.g. ~ 2) and the QNN is too shallow, the unitary is not complicated enough that a local operator cannot be 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), and their differences also become insignificant.

Learning Efficiency. To relate the learning efficiency to 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

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

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 sufficient long time to ensure a chaotic unitary dynamics, which yields $|\psi^d\rangle = e^{-i\hat{H}t}|\phi^d\rangle$. For 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 dataset. 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 QNN given by Eq. 2 with input $|\psi^d\rangle$. In Fig. 3(a-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 [41]. 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,

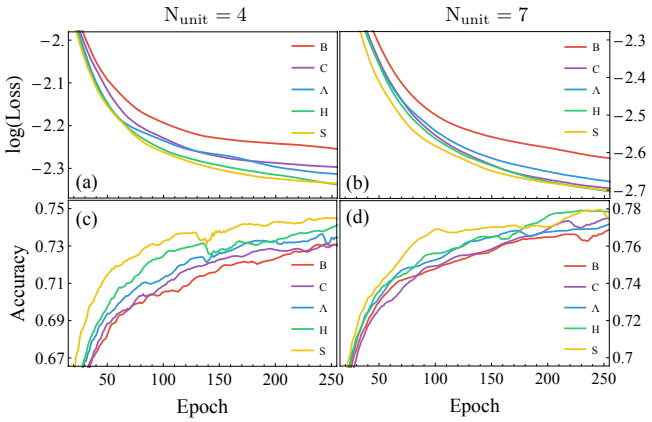

FIG. 3: Performance of different architectures for two different tasks. (a-b) Loss as a function of training epoch for the information recovering task on the quantum spin problem. (c-d) Prediction accuracy as a function of training epoch for the second classification problem of RGB images of numbers. 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 10 different initializations.

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

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

After leaning, we let the QNN to make predictions on a testing dataset $\{\{|\psi^d\rangle, y^d\}, d = 1, \ldots, N_{\text{test}}\}$. 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 Fig. 3(c-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 Fig. 3(a-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 Fig. 3(c-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 $\overline{\text{Size}}$ defined for different architectures. This means that for a fixed target loss value or prediction accuracy, the architecture with the largest $\overline{\text{Size}}$ 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 intuition of the positive correlation between scrambling ability and learning efficiency.

We also note that this correlation is most pronounced for intermediate training epochs and for intermediate depths of the QNN. This is because $\overline{\text{Size}}$ quantifies the scrambling ability of architectures with generic parameters, but for sufficiently long training, the QNN can always reach the optimal parameters. Also, for 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, the differences in learning efficiency also become less significant, as one can see by comparing Fig. 3(b)(d) with (a)(c).

Outlook. To the best of our knowledge, this work is the first attempt to understand how to design the most efficient architectures in 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 [42–44], quantum recurrent neural networks [45, 46], and quantum autoencoders [47–49].

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

* Electronic address: pengfeizhang.physics@gmail.com
† Electronic address: hzhai@tsinghua.edu.cn

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Couville, *Deep Learning* (2016).
[2] John Preskill, "Quantum computing in the NISQ era and beyond," Quantum **2**, 79 (2018).
[3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, Q"uantum machine learning," Nature **549**, 195–202 (2017).
[4] Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz, "Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models," Phys. Rev. X **7**, 041052 (2017).
[5] E Torrontegui, and JJ García-Ripoll, "Unitary quantum perceptron as efficient universal approximator," EPL (Europhysics Letters) **125**, 30004 (2019).
[6] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro PerdomoOrtiz, "A generative modeling approach for benchmarking and training shallow quantum circuits," npj Quantum Inf. **5**, 45 (2019).
[7] Edward Farhi, and Hartmut Neven, "Classification with Quantum Neural Networks on Near Term Processors," arXiv:1802.06002.
[8] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan

Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nat. Commun. **9**, 4812 (2018).

[9] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Phys. Rev. A **98**, 032309 (2018).

[10] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire, "Towards quantum machine learning with tensor networks," Quantum Sci. Technol. **4**, 024001 (2019).

[11] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe, "Circuit-centric quantum classifiers," Phys. Rev. A **101**, 032308 (2020).

[12] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G Green, and Simone Severini, "Hierarchical quantum classifiers," npj Quantum Inf. **4**, 65 (2018).

[13] Jin-Guo Liu and Lei Wang, "Differentiable learning of quantum circuit Born machines," Phys. Rev. A **98**, 062324 (2018).

[14] Guillaume Verdon, Jason Pye, and Michael Broughton, "A Universal Training Algorithm for Quantum Deep Learning," arXiv:1806.09729.

[15] Jinfeng Zeng, Yufeng Wu, Jin-Guo Liu, Lei Wang, and Jiangping Hu, "Learning and inference on generative adversarial quantum circuits," Phys. Rev. A **99**, 052306 (2019).

[16] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao, "The Expressive Power of Parameterized Quantum Circuits," arXiv:1810.11922.

[17] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J Osborne, Robert Salzmann, and Ramona Wolf, "Efficient Learning for Deep Quantum Neural Networks," arXiv:1902.10445.

[18] Matthew J. S. Beach, Roger G. Melko, Tarun Grover, and Timothy H. Hsieh, "Making trotters sprint: A variational imaginary time ansatz for quantum many-body systems," Phys. Rev. B **100**, 094434 (2019).

[19] Stephen H. Shenker, and Douglas Stanford, "Black holes and the butterfly effect," J. High Energy Phys. **2014**,03 (2014).

[20] Daniel A. Roberts, Douglas Stanford, and Leonard Susskind, "Localized shocks," J. High Energy Phys. **2015**,51 (2015).

[21] Daniel A. Roberts, and Douglas Stanford, "Diagnosing chaos using four-point functions in two-dimensional conformal field theory," Phys. Rev. Lett. **115**, 131603 (2015).

[22] Juan Maldacena, Stephen H. Shenker, and Douglas Stanford, "A bound on chaos," J. High Energy Phys. **2016**,106 (2016).

[23] Pavan Hosur, Xiao-Liang Qi, and Daniel Roberts, "Chaos in quantum channels," J. High Energy Phys. **2016**,4 (2016).

[24] Ruihua Fan, Pengfei Zhang, Huitao Shen and Hui Zhai, "Out-of-Time-Order correlation for many-body localization," Sci. Bulletin, 62(10), 707-711 (2016).

[25] Mukund Rangamani, and Massimiliano Rota, "Entanglement structures in qubit systems," J. Phys. A: Math. Theor. 48, 385301 (2015).

[26] Alexei Kitaev, and John Preskill, "Topological entanglement entropy," Phys. Rev. Lett. **96**, 110404 (2006).

[27] Michael Levin, and Xiao-Gang Wen, "Detecting topological order in a ground state wave function," Phys. Rev. Lett. **96**, 110405 (2006).

[28] Christoph Sünderhauf, Lorenzo Piroli, Xiao-Liang Qi, Norbert Schuch, and J. Ignacio Cirac, "Quantum chaos in the Brownian SYK model with large finite N : OTOCs and tripartite information", Journal of High Energy Physics, **2019**, 38 (2019).

[29] Daniel A. Roberts, Douglas Stanford, and Alexandre Streicher, "Operator growth in the SYK model," J. High Energy Phys. **2018**,122 (2018).

[30] Adam Nahum, Sagar Vijay, and Jeongwan Haah, "Operator spreading in random unitary circuits," Phys. Rev. X **8**,021014 (2018).

[31] C. W. von Keyserlingk, Tibor Rakovszky, Frank Pollmann, and S. L. Sondhi, "Operator hydrodynamics, OTOCs, and entanglement growth in systems without conservation laws," Phys. Rev. X **8**,021013 (2018).

[32] Vedika Khemani, Ashvin Vishwanath, and David A. Huse, "Operator Spreading and the Emergence of Dissipative Hydrodynamics under Unitary Evolution with Conservation Laws," Phys. Rev. X **8**,031057 (2018).

[33] Tibor Rakovszky, Frank Pollmann, and C. W. von Keyserlingk, " Diffusive hydrodynamics of Out-of-Time-Order Correlators with Charge Conservation," Phys. Rev. X **8**,031058 (2018).

[34] Xiao Chen, and Tianci Zhou, "Operator scrambling and quantum chaos," arXiv: 1804.08655 (2018).

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

[36] Shenglong Xu, and Brian Swingle, "Locality, quantum fluctuations, and scrambling," Phys. Rev. X. **9**, 031048 (2019).

[37] Xiao-Liang Qi, and Alexandre Streicher, "Quantum epidemiology: operator growth, thermal effects and SYK," J. High Energy Phys. **2019**,12 (2019).

[38] Shenglong Xu, and Brian Swingle, "Accessing scrambling using matrix product operators," Nat. Phys. **16**, 199 – 204 (2020).

[39] Yuri D. Lensky, Xiao-Liang Qi, and Pengfei Zhang, "Size of bulk fermions in the SYK model," J. High Energy Phys. **2020**,53 (2020).

[40] Huitao Shen, Pengfei Zhang, Yi-Zhuang You, and Hui Zhai, "Information scrambling in quantum neural networks," Phys. Rev. Lett. **124**, 200504 (2020).

[41] See the supplementary information for details of how to encode the information of a RGB image into a quantum wave function.

[42] Iris Cong, Soonwon Choi, and Mikhail D. Lukin, "Quantum convolutional neural networks," Nat. Phys. **15**,1273 – 1278 (2019).

[43] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook,"Quanvolutional neural networks: powering image recognition with quantum circuits," arXiv: 1904.04767 (2019).

[44] Seunghyeok Oh, Jaeho Choi, and Joongheon Kim, "A tutorial on quantum convolutional neural networks (QCNN)," arXiv: 2009.09423 (2020).

[45] Johannes Bausch, "Recurrent quantum neural networks," arXiv: 2006.14619.

[46] Laxmidhar Behera, Indrani Kar, and Avshalom C. Elitzur, "Recurrent quantum neural network and its applications," *The Emerging Physics of Consciousness* (2006).

[47] Jonathan Romero, Jonathan P. Olson, and Alan Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," Quantum Sci. Technol. **2**,045001 (2017).

[48] Dmytro Bondarenko, and Polina Feldmann, "Quantum autoencoders to denoise quantum data," Phys. Rev. Lett. **124**,130502 (2020).

[49] Alex Pepper, Nora Tischler, and Geoff J. Pryde, "Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning," Phys. Rev. Lett. **122**,060501 (2019).

[50] For supercube circuits (S), we set the last unit $\hat{U}_1$ the same as the Hyperbolic unit Fig. 1(e). Other units $\hat{U}_{l\geq 2}$ are choosen such that the circuit $\hat{U}_L \cdots \hat{U}_2$ repeats the structure of Fig.1(f) and each unit contains 7 two-qubit gates. Such a structure eliminates redundant gates in U_1 and is found to be optimal for the operator size growth.