A quantum neural network with efficient optimization and interpretability

Pei-Lin Zheng, Jia-Bao Wang, and Yi Zhang∗
International Center for Quantum Materials, School of Physics, Peking University, Beijing, 100871, China
(Dated: October 2022)

As the quantum counterparts to the classical artificial neural networks underlying widespread machine-learning applications, unitary-based quantum neural networks are active in various fields of quantum computation. Despite the potential, their developments have been hampered by the elevated cost of optimizations and difficulty in realizations. Here, we propose a quantum neural network in the form of fermion models whose physical properties, such as the local density of states and conditional conductance, serve as outputs, and establish an efficient optimization comparable to back-propagation. In addition to competitive accuracy on challenging classical machine-learning benchmarks, our fermion quantum neural network performs machine learning on quantum systems with high precision and without preprocessing. The quantum nature also brings various other advantages, e.g., quantum correlations entitle networks with more general and local connectivity facilitating numerical simulations and experimental realizations, as well as novel perspectives to address the vanishing gradient problem long plaguing deep networks. We also demonstrate the applications of our quantum toolbox, such as quantum-entanglement analysis, for interpretable machine learning, including training dynamics, decision logic flow, and criteria formulation.

Introduction—Artificial neural networks (ANNs) lay the foundation of cutting-edge machine learning research and witness profound successes in artificial intelligence applications [1–8]. Their successes largely lie in their versatile expression and efficient optimization, as back-propagation collectively determines the model’s descending gradients in deep, feed-forward architecture [1–3]. On the other hand, such structural complexity comes at the price of increasingly obscure interpretability and the vanishing gradient problem [9, 10] - frustrating obstacles to a new generation of machine learning with advanced control and effectiveness, especially for scientific utilities. Recently, various studies have applied ANNs to quantum data and systems [11–34], where classical ANNs witness typical compatibility issues and usually require suitable bridging and preprocessing of quantum inputs [24–34].

A quantum neural network (QNN) is a quantum circuit of unitaries commonly used in quantum computations [35] and quantum machine learning [36, 37]. Compared with classical ANNs, the QNNs may possess a quantum advantage [38–47], especially in their expressive and computational potential [48–52]. However, as quantum many-body systems, the QNNs are generally costly to simulate on classical computers [38–44] and, notably, more expensive to optimize [53–57] - due to the lack of efficient optimization, various studies had to resort to brute-force gradient calculations or extensive circuit samplings based on parameter-shift rules [58, 59]. Consequently, most machine-learning demonstrations of QNNs are limited to small systems and mediocre performances compared to the ANN benchmark in the current noisy intermediate-scale quantum (NISQ) era.

Here, we consider a fermion bilinear model over a network of sites (black dots), whose hopping amplitudes \( t_{rr'} \) and onsite potentials \( \mu_r \) are parameters for supervised machine learning. It integrates the classical data or quantum model inputs as the \( t = 0 \) layer from the left, and its subsequent physical properties, such as the LDOS on the right, serve as the outputs. Such properties are available via Green’s function \( G(l) \) recursively, adding one layer (\( \hat{T}_{l-1} \) and \( \hat{H}_l \)) at a time. Therefore, for efficient optimization, we only allow finite \( t_{rr'} \) intra-layer (orange lines) or between the neighboring layers (green lines). For clarity, we only show the hopping concerning a single site (dashed circle). In addition to such fully-connected FQNNs, locally-connected FQNNs are also applicable for machine learning, thanks to quantum correlation.

on site potential over the neurons, granting it extensive degrees of freedom. Its specific physical properties, such as local density of states (LDOS) and conditional conductance (CC), possess rich expressions and serve as outputs. We name such a quantum model as a fermion quantum neural network (FQNN) and show that it possesses the benefits of both classical and quantum worlds. On the one hand, for an FQNN with hopping only between the nearest layer and within each layer (Fig. 1), we establish efficient optimization similar to the back-propagation of

---

* frankzhangyi@gmail.com
classical feed-forward ANNs. On the other hand, FQNN connects directly with target quantum data and models without the need for preprocessing and also entities our comprehensive quantum toolboxes, such as quantum entanglement and correlation. We demonstrate FQNNs on classical MNIST [60] benchmarks and quantum models with excellent efficiency, accuracy, and compatibility; further, we show that quantum analysis brings unique interpretable machine learning [32, 61–63]: training dynamics, decision logic flow, and criteria formulation. Notably, while FQNNs are straightforwardly operable on classical computers, their quantum nature allows for not only more general and local networks, thus better feasibility overall, but also a perspective and potential solution towards the long-standing vanishing gradient problem.

**FQNN for machine learning**—Without loss of generality, our FQNN model takes the following form:

\[
\hat{H} = \sum_{rr'} t_{rr'} c^\dagger_r c_{r'} + \sum_r \mu_r c^\dagger_r c_r,
\]

where the hopping amplitudes \(t_{rr'} = t_{r'r}\), and the onsite potentials \(\mu_r\) are the model parameters adjusted via machine learning, similar to the weights and biases of a classical ANN. We label each neuron (site) with \(r = (l, m)\), where \(l \in [0, L]\) denotes the layer and \(m\) denotes the intra-layer coordinate, respectively. We demand finite \(t_{rr'}\) only for \(|l - l'| \leq 1\); we do allow intra-layer hopping, unlike feed-forward classical ANNs; see an example of FQNN architecture in Fig. 1.

The input data or quantum system \((l = 0)\) connects to FQNN via \(T_0\) to the first layer \((l' = 1)\), after which the outputs take the form of specific physical properties of the entire quantum model. For example, we may use the LDOS at the neurons in the last layer \((l = L, k = 1, \cdots, M_L)\) as outputs:

\[
y_k = -\frac{1}{\pi} \Im \left[ \mathcal{G}^{(L)}_{L,i,L} \right]_{k,k},
\]

where \(\mathcal{G}^{(L)}_{l,m,m'}\) are FQNN’s Green’s functions between the \((l, m)\) and \((l', m')\) sites. For a classification problem, we can conveniently set the number of output neurons \(M_L\) to the number of categories, where the neuron with the largest LDOS signals the FQNN’s decision. Likewise, we may use the FQNN’s CC, i.e., the localization property across its depth:

\[
y = \sum_m \left| \mathcal{G}^{(L)}_{0,i,L} \right|_{m,1}^2,
\]

which offers a binary output as the FQNN switches on and off like a PN-junction depending on our chosen target. Without loss of generality, we have considered an FQNN with \(M_L = 1\) and averaged over the contributions from all sites \(r = (0, m)\) in the input layer. These measurements are more efficient than the tomography on unitary-based QNNs [64–66].

**Examples and results**—First, we apply FQNNs to the MNIST dataset as a classical benchmark. We encode each image’s pixel grayscale \(x_{\bar{m}} \in [0, 1]\) as the onsite potential \(\mu_{\bar{m}}\) of a model, integrated into the FQNN as the \(l = 0\) layer, see Fig. 1:

\[
\hat{H}_0 = \sum_{\bar{m}} x_{\bar{m}} c^\dagger_{\bar{m}} c_{\bar{m}},
\]

where \(\bar{m} = (m_1, m_2)\) labels the pixels’ 2D coordinates.

We consider different FQNN architectures: first, we consider a fully-connected FQNN with \([100, 64, 10]\) neurons in 3 layers, where \(t_{rr'}\) fully connects each neuron to its own layer and adjacent layers (Fig. 1); then, we consider a locally-connected FQNN with \([13 \times 13, 6 \times 6, 10]\) Hutchinson.
neurons, where hopping is local on the 2D lattices in the $l = 1, 2$ layers and the $l = 3$ layer remains fully-connected. The LDOS over the $l = 3$ layer stands for the 10 MNIST categories. We set learning rate $\eta = 0.005$, level width $\gamma = 0.005$, and weight decay $\lambda = 0.001$. Further details and examples on FQNN architecture and locality are in the Supplemental Material [68].

With the aforementioned optimization approach, we achieve high-quality convergences in the supervised machine learning of MNIST. With a fully-connected FQNN, we achieve 98.13% accuracy (Fig. 2 and Ref. [68]), comparable with classical fully-connected ANNs with similar compact scales ($\sim$ 98.5% [69]) and blowing past conventional QNNs that had to settle with small systems, coarse-graining on big data, high cost, and meager accuracy. In addition, with an efficient algorithm and fast convergence, FQNNs have much potential to expand further on their architectures [70]. Interestingly, we still achieve satisfactory accuracy - 97.36% on MNIST - using FQNNs with only local hopping, a vital step towards feasibility on quantum devices. Indeed, quantum correlation naturally extends beyond the hopping range, and the extra intra-layer hopping, absent in classical ANNs, further boosts the communication and network generality in FQNN.

We note the diverse possibilities over FQNN input and output formalisms as long as we keep the consistency between machine learning and subsequent applications. For instance, instead of the model in Eq. 5, we may encode an image as an external, static LDOS field $\mathcal{G}_{\text{ext}}(\vec{m}) \propto x(\vec{m})$ directly coupled to the FQNN. We set $\eta = 0.001$, $\gamma = 0.001$, and $\lambda = 0.001$. The FQNN achieves an (even higher) accuracy of 98.54% on the MNIST dataset; see further details in the Supplemental Material [68]. We discuss other output formalisms in the following example.

Next, we consider training FQNN to distinguish Chern insulators (Chern number $C = 1$) and normal insulators ($C = 0$) with the following fermion bilinear models on a $12 \times 12$ square lattice $\vec{r} = (x, y)$:

$$\hat{H}(\kappa) = \sum_{\vec{r}} (-1)^y c_{\vec{r}+\hat{x}}^\dagger c_{\vec{r}} + [1 + (-1)^y (1 - \kappa)] c_{\vec{r}+\hat{y}}^\dagger c_{\vec{r}} + (-1)^y \frac{\tilde{\kappa}}{2} [c_{\vec{r}+\hat{x}+\hat{y}}^\dagger c_{\vec{r}} - c_{\vec{r}+\hat{x}-\hat{y}}^\dagger c_{\vec{r}}] + \text{h.c.}, \quad (6)$$

which yields a Chern insulator if $\kappa > \kappa_C = 0.5$ and a normal insulator otherwise at $\kappa = 0$. In practice, we also add quenched disorder among the onsite potentials as well as nearest and next nearest neighbor hopping amplitudes to Eq. 6 for a diverse training set and apply the real-space Kubo formula [24, 71] for each sample’s topological category [68].

FQNN incorporates each sample directly as its $l = 0$ layer; in comparison, classical ANNs are incompatible with such quantum models and have to resort to proper preprocessing, e.g., quantum operators, quantum entanglement or surface states. For the binary decision of whether the input is a Chern insulator or not, we may use either the LDOS over two neurons in the $l = L$ layer or the CC across the entire FQNN. For the latter, the FQNN is localized at $E \approx 0$ with Green’s functions decaying exponentially thus vanishing between the first and last layers if $\tilde{H}_0$ is a normal insulator, and vice versa. We employ FQNNs with $[100, [64] \times 3, 2]$ neurons in the 5 layers for the former case and with $[100, [64] \times 5, 1]$ neurons distributed among 7 layers for the latter. We set $\eta = 0.001 \sim 0.005$, $\gamma = 0.01$, and $\lambda = 0.001$.

After the supervised machine learning reaches convergence, we apply the FQNNs to examine various sample models. We obtain 100% accuracy on test models without the disorder, whose phase diagrams in $\kappa$ in Fig. 3(a)(c) display an apparent topological transition at $\kappa = 0.5$ fully consistent with the theory benchmark. Even on test models with a moderate disorder, the FQNN still achieves an accuracy of 99.83% for LDOS and 99.51% for CC [72], as well as 100% and 99.99% test area under the receiver operating characteristic curve (AUROC) [73, 74], respectively. Interestingly, responsible for the CC behavior, an extended state emerges at $E \approx 0$ in the gap when and only when the FQNN meets up with a Chern insulator (Fig. 3(b)(d)), which is useful for interpretability as we show later.

Interpretability—FQNN has inherited classical ANNs’ merits of efficient optimization and powerful expression while exhibiting compatibility in both the classical and quantum worlds. In addition, its quantum nature allows us to take advantage of our quantum toolbox, such as quantum entanglement [75, 76], for novel perspectives on interpretable quantum machine learning. For example, mutual information [77, 78] measures the information de-
dependence between two subregions A and B:
\[ I_{AB} = S_A + S_B - S_{AB}, \]  
(7)
where \( S_A = -\text{tr}(\rho_A \log \rho_A) \), \( \rho_A = \text{tr} \bar{\rho}_A(\rho) \) are the entanglement entropy and (reduced) density operator of A. The mutual information \( I_{l\ell_0} \) between the \( l = 0 \) and \( l = L \) layers evaluate the FQNN’s control on converting inputs into outputs, thus its machine learning progress; see Fig. 2 for an illustration of \( I_{l\ell_0} \) on mid-training dynamics.

For a post-learning FQNN, we can also employ mutual information to track the input-to-output logic flow through its full depth. First, given a target input, we can establish mutual information between every neuron and the output neurons, whose LDOS holds the FQNN’s decision. Further, we can apply a unitary transformation to each layer so that the mutual information concentrates among a few selected neurons, simplifying the logic flows. For this purpose, we solve for a new basis where most neurons do not correlate with the output neurons; see more details in the Supplemental Material [68]. For the FQNN in Fig. 3(a), a logic flow example given a Chern insulator input and the mutual information distributions before and after the unitary transformation are in Fig. 4(a)(b). We note that such unitary transformations are controlled, invertible, and sample-wise applicable - no transformation of this kind is available to classical ANNs [79].

There are also interpretive machine learning options for an FQNN with CC as its output. In the presence of a Chern insulator \( \hat{H}_0 \) at the input layer, the overall quantum fermion model should possess an extended state at \( E = 0 \) (Fig. 3(d)), suggesting:
\[
\det \begin{pmatrix} \hat{H}_0 & \hat{T}_0 & \hat{T}_0^\dagger & \hat{Q} \end{pmatrix} = 0,
\]  
(8)
which offers a natural formulation of the FQNN’s criteria on the target model \( \hat{H}_0 \) that lead to a Chern-insulator decision. Here, \( \hat{Q} \) is the Hamiltonian of the \( l = 1, 2, \cdots, L \) layers of the FQNN, connected to the input model via \( \hat{T}_0 \) and \( \hat{T}_0^\dagger \). \( \hat{T}_0^\dagger, \hat{T}_0, \hat{T}_0^\dagger \) consist of the model parameters \( t_{rr^{'}} \) and \( \mu_r \) and are fixed by training. Also, given a Chern insulator model \( \hat{H}_{\text{ch}} \), we design a Chern-insulator criteria:
\[
\hat{H}_0 = \hat{H}_{\text{ch}} + \sum_i \omega_i \hat{h}_{\delta i} + \omega_i^* \hat{h}_{\delta i}^{'},
\]  
(9)
for any sufficiently small \( \omega_i \), where \( \hat{h}_{\delta i} \) \((i = 1, 2, \cdots)\) are normalized columns linearly depend on the overall matrix and local in \( \hat{H}_0 \). To verify such formulation, we start with a Chern insulator model \( H_{\text{ch}} \) close to the topological transition, e.g., Eq. 6 with \( \kappa = 0.52 \), and evaluate the consequences of perturbations with an \( \hat{h}_{\delta} \) in Eq. 9 either random or following our criteria. Example results are in Fig. 4(c). While random perturbations may topple the Chern insulator relatively easily, the designed perturbations are indeed pro-topological-phase, which steadily persists till the perturbation amplitude is fairly large.

Discussion —FQNNs can take advantage of their quantum nature, and their physical properties’ nonlinear recursions from model parameters endow them with as powerful expressions and efficient optimization as classical ANNs. Moreover, FQNN may even exceed the classical ANNs in optimization, which widely suffer from the vanishing-gradient problem as the layers further away from the outputs experience exponentially decreasing gradients, thus unproductive optimizations. The introductions of convolutional neural networks [60, 80] and residue blocks [81] that renovated deep ANNs only partially circumvent such issues. Physically, the vanishing gradient is consistent with the increasing distance and decaying correlation between the target model parameters and the ANN outputs. In contrast, as CC engages the entire depth of the FQNN, its contributions are naturally more equally distributed over and dependent on parameters from all layers. The effect of our remedy to the vanishing gradient is also rooted in the gradients’ recursive expressions [68] and explicitly observable from the example results in Fig. 5 on FQNNs and classical ANNs’ gradient distributions during machine learning.
FIG. 5. The modulus gradients distributed among the layers show that, unlike the ANN that suffers a vanishing gradient for neurons farther from the output, the FQNN with CC output enjoys a more evenly distributed gradient - more advantageous for optimization. We employ identical layer architecture, initialization methods, and Chern/normal insulator models for ANN and FQNN, where the dataset for the former is preprocessed via quantum loop topography [24]. We average the gradients over the first three epochs and normalize the results by setting the last layer’s values as 1 (not shown).

Unlike a quantum-circuit-based QNN, FQNN functions as a non-dynamical quantum fermion model, which can become relatively simpler to implement in analog quantum simulations [82, 83], especially given FQNN’s locality where hopping \( t_{rr'} \) exists only between near neighbors. Recent quantum simulations have exercised precisely controls on the inter-site couplings and onsite chemical potentials [84, 85], and reach system scales of hundreds of sites [86–88], especially in the quantum-dot arrays [84, 89–93] and the Rydberg atom arrays [85–88, 94–97]. In addition to speeding up applications, the machine learning and quantum realization of FQNN also offer a new arena for designing novel phenomena and functional properties, such as CC under diverse circumstances.

We also note that our studies may generalize to non-Hermitian systems [98, 99] FQNNs, where Eq. 1 attains non-reciprocal hopping \( t_{rr'} \neq t_{r'r} \) and more closely resembles the unidirectional information flow in classical ANNs.

Acknowledgments — We thank Zhenduo Wang for insightful discussions, and support from the National Key R&D Program of China (Grant No. 2021YFA1401900) and the National Science Foundation of China (Grant No. 12174008). The computation was supported by the High-performance Computing Platform of Peking University. The source code is available in Ref. [100].

Conflict of interest — The authors declare no conflict of interest.

Author contributions — Y.Z. conceived the idea, P.-L.Z. carried out the calculations, and all authors participated in the discussions and the work’s writing.

Methods — To characterize the quantum model in Eq. 1, we can evaluate the expectation values \( \langle \Phi | O | \Phi \rangle \) for the many-fermion ground state \( | \Phi \rangle = \prod_{\epsilon \leq E} c_\epsilon | 0 \rangle \), where \( c_\epsilon \) corresponds to the energy eigenstates \( \hat{H} = \sum_\epsilon \epsilon c_\epsilon c_\epsilon^\dagger \) and we set the Fermi energy \( E = 0 \). Alternatively, we may resort to Green’s functions:

\[
\hat{G}(z) = \left( \hat{z}^{\dagger} - \hat{H} \right)^{-1},
\]

where \( z = E + i\gamma \) possesses a small imaginary part \( \gamma > 0 \) that attributes a finite level width and avoids singularities. Interestingly, when hopping exists only within the same or adjacent layers, we can evaluate systematic properties via the recursive Green’s function method [101]:

\[
\begin{align*}
\hat{G}^{(N)}_{N,N} &= \left[ z^{\dagger} I_N - \hat{T}_N \hat{G}^{(N-1)}_{N-1,N-1} \hat{T}_N \right]^{-1}, \\
\hat{G}^{(N)}_{l,N} &= \hat{G}^{(N-1)}_{l,N-1} \hat{T}_{N-1} \hat{G}^{(N)}_{N,N},
\end{align*}
\]

where \( \hat{H}_l \) is the Hamiltonian defined within the \( l \)th layer, and \( \hat{T}_l \) is the hopping matrix elements from the \((l + 1)\)th to the \( l \)th layer: \( \hat{H} = \hat{H}_0 + \sum_{l=1}^L \hat{H}_l + \hat{T}_{l-1} + \hat{T}_{l-1}^{\dagger} \). The recursion starts with \( \hat{G}^{(0)}_{0,0} = \left[ z^{\dagger} I_0 - \hat{H}_0 \right]^{-1} \) for the input model on the left \((l = 0)\), adding one layer to the right at a time for a system of \( N \) layers at the \( N \)th iteration, as denoted by the superscript \( N \in [1, L] \) in the braces; see Fig. 1. The recursive expressions for the rest of \( \hat{G} \) components are in the Supplemental Material [68].

Importantly, the left-hand sides of Eq. 11 at the \( N \)th iteration only depend on the right-hand sides at the \( (N - 1) \)th iteration. Therefore, the time complexity of each layer is only polynomially related to the number of neurons \( M_l \) in the current layer, similar to classical ANNs. Also, the functions mapping each iteration’s inputs to the outputs are highly nonlinear. At the very end of the recursion lies \( \hat{G}^{(L)} \), which encodes various physical properties of the entire FQNN and provides outputs, e.g., the LDOS on the last layer’s neurons \((l = L)\) or the CC across the depth of FQNN. These features, like the feed-forward architecture and the nonlinear activation function of classical ANNs, endow the FQNN with powerful expressive potential and collective gradient solution through the chain rule - major merits that made ANN’s name in machine learning and that prior QNNs lack the most.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature 521, 436 (2015).
[2] M. Jordan and T. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349, 255 (2015).
[3] Michael A. Nielsen, Neural Networks and Deep Learning (Determination Press, 2015).
[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529, 484 (2016).

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, Mastering the game of go without human knowledge, Nature 550, 354 (2017).

[6] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tuyunasunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Ziehe, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, Highly accurate protein structure prediction with alphafold, Nature 596, 583 (2021).

[7] A. Davies, P. Velickovic, L. Buesing, S. Blackwell, D. Zheng, N. Tomasev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhasz, M. Lackenby, G. Williamson, D. Hassabis, and P. Kohli, Advancing mathematics by guiding human intuition with ai, Nature 600, 70 (2021).

[8] F. Agostinelli, S. McAler, A. Shmakov, and P. Baldi, Solving the rubik’s cube with deep reinforcement learning and search, Nature Machine Intelligence 1, 356 (2019).

[9] J. F. Kolen and S. C. Kremer, Gradient flow in recurrent nets: The difficulty of learning longterm dependencies, in A Field Guide to Dynamical Recurrent Networks (2001) pp. 237–243.

[10] J. Han and C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning, in From Natural to Artificial Neural Computation, edited by J. Mira and F. Sandoval (Springer Berlin Heidelberg, Berlin, Heidelberg, 1995) pp. 195–201.

[11] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. 91, 045002 (2019).

[12] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nature Physics 13, 431 (2017).

[13] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science 355, 602 (2017).

[14] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, Phys. Rev. B 96, 195145 (2017).

[15] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nature Physics 14, 447 (2018).

[16] A. A. Melnikov, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Active learning machine learns to create new quantum experiments, Proceedings of the National Academy of Sciences 115, 1221 (2018).

[17] Y. Zhang, A. Mesgaros, K. Fujita, S. Edkins, M. Hamidian, K. Ch’ng, H. Eisaki, S. Uchida, J. S. Davis, E. Khatami, et al., Machine learning in electronic-quantum-matter imaging experiments, Nature 570, 484 (2019).

[18] D. Luo and B. K. Clark, Backflow transformations via neural networks for quantum many-body wave functions, Phys. Rev. Lett. 122, 226401 (2019).

[19] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, Nature Physics 15, 917 (2019).

[20] W. Lian, S.-T. Wang, S. Lu, Y. Huang, F. Wang, X. Yuan, W. Zhang, X. Ouyang, X. Wang, X. Huang, L. He, X. Chang, D.-L. Deng, and L. Duan, Machine learning topological phases with a solid-state quantum simulator, Phys. Rev. Lett. 122, 210503 (2019).

[21] Y.-H. Zhang, P.-L. Zheng, Y. Zhang, and D.-L. Deng, Topological quantum compiling with reinforcement learning, Phys. Rev. Lett. 125, 170501 (2020).

[22] R. Fournier, L. Wang, O. V. Yazeyev, and Q. Wu, Artificial neural network approach to the analytic continuation problem, Phys. Rev. Lett. 124, 056401 (2020).

[23] G. Koolstra, N. Stevenson, S. Barzili, L. Burns, K. Siva, S. Greenfield, W. Livingston, A. Hashim, R. K. Naik, J. M. Kreilebaum, K. P. O’Brien, D. I. Santiago, J. Dressel, and I. Siddiqi, Monitoring fast superconducting qubit dynamics using a neural network, Phys. Rev. X 12, 031017 (2022).

[24] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. 118, 216401 (2017).

[25] T. Ohtsuki and T. Ohtsuki, Deep Learning the Quantum Phase Transitions in Random Two-Dimensional Electron Systems, Journal of the Physical Society of Japan 85, 123706 (2016).

[26] T. Ohtsuki and T. Ohtsuki, Deep learning the quantum phase transitions in random electron systems: Applications to three dimensions, Journal of the Physical Society of Japan 86, 044708 (2017).

[27] E. van Nieuwenburg, Y.-H. Liu, and S. Huber, Learning phase transitions by confusion, Nature Physics 13, 435 (2017).

[28] P. Zhang, H. Shen, and H. Zhai, Machine learning topological invariants with neural networks, Phys. Rev. Lett. 120, 066401 (2018).

[29] Y. Zhang, R. G. Melko, and E.-A. Kim, Machine learning Z2 quantum spin liquids with quasiparticle statistics, Phys. Rev. B 96, 245119 (2017).

[30] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knapp, Classifying snapshots of the doped hubbard model with machine learning, Nature Physics 15, 921 (2019).

[31] V. Peano, F. Sapper, and F. Marquardt, Rapid exploration of topological band structures using deep learning, Phys. Rev. X 11, 021052 (2021).

[32] J. Arnold and F. Schäfer, Replacing neural networks by optimal analytical predictors for the detection of phase transitions, Phys. Rev. X 12, 031044 (2022).

[33] P.-L. Zheng, S.-J. Du, and Y. Zhang, Ground-state properties via machine learning quantum constraints, Phys. Rev. Research 4, L032043 (2022).

[34] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in
learning from experiments, Science 376, 1182 (2022).
[35] M. A. Nielsen and I. Chuang, Quantum computation and quantum information (2002).
[36] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature 549, 195 (2017).
[37] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, Nature Computational Science 2, 567 (2022).
[38] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature 549, 203 (2017).
[39] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Review 41, 303 (1999), https://doi.org/10.1137/S0036144598347011.
[40] B. M. Terhal and D. P. DiVincenzo, Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games, Quantum Info. Comput. 4, 134–145 (2004).
[41] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, Theory of Computing 9, 143 (2013).
[42] S. Aaronson and L. Chen, Complexity-theoretic foundations of quantum supremacy experiments, in Proceedings of the 32nd Computational Complexity Conference, CCC’17 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, 2017).
[43] S. Bravyi, D. Gosset, and R. Konig, Quantum advantage with shallow circuits, Science 362, 308 (2018).
[44] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in learning from experiments, Science 376, 1182 (2022).
[45] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysy, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindermay, E. Lucero, D. Lyakh, S. Mandra, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C.Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, Quantum supremacy through a programmable superconducting processor, Nature 574, 505 (2019).
[46] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Quantum computational advantage using photons, Science 370, 1460 (2020).
[47] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, L. Wang, S. Wang, D. Wu, Y. Xu, K. Yan, W. Yang, Y. Yang, Y. Ye, J. Yin, C. Ying, J. Yu, C. Zha, C. Zhang, H. Zhang, K. Zhang, Y. Zhang, H. Zhao, Y. Zhao, L. Zhou, Q. Zhu, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. 127, 180501 (2021).
[48] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics 3, 625 (2021).
[49] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nature Physics 15, 1273 (2019).
[50] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nature Physics 17, 1013 (2021).
[51] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nature Computational Science 1, 403 (2021).
[52] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nature Communications 12, 2631 (2021).
[53] L. Bittl and M. Kliesch, Training variational quantum algorithms is np-hard, Phys. Rev. Lett. 127, 120502 (2021).
[54] Z.-D. Wang, P.-L. Zheng, B. Wu, and Y. Zhang, Quantum dropout for efficient quantum approximate optimization algorithm on combinatorial optimization problems (2022).
[55] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nature Communications 9, 4812 (2018).
[56] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nature Communications 12, 1791 (2021).
[57] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, Nature Communications 12, 6961 (2021).
[58] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A 98, 032309 (2018).
[59] M. Schulz, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A 99, 032331 (2019).
[60] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86, 2278 (1998).
[61] Y. Zhang, P. Ginsparg, and E.-A. Kim, Interpreting machine learning of topological quantum phase transitions, Phys. Rev. Research 2, 023283 (2020).
[62] M. T. Ribeiro, S. Singh, and C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16 (Association for Computing Machinery, New York, NY, USA, 2016) p. 1135–1144.
[63] A. Barredo Arrieta, N. Diaz-Rodriguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and
F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Inf. Fusion 58, 82–115 (2020).

[64] J. Tersoff and D. R. Hamann, Theory of the scanning tunneling microscope, Phys. Rev. B 31, 805 (1985).

[65] M. F. Crommie, C. P. Lutz, and D. M. Eigler, Confinement of electrons to quantum corrals on a metal surface, Science 262, 218 (1993).

[66] B. J. van Wees, H. van Houten, C. W. J. Beenakker, J. G. Williamson, L. P. Kouwenhoven, H. van der Marel, and C. T. Foxon, Quantized conductance of point contacts in a two-dimensional electron gas, Phys. Rev. Lett. 60, 848 (1988).

[67] A. Krogh and J. A. Hertz, A simple weight decay can improve generalization, in Proceedings of the 4th International Conference on Neural Information Processing Systems, NIPS’91 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991) p. 950–957.

[68] Please refer to Supplemental Materials for the details on additional examples and details on FQNN inputs, architectures, regularization, local unitary transformations, the gradient chain rule, and the insulator models, which include Ref. [102–110].

[69] The benchmark accuracy obtained by our own attempts is 98.99%, while the benchmark accuracy of an ANN of approximately the same size at http://yann.lecun.com/exdb/mnist/ is 98.47%.

[70] The state-of-the-art accuracy of 99.91% on MNIST is by ensemble learning with convolutional neural networks (CNN) [111], refer to https://paperswithcode.com/sota/image-classification-on-mnist.

[71] A. Kitaev, Anyons in an exactly solved model and barton, Annals of Physics 321, 2 (2006).

[72] For better figurative illustrations, the FQNNs in Fig. 3 differ from these best-accuracy ones and achieve 99.07% and 98.1% accuracy, respectively.

[73] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recogn. 30, 1145–1159 (1997).

[74] T. Fawcett, An introduction to roc analysis, Pattern Recogn. Lett. 27, 861–874 (2006).

[75] L. Amico, R. Fazio, A. Osterloh, and V. Vedral, Entanglement in many-body systems, Rev. Mod. Phys. 80, 517 (2008).

[76] J. Eisert, M. Cramer, and M. B. Plenio, Colloquium: Many-body physics with individually controlled rydberg atoms, Nature Physics 16, 132 (2020).

[77] S. Scholl, M. Schuler, H. J. Williams, A. A. Eberharder, D. Barredo, K.-N. Schymik, V. Lienhard, L-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms, Nature 595, 233 (2021).

[78] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, Quantum optimization of maximum independent set using rydberg atom arrays, Science 376, 1209 (2022).

[79] D. Bluvstein, A. Omran, H. Levine, A. Keesling, G. Semeghini, S. Ebadi, T. T. Wang, A. A. Michaelidis, N. Maskara, W. W. Ho, S. Choi, M. Serbyn, M. Greiner, V. Vuletić, and M. D. Lukin, Controlling quantum many-body dynamics in driven rydberg atom arrays, Science 371, 1355 (2021).

[80] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, Spins in few-electron quantum dots, Rev. Mod. Phys. 79, 1217 (2007).

[81] T. Byrnes, N. Y. Kim, K. Kusudo, and Y. Yamamoto, Quantum simulation of fermi-hubbard models in semiconductor quantum-dot arrays, Phys. Rev. B 78, 075320 (2008).

[82] F. R. Braakman, P. Barthelemy, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, Long-distance coherent coupling in a quantum dot array, Nature Nanotechnology 8, 432 (2013).

[83] T. Hengsens, T. Fujita, L. Janssen, X. Li, C. J. Van Diepen, C. Reichl, W. Wegscheider, S. Das Sarma, and L. M. K. Vandersypen, Quantum simulation of a fermi–hubbard model using a semiconductor quantum dot array, Nature 548, 70 (2017).

[84] C. J. van Diepen, T.-K. Hsiao, U. Mukhopadhyay, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, Quantum simulation of antiferromagnetic heisenberg chain with gate-defined quantum dots, Phys. Rev. X 11, 041025 (2021).

[85] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with rydberg atoms, Rev. Mod. Phys. 82, 2313 (2010).

[86] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, A rydberg quantum simulator, Nature Physics 6, 382 (2010).

[87] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, J. Cui, M. Rossignolo, P. Rembold, S. Montangero, T. Calarco, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, Generation and manipulation of schrödinger cat states in rydberg atom arrays, Science 365, 570 (2019).
Quantum nonlinear optics based on two-dimensional rydberg atom arrays, Phys. Rev. Lett. **127**, 263602 (2021).

[98] Y. Ashida, Z. Gong, and M. Ueda, Non-hermitian physics, Advances in Physics **69**, 249 (2020).

[99] E. J. Bergholtz, J. C. Budich, and F. K. Kunst, Exceptional topology of non-hermitian systems, Rev. Mod. Phys. **93**, 015005 (2021).

[100] https://github.com/PeilinZHENG/FQNN.

[101] A. Croy, R. A. Römer, and M. Schreiber, Localization of electronic states in amorphous materials: Recursive green’s function method and the metal-insulator transition at $e \neq 0$, in *Parallel Algorithms and Cluster Computing*, edited by K. H. Hoffmann and A. Meyer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 203–226.

[102] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Array programming with NumPy, Nature **585**, 357 (2020).

[103] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, Vol. 32 (Curran Associates, Inc., 2019).

[104] J. D. Hunter, Matplotlib: A 2d graphics environment, Computing in Science & Engineering **9**, 90 (2007).

[105] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929–1958 (2014).

[106] K. B. Petersen and M. S. Pedersen, The matrix cookbook (2012), version 20121115.

[107] W. Smith and S. Erdman, A note on the inversion of complex matrices, IEEE Transactions on Automatic Control **19**, 64 (1974).

[108] I. Peschel and V. Eisler, Reduced density matrices and entanglement entropy in free lattice models, Journal of Physics A: Mathematical and Theoretical **42**, 504003 (2009).

[109] I. Peschel, Calculation of reduced density matrices from correlation functions, Journal of Physics A: Mathematical and General **36**, L205 (2003).

[110] A. M. Turner, Y. Zhang, and A. Vishwanath, Entanglement and inversion symmetry in topological insulators, Phys. Rev. B **82**, 241102 (2010).

[111] S. An, M. Lee, S. Park, H. Yang, and J. So, An ensemble of simple convolutional neural network models for mnist digit recognition (2020).
Supplemental Material: A quantum neural network with efficient optimization and interpretability

Pei-Lin Zheng, Jia-Bao Wang, and Yi Zhang‡

International Center for Quantum Materials, School of Physics, Peking University, Beijing, 100871, China

(Dated: October 2022)

I. ADDITIONAL EXAMPLES ON FQNN

INPUTS: ENCODING CLASSICAL DATA AS EXTERNAL LDOS

Given classical data, there are diverse encoding methods for FQNNs’ inputs. For example, in addition to the onsite potential $\mu_{\vec{m}}$ of a model, we may also regard each sample as $\mathcal{I}(\hat{G}_{00})_{\vec{m}\vec{m}} = -\pi \cdot x(\vec{m})$, an external and static LDOS field coupled to the FQNN’s first layer ($l = 1$) via $T_0$ and $T_0^\dagger$. As such inputs do not specify an explicit $\hat{H}_0$, the mutual information $I_{0L}$ between the input and output layers is no longer directly available.

Applying such encoding to the classical MNIST dataset, we carry out supervised machine learning with FQNNs and settings similar to the main text. The LDOS over the 10 neurons in the last layer serves as outputs. We summarize the performance on training and convergence in Fig. 1. With adequate tuning and training, FQNNs can achieve 98.54% accuracy on the MNIST dataset, fully comparable to classical ANNs with similar scale and complexity.

Also, we have investigated adding nearest neighbor hopping to the onsite potential Hamiltonian in Eq. (7) in the main text. The performance remains on par with the scenario without hopping.

II. ADDITIONAL EXAMPLES ON FQNN ARCHITECTURE

We study more FQNN architectures to compare the effect of connectivity and locality on machine-learning performance. We consider FQNNs with the neurons in each layer forming a 2D square lattice and either no intra-layer hopping (unconnected), nearest neighbor intra-layer hopping, or nearest and next-nearest neighbor intra-layer hopping. We also consider two inter-layer connection formalisms: first, we consider FQNNs with $[14 \times 14, 7 \times 7, 10]$ neurons per layer, and tree-type inter-layer hopping - each neuron in the 1st (2nd) layer connects with $2 \times 2$ neurons in the 0th (1st) layer and without overlap; then, we also consider FQNNs with $[13 \times 13, 6 \times 6, 10]$ neurons per layer, where the neighboring neurons’ inter-layer partners share an overlap - inter-layer $t_{rr'}$ connects neuron at $(m_1, m_2)$ in the 1st (2nd) layer and neurons at $(m'_1, m'_2)$ in the 0th (1st) layer, $m'_j \in [2m_j - 1, 2m_j + 2]$.

In all scenarios, the $l = 3$ layer consists of fully-connected neurons whose LDOS represents FQNNs’ classification outputs.

| Intra-layer | Tree | Overlapping |
|-------------|------|-------------|
| Unconnected | 93.43% | 97.17% |
| Upto nearest neighbor | 95.61% | 97.26% |
| Upto next nearest neighbor | 96.19% | 97.36% |

TABLE I. FQNNs’ accuracy on the MNIST test dataset shows relatively sensitive dependence on the architecture - the level of locality and connectivity. The image’s pixel grayscale is encoded as the zeroth layer model’s onsite potentials; see Eq. (7) in the main text.

| Intra-layer | Tree | Overlapping |
|-------------|------|-------------|
| Unconnected | 93.40% | 97.84% |
| Upto nearest neighbor | 96.27% | 97.68% |
| Upto next nearest neighbor | 96.16% | 97.72% |

TABLE II. FQNNs’ accuracy on the MNIST test dataset shows relatively sensitive dependence on the architecture - the level of locality and connectivity. The image’s pixel grayscale is encoded as an external, static LDOS field coupled to the FQNN’s first layer; see Sec. I.

* frankzhangyi@gmail.com

FIG. 1. The loss function on the training set and the accuracy over the test set consistently demonstrate the convergence of a fully-connected FQNN in supervised machine learning on the MNIST dataset encoded as external LDOS. The shades illustrate typical variance over 10 trails.
Applying such FQNNs to supervised machine learning for the MNIST dataset, encoded as onsite potentials in Eq. (7) in the main text or as an external LDOS field in Sec. I, we summarize the optimal performances in Tabs. I and II. We note that poor connectivity, e.g., the tree architecture (‘tree column’ and ‘unconnected’ row), may adversely impact the FQNN capacity - such locality is too strict to allow sufficient correlations between neurons in the first layers. Fortunately, unlike classical ANNs, FQNNs allow intra-layer connections, adding to available inter-layer hopping, which we observe bears more significant contributions. Given sufficient connectivity above the tree architecture, FQNNs’ performances increase quickly and saturate to the level of fully-connected FQNNs. Further reduction of locality tends to receive a diminishing margin.

III. DETAILS ON REGULARIZATION

Regularization plays a vital role in reducing over-fitting and reaching optimal performance in supervised machine learning. Common regularization approaches for classical ANNs are weight decay [1] and dropout [2]. Noting the similarity between FQNN’s $\tilde{T}_{(-1)}$ and $\tilde{H}_l$ with classical ANN’s weights and biases, we can carry over the weight decay, also known as L2 regularization, for the benefit of FQNN’s supervised machine learning. Indeed, we have observed improved results with the inclusion of a small weight decay of $\lambda = 0.001$ in various scenarios; see Fig. 2 for examples. In addition, most of the optimal results in Tabs. I and II are obtained with the weight decay. As a caveat, we also observe increased fluctuations in mid-training performances with such regularization.

Another regularization is the small imaginary part $\gamma$ added to the energy $E$. Physically, such $\gamma$ introduces a finite level width, reduces singularities, and may originate from the averaged effect of random quenched disorder. Like dropout introduces effective averaging over multiple descendants of a classical ANN, a proper value of $\gamma$ may represent an average over various random disorder configurations over the FQNN, thus reducing sensitivity towards details and enhancing generality. A closer comparison with dropout for classical ANNs would suggest training with variable disorder configurations, which are then replaced with a corresponding $\gamma$ for tests and applications. We leave such endeavors and more throughout investigations on FQNN regularization to future studies.

IV. DETAILS ON DISORDERED INSULATOR MODELS AND THEIR TOPOLOGICAL CHARACTERIZATION

To increase the diversity of the insulator models in Eq. (8) in the main text, we also include in their Hamiltonians the following terms on random quenched disorder:

$$\hat{H}_{\text{dis}} = \sum_{\vec{r}} w^{(1)}_{\vec{r}} c^\dagger_{\vec{r}+\vec{x}} c + w^{(-1)}_{\vec{r}} c^\dagger_{\vec{r}+\vec{y}} c + w^{(2)}_{\vec{r}} c^\dagger_{\vec{r}+\vec{x}+\vec{y}} c + w^{(-2)}_{\vec{r}} c^\dagger_{\vec{r}+\vec{x}-\vec{y}} c + \text{h.c.} = w^{(0)}_{\vec{r}} c^\dagger_{\vec{r}} c, \quad (1)$$

where $w^{(0)}_{\vec{r}} \in [-W_0, W_0]$, $w^{(\pm 1)}_{\vec{r}} \in [-W_1, W_1]$, $w^{(\pm 2)}_{\vec{r}} \in [-W_2, W_2]$ are perturbations to the onsite potential, nearest-neighbor hopping and next-nearest neighbor hopping, respectively. In practice, we set $W_0 \in [1.0, 3.0]$, $W_1 \in [0, 1.0]$, $W_2 \in [0, 0.5]$.

Since disorder and perturbations may modify the topological character of a model, it may be rash to determine whether samples with $\hat{H}_{\text{dis}}$ are Chern or normal insulators just with their $\kappa$ value in the clean limit. Instead, we employ the real-space Kubo formula [3, 4] to evaluate the transverse conductance in accord with the Chern
number:
\[ C = \frac{4\pi i}{N} \sum_{jkl} P_{jk} P_{kl} P_{lj} S_{\triangle jkl}, \]
(2)

where \( P_{jk} = \langle c_j^\dagger c_k \rangle \) is the two-point correlator between sites \( i \) and \( j \), \( S_{\triangle jkl} \) is the signed area of triangle \( jkl \), and \( N \) is the total number of sites. Given the locality of the insulator models, we limit the summation in Eq. 2 to triangles no larger than a cut-off length scale \( d = 3 \) for simplicity. Finally, for model samples with an apparent spectral gap and \( 0.7 \leq C \leq 1.0 \) or \( 0 \leq C \leq 0.3 \) after the cut-off, we include them in our training set as Chern insulators and normal insulators, respectively.

V. DETAILS ON THE LOSS FUNCTION GRADIENT VIA THE CHAIN RULE AND RECURSIVE GREEN’S FUNCTION METHOD

The recursive Green’s function method [5] derives the Green’s functions of the entire system in Eq. (2) in the main text via recursion. Starting from the system’s first layer, the recursive expressions \((i, j < N)\):
\[ \hat{G}^{(N)}_{N,N} = \left[ z \hat{I} - \hat{H}_N - \hat{T}_{N-1} \hat{G}^{(N-1)}_{N-1,N-1} \hat{T}_{N-1} \right]^{-1}, \]
\[ \hat{G}^{(N)}_{i,N} = \hat{G}^{(N-1)}_{i,N-1} \hat{T}_{N-1} \hat{G}^{(N)}_{N,N} , \]
\[ \hat{G}^{(N)}_{N,j} = \hat{G}^{(N)}_{N-1,j} \hat{T}_{N-1} \hat{G}^{(N)}_{N,N} , \]
\[ \hat{G}^{(N)}_{i,j} = \hat{G}^{(N-1)}_{i,j} + \hat{G}^{(N-1)}_{i,N-1} \hat{T}_{N-1} \hat{G}^{(N)}_{N,N} \hat{T}_{N-1} \hat{G}^{(N-1)}_{N-1,j} , \]
(3)
yield the Green’s functions of a system with the first \( N \) layers:
\[ \hat{G}^{(N)} = \begin{pmatrix} 
\hat{G}^{(N)}_{0,0} & \cdots & \hat{G}^{(N)}_{0,N-1} & \hat{G}^{(N)}_{0,N} \\
\vdots & \ddots & \vdots & \vdots \\
\hat{G}^{(N)}_{N-1,0} & \cdots & \hat{G}^{(N)}_{N-1,N-1} & \hat{G}^{(N)}_{N,N} \\
\hat{G}^{(N)}_{0,0} & \cdots & \hat{G}^{(N)}_{0,N-1} & \hat{G}^{(N)}_{0,N} 
\end{pmatrix}, \]
(4)
given those of a system with the first \( N - 1 \) layers,
\[ \hat{G}^{(N-1)} = \begin{pmatrix} 
\hat{G}^{(N-1)}_{0,0} & \cdots & \hat{G}^{(N-1)}_{0,N-1} \\
\vdots & \ddots & \vdots \\
\hat{G}^{(N-1)}_{N-1,0} & \cdots & \hat{G}^{(N-1)}_{N-1,N-1} 
\end{pmatrix}, \]
(5)
adding one layer at each iteration until we reach the entire system \((N = L)\), whose physical properties correspond to the FQNN outputs. In practice, one first solves for \( \hat{G}^{(N)}_{N,N} \) in the first line in Eq. 3, which is then plugged into the second to last lines.

Such iterative dependencies also allow us to utilize the chain rule and collectively obtain the differentials of the loss function \( \mathcal{L} \) with respect to the FQNN parameters implicitly in \( \hat{T} \) and \( \hat{H} \), whose definitions are in the main text. For convenience, we define the following real matrices [6]:
\[ T^{(N-1)} = \begin{pmatrix} 
\Re \left( \hat{T}_{N-1} \right) & \Im \left( \hat{T}_{N-1} \right) \\
-\Im \left( \hat{T}_{N-1} \right) & \Re \left( \hat{T}_{N-1} \right) 
\end{pmatrix}, \]
\[ H^{(N)} = \begin{pmatrix} 
\Re \left( \hat{H}_N \right) & \Im \left( \hat{H}_N \right) \\
-\Im \left( \hat{H}_N \right) & \Re \left( \hat{H}_N \right) 
\end{pmatrix}, \]
\[ O^{(N)} = \begin{pmatrix} 
\Re \left( \hat{G}^{(N)}_{0,0} \right) & \Im \left( \hat{G}^{(N)}_{0,0} \right) \\
-\Im \left( \hat{G}^{(N)}_{0,0} \right) & \Re \left( \hat{G}^{(N)}_{0,0} \right) 
\end{pmatrix} = \left( X^{(N)} \right)^{-1}, \]
\[ X^{(N)} = z \hat{I} - \hat{H}^{(N)} - \left( T^{(N-1)} \right)^T O^{(N-1)} T^{(N-1)} , \]
\[ Y^{(N)} = \begin{pmatrix} 
\Re \left( \hat{G}^{(N)}_{0,0} \right) & \Im \left( \hat{G}^{(N)}_{0,0} \right) \\
-\Im \left( \hat{G}^{(N)}_{0,0} \right) & \Re \left( \hat{G}^{(N)}_{0,0} \right) 
\end{pmatrix} = Y^{(N-1)} T^{(N-1)} O^{(N)} , \]
(6)
where \( X^{(N)} \), \( Y^{(N)} \) and \( O^{(N)} \) are the input and output of the \( N \)th iteration (first two lines in Eq. 3), with parametric dependence on the \( N \)th layer.

First, we consider the case where the FQNN outputs \( y \) are the LDOS over the \( M_L \) neurons in the last layer \( N = L \); see Eq. 4 in the main text. \( \mathcal{L} \) depends explicitly on \( y \), e.g., a mean-square or cross-entropy error loss function. With the chain rule, we have:
\[ \frac{\partial \mathcal{L}}{\partial T_{ij}}^{(N-1)} = \frac{\partial \mathcal{L}}{\partial O_{mn}} \frac{\partial O_{mn}}{\partial T_{ij}}^{(N-1)} = \epsilon^{(N)}_{mn} \delta_{mm,ij}, \]
\[ \frac{\partial \mathcal{L}}{\partial H_{ij}} = \frac{\partial \mathcal{L}}{\partial O_{mn}} \frac{\partial O_{mn}}{\partial H_{ij}} = \epsilon^{(N)}_{mn} \delta_{mm,ij}, \]
(7)
where:
\[ \epsilon^{(L)}_{mn} = \sum_{k=1}^{M_L} \frac{\partial \mathcal{L}}{\partial y_k} \frac{\partial y_k}{\partial O_{mn}} \]
\[ = \sum_{k=1}^{M_L} \frac{\partial \mathcal{L}}{\partial y_k} \left( \delta_{k+M_L,m} \delta_{kn} - \delta_{km} \delta_{k+M_L,n} \right) \pi \]
\[ = \frac{\partial \mathcal{L}}{\partial y_n} \left( \delta_{n+M_L,m} - \delta_{m+M_L,n} \right) \pi , \]
(8)
and the rest \((N < L)\) are obtainable via a second (backward) recursion:
\[ \epsilon^{(N)}_{mn} = \frac{\partial \mathcal{L}}{\partial O_{mn}} = \sum_{kl} \epsilon^{(N+1)}_{kl} \frac{\partial O_{mn}}{\partial O_{kl}} \]
\[ = \sum_{kl} \epsilon^{(N+1)}_{kl} \gamma_{kl,mn} , \]
(9)

We note that the values of \( \alpha, \beta, \) and \( \gamma \) necessary for 7 and 9 are fully determined in the previous recursion.
towards the FQNN outputs:
\[
\alpha_{mn,ij}^{(N)} = \frac{\partial O_{mn}^{(N)}}{\partial T_{ij}^{(N-1)}} = -\sum_p O_{mp}^{(N)} O_{jn}^{(N)} \left[ \left(T_{(N-1)}^{(N-1)}\right)^T O_{jn}^{(N)} \right]_{pi} + O_{mj}^{(N)} O_{pn}^{(N)} \left[ O_{(N-1)T_{(N-1)}}^{(N-1)} \right]_{ip},
\]
\[
\beta_{mn,ij}^{(N)} = \frac{\partial O_{mn}^{(N)}}{\partial H_{ij}^{(N)}} = O_{mi}^{(N)} O_{jn}^{(N)},
\]
\[
\gamma_{kl,mn}^{(N+1)} = \frac{\partial O_{mn}^{(N+1)}}{\partial O_{kl}^{(N)}} = \sum_{pq} O_{kp}^{(N+1)} O_{ql}^{(N+1)} \left( T_{ij}^{(N)} \right)^T T_{mn}^{(N+1)} T_{pq}^{(N+1)},
\]
where we have employed the following formulas [7]:
\[
\frac{\partial O_{mn}^{(N)}}{\partial X_{pq}^{(N)}} = -O_{mp}^{(N)} O_{qn}^{(N)},
\]
\[
\frac{\partial X_{pq}^{(N)}}{\partial T_{ij}^{(N-1)}} = \delta_{ij} \left[ \left(T_{(N-1)}^{(N-1)}\right)^T O_{jn}^{(N-1)} \right]_{pi} + \delta_{pj} \left[ O_{(N-1)T_{(N-1)}}^{(N-1)} \right]_{iq},
\]
\[
\frac{\partial X_{pq}^{(N+1)}}{\partial H_{ij}^{(N+1)}} = -\delta_{pi} \delta_{qj},
\]
Next, we consider the case where the FQNN output takes the form of the CC, Eq. 5 in the main text. Once again, we apply the chain rule to get:
\[
\frac{\partial L}{\partial T_{ij}^{(N-1)}} = \frac{\partial L}{\partial y} \sum_{kl} \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(L)}} \left[ \frac{\partial y_{uv}^{(L)}}{\partial y_{uv}^{(N)}} \frac{\partial y_{uv}^{(N)}}{\partial T_{ij}^{(N-1)}} \right]_{kl} + \frac{\partial L}{\partial y} \sum_{N=N-1}^{N=N} \left[ \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(N)}} \frac{\partial y_{uv}^{(N)}}{\partial O_{mn}^{(N)}} \frac{\partial O_{mn}^{(N)}}{\partial T_{ij}^{(N-1)}} \right]_{kl},
\]
\[
\frac{\partial L}{\partial H_{ij}^{(N-1)}} = \frac{\partial L}{\partial y} \sum_{kl} \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(L)}} \left[ \sum_{N=N-1}^{N=N} \left[ \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(N)}} \frac{\partial y_{uv}^{(N)}}{\partial O_{mn}^{(N)}} \frac{\partial O_{mn}^{(N)}}{\partial H_{ij}^{(N-1)}} \right]_{kl} \right],
\]
\[
\frac{\partial L}{\partial y} \sum_{kl} \frac{y_{kl}^{(L)}}{y_{ij}^{(L)}} \left[ \sum_{N=N-1}^{N=N} \theta_{kl,uv}^{(N)} \phi_{uv,mm}^{(N)} \gamma_{mn,ij}^{(N)} \right],
\]
where:
\[
\theta_{kl,uv}^{(L)} = \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(L)}} = \delta_{ku} \delta_{lv},
\]
\[
\phi_{uv,mm}^{(N)} = \frac{\partial y_{uv}^{(N)}}{\partial O_{mn}^{(N)}} = \delta_{vn} \sum_r y_{ur}^{(N-1)} T_{rm}^{(N-1)},
\]
and the rest (\(N < L \) for \(\theta \) and \(N < N' \) for \(\phi \)) are obtainable via (back-ward) recursions:
\[
\theta_{kl,uv}^{(N')} = \frac{\partial y_{kl}^{(L)}}{\partial y_{uv}^{(N')}} = \sum_{pq} \theta_{kl,pq}^{(N'+1)} \frac{\partial y_{pq}^{(N+1)}}{\partial y_{uv}^{(N')}}
= \sum_{qr} \theta_{kl,qr}^{(N'+1)} T_{qr}^{(N+1)},
\]
\[
\phi_{uv,mm}^{(N')} = \frac{\partial y_{uv}^{(N')}}{\partial O_{mn}^{(N)}} = \sum_{pq} \phi_{uv,pq}^{(N'+1)} \gamma_{pq,mm}^{(N+1)}.
\]
Interestingly, due to the extra summation of \(N' \) from \(N \) to \(L \) in Eq. 12, an earlier layer receives a concentration and boost in its corresponding differentials from multiple channels contributed by all later layers (Fig. 3), thus circumventing the gradient decay like in classical ANNs; see Fig. 5 in the main text. Such multi-purposes of gradients towards the earlier layers are also present in residual neural networks [8] via additional connectivity in ANN architecture. In comparison, even with local FQNN connectivity, the additional contributions rise naturally due to the CC’s global physical dependence.

VI. DETAILS ON MUTUAL INFORMATION AND LOCAL UNITARY TRANSFORMATIONS FOR LOGIC FLOW

Given a fermion tight-binding model, e.g., the FQNN Hamiltonian in Eq. 1 in the main text, we can analyze its ground-state entanglement entropy [9, 10] and mutual information [11, 12] starting from its two-point correlators.
$C_{ij} = \langle c_i^\dagger c_j \rangle$. With the correlation matrix $C_A$ defined in a subsystem $A$, the corresponding entanglement Hamiltonian $h_A$ is:

$$h_A = \{ \ln [ (1 - C_A) / C_A ] \}^T ,$$

which gives the reduced density matrix for the subsystem $A$ [13]:

$$\rho_A = \frac{1}{Z_A} \exp \left[ - \sum_{i,j \in A} (h_A)_{ij} c_i^\dagger c_j \right].$$

Since the partition function takes the form:

$$Z_A = \text{tr} \left( e^{-h_A} \right) = \prod_k \left( 1 + e^{-\varepsilon_k^A} \right),$$

$$\varepsilon_k^A = \ln \left[ (1 - \zeta_k^A) / \zeta_k^A \right],$$

where $\varepsilon_k^A$ and $\zeta_k^A$ are the respective eigenvalues of $h_A$ and $C_A$ [14, 15], we can straightforwardly derive the Von Neumann entanglement entropy we use in the main text as:

$$S_A = -\text{tr} (\rho_A \ln \rho_A) = \sum_k \left[ \frac{\varepsilon_k^A}{e^{\varepsilon_k^A} + 1} + \ln \left( 1 + e^{-\varepsilon_k^A} \right) \right],$$

which we can calculate from first $C_{ij}$ and then $\zeta_k^A$. As a side note, we can also evaluate the Renyi entanglement entropy as:

$$(S_A)_\alpha = \frac{1}{1 - \alpha} \ln \left[ \text{tr} (\rho_A^n) \right] = \frac{1}{1 - \alpha} \sum_k \ln \left( 1 + \alpha e^{\varepsilon_k^A} \right) - \alpha \ln \left( 1 + e^{-\varepsilon_k^A} \right).$$

For the FQNN in Eq. 1 in the main text, we first diagonalize the Hamiltonian:

$$\hat{H} = (c^\dagger Q) \Lambda (Q^\dagger c) = \alpha^\dagger \Lambda \alpha,$$

where $\Lambda = \text{diag} \{ \{ \epsilon_n \} \}$ are the energy eigenvalues and $\alpha_n^\dagger = \sum_m c_m^\dagger Q_{mn}$ are the corresponding eigenstates. Then, since the ground state fills (leaves empty) all states with $\epsilon_n \leq 0$ ($\epsilon_n > 0$):

$$|\Phi\rangle = \prod_{\epsilon_n \leq 0} \alpha_n^\dagger |0\rangle,$$

where $|0\rangle$ is the vacuum state, we can obtain the correlation matrix:

$$C = \left( Q \hat{A} Q^\dagger \right)^T ,$$

where $\hat{A}$ is the occupation matrix with 1’s in the diagonal for all $\epsilon_n \leq 0$.

In order to unambiguously locate the logic flow, we need to perform a unitary transformation $U$ on each layer so that the mutual information (Eq. 9 in the main text) vanishes between the last layer and as many neurons in that layer as possible. We note that the mutual information $I_{AB} = 0$ between two subsystems $A$ and $B$ if and only if the correlation matrix is block-diagonal:

$$C_{AB} = \begin{pmatrix} C_A & 0 \\ 0 & C_B \end{pmatrix},$$

and so does the entanglement Hamiltonian $h_{AB}$, yielding eigenvalues and entanglement entropy fully canceled by those from the separate subsystems $h_A$ and $h_B$. Physically, the subsystems $A$ and $B$ behave decoupled.

To preserve the layered FQNN architecture, we considered layer-wise unitary transformations, which take a block-diagonal form:

$$U = \begin{pmatrix} U_0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & U_{L-1} & 0 & \vdots \\ 0 & \cdots & 0 & 0 & U_L \end{pmatrix},$$

where $U_l$ is a unitary matrix of $M_l \times M_l$ acting on the $l^{th}$ layer. $U$ transforms the overall correlation matrix (and simultaneously the FQNN model) as follows:

$$C' = U^\dagger C U = \begin{pmatrix} C'_{0,0} & \cdots & \cdots & \cdots & C'_{0,L} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & C'_{L,L} & \cdots & \vdots \\ C'_{L,0} & \cdots & C'_{L,1} & \cdots & C'_{L,L} \\ C'_{L,L-1,0} & \cdots & C'_{L,L-1,1} & \cdots & C'_{L,L} \end{pmatrix},$$

where $C'_{l,l'} = U_l^\dagger C_{l,l'} U_{l'}$. Importantly, we wish to zero as many rows from the $M_l \times M_l$ matrix $C'_{l,L}$ as possible so that these neurons in the $l^{th}$ layer do not correlate with the last layer, diminishing their mutual information. Commonly, we have more hidden neurons than output neurons, $M_l > M_{L}$. For such purpose, we can transform $C'_{l,L}$ into an upper triangular matrix:

$$C'_{l,L} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,M_L} \\ 0 & a_{22} & \cdots & a_{2,M_L} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{M_L,M_L} \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$
by making $U_L = I$ an identity so that the output neurons retain their meanings and obtaining $U_l^*$ via the Gram-Schmidt orthogonalization. In the new FQNN after the overall unitary transformation $U$, only the first $M_L$ neurons in a layer may possess finite mutual information with the last layer. An example transformation yields the mutual information distributions and the decision logic flow shown in Fig. 4(a)(b) in the main text.

[1] A. Krogh and J. A. Hertz, A simple weight decay can improve generalization, in Proceedings of the 4th International Conference on Neural Information Processing Systems, NIPS’91 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991) p. 950–957.

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15, 1929–1958 (2014).

[3] A. Kitaev, Anyons in an exactly solved model and beyond, Annals of Physics 321, 2 (2006).

[4] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. 118, 216401 (2017).

[5] A. Croy, R. A. Römer, and M. Schreiber, Localization of electronic states in amorphous materials: Recursive green’s function method and the metal-insulator transition at $e \neq 0$, in Parallel Algorithms and Cluster Computing, edited by K. H. Hoffmann and A. Meyer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 203–226.

[6] W. Smith and S. Erdman, A note on the inversion of complex matrices, IEEE Transactions on Automatic Control 19, 64 (1974).

[7] K. B. Petersen and M. S. Pedersen, The matrix cookbook (2012), version 20121115.

[8] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016).

[9] L. Amico, R. Fazio, A. Osterloh, and V. Vedral, Entanglement in many-body systems, Rev. Mod. Phys. 80, 517 (2008).

[10] J. Eisert, M. Cramer, and M. B. Plenio, Colloquium: Area laws for the entanglement entropy, Rev. Mod. Phys. 82, 277 (2010).

[11] C. Adami and N. J. Cerf, von neumann capacity of noisy quantum channels, Phys. Rev. A 56, 3470 (1997).

[12] B. Groisman, S. Popescu, and A. Winter, Quantum, classical, and total amount of correlations in a quantum state, Phys. Rev. A 72, 032317 (2005).

[13] I. Peschel and V. Eisler, Reduced density matrices and entanglement entropy in free lattice models, Journal of Physics A: Mathematical and Theoretical 42, 504003 (2009).

[14] I. Peschel, Calculation of reduced density matrices from correlation functions, Journal of Physics A: Mathematical and General 36, L205 (2003).

[15] A. M. Turner, Y. Zhang, and A. Vishwanath, Entanglement and inversion symmetry in topological insulators, Phys. Rev. B 82, 241102 (2010).