

ARTICLE OPEN



Quantum generative adversarial networks with multiple superconducting qubits

Kaixuan Huang^{1,8}, Zheng-An Wang^{2,8}, Chao Song^{3,8}, Kai Xu², Hekang Li³, Zhen Wang³, Qiujiang Guo³, Zixuan Song³, Zhi-Bo Liu¹, Dongning Zheng^{2,4}, Dong-Ling Deng^{5,6}, H. Wang³, Jian-Guo Tian¹ and Heng Fan^{2,7}

Generative adversarial networks are an emerging technique with wide applications in machine learning, which have achieved dramatic success in a number of challenging tasks including image and video generation. When equipped with quantum processors, their quantum counterparts—called quantum generative adversarial networks (QGANs)—may even exhibit exponential advantages in certain machine learning applications. Here, we report an experimental implementation of a QGAN using a programmable superconducting processor, in which both the generator and the discriminator are parameterized via layers of single- and two-qubit quantum gates. The programmed QGAN runs automatically several rounds of adversarial learning with quantum gradients to achieve a Nash equilibrium point, where the generator can replicate data samples that mimic the ones from the training set. Our implementation is promising to scale up to noisy intermediate-scale quantum devices, thus paving the way for experimental explorations of quantum advantages in practical applications with near-term quantum technologies.

npj Quantum Information (2021)7:165; <https://doi.org/10.1038/s41534-021-00503-1>

INTRODUCTION

The interplay between quantum physics and machine learning gives rise to an emergent research frontier of quantum machine learning that has attracted tremendous attention recently^{1–4}. In particular, certain carefully designed quantum algorithms for machine learning, or more broadly artificial intelligence, may exhibit exponential advantages compared to their best possible classical counterparts^{2–8}. An intriguing example concerns quantum generative adversarial networks (QGANs)⁷, where near-term quantum devices have the potential to showcase quantum supremacy⁹ with real-life practical applications. Indeed, applications of QGANs with potential quantum advantages in generating high-resolution images^{10,11}, loading classical data¹², and discovering small molecular drugs¹³ have been investigated actively at the current stage.

The general framework of QGANs consists of a generator learning to generate statistics for data mimicking those of a true data set, and a discriminator trying to discriminate generated data from true data^{7,12,14–18}. The generator and discriminator follow an adversarial learning procedure to optimize their strategies alternatively and arrive at a Nash equilibrium point, where the generator learns the underlying statistics of the true data and the discriminator can no longer distinguish the difference between the true and generated data. In a previous QGAN experiment¹⁶, the generator is trained via the adversarial learning process to replicate the statistics of the single-qubit quantum data output from a quantum channel simulator. However, the implementation of quantum gradient, which is crucial for training QGANs⁶, is still absent. In addition, the involvement of entanglement in QGANs, which is a characterizing feature of quantumness and a vital

resource for quantum supremacy, has not yet been achieved during the learning process³.

In this article, we add these two crucial yet missing blocks by reporting an experiment realization of a QGAN based on a programmable superconducting processor with multiple qubits and all-to-all qubit connectivity previously reported in ref. ¹⁹. Superconducting qubits are a promising platform for realizing QGANs, owing to their flexible design, excellent scalability, and remarkable controllability. In our implementation, both the generator and discriminator are composed of multiqubit parameterized quantum circuits, also referred to as quantum neural networks in some contexts^{20–23}. Here, we benchmark the functionality of the quantum gradient method by learning an arbitrary mixed state, where the state is replicated with a fidelity up to 0.999. We further utilize our QGAN to learn a classical XOR gate, and the generator is successfully trained to exhibit a truth table close to that of the XOR gate.

RESULTS

Framework of QGAN algorithm

We first introduce a general recipe for our QGAN and then apply it to two typical scenarios: learning quantum states and the classical XOR gate. The overall structure of the QGAN is outlined in Fig. 1a, which includes an input label, a real source (R), a generator (G), a discriminator (D), and a quantum gradient subroutine. The input label sorts the training data stored in R, and also instructs G to generate data samples mimicking R. D receives the label and corresponding data samples from either G or R, and then evaluates with appropriate scores, based on which a loss function

¹The Key Laboratory of Weak Light Nonlinear Photonics, Ministry of Education, Teda Applied Physics Institute and School of Physics, Nankai University, 300457 Tianjin, China.

²Institute of Physics, Chinese Academy of Sciences, 100190 Beijing, China. ³Interdisciplinary Center for Quantum Information, State Key Laboratory of Modern Optical Instrumentation, and Zhejiang Province Key Laboratory of Quantum Technology and Device, Department of Physics, Zhejiang University, 310027 Hangzhou, China. ⁴CAS Center for Excellence in Topological Quantum Computation, University of Chinese Academy of Sciences, 100190 Beijing, China. ⁵Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, 100084 Beijing, China. ⁶Shanghai Qi Zhi Institute, 41th Floor, AI Tower, No. 701 Yunjin Road, Xuhui District, 200232 Shanghai, China. ⁷School of Physical Sciences, University of Chinese Academy of Sciences, 100190 Beijing, China. ⁸These authors contributed equally: Kaixuan Huang, Zheng-An Wang, Chao Song. ✉email: liuzb@nankai.edu.cn; dldeng@tsinghua.edu.cn; hfan@iphy.ac.cn

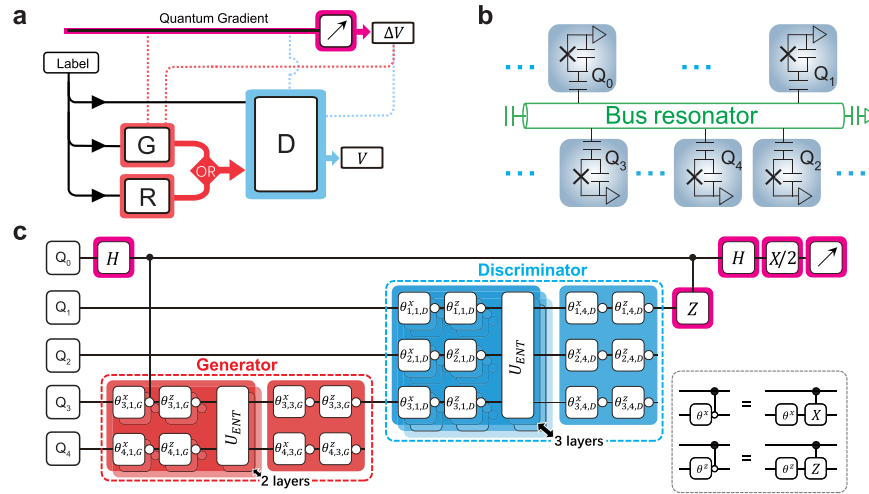


Fig. 1 QGAN algorithm and its implementation. **a** Overview of the QGAN logic. **b** Sketch of the superconducting processor used to implement the QGAN algorithm, where the five qubits, Q_0 – Q_4 , are interconnected by the central bus resonator. **c** An instance of the experimental sequences for fulfilling the QGAN algorithm to learn the classical XOR gate. Both G and D are parameterized quantum circuits consisting of n layers of the multiqubit entangling gate U_{ENT} and $n + 1$ single-qubit rotations $\theta_{j,l,m}^{x/z}$, where $l = n + 1$ is the layer index of single-qubit rotations, $m \in \{G, D\}$, and the superscript (x or z) refers to the axis in the Bloch sphere around which the state of Q_j is rotated by the angle θ . Shown is the training sequence on G to optimize its parameter $\theta_{3,1,G}^x$ based on the quantum gradient subroutine, which includes two Hadamard gates (H), two controlled rotation gates, and a $\pi/2$ rotation around x -axis ($X/2$). In this instance, both Q_1 – Q_2 and Q_3 – Q_4 store the input label, and D's output score is encoded in Q_1 by $S_n^{\text{D,R/G}} = \langle \sigma_1^z \rangle / 2 + 1/2$, which can be obtained by directly measuring Q_1 . Q_0 's $\langle \sigma_0^z \rangle$ gives the score derivative with respect to the rotational angle parameter right before the first controlled rotation gate, i.e., $\partial \langle \sigma_1^z \rangle / \partial \theta_{3,1,G}^x = -\langle \sigma_0^z \rangle$ for the sequence displayed here. The first controlled rotation gate can be either a controlled X (CNOT) or controlled Z (CZ) gate depending on the single-qubit rotation axis it follows as shown in the lower right box.

V is constructed to differentiate between R and G. The adversarial training procedure is repeated in conjugation with the quantum gradient subroutine, which yields the partial derivatives of V with respect to the parameters constructing D or G, so as to maximize V for the optimal configuration of D and to minimize V for the optimal G in terms of the chosen values of the constructing parameters.

Specifically, denoting the sets of parameters constructing G and D as θ_G and θ_D , respectively, the loss function V is written as

$$V = \frac{1}{N} \sum_{n=1}^N \left[S_n^{\text{D,R}}(\vec{\theta}_D) - S_n^{\text{D,G}}(\vec{\theta}_D, \vec{\theta}_G) \right],$$

where N is the total number of data samples selected for training and $S_n^{\text{D,R}}$ ($S_n^{\text{D,G}}$) represents the score of the n th data sample from R (G) evaluated by D. G and D are trained alternately with D being trained first. In D's turn, we maximize the loss function by iteratively optimizing $\vec{\theta}_D$ according to $\vec{\theta}_D^{i+1} = \vec{\theta}_D^i + \alpha_D \nabla_{\vec{\theta}_D} V$, where i is the iteration step index and $\nabla_{\vec{\theta}_D} V$ denotes the gradient vector of the loss function at the i th step; we train G by minimizing the square of the loss function with an iteration relation $\vec{\theta}_G^{i+1} = \vec{\theta}_G^i - \alpha_G \nabla_{\vec{\theta}_G} V^2$. The learning rates can be adjusted by tuning α_D and α_G which are typically on the order of unity.

An important quantity that plays a vital role in training our QGAN is the gradient of the loss function with respect to a given parameter. Interestingly, owing to the special structures of our quantum circuits for the generator and the discriminator, a common approach to obtain such a gradient is to shift the corresponding parameter by $\pm\pi/2$ and measure the loss function at the shifted values²⁴. In our experiment, we employ an effective quantum method—the Hadamard test quantum algorithm²⁵—to obtain the gradient for the first time. This algorithm can reduce half of the time in comparison with the common approach to finish the training, at the expense of an additional auxiliary qubit

and two controlled gates (see Supplementary Note 4 for more details). At each round of the training process, the parameters of the discriminator (or generator) are updated simultaneously after the gradients for all parameters are obtained, and a quantum process tomography (QPT) is performed to characterize the overlap fidelity between the generated data based on the updated parameters and the true data.

Experimental implementation of QGAN

The above mentioned QGAN is experimentally realized on a superconducting quantum processor using five frequency-tunable transmon qubits labeled as Q_j for $j = 0$ – 4 , where all qubits are interconnected by a central bus resonator as illustrated in Fig. 1b. The role arrangement of Q_0 – Q_4 can be visualized by the exemplary experimental sequence shown in Fig. 1c. Q_0 is to assist the quantum gradient subroutine. Q_1 – Q_2 stores the label which is passed to D, with the output score encoded in Q_1 by $S_n^{\text{D,R/G}} = \langle \sigma_1^z \rangle / 2 + 1/2$. For the experimental instances with G inserted, Q_3 – Q_4 also stores the label as the input to G and the output data sample from G is passed to D via Q_3 ; for the experimental instances with R in replacement of G, only Q_3 stores the data sample from R as designated by the label. Details of the device parameters can be found in the “Methods” section and Supplementary Note.

By tuning the qubits on resonance but detuned from the bus resonator in frequency, these qubits can be all effectively connected, which enables the flexible realizations of the multi-qubit entangling gates among arbitrarily selected qubits. The all-to-all interactions are described in the dispersive regime by the effective Hamiltonian $H_1 = \sum \lambda_{jk} (\sigma_j^+ \sigma_k^- + \sigma_j^- \sigma_k^+)$, where σ_j^+ (σ_j^-) denotes the raising (lowering) operator of Q_j , and λ_{jk} is the effective coupling strength between Q_j and Q_k mediated by the bus resonator. Evolution under this Hamiltonian for an interaction time τ leads to the entangling operator with the form of $U_{\text{ENT}} = e^{-iH_1\tau}$, which can steer the interacting qubits into highly entangled state. In our QGAN, the parameterized quantum

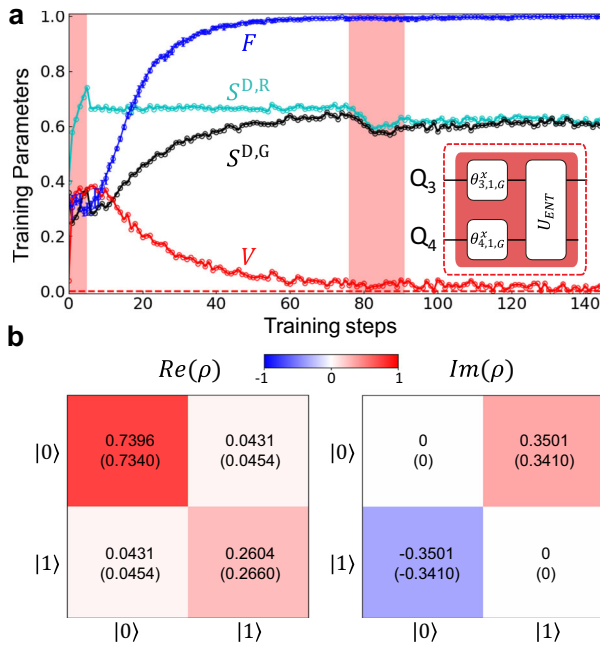


Fig. 2 QGAN performance in learning a mixed state. **a** Tracking of the loss function V , the output scores $S^{D,R/G}$, and the state fidelity between R/G's output states F during the adversarial training procedure with learning rates $\alpha_D = 0.8$ and $\alpha_G = 0.6$. The alternate training stages of D and G are marked by red and white regions, respectively. In each stage, the maximum step number is limited to 50 for D and 100 for G. Inset: the quantum circuit for generating ρ_R for Q_3 . The same circuit is also used for G, with random initial guesses for the two rotational angles which are then optimized. **b** Elements of the real and imaginary parts of the output density matrices of R(G). The numbers in the parentheses denotes the corresponding values for G.

circuits that comprise G and D leverage the naturally available multiqubit U_{ENT} s, with the interaction time of the two-qubit U_{ENT} fixed at around 50 ns for G and the three-qubit one fixed at around 55 ns for D. We stress that, compared to architectures with limited connectivity^{8,9,26}, the all-to-all connectivity for the device used in this experiment could reduce the total circuit depths (hence the running time) in implementing both the generator and discriminator, which is a crucial merit given that the coherence time is limited.

As laid out in Fig. 1c, the entangling operators U_{ENT} are interleaved with the single-qubit X and Z rotations, which successively rotate Q_j around x - and z -axis in the Bloch sphere by angles of $\theta_{j,l,m}^x$ and $\theta_{j,l,m}^z$, where l is the layer index and $m \in \{G, D\}$. The lengths of the X and Z rotations are fixed at 30 and 20 ns, respectively. Taking into account the experimental imperfections, we perform numerical simulations to decide the depths of the interleaved layers consisting of U_{ENT} and the single-qubit rotations, for a balance between the learning fidelity and efficiency, see Supplementary Discussion for the discussion of the circuit depth. For example, to learn the XOR gate, the circuit layer depths are set to be $n = 2$ and 3 for G and D, respectively, as shown in Fig. 1c.

The QGAN learning process is guided by the gradient of the loss function with respect to θ_G and θ_D . To obtain these gradients, we adopt the method of Hadamard test^{14,25}, which is illustrated in the sequence instance in Fig. 1c. For the partial derivative with respect to the parameter $\theta_{j,l,m}^x$ ($\theta_{j,l,m}^z$), we insert the first controlled- X (Z) gate right after the single-qubit X (Z) rotation containing this parameter, with Q_j as the target. The second controlled- Z gate is

applied at the end of the training sequence with Q_1 as the target. The partial derivative of Q_1 's $\langle \sigma_1^z \rangle$, which relates to D's output score, is given by $\partial \langle \sigma_1^z \rangle / \partial \theta_{j,l,m}^{x(z)} = -\langle \sigma_0^z \rangle$, which can be directly obtained by measuring Q_0 and used in computing the gradient of the loss function. More details about the experimental realizations of the controlled- X (Z) gates, as well as the theoretical and experimental verifications of the quantum gradient method are presented in Supplementary Note 4.

Learning an arbitrary single-qubit quantum state

To benchmark the functionality of the quantum gradient method and the learning efficiency of our QGAN circuit, we first train an arbitrary mixed state as data which is a simulation of quantum channel. As shown in the inset of Fig. 2a, the mixed state for Q_3 reads $\rho_R = \begin{pmatrix} 0.7396 & 0.0431 + 0.3501i \\ 0.0431 - 0.3501i & 0.2604 \end{pmatrix}$, which is generated by applying two single-qubit X rotations (with the rotation angles of 1.35 on Q_3 and 0.68 on Q_4) followed by the U_{ENT} gate on Q_3 and Q_4 . Correspondingly, G is set up with a single layer and two parameters describing the single-qubit X rotation angles during the training, while D remains the one shown in Fig. 1c with 3 layers and all 18 parameters being trained. The trajectories of the loss function and scores of data from R/G during the training process are recorded and plotted in Fig. 2a. We optimize D at the beginning of the training to enlarge the distance between $S^{D,R}$ and $S^{D,G}$. At the end of this turn, D can discriminate datasets from R and G with the maximum probability. In G's turn, $S^{D,G}$ moves towards $S^{D,R}$ which means that G is learning the behavior of R. Each turn ends when the optimal point of the loss function is reached, or the iteration number goes over a preset limit. As the adversarial learning process goes on, the value of the loss function oscillates from turn to turn and eventually converges to 0 indicating that the learning arrives at a Nash equilibrium point, where G is able to produce a mixed state ρ_G which resembles ρ_R and D can no longer distinguish between them^{7,14}. The training process is characterized by the similarity between datasets generated by G and R, which is quantified by the state fidelity $F(\rho_R, \rho_G) = \text{Tr}(\sqrt{\rho_R \rho_G} \sqrt{\rho_R})$. As shown in Fig. 2a, The average fidelity increases rapidly with the iteration steps, indicating the effectiveness of the adversarial learning. The density matrices ρ_R and the final ρ_G are plotted in Fig. 2b, which yields a state fidelity of around 0.999. We mention that the QGAN demonstrated in our experiment is distinct from quantum tomography or state preparation in essential ways. Here, we carry out the tomography process merely to benchmark the performance of the QGAN. In practical applications of QGAN with more qubits, this tomography process is not necessary and one may use other more efficient approaches to measure the performance.

Learning the statistics of an XOR gate

Now we apply the recipe to train the QGAN to replicate the statistics of an XOR gate, which is a classical gate with input-output rules $00 \rightarrow 0$, $01 \rightarrow 1$, $10 \rightarrow 1$, and $11 \rightarrow 0$. The input bit values store as labels in Q_1 and Q_2 , which is performed to let the discriminator know the corresponding input state of the state generated by the generator. We use the computational basis state $|0\rangle$ and $|1\rangle$ to encode the classical data 0 and 1, respectively. We randomly initialize the parameters of both D and G, and then update them alternately following the rules outlined above. The trajectories of the key parameters benchmarking the QGAN performance during the training process are plotted in Fig. 3a, with the evolutions of two representative parameters constructing D and G shown in Fig. 3b. Again, the loss function exhibits a typical oscillation during the adversarial process, and the training reaches its equilibrium after about 190 steps with an average state fidelity of 0.927. In addition, after training G successfully exhibits a

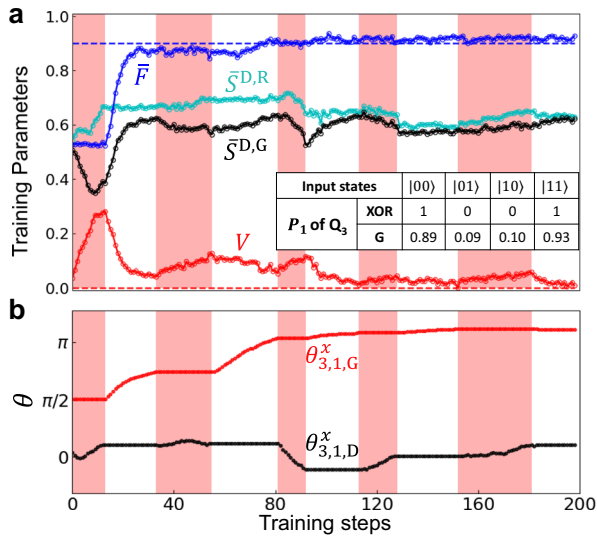


Fig. 3 QGAN performance in learning the XOR gate. **a** Tracking of the loss function V , the averaged output scores $\bar{S}^{D,R/G}$, and the averaged state fidelity between R/G's output states \bar{F} during the adversarial training process with learning rates $\alpha_D = 1.0$ and $\alpha_G = 1.5$. The blue dashed line denotes the training fidelity of 0.9. The maximum step number is limited to 50 for both G and D. $\bar{S}^{D,R/G}$ and \bar{F} are averaged over four possible inputs. Inset: The truth table of G after training in comparison with that of the XOR gate. **b** Trajectories of two representative parameters constructing D and G, $\theta_{3,1,D}^x$ and $\theta_{3,1,G}^x$, during the QGAN training. All parameters in G and D are initialized randomly between 0 and π and optimized alternately during the training.

truth table close to that of an XOR gate, as shown in the inset of Fig. 3a. Note that the fidelity of trained XOR gate is lower than the fidelity of the above mentioned single qubit mixed state. Due to the complexity of learning task, generator consisting of more layers and parameters is employed for training XOR gate, which would introduce more two-qubit gate infidelities and decoherence error during the learning.

DISCUSSION

We have experimentally implemented a multi-qubit QGAN equipped with a quantum gradient algorithm on a programmable superconducting processor. The results demonstrate the feasibility of QGAN in learning data with both classical and quantum statistics for small system sizes. The parameterized quantum circuits for constructing quantum generators and discriminators do not require accurate implementations of specific quantum logics and can be achieved on the near-term quantum devices across different physical platforms. QGAN has far-reaching effects in solving the quantum many-body problem, which can directly extend to the optimal control and self-guided quantum tomography, especially when the system size goes large^{27,28}. Our implementation paves the way to the much-anticipated computing paradigm with combined quantum-classical processors, and holds the intriguing potential to realize practical quantum supremacy⁹ with noisy intermediate-scale quantum devices²⁹.

We note that there might be challenges with the random initialization and hardware efficient forms of G and D with large depth. For instance, for larger circuits we may need larger depths to gain enough representation power. As a result, the coherence time in our experiment needs to be increased. Moreover, another possible challenge for larger circuits is the so called “barren plateaus” problem, namely that the gradient might be vanishing for most of the parameter regions (see ref. 20–23 for more details). In fact, these

challenges are the common ones facing most of the current experiments on variational quantum circuits. We also would like to mention that when scale beyond a handful of qubits, G and D can be replaced by deep quantum neural networks³⁰ and the corresponding gradient may as well be efficiently evaluated by the backward propagation algorithm³¹, similar to the case of training classical deep neural networks.

METHODS

Details about the quantum device

Our experimental device is a superconducting circuit consisting of 20 transmon qubits interconnected by a central bus resonator with frequency fixed at around $\omega_R/2\pi \approx 5.51$ GHz. Five qubits, denoted as Q_j for $j = 0-4$, are actively used in this work. The frequencies of the used qubits are carefully arranged to minimize any possible unwanted interactions and crosstalk errors among qubits during single-qubit operations. Each qubit has its own microwave control and flux bias lines for implementations of XY and Z rotations, respectively. Meanwhile, each qubit is dispersively coupled to its own readout resonator for qubit-state measurement, and all the qubits can be measured simultaneously using the frequency-domain multiplexing technique.

Multi-qubit entangling gates generation

The multi-qubit entangling gates that comprise G and D in our QGANs are generated by tuning all the involved qubits on-resonance at around 5.165 GHz, which is detuned from the resonator frequency by 345 MHz. Single-qubit phase gates, which are realized by amplitude-adjustable Z square pulses with a width of 20 ns, are added on each qubit before and after the interaction process to cancel out the dynamical phases accumulated during it. In our experiment, the two- and three-qubit entangling gates contain Q_3 , Q_4 and Q_1 , Q_2 , Q_3 respectively. The characteristic interaction time t_{gate} is fixed at $\pi/4|\lambda|$, where λ is negative and approximately equals to \tilde{g}/Δ . \tilde{g} is defined as the average qubit-resonator coupling strength and Δ denotes the detuning between the interaction frequency and resonator frequency. $\lambda/2\pi$ are around -2.48 and -2.27 MHz for the two- and three-qubit cases, respectively.

DATA AVAILABILITY

The raw experimental data for generating the plots in this paper are available upon reasonable request.

Received: 12 July 2021; Accepted: 9 November 2021;

Published online: 01 December 2021

REFERENCES

- Carleo, G. et al. Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019).
- Sarma, S. D., Deng, D.-L. & Duan, L.-M. Machine learning meets quantum physics. *Phys. Today* **72**, 48–54 (2019).
- Biamonte et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
- Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep. Prog. Phys.* **81**, 074001 (2018).
- Gao, X., Zhang, Z.-Y. & Duan, L.-M. A quantum machine learning algorithm based on generative models. *Sci. Adv.* **4**, eaat9004 (2018).
- Harrow, A. & Napp, J. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Phys. Rev. Lett.* **126**, 140502 (2021).
- Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**, 040502 (2018).
- Havlicek, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
- Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
- Rudolph, M. S., Bashige, N. T., Katarbarwa, A., Johr, S. & Peropadre, B. Generation of high resolution handwritten digits with an ion-trap quantum computer. Preprint at *bioRxiv* <https://arxiv.org/abs/2012.03924> (2020).
- Huang, H.-L. et al. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.* **16**, 024051 (2021).

12. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Inf.* **5**, 103 (2019).
13. Li, J., Topaloglu, R. & Ghosh, S. Quantum generative models for small molecule drug discovery. Preprint at *bioRxiv* <https://arxiv.org/abs/2101.03438> (2021).
14. Dallaire-Demers, P.-L. & Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **98**, 012324 (2018).
15. Zeng, J., Wu, Y., Liu, J.-G., Wang, L. & Hu, J. Learning and inference on generative adversarial quantum circuits. *Phys. Rev. A* **99**, 052306 (2019).
16. Hu, L. et al. Quantum generative adversarial learning in a superconducting quantum circuit. *Sci. Adv.* **5**, eaav2761 (2019).
17. Romero, J. & Aspuru-Guzik, A. Variational quantum generators: generative adversarial quantum machine learning for continuous distributions. *Adv. Quantum Technol.* **4**, 2000003 (2021).
18. Anand, A., Romero, J., Degroote, M. & Aspuru-Guzik, A. Experimental demonstration of a quantum generative adversarial network for continuous distributions. Preprint at *bioRxiv* <https://arxiv.org/abs/2006.01976> (2020).
19. Song, C. et al. Generation of multicomponent atomic Schrödinger cat states of up to 20 qubits. *Science* **365**, 574–577 (2019).
20. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).
21. Cerezo, M., Sone, A., Volkoff, T. & Coles, P. J. Cost function dependent barren plateaus in shallow quantum neural networks. *Nat. Commun.* **12**, 1791 (2021).
22. Skolik, A., McClean, J. R., Mohseni, M., Smagt, P. & Leib, M. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.* **3**, 5 (2021).
23. Huembeli, P. & Dauphin, A. Characterizing the loss landscape of variational quantum circuits. *Quantum Sci. Technol.* **6**, 025011 (2021).
24. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A* **99**, 032331 (2019).
25. Mitarai, K. & Fujii, K. Methodology for replacing indirect measurements with direct measurements. *Phys. Rev. Res.* **1**, 013006 (2019).
26. Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
27. Chapman, R. J., Ferrie, C. & Peruzzo, A. Experimental demonstration of self-guided quantum tomography. *Phys. Rev. Lett.* **117**, 040402 (2016).
28. Rambach, M. et al. Robust and efficient high-dimensional quantum state tomography. *Phys. Rev. Lett.* **126**, 100402 (2021).
29. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
30. Beer, K. et al. Training deep quantum neural networks. *Nat. Commun.* **11**, 808 (2020).
31. Goncalves, C. Quantum neural machine learning: backpropagation and dynamics. *NeuroQuantology* **15**, 22 (2016).

ACKNOWLEDGEMENTS

Devices were made at the Nanofabrication Facilities at Institute of Physics in Beijing and National Center for Nanoscience and Technology in Beijing. The experiment was performed on the quantum computing platform at Zhejiang University. This work was supported by National Basic Research Program of China (Grants Nos. 2017YFA0304300, 2016YFA0302104 and 2016YFA0300600), National Natural Science

Foundation of China (Grants Nos. 11934018 and 11725419), the Zhejiang Province Key Research and Development Program (Grant No. 2020C01019), the start-up fund from Tsinghua University (Grant No. 53330300320), the Shanghai Qi Zhi Institute, and Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB28000000).

AUTHOR CONTRIBUTIONS

Z.A.W., D.L.D., and H.F. proposed the idea. K.H., C.S., K.X., and Q.G. conducted the experiment, with help from Z.B.L., J.G.T., and H.W.; K.H. and Z.A.W. performed the numerical simulation, with help from H.F.; H.L. and D.Z. fabricated the device; K.H., Z.A.W., C.S., Z.W., D.L.D., H.W., and H.F. cowrote the manuscript. All authors contributed to the experimental setup, discussions of the results, and development of the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41534-021-00503-1>.

Correspondence and requests for materials should be addressed to Zhi-Bo Liu, Dong-Ling Deng or Heng Fan.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021