Active learning on a programmable photonic quantum processor

Chen Ding, Xiao-Yue Xu, Yun-Fei Niu, Shuo Zhang, He-Liang Huang and Wan-Su Bao

1 Henan Key Laboratory of Quantum Information and Cryptography, Zhengzhou, Henan 450000, People's Republic of China
2 Hefei National Laboratory, University of Science and Technology of China, Hefei 230088, People's Republic of China
3 The first two authors contributed equally.

* Authors to whom any correspondence should be addressed.
E-mail: quanhhl@ustc.edu.cn and bws@qiclab.cn

Keywords: quantum machine learning, linear optical quantum computing, active learning

Abstract
Training a quantum machine learning model generally requires a large labeled dataset, which incurs high labeling and computational costs. To reduce such costs, a selective training strategy, called active learning (AL), chooses only a subset of the original dataset to learn while maintaining the trained model’s performance. Here, we design and implement two AL-powered variational quantum classifiers to investigate the potential applications and effectiveness of AL in quantum machine learning. Firstly, we build a programmable free-space photonic quantum processor, which enables the programmed implementation of various hybrid quantum-classical computing algorithms. Then, we code the designed variational quantum classifier with AL into the quantum processor, and execute comparative tests for the classifiers with and without the AL strategy. The results validate the great advantage of AL in quantum machine learning, as it saves at most 85% labeling efforts and 91.6% percent computational efforts compared to the training without AL on a data classification task. Our results inspire AL’s further applications in large-scale quantum machine learning to drastically reduce training data and speed up training, underpinning the exploration of practical quantum advantages in quantum physics or real-world applications.

1. Introduction

The fastly developing quantum machine learning algorithms could help solve numerous physical and real-world data analysis issues [1–10], and their potential computational advantages over classical counterparts have been investigated and even proven in a variety of theoretical and experimental works [11–24]. Despite the fact that the future subversion of quantum machine learning applications is rapidly approaching, there are still issues that need to be resolved for large-scale practical applications. One of the remaining problems is the high cost of labeling data items for supervised learning applications [25–27]. Since the labeling task requires finding a certain amount of answers to the question-to-solve, such work usually involves inefficient data collection or large human labor, especially for quantum machine learning models. In applications like training quantum autoencoder [28, 29], learning quantum dynamics [10, 30] and solving real-world problems with powerful variational quantum models [8, 9, 19, 31–33], preparing quantum states and labeling quantum state or quantum processes usually consumes huge physical resources. This drawback severely hurdles the further development of their practical applications.

Active learning (AL) [34–36] is an effective method to resolve the above problem. Instead of acquiring all the labeled data to do the training job, AL strategies enable the model to query only a subset of the original data, while keeping the capability to faithfully extract the most critical features from the whole dataset.

In this paper, we implement two AL-powered variational quantum classifiers on a programmable photonic quantum processor. The whole system is built by high-quality single-photon sources, and circuit training and measurement devices connected to a control server, enabling fully programmed and
high-precision manipulation (the average fidelity of the programmable single-qubit gate is 99.88\%\) of the
expectation measurement, gradient evaluation and circuit training in the quantum processor. With the AL
strategies coded into the program, we apply them to the variational quantum classifier and the
nonlinearity-enhanced variational quantum classifier (NEVQC). Our comparative results confirm the
efficacy of AL in reducing the cost of quantum machine learning, and its robust adaptability to different
quantum machine learning models. Furthermore, our techniques also demonstrate a path towards optical
implementation of various types of hybrid quantum–classical algorithms in free-space photonic quantum
processors.

2. AL

AL is a type of strategies that applies selective technique to reduce the training set size, while preserving the
performance of the original machine learning algorithm. Generally, we can view the training process of AL as
the cooperation work of a learner and a labeler, as shown in figure 1. The learner is familiar with the model,
the dataset, but unaware of their labels. Meanwhile, the labeler owns a labeling oracle and knows nothing
else. The training works as follows:

1. In the beginning, the labeler sends a few labeled data items to the learner. The learner builds a prototype of
   the model using the given data as the training set.
2. By a certain selecting strategy, the learner selects one data item from the unlabeled data pool to query the
   labeler, then adds the labeled item to the training set.
3. The learner trains the model with the update labeled training set.
4. Repeat Step 2 and 3 until convergence.

To ensure effectiveness, the most representative data items could be extracted by some advanced selecting
strategy, including uncertainty sampling (USAMP) \cite{37–39}, Query-by-Committee (QBC) \cite{40}, evaluating
expected model change \cite{41}, error reduction \cite{42}, etc. Those methods assess the informativeness of data
items from different perspectives. Among them, two most typical techniques \cite{43}, USAMP and QBC, are
chosen for our AL experiments and thus will be briefly introduced below.

Denote the training set as \(\{x_i\}_i=1,\ldots,m\), their corresponding labels as \(y_i\) in set \(\mathcal{L}\). The USAMP
method evaluates each data item’s uncertainty for model \(M_\theta\) as
\[
U(x_i) = -\max_{y \in \mathcal{L}} P(M_\theta(x_i) = y),
\]
where \(P(\cdot)\) is the probability measure, and select the data item with most uncertainty to query the labeler.

The QBC method appoints a committee of several machine learning models, noted as \(M_1,\ldots,M_C\). For
each data item, the committee members first guess its label separately, and then take a vote. The final
selection goes to the most divergent item with the highest vote entropy
\[
E(x_i) = -\sum_j V(y_j|x_i) \cdot \ln \frac{V(y_j|x_i)}{C},
\]
where \(V(y_j|x_i) = \sum_{k=1}^C \delta_{M_k(x_i),y_j}\) denotes to the number of votes for label \(y_j\) in the committee.

In our experiment, ‘learner’ contains both quantum processor and classical processor, while ‘labeler’
refers to the labeling function coded in the classical processor. We next introduce our experiment
implementation and results.

3. Experiment and results

We implement the AL experiments on a programmable free-space photonic quantum processor (see
figure 2). The experimental setup consists of two main modules, data encoding and variational quantum
circuits (VQCs), both of which are programmable. Two types of VQCs with different capabilities are
implemented in our experiments for the classification task. The results show that AL strategy has strong
adaptability in different machine learning models, and can greatly reduce the amount of data and
computational cost.
Figure 1. The schematic diagram of active learning. From the unlabeled data pool, the learner sequentially selects a most representative data item, by evaluating the data uncertainty (USAMP) or consulting a committee of machine learning models (QBC), to query with the labeler. The labeler answers each query with the true label. The learner then trains the model for a short time period, and continues to select a new data item to learn until convergence.

Figure 2. The experiment design and implementation. (a) Three patterns of dividing the data items on an arc, designed to variate the difficulty of the classification task for variational quantum classifier. (b) The circuit of our variational quantum classifier (VQC) and the nonlinearity-enhanced variational quantum classifier (NEVQC). The nonlinearity, enhancing the performance of VQC on the data dividing patterns, is introduced by an entangling operations followed by postselecting the ancilla photon as |0⟩. (c) The experimental setups on the photonic quantum processor. The implementations of data encoding (introduced in section 3.1), VQC (section 3.2), NEVQC (section 3.3) are placed on three boards, connected through optical fibers. The photon detectors (noted in white numbers as 1–6) and electric rotation stages are also connected to the classical computer, automating the process of expectation measurement, gradient evaluation, and circuit training. The data photon and ancilla photon are noted as red letters as a and b. On these boards, all photons are spectrally filtered with 3 nm bandwidth filters. Devices: C-BBO, sandwich-like BBO + HWP + BBO combination; QWP, quarter-wave plat; SC-YVO4, YVO4 crystal for spatial compensation; TC-YVO4, YVO4 crystal for temporal compensation; Babinet, Soleil–Babinet compensators.

3.1. Data encoding

As shown in figure 2(a), normalized vectors (cos xi, sin xi) distributed on an arc of length π are taken as the data for classification. These vectors are divided into two classes, labeling as yi = ±1 (marked red and blue, respectively). To variate the difficulty of the classification, three patterns of dividing these vectors are designed. We introduce the difficulties of the classification on the three patterns in appendix D.
Figure 3. The calibration of HWP during long-term automatic rotation. (a) The response curve of HWP during 10,000 step rotation. We constantly rotate the HWP from angle 0 to 100π with step size 0.01π. A single photon of horizontal polarization state passes through the HWP and is measured for 0.1 s after each rotation step. The dotted line shows the theoretical $|H\rangle$ probability in the photon, while the circles show the experimental measurement results, with around 2000 shots. We abbreviate the middle segment of the response curve for clarity of the result. (b) The distribution of calibrated gate infidelities of the HWP during the long-term rotation. The average infidelity is around 0.12%. Our calibration results show a high fidelity and stability of the single-qubit gate formed by HWP during long-term automatic rotation.

To encode these data into quantum states with high fidelity, we first build a single-qubit photon source. As shown on the Board I in figure 2(c), laser pulses with a central wavelength of 390 nm, pulse duration of 150 fs, and repetition rate of 80 MHz pass through a half-wave plate (HWP) sandwiched by two β-barium borate (BBO) crystals. By the spontaneous parametric down-conversion (SPDC) process, entangled photon pairs of visibility more than 100 : 1 are produced on the two sides. The two photons are in an entangled state $(|HV\rangle + |VH\rangle)/\sqrt{2}$, where $|H\rangle$ represents the horizontal polarization and $|V\rangle$ represents the vertical polarization. We also note $|H\rangle$ as $|0\rangle$, and $|V\rangle$ as $|1\rangle$.

Then, a polarizing beam splitter (PBS) on one side postselects the photons as $|HV\rangle$, which disentangled them. We denote the photon of horizontal polarization as the ‘data photon’, and the other photon as the ‘ancilla photon’. A single-qubit gate formed by a HWP at angle $x/2$ (denote by $\text{HWP}(x)$) encodes the data to the photon as

$$
\begin{pmatrix}
\cos x & \sin x \\
\sin x & -\cos x
\end{pmatrix} |H\rangle = \cos x |H\rangle + \sin x |V\rangle.
$$

(3)

The whole dataset $\{(|\cos x_i|, |\sin x_i|)\}_{i=1,...,m}$ is encoded to the classifier successively. In a single epoch, each angle $x_i$ is visited by the HWP for a short duration, to evaluate a separate loss function or gradient. Then a classical computer will sum all the terms together and yield the parameters-to-be in the next epoch. Since the size of dataset is generally large in machine learning, a heavy HWP rotation work is brought. For instance, evaluating the loss function for 1000 epochs with dataset size of 100 will need about $10^5$ rotations of distance $500\pi$ on the encoding HWP, making manual experiment impossible, not to mention the subsequent low operation accuracy.

To overcome this problem and accurately encode large datasets, we automate the rotation of the HWP by installing it into electric rotation stage and leave the large rotation work to a classical computer controlling the stage. We calibrate the response curve of the HWP to investigate their fidelity and stability during long-term electronically controlled rotations. The results in figure 3 show the HWP’s response is in good agreement with the theoretical estimation during long-term automatic rotation, with the average fidelity estimated by the squared statistical overlap [44]

$$
\bar{F} = \frac{1}{10000} \sum \left( 1 - \frac{\sqrt{p_i q_i} + \sqrt{(1-p_i)(1-q_i)}}{2} \right)^2,
$$

as 99.88% ($p_i, q_i$ are the theoretical and experimental $|H\rangle$ probability in the outcome photon, respectively, during the 10,000 step rotation.). Our high-performance technique makes possible the training of large quantum machine learning models, as we also apply it to the iterations of ansatz parameters in model training.
3.2. Variational quantum classifier (VQC)

3.2.1. Ansatz
For classifying the one-qubit data, we employ single-qubit parameterized quantum circuit consisting of a trainable single-qubit gate formed by a HWP, denoted by \( \text{HWP}(\theta) \), as the trainable ansatz, as shown on the top panel in figure 2(b). The variational ansatz applies on the input states and analyzes the encoded information. The output states are all measured in Pauli-Z basis. The classification rule for state \(|x\rangle\) is set as

\[
\langle Z(x, \theta) \rangle > 0 \rightarrow 1 \\
\langle Z(x, \theta) \rangle < 0 \rightarrow -1,
\]

in which \( V(\cdot) \) is the parameterized quantum circuit, \( \langle Z(x, \theta) \rangle = \langle x | V(\theta) Z V(\theta) | x \rangle \) is the measured expectation. The circuit parameter \( \theta \) is optimized to minimize the mean squared error loss function on the given dataset

\[
C(\theta) = \sum_i (\langle Z(x_i, \theta) \rangle - y_i)^2.
\]  
(4)

Generally, the optimization method is gradient descent, in which the cost function gradient

\[
\frac{\partial C(\theta)}{\partial \theta} = \sum_i (\langle Z(x_i, \theta) \rangle - y_i) \frac{\partial \langle Z(x_i, \theta) \rangle}{\partial \theta},
\]  
(5)

is evaluated by parameter shift rule [45]

\[
\frac{\partial \langle Z(x_i, \theta) \rangle}{\partial \theta} = \frac{1}{2} \left( \langle Z(x_i, \theta + \frac{\pi}{4}) \rangle - \langle Z(x_i, \theta - \frac{\pi}{4}) \rangle \right).
\]  
(6)

3.2.2. Experimental implementation
Our implementation of the circuit is shown on the Board II in figure 2(c), where the photon-a is used to perform the data encoding and variational quantum classifier. The measurement of the data photon is achieved by a PBS and two detectors on the transmission and reflection direction. The expectation is then calculated as

\[
\langle Z \rangle = \frac{N_{13} - N_{23}}{N_{13} + N_{23}},
\]  
(7)

in which \( N_{13}, N_{23} \) are the number of coincidence events among detector 1–3 in 0.1 s. In equation (7), the detection signal of ancilla photon (photon-b) is used as the trigger for the detection of the data photon. Partial higher-order events (e.g. double-pair emission) during SPDC, such as the noise events \( N_{123} \), are automatically subtracted to improve the accuracy of the experiment. [46]. For each \( x_i, \theta \), the overall number of measurement shots is around 2000.

We connect all the detectors to a classical computer that automatically evaluates equations (4), (5) and (7) with the detection raw data. Since the HWPs are installed on electric rotation stage, we let the classical computer directs the iteration of parameter \( \theta \) with the evaluated gradient and Adam optimizer [47], which forms a closed control loop.

In each epoch, according to equations (5) and (6), expectations \( \langle Z(x_i, \theta + \frac{\pi}{4}) \rangle \) and \( \langle Z(x_i, \theta) \rangle \) need to be evaluated, which means the HWPs need to rotate to the angles \((x_i, \theta)\) and \((x_i, \theta + \frac{\pi}{4})\). Since the HWPs can only be continuously rotating, the total rotation distance (also the time consumption, given a constant rotating speed) during the training crucially depends on the rotation sequence of HWP(\(\theta\) and HWP(x), presenting as the route passing through these angle positions. Though finding a shortest route passing the discrete points in the space (the travelling salesman problem), is generally hard to solve [48], we develop following two techniques to speedup the training process and subsequently improve the computing accuracy.

Firstly, from the design of the single-qubit parameterized quantum circuit, it is easy to find

\[
\langle Z(x_i, \theta + \frac{\pi}{4}) \rangle + \langle Z(x_i, \theta - \frac{\pi}{4}) \rangle = 1.
\]  
(8)

Then equation (5) can be simplified as

\[
\frac{\partial C(\theta)}{\partial \theta} = \sum_i (\langle Z(x_i, \theta) \rangle - y_i) \left( 2 \langle Z(x_i, \theta + \frac{\pi}{4}) \rangle - 1 \right),
\]  
(9)
which reduces the required visiting positions to only \((x_i, \theta)\) and \((x_i, \theta + \pi/4)\), saving both the rotation steps and the rotation distance, which further reduces the cumulative error during the whole epoch.

Secondly, we note the positions we need to pass through are distributed on two parallel lines, where the number of \(x_i\) (also the size of training set) is generally large. To avoid the frequent round trip of HWP(\(\theta\)), we let the program travels along the \(x_i\) direction with priority, which forms a U-shaped route demonstrated in figure 4(a).

![Figure 4](image)

**Figure 4.** The optimized rotation sequences for the two variational quantum classifiers. (a) The rotation sequence of VQC. The points show the required visiting positions \((x_i, \theta), (x_i, \theta + \pi/4)\) in one iteration. The arrows show the rotation route. We first rotate HWP(\(\theta\)) to \(\theta + \pi/4\) and let HWP(\(x\)) rotates from \(x_i\) to \(x_m\). Then we rotate HWP(\(\theta\)) to \(\theta\) and let HWP(\(x\)) rotates back from \(x_m\) to \(x_i\). (b) The rotation sequence of NEVQC. The positions \((x_i, \theta_1, \theta_2), (x_i, \theta_1 \pm \pi/4, \theta_2), (x_i, \theta_1, \theta_2 \pm \pi/4)\) are distributed in three-dimensional space. Since the number of data items \(x_i\) is generally large, we divided the positions into five groups. Each group contains a set of positions with same \(\theta_i\). We let the classifier preferentially travels in the \(x_i\) direction, which is illustrated as the color fading in each group.

### 3.2.3. Results

We first train the classifier without AL for 35 steps with 20 labeled data vectors of three dividing patterns (We explain 20 data items are enough to show the advantage of AL in appendix E). Then we adopt the USAMP and the QBC strategies to the classifier respectively to observe the performance improvement of AL. We let the classifier actively selects 10 samples to learn from an unlabeled data pool of 20 data items, starting from the given 2 (for USAMP) or 3 (for QBC) data items with different labels. For each sample, the classifier trains itself for 10 epochs with the data it has. The epoch numbers in the experiment are chosen to make sure the training reaches convergence points, defined as the point where the decrease of cost function is less than \(10^{-3}\). The members of the committee for QBC strategy are support vector classifier (SVC) \([49, 50]\) with the radial basis function (RBF) kernel, \(k\)-nearest neighbor classification \([51, 52]\) with \(k = 3\), linear discriminant analysis algorithm \([53, 54]\), and decision tree \([55, 56]\) with max depth 7. These models are detailedly introduced in appendix A. Their hyperparameters are pre-optimized to fit our classification tasks here. To monitor the classifiers’ performance during the training process, we timely benchmark the classifier’s performance. We do the test for 5 (for without AL) or 10 (for AL) epochs by a test set of the corresponding training set pattern with 500 vectors. A specific demonstration of the training process in the experiment is shown in appendix B. We also conduct corresponding numerical simulations (see results in appendix C) to validate the experiment results.

The classification rates during the whole training process are shown in figures 5(a)–(c). The performance of training without AL, and the training with two AL strategies are compared together with the same ‘number of evaluations’, which is calculated as the cumulative times of expectation evaluation, directly corresponding to the realistic time consumption. The classification rate curves of AL in figure 5 do not increase monotonically, mainly due to the additions of new data samples at each step. The results suggest the AL training strategies generally accelerate the training as they achieves same classification rate with smaller number of evaluations than training without AL.

We show the ratios of labeling and computational costs (AL/non-AL) in table 1. The labeling costs of the methods are the numbers of labeled data items for training, and the computational costs are calculated as their number of evaluations. From the table, we can see that the AL can achieve a maximum reduction of 91.6% computing resources, compared with the classifier without AL, in the case of using the USAMP strategy for Patten 3.

Figures 5(g)–(i) show the classification rates varying with training set size. We find the performance of both AL strategies becomes better as they expand the training set size. In most cases, compared to the classifier without AL, the classifier with AL achieves the same classification rate (near convergence point) in the training set of less than 10 data items. As can be seen from table 1, the labeling efforts can be reduced by up to 85%, in the case of using the USAMP strategy for Patten 3. The only exception is VQC/NEVQC with
Figure 5. The performance of VQC, NEVQC with and without AL strategies during the whole training process. (a)–(f) The classification rates of VQC/NEVQC on the test set with the varying number of expectation evaluations. Each line shows an average result over 4 identical independent experiments and the bands show the standard deviation. The blue dashed lines mark the classification rates of VQC (without AL) at convergence and the corresponding numbers of evaluations. The green and orange dashed lines show the required numbers of evaluations of VQC (with USAMP and QBC, respectively) that achieve the same classification rates as VQC without AL. (g)–(i) The classification rates of VQC/NEVQC (with AL training strategies) on the test set with varying training set size. The color of each horizontal bar shows the average classification rate with the corresponding training set size marked on the x-axis. The green triangles on the bars show the required training set size to achieve the same classification rates as VQC without AL. The blue triangles suggest cases that the final classification rates of VQC/NEVQC with QBC strategy and the 10-data-item training set cannot match the final classification rates of VQC/NEVQC without AL (but already very close). Among these subfigures, each column corresponds to each dataset dividing patterns, shown in figure 2(a).

Table 1 shows the corresponding labeling and computational cost ratios for classifiers with and without AL.

| Pattern | Ratios(%) | USAMP | QBC | USAMP | QBC | USAMP | QBC |
|---------|-----------|-------|-----|-------|-----|-------|-----|
| VQC     | Labeling  | 50    | ×   | 30    | 50  | 15    | 25  | 15   | 34  |
|         | Computation | 77.9  | ×   | 29.6  | 75  | 8.4   | 18.2 | 8.4  | 46.8 |
| NEVQC   | Labeling  | 35    | ×   | 25    | 25  | 30    | 25  | 29   | 29  |
|         | Computation | 38.9  | ×   | 20.4  | 17.6| 29    | 26.1| 17.6 | 26.4 |

QBC strategy on Pattern 1, the final classification rates using QBC (reaching 0.955 for VQC, 0.9175 for NEVQC) are slightly worse than the ones without using AL strategy (0.9605 for VQC, 0.9365 for NEVQC). It may suggest the training set in Pattern 1 selected by QBC is slightly less representative. But QBC apparently helped accelerating the training and saving labeling costs, as figures 5(a) and (d) show.

The single-qubit classifier performs best on Pattern 1, while worst on Pattern 3, showing its endogenous classification ability, as its dividing boundaries are always vertical (see proof in appendix D). To further boost the performance, we strengthen the classifier by adding a nonlinear operation to the model.
3.3. NEVQC

We introduce nonlinearity to the ansatz to enhance the power of our variational quantum classifier [57, 58], by introducing an ancilla qubit and some additional quantum operations. As shown on the bottom panel in figure 2(b) and the Board III in figure 2(c). We first apply a Hadamard gate on the ancilla photon to prepare \((|H⟩ + |V⟩)/\sqrt{2}\). Then, we place a PBS on the crossing routes of the data photon and ancilla photon and balance the optical distance difference between the two photons such that they are temporally and spatially superposed on the PBS. Two parameterized single-qubit gates are inserted into the circuit for training the model. Finally, we postselect the events where there is exactly one photon exiting each output of the PBS and the ancilla is in state \(0\). We measure the input qubit in the Z basis. The ancilla photon, an additional Hadamard gate and PBS operation, and postselection of measurements, constitute the introduced nonlinearity.

The measured Z expectation for photon-a is calculated as

\[
⟨Z⟩ = \frac{N_{45} - N_{46}}{N_{45} + N_{46}},
\]

in which \(N_{45}, N_{46}\) are the number of coincidence events among detector 4–6 in one second. The number of measurement shots is 5500.

The same experiment settings, including the cost function, the AL strategies, the dataset configuration, are applied for NEVQC as for VQC. However, the automatic rotation scheme of the electrically driven HWP (trainable single-qubit gate) slightly different, since we now have three electrically driven HWPs (HWP(x) for encoding the data and HWP(\(θ_1\)), HWP(\(θ_2\)) for the two trainable single-qubit gates), thus the corresponding route is then in 3-dimensional space. Moreover, equation (8) no longer holds. The program needs to do the measurements with the angles as \((x, θ_2, θ_1), (x, θ_2 ± π/4, θ_1), (x, θ_2, θ_1 ± π/4)\) for the evaluation of derivatives \(\partial C/\partial θ_2\) and \(\partial C/\partial θ_1\). Our strategy is yet similar. As shown in figure 4(b), same as in the training of VQC, we let the program preferentially travel in the \(x_1\) direction, and then seek the shortest path among the five parameter configurations \(\{(θ_1, θ_2), (θ_1 ± π/4, θ_2 ± π/4)\}\) in the left two dimension.

The classification rates during the whole training process in figures 5(d)–(f) show the advantage of NEVQC compared to VQC, as it generally achieves higher classification rates on the three patterns, which fully demonstrates the performance improvement after we introduce nonlinear operation. The efficacy of AL is also validated on NEVQC, as USAMP and QBC achieves same classification rate in less time than the training without AL in most cases. As shown in table 1, the training with AL strategies require only 17.6% number of evaluations to match the performance of training without AL in the best case. Meanwhile, the AL methods also enhance the upper bound of the model’s performance, which may originate from their representative choice of data. Figures 5(g)–(i) shows the classification rates varying with training set size. The phenomena is similar with the ones of VQC. In the best case, training with AL strategies only require 25% labeling efforts to match the performance of training without AL.

4. Conclusion

We build a fully-programmable photonic quantum processor with high gate fidelities, which allows flexible execution of hybrid quantum-classical computing schemes. The AL strategies are coded as classical programs and deployed to our designed variational quantum classifiers on the quantum processor. Comparative experiments demonstrate a very positive effect of AL in the practical application of quantum machine learning, as it sharply cuts 66% labeling requirements and 53% computations on average, and 85% labeling efforts and 91.6% computations at most.

Besides, the implementation of high-fidelity operations, programmability, and speedup techniques during the training in the experiments make possible the implementations of various quantum machine learning algorithms in the free-space optical quantum system. And the implementation of nonlinearity in our experiment also demonstrates the path towards designing powerful and practical quantum machine learning models. By introducing the large-scale interferometer similar to the state-of-the-art photonic quantum processor, JiuZhang [59, 60], it is possible to immediately form large-scale quantum machine learning applications.

Data availability statement

The data generated and/or analyzed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.
Acknowledgment

H-L H acknowledges support from the Youth Talent Lifting Project (Grant No. 2020-JCJQ-QT-030), National Natural Science Foundation of China (Grants Nos. 11905294, 12274464), China Postdoctoral Science Foundation, and the Open Research Fund from State Key Laboratory of High Performance Computing of China (Grant No. 201901-01).

Appendix A. Members of committee in QBC selection strategy

The QBC is one of the selection strategies to evaluate the importance of data in our AL experiments. The QBC committee consists of four members. They are SVC with the RBF kernel, \(k\)-nearest neighbor with \(k = 3\), linear discriminant analysis algorithm (LDA), and decision tree with max depth 7.

SVC [49, 50] is a supervised binary classification model. It learns a boundary that leads to the max margin from both classes of data. Rather than fitting all the training data, it focuses on the points close to the boundary, which named support vectors. Combined with the kernel function, SVC can express a very complex classification boundary to achieve better classification performance. The kernel function is a special nonlinear function which maps the low-dimensional feature space to a higher one. So the original linearly inseparable problem in the low-dimensional space transforms to that linearly separable in the high-dimensional space. The RBF kernel is one of typical kernel functions which we choose to enhance our SVC model.

KNN (\(k\)-nearest neighbor) [51, 52] is one of the most classic and simplest supervised machine learning models, which contains no explicit learning or training procedures. KNN concentrates on the \(k\) nearest data points to the test data. If most neighbor data points belong to a certain class, the test data is also divided into this class. The hyperparameters in the KNN algorithm are the distance metric and the \(k\) value. We use the Euclidean distance metric and set the \(k\) value to 3.

LDA [53, 54], also called Fisher’s linear discriminant, is a fundamental supervised dimensionality reduction method. The key idea of LDA is to maximize the between-class variance and minimize the within-class variance. Take binary classification as an example, LDA first projects the data points in the feature space onto a line. Then it learns to optimize the line to make the projections of data points in the same class on the line as close as possible. Meanwhile, for projections of data points in different classes, the distances are maximized. To predict the test data, we calculate the projection of the test data point on the optimized line. Then the estimated label is judged according to the position of the projection.

Decision tree algorithm [55, 56] is a common supervised learning algorithm. It depends on a directed decision tree to make decisions on multi-classification task. During the training process, the decision tree is built according to the principle of minimizing the loss function. A decision tree consists of nodes and directed edges. There are two types of nodes: internal node (records a feature of dataset) and leaf node (denotes a class). Each directed edge corresponds to the certain value of a feature. When predicting a test data, which branch to enter is determined according to the judgment result of the internal node. Until the test data reaches the leaf node, we obtain the final classification result.

Appendix B. Demonstration of the training process with AL strategy

The AL training process of VQC and NEVQC (both using USAMP strategy) on Patterns 1 and 2 is given in figure B1. We choose one set of experimental results to demonstrate the actual training process. The classifier iteratively adds data item to the labeled data pool and trains itself. We record the model parameters and draw the training data set and classification lines in figure B1. We set the initial training data set by randomly choose two data items with different labels. The blue and red dots respectively represent the data points in class 1 and class 0, and the orange triangle denotes the newly chosen data item in the next epoch.

We can see that as the training data size increases, the model classification lines are trained to get closer to the actual dividing margins, providing a more intuitive understanding of the model ability mentioned in previous section. Figures B1(a) and (b) show that the classification lines of VQC are always maintaining a vertical relationship while optimization. But for the NEVQC given in figures B1(c) and (d), the included angle of two classification lines is continuously optimized as training data size increases. The classification lines are optimized to cross through data points in the same class, which supports the derivation of equation (D.1).

Moreover, during the AL training process, the newly chosen data item is always close to the model classification line at current model parameters, which is on great agreement with the least confidence principle in USAMP.
Figure B1. The active learning training process of (a), (b) VQC and (c), (d) NEVQC using USAMP strategy. We choose one set of experimental results to demonstrate the training process. The classifier iteratively adds data item to the labeled data pool and trains itself. We record the model parameters and draw the current training data set and classification lines in the pictures. The red and blue circles respectively refer to the data items of class 1 and class 0 in the labeled data pool, and the newly chosen data item in the next round is represented by an orange triangle.

**Appendix C. Classical simulation**

**C.1. Numerical simulation**
We perform abundant numerical simulations to support our experimental findings. In our AL experiments, the classifier iteratively selects 10 data items from an unlabeled data pool with size 20. After a newly chosen data item is added, the classifier trains itself for 10 epochs with its labeled training dataset. As for training process without AL, we train the classifier for 35 training epochs with 20 labeled training dataset. The reasonability and efficacy of these settings is tested in numerical simulations.

For each classifier and data pattern, 100 independent numerical experiments are implemented to calculate the mean and standard deviation of the results. Figure C1 shows the comparison among different training epoch after a newly chosen data item added. During the training process, we iteratively increase the labeled training set size up to 20. We can see that for training epoch 10, 20 and 30, the classification rate curve reaches the final convergence with little difference. The green dashed line indicates the position where 10 data items have been labeled to train the classifier. It is obvious that 10 labeled data items are sufficient for training the classifier close to its maximum performance. Therefore, considering the balance between classification performance and training time consumption, we finally set the number of selected data items to 10. And for each newly added item, we train the classifier for 10 training epoch with its current labeled dataset.

**C.2. Experimental performance compared to numerical simulation**
We also compare our experimental outcomes to numerical simulations, to show the high-precision manipulation of our experiments. For experiments and simulations, we take the same initial data set and
Figure C1. Comparison among different training epochs after a newly chosen data item added. The classification rate plots show the performance of classifier with training epochs 10, 20 and 30. We iteratively increase the labeled training set size up to 20. The vertical green dashed line refers to the position where 10 data items have been added to the labeled training data pool.

Figure C2. Comparison between experiment and simulation results in training of NEVQC on Pattern 3. (a)–(i) Show the loss function values of USAMP training with growing number of training data samples. (j) Shows the classification rate growth with increasing size of training set. (k) and (l) shows the loss function and classification rate values of non-AL training with growing epochs.
Figure C3. The average absolute classification rate error between experiments and their simulations. For active learning process, the absolute classification rate error is calculated every time a newly chosen data item is added to the labeled data pool. As for training without active learning, we test the model performance on test dataset every five training epoch.

Figure C4. The average absolute loss error between four sets of experiments and their simulations. The way to collect and display the comparison results is similar to that of classification rate error.

Furthermore, in order to quantify the average performance of the whole experiments, we present the average absolute loss and classification rate error of experiments by comparing to their theoretical simulations in figures C3 and C4. We compute the average absolute loss and classification rate error using the following equation

\[ \text{Error} = \frac{1}{L} \sum_{i=1}^{L} |v_i^{\text{expr}} - v_i^{\text{simu}}|, \]  

(C.1)

where \( v_i^{\text{expr}} \) and \( v_i^{\text{simu}} \) represent the value of loss or classification rate of experiments and simulations, and \( L \) refers to the training data size (or test data size) for loss error (or classification rate error). For AL process, the absolute classification rate error is calculated each time a newly chosen data item is added to the labeled data pool. As for training without AL, we test the model performance on test dataset every five training epoch. Four independent sets of experiments are implemented for the VQC and NEVQC classifier to report the average value and the standard deviation.
As can be seen, there is a good agreement between the experimental results and simulations, both for VQC and NEVQC. Furthermore, we notice that VQC has shown better experimental classification rate and stability, with maximum classification rate error 0.01 and maximum loss error 0.005. This phenomenon is understandable because VQC uses fewer qubits and quantum operations than NEVQC.

Appendix D. The classification power of VQC and NEVQC

We provide rigorous derivation of the maximum classification rates and generalization power for VQC and NEVQC.

D.1. Variational quantum classifier

The implementation of VQC contains only one parameterized single-qubit gate formed by the HWP. By evaluating the sign of the measured Z expectation, VQC actually determines the predicted label through a comparison between probability \( P_0 \) and \( P_1 \), where \( P_0 \) and \( P_1 \) represent the probability of \(|0\rangle\) and \(|1\rangle\) events in the detectors. To obtain the analytical form of the probability, we consider the transformation within the classifier.

\[
|\psi\rangle_{\text{VQC}}^{\text{en}} = x_i|0\rangle + y_i|1\rangle \\
\xrightarrow{\text{HWP}(\theta)} (x_i \cos 2\theta + y_i \sin 2\theta)|0\rangle + (x_i \sin 2\theta - y_i \cos 2\theta)|1\rangle,
\]

where \(|\psi\rangle_{\text{VQC}}^{\text{en}}\) refers to the input state of VQC and \((x_i, y_i)\) is the encoding data.

For the case of \((Z) > 0\), we have

\[
(x_i \cos 2\theta + y_i \sin 2\theta)^2 > (x_i \sin 2\theta - y_i \cos 2\theta)^2.
\]

With a simple deformation, the inequality is re-expressed as

\[
\left( x_i \sin \left( 2\theta + \frac{\pi}{4} \right) + y_i \sin \left( 2\theta - \frac{\pi}{4} \right) \right) \cdot \left( -x_i \sin \left( 2\theta - \frac{\pi}{4} \right) + y_i \sin \left( 2\theta + \frac{\pi}{4} \right) \right) > 0.
\]

Therefore, the classification function of VQC is realized through the geometric relationship of the two classification lines \( y + k_1x = 0 \) and \( y + k_2x = 0 \), where \( k_1 = \sin(\theta + \frac{\pi}{4}) \) and \( k_2 = -\frac{\sin(\theta - \frac{\pi}{4})}{\sin(\theta + \frac{\pi}{4})} \). When \((y + k_1x) \cdot (y + k_2x) \geq 0\) we assign the predicted label to 1, otherwise –1. It is obvious that \( k_1 \cdot k_2 = -1 \), indicating the two classification lines are mutual perpendicular. The expressibility of VQC is limited by this vertical characteristic due to the independence of the model parameter.

Now we will prove that, due to the property of Pattern 1, VQC can theoretically achieve a 100% maximum classification rate with sufficient training data set and training epoch. However, for Patterns 2 and 3 whose geometric property is not a type of vertical margin, VQC can only achieve a maximum classification rate much lower than 100%.

As shown in figure D1, the orange solid lines with slope angle \( \beta_1 \) and \( \beta_2 \) represent the actual dividing margin of a certain pattern. And the green dashed lines with slope angle \( \gamma_1 \) and \( \gamma_2 \) refer to the classification lines at current model parameters. The blue arrows indicate the correctly classified zone and the red arrows point to the error zone. Two situations may arise during the training process. We first consider a general situation given in figure D1(a), where two classification lines cross through data points in the same class. In this situation, data items of at least one class are exactly classified, and the maximum classification rate is

\[
\text{acc} = 1 - \min_{\gamma_1, \gamma_2} \left| \frac{|\beta_1 - \beta_2| - |\gamma_1 - \gamma_2|}{\pi} \right|,
\]

\[
= 1 - \min_{\theta} \left| \frac{|\beta_1 - \beta_2| - \arctan \left( \frac{k_1 - k_2}{1 + k_1k_2} \right)}{\pi} \right|.
\]

(D.1)

For the other situation shown in figure D1(b), where two classification lines cross through data points in different classes, the classifier can easily move the situation to the previous case through adjusting the value of parameter single-qubit gate (HWP(\(\theta\))), so equation (D.1) still holds.

As for VQC, there is a constrain between \( \gamma_1 \) and \( \gamma_2 \) due to the perpendicular relationship of two classification lines, which can be expressed as \( \gamma_1 = \gamma_2 + \frac{\pi}{4} \). Therefore, the classification rate

\[
\text{acc} = 1 - \frac{|\beta_1 - \beta_2|}{\pi},
\]

implying that the maximum classification rate totally depends on the proportion of each class for a certain pattern. As is shown in figure 2(a), the slope angle difference \( |\beta_1 - \beta_2| \) of the actual dividing margins for Patterns 1–3 are \( \pi/2, \pi/4 \) and \( \arctan(\frac{1}{2}) \) respectively. Then we can easily deduce a
maximum classification rate of 100% for Pattern 1, 75% for Pattern 2 and 57.79% for Pattern 3, which is validated in figure 5.

### D.2. NEVQC

NEVQC is realized by introducing an ancilla qubit (prepared as $|+\rangle$ state), a PBS operation, and postselection on the ancilla qubit after a parameterized single-qubit gate (HWP($\theta_2$)) applied on it. As shown in figure 2(c), we place a triangular prism on the path of the data qubit. The electric linear translation stage below it can be used to adjust the optical path difference between the two photons. We scan the position of the triangular prism to check the indistinguishability of the outcome photon. When the triangular prism reaches the interference point, by postselecting the events where there is exactly one photon exiting each output of the PBS, the quantum state is

$$ \frac{1}{\sqrt{2}} (\alpha |0\rangle + \beta |1\rangle) $$

where $\alpha |0\rangle$ is the data qubit after applying the first parameterized single-qubit gate (HWP($\theta_1$)).

For the maximum classification rate of NEVQC, similar to the analysis of VQC, assuming $\langle Z\rangle > 0$, we have

$$ (\cos 2\theta_2(x; \cos 2\theta_1 + y; \sin 2\theta_1))^2 > (\sin 2\theta_2(x; \sin 2\theta_1 - y; \cos 2\theta_1))^2. $$

Then

$$ (x; \cos(2\theta_1 - 2\theta_2) + y; \sin(2\theta_1 - 2\theta_2)) \cdot (x; \cos(2\theta_1 + 2\theta_2) + y; \sin(2\theta_1 + 2\theta_2)) > 0. $$

We notice that two classification lines of NEVQC have slopes $k_1 = \cot(2\theta_1 - 2\theta_2)$ and $k_2 = \cot(2\theta_1 + 2\theta_2)$, with an included angle

$$ \gamma_1 - \gamma_2 = \arctan \left( \frac{k_1 - k_2}{1 + k_1 k_2} \right) = \arctan(\tan(4\theta_2)). $$

Using equation (D.1), we get the maximum classification rate

$$ \text{acc} = 1 - \min_{\theta_1, \theta_2} \frac{||\beta_1 - \beta_2| - \arctan(\frac{k_1 + k_2}{1 + k_1 k_2})||}{\pi} $$

$$ = 1 - \min_{\theta_2} \frac{||\beta_1 - \beta_2| - \arctan(\tan(4\theta_2))||}{\pi}. $$

(D.3)

Although $\theta_1$ does not contribute to the maximum classification rate, it ensures that two classification lines can always transfer to the first classification situation shown in figure D1. Along with the training process, NEVQC iteratively adjusts the slope angles of classification lines while optimizing their included angle. Hence, the model expressibility of NEVQC is well enhanced. According to equation (D.3), there exists at least one solution $\theta_2 = \frac{\theta_1 - \theta_1}{4}$ to achieve a maximum classification rate 100% for any pattern. We find NEVQC will generally outperforms its single-qubit counterpart VQC.

---

**Figure D1.** The illustration for two classification situations on an arc. The green dashed line with slope angle $\gamma_1$ or $\gamma_2$ represents the classification line at current model parameters. The orange solid line with slope angle $\beta_1$ or $\beta_2$ refers to the actual dividing margin of a certain pattern. There are two general classification situations appearing in the training process. (a) Two classification lines cross through data points in the same class. In this situation, data items of at least one class are exactly classified. (b) Two classification lines cross through data points in different classes. Moreover, the blue arrows indicate the correctly classified zone and the red arrows point to the error zone.
Table D1. The expected decision boundaries $E[\hat{a}], E[\hat{b}]$ and generalization errors of a fully optimized NEVQC on the three data patterns.

| Pattern | Boundary $a$ | $E[\hat{a}]$ | Boundary $b$ | $E[\hat{b}]$ | Classification rate | Generalization error |
|---------|--------------|--------------|--------------|--------------|---------------------|---------------------|
| Pattern 1 | $\pi$ | $\frac{11}{12} \pi$ | $\frac{1}{2} \pi$ | $\frac{31}{42} \pi$ | 97.6% | 2.38% |
| Pattern 2 | $\frac{\pi}{4}$ | $\frac{1}{4} \pi$ | $\frac{1}{4} \pi$ | $\frac{1}{4} \pi$ | 97.6% | 2.38% |
| Pattern 3 | $\arctan \frac{1}{4}$ | 0.308 | $\pi$ | $\frac{41}{42} \pi$ | 95.6% | 4.39% |

D.3. The generalization power of NEVQC trained on 20 data samples

A model’s generalization power can be quantified as the difference of its classification rates over training set and test set. In our experiment, we generate the $N = 20$ data samples of training set $T = \{ (\cos x_i, \sin y_i) \}_{i=1,...,N}$ and $M = 500$ samples of test set by randomly sampling $x_i$ from the uniform distribution on $[0, \pi]$. The classification rates of VQC and NEVQC on the test sets, shown in figure 5, can generally achieve their theoretical maximum classification rate derived in the last two subsections. Therefore, an empirical strong generalization power is found for the two classifiers, even though they are only trained with 20 or less data samples.

Theoretically, to investigate NEVQC’s generalization power, we assume a NEVQC model is fully optimized over the given $N$ data samples, and compare the model’s expected classification results with our designed data patterns. The generalization over given patterns implies the generalization over all test sets that follow uniform distribution on $[0, \pi]$. We relabel the data samples in an increasing order of their angles. According to theory of order statistics, $\frac{x_i}{N}$ follows Beta distribution. Denote $x_0 = 0$ and $x_{N+1} = \pi$. For any pattern, suppose a boundary between the $\pm 1$ classes is $a$ (We directly use the corresponding angle to represent the boundary line.). Denote the neighboring data samples that sandwich $a$ as $x_{(a)} < a \leq x_{(a)+1}$, in which $(a)$ is the number of samples that is less than $a$ as $(a) := \# \{ i: x_i < a, i = 1, \ldots, N \}$. For arbitrary values of $(a)$, a fully optimized NEVQC will correctly classify both $x_{(a)}$ and $x_{(a)+1}$. It’s decision boundary, denoted at therefore lines between them. Assume it follows a uniform distribution on the interval $(x_{(a)}, x_{(a)+1}]$. Then the expected positions of the model’s decision boundary is

$$E[\hat{a}] = E[E[\hat{a}|x_1, \ldots, x_N]] = \frac{x_{(a)} + x_{(a)+1}}{2} = \frac{\sum_{l=0}^{N} \mathbb{P}(\{a\} = l)E \left[ \frac{x_l + x_{l+1}}{2} \right]}{\sum_{l=0}^{N} \binom{N}{l} \left( \frac{a}{\pi} \right)^l \left( 1 - \frac{a}{\pi} \right)^{N-l} \frac{2l+1}{2(N+1)}} = \frac{2an + \pi}{2(N+1)}.$$  

All three patterns have two $\pm 1$ boundaries. According to the results above, we yield the expected model’s classification rates on the patterns, as shown in table D1. The rates are all higher than 95%, which confirms the generalization of NEVQC trained by 20 data samples.

Appendix E. Discussion on the choice of training set size

Our choice of training set size seems small. But it is sufficient for showing the advantage of AL for two reasons:

1. Our designed data patterns have simple topologies that can be described with a relatively low number of data points. As shown in appendix D.3, the models trained with 20 data samples generalizes well on all uniform distributed random test sets over the patterns. Specifically, we prove a NEVQC trained by 20 data samples can generally achieve over 95% classification rates. Besides, figure 5 in the manuscript suggests classifiers with AL strategies can achieve their theoretical maximum classification rate on 500-size test set with only 12 or 13 training data. Therefore, increasing the size of training set cannot further improve the performance of VQC/NEVQC with AL. The training set of size 20 is enough to show the AL’s ability to reach the same classification rate with less data.

2. Figure B1 shows the classifiers with USAMP can quickly learn the correct decision boundaries with very few data items. Increasing the size of training set decelerates, instead of accelerating, such learning process,
because the processors will need to evaluate more data samples for selection. Since the models with AL can achieve high performance with few data items, the contrast of training set size will be greater by enlarging the original data pool. However, such phenomenon can be arbitrarily emphasized by adding more data samples, and is rather irrelevant to the power of AL. Therefore, more data cannot show the advantage of AL better.

Appendix F. Probability vanishment in postselection

When there are no \(|0\rangle\) events happened in the ancilla in NEVQC, we call this situation probability vanishment. Probability vanishment is an instinctive phenomenon rather than an erroneous occurrence. As for NEVQC, the probability of \(|0\rangle\) events in the ancilla is

\[
P_0^\ast = (\cos 2\theta_1 (x_i \cos 2\theta_1 + y_i \sin 2\theta_1))^2 + (\sin 2\theta_1 (x_i \sin 2\theta_1 - y_i \cos 2\theta_1))^2.
\]

We note the probability value \(P_0^\ast\) may approach zero when \(\theta_1 \approx 0\) or direction vectors \((x_i, y_i)\) and \((\cos \theta_1, \sin \theta_1)\) are approximately orthogonal. In such cases the probability \(P_0^\ast\) becomes so negligible that it is hard to observe any \(|0\rangle\) event in the experiment, and our calculation of equation (10) faces the problem of zero-denominator. This exception could indeed interrupt our experiments, because we cannot avoid the vanishment from happening, or extract information of the expectation through other approaches.

We hence take a break strategy. Once probability vanishment occurs, we skip the postselection and set \(\langle Z(x_i, \theta) \rangle\) to zero, the unbiased estimation of the expectation assuming its uniform distribution in the interval \([-1, 1]\). During our experiments, we observe that probability vanishment is a rather occasional phenomenon and the experimental results are in good agreement with our numerical simulations, as is shown in figures C3 and C4.

ORCID iDs

Chen Ding @ https://orcid.org/0000-0003-2227-2923
He-Liang Huang @ https://orcid.org/0000-0002-5121-3028

References

[1] Huang H-L, Xu X-Y, Guo C, Tian G, Wei S-J, Sun X, Bao W-S and Long G-L 2023 Near-term quantum computing techniques: variational quantum algorithms, error mitigation, circuit compilation, benchmarking and classical simulation Sci. China Phys. Mech. Astron. 66 250302
[2] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning Nature 549 195–202
[3] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets Nature 549 242–6
[4] Havlivček V, Córdoba A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces Nature 567 209–12
[5] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces Phys. Rev. Lett. 122 040504
[6] McArdle S, Endo S, Aspuru-Guzik A, Benjamin S C and Yuan X 2020 Quantum computational chemistry Rev. Mod. Phys. 92 015003
[7] Arute F et al (Google AI Quantum and Collaborators) 2020 Hartree-fock on a superconducting qubit quantum computer Science 369 1084–9
[8] Huang H-L et al 2021 Experimental quantum generative adversarial networks for image generation Phys. Rev. Appl. 16 024051
[9] Liu J, Hui Lim K H, Wood K L, Huang W, Guo C and Huang H-L 2021 Hybrid quantum-classical convolutional neural networks Sci. China Phys. Mech. Astron. 64 290311
[10] Gong M et al 2023 Quantum neural sensing of quantum many-body states on a 61-qubit programmable superconducting processor Sci. Bull. 68 906–12
[11] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism and implementation on near-term devices Phys. Rev. X 10 021067
[12] Abbas A, Sutter D, Zoufal C, Lucchi Aelien, Figalli A and Woerner S 2021 The power of quantum neural networks Nat. Comput. Sci. 1 403–9
[13] Gentili L, Cacciani A, Pirandola S, Verrucchi P and Banchi L 2020 Noise-resilient variational hybrid quantum-classical optimization Phys. Rev. A 102 052411
[14] Bittel L and Kilesch M 2021 Training variational quantum algorithms is NP-hard Phys. Rev. Lett. 127 120502
[15] Zheng H, Li Z, Liu J, Strelchuk S and Kendon V 2023 Speeding up learning quantum states through group equivariant convolutional quantum ansätze PRX Quantum 4 020327
[16] Golden J, Bärtisch A, Eidenbenz S and O’Malley D 2022 Evidence for super-polynomial advantage of qaoa over unstructured search (arXiv:2202.00648)
[17] Saggio V et al 2021 Experimental quantum speed-up in reinforcement learning agents Nature 591 229–33
[18] Huang H-Y et al 2022 Quantum advantage in learning from experiments Science 376 1182–6
[19] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning Nat. Phys. 17 1015–7
[20] Yang C, Garner A, Liu F, Tischler N, Thompson J, Yung M-H, Gu M and Dahlsten O 2021 Provable superior accuracy in machine learned quantum models (arXiv:2105.14434)
