Quantum Curriculum Learning

Quoc Hoan Tran, Yasuhiro Endo, and Hirotaka Oshima
Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Kawasaki, Kanagawa 211-8588, Japan

Quantum machine learning (QML) is considered to require significant quantum resources to achieve quantum advantage. Research should prioritize both the efficient design of quantum architectures and learning strategies to optimize resource usage. We propose a framework of quantum curriculum learning (Q-CurL) for quantum data, where the curriculum introduces auxiliary tasks or data to the learning model before progressing to more challenging ones. Theoretically, we define the curriculum criteria based on the data density ratio between tasks to determine the curriculum order and implement a data-based dynamic learning schedule in optimizing the loss function. Empirical evidence shows that Q-CurL enhances the training convergence and the generalization for unitary learning tasks and improves the robustness of quantum phase recognition tasks. Our framework provides a general learning strategy, bringing QML closer to realizing practical advantages.

Introduction.— There is a question as to whether speed is the only metric by which quantum machine learning (QML) algorithms should be judged. The hope is that QML can detect correlations in the data or generate new patterns that would be very difficult for classical algorithms to achieve, even though there is no clear evidence that classical data inherently requires quantum effects. This suggests a fundamental shift in the research community's perspective: it is preferable to use QML on data that is already quantum in nature.

Without being confined to proving a speed-up, how can we improve current QML algorithms? This question refocuses our attention on the concept of learning. In machine learning (ML), learning refers to the process through which a computer system enhances its performance on a specific task over time by acquiring and integrating knowledge or patterns from data. We can improve current QML algorithms by making this learning process more efficient.

Curriculum learning is inspired by the human learning process, based on the intuitive observation that we often begin with simpler concepts before progressing to more complex ones. In ML, this insight leads to the development of a strategy for sampling or task scheduling—a curriculum [1]. The investigation of curriculum learning in the QML field, especially regarding quantum data, is still in the early stages. The most relevant research has focused on investigating model transfer learning within hybrid classical-quantum neural networks [2]. Typically, this involves starting with a pre-trained classical network that is then modified and enhanced by adding a variational quantum circuit. Despite the potential benefits, there is still a lack of concrete evidence that effectively using a curriculum learning framework to schedule tasks and samples improves QML techniques.

We demonstrate the feasibility of implementing curriculum learning in a quantum data framework called quantum curriculum learning (Q-CurL). We present two principal approaches: task-based [Fig. 1(a)] and data-based [Fig. 1(b)] Q-CurL. In the task-based approach, we examine scenarios where a main task, potentially challenging or limited by data availability, can be facilitated through the preparatory parameter adjustment of an auxiliary task, which is comparatively easier or more data-rich. In the data-based approach, we introduce a dynamic learning schedule that adjusts data weights, thereby prioritizing the importance of data in minimizing the loss function.

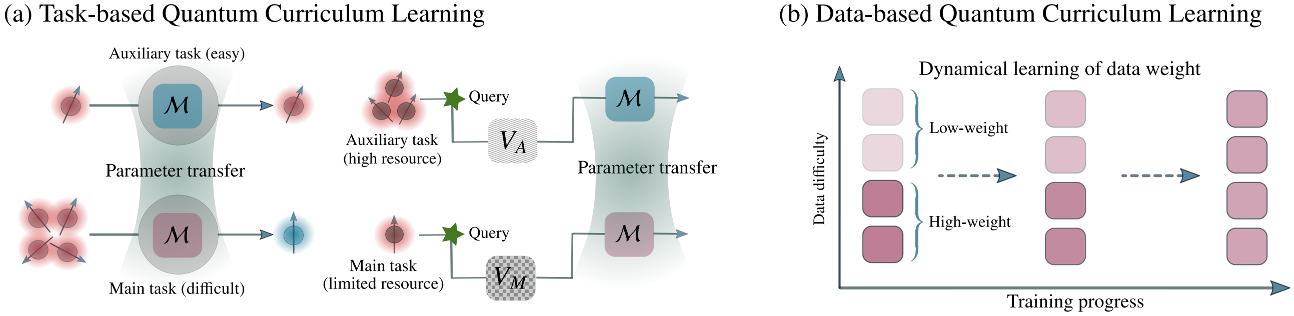


FIG. 1. Overview of two principal methodologies in quantum curriculum learning: (a) task-based and (b) data-based approaches. In the task-based approach, a model $\mathcal{M}$, designated for a main task that may be challenging or constrained by data accessibility, benefits from pre-training on an auxiliary task. This auxiliary task is either relatively simpler (left panel of (a)) or has a richer dataset (right panel of (a)). In the data-based approach, we implement a dynamic learning schedule to modulate data weights, thereby emphasizing the significance of quantum data in optimizing the loss function to reduce generalization error.

Task-based Q-CurL.— Unlike classical ML, which generally assumes a fixed amount of training data for all tasks, QML must navigate limited quantum resources. Therefore, the order of tasks and the allocation of training data to each task are crucial in QML. Efficient task scheduling can reduce the resources needed for training the main task, bringing QML closer to practical, real-world applications.

The goal of learning is to find a hypothesis model $h : \mathcal{X} \rightarrow \mathcal{Y}$ within a hypothesis set $\mathcal{H}$ that approximates the true function f such that $h(\boldsymbol{x}) \approx f(\boldsymbol{x})$. The loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is used to measure the approximation error $\ell(h(\boldsymbol{x}), \boldsymbol{y})$ between the prediction $h(\boldsymbol{x})$ and the target $\boldsymbol{y}$. We aim to find a hypothesis $h \in \mathcal{H}$ that minimizes the expected risk over the distribution $P(\mathcal{X}, \mathcal{Y})$. In practice, since the data generation distribution $P(\mathcal{X}, \mathcal{Y})$ is unknown, we use the observed dataset $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$ to minimize the empirical risk, defined as the average loss over the training data: $\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} \ell(h(\boldsymbol{x}_i), \boldsymbol{y}_i)$.

Given a main task $\mathcal{T}_M$, the goal of task-based Q-CurL is to design a curriculum for solving auxiliary tasks to enhance performance compared to solving the main task alone. We consider $\mathcal{T}_1, \ldots, \mathcal{T}_{M-1}$ as the set of auxiliary tasks. The training data $\left(\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)}\right)$ for task $\mathcal{T}_m$ are drawn from the probability distribution $P^{(m)}(\mathcal{X}^{(m)}, \mathcal{Y}^{(m)})$ with the density $p^{(m)}(\mathcal{X}^{(m)}, \mathcal{Y}^{(m)})$. We assume that all tasks share the same data spaces $\mathcal{X}^{(m)} \equiv \mathcal{X}$ and $\mathcal{Y}^{(m)} \equiv \mathcal{Y}$, as well as the same hypothesis h and loss function ℓ for all m.

We propose the curriculum weight $c_{M,m}$, where a larger $c_{M,m}$ indicates a greater benefit of solving $\mathcal{T}_m$ for improving the performance on $\mathcal{T}_M$. We evaluate the contribution of solving task $\mathcal{T}_i$ to the main task $\mathcal{T}_M$ by transforming the expected risk of training $\mathcal{T}_M$ as $R_{T_M}(h) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P^{(M)}}\left[\ell(h(\boldsymbol{x}), \boldsymbol{y})\right] = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P^{(m)}}\left[\dfrac{p^{(M)}(\boldsymbol{x}, \boldsymbol{y})}{p^{(m)}(\boldsymbol{x}, \boldsymbol{y})}\ell(h(\boldsymbol{x}), \boldsymbol{y})\right]$.

The curriculum weight $c_{M,m}$ can be determined using the density ratio $r(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{p^{(M)}(\boldsymbol{x}, \boldsymbol{y})}{p^{(m)}(\boldsymbol{x}, \boldsymbol{y})}$ without requiring the density estimation of $p^{(M)}(\boldsymbol{x}, \boldsymbol{y})$ and $p^{(m)}(\boldsymbol{x}, \boldsymbol{y})$. The key idea is to model the density ratio function $r(\boldsymbol{x}, \boldsymbol{y})$ using a linear model: $\hat{r}(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{\alpha}^{\top}\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{N_M} \alpha_i \phi_i(\boldsymbol{x}, \boldsymbol{y})$, where the vector of basis functions is $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}) = (\phi_1(\boldsymbol{x}, \boldsymbol{y}), \ldots, \phi_{N_M}(\boldsymbol{x}, \boldsymbol{y}))$, and the parameter vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{N_M})^{\top}$ is learned from data.

For quantum data, $\phi_l(\boldsymbol{x}, \boldsymbol{y})$ is naturally defined as the product of global fidelity used to compare two pairs of input and output states as $\phi_l(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{Tr}[\boldsymbol{x}\boldsymbol{x}_l^{(M)}]\,\mathrm{Tr}[\boldsymbol{y}\boldsymbol{y}_l^{(M)}]$. The parameter vector $\boldsymbol{\alpha}$ is estimated by minimizing

$$\frac{1}{2N_m}\sum_{i=1}^{N_m} \hat{r}^2_{\boldsymbol{\alpha}}(\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)}) - \frac{1}{N_M}\sum_{i=1}^{N_M} \hat{r}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i^{(M)}, \boldsymbol{y}_i^{(M)}) + \frac{\lambda}{2}\|\boldsymbol{\alpha}\|_2^2. \tag{1}$$

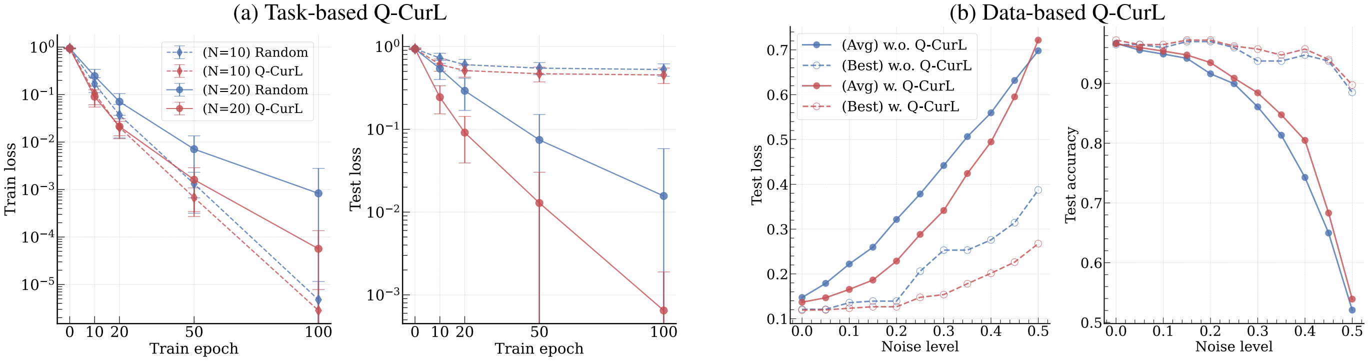
We can consider each $\hat{r}(\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)})$ as the contribution of the data $(\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)})$ from the auxiliary task $\mathcal{T}_m$ to the main task $\mathcal{T}_M$. We then define the curriculum weight $c_{M,m}$ as:

$$c_{M,m} = \frac{1}{N_m}\sum_{i=1}^{N_m} \hat{r}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)}). \tag{2}$$

We propose a Q-CurL game to examine the effect of Q-CurL. In this game, Alice has a machine learning model $\mathcal{M}(\boldsymbol{\theta})$ and wants to solve the main task $\mathcal{T}_M$, but she needs to solve all the auxiliary tasks $\mathcal{T}_1, \ldots, \mathcal{T}_{M-1}$ first. An assumption for data-access efficiency is data forgetting in task transfer, meaning that after solving task A, only the trained parameters $\boldsymbol{\theta}_A$ are transferred as the initial parameters for task B. We propose the following greedy algorithm to decide the curriculum order $\mathcal{T}_{i_1} \rightarrow \mathcal{T}_{i_2} \rightarrow \ldots \rightarrow \mathcal{T}_{i_M = M}$ before training. Starting from the main task $\mathcal{T}_{i_M}$, we find the auxiliary task $\mathcal{T}_{i_{M-1}}$ ($i_{M-1} \in \{1, 2, \ldots, M-1\}$) with the highest curriculum weights $c_{i_M, i_{M-1}}$. Similarity, to solve the task $\mathcal{T}_{i_{M-1}}$, we find the auxiliary task $\mathcal{T}_{i_{M-2}}$ in the remaining tasks with the highest $c_{i_{M-1}, i_{M-2}}$, and so on.

As a demonstration of the curriculum criteria based on $c_{M,m}$, we consider learning the unitary evolution of the XY-Hamiltonian $H_{XY} = \sum_{j=1}^{N}\left(\sigma_j^x\sigma_{j+1}^x + \sigma_j^y\sigma_{j+1}^y + h_j\sigma_j^z\right)$, where $h_j \in \mathbb{R}$ and $\sigma_j^x, \sigma_j^y, \sigma_j^z$ are the Pauli operators acting on qubit j. We represent the time evolution of H_{XY} via the ansatz V_{XY}, which is similar to the Trotterized version of $\exp(-i\tau H_{XY})$. The unitary for the main task, $V_{XY}^{(M)} = \prod_{l=1}^{L_M} V^{(l)}(\boldsymbol{\beta}_l)\prod_{l=1}^{L_F} V_{\text{fixed}}^{(l)}$, consists of $L_M = 20$ repeating layers, where each layer $V^{(l)}(\boldsymbol{\beta}_l)$ includes parameterized z-rotations RZ (with assigned parameter $\boldsymbol{\beta}_l$) and non-parameterized nearest-neighbor $\sqrt{i\mathrm{SWAP}}$ gates. Additionally, we include the fixed-depth unitary $\prod_{l=1}^{L_F} V_{\text{fixed}}^{(l)}$ with $L_F = 20$ layers at the end of the circuit $V^{(l)}$ to increase expressivity. Similarity, keeping the same parameters β_l, we create the target unitary for the auxiliary tasks $\mathcal{T}_m$ as $V_{XY}^{(m)} = \prod_{l=1}^{L_m} V^{(l)}(\boldsymbol{\beta}_l)\prod_{l=1}^{L_F} V_{\text{fixed}}^{(l)}$, with $L_m = 1, 2, \ldots, 19$.

Figure 2(a) depicts the average training loss and test loss for different training epochs and different numbers of training data ($N = 10, 20$). Introducing Q-CurL can significantly improve the trainability (lower training loss) and generalization (lower test loss) when compared with random order in Q-CurL game.


(a) Task-based Q-CurL

(b) Data-based Q-CurL

FIG. 2. (a) The average training loss and test loss for different training epochs and different numbers N of training data in the Q-CurL game, considering both random and Q-CurL orders. (b) The generalization of the trained QCNN on the ground state data in the quantum phase recognition task with $Q = 8$ qubits under varying noise levels in corrupted labels.

Data-based Q-CurL.— We present a form of data-based Q-CurL that dynamically predicts the easiness of each sample at each training epoch, such that easy samples are emphasized with large weights during the early stages of training and vice versa. Apart from improving generalization, this approach's benefit lies in its resistance to noise, which is especially needed in QML. For example, as observed in Ref. [3], existing QML models can accurately fit partially corrupted labels to quantum states in the training data but fail on the test data. We show that data-based Q-CurL can enhance the robustness based on the dynamic weighting of the difficulty fitting to corrupted labels.

Inspired by the confidence-aware techniques in classical ML, the key idea is to modify the empirical risk as $\hat{R}(h, \boldsymbol{w}) = \dfrac{1}{N}\sum_{i=1}^{N}\left((\ell_i - \eta)e^{w_i} + \gamma|w_i|^2\right)$. Here, $\boldsymbol{w} = (w_1, \dots, w_N)$, $\ell_i = \ell(h(\boldsymbol{x}_i), \boldsymbol{y}_i)$, and $|w_i|^2$ is the regularization term controlled by the hyper-parameter $\gamma > 0$. The threshold η distinguishes easy and hard samples with w_i emphasizing the loss $l_i \ll \beta$ (easy sample) and neglecting the loss $l_i \gg \beta$ (hard samples, such as training data with corrupted labels). The minimization problem is reduced to $\min_{\boldsymbol{\theta}}\min_{\boldsymbol{w}}\hat{R}(h, \boldsymbol{w})$, where $\min_{\boldsymbol{w}}\hat{R}(h, \boldsymbol{w})$ is decomposed at each loss ℓ_i and solved without quantum resources as $w_i = \operatorname{argmin}_w (l_i - \eta)e^{w} + \gamma|w|^2$. To control the difficulty of the samples, in each training epoch, we set η as the average value of all ℓ_i obtained from the previous epoch. Therefore, η changes dynamically during the early stages of training but remains constant during the convergence periods.

We apply the data-based Q-CurL to the quantum phase recognition task investigated in Ref. [4] to demonstrate that it can improve the generalization of the learning model. We employ the quantum convolutional neural network (QCNN) model with binary cross-entropy loss. Without Q-CurL, we use the conventional loss $\hat{R}(h) = (1/N)\sum_{i=1}^{N}\ell_i$ for the training and test phase. In data-based Q-CurL, we train the QCNN with the loss $\hat{R}(h, \boldsymbol{w})$, while using $\hat{R}(h)$ to evaluate the generalization on the test data set.

To evaluate the effectiveness of Q-CurL in emphasizing the loss function based on the difficulty of the training data, we consider the scenario of fitting corrupted labels. Given a probability p $(0 \leq p \leq 1)$ representing the noise level, the label y_i of quantum state $|\psi_i\rangle$ is transformed to the corrupted label $1 - y_i$ with probability p, while it remains the true label with probability $1 - p$. Figure 2(b) illustrates the performance of trained QCNN on test data across different noise levels in corrupted training labels. There is no significant difference at low noise levels, but as the noise level increases, the conventional training procedure fails to generalize effectively. In this case, introducing data-based Q-CurL in the training process (red lines) reduces the test loss and enhances testing accuracy compared to the conventional training method (blue lines).

Conclusion.— The proposed Q-CurL framework can enhance training convergence and generalization in QML with quantum data. Future investigations should examine whether Q-CurL can be designed to improve trainability in QML, such as by avoiding the barren plateau problem.

[1] A. Danyluk, L. Bottou, M. Littman, Y. Bengio, J. Louradour, R. Collobert, and J. Weston, Proceedings of the 26th Annual International Conference on Machine Learning , 41 (2009).
[2] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, Quantum **4**, 340 (2020).
[3] E. Gil-Fuster, J. Eisert, and C. Bravo-Prieto, Nat. Comm. **15**, 2277 (2024).
[4] I. Cong, S. Choi, and M. D. Lukin, Nat. Phys. **15**, 1273 (2019).