

# Quantum Adversarial Transfer Learning

Longhan Wang, Yifan Sun \*  and Xiangdong Zhang \*

Key Laboratory of Advanced Optoelectronic Quantum Architecture and Measurements of Ministry of Education, Beijing Key Laboratory of Nanophotonics & Ultrafine Optoelectronic Systems, School of Physics, Beijing Institute of Technology, Beijing 100081, China

\* Correspondence: yfsun@bit.edu.cn (Y.S.); zhangxd@bit.edu.cn (X.Z.)

**Abstract:** Adversarial transfer learning is a machine learning method that employs an adversarial training process to learn the datasets of different domains. Recently, this method has attracted attention because it can efficiently decouple the requirements of tasks from insufficient target data. In this study, we introduce the notion of quantum adversarial transfer learning, where data are completely encoded by quantum states. A measurement-based judgment of the data label and a quantum subroutine to compute the gradients are discussed in detail. We also prove that our proposal has an exponential advantage over its classical counterparts in terms of computing resources such as the gate number of the circuits and the size of the storage required for the generated data. Finally, numerical experiments demonstrate that our model can be successfully trained, achieving high accuracy on certain datasets.

**Keywords:** quantum transfer learning; quantum generative adversarial network; quantum machine learning; quantum computation

## 1. Introduction

Machine learning (ML) methods have successfully been applied to various fields such as speech recognition, visual object recognition, and object detection [1,2]. In recent years, ML research has been extended to applications in more complicated but ordinary scenarios, such as situations involving datasets belonging to different domains. The predicament in these scenarios is that the ML model trained on a dataset in one domain does not often work for tasks in a different domain of interest. One resource-consuming strategy used to solve the above issue is transfer learning (TL) [3–8], which estimates the usefulness of the knowledge learned in a source domain and transfers it to help the learning task in a target domain. However, conventional TL usually lacks efficiency when the distribution of the target domain data is completely different from the source domain. Recently, an adversarial transfer learning (ATL) method has been proposed to solve the issue. Such an ML scheme has the potential for a broad range of applications; it has been proven to be beneficial in areas such as natural language processing, autonomous cars, robotics, and image understanding [9–19].

The basic idea of the ATL method is to introduce generative models to bridge the gap between the datasets of different domains. This method is a new version of ML and cannot be provided by simply combining methods such as generative adversarial networks (GANs) [20–22] and TL. As schematically shown in Figure 1a, ATL controls the samples of a given source domain dataset and a target domain dataset, denoted as  $X^{R_s}$  and  $X^{R_t}$ , respectively. A generator  $G$  is employed to produce a fake sample  $X^G$  using  $X^{R_s}$  and a noise vector  $\vec{z}$ .  $X^G$  and  $X^{R_t}$  are then sent to a discriminator  $D$  to judge the probabilistic likelihood. The general purpose of training  $G$  and  $D$  is an adversarial game. Generator  $G$  is required to generate  $X^G$  as close as possible to the target data sample  $X^{R_t}$ , cheating the discriminator  $D$ . Discriminator  $D$  attempts to accurately distinguish  $X^G$  from  $X^{R_t}$  and avoid being cheated. The game finally reaches the Nash equilibrium [23] after the



**Citation:** Wang, L.; Sun, Y.; Zhang, X. Quantum Adversarial Transfer Learning. *Entropy* **2023**, *25*, 1090. <https://doi.org/10.3390/e25071090>

Academic Editors: Durdu Guney, Rosario Lo Franco and David Petrosyan

Received: 8 May 2023

Revised: 9 June 2023

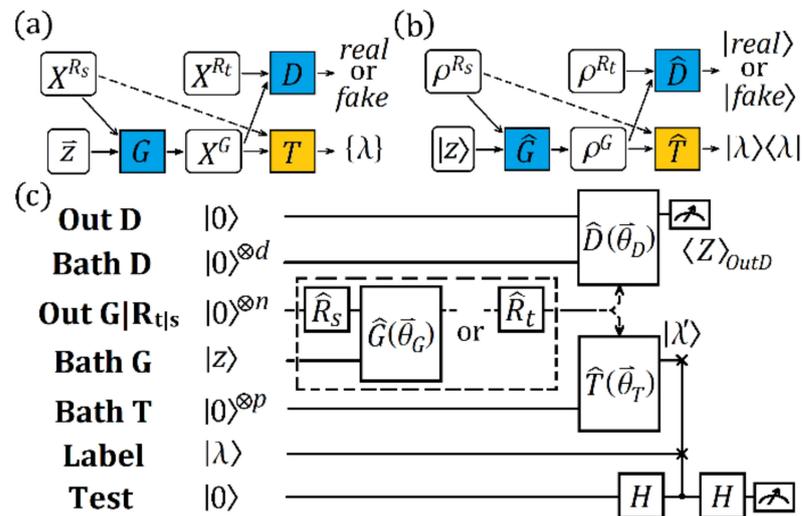
Accepted: 8 July 2023

Published: 20 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

parameters of the model are optimized. In cases of classification, a classifier  $T$  is correctly applied to label  $X^G$  according to the label of  $X^{R_s}$ . In certain cases, the source domain dataset is also employed as the input of  $T$ , which increases the efficiency of the training. The key ingredient in such a scheme is the cost function of the model, which finally converges to an equilibrium point.



**Figure 1.** (a) Model architecture of ATL. The generator  $G$  produces a data sample  $X^G$  on a source data sample  $X^{R_s}$  and a noise vector  $\vec{z}$ . The discriminator  $D$  distinguishes  $X^G$  from the target data sample  $X^{R_t}$ , assigning the judgement as “real” or “fake”. The classifier  $T$  assigns task-specific labels  $\{\lambda\}$  to fake data sample  $X^G$ . Note that source data sample  $X^{R_s}$  is only accepted into the next step when the efficiency of the training is increased, as marked by the dashed arrow. (b) The corresponding QATL scheme. The data samples ( $X^{R_s}$ ,  $X^{R_t}$ , and  $X^G$ ) are encoded by the quantum state  $X^{R_s}$  ( $\rho^{R_s}$ ,  $\rho^{R_t}$ , and  $\rho^G$ ), respectively. The functioning of  $G$ ,  $D$ , and  $T$  is implemented by the quantum operators  $\hat{G}$ ,  $\hat{D}$ , and  $\hat{T}$ , respectively. The judgement of  $\hat{D}$  is given by a quantum state ( $|real\rangle$  or  $|fake\rangle$ ) and the label is also encoded by  $|\lambda\rangle$ . (c) The quantum circuit of QATL. The qubit numbers of the registers **Bath D**, **Out G | R<sub>t</sub>s**, and **Bath T** are  $d$ ,  $n$ , and  $p$ , respectively. The registers **Out D** and **Test** both contain one qubit. Their qubits are initialized as  $|0\rangle$ . The register **Bath G** stores an  $m$ -qubit random state  $|z\rangle$  generated by the environmental coupling. The register **Label** has the initial state  $|\lambda\rangle$ .  $\hat{R}_t$ ,  $\hat{R}_s$ ,  $\hat{G}(\vec{\theta}_G)$ ,  $\hat{D}(\vec{\theta}_D)$ , and  $\hat{T}(\vec{\theta}_T)$  represent the unitary operators. The dashed box marks an option of applying two types of setups.  $H$  is the Hadamard gate.  $\langle Z_{outD} \rangle$  is the expectation value of operator  $Z$ , as defined in the main text.

Despite the success of the above-mentioned method, the increasing requirements of data processing present a significant challenge to all computing strategies, including ML. As quantum computing provides an opportunity to overcome this challenge, researchers have considered addressing ML tasks using quantum machines [24–57]. Proposed quantum machine learning algorithms include quantum support vector machines [29,30], quantum deep learning [32–38], quantum Boltzmann machines [39,40], quantum generative adversarial learning [41–45], and quantum transfer learning (QTL) [47–51]. A quantum counterpart of ATL has not yet been provided; this could exhibit an advantageous performance in cross-domain learning tasks. To propose such a counterpart is complex because a suitable quantum cost function must be established so that the quantum adversarial game for knowledge transfer can finally converge to an equivalent point, as in classical cases. How to develop such a function in the quantum regime and how to obtain a quantum version of ATL remain unknown.

In this paper, we demonstrate how we solved the above problem, and we propose quantum adversarial transfer learning (QATL) schemes. The training process of our QATL was equivalent to an adversarial game of a quantum generator and a quantum discriminator,

and we demonstrate that an equilibrium point also existed in the model. Specifically, a quantum cost function for adversarial training is provided; a measurement-based judgment of the data label and a quantum subroutine to compute the gradients are also discussed in detail. We prove that our proposal has an exponential advantage over classical counterparts in terms of computing resources such as the gate number of the circuits and the size of the storage for the generated data. This is of benefit to the transfer of complicated knowledge, during which a module is extensively called upon and a large amount of data are generated. We applied this scheme to a classification task based on the Iris dataset and the states on a Bloch sphere to prove that an extremely high classification accuracy could be achieved using this method.

## 2. Materials and Methods

Our QATL scheme is shown in Figure 1b. Here, the density matrix of the state that encodes a source (target) domain data sample is denoted by  $\rho^{R_s}$  ( $\rho^{R_t}$ ). A unitary operator is employed as the quantum generator  $\hat{G}$ , whose parameters are denoted by a vector  $\vec{\theta}_G$ . It operates on the state  $\rho^{R_s}$  and a quantum noise state  $|z\rangle\langle z|$ , outputting a *fake* data sample  $\rho^G$ . Another unitary operator, parameterized by vector  $\vec{\theta}_D$ , is used as the quantum discriminator  $\hat{D}$ . It operates on the state  $\rho^G$  generated by  $\hat{G}$  and the state  $\rho^{R_t}$ ; it then outputs the state  $|real\rangle\langle real|$ . If it operates on  $\rho^G$ , the state is  $|fake\rangle\langle fake|$ . As with ATL, the general training purpose of QATL is to maximize the probability of  $\rho^G$  passing the test of discriminator  $\hat{D}$ , simultaneously minimizing the probability of  $\hat{D}$  being cheated if  $\rho^G$  is fake. The optimization of the parameters  $\vec{\theta}_G$  and  $\vec{\theta}_D$  are addressed when the above quantum adversarial game reaches the Nash equilibrium. We considered the classification task and applied a unitary operator parameterized by  $\vec{\theta}_T$  as a quantum classifier  $\hat{T}$  [58–61]. This provided a mapping of the data sample ( $\rho^G$  or  $\rho^{R_s}$ ) to a label state  $L = |\lambda'\rangle\langle\lambda'|$ ;  $\vec{\theta}_T$  was updated during the training.

Next, we discuss the state evolution of the above scheme on a circuit and demonstrate how an equilibrium point of the game was reached. The quantum circuit of Figure 1b is displayed in Figure 1c. The whole circuit ran on seven quantum registers. The register **Out G** |  $R_{t|s}$  containing  $n$  qubits encoded the target data, source data, or the generated data. The encoding of the target (source) data was described by the operator  $\hat{R}_t$  ( $\hat{R}_s$ ), which acted on the register **Out G** |  $R_{t|s}$ . The register **Bath G**, containing  $m$  qubits, encoded the quantum noise state  $|z\rangle\langle z|$ . The generator  $\hat{G}(\vec{\theta}_G)$  operated on **Out G** |  $R_{t|s}$  and **Bath G**, producing the generated data. The register **Bath D**, containing  $d$  qubits, was employed as the internal workspace of the discriminator  $\hat{D}$ , whose outputs ( $|real\rangle\langle real|$  or  $|fake\rangle\langle fake|$ ) were stored by the register **Out D**. The register **Bath T** contained  $p$  qubits and was employed as the workspace of the classifier  $\hat{T}$ . The label state  $|\lambda'\rangle$  outputted by  $\hat{T}$  was compared with the single qubit label state  $|\lambda\rangle$  stored by the register **Label**. The register **Test** was used to perform the estimation of the likelihood of  $|\lambda'\rangle$  and  $|\lambda\rangle$ , so that  $\rho^G$  could be properly labeled. All qubits were initialized to be  $|0\rangle$ , except for those in **Bath G** and **Label**. The initial state of the quantum circuit shown in Figure 1c could then be denoted by

$$\rho^{in}(z) = |0\rangle\langle 0|^{\otimes(1+d+n)} \otimes |z\rangle\langle z| \otimes |0\rangle\langle 0|^{\otimes p} \otimes |\lambda\rangle\langle\lambda| \otimes |0\rangle\langle 0|, \tag{1}$$

where  $\otimes$  is the tensor product and  $|0\rangle\langle 0|^{\otimes p}$  is the tensor product of  $p$   $|0\rangle\langle 0|$ s; this was also applicable for other similar terms. The unitary operators of such a state encoding the source domain data, target domain data, and generated data were denoted by  $U_{R_s}$ ,  $U_{R_t}$ , and  $U_G(\vec{\theta}_G)$ , respectively. They were given by

$$U_{R_s} = I^{\otimes(1+d)} \otimes \hat{R}_s \otimes I^{\otimes(m+p+s+1)}, \tag{2}$$

$$U_{R_t} = I^{\otimes(1+d)} \otimes \hat{R}_t \otimes I^{\otimes(m+p+s+1)}, \tag{3}$$

$$U_G(\vec{\theta}_G) = I^{\otimes(1+d)} \otimes \hat{G}(\vec{\theta}_G) \otimes I^{\otimes(p+s+1)} \tag{4}$$

where  $I$  represents the  $2 \times 2$  identity matrix. The states  $\rho^{R_s}$  and  $\rho^{R_t}$  were then given by

$$\rho^{R_s} = U_{R_s} \rho^{in}(0) U_{R_s}^\dagger, \tag{5}$$

$$\rho^{R_t} = U_{R_t} \rho^{in}(0) U_{R_t}^\dagger. \tag{6}$$

We set **Bath G** to be  $|0\rangle^{\otimes m}$  instead of rotating it to a random quantum state with extra control. A random quantum state can naturally be generated by the entanglement of the register state and environmental degrees [44]. After  $|z\rangle\langle z|$  was generated according to the requirements, the state  $\rho^G$  could be expressed by

$$\rho^G = U_G(\vec{\theta}_G) \rho^{R_s} U_G^\dagger(\vec{\theta}_G). \tag{7}$$

The discriminator  $\hat{D}(\vec{\theta}_D)$  was applied to estimate the likelihood of  $\rho^G$  and  $\rho^{R_t}$ . The unitary operator of  $\hat{D}(\vec{\theta}_D)$  was given by

$$U_D(\vec{\theta}_D) = \hat{D}(\vec{\theta}_D) \otimes I^{\otimes(m+p+s+1)}. \tag{8}$$

Therefore, the resultant states of  $\rho^{R_t}$  and  $\rho^G$  after being operated by  $U_D(\vec{\theta}_D)$  were given by

$$\rho^{DR_t}(\vec{\theta}_D) = U_D(\vec{\theta}_D) \cdot \rho^{R_t} \cdot U_D^\dagger(\vec{\theta}_D), \tag{9}$$

$$\rho^{DG}(\vec{\theta}_D, \vec{\theta}_G, z) = U_D(\vec{\theta}_D) \cdot \rho^G(\vec{\theta}_G, z) \cdot U_D^\dagger(\vec{\theta}_D). \tag{10}$$

The expectation value of operator  $Z \equiv |real\rangle\langle real| - |fake\rangle\langle fake|$  could be measured on the register **Out D**. Such a value was close to 1 when  $\hat{D}(\vec{\theta}_D)$  generated the state  $|real\rangle\langle real|$ ; it was close to  $-1$  when  $\hat{D}(\vec{\theta}_D)$  outputted the state  $|fake\rangle\langle fake|$ .

The quantum classifier  $\hat{T}$  we considered was forced to output the label of  $\rho^G$  according to its closeness to  $\rho^{R_s}$ . We also introduced this method to judge whether the label was correct. The unitary operator of  $\hat{T}$  was defined by

$$U_T(\vec{\theta}_T) = I^{\otimes(1+d)} \otimes \hat{T}_1(\vec{\theta}_{T_1}) \otimes I^{\otimes m} \otimes \hat{T}_2(\vec{\theta}_{T_2}) \otimes I^{\otimes(1+s)} \tag{11}$$

where  $\hat{T}_1$  and  $\hat{T}_2$  are two unitary operators on **Out G | R<sub>t</sub>s** and **Bath T**, respectively. Their parameters were denoted by  $\vec{\theta}_{T_1}$  and  $\vec{\theta}_{T_2}$  correspondingly. Thus,  $\vec{\theta}_T = (\vec{\theta}_{T_1}, \vec{\theta}_{T_2})$ . The

resultant states of  $\rho^{R_s}$  and  $\rho^G$  after being operated by  $U_T(\vec{\theta}_T)$  were given by

$$\rho^{TR_s}(\vec{\theta}_T, \vec{\theta}_D, z) = U_T(\vec{\theta}_T) \cdot \rho^{R_s}(\vec{\theta}_G, z) \cdot U_T^\dagger(\vec{\theta}_T), \tag{12}$$

$$\rho^{TG}(\vec{\theta}_T, \vec{\theta}_D, z) = U_T(\vec{\theta}_T) \cdot \rho^G(\vec{\theta}_G, z) \cdot U_T^\dagger(\vec{\theta}_T). \tag{13}$$

One qubit of the above output state was the predicted label  $|\lambda'\rangle$  by  $\hat{T}$ . When comparing it with the label  $|\lambda\rangle$  of the input  $\rho^{R_s}$  by the swap test circuit [62] (which was composed of two Hadamard gates and one Fredkin gate, as shown in Figure 1c), the label of  $\rho^G$  was set to be  $|\lambda\rangle$  if  $|\lambda'\rangle$  and  $|\lambda\rangle$  were close enough. The closeness of  $|\lambda'\rangle$  and  $|\lambda\rangle$  could be estimated by the measurement of the register **Test**. The probability of the register **Test** being in state  $|0\rangle$  was given by

$$P_0 = \frac{1}{2}(1 + P_c) \tag{14}$$

where  $P_c$  is the overlap of  $|\lambda'\rangle$  and  $|\lambda\rangle$ , defined by  $P_c = \text{Tr}\left\{|\lambda\rangle\langle\lambda|L(\vec{\theta}_T, \vec{\theta}_G, z)\right\}$ . Therefore,  $P_c$  could be estimated by  $P_c = 2P_0 - 1$ . In specific cases, we could set a suitable boundary for accepting  $|\lambda\rangle$  to be the label of  $\rho^G$  (see the example below). Finally, the training of the quantum model could be performed by finding the following minimax point of the quantum cost function  $V(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T)$ , given by

$$\begin{aligned} \tilde{V}(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T) = & \frac{1}{M} \sum_{k=1}^M \left[ \frac{1}{2} + \frac{1}{2} \cos^2 \phi \text{Tr}\{Z\rho_k^{DR_i}(\vec{\theta}_D)\} \right. \\ & - \frac{1}{2} \sin^2 \phi \text{Tr}\{Z\rho_k^{DG}(\vec{\theta}_D, \vec{\theta}_G, z)\} \\ & \left. - \text{Tr}\{|\lambda_k\rangle\langle\lambda_k|[L_k(\vec{\theta}_T, \vec{\theta}_G, z) + L_k^{R_s}(\vec{\theta}_T)]\} \right]. \end{aligned} \tag{15}$$

The minimax point of  $\tilde{V}(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T)$  was obtained by tuning  $\vec{\theta}_G$  and  $\vec{\theta}_T$  to the minimum and tuning  $\vec{\theta}_D$  to the maximum. The subscript  $k$  represents the different samples fed into the model. The number of the samples is denoted by  $M$ .  $L_k^{R_s}(\vec{\theta}_T)$  is the label state outputted by  $\hat{T}$  when the input was the source domain data sample  $\rho_k^{R_s}$ . Such a term provides a bias; this could increase the training efficiency. The angle  $\phi$  is the parameter used to adjust the weight of the target domain data and the generated data in the cost function, and it could be set according to the requirements of the specific tasks. A usual form was given by considering that they were equally weighted. By setting  $\phi = \pi/4$ , we obtained

$$\begin{aligned} V(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T) = & \frac{1}{M} \sum_{k=1}^M \left[ \frac{1}{2} + \frac{1}{4} \text{Tr}\{Z\rho_k^{DR_i}(\vec{\theta}_D)\} - \frac{1}{4} \text{Tr}\{Z\rho_k^{DG}(\vec{\theta}_D, \vec{\theta}_G, z)\} \right. \\ & \left. - \text{Tr}\{|\lambda_k\rangle\langle\lambda_k|[L_k(\vec{\theta}_T, \vec{\theta}_G, z) + L_k^{R_s}(\vec{\theta}_T)]\} \right]. \end{aligned} \tag{16}$$

The minimax optimization problem of Equation (16) was implemented by alternating between two steps. In the first step, we updated the discriminator and the classifier parameters  $\vec{\theta}_D$  and  $\vec{\theta}_T$  whilst keeping the generator parameter  $\vec{\theta}_G$  fixed. In the second step, we fixed  $\vec{\theta}_D$  and  $\vec{\theta}_T$  and updated  $\vec{\theta}_G$ . We used the gradient descent method to update parameters  $\vec{\theta}_G$ ,  $\vec{\theta}_D$ , and  $\vec{\theta}_T$ , respectively. The values of  $\vec{\theta}_G$ ,  $\vec{\theta}_D$ , and  $\vec{\theta}_T$  given by the  $l$ th update were denoted by  $\vec{\theta}_G^l$ ,  $\vec{\theta}_D^l$ , and  $\vec{\theta}_T^l$ , respectively, with integer  $l = 1, 2, \dots$ ;  $\vec{\theta}_G^0$ ,  $\vec{\theta}_D^0$ , and  $\vec{\theta}_T^0$  as the initial values. Hence, the update rule was expressed by

$$\begin{aligned}
\vec{\theta}_G^{l+1} &= \vec{\theta}_G^l - \chi_G^l \nabla_{\vec{\theta}_G} V(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T), \\
\vec{\theta}_D^{l+1} &= \vec{\theta}_D^l - \chi_D^l \nabla_{\vec{\theta}_D} V(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T), \\
\vec{\theta}_T^{l+1} &= \vec{\theta}_T^l - \chi_T^l \nabla_{\vec{\theta}_T} V(\vec{\theta}_G, \vec{\theta}_D, \vec{\theta}_T),
\end{aligned} \tag{17}$$

where  $\chi_G^l$ ,  $\chi_D^l$ , and  $\chi_T^l$  are the learning rates. In the above scheme, we used the variational quantum circuit in [44] and [58] to implement the generator  $\hat{G}$ , the discriminator  $\hat{D}$ , and the classifier  $\hat{T}$ , as shown in Figure 1c. In [44], each quantum gate of a quantum circuit corresponded with a parameter of the quantum circuit. The number of quantum gates was polynomial in the number of qubits; that is,  $3/2Lq$ .  $L$  was the number of layers of quantum circuits and  $q$  was the number of qubits. Therefore, the number of parameters was also  $3/2Lq$ . In [58], each quantum gate of the circuit had three parameters. The number of quantum gates was polynomial in the number of qubits, which was  $2Lq + 1$ . Therefore, the number of parameters was  $6Lq + 3$ . In our scheme, the total number of quantum gates was bound by those operating on the register **Out G** |  $\mathbf{R}_{\text{ts}}$ , which was the sum of the number of quantum gates employed by the generator–discriminator sequence or the classifier. Based on the gate number of the above two circuits for the unitary operations we considered, the sum of the number of quantum gates remained polynomial in the number of qubits. We added two registers, **Label** and **Test**. Their gate number was constant and their impact on the complexity was negligible. In the process of obtaining the quantum cost function, the number of gates and the number of parameters of our scheme were polynomial in their number of qubits, which required exponentially fewer resources than the classical counterparts [10]. This property potentially benefits future applications for tasks involving complicated data structures. In general, if the information in a target dataset is insufficient, a large amount of data must be generated to effectively transfer the knowledge required to improve the training of the target dataset. This indicates a large storage requirement for the data and extensive calls of the modules. Otherwise, the gap between the features of different datasets is not bridged and the helpful information required to improve the training process is not captured and transferred. The improvement in computing resources by our QATL could loosen the requirements on storage and boost the efficiency of running the subroutines of the module, facilitating the execution of tasks.

To provide a more direct connection between the computation of the cost function and the updates of the parameters, we also applied two quantum circuits to calculate the gradients. Their theoretical scheme and circuit design are shown in Section SI of the Supplementary Materials. This scheme could be implemented on the recent quantum experimental platform because it was mainly based on variational quantum algorithms (VQAs) [63–66].

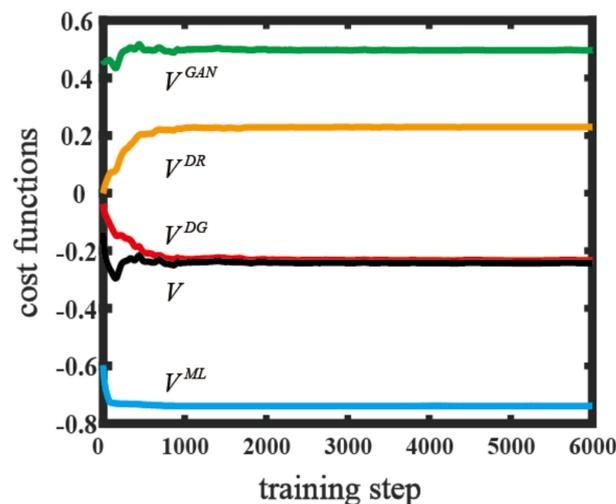
### 3. Results

To demonstrate the feasibility of QATLs, we considered their application in cases in which the probability distributions of the data in the two domains were different. The examples used the Iris dataset [67] and the states of the Bloch sphere.

The main task we considered was a classification of the Iris dataset in a numerical simulation. We chose two types of Iris flowers from the Iris dataset: *versicolor* and *virginica*. Each type contained 50 data samples. All data samples contained four attributes: sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW). We considered encoding the four attributes based on the amplitudes of a two-qubit state (amplitude-encoding strategy [47]). The source domain dataset was composed of 40 data samples; 20 were picked from the *Iris versicolor* data samples and labeled  $|0\rangle\langle 0|$ . The other 20 were picked from *Iris virginica* data and labeled  $|1\rangle\langle 1|$ . The target domain dataset was also a 20/20 set chosen from samples other than those applied to the source domain dataset and without labeling. The last 10 data samples of both types were used to obtain a test dataset to check the performance of the model. The classification task labeled the target domain

data samples using differently distributed source domain data. To enlarge the statistical difference of the source domain dataset and the target domain dataset and to improve the difficulty of classifying Iris flowers, we preprocessed the data samples by reducing the SL by 7 cm, the SW by 3 cm, the PL by 4 cm, and set the value of PW to 0. An illustration of the distribution of the data can be found in [67].

To show the convergence of terms of the cost function on the above datasets, we plotted the cost function  $V$  and its components as a function of the training step in Figure 2a. The cost function  $V$  converged to  $-0.217$ . The component  $V^{DG}$  was defined as  $V^{DG} = -1/4M \cdot \sum_{k=1}^M \text{Tr} \left\{ Z \rho_k^{DG} \left( \vec{\theta}_D, \vec{\theta}_G, z \right) \right\}$ , which represented the average probability of  $\rho_k^{DG} \left( \vec{\theta}_D, \vec{\theta}_G, z \right)$  being  $|real\rangle\langle real|$ , weighted by  $-1/4$ .  $V^{DR} = 1/4M \cdot \sum_{k=1}^M \text{Tr} \left\{ Z \rho_k^{DR_t} \left( \vec{\theta}_D \right) \right\}$  represented the average probability of  $\rho_k^{DR_t} \left( \vec{\theta}_D \right)$  being  $|real\rangle\langle real|$ , weighted by  $-1/4$ .  $V^{GAN} \equiv 1/2 + V^{DG} + V^{DR}$ ; this reflected the capability of  $\hat{D}$  to identify the generated data.  $V^{DG}$ ,  $V^{DR}$ , and  $V^{GAN}$  converged to  $1/4$ ,  $-1/4$ , and  $1/2$ , respectively; thus,  $\hat{D}$  could not provide a fair judgement and designate all the density matrices as  $|real\rangle$ . The average probability of matching the label with the generated data was given by  $V^{ML} = -1/M \cdot \sum_{k=1}^M \text{Tr} \left\{ |\lambda_k\rangle\langle \lambda_k| \left[ L_k \left( \vec{\theta}_T, \vec{\theta}_G, z \right) + L_k^{R_s} \left( \vec{\theta}_T \right) \right] \right\}$ , which eventually reached  $-0.739$ . As Figure 2 clearly demonstrates, the cost function and its components converged to stable values in only 1500 training steps. We tested the trained model on the classification using the test dataset mentioned above. The classification accuracy reached 95%. The results demonstrated that an equilibrium point existed for the proposed cost function and the QATL model was workable on the dataset. Due to the generative model and adversarial training, the gap of knowledge transfer between the differently distributed datasets was effectively bridged; therefore, the accuracy of the classification was largely improved compared with previous quantum algorithms [51]. The method is shown in Appendix A. We also applied our scheme to the classification of the states of the Bloch sphere; detailed descriptions are shown in Section SII of the Supplementary Materials. A high-efficiency classification accuracy was also observed.



**Figure 2.** The values of quantum cost functions as a function of the training step on the Iris dataset.  $V$  is the total cost function.  $V^{ML}$ ,  $V^{GAN}$ ,  $V^{DG}$ , and  $V^{DR}$  represent various components of the cost function and are indicated by blue, green, red, and orange lines, respectively.

#### 4. Conclusions

We demonstrated an efficient quantum machine learning scheme for cross-domain learning problems—QATL—which was a quantum counterpart of the recent and well-

received ATL. Using the well-defined quantum cost function, an adversarial training process was applied to the transfer of knowledge, which was independent of specific tasks. Our numerical experiments demonstrated that the QATL model could successfully be trained and outperformed state-of-the-art algorithms in the same tasks. The complexity of the algorithm was logarithmic in terms of the number of quantum gates and training parameters, showing an advance over ATL.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/e25071090/s1>. References [44,58,62] have been cited in the Supplementary Materials.

**Author Contributions:** Conceptualization, L.W., Y.S. and X.Z.; methodology and software, L.W.; validation, L.W., Y.S. and X.Z.; formal analysis and investigation, L.W.; writing—original draft preparation, L.W.; writing—review and editing, L.W., Y.S. and X.Z.; supervision, Y.S. and X.Z.; project administration, X.Z.; funding acquisition, Y.S. and X.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China, grant no. 2017YFA0303800, and the National Natural Science Foundation of China, grant nos. 91850205 and 11904022.

**Data Availability Statement:** The Irish dataset, accessed 8 February 2023, used in our work can be downloaded from <https://archive-beta.ics.uci.edu/>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

To complete the numerical simulation task of the theoretical scheme, we used a variational quantum circuit with eight entangled qubits from [58] to implement  $\hat{G}(\vec{\theta}_G)$ ,  $\hat{D}(\vec{\theta}_D)$ , and  $\hat{T}(\vec{\theta}_T)$ . The quantum circuit is shown in Supplementary Materials Section S1. In our task, the register **Out G** |  $\mathbf{R}_{t|s}$  required two qubits. As the data sample was encoded by a pure state, **Bath G** did not need to generate entropy. The generator  $\hat{G}(\vec{\theta}_G)$ , involving 15 variational parameters, could generate the data for this task. The register **Bath D** was not necessary. Therefore,  $\hat{D}(\vec{\theta}_D)$ , involving 42 variational parameters, operated on three qubits; one qubit served as **Out D** and the other two belonged to **Out G** |  $\mathbf{R}_{t|s}$ . For the classifier  $\hat{T}(\vec{\theta}_T)$ , we noted that **Bath T** was not necessary. Thus,  $\hat{T}(\vec{\theta}_T)$ , which had 15 variational parameters, only operated on two qubits of **Out G** |  $\mathbf{R}_{t|s}$ . The registers **Label** and **Test** required only one qubit, respectively. An additional qubit was employed to store the information of the gradient. Therefore, the workspace of the whole scheme was composed of five qubits in total.

In this scenario, we trained the gradient steps for this scheme according to the updated rules of Equation (17). At the beginning of the training sequence, the parameters were randomly chosen. Similarly, we adopted RMSProp [68] to update the learning rate. The idea of RMSProp is to divide the learning rate of a weight by a running average of the magnitudes of recent gradients for that weight. With the help of RMSProp, the learning rate is largely updated in flat directions and finely tuned in steep directions. This speeds up the training process; it has shown a good adaptation of learning rates in different applications. Here, the initial learning rates of  $\hat{G}$ ,  $\hat{D}$ , and  $\hat{T}$  were set to 0.001, 0.001, and 0.005, respectively. The training progress terminated; after that, the cost function converged at an equilibrium point. The numerical simulation results are shown in Figure 2.

## References

1. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2016.
2. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255. [[CrossRef](#)] [[PubMed](#)]

3. Yang, Q.; Zhang, Y.; Dai, W.; Pan, S.J. *Transfer Learning*, 1st ed.; Cambridge University Press: Cambridge, UK, 2020.
4. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; IEEE: New York, NY, USA, 2019; pp. 4171–4186.
5. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 1–24 September 2017; IEEE: New York, NY, USA, 2017; pp. 31–36.
6. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; Van der Maaten, L. Exploring the limits of weakly supervised pretraining. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; IEEE: New York, NY, USA, 2018; pp. 185–201.
7. Liu, B.; Wei, Y.; Zhang, Y.; Yan, Z.; Yang, Q. Transferable contextual bandit for cross-domain recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
8. Pan, W.; Xiang, E.; Liu, N.; Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, NY, USA, 11–15 July 2010.
9. Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; Krishnan, D. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Honolulu, HI, USA, 2017; pp. 95–104.
10. Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Brisbane, QLD, Australia, 2018; pp. 4243–4250.
11. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning from Simulated and Unsupervised Images through Adversarial Training. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Honolulu, HI, USA, 2017; pp. 2242–2251.
12. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Honolulu, HI, USA, 2017; pp. 5967–5976.
13. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. *Domain-Adversarial Training of Neural Networks*; Springer: Cham, Switzerland, 2017; pp. 189–209.
14. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; ACM: Sydney, Australia, 2017; pp. 1857–1865.
15. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Venice, Italy, 2017; pp. 2868–2876.
16. Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; Erhan, D. Domain separation networks. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; ACM: Barcelona, Spain, 2016.
17. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Honolulu, HI, USA, 2017; pp. 2962–2971.
18. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; ACM: Sydney, Australia, 2017.
19. Pei, Z.; Cao, Z.; Long, M.; Wang, J. Multi-Adversarial Domain Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2018; ACM: New Orleans, LA, USA, 2018.
20. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139. [[CrossRef](#)]
21. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53. [[CrossRef](#)]
22. Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.-Y. Generative adversarial networks: Introduction and outlook. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 588. [[CrossRef](#)]
23. Holt, C.A.; Roth, A.E. The Nash equilibrium: A perspective. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 3999. [[CrossRef](#)]
24. Gu, M.; Wiesner, K.; Rieper, E.; Vedral, V. Quantum mechanics can reduce the complexity of classical models. *Nat. Commun.* **2012**, *3*, 762. [[CrossRef](#)]
25. Pudenz, K.L.; Lidar, D.A. Quantum adiabatic machine learning. *Quantum Inf. Process.* **2013**, *12*, 2027. [[CrossRef](#)]
26. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195. [[CrossRef](#)]
27. Cai, X.-D.; Wu, D.; Su, Z.-E.; Chen, M.-C.; Wang, X.-L.; Li, L.; Liu, N.-L.; Lu, C.-Y.; Pan, J.-W. Entanglement-Based Machine Learning on a Quantum Computer. *Phys. Rev. Lett.* **2015**, *114*, 110504. [[CrossRef](#)]

28. Hentschel, A.; Sanders, B.C. Machine Learning for Precise Quantum Measurement. *Phys. Rev. Lett.* **2010**, *104*, 063603. [[CrossRef](#)]
29. Reberntrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [[CrossRef](#)]
30. Li, Z.; Liu, X.; Xu, N.; Du, J. Experimental Realization of a Quantum Support Vector Machine. *Phys. Rev. Lett.* **2015**, *114*, 140504. [[CrossRef](#)]
31. Wossnig, L.; Zhao, Z.; Prakash, A. Quantum Linear System Algorithm for Dense Matrices. *Phys. Rev. Lett.* **2018**, *120*, 050502. [[CrossRef](#)]
32. Levine, Y.; Sharir, O.; Cohen, N.; Shashua, A. Quantum Entanglement in Deep Learning Architectures. *Phys. Rev. Lett.* **2019**, *122*, 065301. [[CrossRef](#)]
33. Dunjko, V.; Taylor, J.M.; Briegel, H.J. Quantum-Enhanced Machine Learning. *Phys. Rev. Lett.* **2016**, *117*, 130501. [[CrossRef](#)]
34. Temme, K.; Bravyi, S.; Gambetta, J.M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*, 180509. [[CrossRef](#)]
35. Cong, I.; Choi, S.; Lukin, M.D. Quantum convolutional neural networks. *Nat. Phys.* **2019**, *15*, 1273. [[CrossRef](#)]
36. Nguyen, N.; Chen, K.-C. Bayesian Quantum Neural Networks. *IEEE Access* **2022**, *10*, 54110. [[CrossRef](#)]
37. Zhao, Z.; Pozas-Kerstjens, A.; Reberntrost, P.; Wittek, P. Bayesian deep learning on a quantum computer. *Quantum Mach. Intell.* **2019**, *1*, 41. [[CrossRef](#)]
38. Li, Y.; Zhou, R.-G.; Xu, R.; Luo, J.; Hu, W. A quantum deep convolutional neural network for image recognition. *Quantum Sci. Technol.* **2020**, *5*, 044003. [[CrossRef](#)]
39. Amin, M.H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. Quantum Boltzmann Machine. *Phys. Rev. X* **2018**, *8*, 021050. [[CrossRef](#)]
40. Kieferová, M.; Wiebe, N. Tomography and generative training with quantum Boltzmann machines. *Phys. Rev. A* **2017**, *96*, 062327. [[CrossRef](#)]
41. Niu, M.Y.; Zlokapa, A.; Broughton, M.; Boixo, S.; Mohseni, M.; Smelyanskiy, V.; Neven, H. Entangling Quantum Generative Adversarial Networks. *Phys. Rev. Lett.* **2022**, *128*, 220505. [[CrossRef](#)] [[PubMed](#)]
42. Lloyd, S.; Weedbrook, C. Quantum Generative Adversarial Learning. *Phys. Rev. Lett.* **2018**, *121*, 040502. [[CrossRef](#)] [[PubMed](#)]
43. Hu, L.; Wu, S.H.; Cai, W.; Ma, Y.; Mu, X.; Xu, Y.; Wang, H.; Song, Y.; Deng, D.; Zou, C.; et al. Quantum generative adversarial learning in a superconducting quantum circuit. *Sci. Adv.* **2019**, *5*, eaav2761. [[CrossRef](#)]
44. Dallaire-Demers, P.L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324. [[CrossRef](#)]
45. Li, T.; Zhang, S.; Xia, J. Quantum Generative Adversarial Network: A Survey. *Mater. Contin.* **2020**, *64*, 401. [[CrossRef](#)]
46. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242. [[CrossRef](#)]
47. Qi, J.; Tejedor, J. Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks. In Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; IEEE: Singapore, 2022; pp. 8627–8631.
48. Mari, A.; Bromley, T.R.; Izaac, J.; Schuld, M.; Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **2020**, *4*, 340. [[CrossRef](#)]
49. Gokhale, A.; Pande, M.B.; Pramod, D. Implementation of a quantum transfer learning approach to image splicing detection. *Int. J. Quantum Inf.* **2020**, *18*, 2050024. [[CrossRef](#)]
50. Zen, R.; My, L.; Tan, R.; Hébert, F.; Gattobigio, M.; Miniatura, C.; Poletti, D.; Bressan, S. Transfer learning for scalability of neural-network quantum states. *Phys. Rev. E* **2020**, *101*, 053301. [[CrossRef](#)]
51. Wang, L.; Sun, Y.; Zhang, X. Quantum deep transfer learning. *New J. Phys.* **2021**, *23*, 103010. [[CrossRef](#)]
52. Liu, J.; Zhong, C.; Otten, M.; Chandra, A.; Cortes, C.L.; Ti, C.; Gray, S.K.; Han, X. Quantum Kerr learning. *Mach. Learn. Sci. Technol.* **2023**, *4*, 025003. [[CrossRef](#)]
53. Haug, T.; Self, C.N.; Kim, M.S. Quantum machine learning of large datasets using randomized measurements. *Mach. Learn. Sci. Technol.* **2023**, *4*, 015005. [[CrossRef](#)]
54. Sheng, Y.-B.; Zhou, L. Distributed secure quantum machine learning. *Sci. Bull.* **2017**, *62*, 1025. [[CrossRef](#)]
55. Li, W.; Deng, D.L. Recent advances for quantum classifiers. *Sci. China Phys. Mech. Astron.* **2022**, *65*, 220301. [[CrossRef](#)]
56. Li, W.; Lu, S.; Deng, D.L. Quantum federated learning through blind quantum computing. *Sci. China Phys. Mech. Astron.* **2021**, *64*, 100312. [[CrossRef](#)]
57. Hu, F.; Wang, B.-N.; Wang, N.; Wang, C. Quantum machine learning with d-wave quantum computer. *Quantum Eng.* **2019**, *1*, e12. [[CrossRef](#)]
58. Schuld, M.; Bocharov, A.; Svore, K.M.; Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **2020**, *101*, 032308. [[CrossRef](#)]
59. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [[CrossRef](#)]
60. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [[CrossRef](#)] [[PubMed](#)]
61. Park, G.; Huh, J.; Park, D.K. Variational quantum one-class classifier. *Mach. Learn. Sci. Technol.* **2023**, *4*, 015006. [[CrossRef](#)]
62. Schuld, M.; Petruccione, F. *Supervised Learning with Quantum Computers*; Springer: Berlin, Germany, 2018.
63. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O’Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [[CrossRef](#)] [[PubMed](#)]

64. Liu, H.L.; Wu, Y.S.; Wan, L.C.; Pan, S.J.; Qin, S.J.; Gao, F.; Wen, Q.Y. Variational quantum algorithm for the Poisson equation. *Phys. Rev. A* **2021**, *104*, 022418. [[CrossRef](#)]
65. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023. [[CrossRef](#)]
66. Blank, C.; Da Silva, A.J.; De Albuquerque, L.P.; Petruccione, F.; Park, D.K. Compact quantum kernel-based binary classifier. *Quantum Sci. Technol.* **2022**, *7*, 045007. [[CrossRef](#)]
67. Yang, Z.; Zhang, X.D. Entanglement-based quantum deep learning. *New J. Phys.* **2020**, *22*, 033041. [[CrossRef](#)]
68. Xu, D.; Zhang, S.; Zhang, H.; Mandic, D.P. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Netw.* **2021**, *139*, 17. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.