applied sciences

IMPACT FACTOR 2.5

CITESCORE 5.3

Article

Quantifying Quantum Coherence Using Machine Learning Methods

Lin Zhang, Liang Chen, Qiliang He and Yeqi Zhang

Topic
Quantum Information and Quantum Computing, 2nd Volume

Edited by
Dr. Durdu Guney and Dr. David Petrosyan

MDPI

https://doi.org/10.3390/app14167312

applied sciences

Article

Quantifying Quantum Coherence Using Machine Learning Methods

Lin Zhang [1], Liang Chen [1,2,3], Qiliang He [4] and Yeqi Zhang [1,2,*]

[1] School of Mathematics and Physics, North China Electric Power University, Beijing 102206, China; 120212209097@ncepu.edu.cn (L.Z.); slchern@ncepu.edu.cn (L.C.)
[2] Institute of Condensed Matter Physics, North China Electric Power University, Beijing 102206, China
[3] Hebei Key Laboratory of Physics and Energy Technology, North China Electric Power University, Baoding 071003, China
[4] School of Physics and Electronics, Guizhou Normal University, Guiyang 550001, China; heliang005@163.com
[*] Correspondence: yqzhang@ncepu.edu.cn

Abstract: Quantum coherence is a crucial resource in numerous quantum processing tasks. The robustness of coherence provides an operational measure of quantum coherence, which can be calculated for various states using semidefinite programming. However, this method depends on convex optimization and can be time-intensive, especially as the dimensionality of the space increases. In this study, we employ machine learning techniques to quantify quantum coherence, focusing on the robustness of coherence. By leveraging artificial neural networks, we developed and trained models for systems with different dimensionalities. Testing on data samples shows that our approach substantially reduces computation time while maintaining strong generalizability.

Keywords: quantum coherence; robustness of coherence; machining learning; artificial neural network

1. Introduction

Quantum coherence refers to the possibility of creating a superposition of a set of orthogonally distinguishable states in quantum physics and quantum information science [1]. It is an indispensable element in multi-particle interference, entanglement, and other phenomena, and can also be regarded as a quantum resource in some quantum processing tasks [2–4]. Especially in the field of thermodynamics, the theory of coherent quantum resources has brought new enlightenment and has energy value in terms of thermodynamic work [5,6]. Recent studies have shown that quantum coherence can serve as a useful resource for improving the performance of some thermal machines [7–9] and the charging power of quantum batteries [10,11].

There are many types of quantum coherence measures like coherence measure based on entanglement and discord [12–14], intrinsic randomness of coherence [15], the basis independent set coherence [16], coherence distillation and cost [3,17], measures based on Fisher information [18], and relative entropy of coherence [19,20]. The robustness of coherence [21,22], serving as an operational measure of quantum coherence, is not only feasible to observe experimentally but also numerically assessable through semidefinite programming. In practice, fast and accurate calculations of coherence values are crucial. However, this is a challenge because this computational process relies on convex optimization techniques, and the time required can increase significantly as the state dimension of the processing increases.

Machine learning methods have been recently brought to the field of quantum physics and quantum information to address problems such as quantum-state tomography and estimation [23,24], quantum error correction code [25], wave function reconstruction [26]. In particular, it is found that machine learning methods are useful in entanglement detection [27–30] and quantification [31]. Meanwhile, machine learning methods are adopted in

check for updates

Citation: Zhang, L.; Chen, L.; He, Q.; Zhang, Y. Quantifying Quantum Coherence Using Machine Learning Methods. *Appl. Sci.* **2024**, *14*, 7312. https://doi.org/10.3390/app14167312

Academic Editor: Wilhelm Becker

Received: 11 July 2024
Revised: 12 August 2024
Accepted: 14 August 2024
Published: 20 August 2024

the classification of Einstein–Podolsky–Rosen (EPR) steering [32]. Recently, Zhang et al. presented a semisupervised support vector machine method on EPR steering that showed a significant improvement in accuracy and labor-saving [33]. Methods of artificial neural network to quantify two-qubit steerability have also been proposed [34]. Therefore, machine learning plays a positive role in the detection and quantification of quantum information resources.

Inspired by these recent research progresses, in this work we combine machine learning methods with the quantification of quantum coherence measured by the robustness of coherence. We first obtain the dataset of labeled quantum states by using the semidefinite programming. Then, we train artificial neural networks (ANNs) for two-qubit, qubit-qutrit and three-qubit systems. ANNs are computational models inspired by biological neural systems. In this work, we utilize a backpropagation (BP) neural network, a type of multi-layer feedforward network trained via the error backpropagation algorithm, which is among the most widely used neural network models [35]. This method can straightforwardly be extended to arbitrary dimensional quantum systems. By applying the trained model to test samples and an example of non-Markovian dynamics of coherence, we show the trained models have strong generalization ability.

This paper is organized as follows. We briefly review the ROC from the viewpoint of quantum resource theory, the feedforward neural network and the methods for data preprocessing in Section 2. We describe our results of training the model for two-qubit, qubit-qutrit and three-qubit systems in Section 3. A conclusion is given in Section 4.

2. Materials and Methods

In this section, we first briefly introduce the concepts of quantum coherence and the quantifier concerned in Section 2.1. Then, we briefly review the machine learning method we adopted, i.e., the feedforward neural network in Section 2.2. Finally, we describe the methods for data processing in Section 2.3.

2.1. The Robustness of Coherence

Let us analyze the quantifier of quantum coherence within the framework of quantum resource theory (QRT). A QRT consists of three main components: free states, resource states, and free or restricted operations. Mathematically, a quantum state is represented by a normalized Hermitian operator ρ that satisfies $\rho^\dagger = \rho$ and $\mathrm{tr}\{\rho\} = 1$. In a specific basis, it is simply a Hermitian matrix with a trace of one. A quantum operation is defined as a linear, completely positive map from the set of density operators to itself [36]. A consistent QRT ensures that resource states cannot be generated from free states using free operations.

We adhere to the framework of quantum coherence QRT as described by Baumgratz et al. [2]. In this framework, the free states are those with diagonal density matrices in a specific basis, represented as

$$\delta = \sum_i \delta_i |i\rangle\langle i|, \tag{1}$$

where $\{|i\rangle\}$ is a fixed reference basis in a finite-dimensional Hilbert space and δ_i forms a probability distribution. In the context of quantum coherence QRT, these free states are referred to as incoherent states, and the set of incoherent states is denoted by $\mathcal{I}$. The resource states, known as coherent states, are those that cannot be expressed in this form.

In order to characterize the set of free operations, we recall that quantum operations are specified by a set of Kraus operators $\{K_n\}$ satisfying $\sum_n K_n^\dagger K_n = I$. For a quantum operation, the corresponding Kraus representation is not unique. The free operations, called incoherent operations, are thus defined as those operations for which there exist a Kraus representation $\{K_n\}$ such that $K_n \rho K_n^\dagger / \,\mathrm{tr}\big(K_n \rho K_n^\dagger\big) \in \mathcal{I}$ for all n and all $\rho \in \mathcal{I}$. The restriction guarantees that in an overall quantum operation $\rho \mapsto \sum_n K_n \rho K_n^\dagger$, quantum coherence cannot be generated from incoherent input states, even if someone has access to individual measurement outcomes n, not even probabilistically.

Now, let us look at the robustness quantifier for quantum coherence, i.e., the robustness of coherence (ROC). Note that quantum coherence can be treated as a useful resource in certain quantum tasks, and the robustness of a resource can be defined within a general resource theory [37]. Let $\mathcal{D}(\mathbb{C}^d)$ be the convex set of density operators acting on a d-dimensional Hilbert space. The ROC of a quantum state $\rho \in \mathcal{D}(\mathbb{C}^d)$ is defined as

$$\mathcal{C}_{\mathcal{R}}(\rho) = \min_{\tau \in \mathcal{D}(\mathbb{C}^d)} \left\{ s \geq 0 \middle| \frac{\rho + s\tau}{1+s} =: \delta \in \mathcal{I} \right\}. \tag{2}$$

Intuitively, given a quantum state ρ, one can mix it with another state τ according to the weight s. The resulting normalized state may be either coherent or not. Hence, the ROC of ρ can be viewed as the minimum weight of another state τ, when their convex mixture yields an incoherent state δ. For an incoherent state ρ which already belongs to $\mathcal{I}$, it is no need to be mixed with another state to achieve incoherence. Hence, the ROC for an incoherent state is zero. Moreover, in the Equation (2), τ can be a coherent state as well as an incoherent state, otherwise, the minimum s would diverge for any state ρ that possesses nonzero coherence, for the non-diagonal elements of the state matrix cannot be eliminated.

In order to show the property of computability for the ROC, one must first realize that any state ρ can be reduced to an incoherent one through a dephasing operation $\Delta(\rho) = \sum_i |i\rangle\langle i|\rho|i\rangle\langle i|$ in the reference basis $\{|i\rangle\}$. Then, one must introduce the notion of coherence witnesses. A coherence witness is an observable represented by a Hermitian operator W satisfying $\Delta(W) \geq 0$ if and only if $\mathrm{tr}(\delta W) = \mathrm{tr}(\delta\Delta(W)) \geq 0$ for all incoherent state $\delta \in \mathcal{I}$. Observing $\mathrm{tr}(\rho W) < 0$ indicates that the state ρ has coherence. Then, the evaluation of $\mathcal{C}_{\mathcal{R}}$ can be recast [22] as a semidefinite program (SDP) [38]:

$$\begin{aligned} \max \quad & -\mathrm{tr}(W\rho), \\ \text{s.t.} \quad & W \leq I, \\ & \Delta(W) \geq 0. \end{aligned} \tag{3}$$

Using the open-source MATLAB-based CVX modeling system for convex optimization [39,40], Piani et al. [21] developed MATLAB code to assess the robustness of asymmetry and coherence in arbitrary quantum states. Notably, linking the robustness of coherence (ROC) with witness operators makes ROC especially useful for detecting coherence effects in energy transport phenomena within light-harvesting systems [41,42].

2.2. The Feedforward Neural Network

The feedforward neural network (FNN) is the earliest and simplest form of artificial neural networks (ANNs) introduced [43]. In this network structure, data flow in a single direction, from the input layer to the output layer. There are also other types of ANNs, such as recurrent neural networks, wherein connections between nodes may form loops. We adopt an FNN in the present paper. If using the backpropagation algorithm [44] to train the FNN, one also refers to this network model as BP neural network.

In Figure 1, we depict the structure of the BP neural network. It comprises five layers: one input layer, three hidden layers, and one output layer. The number of hidden layers and the number of nodes in each hidden layer may be adjusted based on the specific problem and the performance of the predictions. The BP neural network can address both classification and regression problems. Since the network needs to output the ROC value of a quantum state, which is a continuous number, there is only one node in the output layer. The number of nodes in the input layer is related to the dimensionality of the density matrix and should equal the number of variables contained in the matrix elements. Specifically, if $\rho \in \mathcal{D}(\mathbb{C}^d)$, the number of nodes in the input layer will be d^2 if we restrict to real entries, because it must match the number of matrix elements. While designing the number of nodes for the input and output layers is straightforward, designing the hidden layers is more complex [45]. There are several heuristics useful for designing the hidden layers, and the actual structure should ultimately be determined by its performance.

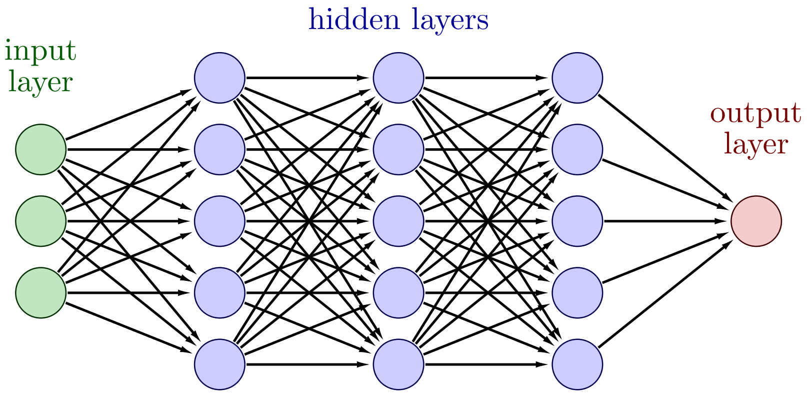

Figure 1. A schematic diagram of neural network for regression purpose with five layers. From left to right, there is one input layer, three hidden layers, and one output layer. Note that there is only one output unit for our regression purposes.

Let us explore how backpropagation (BP) is utilized for training feedforward neural networks (FNNs). Each element in our dataset consists of matrix elements arranged in a vector $\boldsymbol{x}$ and the corresponding ROC value y. If the dataset $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_N, y_N)\}$ with N samples is collected, we have to randomly split it into two sets: one for training the model and the other one for testing the performance. The overall network operates through $g(\boldsymbol{x}) = f^L(W^L f^{L-1}(W^{L-1} \cdots f^1(W^1 \boldsymbol{x}) \cdots))$, where L represents the total number of layers, while $W^l = (w^l_{jk})$ denotes the weight matrix connecting layer $l-1$ to layer l, with w^l_{jk} indicating the weight between the k-th node in layer $l-1$ and the j-th node in layer l. The function f^l denotes the activation function at layer l, which is crucial for capturing nonlinear properties in the problem. In this study, we employ the ReLU activation function $\mathrm{ReLU}(x) = \max(0, x)$.

The initialization of weights is typically randomized. Training the network is essential for adjusting these weights to minimize the average discrepancy between the network's prediction $g(\boldsymbol{x})$ and the actual target y. This discrepancy is measured using a loss function. We choose the mean-squared error (MSE) as the loss function:

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (g(\boldsymbol{x}_i) - y_i)^2, \tag{4}$$

because for small errors, MSE can be advantageous due to its sensitivity to error magnitudes [46].

Updating the weight w^l_{ij} involves assessing how changes in w^l_{ij} impact the loss function E. If $\partial E / \partial w^l_{ij} > 0$, increasing w^l_{ij} increases E, so we subtract a suitable value Δw^l_{ij}. To facilitate this, we employ a fixed learning rate $\eta > 0$. Similar reasoning applies when $\partial E / \partial w^l_{ij} < 0$. This technique is known as gradient descent, where $\Delta w^l_{ij} = -\eta(\partial E / \partial w^l_{ij})$.

A common issue in neural networks is overfitting, where the model performs exceptionally well on the training data but poorly on new, unseen data. To detect overfitting, a portion of the training data is reserved as a validation set. Monitoring the accuracy on this set ensures the model comprehends the data rather than simply memorizing it. Training halts once accuracy on the validation data plateaus, a strategy known as early stopping.

Post-training, evaluating the model on a test set is crucial. In addition to MSE, we use R^2 as an evaluation metric:

$$R^2 = 1 - \frac{\sum_i (y_i - g(\boldsymbol{x}_i))^2}{\sum_i (y_i - \bar{y})^2}, \tag{5}$$

where $\bar{y}$ represents the mean of the target values in the test set. The R^2 value ranges from 0 to 1, with higher values signifying a closer fit between the predictions and the actual values.

2.3. Data Preparation

We begin by preparing datasets for training BP neural networks. Initially, we focus on two-qubit systems, where each Hilbert space dimension is 2, resulting in 4×4 matrices for states ρ. Drawing inspiration from [47], we generate 6×10^6 density matrices, each containing 16 matrix elements. These matrices are reshaped into vectors $\boldsymbol{x}_i$ with 16 entries, using real entries for simplicity. For each density matrix ρ_i, we compute its corresponding ROC and label it as the target y_i. This completes the dataset collection, where each data point is represented as $\{x_i, y_i\}$ with 17 entries.

Next, we extend our discussion to qubit-qutrit and three-qubit systems, where the Hilbert space dimensions are 6 and 8, respectively. States in these systems are described by 6×6 and 8×8 matrices. Consequently, each dataset entry consists of 37 and 65 entries, respectively, consisting of ROC values and matrix elements. We partition the entire dataset into training and test sets, reserving 20% for testing and utilizing the remaining 80% for training. The training set is further divided, with 20% allocated to a validation set and the remaining 80% serving as the training subset.

Our neural networks are implemented using TensorFlow [48], featuring 3 hidden layers. We adopt ReLU as the activation function and Adam as the gradient descent optimizer. The learning rate is set to 0.001, and the batch size is 64. To mitigate overfitting, we employ an early stopping strategy. Notably, achieving an optimal network structure often requires hyperparameter optimization [49]. In this study, we systematically adjust parameters such as layer count, node density per layer, and learning rates based on observed performance to finalize our network configurations. The machine learning code and data generation scripts used in this paper can be found in [50].

3. Results

Now, let us train the model for ρ for the two-qubit systems. The numbers of nodes in the three hidden layers are all set to 64 and the number of trainable parameters for this network is 9473. The loss functions (MSE) are plotted against epoch for both the training set and validation set in Figure 2. It can be shown in Figure 2 that both errors decrease dramatically in the first few epochs. As we continue the training, the MSE for the training set keeps decreasing, while the MSE for the validation set has some ups and downs. This indicates the possibility of overfitting. Now, we use the early stopping strategy that in case the loss function does not improve over 10 epochs, the training stops. Eventually, the training process finishes at an epoch of about 38.

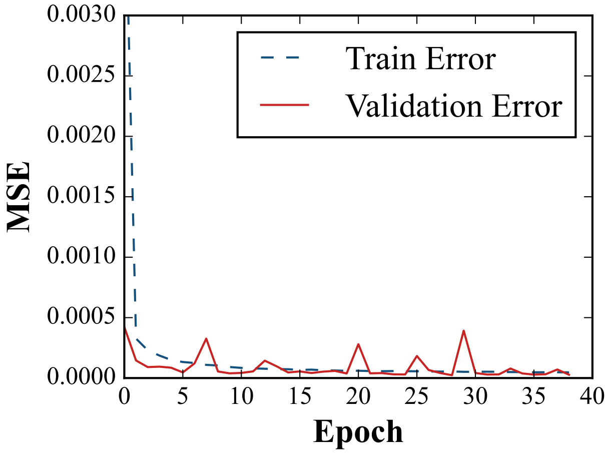

Figure 2. The loss functions (MSE) are plotted against epoch for both training set (dashed line) and validation set (solid line) with $d = 4$. Early stopping strategy has been used.

In order to see the generalization ability of the trained model, we apply it to the test set. The MSE for the test set is 2.3644×10^{-4}. These error values are small and imply the trained network has a good generalization on the test set. In order to see this more clearly, we plot the predicted ROC versus the actual ROC for the test set in Figure 3. The horizontal and vertical coordinates of each hollow circle in the figure represent the actual values and predicted values of ROC, respectively, by the trained model for one sample in the test set. The red line indicates the case that the predicted ROC equals the actual ROC. The closer these dots are to the line, the more accurate the predicted ROC is. Moreover, we also calculate the $R^2 = 0.9994$, which is high and close to 1. We can see that the trained model has a strong generalization ability on the test set.

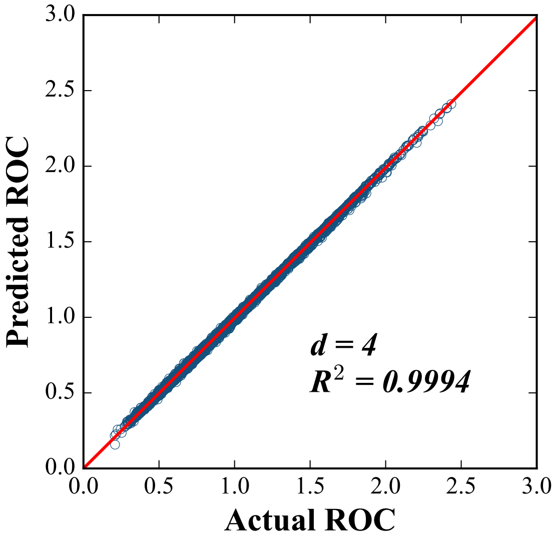

Figure 3. The predicted ROC versus the actual ROC for 4-dimensional space test set are marked with blue circles. The red line indicates the case that the predicted ROC equals to the actual ROC.

As an application, we now apply the trained model to the coherence dynamics for a two-qubit system under a non-Markovian environment. This system is composed of two parts, each one consisting of a two-level system interacting with a reservoir [51]. The Hamiltonian of the system is

$$H = \hbar\omega_0\sigma_+\sigma_- + \sum_k \left[\hbar\omega_k b_k^\dagger b_k + \left(g_k\sigma_+ b_k + g_k^*\sigma_- b_k^\dagger\right)\right], \tag{6}$$

where ω_0 is the transition frequency, $\sigma_\pm$ is the system raising and lowering operators for the qubit, $b_k^\dagger$ (b_k) is the creation (annihilation) operator, and g_k is the coupling of the mode k with frequency ω_k. If the initial state is the Werner-like state

$$\rho = r|\Phi\rangle\langle\Phi| + \frac{1-r}{4}I \tag{7}$$

with r the purity and the coupling are non-Markovian with spectral density

$$J(\omega) = \frac{1}{2\pi}\frac{\Gamma\lambda^2}{(\omega_0-\omega)^2+\lambda^2}, \tag{8}$$

where Γ is the system–reservoir coupling constant and λ is the spectral width of the coupling. We obtain the density matrix of the qubits system according to Ref. [51] and calculate the ROC using both the SDP method and the FNN. In Figure 4, we plot the predicted and the actual ROC as a function of the dimensionless quantity Γt. It can be seen that our trained model has a good generalization. Moreover, it is worth noting that the time required to process a single state using the FNN is approximately 10^{-6} s, whereas using the SDP method directly takes around 10^{-1} s. This means our method is roughly 10^5 times faster than the SDP approach.

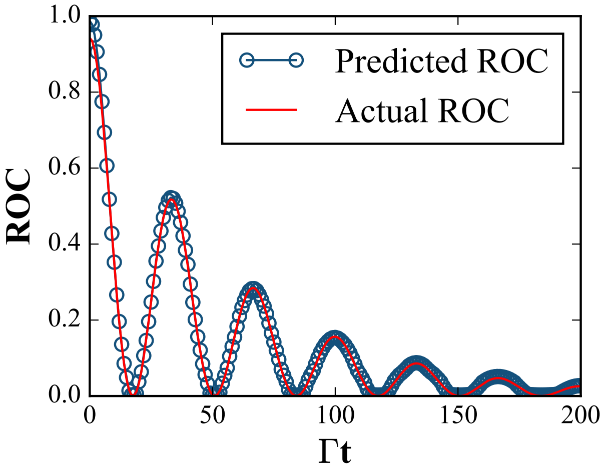

Figure 4. The predicted and the actual ROC as a function of the dimensionless quantity Γt for initial state $\rho^{\Phi}(0)$ with $r = 1$, $\alpha^2 = 1/3$, and $\lambda/\Gamma = 0.01$.

Next, we discuss larger dimension states of the qubit-qutrit and the three-qubit systems. The Hilbert space dimensions of these two systems are 6 and 8, respectively. In these cases, the performance decreases when using the same structure of neural network model as in previous two-qubit system. Therefore, we increase the number of nodes in each hidden layer to 256. As a result, the number of trainable parameters will increase. For example, in three-qubit case, we have a total of 148,481 trainable parameters. Apart from this change, methods of data preparation and training processes of the neural network are the same with those of the two-qubit system. As shown in Figure 5, the trainings for the qubit-qutrit system and the three-qubit system accomplish within training epochs of 38 and 72, respectively. The MSE of the qubit-qutrit system is 8.5269×10^{-4}. For the three-qubit system, the MSE is 0.0035, which is relatively larger than the qubit-qutrit system. This is because using the same neural network structure for more complex problems may result in slightly inferior outcomes. The predicted ROC curve overlaps extremely well with the actual ROC curve, as shown in Figure 6, with R^2 values reaching 0.9988 and 0.9688, respectively. Therefore, we can conclude that in the case of the qubit-qutrit and the three-qubit systems, the trained neural network method exhibit strong generalization ability.

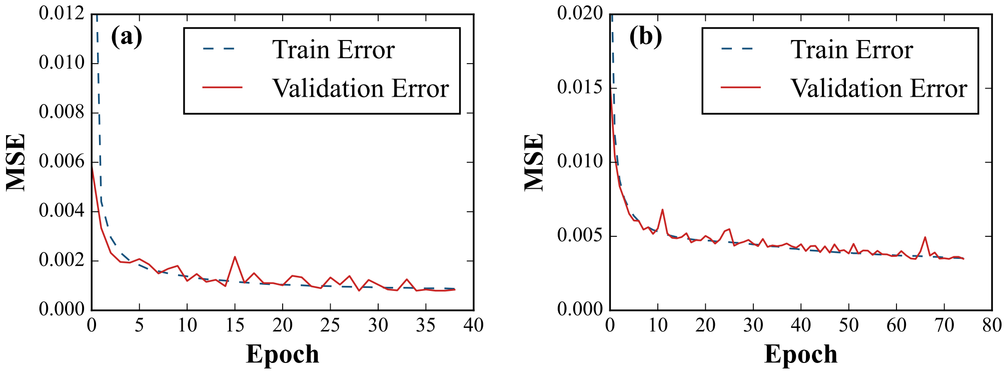

Figure 5. The loss functions (MSE) are plotted against epoch for both training set (dashed line) and validation set (solid line) with (**a**) $d = 6$ and (**b**) $d = 8$, respectively. Early stopping strategy has been used.

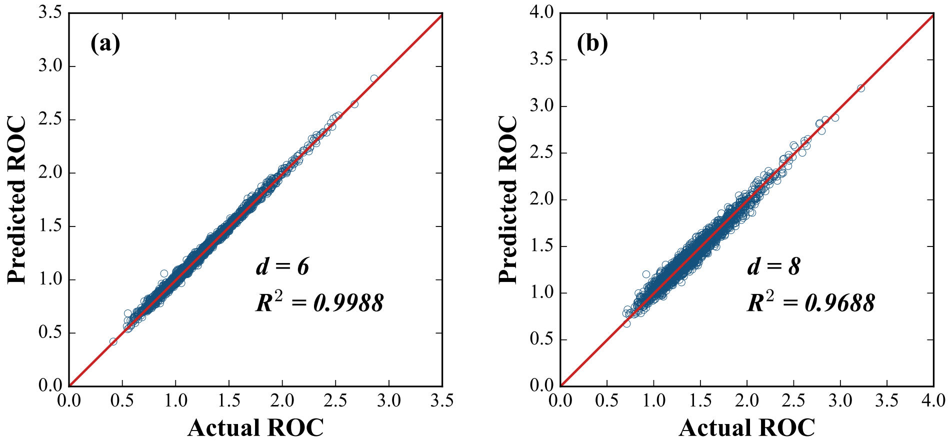

Figure 6. The predicted ROC vs. the actual ROC for (**a**) 6- and (**b**) 8-dimensional spaces are marked with blue circles, respectively. The red line indicates the case that the predicted ROC equals to the actual ROC.

4. Conclusions

In this study, we have applied machine learning to address the quantification of quantum coherence. Utilizing the SDP program, we computed the ROC and created labeled datasets for two-qubit, qubit-qutrit, and three-qubit systems, subsequently training FNNs for each. This method can be easily extended to other quantum systems with varying dimensions. We demonstrated the robust generalization capabilities of the FNNs by testing them on sample datasets. Additionally, we predicted the quantum coherence dynamics in a two-qubit system within a non-Markovian environment and compared these predictions with those obtained using the CVX method. It turns out that this approach is very accurate and time-saving. Our results highlight the considerable potential of machine learning techniques in the quantification of quantum coherence.

Author Contributions: Conceptualization, L.Z. and Y.Z.; methodology, L.C. and L.Z.; investigation, L.Z.; data curation, L.Z. and Q.H.; writing—original draft preparation, L.Z.; writing—review and editing, L.C., Q.H. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Y.Z. acknowledges financial support from the NSFC (under Grant No. 11805065). L.C. was supported by the NSFC (under Grant No. 12174101) and the Fundamental Research Funds for the Central Universities (under Grant No. 2022MS051). Q.H. was supported by the NSFC (under Grant No. 11364006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Streltsov, A.; Adesso, G.; Plenio, M.B. Colloquium: Quantum coherence as a resource. *Rev. Mod. Phys.* **2017**, *89*, 041003. [CrossRef]
2. Baumgratz, T.; Cramer, M.; Plenio, M.B. Quantifying Coherence. *Phys. Rev. Lett.* **2014**, *113*, 140401. [CrossRef]
3. Winter, A.; Yang, D. Operational Resource Theory of Coherence. *Phys. Rev. Lett.* **2016**, *116*, 120404. [CrossRef]
4. Cunden, F.D.; Facchi, P.; Florio, G.; Gramegna, G. Generic aspects of the resource theory of quantum coherence. *Phys. Rev. A* **2021**, *103*, 022401. [CrossRef]
5. Bernardo, B.d.L. Unraveling the role of coherence in the first law of quantum thermodynamics. *Phys. Rev. E* **2020**, *102*, 062152. [CrossRef]
6. Kwon, H.; Jeong, H.; Jennings, D.; Yadin, B.; Kim, M.S. Clock–Work Trade-Off Relation for Coherence in Quantum Thermodynamics. *Phys. Rev. Lett.* **2018**, *120*, 150602. [CrossRef] [PubMed]

7. Camati, P.A.; Santos, J.F.G.; Serra, R.M. Coherence effects in the performance of the quantum Otto heat engine. *Phys. Rev. A* **2019**, *99*, 062103. [CrossRef]
8. Van Vu, T.; Saito, K. Finite-Time Quantum Landauer Principle and Quantum Coherence. *Phys. Rev. Lett.* **2022**, *128*, 010602. [CrossRef]
9. Hammam, K.; Hassouni, Y.; Fazio, R.; Manzano, G. Optimizing autonomous thermal machines powered by energetic coherence. *New J. Phys.* **2021**, *23*, 043024. [CrossRef]
10. Monsel, J.; Fellous-Asiani, M.; Huard, B.; Auffèves, A. The Energetic Cost of Work Extraction. *Phys. Rev. Lett.* **2020**, *124*, 130601. [CrossRef]
11. Seah, S.; Perarnau-Llobet, M.; Haack, G.; Brunner, N.; Nimmrichter, S. Quantum Speed-Up in Collisional Battery Charging. *Phys. Rev. Lett.* **2021**, *127*, 100601. [CrossRef] [PubMed]
12. Streltsov, A.; Singh, U.; Dhar, H.S.; Bera, M.N.; Adesso, G. Measuring Quantum Coherence with Entanglement. *Phys. Rev. Lett.* **2015**, *115*, 020403. [CrossRef] [PubMed]
13. Yao, Y.; Xiao, X.; Ge, L.; Sun, C.P. Quantum coherence in multipartite systems. *Phys. Rev. A* **2015**, *92*, 022112. [CrossRef]
14. Hu, M.L.; Hu, X.; Wang, J.; Peng, Y.; Zhang, Y.R.; Fan, H. Quantum coherence and geometric quantum discord. *Phys. Rep.* **2018**, *762–764*, 1–100. [CrossRef]
15. Yuan, X.; Zhou, H.; Cao, Z.; Ma, X. Intrinsic randomness as a measure of quantum coherence. *Phys. Rev. A* **2015**, *92*, 022124. [CrossRef]
16. Designolle, S.; Uola, R.; Luoma, K.; Brunner, N. Set Coherence: Basis-Independent Quantification of Quantum Coherence. *Phys. Rev. Lett.* **2021**, *126*, 220404. [CrossRef]
17. Chitambar, E.; Streltsov, A.; Rana, S.; Bera, M.N.; Adesso, G.; Lewenstein, M. Assisted Distillation of Quantum Coherence. *Phys. Rev. Lett.* **2016**, *116*, 070402. [CrossRef]
18. Li, L.; Wang, Q.W.; Shen, S.Q.; Li, M. Quantum coherence measures based on Fisher information with applications. *Phys. Rev. A* **2021**, *103*, 012401. [CrossRef]
19. Bu, K.; Singh, U.; Fei, S.M.; Pati, A.K.; Wu, J. Maximum Relative Entropy of Coherence: An Operational Coherence Measure. *Phys. Rev. Lett.* **2017**, *119*, 150405. [CrossRef]
20. Rastegin, A.E. Quantum-coherence quantifiers based on the Tsallis relative α entropies. *Phys. Rev. A* **2016**, *93*, 032136. [CrossRef]
21. Napoli, C.; Bromley, T.R.; Cianciaruso, M.; Piani, M.; Johnston, N.; Adesso, G. Robustness of Coherence: An Operational and Observable Measure of Quantum Coherence. *Phys. Rev. Lett.* **2016**, *116*, 150502. [CrossRef] [PubMed]
22. Piani, M.; Cianciaruso, M.; Bromley, T.R.; Napoli, C.; Johnston, N.; Adesso, G. Robustness of asymmetry and coherence of quantum states. *Phys. Rev. A* **2016**, *93*, 042107. [CrossRef]
23. Quek, Y.; Fort, S.; Ng, H. Adaptive quantum state tomography with neural networks. *NPJ Quantum Inf.* **2021**, *7*, 105. [CrossRef]
24. Lohani, S.; Kirby, B.T.; Brodsky, M.; Danaci, O.; Glasser, R.T. Machine learning assisted quantum state estimation. *Mach. Learn. Sci. Technol.* **2020**, *1*, 035007. [CrossRef]
25. Nautrup, H.P.; Delfosse, N.; Dunjko, V.; Briegel, H.J.; Friis, N. Optimizing Quantum Error Correction Codes with Reinforcement Learning. *Quantum* **2019**, *3*, 215. [CrossRef]
26. Beach, M.J.S.; Vlugt, I.D.; Golubeva, A.; Huembeli, P.; Kulchytskyy, B.; Luo, X.; Melko, R.G.; Merali, E.; Torlai, G. QuCumber: wavefunction reconstruction with neural networks. *SciPost Phys.* **2019**, *7*, 9. [CrossRef]
27. Khoo, J.Y.; Heyl, M. Quantum entanglement recognition. *Phys. Rev. Res.* **2021**, *3*, 033135. [CrossRef]
28. Harney, C.; Pirandola, S.; Ferraro, A.; Paternostro, M. Entanglement classification via neural network quantum states. *New J. Phys.* **2020**, *22*, 045001. [CrossRef]
29. Lu, S.; Huang, S.; Li, K.; Li, J.; Chen, J.; Lu, D.; Ji, Z.; Shen, Y.; Zhou, D.; Zeng, B. Separability-entanglement classifier via machine learning. *Phys. Rev. A* **2018**, *98*, 012315. [CrossRef]
30. Chen, Y.; Pan, Y.; Zhang, G.; Cheng, S. Detecting quantum entanglement with unsupervised learning. *Quantum Sci. Technol.* **2021**, *7*, 015005. [CrossRef]
31. Roik, J.; Bartkiewicz, K.; Černoch, A.; Lemr, K. Entanglement quantification from collective measurements processed by machine learning. *arXiv* **2022**, arXiv:2203.01607. [CrossRef]
32. Ren, C.; Chen, C. Steerability detection of an arbitrary two-qubit state via machine learning. *Phys. Rev. A* **2019**, *100*, 022314. [CrossRef]
33. Zhang, L.; Chen, Z.; Fei, S.M. Einstein-Podolsky-Rosen steering based on semisupervised machine learning. *Phys. Rev. A* **2021**, *104*, 052427. [CrossRef]
34. Zhang, Y.Q.; Yang, L.J.; He, Q.l.; Chen, L. Machine learning on quantifying quantum steerability. *Quantum Inf. Process.* **2020**, *19*, 263. [CrossRef]
35. Li, J.; Cheng, J.h.; Shi, J.y.; Huang, F. Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement. In Proceedings of the Advances in Computer Science and Information Engineering, Zhengzhou, China, 19–20 May 2012 ; Jin, D., Lin, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 553–558.
36. Takhtadzhian, L. *Quantum Mechanics for Mathematicians*; Graduate Studies in Mathematics; American Mathematical Society: Providence, RI, USA, 2008.
37. Brandão, F.G.S.L.; Gour, G. Reversible Framework for Quantum Resource Theories. *Phys. Rev. Lett.* **2015**, *115*, 070503. [CrossRef]
38. Vandenberghe, L.; Boyd, S. Semidefinite Programming. *SIAM Rev.* **1996**, *38*, 49–95. [CrossRef]

39. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming, Version 2.1. 2014. Available online: http://cvxr.com/cvx(accessed on 10 June 2024).
40. Grant, M.; Boyd, S. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*; Blondel, V., Boyd, S., Kimura, H., Eds.; Lecture Notes in Control and Information Sciences; Springer: Berlin/Heidelberg, Germany, 2008; pp. 95–110. Available online: https://web.stanford.edu/~boyd/papers/pdf/graph_dcp.pdf (accessed on 10 June 2024).
41. Lloyd, S. Quantum coherence in biological systems. *J. Phys. Conf. Ser.* **2011**, *302*, 012037. [CrossRef]
42. Huelga, S.; Plenio, M. Vibrations, quanta and biology. *Contemp. Phys.* **2013**, *54*, 181–207. [CrossRef]
43. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
44. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 10 June 2024).
45. Nielsen, M.A. *Neural Network and Deep Learning*; Determination Press: Los Angeles, CA, USA, 2015. Available online: http://neuralnetworksanddeeplearning.com (accessed on 10 June 2024).
46. Botchkarev, A. A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. *Interdiscip. J. Inf. Knowl. Manag.* **2019**, *14*, 045–076. [CrossRef] [PubMed]
47. Życzkowski, K.; Penson, K.A.; Nechita, I.; Collins, B. Generating random density matrices. *J. Math. Phys.* **2011**, *52*, 062201. [CrossRef]
48. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: https://www.tensorflow.org/install?hl=zh-cn (accessed on 10 June 2024).
49. Yu, T.; Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications. *arXiv* **2020**, arXiv:2003.05689.
50. GitHub—lruicckhy/CoherenceMachineLearning at Master—github.com. Available online: https://github.com/lruicckhy/CoherenceMachineLearning/tree/master (accessed on 10 August 2024).
51. Bellomo, B.; Lo Franco, R.; Compagno, G. Entanglement dynamics of two independent qubits in environments with and without memory. *Phys. Rev. A* **2008**, *77*, 032342. [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.