PAPER • **OPEN ACCESS**

Detecting quantum entanglement with unsupervised learning

To cite this article: Yiwei Chen *et al* 2022 *Quantum Sci. Technol.* **7** 015005

Quantum Science and Technology

PAPER

Detecting quantum entanglement with unsupervised learning

Yiwei Chen[1], Yu Pan[1,*], Guofeng Zhang[2,3,*] and Shuming Cheng[4,5,6,*]

[1] Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, People's Republic of China
[2] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong Special Administrative Region of China, People's Republic of China
[3] Shenzhen Research Institute, The Hong Kong Polytechnic University, Shenzhen 518057, People's Republic of China
[4] The Department of Control Science and Engineering, Tongji University, Shanghai 201804, People's Republic of China
[5] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 201804, People's Republic of China
[6] Institute for Advanced Study, Tongji University, Shanghai, 200092, People's Republic of China
[*] Authors to whom any correspondence should be addressed.

E-mail: ypan@zju.edu.cn, Guofeng.Zhang@polyu.edu.hk and shuming_cheng@tongji.edu.cn

Keywords: entanglement detection, machine learning, quantum resource

OPEN ACCESS

RECEIVED
19 June 2021

REVISED
8 October 2021

ACCEPTED FOR PUBLICATION
19 October 2021

PUBLISHED
3 November 2021

Abstract

Quantum properties, such as entanglement and coherence, are indispensable resources in various quantum information processing tasks. However, there still lacks an efficient and scalable way to detecting these useful features especially for high-dimensional and multipartite quantum systems. In this work, we exploit the convexity of samples without the desired quantum features and design an unsupervised machine learning method to detect the presence of such features as anomalies. Particularly, in the context of entanglement detection, we propose a complex-valued neural network composed of pseudo-siamese network and generative adversarial net, and then train it with only separable states to construct non-linear witnesses for entanglement. It is shown via numerical examples, ranging from two-qubit to ten-qubit systems, that our network is able to achieve high detection accuracy which is above 97.5% on average. Moreover, it is capable of revealing rich structures of entanglement, such as partial entanglement among subsystems. Our results are readily applicable to the detection of other quantum resources such as Bell nonlocality and steerability, and thus our work could provide a powerful tool to extract quantum features hidden in multipartite quantum data.

1. Introduction

Peculiar quantum features, signalled by quantum entanglement [1] and coherence [2], enable us to accomplish tasks impossible for classical systems [3], such as ensuring the security of communications and speeding up certain hard computational tasks [4, 5]. Hence, an important question naturally arises: how can the presence of these features be efficiently detected for any given quantum system? Indeed, this is a challenging task for high-dimensional and multipartite systems because quantum features usually imply correlated patterns hidden within subsystems. Taking entanglement for example, except for low-dimensional systems, e.g. $2 \otimes 2$ and $2 \otimes 3$, of which entanglement could be detected faithfully via the positive partial transpose (PPT) criterion [6], generically, it is an NP-hard problem [7]. Besides, even though at least one linear entanglement witness could be found to witness any entangled state [1, 8–10] as displayed in figure 1, there still lacks a universal and scalable way to construct such an appropriate witness for an arbitrary state in practice.

In this work, we turn to the machine learning technique which is powerful in extracting features or patterns hidden in large multipartite datasets to tackle the quantum detection problem. Recently, much progress has been achieved in this inter-disciplinary field of quantum machine learning [11]. For example, on one hand, many quantum or quantum-inspired algorithms have been developed to speed up some well-known machine learning algorithms [12–14]. On the other hand, machine learning is also a natural candidate to extract correlated features of high-dimensional quantum systems, which has found wide

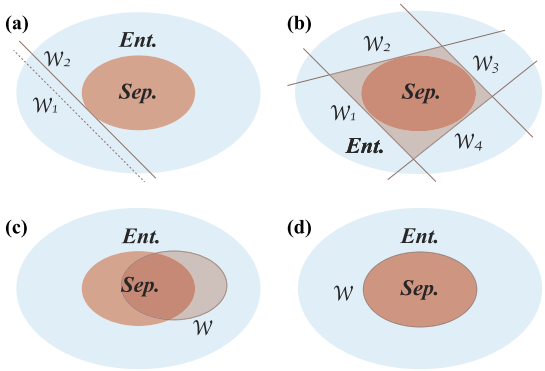

Figure 1. The geometrical interpretation of entanglement detection via entanglement witnesses. (a) Standard linear entanglement witnesses. The witness $\mathcal{W}_2$ is finer than $\mathcal{W}_1$. (b) A proper set of linear entanglement witnesses is able to form a closed area which encloses all separable states. Here $\mathcal{W}_1, \ldots, \mathcal{W}_4$ are used as examples. (c) An imperfect nonlinear entanglement witness could be generated via supervised learning method if we cannot label enough samples to cover the space of entangled states. (d) A near-perfect nonlinear witness $\mathcal{W}$ can be approximately constructed by the unsupervised neural network if the generated training data span the space of separable states.

applications in quantum control [15], state tomography [16], measurement [17, 18], and many-body problems [19–21]. Especially, the task of quantum entanglement detection can be formulated as a binary classification problem. As a consequence, various classical neural nets, trained with both entangled and separable samples, have been constructed to solve this problem via supervised learning [22–24]. However, the supervised training method requires a large pre-labelled dataset. In practice, it is time-consuming or even impossible to faithfully label a large number of entangled states in a high-dimensional space [7], thus leading these supervised methods into a dilemma.

Here, we instead build up an unsupervised model to accomplish the task of entanglement detection beyond the above issues. Following from the fact that separable states form a convex set, it becomes an anomaly detection problem of which all separable samples are labelled as normal and entangled ones are abnormal. Particularly, as shown in figure 3, a class of complex-valued neural networks composed of a pseudo-siamese network and a generative adversarial net (GAN), is constructed and then trained with very few normal samples to detect entanglement for multipartite systems, ranging from two-qubit to ten-qubit states. It is noted that our model is much more feasible than anomaly detection methods proposed in [25, 26] which require quantum hardware.

It is further illustrated in figure 1 that our unsupervised neural nets are essentially trained to search for proper nonlinear entanglement witnesses which near-perfectly construct the boundary between separable and entangled samples. Numerical results show that it is able to achieve extremely high accuracy of entanglement detection with above 97.5% on average, and even capable to detect partial entanglement within subsystems, e.g. bi-separable states in three-qubit system with accuracy above 97.7%.

Our work is organised as follows. In section 2, we give a brief introduction to the task of entanglement detection and unsupervised learning method. Then we propose an unsupervised learning neural network targeted for the detection of generic quantum features. In section 3, multipartite entanglement detection is taken as examples to illustrate the performance of our model, with only separable samples used for training. Finally, we conclude this work with a summary in section 4.

2. Unsupervised entanglement detection

2.1. The task of detecting entanglement

Entanglement is not only of significant importance to understand quantum theory at the fundamental level [1], but also has found applications in information protocols, such as quantum teleportation [27]. For a given n-partite quantum system, entanglement associated with the state is defined in a passive way in which a state ρ is entangled if and only if it cannot be described in a fully-separable form of [28]

$$\rho_{\text{sep}} = \sum_{i=1}^{m} \lambda_i \rho_i^1 \otimes \cdots \otimes \rho_i^j \otimes \cdots \otimes \rho_i^n \tag{1}$$

with non-negative coefficients satisfying $\sum_{i=1}^{m} \lambda_i = 1$. Here ρ_i^j denotes the state density matrix of the jth subsystem. Obviously, all of the separable states as per equation (1) form a convex set in the sense that any convex combination of these states in this set also belong to the same state set. It is noted that the above

definition of entanglement does not fully capture the entangled structure in the state, e.g. the partial entanglement [29], which will be discussed later.

In practice, whether a given state ρ is entangled or not, can be experimental-friendly determined via an entanglement witness [1, 10]. Indeed, as shown in figure 1(a), an entanglement witness essentially defines a hyperplane which separates the entangled state from the convex set of separable states. Furthermore, it has been shown in [1] that it is impossible for one linear witness to detect all entangled states, implying that a large set of linear witnesses illustrated in figure 1(b) (could be impractical) or certain nonlinear witness shown in figure 1(d) may be required. Besides, it becomes extremely inefficient and impractical to construct a proper witness for an arbitrary state, especially in multipartite systems. The entanglement witnesses as neural networks are experimentally accessible and has been demonstrated in [24]. In fact, since neural networks are learning the linear and nonlinear correlations on the quantum states to form a classifier, a properly parameterized neural network layer is equivalent to a set of generalized Bell's inequalities for the experimental detection of entanglement. In the following, we propose a complex-valued neural network trained in unsupervised manner to search for the nonlinear entanglement witnesses as desired.

2.2. Unsupervised learning

The unsupervised model refers to the process of learning a probability distribution over the data that has not been classified or categorized. In this situation, automated methods or algorithms must explore the underlying features from the available data and group them with similar characteristics. Specifically, the unsupervised model only receives a training set $\mathcal{S}$ that contains

$$\mathcal{S} = \{x_1, x_2, x_3, \ldots\} \tag{2}$$

without supervised target outputs $\{y_1, y_2, y_3, \ldots\}$. In contrast to supervised learning where tagging data requires a large amount of time, unsupervised learning exhibits high efficiency and self-organization in capturing patterns from untagged data.

Autoencoder [30] is a widely used unsupervised learning method that aims to learn efficient representations for a set of data. Typically, an autoencoder consists of two modules, namely encoder $\mathcal{E}$ and decoder $\mathcal{D}$, where the former learns the latent representation (encoding) for input data, and the latter is trained to generate an output as close as possible to its original input from the latent representation. Another well-known unsupervised learning method is GAN [31, 32]. Specifically, two neural networks, namely generator $\mathcal{G}$ and discriminator $\mathcal{D}$, contest with each other in the form of a zero-sum game in GAN, where the gain of one module is the loss of the other. This technique learns to generate new data with the same statistics as the training set. The siamese network [33, 34], as shown in figure 2, contains a pair of neural networks built by the same parameters, which receives two inputs and detects their difference by comparing the output vectors of the networks. The siamese network is capable of learning generic features for making predictions about an unknown distribution even when few examples from the distribution are available, which provides a competitive approach for pattern recognition without the domain-specific knowledge. In particular, the siamese network can be trained in an unsupervised manner, as the labels of the input data are not needed.

For these reasons, the method proposed in this paper has been built upon the siamese network, which is suitable for one-class unsupervised learning. The basic idea is similar to one-class support vector machine for anomaly detection [35]. That is, given a set of training samples, we aim to model the underlying distribution of the data and detect the soft boundary of this set, in order to classify new inputs as belonging to this set or not. In this case, the model will only take a training dataset without class labels as input, which means the model is a type of unsupervised learning methods.

2.3. Constructing the complex-valued neural networks

As shown in figure 3, our networks could be decomposed into two parts: one is the pseudo-siamese neural network (in the red dashed box) and the other is the GAN (in the purple dashed box). The complex-valued neural network receives the density state matrix as the input. The building modules for these networks are detailed in appendix A.

The pseudo-siamese neural network consists of two encoders sharing the same network structure, labelled as $\mathcal{E}_\mathrm{r}$ and $\mathcal{E}_\mathrm{g}$, respectively. In contrast to the original siamese network [34] which requires quadratic pairs as input, the pseudo-siamese network only requires a single input ρ_real be fed to the first encoder $\mathcal{E}_\mathrm{r}$. The second input ρ_gen to the second encoder $\mathcal{E}_\mathrm{g}$ is automatically generated by the decoder $\mathcal{G}$ whose aim is to reconstruct ρ_real. Therefore, the pseudo-siamese network trains much faster than the original siamese network while inherits its few-shot learning ability. In principle, these two encoders competes with each other to produce a pair of indistinguishable feature vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. The performance is evaluated by the

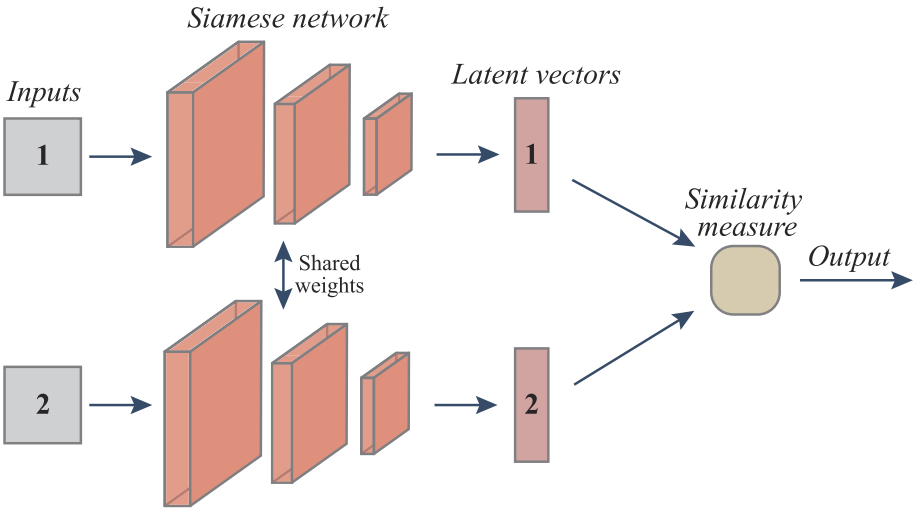

Figure 2. Pipeline of the siamese network. Two inputs are encoded into latent vectors whose difference is detected by a similarity measure.

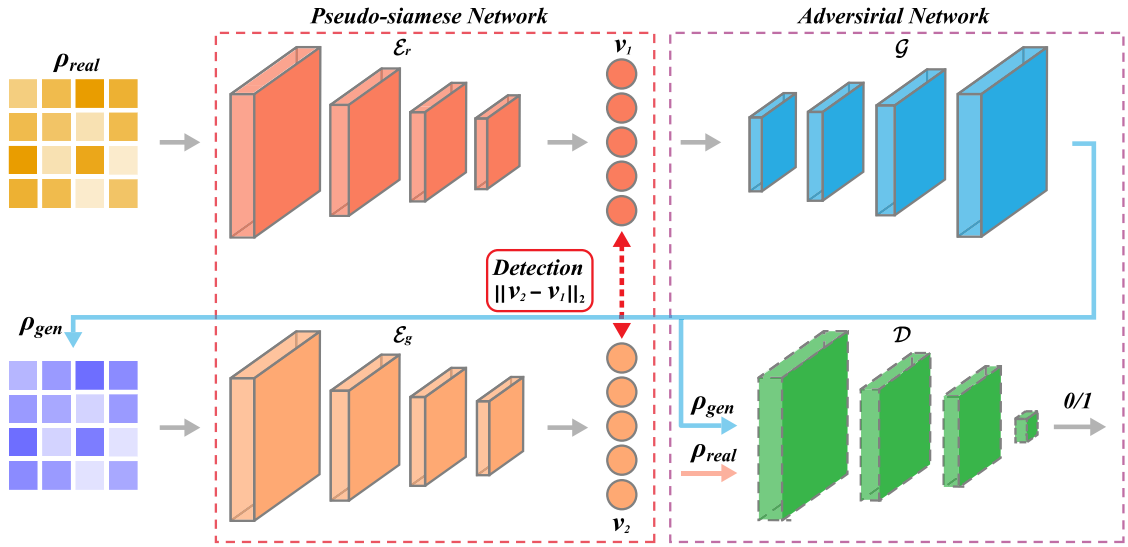

Figure 3. Structure of the complex-valued neural network. The complex-valued network is composed of two parts: one is the pseudo-siamese network in the red dashed box and the other is a GAN in the purple dashed box. The pseudo-siamese neural network consists of two encoders $\mathcal{E}_r$ and $\mathcal{E}_g$ that share the same network structure. ρ_{gen} is generated by $\mathcal{G}$. The discriminator network $\mathcal{D}$ is a binary classifier which outputs either 0 or 1. The generator $\mathcal{G}$ and discriminator $\mathcal{D}$ form the GAN, which aims to produce a ρ_{gen} that is as close as possible to ρ_{real}.

cost function

$$\mathcal{L}_1 = \mathbf{E}_{\rho_{real}} \|\mathcal{E}_r(\rho_{real}) - \mathcal{E}_g(\mathcal{G}(\mathcal{E}_r(\rho_{real})))\| = \mathbf{E}_{\rho_{real}} \|v_1 - v_2\|, \tag{3}$$

where the norm $\|x\|$ could be the L_p-norm of any complex vector x with $\|x\|_p \equiv (|\Re(x)|^p + |\Im(x)|^p)^{1/p}$. Here two-norm is chosen for equation (3). As the two inputs to the encoders $\mathcal{E}_r$ and $\mathcal{E}_g$ are slightly different, the two encoders would not share the same weight parameters after training [36].

Combining the encoder $\mathcal{E}_r$ with $\mathcal{G}$ yields an encoder–decoder structure which aims to produce fake samples that are close to real ones. Thus we introduce the loss function

$$\mathcal{L}_2 = \mathbf{E}_{\rho_{real}} \|\rho_{real} - \mathcal{G}(\mathcal{E}_r(\rho_{real}))\| = \mathbf{E}_{\rho_{real}} \|\rho_{real} - \rho_{gen}\| \tag{4}$$

to quantify its performance. In analogy to classical autoencoders [37], it is found that L_1-norm achieves better performance than that of $p = 2$ for this loss term.

An optional discriminator network could be introduced for additional adversarial training. The discriminator $\mathcal{D}$ and generator $\mathcal{G}$ form the GAN (figure 3) which could enhance the ability of $\mathcal{G}$ to produce more realistic quantum samples. Indeed, $\mathcal{D}$ is a binary classifier trained to discriminate fake samples from

real ones. The two cost functions for this adversarial net are given by

$$\mathcal{L}_{\text{adv1}} = \mathbf{E}_{\rho_{\text{real}}}(-\mathcal{D}(\rho_{\text{real}}) + \mathcal{D}(\mathcal{G}(\mathcal{E}_{\text{r}}(\rho_{\text{real}})))), \tag{5}$$

$$\mathcal{L}_{\text{adv2}} = \mathbf{E}_{\rho_{\text{real}}}(-\mathcal{D}(\mathcal{G}(\mathcal{E}_{\text{r}}(\rho_{\text{real}})))), \tag{6}$$

which are alternatively minimized via gradient descent method. Specifically, the gradients are clipped between -1 and 1, turning the network into a Wasserstein GAN which is easy to train [32]. In each round, the parameters of $\mathcal{D}$ are updated by minimizing $\mathcal{L}_{\text{adv1}}$, while the parameters of $\mathcal{G}$ and $\mathcal{E}_{\text{r}}$ are updated by minimizing $\mathcal{L}_{\text{adv2}}$.

Finally, by combining (3)–(6), the complex-valued neural network is trained by alternatively minimizing $\mathcal{L}_{\text{adv1}}$ and

$$\mathcal{L}_3 = w_1 \cdot \mathcal{L}_1 + w_2 \cdot \mathcal{L}_2 + w_a \cdot \mathcal{L}_{\text{adv2}}, \tag{7}$$

with the weight parameters w_1, w_2, and w_a being chosen adaptively.

2.4. Training the networks via unsupervised learning

Suppose the complex-valued network is trained with separable states only, an entangled state would result in a feature vector $\boldsymbol{v}_{\text{ent}}$ distinct from that of the generated one in the latent space. Indeed, the entire training and prediction process can be divided into three steps as follows.

(a) Preparing separable states as training samples. Following equation (1), each ρ_i^j is generated via $HH^\dagger/(\text{tr}\, HH^\dagger)$, where H is a complex-valued matrix whose real and imaginary parts of each entry are sampled from independent Gaussian distributions. It is noted that this sampling method could cover the whole space of separable states [38].

(b) Training the neural network on the generated set of separable states by alternatively minimizing $\mathcal{L}_{\text{adv1}}$ as per equation (5) and $\mathcal{L}_3$ as per equation (7) via the gradient descent method.

(c) Determining the decision threshold value b on the test set after training. We choose b to satisfy

$$\frac{\text{FN}}{\text{TP}+\text{FN}} = \frac{\text{FP}}{\text{FP}+\text{TN}}, \tag{8}$$

where TP, FP, TN, and FN refer to the number counts of true positive, false positive, true negative, and false negative samples. Here, being positive or negative stands for a separable or entangled sample. Choosing b to satisfy equation (8) implies that the probabilities of misclassifying entangled and separable states are the same on the test set. Hence, if the score of a quantum state is larger than this b, then it will be detected as entangled.

For each ρ in the test set, its score for entanglement detection can be defined as

$$\mathcal{A}(\rho) = \left\|\mathcal{E}_{\text{r}}(\rho) - \mathcal{E}_{\text{g}}(\mathcal{G}(\mathcal{E}_{\text{r}}(\rho)))\right\|_2. \tag{9}$$

It could be further expressed in a witness-like form of

$$\mathcal{A}(\rho) = \|(\mathcal{W}_{\mathcal{E}_{\text{g}}}\mathcal{W}_{\mathcal{G}} - \mathcal{I})\mathcal{W}_{\mathcal{E}_{\text{r}}} \cdot \text{vec}(\rho)\|_2 = \|\mathcal{W} \cdot \text{vec}(\rho)\|_2, \tag{10}$$

where $\mathcal{W}_{\mathcal{E}_{\text{f}}(\mathcal{G})}$ denotes the weight tensor which generates the corresponding linear and nonlinear network transformations. For this reason, the neural network model can be regarded as trying to determine the nonlinear witness $\mathcal{W}$ which approximately characterizes the boundary between separable and entangled states, without relying on samples of entangled states during training.

Alternately, there is another way to implement the model for prediction without the test dataset, making both training and prediction independent of any information of entangled states. This is achieved by determining b as

$$b = \max_{\rho_{\text{sep}}} \mathcal{A}(\rho). \tag{11}$$

Obviously, this approach leads to a higher detection accuracy than using equation (8). Since both the training and implementation do not rely on entangled samples, this approach is computationally efficient. More importantly, the major advantage of our unsupervised learning framework lies in its scalability, as generating sufficient entangled states for training becomes impractical for high-dimensional quantum systems.

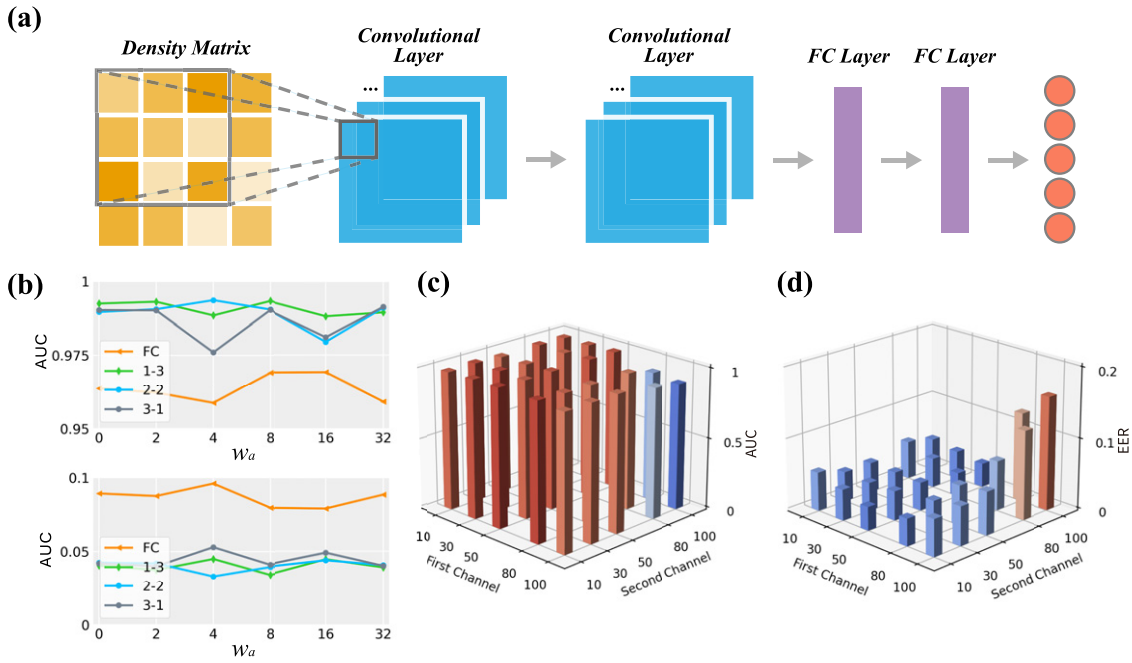

Figure 4. Detection of two-qubit states. (a) Four-layer neural network structure of the encoder, including two convolutional layers followed by two FC layers. The structures of the encoder and decoder are symmetric. The structure of the discriminator is the same as the encoder, with an additional normalization layer to produce a scalar output. (b) The performance of the neural networks with the first two layers being convolutional or FC. The convolutional kernel size combinations that have been tested are 1×1 and 3×3, 2×2 and 2×2, 3×3 and 1×1, with the number of output channels being 10 and 30, respectively. If the first two layers are FC, the number of output channels is set to 32 and 128, respectively. (e) and (f) AUCs and EERs with different number of output channels for convolutional layers. The convolutional kernel sizes are 2×2 and 2×2.

3. Numerical results

3.1. Evaluation metrics

We use two evaluation metrics of binary classification in our experiments. The first metric is the area under curve (AUC) of the receiver operating characteristic curve, which is created by plotting the true positive rate (TPR $=$ TP/(TP $+$ FN)) against the false positive rate (FPR $=$ FP/(FP $+$ TN)) using the similarity score defined in (9) for various values of b [39]. The second metric is equal error rate (EER), which is defined as FN/(TP $+$ FN) when equation (8) holds [40].

3.2. Detecting two-qubit entangled states

The number of training samples for two-qubit case is 160 000, all composed of separable states. The number of testing samples is 80 000, including 40 000 separable states and 40 000 entangled states. Two-qubit separable states are generated by

$$\rho_{\text{sep}} = \sum_{i=1}^{m} \lambda_i \rho_i^1 \otimes \rho_i^2, \tag{12}$$

where $\sum_{i=1}^{m} \lambda_i = 1$ and $0 \leqslant \lambda_i \leqslant 1$, with m iterating from 1 to 20. Entangled states are selected from randomly generated states of the entire system using PPT criterion.

The structure of the four-layer encoder is illustrated in figure 4(a). The last two layers of the encoder are fully-connected (FC) layers, with output channels being 64 and 10, respectively. The first two layers can be convolutional with different kernels and different number of output channels, or fully connected as tested in figure 4(b). The best performance of the model has been achieved with the convolutional kernel size of the first two layers being 2×2 and 2×2. The best AUC is 0.99 and EER is 2.99%, attained at a small w_a which is the weight of adversarial cost for training. As shown in figure 4(c), convolutional layer performs much better than FC layer, with AUC being consistently higher than 0.975 and EER lower than 5%. Figures 4(e) and (f) shows the performance of convolutional neural networks when the number of output channels varies, indicating that a small number of output channels is enough to extract the features of entanglement for two-qubit states.

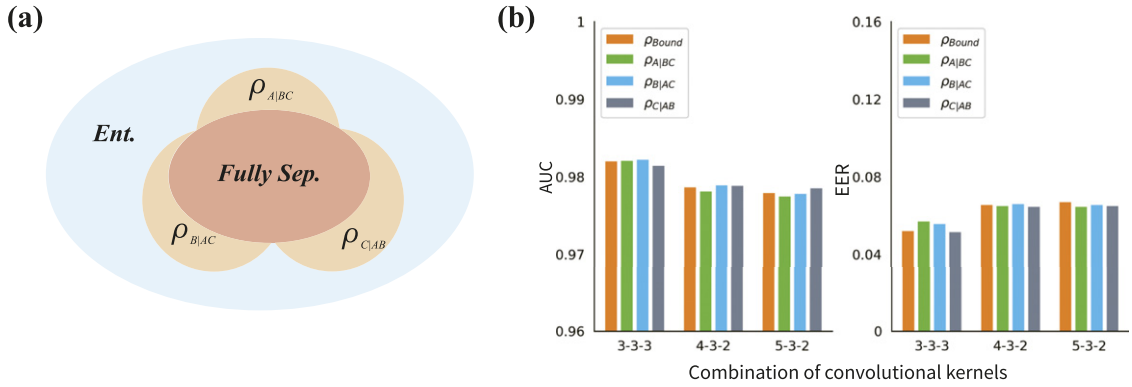

Figure 5. Detection of three-qubit states. (a) Distribution of three-qubit states. $\rho_{A|BC}$ is a bi-separable state with qubit B and C entangled. (b) EERs and AUCs with different combinations of convolutional kernels, where i–j–k stands for the kernel sizes of the three convolutional layers.

3.3. Detecting three-qubit entangled states

An entangled three-qubit state can be classified into several types, e.g. bi-separable states and bound entangled states [24]. The three-qubit state is fully-separable if

$$\rho_{\text{sep}} = \sum_{i=1}^{m} \lambda_i \rho_i^{\text{A}} \otimes \rho_i^{\text{B}} \otimes \rho_i^{\text{C}}. \tag{13}$$

The distribution of three-qubit states is illustrated in figure 5(a). In this case, successful supervised learning requires that one can generate enough and balanced samples for all types of entanglement, which cannot be guaranteed by the current random sampling techniques. In contrast, a universal entanglement detector could be built using only the fully-separable samples if unsupervised learning method is employed.

The numerical results in figure 5(b) are based on a dataset consisting of 160 000 training samples and 200 000 test samples. The training samples are fully-separable states, and the test samples include 40 000 fully-separable states, 40 000 bound entangled states and 120 000 bi-separable states (40 000 for each subtype). To accommodate the 8×8 density matrix input, a third convolutional layer is added. The number of the output channels for the three convolution layers is 10, 30, 50, respectively. Since the unsupervised model focuses on detecting the feature of separability instead of the features of different types of entanglement, it has achieved similar detection accuracy on four types of entangled samples.

The proposed unsupervised learning method is applicable to the detection of partial entanglement and genuine entanglement. Here we take the detection of bi-separable states of a three-qubit system as an example [41]. Suppose the task is to discriminate the bi-separable states $\rho_{\text{A|BC}}$ (B and C are entangled) from the other states. By generating the entangled states for subsystem BC using the PPT criterion, the samples of bi-separable states are given by

$$\rho_{\text{A|BC}} = \sum_{i=1}^{m} \lambda_i \rho_i^{\text{A}} \otimes \rho_i^{\text{BC}}. \tag{14}$$

A classifier for A|BC separability can be obtained by training on these samples in an unsupervised manner. Particularly, if we replace ρ_i^{BC} in (14) by a generic two-qubit state, the anomalies detected would be the quantum states that are entangled between A and BC (page 10). Furthermore, if we generate the samples as

$$\rho_{\text{ABC}} = \sum_{i=1} \lambda_i^1 \rho_i^{\text{A}} \otimes \rho_i^{\text{BC}} + \sum_{j=1} \lambda_j^2 \rho_j^{\text{B}} \otimes \rho_j^{\text{AC}} + \sum_{k=1} \lambda_k^3 \rho_k^{\text{C}} \otimes \rho_k^{\text{AB}}, \tag{15}$$

the abnormal samples detected by the unsupervised model would be quantum states which are not bi-separable. In other words, the genuine entanglement of the three-qubit state can be detected as an anomaly.

3.4. Scalability up to ten-qubit states

The unsupervised learning method is applied on four- to ten-qubit states to study its scalability. We have found that the generation of separable states for training is very efficient even for tens of qubits, because the generation of separable pure states is very efficient, which is done by generating single qubit states and calculating their Kronecker products. Consequently, mixed (fully and partial) separable states can be constructed as linear combinations of pure states, which does not take much time. In this work, it takes less

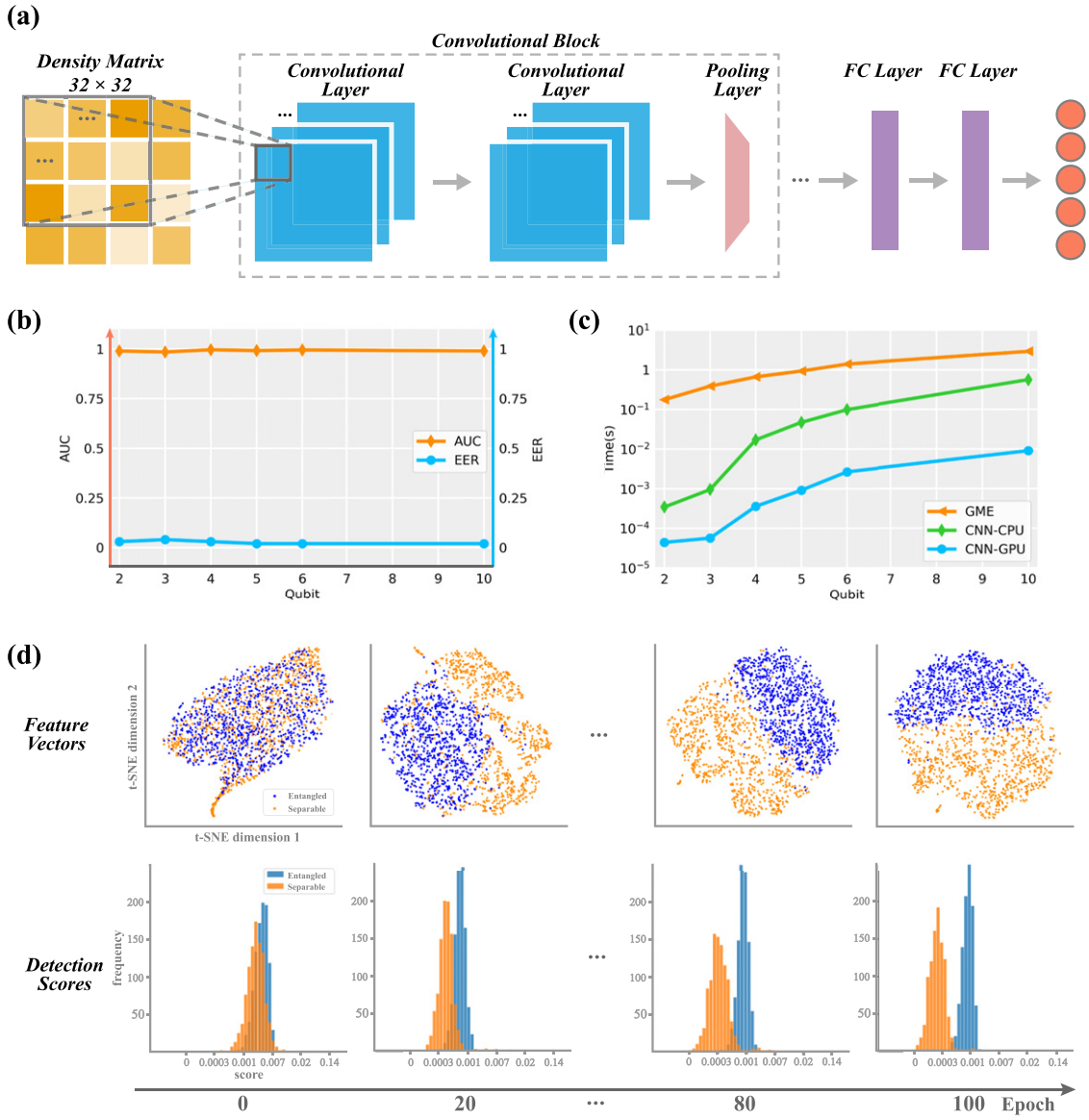

Figure 6. Scalability of unsupervised learning. (a) The neural network structure of the encoder for five-qubit pure states. The convolutional block is composed of two convolutional layers and a 2 × 2 max pooling layer. (b) AUCs and EERs achieved by the unsupervised learning method as the number of qubits increases. (c) The comparison of inference time between the up-to-date numerical method for computing the GME and the neural network method. (d) The evolution of feature vectors and detection scores for 1000 separable and 1000 entangled samples of five-qubit states during training.

than 10 min to generate enough pure separable samples for ten-qubit states on a desktop computer, and mixing the samples takes less than 3 min. Moreover, we have observed a linear increase on the generation time with the dimension. Here we used pure four- to ten-qubit states for training because the test samples of mixed states (mixed entangled states) are hard to label for high-dimensional system, while we have developed an efficient algorithm [42] that can tell whether a randomly generated ten-qubit pure state is entangled or not within 5 s. However, test samples are just used to measure the accuracy of the model. The model is trained using the separable samples only, which can be generated efficiently. The trained model can be implemented without using test samples as shown in (11). Therefore, the model can also be trained and implemented with mixed state samples for high-dimensional cases. Note that the geometrical measure is only used to label the entangled states for the test dataset. The separable pure states are generated by

$$|\psi_{\text{sep}}\rangle = |\psi_i^1\rangle \otimes \cdots |\psi_i^j\rangle \cdots \otimes |\psi_i^n\rangle, \tag{16}$$

where $|\psi_i^j\rangle$ is a randomly generated pure state vector of the jth qubit. The real and imaginary parts of the complex-valued vector are sampled from an independent Gaussian distribution. The density matrix $\rho_{\text{sep}} = |\psi_{\text{sep}}\rangle\langle\psi_{\text{sep}}|$ is used as the input to the neural network. Figure 6(a) depicts the network structure of the encoder for entanglement detection in five-qubit states, where a max pooling layer has been added to

handle the increased dimension of the input. For ten-qubit states, we adopt three convolutional layers and increase the max pooling size to 4×4. The training dataset is composed of 160 000 separable states, and the test dataset is composed of 40 000 separable and 40 000 entangled states. The entangled states are found by randomly generating four- to ten-qubit pure states and computing their entanglement measures using the numerical method from [42]. See appendix B for the details of the algorithm.

As shown in figure 6(b), the unsupervised model achieves an AUC of 0.9952 and an EER of 2.02% for entanglement detection in ten-qubit states. The EER is 0.54% for entanglement detection in five-qubit states, which means only 54 in 10 000 states are misclassified. The short inference time is another advantage of the neural network model. The inference time of the neural network model on GPU is about tens of microseconds to hundreds of microseconds for up to 10 qubits (figure 6(c)), which is significantly faster than the up-to-date numerical method which takes the state vector instead of density matrix as the input for computing the geometrical entanglement measure (GME). The time needed for generating training dataset is greatly reduced as compared to supervised learning methods, since there is no need to label the entangled states. For example, suppose the ten-qubit training dataset of the supervised method consists of 100 000 samples, which must be labelled by numerically computing the GME. The total time needed for generating the dataset is about 138 h (labelling each sample takes 5 s in average). In contrast, generating separable training samples of the ten-qubit system is much more simple, which only takes several minutes.

The upper half of figure 6(d) shows the evolution of feature vectors of 1000 separable and 1000 entangled states in the training process for five-qubit states. We visualize the evolution by t-SNE method [43] which maps the feature vectors to two-dimensional space. In the first 10 epochs, the entangled and separable states are mixed up in the latent space and difficult to distinguish. After 20 epochs, the feature vectors start to split into two set. In the last 20 epochs, the feature vectors of separable states are separated completely from the feature vectors of entangled states, with very few exceptions. A similar evolution can be seen in the distribution of detection scores of the input states. After training, the detection scores of separable states are more closed to zero, while the scores of entangled states are concentrated around 0.001.

4. Conclusions and discussions

We have proposed an efficient and scalable method with unsupervised learning to detect quantum entanglement. Specifically, we build up a class of complex-valued pseudo-siamese neural networks which is easy to implement as it is trained without entangled samples. Moreover, it is scalable to detect entanglement of multipartite systems where sufficient labelled entangled samples become difficult to obtain, and our numerical analysis finds that we could still obtain a rather high accuracy with above 97.5% on average for multipartite systems from two-qubit to ten-qubit. For this reason, we believe that our work provides a promising tool to detect quantum features of high-dimensional quantum data.

Finally, it is noted that we exploit the convexity of separable samples and thus reformulate entanglement detection as an anomaly detection problem, for which the unsupervised neural networks are suitable. Since other useful quantum features, such as Bell nonlocality and Einstein–Podolsky–Rosen steerability, also share the same property that it is defined as a distinguishable sample from a convex set, it is evident that our work can be readily generalized to solve the similar detection problem.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants Nos. 62173296 and 62088101. G F also acknowledges support from Hong Kong Research Grant Council (Grants Nos. 1520841, 15203619, and 15506619), Shenzhen Fundamental Research Fund, China, under Grant No. JCYJ20190813165207290, and the CAS AMSS-polyU Joint Laboratory of Applied Mathematics.

Data availability statement

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

Appendix A. Complex-valued neural network

We build the complex-valued neural network based on the work of [44]. The codes are available at https://github.com/ewellchen/Entanglement_detection. The two-dimensional convolutional (denoted as *)

and FC (denoted as $\cdot$) operations of the weight w and input z in the complex domain are defined by

$$w * z = \Re\{w\} * \Re\{z\} - \Im\{w\} * \Im\{z\} \tag{A.1}$$

$$+ \mathrm{i}(\Im\{w\} * \Re\{z\} + \Re\{w\} * \Im\{z\}), \tag{A.2}$$

$$w \cdot z = \Re\{w\}\Re\{z\} - \Im\{w\}\ \Im\{z\} \tag{A.3}$$

$$+ \mathrm{i}(\Re\{w\}\Im\{z\} + \Im\{w\}\ \Re\{z\}), \tag{A.4}$$

where $\Re$ and $\Im$ represent the real and imaginary part of the vector or matrix, respectively. The formulation of the complex-valued rectified linear unit (CReLU) is given by

$$\mathrm{CReLU}(z) = \mathrm{ReLU}(\Re(z)) + \mathrm{iReLU}(\Im(z)), \tag{A.5}$$

which introduces nonlinearity into the network transformation. The batch normalization (BN) layer is implemented by multiplying the 0-centered data $(\boldsymbol{z} - \mathbf{E}[\boldsymbol{z}])$ with the inverse square root of the covariance as

$$\mathcal{V} = \begin{pmatrix} \mathrm{Cov}(\Re\{\boldsymbol{z}\}, \Re\{\boldsymbol{z}\}) & \mathrm{Cov}(\Re\{\boldsymbol{z}\}, \Im\{\boldsymbol{z}\}) \\ \mathrm{Cov}(\Im\{\boldsymbol{z}\}, \Re\{\boldsymbol{z}\}) & \mathrm{Cov}(\Im\{\boldsymbol{z}\}, \Im\{\boldsymbol{z}\}) \end{pmatrix},$$

$$\tilde{\boldsymbol{z}} = (\mathcal{V})^{-\frac{1}{2}}(\boldsymbol{z} - \mathbf{E}[\boldsymbol{z}]),$$

$$\mathrm{BN}(\tilde{\boldsymbol{z}}) = \begin{pmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ri} & \gamma_{ii} \end{pmatrix} \tilde{\boldsymbol{z}} + \boldsymbol{\beta}. \tag{A.6}$$

The parameters $\gamma_{r(i)r(i)}$ and β are trainable. Each convolutional layer is composed of a convolutional operation, a CReLU and a BN layer. The first FC layer is composed of a FC operation and a CReLU. The last FC layer generates the final output directly via a FC operation. The operations defined above are differentiable, which means the neural network could be trained efficiently with back-propagation. The gradient is calculated with respect to the real-valued cost function $\mathcal{L}$ as

$$\nabla_{\mathcal{L}}(z) = \frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial z_r} + \mathrm{i}\frac{\partial \mathcal{L}}{\partial z_i} = \Re(\nabla_{\mathcal{L}}(z)) + \mathrm{i}\Im(\nabla_{\mathcal{L}}(z)). \tag{A.7}$$

The back-propagation updates the complex-valued parameter $t = t_r + \mathrm{i}t_i$ of the neural network by

$$\nabla_{\mathcal{L}}(t) = \frac{\partial \mathcal{L}}{\partial t} = \frac{\partial \mathcal{L}}{\partial t_r} + \mathrm{i}\frac{\partial \mathcal{L}}{\partial t_i} \tag{A.8}$$

$$= \frac{\partial \mathcal{L}}{\partial z_r}\frac{\partial z_r}{\partial t_r} + \frac{\partial \mathcal{L}}{\partial z_i}\frac{\partial z_i}{\partial t_r} + \mathrm{i}\left(\frac{\partial \mathcal{L}}{\partial z_r}\frac{\partial z_r}{\partial t_i} + \frac{\partial \mathcal{L}}{\partial z_i}\frac{\partial z_i}{\partial t_i}\right) \tag{A.9}$$

$$= \frac{\partial \mathcal{L}}{\partial z_r}\left(\frac{\partial z_r}{\partial t_r} + \mathrm{i}\frac{\partial z_r}{\partial t_i}\right) + \frac{\partial \mathcal{L}}{\partial z_i}\left(\frac{\partial z_i}{\partial t_r} + \mathrm{i}\frac{\partial z_i}{\partial t_i}\right) \tag{A.10}$$

$$= \Re(\nabla_{\mathcal{L}}(z))\left(\frac{\partial z_r}{\partial t_r} + \mathrm{i}\frac{\partial z_r}{\partial t_i}\right) \tag{A.11}$$

$$+ \Im(\nabla_{\mathcal{L}}(z))\left(\frac{\partial z_i}{\partial t_r} + \mathrm{i}\frac{\partial z_i}{\partial t_i}\right), \tag{A.12}$$

which could be implemented using Pytorch [45].

Appendix B. Computing the GME of quantum pure states

We employ the algorithm proposed in [42] to compute the GME for an arbitrary quantum pure state. The algorithm is based on a tensor version of the Gauss–Seidel method for computing unitary eigenpairs (U-eigenpairs) of a non-symmetric complex tensor $\mathcal{A}$ which corresponds to the given quantum pure state.

Algorithm 1 [42]. Computing the U-eigenpairs of an $n_1 \times \cdots \times n_m$ non-symmetric complex tensor $\mathcal{A}$.

Step 1 (initial step): let $\mathcal{S} = \mathbf{sym}(\mathcal{A})$ be the symmetric embedding of $\mathcal{A}$, and $n = n_1 + \cdots + n_m$. Choose a starting point $\mathbf{x}_0 \in \mathbf{C}^n$ with $\|\mathbf{x}_0\| = 1$, and $0 < \alpha_{\mathcal{S}} \in \mathbf{R}$. Let $\lambda_0 = \mathcal{S}^*\mathbf{x}_0^m$.

Step 2 (iterating step):

for $k = 1, 2, \ldots,$ **do**

$$\hat{\mathbf{x}}_k = \lambda_{k-1}\mathcal{S}\mathbf{x}_{k-1}^{*m-1} + \alpha_{\mathcal{S}}\mathbf{x}_{k-1}, \tag{B.1}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k/\|\hat{\mathbf{x}}_k\|, \tag{B.2}$$

$$\lambda_k = \mathcal{S}^*\mathbf{x}_k^m. \tag{B.3}$$

end for.
return:
Unitary symmetric eigenpair (US-pair): $\lambda_{\mathcal{S}} = |\lambda_k|$, and $\mathbf{x} = (\frac{\lambda_{\mathcal{S}}}{\lambda_k})^{1/m}\mathbf{x}_k$.
Let $\mathbf{x} = (\mathbf{x}^{(1)\top}, \ldots, \mathbf{x}^{(m)\top})^\top$, $\mathbf{x}^{(i)} \in \mathbf{C}^{n_i}$, for all $i = 1 : m$.
U-eigenvalue $\lambda_{\mathcal{A}} = \frac{(\sqrt{m})^m}{m!}\lambda_{\mathcal{S}}$.
U-eigenvector $\{\sqrt{m}\mathbf{x}^{(1)}, \ldots, \sqrt{m}\mathbf{x}^{(m)}\}$.

ORCID iDs

Yu Pan https://orcid.org/0000-0001-6900-4016

References

[1] Horodecki R, Horodecki P, Horodecki M and Horodecki K 2009 Quantum entanglement *Rev. Mod. Phys.* **81** 865
[2] Streltsov A, Adesso G and Plenio M B 2017 Colloquium: quantum coherence as a resource *Rev. Mod. Phys.* **89** 041003
[3] Chitambar E and Gour G 2019 Quantum resource theories *Rev. Mod. Phys.* **91** 025001
[4] Nielson M A and Chuang I I 2000 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)
[5] Deutsch I H 2020 Harnessing the power of the second quantum revolution *PRX Quantum* **1** 020101
[6] Peres A 1996 Separability criterion for density matrices *Phys. Rev. Lett.* **77** 1413
[7] Gurvits L 2003 Classical deterministic complexity of Edmonds' problem and quantum entanglement *Proc. of the 35th Annual ACM Symp. on Theory of Computing* pp 10–9
[8] Horodecki R, Horodecki M and Horodecki P 1996 Teleportation, Bell's inequalities and inseparability *Phys. Lett.* A **222** 21–5
[9] Terhal B M 2000 Bell inequalities and the separability criterion *Phys. Lett.* A **271** 319–26
[10] Gühne O and Tóth G 2009 Entanglement detection *Phys. Rep.* **474** 1–75
[11] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
[12] Xiao J, Yan Y, Zhang J and Tang Y 2010 A quantum-inspired genetic algorithm for *k*-means clustering *Expert Syst. Appl.* **37** 4966–73
[13] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3
[14] Tang E 2019 A quantum-inspired classical algorithm for recommendation systems *Proc. of the 51st Annual ACM SIGACT Symp. on Theory of Computing* pp 217–28
[15] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev.* X **8** 031086
[16] Chapman R J, Ferrie C and Peruzzo A 2016 Experimental demonstration of self-guided quantum tomography *Phys. Rev. Lett.* **117** 040402
[17] Magesan E, Gambetta J M, Córcoles A D and Chow J M 2015 Machine learning for discriminating quantum measurement trajectories and improving readout *Phys. Rev. Lett.* **114** 200501
[18] Hentschel A and Sanders B C 2010 Machine learning for precise quantum measurement *Phys. Rev. Lett.* **104** 063603
[19] Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–6
[20] Huang L and Wang L 2017 Accelerated Monte Carlo simulations with restricted Boltzmann machines *Phys. Rev.* B **95** 035105
[21] Carrasquilla J and Melko R G 2017 Machine learning phases of matter *Nat. Phys.* **13** 431–4
[22] Lu S *et al* 2018 Separability-entanglement classifier via machine learning *Phys. Rev.* A **98** 012315
[23] Yang M *et al* 2019 Experimental simultaneous learning of multiple nonclassical correlations *Phys. Rev. Lett.* **123** 190401
[24] Ma Y-C and Yung M-H 2018 Transforming Bell's inequalities into state classifiers with machine learning *npj Quantum Inf.* **4** 34
[25] Liu N and Rebentrost P 2018 Quantum machine learning for quantum anomaly detection *Phys. Rev.* A **97** 042315
[26] Liang J-M, Shen S-Q, Li M and Li L 2019 Quantum anomaly detection with density estimation and multivariate Gaussian distribution *Phys. Rev.* A **99** 052310
[27] Bennett C H, Brassard G, Crépeau C, Jozsa R, Peres A and Wootters W K 1993 Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels *Phys. Rev. Lett.* **70** 1895
[28] Werner E E 1989 High-risk children in young adulthood: a longitudinal study from birth to 32 years *Am. J. Orthopsychiatry* **59** 72–81
[29] Dür W, Briegel H-J, Cirac J I and Zoller P 1999 Quantum repeaters based on entanglement purification *Phys. Rev.* A **59** 169
[30] Baldi P 2012 Autoencoders, unsupervised learning, and deep architectures *Proc. of ICML Workshop on Unsupervised and Transfer Learning* pp 37–49
[31] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* pp 2672–80
[32] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks *Int. Conf. on Machine Learning* pp 214–23
[33] Chicco D 2021 Siamese neural networks: an overview *Artificial Neural Networks* (Springer) pp 73–94
[34] Koch G, Zemel R and Salakhutdinov R 2015 Siamese neural networks for one-shot image recognition *ICML Deep Learning Workshop* vol 2 (Lille)
[35] Tax D M J and Duin R P W 2004 Support vector data description *Mach. Learn.* **54** 45–66
[36] Hughes L H, Schmitt M, Mou L, Wang Y and Zhu X X 2018 Identifying corresponding patches in SAR and optical images with a pseudo-siamese CNN *IEEE Geosci. Remote Sens. Lett.* **15** 784–8

[37] Isola P, Zhu J-Y, Zhou T and Efros A A 2017 Image-to-image translation with conditional adversarial networks *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 1125–34
[38] Zyczkowski K and Sommers H-J 2001 Induced measures in the space of mixed quantum states *J. Phys. A: Math. Gen.* **34** 7111
[39] Brown C D and Davis H T 2006 Receiver operating characteristics curves and related decision measures: a tutorial *Chemometr. Intell. Lab. Syst.* **80** 24–38
[40] Zhou X-H, McClish D K and Obuchowski N A 2009 *Statistical Methods in Diagnostic Medicine* vol 569 (New York: Wiley)
[41] Acin A, Bruß D, Lewenstein M and Sanpera A 2001 Classification of mixed three-qubit states *Phys. Rev. Lett.* **87** 040401
[42] Zhang M, Ni G and Zhang G 2020 Iterative methods for computing U-eigenvalues of non-symmetric complex tensors with application in quantum entanglement *Comput. Optim. Appl.* **75** 779–98
[43] van der Maaten L and Hinton G 2008 Visualizing data using t-SNE *J. Mach. Learn. Res.* **9** 2579–605
[44] Trabelsi C *et al* 2018 Deep complex networks *Int. Conf. on Learning Representations*
[45] Paszke A *et al* 2017 Automatic differentiation in Pytorch *NIPS 2017 Workshop on Autodiff* (Long Beach, California, USA) https://openreview.net/forum?id=BJJsrmfCZ