applied sciences

IMPACT FACTOR 2.7

CITESCORE 4.5

Article

Improving Steerability Detection via an Aggregate Class Distribution Neural Network

Yuyang Hao, Kan He and Ying Zhang

MDPI

https://doi.org/10.3390/app13137874

applied sciences

MDPI

Article

Improving Steerability Detection via an Aggregate Class Distribution Neural Network

Yuyang Hao [1], Kan He [2,*] and Ying Zhang [3]

[1] College of Software, Taiyuan University of Technology, Taiyuan 030024, China; hyy123013@163.com
[2] College of Mathematics, Taiyuan University of Technology, Taiyuan 030024, China
[3] College of Information and Computer Science, Taiyuan University of Technology, Taiyuan 030024, China; zhangying-1226@163.com
* Correspondence: hekanquantum@163.com

Abstract: In this paper, we establish an aggregate class distribution neural network (AGGNN) structure to determine whether an arbitrary two-qubit quantum state is steerable. Compared to the classification results obtained using a support vector machine (SVM) and a backpropagation neural network (BPNN), we obtain higher-accuracy quantum-steering classification models via the AGGNN, as well as steerability bounds of generalized Werner states, which are more similar to the theoretical bounds. In particular, when we only know partial information about the quantum states, higher-performance quantum-steering classifiers are obtained compared to those via SVM and BPNN.

Keywords: quantum steering; aggregate class distribution neural network; generalized Werner states; steerability bounds

check for updates

Citation: Hao, Y.; He, K.; Zhang, Y. Improving Steerability Detection via an Aggregate Class Distribution Neural Network. *Appl. Sci.* **2023**, *13*, 7874. https://doi.org/10.3390/app13137874

Academic Editor: Augusto Ferrante

Received: 25 May 2023
Revised: 24 June 2023
Accepted: 24 June 2023
Published: 4 July 2023

1. Introduction

Schrödinger presented the concept of quantum steering [1] based on the well-known EPR paradox [2] in 1935. Although the EPR argument led to long-lasting discussions, there was not much interest in quantum steering. It was not until 2007 when Wiseman, Jones, and Dougherty presented a precise definition and systematic criteria [3] that people began to attach importance to steering. Steering is a quantum correlation between entanglement [4,5] and non-locality [6], wherein one party can steer the state of another distant party by choosing appropriate local measurements. Currently, steering plays an essential role in diverse quantum information-processing tasks [3,7,8] such as one-sided device-independent quantum key distribution [9–14], channel discrimination [15,16], randomness certification [17–19], and tele-amplification [20].

Given the importance of steering in quantum information-processing tasks, it is necessary to detect the steerability of a given bipartite quantum state [3,7,8,15,21–34]. Although various criteria and inequalities for steering have been presented [21–29,32], it is sometimes difficult to determine whether an arbitrary unknown quantum state is steerable, even in the most simple case of a two-qubit state. Theoretically, these criteria can be estimated through semidefinite programming (SDP) [35], but it is very time-consuming and accompanied by misjudgment because of having to consider a wide range of measurement directions in the real computing process [36].

Machine learning can effectively enhance the accuracy and speed of prediction by learning from known data and experience, which has already been adopted for quantum information-processing tasks such as entanglement classification [37], non-locality discrimination [38,39], phase-transition identification [40,41], quantum state tomography [42], and Markovianity [43]. In addition, several machine learning methods, such as the artificial neural network, support vector machine (SVM), decision tree, and semi-supervised machine learning, have been used to detect and quantify the steerability of a given state [44–48].

Among these methods, the SVM [45] and backpropagation neural network (BPNN) [48] are both supervised machine learning methods used to detect steerability. Compared to the SVM [45], BPNN [48] has better learning ability, simpler parameter tuning, and better performance in processing large amounts of data and continuous data features. Therefore, Zhang, He, et al. used this method to construct several models to obtain high-performance quantum-steering classifiers. Nevertheless, the BPNN still has the problem of overfitting inaccurate data.

The AggMatch algorithm improves the performance of semi-supervised learning by integrating the prediction results of similar samples [49]. Inspired by this method, we present a new aggregate class distribution neural network (AGGNN) structure to overcome the problem of overfitting inaccurate quantum states. The AGGNN creates samples of similar features that have a similar prediction distribution based on the similarity measure in order to improve the generalization ability of networks. Thus, for both the whole information and partial information of a quantum state, AGGNN can construct higher-performance quantum-steering classification models compared to the BPNN and SVM models. In addition, we use the AGGNN classification models to predict the steerability bounds of the generalized Werner states. The steerability bounds of the generalized Werner states predicted by the AGGNN are more similar to the theoretical bounds compared to the SVM [45] and BPNN [48] models.

This paper is organized as follows. In Section 2, we introduce the quantum-steering problem and the AGGNN classification model. In Section 3, we illustrate the data generation of different features, demonstrate the parameters and training details of the AGGNN classification model on different features, and compare the steerability classification accuracy and bounds predicted by the AGGNN classification models with those predicted by the SVM and BPNN models. Finally, we summarize our results in Section 4.

2. Preliminaries

2.1. Quantum Steering

In the modern view of quantum steering, it is a concept that is incompatible with the local-hidden-state (LHS) model [36]. For an unknown quantum state ρ shared by Alice and Bob, assume that Alice performs the quantum measurements $\{\mathcal{M}_x^A = \{M_{a|x}^A\}\}$ on her subsystems, and the corresponding measurement outcomes are denoted by a. Based on the quantum theory, after choosing the quantum measurement $M_{a|x}^A$ and obtaining the measurement outcome a, the non-normalized quantum-state assemblage of the subsystem ρ_B is $\{p(a|x), \rho_{a|x}\}$, where

$$\rho_{a|x} = \mathrm{tr}_A[(M_{a|x}^A \otimes I^B)\rho], \tag{1}$$

and the probability distribution

$$p(a|x) = \mathrm{tr}[(M_{a|x}^A \otimes I^B)\rho]. \tag{2}$$

Quantum steering describes a scenario: Bob can fully control their measurements and access the condition state $\rho_{a|x}$ without characterization of Alice's measurements. Namely, Bob performs tomography to reconstruct the set of conditional assemblages $\{\rho_{a|x}\}$ and the results are not dependent on any particular information about how Alice's measurements work. In other words, Alice can steer Bob's local state by performing local measurements and classical communication on the particle she owns. Utilizing the LHS model, quantum steering can be defined as the impossibility of remotely generating ensembles produced by an LHS model. That is, suppose that a source sends a classical message λ to Alice and a corresponding state σ_λ to Bob. If the measurement applied by Alice is x, the classical variable λ determines the probability of her obtaining the measurement outcome a, which is represented by $p(a|x, \lambda)$. Although the probability distribution of the classical message λ

is $p(\lambda)$, since Bob cannot access the classical variable λ, the finial corresponding assemblage that Bob observes is composed of the elements

$$\rho_{a|x} = \int \mathrm{d}\lambda p(\lambda) p(a|x,\lambda)\sigma_\lambda. \tag{3}$$

We say that Alice cannot steer Bob's state if the LHS model can generate a conditional assemblage $\rho_{a|x}$ that corresponds to the state ρ. Otherwise, we say that Alice can steer Bob's state. However, efficiently identifying the steerability of a quantum state is usually very difficult.

Vandenberghe and Boyd gave the numerical calculation for the quantum-steering criterion in terms of SDP [35]. Now let us introduce this computing process. For an arbitrary bipartite state ρ, if Alice performs m measurements $x = \{0, \ldots, m-1\}$ with k outcomes $a = \{0, \ldots, k-1\}$ each, and $\lambda^{'}$ is a map from the measurement set $\{0, \ldots, m-1\}$ to the outcome set $\{0, \ldots, k-1\}$, the deterministic probability distribution is defined as $D(a|x,\lambda^{'}) = \delta_{a,\lambda^{'}(x)}$. Then, Equation (3) can be rewritten as

$$\rho_{a|x} = \sum_{\lambda^{'}=1}^{d} D(a|x,\lambda^{'})\rho_{\lambda^{'}}, \tag{4}$$

where $p(a|x,\lambda) = \sum_{\lambda^{'}=1}^{d} p(\lambda^{'}|\lambda) D(a|x,\lambda^{'})$, and $\rho_{\lambda^{'}} \equiv \int d\lambda p(\lambda) p(\lambda^{'}|\lambda)\sigma_\lambda$.

We write the SDP that can identify the steerability of a state [35] as follows

$$\begin{aligned} &\text{given} \{\rho_{a|x}\}, \{D(a|x,\lambda^{'})\}_{\lambda^{'}} \\ &\min_{\{F_{a|x}\}} \operatorname{tr} \textstyle\sum_{ax} F_{a|x}\rho_{a|x} \\ &\text{s.t.} \textstyle\sum_{ax} F_{a|x} D(a|x,\lambda^{'}) \geqslant 0 \quad \forall \lambda^{'} \\ &\operatorname{tr} \textstyle\sum_{ax\lambda^{'}} F_{a|x} D(a|x,\lambda^{'}) = 1, \end{aligned} \tag{5}$$

where $\{F_{a|x}\}$ are Hermitian matrices. If the objective function value in SDP (5) is negative for some measurements x, then the quantum state ρ is steerable from Alice to Bob. Otherwise, ρ is unsteerable.

The generalized Werner state is identified as a simple family of one-way steerable two-qubit states [50]. It can be given by

$$\rho(\beta,\theta) = \beta|\psi_\theta\rangle\langle\psi_\theta| + (1-\beta)\rho^A \otimes \frac{I^B}{2}, \tag{6}$$

where $|\psi_\theta\rangle = \cos\theta|00\rangle + \sin\theta|11\rangle$, $\rho^A = \mathrm{tr}_B(|\psi_\theta\rangle\langle\psi_\theta|)$, $0 \leqslant \beta \leqslant 1$, $0 < \theta \leqslant \frac{\pi}{4}$. It has been proved that $\rho(\beta,\theta)$ is unsteerable from Alice to Bob when

$$\cos^2 2\theta \geqslant \frac{2\beta - 1}{(2-\beta)\beta^3}. \tag{7}$$

2.2. AGGNN Model

AGGNN aggregates the class distribution of samples by similarity measure during the network training process to enlarge the difference between different class sample features extracted by the model to improve the model's robustness and prediction ability. The network model is divided into three parts: feature extraction layer, classification layer, and aggregate class distribution layer. As shown in Figure 1.

Feature extraction layer. During the training process, we randomly select a batch of examples as $B = \left\{(x_i, y_i) : x_i \in \mathbb{R}^{1\times d}, y_i \in Y, i \in (1, \ldots, n)\right\}$, where x_i is a sample, y_i is a label of $Y = \{0, 1\}$, d is the data dimension of the sample, and n is an even number representing the batchsize. Half of the batches contains unsteerable samples where $y_{i\in\{1,\ldots,n/2\}} = 1$, while the other half contains steerable samples where $y_{i\in\{(n/2)+1,\ldots,n\}} = 0$.

We obtain the features for every sample in each batch of training data through the feature extraction layer, and this process can be expressed as follows

$$\mathbf{V} = \mathrm{FEMLP}(\mathbf{X}), \tag{8}$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^{n \times d}$ is the matrix composed of a batch of samples (the superscript T denotes transpose of matrices), FEMLP is a multilayer perceptron for feature extraction [51], and $\mathbf{V} \in \mathbb{R}^{n \times d_v}$ is the feature matrix of the samples (d_v is the dimension of the feature vector of a sample).

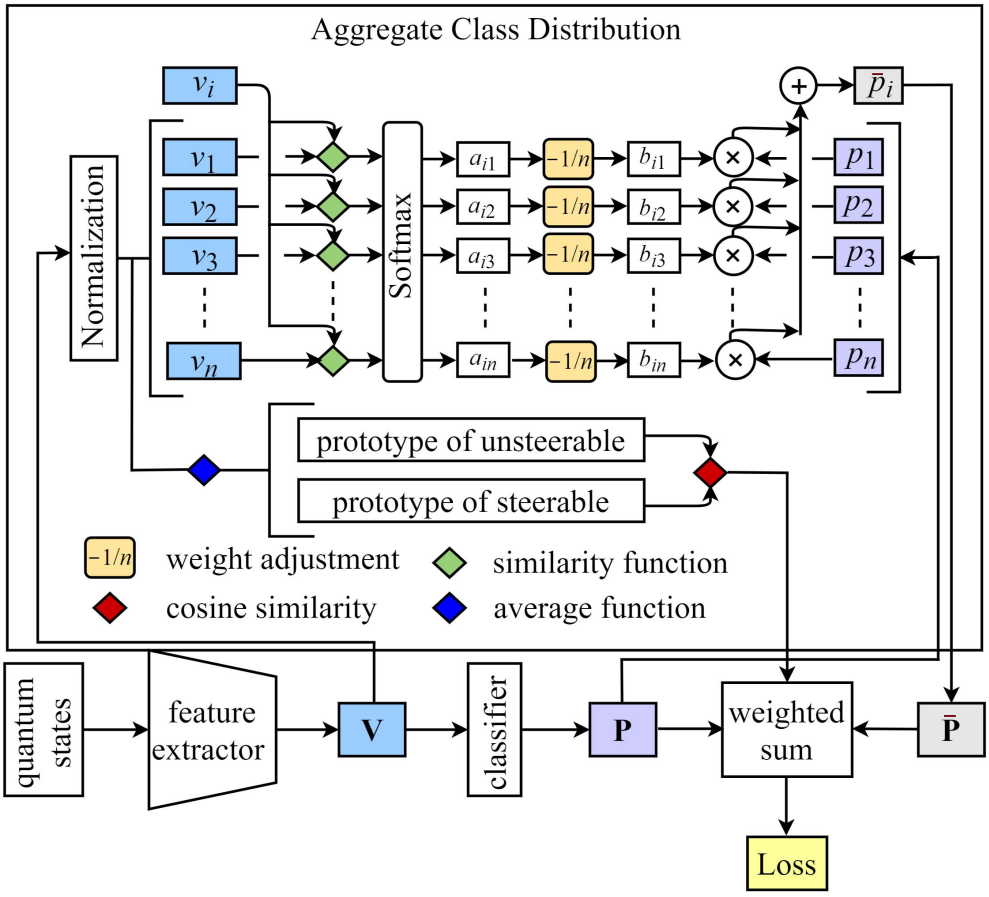

Figure 1. The structure diagram of the AGGNN model's training process. a_{ij} is the element of the ith row and jth column of the similarity matrix **A**, and b_{ij} is the element of the ith row and jth column of the positive and negative correlation similarity matrix **B**. Firstly, we obtain feature **V** and class distribution **P** of the quantum states through the feature extraction layer and classification layer, respectively. Then we input the feature and class distribution into aggregate the class distribution layer to obtain the aggregate class distribution $\bar{\mathbf{P}}$ and cosine similarity between the prototypes. Finally, we calculate the loss for weighted combinations of class distribution, aggregate class distribution and cosine similarity between the prototypes. By reducing the loss of the aggregate class distribution of samples, the model can enlarge the difference between different class sample features extracted by the model to improve the model's robustness and prediction ability.

Classification layer. After obtaining the features of the samples, we make the features pass through the classification layer to obtain the predicted class distribution, and this process can be written as

$$\mathbf{P} = \mathrm{CMLP}(\mathbf{V}), \tag{9}$$

where $\mathbf{P} \in \mathbb{R}^{n\times 2}$ is the predicted class distribution of the samples, and CMLP is a multilayer perceptron for classification [51].

Aggregate distribution layer. In order to accurately calculate the similarity of the samples feature $\mathbf{V}$, we first normalize $\mathbf{V}$. The normalization process is as follows.

$$\begin{aligned} \mu &= \frac{1}{n \times d_v} \sum_{i=1}^{n} \sum_{j=1}^{dv} v_{ij}, \\ \sigma &= \sqrt{\frac{1}{n \times d_v} \sum_{i=1}^{n} \sum_{j=1}^{dv} (v_{ij} - \mu)^2}, \\ v'_{ij} &= \frac{v_{ij} - \mu}{\sqrt{\sigma^2 + \varepsilon}} * g + k, \end{aligned} \tag{10}$$

where μ is the mean value of $\mathbf{V}$, σ is the variance of $\mathbf{V}$, ε is a very small decimal number, preventing division by 0, g and k are trainable parameters to ensure that the normalization operation will not destroy the previous information, v_{ij} is the element of the ith row and jth column of the feature matrix $\mathbf{V}$, and v'_{ij} is the element of the ith row and jth column of the normalized feature matrix $\mathbf{V}'$.

After normalizing the features extracted from the feature extraction layer, we calculate the similarity measure function of the features of the same batch of data to obtain a similarity matrix. Then we normalize the similarity matrix by the Softmax function. The calculation process is as follows

$$\mathbf{A} = \text{Softmax}\left(\mathbf{V}'\mathbf{V}'^{\mathrm{T}} / \sqrt{d_v}\right), \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the feature similarity matrix of the samples, and the superscript T denotes transpose of matrices.

A class distribution independently predicted from an individual sample often contains ambiguous values and often generates erroneous labels. To remedy this, the AggMatch algorithm [49] uses the aggregate class distribution of a sample to refine the ambiguous or noisy class distributions utilizing the relationship between other confidence-aware samples. Specifically, a class distribution p_b for v_b is aggregated by p_l of other samples v_l from the set of B, by considering the similarity between them as

$$\bar{p}_b = \sum_l \left(\frac{\exp(S(v'_b, v'_l)/\tau_{sim})}{\sum_j \exp(S(v'_b, v'_j)/\tau_{sim})} \right) p_l, \tag{12}$$

where l and j are indexes for all samples in the set of B, and τ_{sim} is a temperature. In particular, the similarity function S measures the similarity weights between the samples.

The performance of the aggregate class distribution of the samples is highly dependent on the class distribution of the batch data; therefore, we set the class distribution of our batch data as a uniform distribution. This raises another problem, for the aggregate class distribution of a sample, the similarity weights of samples of different classes from the sample in a batch is close to zero, which is not significant in calculating the gradient in the training process.

Weight adjustment. In order to solve the above problems, we adjusted the similarity weights to better adapt to the training process. We subtract $\mathbf{I}/n$ (the average weight of the same batch of data) from the similarity matrix so that the similarity weights of the features similar to the samples in the same batch are positively correlated, and the similarity weights of the features dissimilar to the samples are negatively correlated. As shown in Figure 2, by adjusting the similarity weights, training the aggregate class distribution of a sample can strengthen both the features of samples similar to the sample and those dissimilar to the sample simultaneously and enlarge the difference between different class sample

features extracted by the model to improve the model's robustness and prediction ability. The calculation process can be written as

$$\mathbf{B} = \mathbf{A} - \mathbf{I}/n, \tag{13}$$

where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is a positive and negative correlation similarity matrix, and $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix.

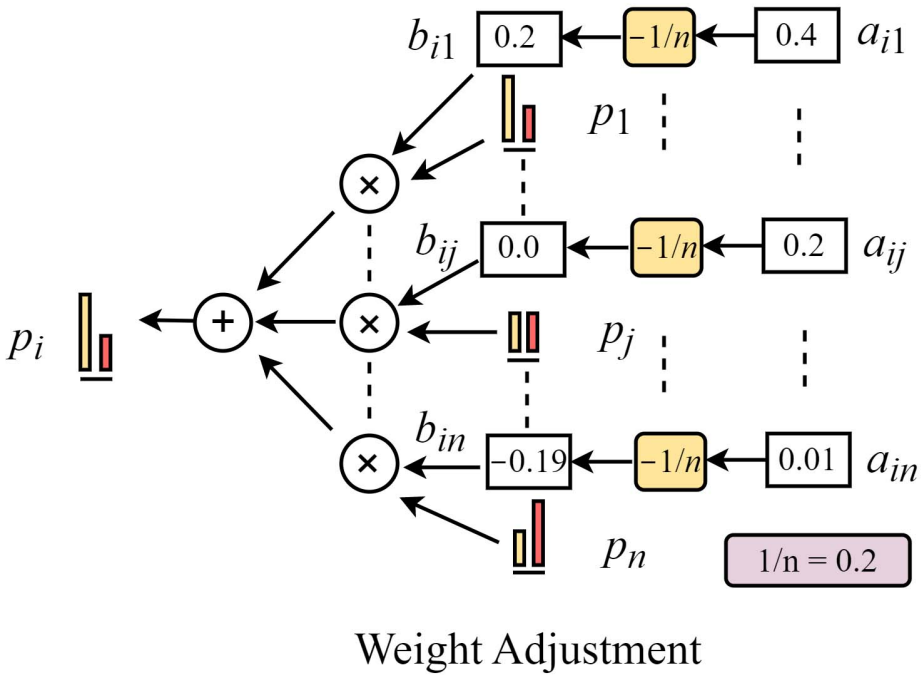

Figure 2. Illustration of weight adjustment. a_{ij} is the element of the ith row and jth column of the similarity matrix $\mathbf{A}$, and b_{ij} is the element of the ith row and jth column of the positive and negative correlation similarity matrix $\mathbf{B}$. By adjusting the similarity weights, the aggregate class distribution of samples can sufficiently combine the class distribution of samples with similar and dissimilar features.

Then we obtain the aggregation distribution by multiplying the similarity matrix and the predicted class distribution,

$$\bar{\mathbf{P}} = \mathbf{B} \cdot \mathbf{P}, \tag{14}$$

where $\bar{\mathbf{P}} \in \mathbb{R}^{n \times 2}$ is the aggregate class distribution.

In order to further improve the performance of the aggregate class distribution, we calculate the d_v-dimensional representation $p_{steerable} \in \mathbb{R}^{d_v}$, $p_{unsteerable} \in \mathbb{R}^{d_v}$, or prototype, of each class and cosine similarity between the prototypes. We then add the coefficient of similarity between prototypes into the loss function to increase the feature difference between different classes. The calculation process can be written as

$$\begin{aligned} p_{unsteerable} &= \tfrac{1}{n/2} \sum_{i \in \{1,\ldots,n/2\}} v'_i, \\ p_{steerable} &= \tfrac{1}{n/2} \sum_{i \in \{(n/2)+1,\ldots,n\}} v'_i, \\ ps &= (p_{steerable} \cdot p_{unsteerable}) / \|p_{steerable}\| \|p_{unsteerable}\|, \end{aligned} \tag{15}$$

where ps is the cosine similarity between the prototypes.

Loss function. The loss function consists of three parts: the first part is the cross-entropy loss of the aggregate class distribution $\bar{\mathbf{P}}$, and the second part is the cross-entropy

loss of **P** obtained by the classification layer, and the third part is the cosine similarity between the prototypes *ps*.

$$\text{Loss} = \lambda \cdot \frac{1}{n}\sum_{i=1}^{n} D(y_i, \bar{p}_i) + \mu \cdot \frac{1}{n}\sum_{i=1}^{n} D(y_i, p_i) + \gamma \cdot ps, \tag{16}$$

where $\bar{p}_i$ and p_i are the aggregate and predicted class distribution of the ith sample, respectively, λ, μ and γ both are weight parameters, and D is the cross-entropy loss function.

Training and validation. In the training process, the data pass through the feature extraction layer, classification layer and aggregate class distribution layer to obtain the loss. In the verification process, the data pass through the feature extraction layer and classification layer to obtain the final prediction result.

Workflow of the AGGNN algorithm. In summary, we show the workflow of the AGGNN algorithm in Algorithm 1. For each batch of training examples, the AGGNN algorithm performs the following operations. First of all, the algorithm obtains the features and the class distribution of the batch examples. Then the network computes the feature similarity matrix and transforms the weights into a positive and negative correlation similarity matrix, obtaining the aggregation distribution for each sample. Finally, the model parameters are optimized by reducing the loss of the model.

Algorithm 1 The AGGNN algorithm.

Require: batch $B = \{(x_i, y_i)\}_{i=1}^{n}$
Require: $y_{i\in\{1,\ldots,n/2\}} = 1, y_{i\in\{(n/2)+1,\ldots,n\}} = 0$
  for t in [1, num_epochs] **do**
    for each minibatch B **do**
      $\mathbf{X} \leftarrow [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$
      $\mathbf{V}, \mathbf{P} \leftarrow p_{\text{ mod } el}(\mathbf{X}; \theta)$
      $\mu \leftarrow \frac{1}{n \times d_v} \sum_{i=1}^{n} \sum_{j=1}^{dv} v_{ij}$
      $\sigma \leftarrow \sqrt{\frac{1}{n \times d_v} \sum_{i=1}^{n} \sum_{j=1}^{dv} (v_{ij} - \mu)^2}$
      $v'_{ij} \leftarrow \frac{v_{ij} - \mu}{\sqrt{\sigma^2 + \varepsilon}} * g + k$
      $\mathbf{A} \leftarrow \text{Softmax}\left(\mathbf{V}\mathbf{V}^{\mathrm{T}} / \sqrt{d_v}\right)$
      $\mathbf{B} \leftarrow \mathbf{A} - \mathbf{I}/n$
      $\bar{\mathbf{P}} \leftarrow \mathbf{B} \cdot \mathbf{P}$
      $p_{unsteerable} \leftarrow \frac{1}{n/2} \sum_{i\in\{1,\ldots,n/2\}} v'_i$
      $p_{steerable} \leftarrow \frac{1}{n/2} \sum_{i\in\{(n/2)+1,\ldots,n\}} v'_i$
      $ps \leftarrow (p_{steerable} \cdot p_{unsteerable}) / \|p_{steerable}\| \, \|p_{unsteerable}\|$
      Loss $\leftarrow \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} D(y_i, \bar{p}_i) + \mu \cdot \frac{1}{n} \sum_{i=1}^{n} D(y_i, p_i) + \gamma \cdot ps$
      Update θ with Adam to minimize Loss
    end for
  end for
Return: θ

3. Detecting the Steerability by AGGNN

The BPNN machine learning model has been explored to detect the steerability of a quantum state [48]. Due to the inevitability of errors in the training data evaluated by the SDP, the BPNN still has the problem of overfitting. The AGGNN makes samples with similar features have a similar class prediction distribution according to the distance measure of sample features, thus reducing the network prediction error and making it

easier to find a model with good performance in the training process. According to the superiority of the AGGNN, we use it to detect the quantum steerability.

3.1. Datasets

In order to train the quantum steering classification models, we need to collect the data for quantum states and select the features for the data. Inspired by [45], we generate two random 4×4 complex matrices, A and B. Then we use the two matrices to generate a Hermitian matrix $C \equiv (A + \mathrm{i}B)(A + \mathrm{i}B)^{\dagger}$, where $\dagger$ means conjugate transpose. Finally, we obtain a density matrix $\rho \equiv C/\mathrm{tr}(C)$. Thus, we can determine the steerability of each sample quantum state by utilizing SDP. We label the steerable state by "1" and the unsteerable state by "0". In this paper, we use the datasets in [45], collected by the following four kinds of feature vectors.

F1: Every feature vector in F1 is a 15-dimensional vector, and the component is one of the set $\{\rho_{ii}$, the real and imaginary part of $\rho_{ij}, i > j\}, i, j \in \{1, 2, 3\}$.

F2: Every feature vector in F2 is a 9-dimensional vector, and the component is one of the set $\{\mathrm{tr}[(\sigma_k \otimes \sigma_l)\rho]\}, k, l \in \{1, 2, 3\}$.

The Bloch representation of an arbitrary two-qubit density operator ρ is as follows

$$\rho = \frac{1}{4}\left(I + \sum_{i=1}^{3} x_i \sigma_i \otimes I^B + \sum_{j=1}^{3} y_j I^A \otimes \sigma_j + \sum_{k,l=1}^{3} t_{kl} \sigma_k \otimes \sigma_l\right), \tag{17}$$

where $t_{kl} = \mathrm{tr}[(\sigma_k \otimes \sigma_l)\rho]$ is a component of the correlation matrix. We extract the partial information by computing $\{t_{kl}\}$ as features, since the correlation matrix can demonstrate a certain quantum correlation. We aim to obtain a high-performance machine learning model with partial information for only three fixed measurement directions x, y, z.

F3: Every feature vector in F3 is a 9-dimensional vector, and the component is one of the set $\{\mathrm{tr}[(\sigma_k \otimes \sigma_l)\rho']\}, k, l \in \{1, 2, 3\}$, where $\rho \to \rho' \equiv (I^A \otimes \sqrt{\rho^B})\rho(I^A \otimes \sqrt{\rho^B})$.

To further explore a high-performance machine learning model with partial information, we convert state ρ into the canonical form ρ' by local unitaries, preserving the steerability of ρ. As proven in [50], the map is given by

$$\rho \to \rho' \equiv (I^A \otimes \sqrt{\rho^B})\rho(I^A \otimes \sqrt{\rho^B}), \tag{18}$$

where $\rho^B = \mathrm{tr}_A \rho$.

Similarly, we extract the coefficients of the correlation terms of ρ' and t'_{kl} to combine the feature vector. Similar to the F2 case, we only need to measure an arbitrary two-qubit state in three fixed directions x, y, z to predict the steerability of it.

F4: Every feature vector is composed of six features in F4, that is, F_3 except for the terms of $\{\sigma_2 \otimes \sigma_1, \sigma_3 \otimes \sigma_1, \sigma_3 \otimes \sigma_2\}$.

To explore a high-performance machine learning model of steering with less information, according to the symmetry, we drop the coefficients of the correlation terms, $\{\sigma_2 \otimes \sigma_1, \sigma_3 \otimes \sigma_1, \sigma_3 \otimes \sigma_2\}$.

According to the number of measurements $m = 2, 3, \ldots, 8$, we can divide the whole dataset into 28 datasets after selecting the feature vectors. For each m, we have at least 5000 examples labelled "1" and 5000 examples with labelled "0" in the corresponding dataset. We then randomly select 1000 positive examples and 1000 negative examples as the test set, and the rest as the training set. We employ the AGGNN to train a model for every dataset.

3.2. Training and Testing

In the AGGNN model, the structure parameters of the network are shown in the Table 1. Features of the batch samples are normalized by layer normalization of dimension $[n, d_v]$. We set the weight parameters of the loss function λ equal to 0.5, μ equal to 0.5 and

γ between 0.01 and 0.001. The optimization algorithm of the network is the Adam gradient descent algorithm. We set the batchsize to 200 and the learning rate to 0.001.

Table 1. The structural parameters of AGGNN.

Feature	The Number of Hidden Layers		The Number of Neurons in Each Layer	d_v
	Feature Extraction Layer	Classification Layer		
F1	2	1	1000	1000
F2	2	1	500	500
F3	2	1	500	500
F4	2	1	200	200

To verify that our model has a better generalization ability and robustness to inaccurate data in quantum-steering detection problems, we trained the models on the F1–F4 feature datasets, comparing their classification accuracy in the cross validation test set, random test set and Werner state validation set, Figure 3a–d. Furthermore, we compared the training difficulty of the AGGNN and BPNN models, as in Figure 4. The classification accuracy of a model in a dataset is the percentage of correctly predicted number of samples corresponding to the size of the dataset.

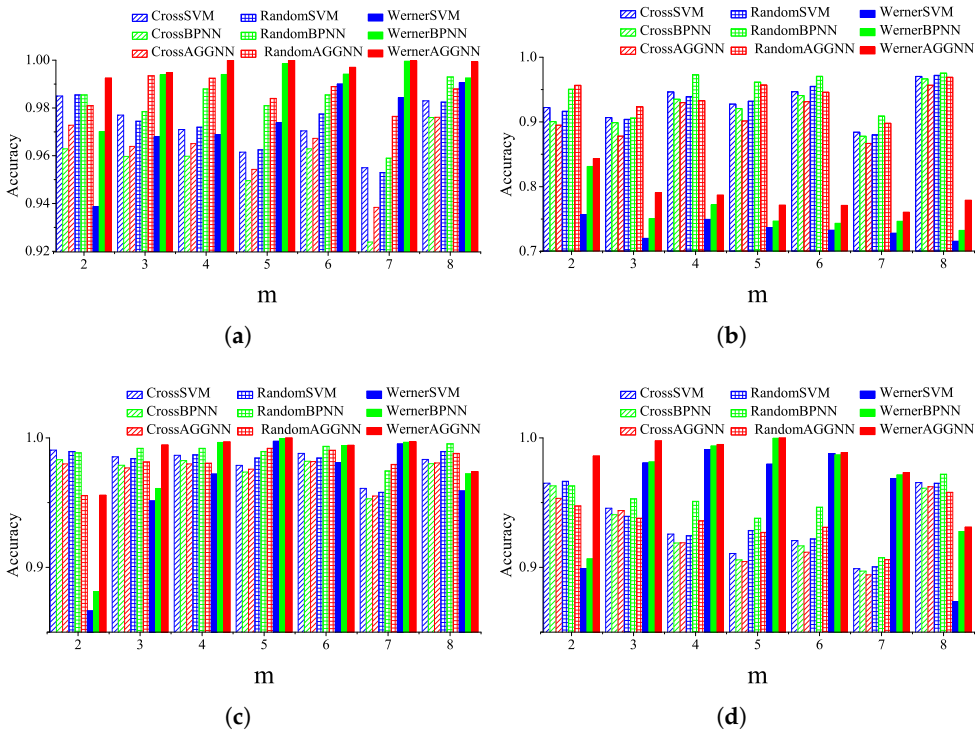

Figure 3. (**a**–**d**) Classification accuracy of the classification models with F1–F4 features. The first (diagonal-filled blue), second (diagonal-filled green) and third (diagonal-filled red) columns illustrate the accuracy of the cross validation of the classification models trained by the SVM, BPNN and AGGNN, respectively. The forth (grid-filled blue), fifth (grid-filled green) and sixth (grid-filled red) columns illustrate the classification accuracy on random states of the classification models trained by the SVM, BPNN and AGGNN, respectively. The seventh (solid blue), eighth (solid green) and ninth (solid red) columns illustrate the classification accuracy on the Werner state of the classification models trained by the SVM, BPNN and AGGNN, respectively. The label m is the number of measurements.

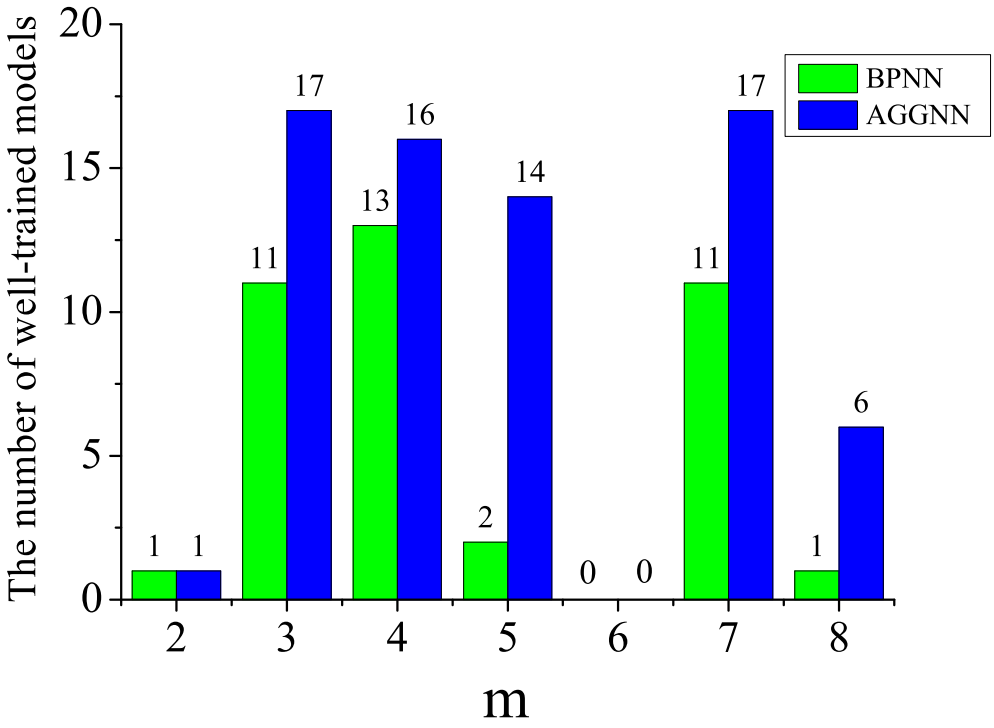

Figure 4. The number of well-trained models by the BPNN, AGGNN on F1 features. The first (diagonal-filled green) and second (diagonal-filled blue) columns depict the number of well-trained models by the BPNN and AGGNN, respectively. The label m is the number of measurements.

In order to make our model have better stability and generalization, inspired by [45], we used four-fold cross validation to train the model. In this method, the data set is divided into four mutually exclusive subsets of similar size, and the data distribution consistency of each subset is maintained as much as possible. Then, each time, the union of three subsets is used as the training set, and the remaining subset is used as the test set. In this way, we can obtain four training test sets to conduct four training and testing experiments to return the mean of the four test results.

First, to verify the model's generalization ability to the training data, we obtained the test accuracy rate of four-fold cross validation. In Figure 3a–d, the four-fold cross-validation accuracies of the classification models trained by the SVM [45], BPNN [48] and AGGNN are depicted by the first (diagonal-filled blue), second (diagonal-filled green) and third (diagonal-filled red) columns, respectively.

Second, in order to verify the learning ability of the model to the dataset, we randomly took 2000 samples from the dataset as a random test set. In Figure 3a–d, the random test set accuracies of the classification models trained by the SVM [45], BPNN [48] and AGGNN are depicted by the fourth (grid-filled blue), fifth (grid-filled green) and sixth (grid-filled red) columns, respectively.

Third, because the data generated by the SDP is inaccurate, the high accuracy of the model on cross-validation test sets and random test sets may lead to overfitting problems. This problem can be avoided by verifying the classification accuracy of the model on the Werner state validation sets. These states have correct labels. Compared to cross-validation test sets and random test sets, having high accuracy on the Werner state validation sets is more likely to be a good model. In Figure 3a–d, the Werner state ($\theta = \frac{\pi}{4}$) validation set's accuracies of the classification models trained by the SVM [45], BPNN [48] and AGGNN are depicted as solid blue, solid green and solid red columns, respectively. The generalized Werner state is unsteerable from Alice to Bob if θ and β satisfy the following inequality,

$$\cos^2 2\theta \geqslant \frac{2\beta - 1}{(2-\beta)\beta^3}. \tag{19}$$

We can clearly see that the theoretical bound value that Alice can steer Bob's state is determined by Equation (19). In this experiment we generate four Werner state test sets with $\theta = \{\frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{8}, \frac{\pi}{12}\}$, respectively. For each Werner state test set, we generated 5000 steerable states and 5000 unsteerable states based on the uniform distribution of β, utilizing them to construct a dataset for every feature (F1–F4), where each instance contains a feature Fi (i = 1,2,3,4), and a label ("0" for steerable and "1" for unsteerable).

Finally, to explored whether our model could achieve better robustness to inaccurate quantum states and a better learning ability. For F1 features we believe the model can be well trained if it reaches more than 99% accuracies on the four validation sets generated by the Werner state ($\theta = \{\frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{8}, \frac{\pi}{12}\}$). We conducted five training sets, obtained 20 models, and then counted the model's accuracy on the four validation sets to obtain the number of well-trained models. In Figure 4, the number of well-trained models by the BPNN are depicted by the first (solid-filled green) column, while those trained by the AGGNN are depicted by the second (solid-filled blue) column. The experiments show that the AGGNN model can be easily trained to produce a good model. Because the data generated by the SDP has errors and an uneven data distribution, there is no well-trained model when $m = 6$. The AGGNN will train better models if the datasets are regenerated or the amount of data increases.

Interestingly, for the F1 feature with whole information of a quantum state, the classification accuracy of the AGGNN models on the Werner state validation sets is higher than the SVM and BPNN models. The classification accuracy of the AGGNN for most cross-validation and random test sets is higher than the BPNN, and the AGGNN is more accessible to train than the BPNN. However, the classification accuracy of the AGGNN for cross-validation test sets is lower than the SVM, but not by much. This indicates that the AGGNN has good generalization abilities and can effectively prevent overfitting. For F2–F4 features with partial information, the classification accuracy of the AGGNN in Werner state validation sets is higher than the SVM and BPNN. However, the classification accuracy of the AGGNN for the cross-validation and random test sets decreases because there are many misjudgments and inaccurate quantum states in the training data. The AGGNN can reduce the learning ability of error data and enlarge the difference between different class sample features extracted by the model to improve the model's robustness and prediction ability, so the classification accuracy of Werner state validation sets can further increase. In contrast, the classification accuracy of cross-validation and random test sets decreases.

3.3. Predicting the Steerability Bounds

In this part, the steerability bounds of the generalized Werner state are predicted by the quantum steering classification models trained by AGGNN and compared with the bounds computed by SDP [35], trained by SVM [45], and trained by BPNN [48].

Figure 5a–d depict the classification models trained with the F1–F4 features, respectively. Figure 5a shows the four subgraphs with $\theta = \{\frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{8}, \frac{\pi}{12}\}$, respectively, as illustrated above each chart. In all charts, the cyan-square lines depict the bounds predicted by the classification models trained by the SVM for $m = 2, \ldots, 8$. The blue-star lines depict the bounds predicted by the classification models trained by the BPNN for $m = 2, \ldots, 8$. The black-plus lines depict the bounds predicted by the classification models trained by the AGGNN for $m = 2, \ldots, 8$. The yellow-circle lines depict the bounds computed by SDP with $m = 2, \ldots, 8$. The red-cross lines depict the steerability bounds from Alice to Bob which are determined by Equation (19). We can clearly see that the predicted steerability bounds by the SVM [45], BPNN [48] and AGGNN are more similar to the theoretical bounds than those calculated by SDP [35], especially concerning the F1 feature. The steerability bounds predicted by the AGGNN are more similar to the theoretical bounds than those predicted by the SVM and BPNN.

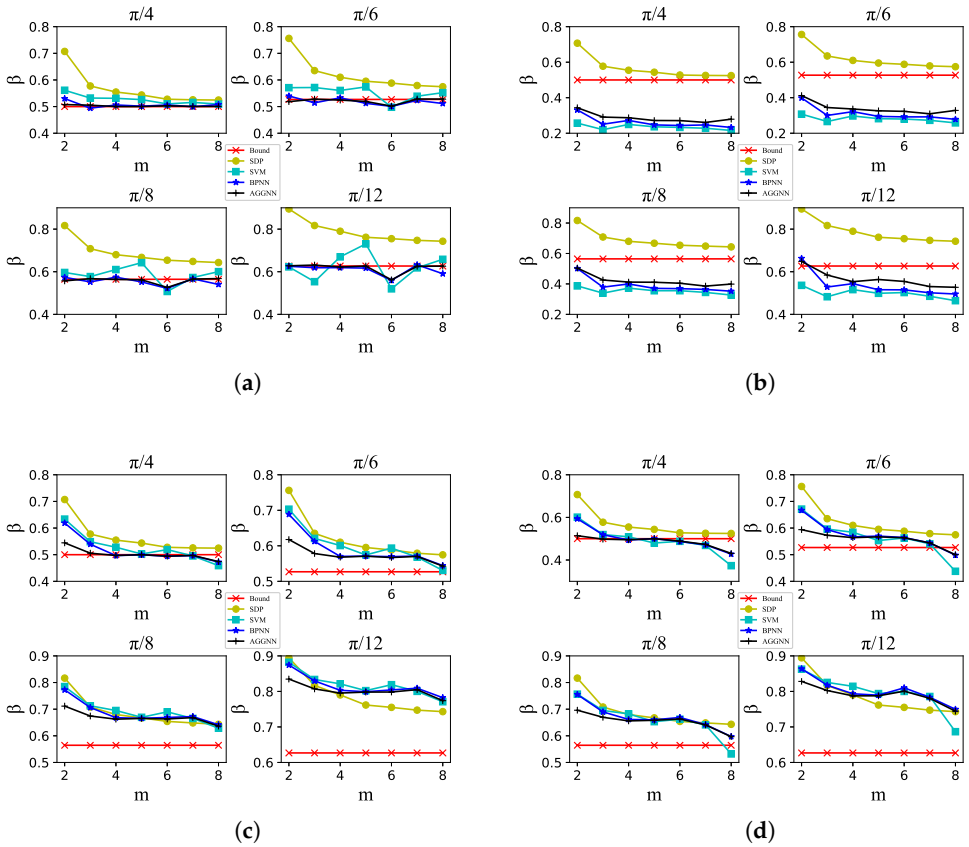

Figure 5. (**a**–**d**) The predictions of the steerability bounds for the generalized Werner states by the trained classification models and SDP with the features F1–F4, respectively. The cyan-square lines depict the bounds predicted by the SVM-trained classification model, the blue-star lines depict the bounds predicted by the BPNN-trained classification model, the black-plus lines are the bounds predicted by the AGGNN-trained classification model, the yellow-circle lines depict the bounds computed by SDP, and the red-cross lines depict the steerability bound from Alice to Bob.

Figure 5a–d illustrate that the steerability bounds predicted by the AGGNN are more similar to the theoretical bounds than those predicted by the SVM [45] and BPNN [48]. This is because the AGGNN makes samples with similar features have a similar class prediction distribution according to the distance measure of the sample features; therefore, the model can reduce the learning of mislabelled or inaccurate quantum states, making it easier for the model to converge to solutions similar to the theoretical boundaries in the training process. Furthermore, when the similarity between the predicted steerability and theoretical bounds increases, the AGGNN will improve the classification accuracy of the Werner state validation sets. For features F2–F4 with partial information of a quantum state, in some cases, the boundary predicted by the SDP is more similar to the theoretical boundaries than the machine learning method, possibly due to data distribution problems caused by inaccurate data.

The above results indicate that the AGGNN is an efficient model for steerability detection. For a quantum state with all information, our model performs better than the BPNN and SVM. For partial information, the predicted steerability boundary of our model is more similar to the theoretical boundary than the BPNN and SVM. Furthermore, the AGGNN can predict the steerability bounds of other states' ensembles, such as Tstate [52]. Moreover, our model has a roughly similar time consumption to the BPNN and SVM. Taking $m = 8$ as an example, the classification model trained by these three machine learning methods spends about 10^{-2} s predicting a quantum state, while the SDP [35]

program spends about 10^2 s. This demonstrates the time superiority of the machine learning methods.

3.4. Comparing the Classification Models Trained with Different Features

In Figure 3a–d, we show the classification models with the F1 feature with a quantum state with whole information and classification models with features F2–F4 with partial information. Naturally, whole information F1 is better than partial information F2–F4. In partial information, F3 and F4 have a better performance than F2. In this part, we compared the classification models trained with features Fi (i = 1,2,3,4) and different measurements m.

It is easy to speculate that the well-trained model will become more accurate and better at learning steerable samples as m rises. To prove this prediction, we randomly took 1000 steerable and 1000 unsteerable samples from the datasets for $m = 8$ to test the classification models of different m. Figure 6 shows the classification accuracy for 1000 random steerable and 1000 random unsteerable samples from the datasets for $m = 8$ and the blue-plus line, orange-circle line, green-cross line, and red-star line depict features F1–F4, respectively. Figure 7 shows the classification accuracy for 1000 random unsteerable samples from the datasets for $m = 8$ and the blue-plus dashed line, orange-circle dashed line, green-cross dashed line, and red-star dashed line depict features F1–F4, respectively. Figure 8 shows the classification accuracy for 1000 random steerable samples from the datasets for $m = 8$ and the blue-plus dotted line, orange-circle dotted line, green-cross dotted line, and red-star dotted line depict features F1–F4, respectively.

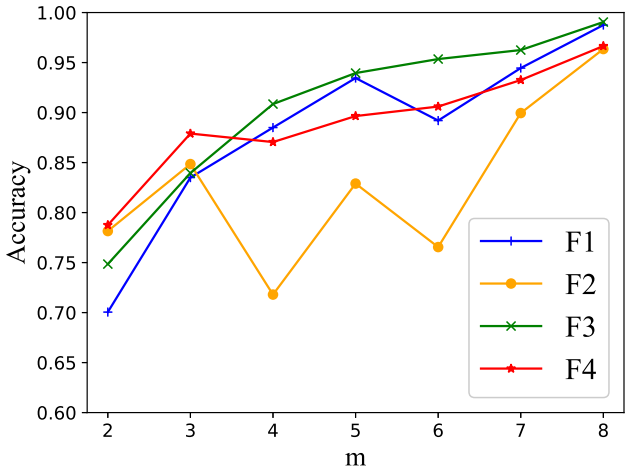

Figure 6. Classification accuracy of 1000 random steerable and 1000 random unsteerable samples for $m = 8$ with different classification models.

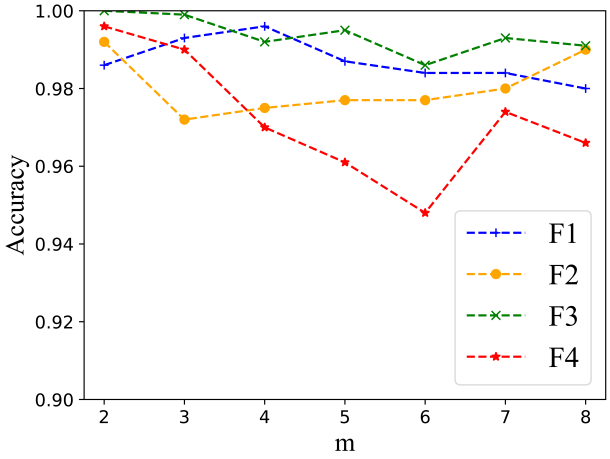

Figure 7. Classification accuracy of 1000 random unsteerable samples for $m = 8$ with different classification models.

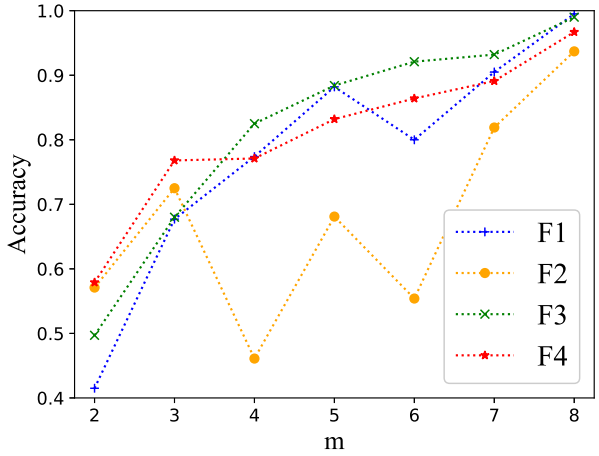

Figure 8. Classification accuracy of 1000 random steerable samples for $m = 8$ with different classification models.

Figures 6–8 show that the classification accuracy of 1000 random steerable and 1000 random unsteerable samples increases as m rises. The classification accuracy of 1000 random unsteerable samples does not change much; the classification accuracy of 1000 random steerable samples is the main factor affecting the total classification accuracy. Although some inflection points exist, this may be due to inaccurate data or an uneven data distribution. Furthermore, F3 has the best performance.

Figure 9 illustrates the classification accuracy for the Werner state validation sets $\theta = \{\frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{8}, \frac{\pi}{12}\}$ for each AGGNN classification models with the features F1–F4, respectively. The first subgraph for F1 features shows that the classification accuracy for the Werner state validation sets are all above 99% except for $m = 6$. The second subgraph shows the steerability classification model of F2 features. The classification accuracy increases as the parameter θ decreases and is over 76%. The third and fourth subgraphs show the classification accuracy for the Werner state validation sets of features F3 and F4. F3 and F4 have similar and correct trends; the accuracy increases with a increase in m, and the accuracy decreases with a decrease in θ. Furthermore, the steerability classification models of features F3 and F4 perform with a relatively higher classification accuracy on the Werner state validation sets.

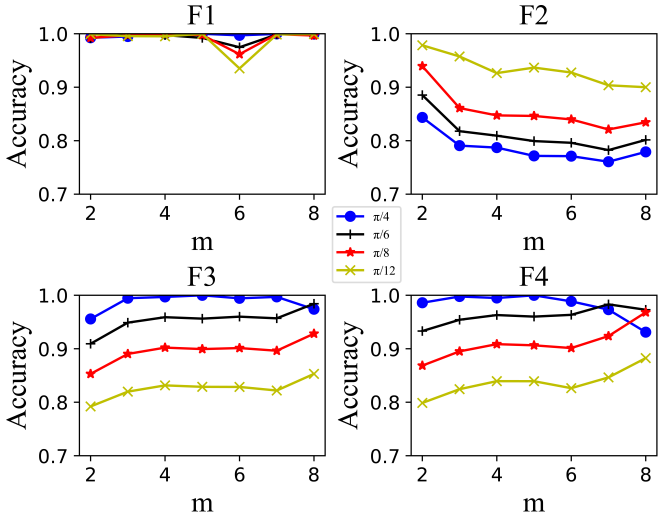

Figure 9. Classification accuracy for the Werner state validation sets for each AGGNN classification model for the features F1–F4, respectively.

4. Conclusions

In this paper, we presented a new neural network classification model, the AGGNN, to detect the steerability of bipartite quantum states and predict the steerability bounds of the generalized Werner states. This study has shown that the AGGNN can reduce the overfitting of inaccurate quantum state and improve the generalization ability of the model. On the one hand, for both whole information and partial information of a quantum state, the AGGNN can construct high-performance quantum steering classification models compared with the BPNN and SVM models. On the other hand, using the AGGNN classification models to predict the steerability bounds of the generalized Werner states are more similar to the theoretical bounds. In summary, the AGGNN can effectively detect the steerability of a large number of arbitrary states in quantum information processing. In the future, quantum-steering detection may be expanded to use deep semi-supervised learning to leverage large amounts of unlabelled data to improve model generalization and the AGGNN for the classification of multi-party quantum steering in large-scale systems.

Author Contributions: Conceptualization, K.H. and Y.H.; methodology, Y.H.; software, Y.H.; validation, Y.H.; formal analysis, K.H.; investigation, Y.H.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.H.; writing—review and editing, Y.Z.; visualization, Y.H.; supervision, K.H.; project administration, K.H.; funding acquisition, K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China [Grants No. 12271394] and Key Research and Development Program of Shanxi Province [Grants No. 202102010101004].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available upon request.

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

References

1. Schrödinger, E. Discussion of probability relations between separated systems. *Math. Proc. Camb.* **1935**, *31*, 555–563. [CrossRef]
2. Einstein, A.; Podolsky, B.; Rosen, N. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.* **1935**, *47*, 777–780. [CrossRef]
3. Wiseman, H.; Jones, S.; Doherty, A. Steering, entanglement, nonlocality, and the Einstein–Podolsky-Rosen paradox. *Phys. Rev. Lett.* **2007**, *98*, 140402. [CrossRef] [PubMed]
4. Gühne, O.; Tóth, G. Entanglement detection. *Phys. Rep.* **2009**, *474*, 1–75. [CrossRef]
5. Horodecki, R.; Horodecki, P.; Horodecki, M.; Horodecki, K. Quantum entanglement. *Rev. Mod. Phys.* **2009**, *81*, 865–942. [CrossRef]
6. Brunner, N.; Cavalcanti, D.; Pironio, S.; Scarani, V.; Wehner, S. Bell nonlocality. *Rev. Mod. Phys.* **2014**, *86*, 419–478. [CrossRef]
7. Jones, S.; Wiseman, H.; Doherty, A. Entanglement, Einstein–Podolsky-Rosen correlations, Bell nonlocality, and steering. *Phys. Rev. A* **2007**, *76*, 052116. [CrossRef]
8. Skrzypczyk, P.; Navascués, M.; Cavalcanti, D. Quantifying Einstein–Podolsky-Rosen steering. *Phys. Rev. Lett.* **2014**, *112*, 180404. [CrossRef]
9. Branciard, C.; Cavalcanti, E.; Walborn, S.; Scarani, V.; Wiseman, H. One-sided device-independent quantum key distribution: Security, feasibility, and the connection with steering. *Phys. Rev. A* **2012**, *85*, 010301(R). [CrossRef]
10. Gehring, T.; Händchen, V.; Duhme, J.; Furrer, F.; Franz, T.; Pacher, C.; Werner, R.; Schnabel, R. Implementation of continuous-variable quantum key distribution with composable and one-sided-device-independent security against coherent attacks. *Nat. Commun.* **2015**, *6*, 8795. [CrossRef]
11. Walk, N.; Hosseini, S.; Geng, J.; Thearle, O.; Haw, J.; Armstrong, S.; Assad, S.; Janousek, J.; Ralph, T.; Symul, T.; et al. Experimental demonstration of Gaussian protocols for one-sided device-independent quantum key distribution. *Optica* **2016**, *3*, 634–642. [CrossRef]
12. Wang, Y.; Bao, W.; Li, H.; Zhou, C.; Li, Y. Finite-key analysis for one-sided device-independent quantum key distribution. *Phys. Rev. A* **2013**, *88*, 052322. [CrossRef]
13. Zhou, C.; Xu, P.; Bao, W.; Wang, Y.; Zhang, Y.; Jiang, M.; Li, H. Finite-key bound for semi-device-independent quantum key distribution. *Opt. Express* **2017**, *25*, 16971–16980. [CrossRef] [PubMed]
14. Kaur, E.; Wilde, M.; Winter, A. Fundamental limits on key rates in device-independent quantum key distribution. *New J. Phys.* **2020**, *22*, 023039. [CrossRef]

15. Piani, M.; Watrous, J. Necessary and sufficient quantum information characterization of Einstein–Podolsky-Rosen steering. *Phys. Rev. Lett.* **2015**, *114*, 060404. [CrossRef]
16. Sun, K.; Ye, X.; Xiao, Y.; Xu, X.; Wu, Y.; Xu, J.; Chen, J.; Li, C.; Guo, G. Demonstration of Einstein–Podolsky-Rosen steering with enhanced subchannel discrimination. *NPJ Quantum Inf.* **2018**, *4*, 12. [CrossRef]
17. Passaro, E.; Cavalcanti, D.; Skrzypczyk, P.; Acín, A. Optimal randomness certification in the quantum steering and prepare-and-measure scenarios. *New J. Phys.* **2015**, *17*, 113010. [CrossRef]
18. Skrzypczyk, P.; Cavalcanti, D. Maximal randomness generation from steering inequality violations using qudits. *Phys. Rev. Lett.* **2018**, *120*, 260401. [CrossRef]
19. Coyle, B.; Hoban, M.; Kashefi, E. One-sided device-independent certification of unbounded random numbers. *EPTCS* **2018**, *273*, 14–26. [CrossRef]
20. He, Q.; Rosales-Zárate, L.; Adesso, G.; Reid, M. Secure continuous variable teleportation and Einstein–Podolsky-Rosen steering. *Phys. Rev. Lett.* **2015**, *115*, 180502. [CrossRef]
21. Reid, M. Demonstration of the Einstein–Podolsky-Rosen paradox using nondegenerate parametric amplification. *Phys. Rev. A* **1989**, *40*, 913–923. [CrossRef] [PubMed]
22. Reid, M.; Drummond, P.; Bowen, W.; Cavalcanti, E.; Lam, P.; Bachor, H.; Andersen, U.; Leuchs, G. Colloquium: The Einstein–Podolsky-Rosen paradox: From concepts to applications. *Rev. Mod. Phys.* **2009**, *81*, 1727–1751. [CrossRef]
23. Cavalcanti, E.; Jones, S.; Wiseman, H.; Reid, M. Experimental criteria for steering and the Einstein–Podolsky-Rosen paradox. *Phys. Rev. A* **2009**, *80*, 032112. [CrossRef]
24. Walborn, S.; Salles, A.; Gomes, R.; Toscano, F.; Souto-Ribeiro, P. Revealing hidden Einstein–Podolsky-Rosen nonlocality. *Phys. Rev. Lett.* **2011**, *106*, 130402. [CrossRef]
25. Schneeloch, J.; Broadbent, C.; Walborn, S.; Cavalcanti, E.; Howell, J. Einstein–Podolsky-Rosen steering inequalities from entropic uncertainty relations. *Phys. Rev. A* **2013**, *87*, 062103. [CrossRef]
26. Pusey, M. Negativity and steering: A stronger Peres conjecture. *Phys. Rev. A* **2013**, *88*, 032313. [CrossRef]
27. Pramanik, T.; Kaplan, M.; Majumdar, A. Fine-grained Einstein–Podolsky-Rosen-steering inequalities. *Phys. Rev. A* **2014**, *90*, 050305(R). [CrossRef]
28. Kogias, I.; Skrzypczyk, P.; Cavalcanti, D.; Acín, A.; Adesso, G. Hierarchy of steering criteria based on moments for all bipartite quantum systems. *Phys. Rev. Lett.* **2015**, *115*, 210401. [CrossRef]
29. Cavalcanti, E.; Foster, C.; Fuwa, M.; Wiseman, H. Analog of the Clauser-Horne-Shimony-Holt inequality for steering. *J. Opt. Soc. Am. B* **2015**, *32*, A74–A81. [CrossRef]
30. Kogias, I.; Lee, A.; Ragy, S.; Adesso, G. Quantification of Gaussian quantum steering. *Phys. Rev. Lett.* **2015**, *114*, 060403. [CrossRef]
31. Zhu, H.; Hayashi, M.; Chen, L. Universal steering criteria. *Phys. Rev. Lett.* **2016**, *116*, 070403. [CrossRef]
32. Nguyen, H.; Vu, T. Necessary and sufficient condition for steerability of two-qubit states by the geometry of steering outcomes. *Europhys. Lett.* **2016**, *115*, 10003. [CrossRef]
33. Costa, A.C.S.; Angelo, R.M. Quantification of Einstein–Podolsky-Rosen steering for two-qubit states. *Phys. Rev. A* **2019**, *100*, 039901. [CrossRef]
34. Ming, F.; Song, X.K.; Ling, J.; Ye, L.; Wang, D. Quantification of quantumness in neutrino oscillations. *Eur. Phys. J. C* **2020**, *80*, 275. [CrossRef]
35. Vandenberghe, L.; Boyd, S. Semidefinite programming. *SIAM Rev.* **1996**, *38*, 49–95. [CrossRef]
36. Cavalcanti, D.; Skrzypczyk, P. Quantum steering: A review with focus on semidefinite programming. *Rep. Prog. Phys.* **2016**, *80*, 024001. [CrossRef] [PubMed]
37. 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]
38. Canabarro, A.; Brito, S.; Chaves, R. Machine learning nonlocal correlations. *Phys. Rev. Lett.* **2019**, *122*, 200401. [CrossRef]
39. Deng, D. Machine learning detection of Bell nonlocality in quantum many-body systems. *Phys. Rev. Lett.* **2018**, *120*, 240402. [CrossRef] [PubMed]
40. Ch'ng, K.; Carrasquilla, J.; Melko, R.; Khatamis, E. Machine learning phases of strongly correlated fermions. *Phys. Rev. X* **2017**, *7*, 031038. [CrossRef]
41. Yoshioka, N.; Akagi, Y.; Katsura, H. Learning disordered topological phases by statistical recovery of symmetry. *Phys. Rev. B* **2018**, *97*, 205110. [CrossRef]
42. Neugebauer, M.; Fischer, L.; Jäger, A.; Czischek, S.; Jochim, S.; Weidemüller, M.; Gärttner, M. Neural-network quantum state tomography in a two-qubit experiment. *Phys. Rev. A* **2020**, *102*, 042604. [CrossRef]
43. Fanchini, F.; Karpat, G.; Rossatto, D.; Norambuena, A.; Coto, R. Estimating the degree of non-Markovianity using machine learning. *Phys. Rev. A* **2021**, *103*, 022425. [CrossRef]
44. Zhang, Y.; Yang, L.; He, Q.; Chen, L. Machine learning on quantifying quantum steerability. *Quantum Inf. Process.* **2020**, *19*, 263. [CrossRef]
45. Ren, C.; Chen, C. Steerability detection of an arbitrary two-qubit state via machine learning. *Phys. Rev. A* **2019**, *100*, 022314. [CrossRef]
46. Yang, M.; Ren, C.; Ma, Y.; Xiao, Y.; Ye, X.; Song, L.; Xu, J.; Yung, M.; Li, C.; Guo, G. Experimental simultaneous learning of multiple nonclassical correlations. *Phys. Rev. Lett.* **2019**, *123*, 190401. [CrossRef]

47. Zhang, L.; Chen, Z.; Fei, S. Einstein–Podolsky-Rosen steering based on semisupervised machine learning. *Phys. Rev. A* **2021**, *104*, 052427. [CrossRef]
48. Zhang, J.; He, K.; Zhang, Y.; Hao, Y.; Hou, J.; Lan, F.; Niu, B. Detecting the steerability bounds of generalized Werner states via a backpropagation neural network. *Phys. Rev. A* **2022**, *105*, 032408. [CrossRef]
49. Kim, J.; Ryoo, K.; Lee, G.; Cho, S.; Seo, J.; Kim, D.; Cho, H.; Kim, S. AggMatch: Aggregating pseudo labels for semi-supervised learning. *arXiv* **2022**, arXiv:2201.10444v1. [CrossRef]
50. Bowles, J.; Hirsch, F.; Quintino, M.; Brunner, N. Sufficient criterion for guaranteeing that a two-qubit state is unsteerable. *Phys. Rev. A* **2016**, *93*, 022121. [CrossRef]
51. Rumelhart, D.; Hinton, G.; Williams, R. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
52. Jevtic, S.; Hall, M.J.W.; Anderson, M.R.; Zwierz, M.; Wiseman, H.M. Einstein–Podolsky–Rosen steering and the steering ellipsoid. *J. Opt. Soc. Am. B* **2015**, *32*, A40–A49. [CrossRef]