Supplementary Material for "State Classification via a Random-Walk-Based Quantum Neural Network"

Lu-Ji Wang, Jia-Yi Lin, and Shengjun Wu
Institute for Brain Sciences and Kuang Yaming Honors School, Nanjing University, Nanjing 210023, China and School of Physics, Nanjing University, Nanjing 210093, China

I. TOPOLOGY OF QSNNS

A. Absence of the Hamiltonian coupling

As described by Eq. (2) of the main text, Hamiltonian couplings h_{ij} can exist between any two neurons ($|i\rangle$ and $|j\rangle$) of the QSNN in general. However, in the tasks of our interests, the couplings h_{ij} only exist between neurons in the input and hidden layer, as the green dashed lines shown in Fig. 1 of the main text. There are only Lindblad operators (orange lines with arrows) between the neurons in the hidden layer and output layer. Because we expect the probability transmission between the hidden layer and output layer is one-way, so that probability converges in the output layer. But the Hamiltonian performs a non-directional transmission between the connected neurons. So, the existence of the couplings h_{ij} between the hidden layer and output layer might slow down the convergence of probability at the output layer, which further makes the QSNN difficult to be trained.

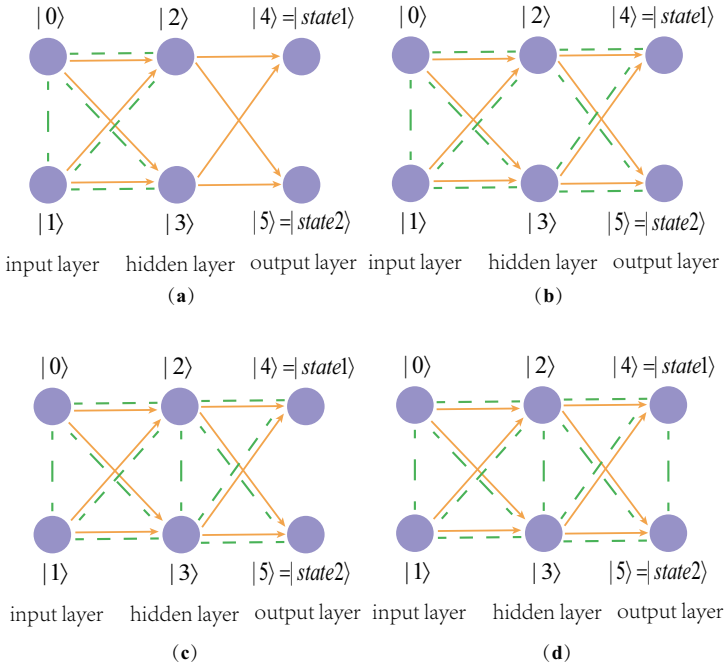

FIG. S1: (a), (b), (c) and (d) are the graph representations of the four QSNNs used for quantum state binary discrimination. The state of a QSNN can be represented by a density matrix in the $N = 6$ dimensional Hilbert space constituted by an orthogonal basis $\{|i\rangle\}_{i=0}^{5}$. The vertices (neurons) are decoherently connected by Lindblad operators (orange lines with arrows) and coherently connected by Hamiltonian elements (green dashed lines).

Then, we give the comparison between the training performances of the four QSNNs (shown in FIG. S1) here. We train them to complete the same task as that described in the main text *Results - Quantum State Binary Discrimination*, i.e., the discrimination between two states represented by

$$|\psi_\theta\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle.$$

The training set is $\{(|\psi_0\rangle, state1), (|\psi_\theta\rangle, state2)\}$. We train the QSNNs with the different training sets separately, where $\theta = 0, \frac{\pi}{6}, \frac{2\pi}{6}, \frac{3\pi}{6}, \cdots, \frac{11\pi}{6}$, and average over the results for these training sets. FIG. S2 shows the training

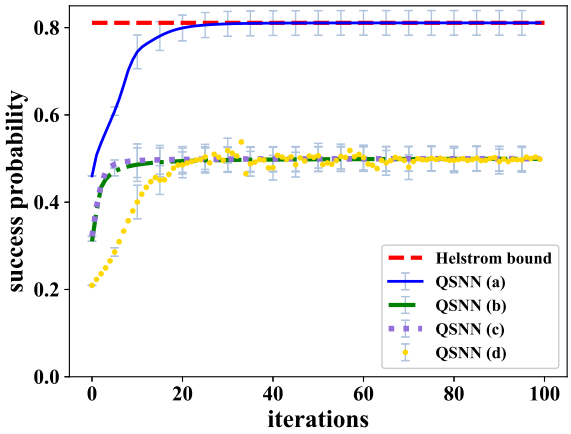

FIG. S2: The training performances of the 4 QSNNs (shown in FIG. S1) in the discrimination of quantum state pairs $(|\psi_0\rangle, |\psi_\theta\rangle)$. The results are given by taking the average over the success probabilities of all training sets with different θ. The red dashed line shows the average optimal theoretical value given by the minimum error discrimination. The other lines in different colors and shapes represent the average success probability of the 4 QSNNs. The error bars are plotted from the variances. We can see the QSNN (a), i.e. the QSNN we used in the main text, performs best. The success probabilities of other QSNNs are about 0.5, which are terrible performances.

performance of the 4 QSNNs. The QSNN shown as FIG. S1(a) performs best, and the other QSNNs all perform poorly.

In addition to the consideration in terms of training performance, we also have to consider the resource consumption of the model. The more parameters that need to be optimized, the more resources we consume.

In summary, the Hamiltonian couplings between the hidden layer and the output layer do not improve the training performance of the network but consume resources, so we use the QSNN shown as FIG. S1(a) without that.

B. Hidden layer

In traditional artificial neural networks, a hidden layer is a layer located between the input and output layer, where neurons take in a set of weighted inputs and produce an output through an activation function. The number and size of hidden layers can sometimes affect the performance of the network. However, more hidden layers or more hidden layer neurons do not necessarily lead to better model performance. One usually selects the appropriate number and size of hidden layers empirically based on the specific task. In our approach, we also follow this convention. For greater clarity, we use $\mathcal{I}-\mathcal{H}-\mathcal{O}$ to describe the structure of a QSNN, where $\mathcal{I}$, $\mathcal{H}$, and $\mathcal{O}$ are the numbers of neurons in the input layer, hidden layer(s), and output layer, respectively. For example, the QSNN we used in the state binary discrimination task is a 2-2-2 network as shown in Fig. 1 of the main text. Then, we give some empirical evidence to show that the QSNNs used in the main text are appropriate for our tasks, if both resource consumption and model performance are considered.

First, because Fig. 3(c) and (d) of the main text show that there is a gap between the success probability given by the 2-2-2 QSNN and the theoretical optimum in discriminating two pure states represented by Eq. (11) of the main text, we take this task as an example to investigate whether the 2-2-2 QSNN used in the main text is appropriate or not. We train the 2-2, 2-3-2, and 2-2-2-2 QSNN with the training sets $\{(|\psi_0\rangle, state1), (|\psi_\varphi\rangle, state2)\}$ which are different in $\varphi = 0, \frac{\pi}{6}, \frac{2\pi}{6}, \frac{3\pi}{6}, \cdots, \frac{11\pi}{6}$. We give the plot of the success probability with respect to the number of iterations used in the training procedure, as shown in FIG. S3. The 2-2 QSNN performs much worse than the other three QSNNs. And there is little difference in the performances of the 2-2-2, 2-3-2, and 2-2-2-2 QSNN. Considering that more hidden layers or more hidden layer neurons in a network will increase the consumption of resources, the 2-2-2 network is appropriate for this task.

Second, we demonstrate that the 4-4-2 QSNN used in the classification task described in the main text *Results - Classification of Entangled and Separable States* is appropriate. We train the 4-2, 4-4-4-2, and 4-5-2 QSNN with the training set as the same as that in *Results - Classification of Entangled and Separable States*, and use the same test set to evaluate the performance of the QSNNs. We show the performances of the 4 QSNNs in FIG. S4. Compared to the 4-4-2 QSNN used in the main text, the performances of the 4-4-4-2 and 4-5-2 QSNN are not greatly improved,

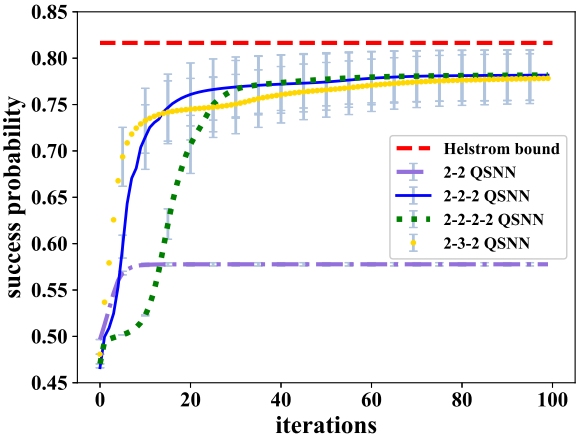

FIG. S3: The training performances of the 4 QSNNs (i.e, 2-2, 2-2-2, 2-3-2, and 2-2-2-2 QSNN) in the discrimination of quantum state pairs $(|\psi_0\rangle, |\psi_\varphi\rangle)$. The results are given by taking the average over the success probabilities for all training sets with different φ. The red dashed line shows the average optimal theoretical value given by the minimum error discrimination. The other lines in different colors and shapes represent the average success probabilities given by the 4 QSNNs. The 2-2 QSNN performs much worse than the other three. There is little difference in the performances of the 2-2-2, 2-3-2, and 2-2-2-2 QSNN. Considering that more hidden layers or more hidden layer neurons in a network will increase the consumption of resources, the 2-2-2 network is appropriate for this task.

and the 4-2 QSNN performs poorly in identifying separate states.

In summary, the QSNN with a single hidden layer is sufficient for the tasks of our interests. In addition, it can also be found that more neurons in the hidden layer also can not improve the performance of the QSNN in the tasks described in the main text.

II. GRADIENT DESCENT

In our work, we use gradient descent to optimize the parameters θ (h_k or γ_k) in the QSNN to minimize the loss function. During the optimization, the parameters θ are updated iteratively according to

$$\theta' = \theta - \eta \frac{\partial \text{Loss}}{\partial \theta}, \tag{S1}$$

where η is the learning rate. According to the chain rule, the gradient of the loss function with respect to θ can be derived as

$$\frac{\partial \text{Loss}}{\partial \theta} = \sum_s w_s \frac{df(\rho^s_{\text{out}}, |l^s\rangle\langle l^s|)}{d\theta} = \sum_s w_s \frac{df}{d\rho^s_{\text{out}}} \frac{d\rho^s_{\text{out}}}{d\theta}, \tag{S2}$$

where f is designed according to different tasks, and $d\rho^s_{\text{out}}/d\theta$ can be derived from the master equation Eq. (1) of the main text. To solve it easier, we can rewrite it as a linear equation

$$\frac{d|\rho\rangle}{dt} = \mathcal{L}|\rho\rangle, \tag{S3}$$

where $\mathcal{L}$ is the operator corresponding to Liouvillian superoperator, and $|\rho\rangle$ is the vectorized density matrix according to the map

$$\rho = \sum_{ij} \rho_{ij} |i\rangle\langle j| \mapsto |\rho\rangle = \sum_{ij} \rho_{ij} |i\rangle \otimes |j\rangle. \tag{S4}$$

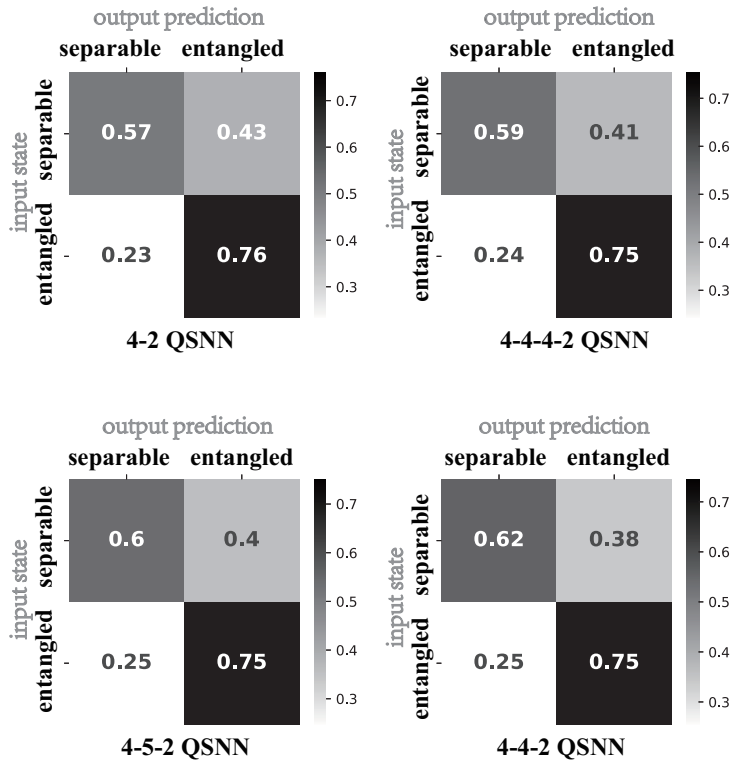

FIG. S4: The classification confusion matrices of the 4-2, 4-4-4-2, 4-5-2, and 4-4-2 QSNN. The numbers in a classification confusion matrix are the mean success and error probabilities of the QSNN classifying the 49 samples represented by Eq. (12) (main text) with $p \in \{0.02 \cdot n\}_{n=1}^{49}$. Compared to the 4-4-2 QSNN used in the main text, the performances of the 4-4-4-2 and 4-5-2 QSNN are not greatly improved, and the 4-2 QSNN performs poorly in identifying separate states.

With this representation, the operator $L(A)$ corresponding to the left multiplication of a matrix A can be calculated as

$$\begin{aligned} A\rho = \sum_{ij} \rho_{ij} A|i\rangle\langle j| \mapsto & \sum_{ij} \rho_{ij} (A|i\rangle) \otimes |j\rangle \\ & = \sum_{ij} \rho_{ij} (A \otimes I)(|i\rangle \otimes |j\rangle) \\ & = (A \otimes I)|\rho\rangle = L(A)|\rho\rangle, \end{aligned} \tag{S5}$$

which indicates that $L(A) = A \otimes I$. Similarly, the operator according to the right multiplication is $R(A) = I \otimes A^T$. With $L(A) = A \otimes I$ and $R(A) = I \otimes A^T$, the operator $\mathcal{L}$ can be calculated explicitly as

$$\begin{aligned} \mathcal{L}(\vec{h}, \vec{\gamma}) = & -i(H \otimes I - I \otimes H^T) \\ & + \sum_k \left[L_k \otimes L_k^* - \frac{1}{2}(L_k^\dagger L_k) \otimes I - \frac{1}{2} I \otimes (L_k^T L_k^*) \right]. \end{aligned} \tag{S6}$$

Here, A^*, A^T, and $A^\dagger$ denote, respectively, the complex conjugate, the transpose, and the adjoint of A. The final state of the QSNN corresponding to the s-th sample can be given by solving Eq. (S3) as

$$|\rho_{\text{out}}^s\rangle = e^{\mathcal{L}(\vec{h},\vec{\gamma})^T} |\rho_{\text{in}}^s\rangle. \tag{S7}$$

The gradient can be given as

$$\begin{aligned}\frac{\partial|\rho_{\text{out}}^s\rangle}{\partial\theta} &= \frac{\partial e^{\mathcal{L}(\vec{h},\vec{\gamma})T}}{\partial\theta}|\rho_{\text{in}}^s\rangle \\ &= \left\{\int_0^T \left[e^{\mathcal{L}(T-t)}\frac{\partial\mathcal{L}(\vec{h},\vec{\gamma})}{\partial\theta}e^{\mathcal{L}t}\right]dt\right\}|\rho_{\text{in}}^s\rangle.\end{aligned} \tag{S8}$$

The partial deviations of the operator $\mathcal{L}$ with respect to parameters can be obtained from Eq. (S6) easily as

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial h_k} &= -i(\mu_k\otimes I - I\otimes\mu_k^T) \\ \frac{\partial\mathcal{L}}{\partial\gamma_k} &= 2\gamma_k\left[\nu_k\otimes\nu_k^* - \frac{1}{2}(\nu_k^\dagger\nu_k)\otimes I - \frac{1}{2}I\otimes(\nu_k^T\nu_k^*)\right],\end{aligned} \tag{S9}$$

where $\mu_k = \frac{\partial H}{\partial h_k}$ and $\nu_k = \frac{\partial L_k}{\partial\gamma_k}$.

III. MIXED STATE DISCRIMINATION

In this part, we investigate the performance of the QSNN in discriminating two mixed quantum states. The training set $\{(\rho_{\theta_1,\varphi_1,r}, state1), (\rho_{\theta_2,\varphi_2,r}, state2)\}$ contains two mixed states, and each of them can be given as

$$\rho_{\theta,\varphi,r} = \frac{I}{2} + \frac{r}{2}(\sin\theta\cos\varphi\hat{\sigma_x} + \sin\theta\sin\varphi\hat{\sigma_y} + \cos\theta\hat{\sigma_z}). \tag{S10}$$

$I, \hat{\sigma_x}, \hat{\sigma_y}, \hat{\sigma_z}$ are Pauli matrices. We randomly choose 100 states pairs $(\rho_{\theta_1,\varphi_1,r}, \rho_{\theta_2,\varphi_2,r})$ on the sphere surface of radius r. We show the numerical result in FIG. S5. For each value of r, the optimal success probability of the QSNN for each training set is shown as an orange circle, and the mean of them is shown as a blue cross. The average values of the Helstrom bound are shown as red squares. We can see that the performance of the QSNN is close to the Helstrom bound. With the decrease of r, the quantum states are harder to be discriminated against, which is intuitive because the distance between two states on a smaller sphere is smaller.

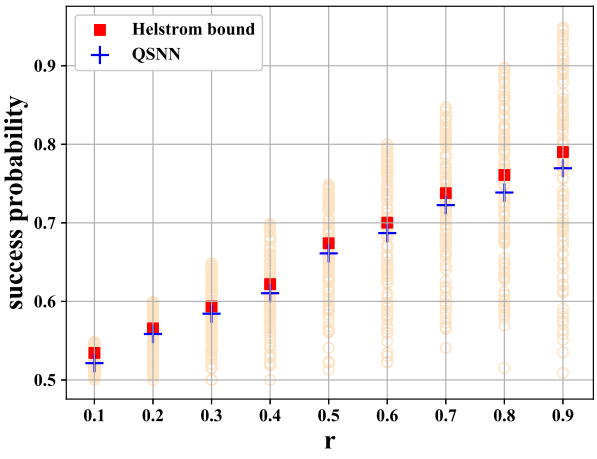

FIG. S5: The blue crosses represent the average optimal success probabilities of the QSNN in mixed state discrimination, which are compared with the optimal theoretical values (red squares). The quantum state pairs are randomly sampled from the sphere surface of radius r, and the QSNN's optimal success probability for each pair is shown as an orange circle. The performance of the QSNN is close to the ideal value given by the ME discrimination.

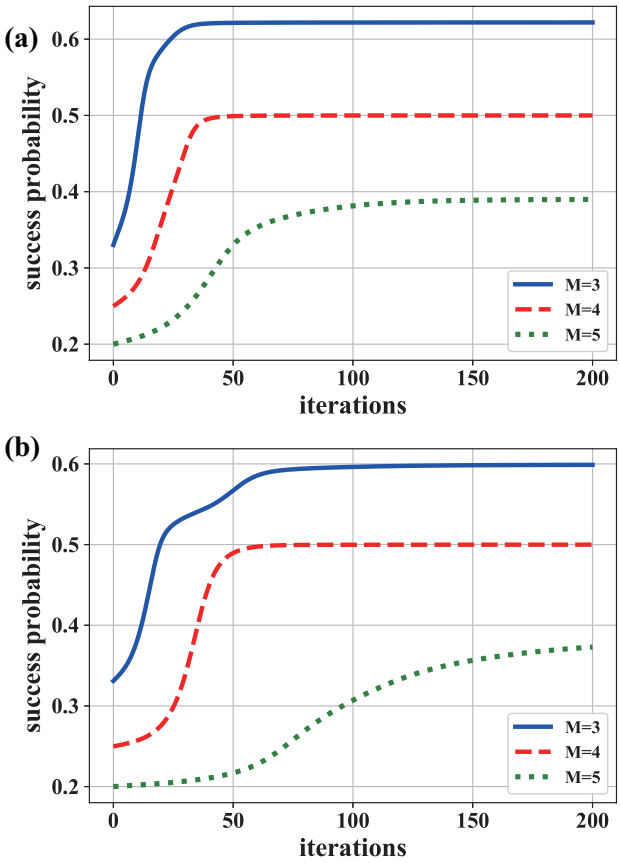

FIG. S6: We train the QSNN to discriminate M pure states in the real vector space (see a) or in the complex space (see b). It is a plot of the success probability of the QSNN with respect to the number of iterations used in the training procedure. As the number of states increases, there is a certain decrease in the success probability of the QSNN. However, the success probability increases with the number of iterations used in the training procedure, so our approach based on training is somewhat helpful for multiple quantum state discrimination.

IV. MULTI-STATE DISCRIMINATION

In traditional methods, one can construct a measurement to discriminate between two quantum states. There are two strategies to design the optimal measurement, i.e., the minimum error discrimination and the unambiguous discrimination. However, in general, if the number of states to be discriminated against is greater than two, the optimal measurement is hard to be analytically developed in the fashion of traditional strategies. In opposite, our approach is not limited by the number of states theoretically.

In this section, we take 1 qubit pure state discrimination as an example to show the performance of our QSNN in discriminating multiple quantum states. The QSNN we used here consists of 2 neurons in the input layer. The 1 qubit states to be processed are encoded in the states of the 2 neurons in the input layer. The number of neurons in the output layer is equal to the number of states to be discriminated against. The QSNN contains a single hidden layer, where the number of neurons is the same as that in the output layer. The connections between neurons are similar to those shown in Fig. 1 of the main text. Namely, the Lindblad operators only connect the neurons in the adjacent layers, and the Hamiltonian couplings only exist between some neurons in the input and hidden layer.

First, we consider discriminating M pure states in the real vector space, and $\{|0\rangle, |1\rangle\}$ is an orthogonal basis of the space. The s-th state can be given as

$$|\psi_s\rangle = \cos\frac{2\pi s}{M}|0\rangle + \sin\frac{2\pi s}{M}|1\rangle, \tag{S11}$$

which corresponds to a label $l^s \in \{N-s\}_{s=1}^{M}$. After the initialization process, the QSNN evolves according to Eq. (1) of the main text, and gives the final state ρ_{out}^s. The prior probabilities w of all the states are equal, i.e., $w = \frac{1}{M}$.

Performing a projective measurement $\Omega = |l^s\rangle\langle l^s|$ on the final state ρ^s_{out} gives

$$P^s_N = \text{Tr}(\rho^s_{\text{out}}\Omega^s), \tag{S12}$$

which is the probability that the QSNN gives the label of the s-th input correctly. So, similar to Eq. (9) of the main text, the loss function can be given as the distance between 1 and the average success probability of the network, that is

$$\text{Loss} = 1 - \frac{1}{M}\sum_{s=1}^{M} P^s_N. \tag{S13}$$

When the number of states M is 3, 4, or 5, we give the plot of the average success probability with respect to the number of iterations used in the training procedure in FIG. S6(a).

Second, we consider discriminating M pure states in a complex space, where the sth state can be represented as

$$|\psi'_s\rangle = \frac{\sqrt{2}}{2}(|0\rangle + e^{\frac{i2\pi s}{M}}|1\rangle). \tag{S14}$$

We also give the plot of the average success probability of the network with respect to the number of iterations, when the number of states M is 3, 4, or 5, in FIG. S6(b).

As the number of states increases, there is a certain decrease in the success probability of the QSNN. However, the success probability increases with the number of iterations used in the training procedure, as shown in FIG. S6(a) or (b). So, our approach based on training is somewhat helpful for the multi-state discrimination that is hard to be completed by traditional strategies.

V. PARAMETERS AND HYPERPARAMETERS

In this part, we provide some training details about the tasks in the main text, i.e., the initial and final values of the parameters, and the values of hyperparameters.

First, we give the initial and final values of the parameters used in each task of the main text. In our approach, the parameters that need to be optimized are the coefficients in the Hamiltonian and Lindblad operators, namely $\vec{h}$ and $\vec{\gamma}$.

Pure states discrimination: In *Results - Quantum State Binary Discrimination* of the main text, we discriminate two pure states represented by Eq. (10) (main text) by training the QSNN with the training set $\{(|\psi_0\rangle, state1), (|\psi_\theta\rangle, state2)\}$. We separately use 12 training sets with different θ to train the QSNN. The training results are shown in Fig. 3(a) and (b) of the main text. There are 12 sets of parameters corresponding to the 12 training sets. We choose 2 sets of parameters from them randomly and show their initial and final values in FIG. S7(a). Then, we discriminate two states represented by Eq. (11) (main text) by training the QSNN with 12 training sets $\{(|\psi_0\rangle, state1), (|\psi_\varphi\rangle, state2)\}$. We separately use 12 training sets with different θ to train the QSNN. The training results are shown in Fig. 3(c) and (d) of the main text. There are also 12 sets of parameters corresponding to the 12 training sets. We choose 2 sets of parameters from them randomly and show their initial and final values in FIG. S7(b).

Mixed states discrimination: In Sec. III, we discriminate two mixed states represented by Eq. (S10) by training the QSNN with the training set $\{(\rho_{\theta_1,\varphi_1,r}, state1), (\rho_{\theta_2,\varphi_2,r}, state2)\}$. We separately use 100 training sets with different θ, φ and the same r to train the QSNNs. And the radius r can be chosen from $0.1, 0.2, \cdots, 0.9$. The result is shown in FIG. S5. There are 900 sets of parameters corresponding to the 900 random training sets. We choose 2 sets of parameters from them randomly and show their initial and final values in the FIG. S7(c).

As shown in FIG. S7(a), (b), and (c), in the state binary discrimination task, the first 5 parameters are coherent strengths h_k in the Hamiltonian, and the last 8 parameters are the dissipation rates γ_k in the Lindblad operators.

Classification of entangled and separate states: In our simulation of *Results - Classification of Entangled and Separable States* in the main text, we train the QSNN to classify an unknown set of Werner-like states represented by Eq. (12) of the main text, for the case that two local unitaries are fixed as $U_1 = \sigma_z$, $U_2 = I$. The initial and final values of the parameters are shown in FIG. S7(d), where the first 22 parameters are coherent strengths h_k in the Hamiltonian, and the last 24 parameters are the dissipation rates γ_k in the Lindblad operators.

Second, we give the values of the hyperparameters used in each task of the main text. A hyperparameter is a parameter whose value is determined before training to control the training process. One of the hyperparameters is the time T it takes for the network to complete an evolution. In our approach, the evolution time T is considered dimensionless (it is of dimension $1/\gamma$ actually, where γ is a typical value of the coupling parameters h_k in Hamiltonian

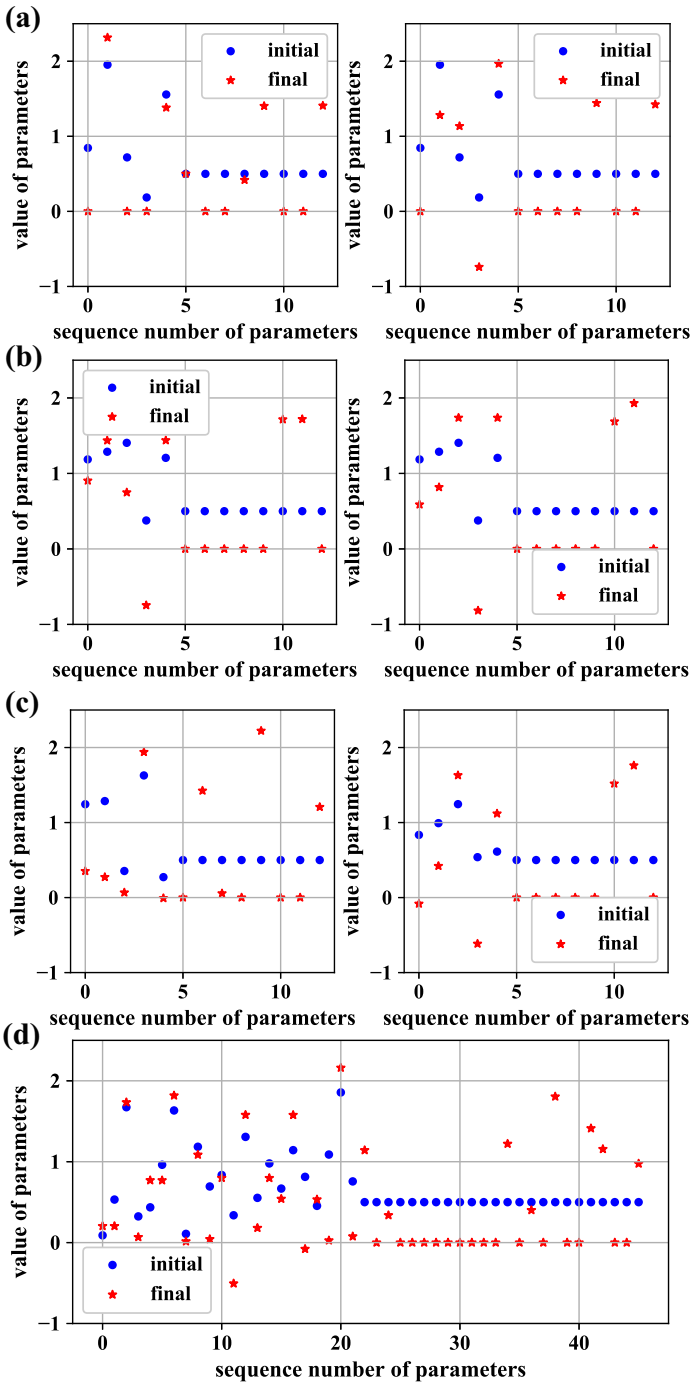

FIG. S7: A plot of the initial (blue dots) and final values (red stars) of the network parameters in the 4 tasks, i.e., (a) discriminating two pure states represented by Eq. (10) (main text), (b) discriminating two pure states represented by Eq. (11) (main text), (c) discriminating two mixed states represented by Eq. (S10), and (d) the classification task described in *Results - Classification of Entangled and Separable States* of the main text. The abscissa of the figure represents the sequence number of the parameters, and the ordinate of the figure represents the value of the parameters. In the state binary discrimination task (see a, b, and c), the first 5 parameters are coherent strengths h_k in the Hamiltonian, and the last 8 parameters are the dissipation rates γ_k in the Lindblad operators. In the classification task (see d), the first 22 parameters are coherent strengths h_k in the Hamiltonian and the last 24 parameters are the dissipation rates γ_k in the Lindblad operators.

or the dissipation rates γ_k in Lindblad operators). We fix time T to find the optimal values of parameters (h_k and γ_k) in the network. In principle, the value of time can be set to any number. However, in practical implementation, the adjustable range of parameters is often determined by the system used to simulate the network. One can adjust the time T to make optimal values of all the parameters fall within the range. In all our simulations of the tasks, we empirically fix T as 10 to make the values of parameters in the order of magnitude 1. It can be seen from FIG. S7, γT is in the order of magnitude 10. The other hyperparameter is the learning rate η. It controls the update step sizes of the parameters during training. The value of η is set as 10 in the training process shown in Fig. 3(a) (main text), and is set as 20 in the other tasks in the main text.

VI. 3-QUBIT STATE DISCRIMINATION

Here, we extend the simulation of state binary discrimination (discussed in *Result – Quantum State Binary Discrimination*) to the unknown 3-qubit states case. The example states here are chosen as two important entanglement states called GreenbergerHorneZeilinger (GHZ) state

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \tag{S15}$$

and W state

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle) \tag{S16}$$

for 3 qubits. In order to encode the 3-qubit input states, the network we used here contains 8 neurons in the input layer, 2 neurons in the hidden layer, and 2 neurons in the output layer. The training result is shown in FIG. S8. The success probability of the QSNN rises rapidly, and after several iterations, it achieves the Helstrom bound. It shows that our model also works for 3-qubit states. So, we have confidence that this model is scalable when experimental or simulation conditions allow.

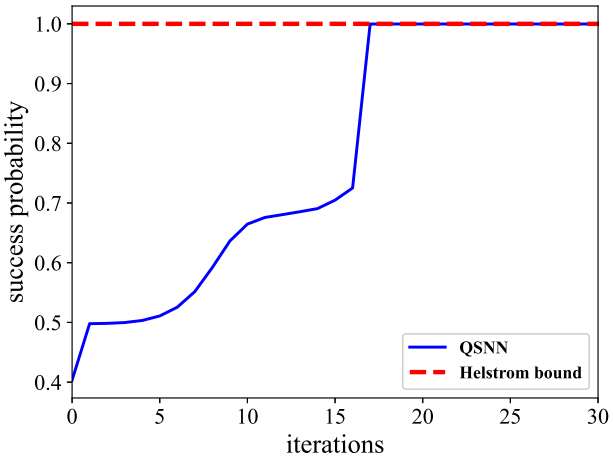

FIG. S8: The blue curve represents the success probability of the QSNN for the discrimination of the quantum state pairs $(|GHZ\rangle, |W\rangle)$. The red dashed line shows the optimal theoretical value given by the minimum error discrimination. The success probability of the QSNN rises rapidly, and after less than 20 iterations, it achieves the Helstrom bound.

VII. FLOW CHARTS OF THE CLASSIFICATION OF ENTANGLED AND SEPARABLE STATES

In this part, we show the flow charts mentioned in *Results – Classification of Entangled and Separable States* as FIG. S9.

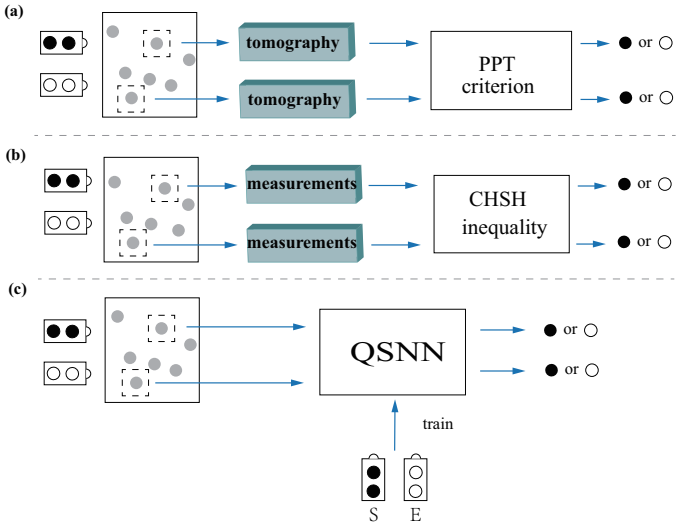

FIG. S9: The flow charts of three methods of the classification of entangled and separable states. There are two devices, one of them produces different unknown entangled quantum states (black dots), and the other produces unknown separable states (white dots). The quantum states prepared by them are then mixed in the ensemble (gray dots). The classification task is to determine which device each state is produced by. (a) Entanglement detection using the PPT criterion. For each quantum state, tomography is needed before the determination. (b) Entanglement detection using the CHSH inequality. Multiple measurements are inevitable in this approach because of the calculation of the left-hand side of the CHSH inequality. (c) QSNN classification. The QSNN can be trained by some samples respectively prepared by two devices. Then, the trained network can classify each unknown state with a single detection.