



Identifying hadronic molecular states with a neural network

Chang Chen^{1,a}, Hao Chen^{1,b}, Wen-Qi Niu^{1,c}, Han-Qing Zheng^{2,d}

¹ Department of Physics and State Key Laboratory of Nuclear Physics and Technology, Peking University, Beijing 100871, China

² College of Physics, Sichuan University, Chengdu 610065, Sichuan, China

Received: 28 June 2022 / Accepted: 30 December 2022
© The Author(s) 2023

Abstract Neural networks are trained to judge whether or not an exotic state is a hadronic molecule of a given channel according its line-shapes. This method performs well in both trainings and validation tests. As applications, it is applied to study $X(3872)$, $X(4260)$ and $Z_c(3900)$. The results show that $Z_c(3900)$ should be regarded as a \bar{D}^*D molecular state but $X(3872)$ not. As for $X(4260)$, it can not be a molecular state of $\chi_{c0}\omega$. Some discussions on $X_1(2900)$ are also provided.

1 Introduction

Exotic hadron states, refer to those hadron states that do not appear to fit with the expectations for an ordinary $q\bar{q}$ or qqq hadron in the quark model. A situation is that an exotic state may have multi-quark constituents, e.g., its valence quarks are $q\bar{q}q\bar{q}$ or $qq\bar{q}\bar{q}$. One explanation is that these states are compact or “elementary” states, where quarks and anti-quarks are building blocks, they form a compact core and interact with each other by exchanging gluons. However, since these exotic states usually appear near the two-hadron thresholds, a natural explanation is that they are hadronic molecules, which means the building blocks of these exotic states are usual hadrons and they interact with each other by exchanging color neutral forces. So, a central question of researching exotic states is to decide whether an exotic state is a molecular state or an “elementary” state (for reviews, see for example, Refs. [1, 2]).

Neural network is a machine learning algorithm which works for regression and classification problems. It has been applied to many fields in physics, like nuclear physics [3],

high energy physics experiments [4], as well as hydrodynamics [5]. Recently, it also been applied to high energy physics phenomenology to study [6, 7], for example, intermediate states in πN scattering [8], NN scattering [9] as well as extracting scattering length and effective range of exotic hadron states [10]. Inspired by these applications, an idea is using neural networks to identify hadronic molecular states.

In this work, the neural networks are trained by the invariant mass spectra generated artificially with labels “0” for molecular states and “1” for elementary states. During training, the validation tests will be done at every training epoch to monitor the network performances and to avoid overfitting. After training, the invariant mass spectrum from real experimental data will be put into trained neural network and the output which is a number from 0 to 1 will describe the possibility to be an elementary state. That means, if the output is closer to 0, then the resonance is more like a molecular state. On the contrary, if the output is closer to 1, then the resonance is more like an elementary state.

In 2003, $X(3872)$ was firstly detected by Belle [11] in the $J/\psi\pi^+\pi^-$ final state. Then, it was also observed near the \bar{D}^*D threshold [12]. Similar situation happened on $Z_c(3900)$, which is both observed in the final state of $J/\psi\pi$ [13] and \bar{D}^*D [14]. So, it is worthwhile to ask whether or not these two exotic states are hadronic molecules of \bar{D}^*D . After $X(3872)$ and $Z_c(3900)$ observed, many experiments are done to study their properties [15–17]. Furthermore, $X(4260)$ has been detected in the $J/\psi\pi\pi$ [18] and $\chi_{c0}\omega$ [19] final state. Since it is near the threshold of $\omega\chi_{c0}$, a valuable question is that wither it is a molecule of $\omega\chi_{c0}$. All these three states have data in at least two channels, especially in the final state whose threshold is near the resonance. So, in this work, the trained neural networks are applied on these three hadronic states.

This paper is organized as follows, In Sect. 2, the method of training data generation and labelling will be introduced.

^a e-mail: 1901110078@pku.edu.cn

^b e-mail: haochen0393@pku.edu.cn (corresponding author)

^c e-mail: 1701110076@pku.edu.cn

^d e-mail: zhenghq@scu.edu.cn

In Sect. 3, the structure of neural network classifier is discussed. The results of training, validation tests, as well as applications are displayed in Sect. 4. Finally, some conclusions and outlooks are given.

2 Training data generation

A neural network should be trained by many data with labels, means that people should not only put lineshapes of resonances into neural network, but also know every lineshape in the training data standing for a molecular state or an elementary state. So, criteria for the nature of resonances are needed. The pole counting rule (PCR) [20] is a convenient choice. According to PCR, the nature of an S-wave resonance is connected to the number of poles near threshold. If there is only one pole near the threshold of the couple channel, the resonance is a molecular state and if there are two poles near the threshold, the resonance is elementary. This method has been used in many works to study the nature of exotic hadron states [21–25]. Then, it demands one to choose a parametrization for resonance amplitudes, in which the parameters can be adjusted easily and then the values of parameters will make difference in pole positions as well as the lineshapes. The Flatté-like parametrization is chosen in this work to generate training data. If two final states are considered, the parametrizations of invariant mass spectra are parametrized as,

$$\frac{d\sigma_i}{d\sqrt{s}} \sim \frac{1}{\sqrt{2\pi}\Delta} \int_{\sqrt{s}-3\Delta}^{\sqrt{s}+3\Delta} \mathcal{R}_i(s') e^{-\frac{(\sqrt{s}-\sqrt{s'})^2}{2\Delta^2}} d\sqrt{s'},$$

$$\mathcal{R}_i(s) = \rho_i(s) \left| \frac{e^{i\phi_i}}{s - M^2 + iM(g_1\rho_1(s) + g_2\rho_2(s))} + bg_i \right|^2 + \rho_i(s)(a_i\sqrt{s} + b_i), \quad (1)$$

where $i = 1, 2$, M is the bare mass for the resonance, bg is coherent background contributions which takes the formula of first order polynomial of \sqrt{s} and $\rho_i(s)(a_i\sqrt{s} + b_i)$ is the incoherent background. $\phi_{1,2}$ are coherent angles. $\rho_1(s)$, $\rho_2(s)$ represent two-body phase space factor for final state 1 (FS1) and final state 2 (FS2), respectively. The FS1 stands for lower couple channel whose threshold is much lower than the resonance mass while FS2 stands for the couple channel whose threshold is near the resonance. It means that if the resonance is a hadronic molecule, it should be the composite of the two hadrons in FS2. As an example, for $Z_c(3900)$, the FS1 stands for $J/\psi\pi$ channel and FS2 stands for \bar{D}^*D channel. In Eq. (1), a Gaussian convolution is added to smooth the line-shapes of the resonances, where Δ is fixed at 3 MeV [16]. This is important for $X(3872)$ and unimportant for $Z_c(3900)$ and $X(4260)$, since the latter have much larger widths.

Table 1 Definition of Riemann sheets

	I	II	III	IV
ρ_1	+	−	−	+
ρ_2	+	+	−	−

In practice, the coupling constant g_1 , g_2 will be adjusted and then lineshapes as well as the pole positions in the complex s plane will change. Different sheets of complex s plane are defined by changing signs of phase space factors, as shown in Table 1. Generally speaking, an elementary state means it hardly couples to FS1 nor FS2, so one should keep g_1 , g_2 small to ensure there are two poles near the threshold of FS2. A molecular state indicates that it strongly couples to the second channels, so one should make $g_2 \gg g_1$ to ensure that only one pole is near the threshold of FS2. According to PCR, if the distance between two poles is larger than ~ 200 MeV, one can say there is only one nearby pole [26]. The pole positions of training data are shown in Fig. 1.

Since the parameter are determined, the training data can be generated using Eq. (1). Firstly, one should choose a proper energy range and it can cover the widths of most exotic hadron states. In this paper, the data of FS1 are from $(E_{th2} - 100)$ MeV to $(E_{th2} + 100)$ MeV and the data of FS2 are from E_{th2} to $(E_{th2} + 100)$ MeV, where E_{thi} , $i = 1, 2$ are the threshold energies of FS1 or FS2, respectively. It is stressed that the absolute value of E_{th} is unimportant since only the lineshapes will be sent into neural network. The energy resolution is fixed at 1 MeV so it can meet the demand that all the inputs sent to a special neural network should have the same size. Furthermore, one can change the energy window sizes in FS1 and FS2, but it can be shown that in most cases, the model can be well trained if the data contain the complete characteristics (the peaks). The choice of energy window sizes in this work comes from the consideration that these energy regions can cover many hadronic resonances since it will make the models more general rather than only focus on a special resonance. However, in the case of $X(3872)$, it is found that the energy window size for FS2 (\bar{D}^*D channel) cannot be taken too small (e.g., 50 MeV), otherwise the results become unstable because the noises from the data become influential here.

In order to simulate experimental data, the effects of error bars are taken into consideration. The values at every energy point calculated from Eq. (1) will be taken as the average value of Gaussian distribution, and 5, 10, 15% of their values will be taken as the standard deviations.¹ Then the Gaussian sampling with these average values and standard deviations gives the data in which the error bar effects are taken into

¹ The values of standard deviation can also be changed as long as they are not too big to affect the line-shapes seriously.

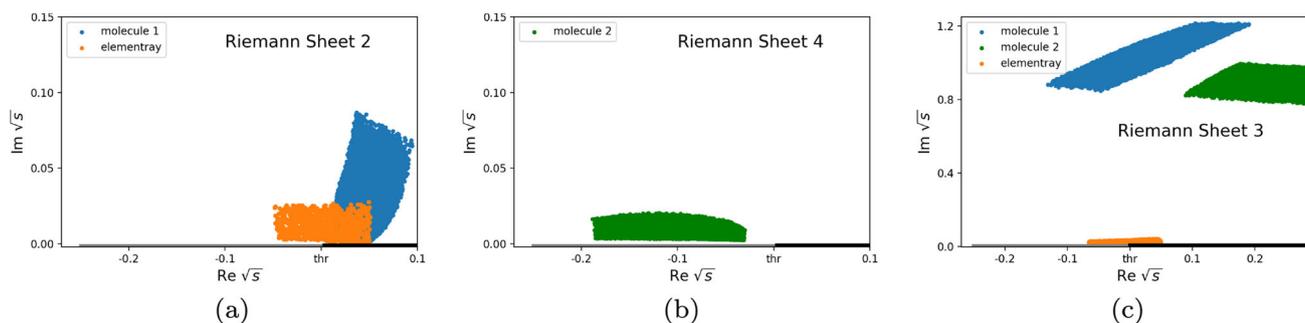


Fig. 1 Pole positions for “elementary” states and molecular states in different Riemann sheets. Here “molecule 1” means a molecular bound state and “molecule 2” means a molecular virtual state

account. At last, the normalization:

$$\{p_i \in \Omega_{\text{training}}\} = \frac{\{p_i \in \Omega_{\text{training}}\}}{\max\{p_i \in \Omega_{\text{training}}\}}, \quad (2)$$

makes points in every sample ranging from 0 to 1, where Ω_{training} is the set of points in one sample.

Using the methods introduced above, 4×10^4 groups of labeled invariant mass spectra are generated in total, including 2×10^4 elementary state with label “1” and 2×10^4 molecular states with label “0”. Each group consists of one invariant mass spectrum in FS1 and one invariant mass spectrum in FS2. In practice, 2.4×10^4 groups of invariant mass spectra are used to train the neural network and 1.6×10^4 groups are used to do validation test. In both training set and testing set, the number of elementary states and molecular states are the same.

3 Construction of neural network model

The goal of the neural network is to establish a map between the input space of invariant mass spectra and the output elementary-molecular classification space. A typical neural network consists of an input layer, some hidden layers and an output layer. In this work, the units in the input layer are numerical values of invariant mass spectra in two final states. Once the inputs are given, they will be linearly transformed with some weights and biases. Then the transformed values will be sent into the activation function, which takes the form of ReLU (rectified linear unit),

$$\text{ReLU}(x) = \max(0, x), \quad (3)$$

for units in hidden layers, and for units in output layer, it takes the form of sigmoid,

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (4)$$

The values passed through activation functions are fed to the units in next layer until meet the output layer, i.e., every unit value is obtained by combining all units in the previous

layer linearly and passing through the activation function. The reason for choosing Eq. (4) as the activation function for output layer is that it can map arbitrary value into a number ranging from 0 to 1, which represents the possibility of being an elementary state.

Once the output is given, which means the network gives the prediction of the nature of input sample, the discrepancy between the label and the prediction will be calculated using loss function. In this problem, the loss function takes the form of,

$$\text{loss} = -\frac{1}{n} \sum_{i=1}^n [q_i \log p_i + (1 - q_i) \log(1 - p_i)], \quad (5)$$

where n is the batch size which means the number of samples in each input and p_i is the output for i th sample, q_i is the label of i th sample in the training batch.

During the training, the weights and biases used to do the linear combination of unit values should be optimized to have more accurate predictions or smaller loss. Usually, the upgrading of weights and biases use gradient descent algorithm which has been integrated into many packages. The optimizer Adam² is used in this work. The process from the input to output called forward pass while the upgrading of weights and biases called backpropagation. With circulation of these two processes, the classifier will be trained accurate enough. All these algorithm are performed with PyTorch³ and more details about neural network can be found in, e.g., Ref. [27] (Fig. 2).

Generally speaking, more hidden layers or hidden layers with more units means there are more parameters including weights and biases involved in the network, so that it can give more accurate predictions with respect to the training data. But on the other hand, if there are too many parameters, the problem of overfitting will arise which means that the network has poor predictive power when it is applied to new data. So, there exist a balance between accuracy and extrap-

² See: <https://pytorch.org>.

³ <https://pytorch.org>.

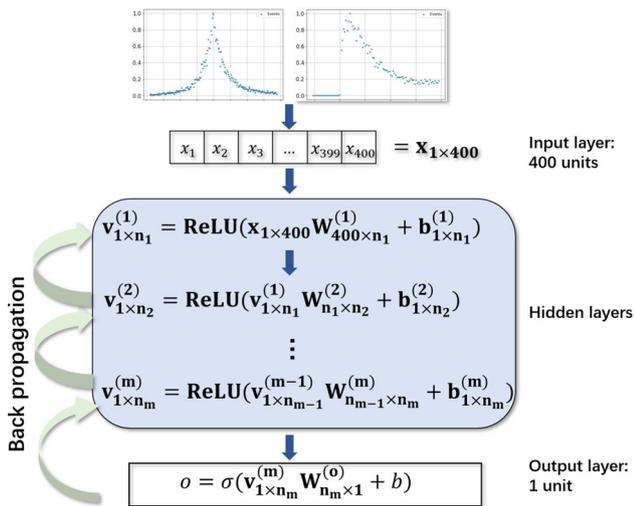


Fig. 2 The process of a sample x crossing the neural network

olation. In this work, neural networks with different number of hidden layers and different number of units in these layers are constructed.

The process of a sample crossing the neural network is displayed in Fig. (2). The sample is consisted of two invariant mass spectrum and is represented by a vector. The weight matrix and the bias vector are denoted as \mathbf{W} , \mathbf{b} , respectively. In Fig. (2), the subscription is the shape of this matrix where n_i is the number of units in the i -th hidden-layer ($i = 1, 2, \dots, m$). The superscription (i) represents that this matrix belongs to the i -th hidden-layer and the superscription (o) means that this matrix belongs to the output layer. The output o is a number ranging from 0 to 1.

4 Training, validation tests and applications

4.1 Training and validation tests

A model with structure 400-[10-10]-1 is trained, which means its input layer has 400 units, output layer has 1 unit and the two hidden layers both have 10 units. As mentioned in Sect. 2, the performances during training are monitored by validation tests. The data in validation test are different from those in training, but they are generated by the same method. Figure 3 shows the loss of training data as well as test data of the model. It is shown that after 500 training epochs, the losses in both training data as well as test data become small. The training loss usually becomes smaller with more training epochs, but it does not mean that the network will perform better and better with as many training epochs as one wants. If the test loss arrives a small value but become larger with more training epoch, then the neural network is thought to be well trained and more training epochs will cause overfit-

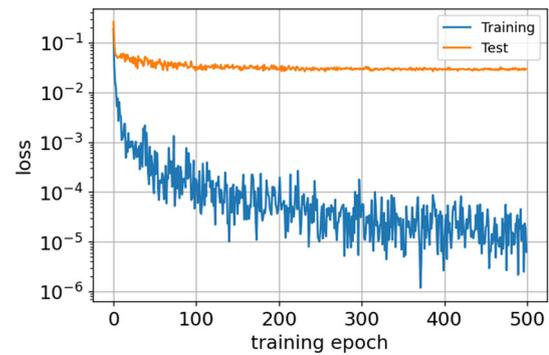


Fig. 3 The loss of training and test data

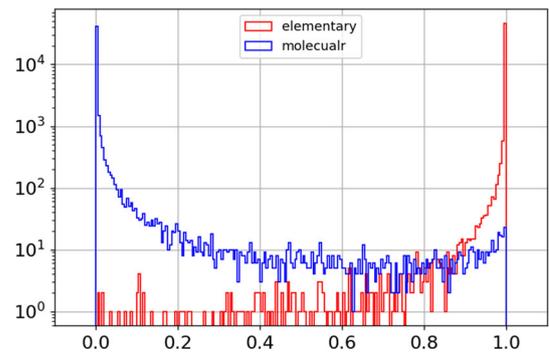


Fig. 4 Output distributions of test data

ting, as shown in Fig. 3. It looks like converging to a local minimum of parameter space in conventional fitting. So, the network with the minimum test loss will be picked up. Furthermore, the output distribution of samples in test data is shown in Fig. 4. From the figure, it is shown that the elementary states labeled by 1 and the molecular states labeled by 0 are classified well since almost all outputs for elementary states are near 1 and for molecular states are near 0.

4.2 Applications

After training and model selection, the trained models is applied to experimental data of $X(3872)$, $X(4260)$ as well as $Z_c(3900)$. For $Z_c(3900)$, the data observed in the $J/\psi\pi$ and \bar{D}^*D final states from Ref. [13] and Ref. [14], respectively are used. As for $X(3872)$, the data observed in the $J/\psi\pi\pi$ and \bar{D}^*D final states from Ref. [16] and Ref. [12] are used. For $X(4260)$, the data observed in $J/\psi\pi\pi$ [18] and $\chi_{c0}\omega$ [19] final states are used.

Before sending the experimental data into the neural network, it is noted that the energy resolution are different among experiments. So, these data should be supplied in order to have the same size as input of the neural network in which the energy resolution is fixed at 1 MeV. To be specific, if the energy resolution of experimental data is larger than 1 MeV (as shown in Fig. 5a, b, d), then the method of

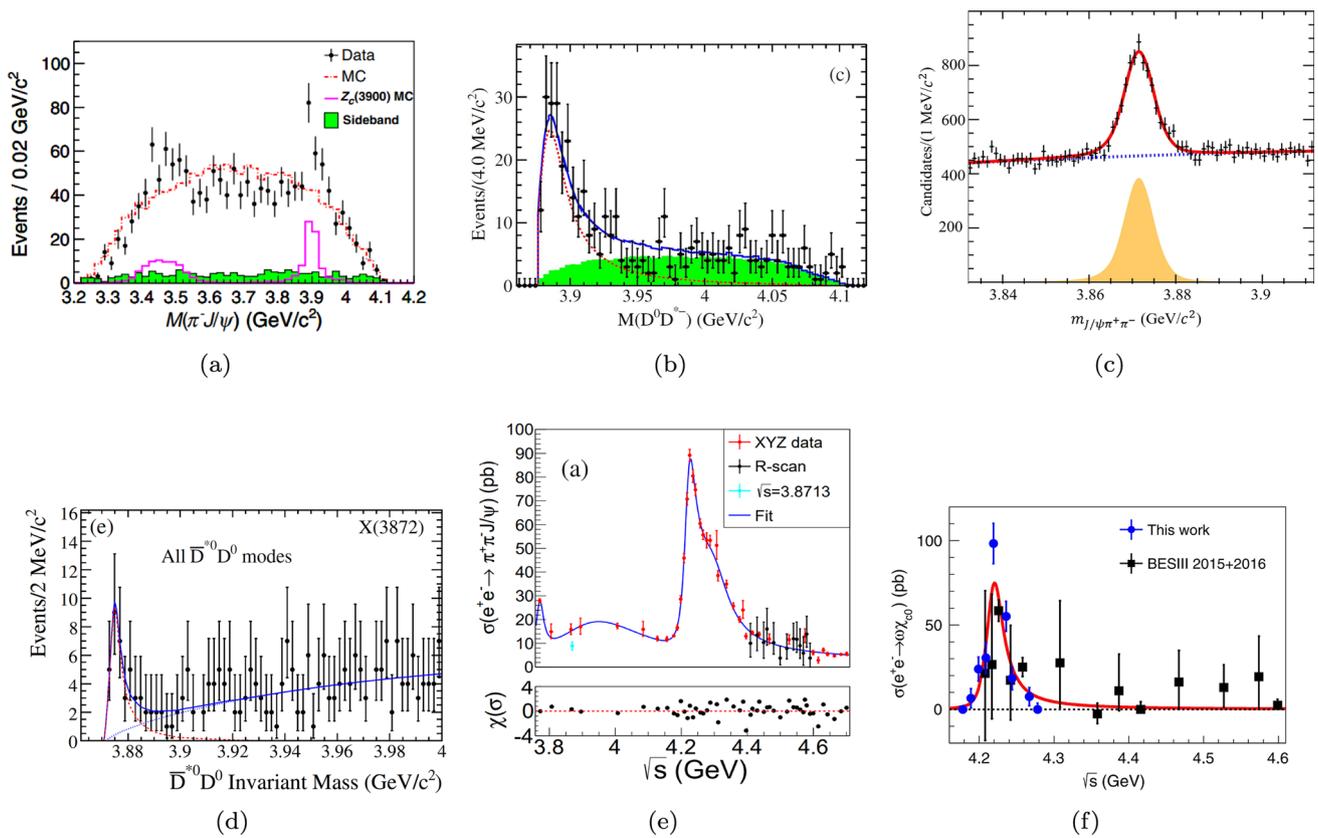


Fig. 5 Experimental data used in the application. **a** $Z_c(3900)$ data from Ref. [13], **b** $Z_c(3900)$ data from Ref. [14], **c** $X(3872)$ data from Ref. [16], **d** $X(3872)$ data from Ref. [12], **e** $X(4260)$ data from Ref. [28], **f** $X(4260)$ data from Ref. [19]

linear interpolation will be employed to supply extra points. For example, the energy resolution of Fig. 5a is 10 MeV, so 9 points will be inserted using linear interpolation between two neighbouring experimental data. Of course, the effects of error bar are also taken into consideration, the final application data are obtained by Gaussian sampling in which the central values of experimental data are regarded as average values and the error bars are regarded as standard deviations. At last, 100 invariant mass spectra are obtained for every group of experimental data using Gaussian sampling and the energy resolution is 1 MeV after linear interpolation. After these procedures, the application data should be normalized into 0 to 1 as done in Eq. (2).

Then, the application data are substituted into trained model to obtain the predictions for the nature of $Z_c(3900)$, $X(4260)$ and $X(3872)$. The predictions for these three states are shown in Fig. 6.

For $Z_c(3900)$, about all 100 samples in the application data have outputs near 0, means that it strongly couples to \bar{D}^*D channel and should be regarded as a hadronic molecule of \bar{D}^*D . This interpretation is consistent with Refs. [23,29–31]. For $X(3872)$, most outputs of 100 samples are near 1, that means the coupling between $X(3872)$ and \bar{D}^*D channel is not strong enough to dominate its production. In other

words, $X(3872)$ behaves more like an elementary state. The neural network prediction for $X(3872)$ is consistent with e.g., Refs. [21,32–34]. The predictions for $X(4260)$ are more coincident. All outputs are near 1, which means that $X(4260)$ couples to $\omega\chi_{c0}$ not strong enough and can not be regarded as a molecule of $\omega\chi_{c0}$. This prediction is consistent with e.g., Refs. [35,36]. Furthermore, the measurement of cross section $e^+e^- \rightarrow \mu^+\mu^-$ performed by BESIII [37] gives the muonic width of $X(4260)$ to be from 1.09 to 1.53 KeV which strongly indicates that $X(4260)$ has a charmonium nature [36].

Furthermore, other model structures are also constructed. For instance, models with structures 400-[10-5]-1, 400-[15-5-5]-1, 400-[20-20]-1 are trained and the outputs for $X(3872)$, $Z_c(3900)$, $X(4260)$ are unchanged.

Of course, some exotic hadron states have data only in one final state (usually only in FS1). At this time, one still can identify whether they are hadronic molecules by neural network. As an example, a network is trained by FS1 data in Eq. (1) alone. This network is used to identify the properties of $X_1(2900)$, which is regarded as a \bar{D}_1K molecule [25]. When the experimental data are sent into network, about a half samples are identified as elementary states and another are molecular states, which means the network is not good enough yet to determine its nature. But this may be caused

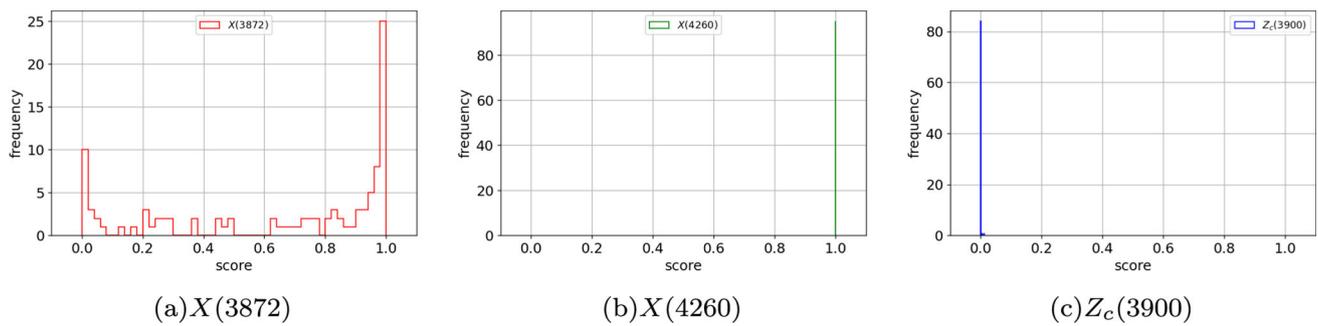


Fig. 6 Output distribution for **a** $X(3872)$, **b** $X(4260)$ and **c** $Z_c(3900)$ using different models

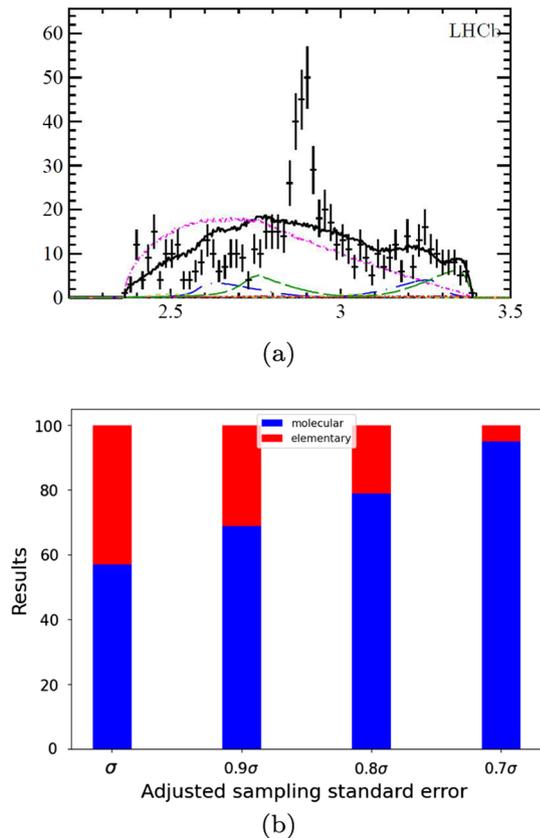


Fig. 7 **a** Experimental data of $X_1(2900)$ [38]. **b** Outputs of $X_1(2900)$. Where the σ stands for the experimental error bar

by the big error bars of the experimental data (see Fig. 7a). When generating samples by Gaussian sampling, their line-shapes can change a lot within error bars which are used as standard deviations. So, one can reduce the standard deviations in Gaussian sampling to avoid misleading. As shown in Fig. 7b, with smaller standard deviations, the outputs are converged to a molecular state, i.e., scores of the most samples are < 0.5 . That means, neural network can also identify hadronic molecules using only one invariant mass spectrum if the experimental error bars are small enough.

5 Summary and outlook

In this work, machine learning models based on neural network are developed and used to decide whether $X(3872)$, $X(4260)$ as well as $Z_c(3900)$ can be regarded as hadronic molecules of special channels. Resonance lineshape data observed in two final states are taken as input, and the output is a number which can be seen as the possibility of this resonance to be an elementary state. In other words, the resonance is more like a hadronic molecule of particles in FS2 whose threshold is near the resonance if the output is closer to 0. The well trained networks are picked up to do these tasks. The results show that $Z_c(3900)$ can be regarded a molecule of \bar{D}^*D but $X(3872)$ is not like a molecule of \bar{D}^*D . Besides, if data of $X(4260)$ in $J/\psi\pi\pi$ and $\omega\chi_{c0}$ channels are taken as input, the predictions of neural networks suggest that it is not like a molecular state of $\omega\chi_{c0}$ ⁴. These interpretations from neural networks are consistent with many previous phenomenological studies. So, the method employed in this work is practicable.

The philosophy of a neural network to do such a binary classification task is to represent the inputs as points in the parameter space and build a hypersurface to separate these points into two classes. The training process is nothing but to adjust the hypersurface to do the separation better. In other words, the output for a set of data reflects that how can the parameters take values to describe such an input. That sounds just like to do a conventional fit. In this work, the data in both two final states perform better compared to data in only one final state, it makes the classification like a joint fitting and the prediction are more reliable.

The method in this work provides a new way to identify the nature of hadron states. It is a data driven, model independent and general method. That means a trained model can be used to classify different resonances in different processes. In the future, the machine learning method can be developed

⁴ It is suggested in Ref. [39] that $X(4260)$ is a $\bar{D}D_1(2420)$ molecule. Since there is no data in this final state, we cannot study this scenario. Furthermore, Ref. [36] suggested $X(4260)$ can well be described as a 4^3S_1 and 3^3D_1 mixing state, the molecule picture is no longer appealing.

better to understand the nature of exotic states. For instance, Dalitz plots are more original data compared with invariant mass spectra and convolution neural network is powerful in image recognition. So, it can be used to classify elementary states and hadronic molecules based on Dalitz plots. It is believed that with the development of algorithms and computer technology, machine learning will become more and more popular in the field of particle physics.

Acknowledgements This work is supported in part by National Nature Science Foundations of China under Contract Number 11975028 and 10925522.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: This analysis doesn't have valuable data to upload. All the numerical results are already shown in figures].

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funded by SCOAP³. SCOAP³ supports the goals of the International Year of Basic Sciences for Sustainable Development.

References

1. A. Hosaka et al., PTEP **2016**, 062C01 (2016). [arXiv:1603.09229](https://arxiv.org/abs/1603.09229)
2. D.-L. Yao et al., Rep. Prog. Phys. **84**, 076201 (2021). [arXiv:2009.13495](https://arxiv.org/abs/2009.13495)
3. A. Boehnlein et al. (2021). [arXiv:2112.02309](https://arxiv.org/abs/2112.02309)
4. A. Gianelle et al. (2022). [arXiv:2202.13943](https://arxiv.org/abs/2202.13943)
5. K. Taradiy et al. (2021). [arXiv:2106.02841](https://arxiv.org/abs/2106.02841)
6. M. Albaladejo et al. (JPAC). (2021). [arXiv:2112.13436](https://arxiv.org/abs/2112.13436)
7. L. Ng et al. (JPAC), Phys. Rev. D **105**, L091501 (2022). [arXiv:2110.13742](https://arxiv.org/abs/2110.13742)
8. D. Sombillo et al., Phys. Rev. D (2021). <https://doi.org/10.1103/PhysRevD.104.036001>. (ISSN 2470-0029)
9. D. Sombillo et al., Few-Body Syst. (2021). <https://doi.org/10.1007/s00601-021-01642-z>. (ISSN 1432-5411)
10. J. Liu et al. (2022). [arXiv:2202.04929](https://arxiv.org/abs/2202.04929)
11. S.-K. Choi et al. (Belle Collaboration), Phys. Rev. Lett. **91**, 262001 (2003). <https://doi.org/10.1103/PhysRevLett.91.262001>
12. B. Aubert et al. (BABAR Collaboration), Phys. Rev. D **77**, 011102 (2008). <https://doi.org/10.1103/PhysRevD.77.011102>
13. M. Ablikim et al. (BESIII Collaboration), Phys. Rev. Lett. **110**, 252001 (2013). <https://doi.org/10.1103/PhysRevLett.110.252001>
14. M. Ablikim et al. (BESIII Collaboration), Phys. Rev. D **92**, 092006 (2015). <https://doi.org/10.1103/PhysRevD.92.092006>
15. R. Aaij et al. (LHCb Collaboration), Phys. Rev. Lett. **110**, 222001 (2013). <https://doi.org/10.1103/PhysRevLett.110.222001>
16. R. Aaij et al. (LHCb Collaboration), Phys. Rev. D **102**, 092005 (2020). <https://doi.org/10.1103/PhysRevD.102.092005>
17. M. Ablikim et al. (BESIII Collaboration), Phys. Rev. Lett. **115**, 112003 (2015). <https://doi.org/10.1103/PhysRevLett.115.112003>
18. M. Ablikim et al. (BESIII Collaboration), Phys. Rev. Lett. **118**, 092001 (2017). <https://doi.org/10.1103/PhysRevLett.118.092001>
19. M. Ablikim et al. (BESIII Collaboration), Phys. Rev. D **99**, 091103 (2019). <https://doi.org/10.1103/PhysRevD.99.091103>
20. D. Morgan, Nucl. Phys. A **543**, 632 (1992)
21. O. Zhang, C. Meng, H.Q. Zheng, Phys. Lett. B **680**, 453 (2009)
22. Q.-F. Cao et al., Phys. Rev. D **100**, 054040 (2019)
23. Q.-R. Gong et al., Phys. Rev. D **94**, 114019 (2016)
24. Q.-F. Cao et al., Chin. Phys. C **45**, 103102 (2021)
25. H. Chen, H.-R. Qi, H.-Q. Zheng, Eur. Phys. J. C **81**, 812 (2021)
26. H. Chen, H.-R. Qi, in 10th International workshop on Chiral Dynamics. (2022). [arXiv:2202.10736](https://arxiv.org/abs/2202.10736)
27. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, New York, 2009). (ISBN 978-0-387-84858-7)
28. M. Ablikim et al. (BESIII). (2022). [arXiv: 2206.08554](https://arxiv.org/abs/2206.08554)
29. E. Wilbring, H.-W. Hammer, U.-G. Meißner, Phys. Lett. B **726**, 326 (2013). (ISSN 0370-2693)
30. M. Albaladejo et al., Phys. Lett. B **755**, 337 (2016)
31. M. Karliner, J.L. Rosner, Phys. Rev. Lett. (2015). <https://doi.org/10.1103/PhysRevLett.115.122001>. (ISSN 1079-7114)
32. J. Ferretti, G. Galatà, E. Santopinto, Phys. Rev. C **88**, 015207 (2013)
33. C. Meng et al., Phys. Rev. D **92**, 034020 (2015)
34. Y. Tan, J.-L. Ping, Phys. Rev. D **100**, 034022 (2019)
35. L. Y. Dai et al., Phys. Rev. D **92**, 014020 (2015). <https://doi.org/10.1103/PhysRevD.92.014020>
36. Q.-F. Cao et al., Eur. Phys. J. C **81**, 83 (2021)
37. Phys. Rev. D **102**, 112009 (2020). [arXiv:2007.12872](https://arxiv.org/abs/2007.12872)
38. R. Aaij et al. (LHCb), Phys. Rev. D **102**, 112003 (2020b). [arXiv:2009.00026](https://arxiv.org/abs/2009.00026)
39. M. Cleven et al., Phys. Rev. D **90**, 074039 (2014). [arXiv:1310.2190](https://arxiv.org/abs/1310.2190)