Classifying the pole of an amplitude using a deep neural network

Denny Lane B. Sombillo,[1,2,*] Yoichi Ikeda,[3] Toru Sato,[2] and Atsushi Hosaka[2]

[1]*National Institute of Physics, University of the Philippines Diliman, Quezon City 1101, Philippines*
[2]*Research Center for Nuclear Physics (RCNP), Osaka University, Osaka 567-0047, Japan*
[3]*Department of Physics, Kyushu University, Fukuoka 819-0395, Japan*

(Received 26 March 2020; accepted 9 July 2020; published 28 July 2020)

Most of the exotic resonances observed in the past decade appear as a peak structure near some threshold. These near-threshold phenomena can be interpreted as genuine resonant states or enhanced threshold cusps. Apparently, there is no straightforward way of distinguishing the two structures. In this work, we employ the strength of deep feed-forward neural network in classifying objects with almost similar features. We construct a neural network model with scattering amplitude as input and the nature of a pole causing the enhancement as output. The training data is generated by an S-matrix satisfying the unitarity and analyticity requirements. Using the separable potential model, we generate a validation data set to measure the network's predictive power. We find that our trained neural network model gives high accuracy when the cutoff parameter of the validation data is within 400–800 MeV. As a final test, we use the Nijmegen partial wave and potential models for nucleon-nucleon scattering and show that the network gives the correct nature of the pole.

DOI: 10.1103/PhysRevD.102.016024

I. INTRODUCTION

Renewed interest in hadron spectroscopy started after the discovery of $X(3872)$ in 2003 [1]. Since then, several candidates of nonstandard exotic hadrons are proposed. One common feature of these phenomena is that they manifest as sharp peak structure near some threshold [2]. The proximity of an enhancement to the threshold introduces several possible nature of peak's origin. One of the appealing possibilities is a weakly bounded hadronic molecule composed of two hadrons [3,4] which can be associated to the presence of a pole near the two-particle threshold. Other possibilities are purely kinematical in nature such as cusps or triangle singularities [5]. On one hand, threshold cusp is always present in s-wave scattering whenever an inelastic channel opens. However, it has been shown in [6–9] that threshold cusp can only produce a significant enhancement provided that there is some near-threshold pole even if it is not located in the relevant region of unphysical sheet. On the other hand, triangle singularity does not need nearby pole to produce a pronounced enhancement but instead requires that intermediate particles be simultaneously on-shell [5,10–12].

The purpose of this paper is to address the origin of the sharp peak observed around the threshold of two-body hadron scattering problems. We specifically focus on the case where a near-threshold pole causes the peak structure and attempt to identify its nature, i.e., whether it is a bound, resonance, or virtual state pole. Until now, there has not been a method to distinguish the pole origin of peak structure around the threshold. In general, this is a difficult program because of the limited resolution of experimental data.

Here, we treat the identification of the nature of the pole causing the enhancement as a classification task [13] and solve it using supervised machine learning. The machine learning technique is ubiquitous even in physical sciences [14] and it is well known that a deep neural network excels in solving a classification task. In this work we demonstrate how a deep neural network can be applied to identify the pole origin of cross section enhancement. This includes defining the appropriate input-output data, setting up the network architecture, and generating the training dataset. As a first effort to apply deep learning in the classification of pole causing a cross section enhancement, we only consider here the single-channel scattering.

This paper is organized as follows. In Sec. II we give a short background on how a neural network works. One of the crucial parts of deep learning is the preparation of the dataset. In Sec. III we describe how the training dataset is generated using the general properties of S-matrix. The performance of our neural network model using the training dataset is discussed in Sec. IV. In Sec. V we explore the

[*] sombillo@rcnp.osaka-u.ac.jp

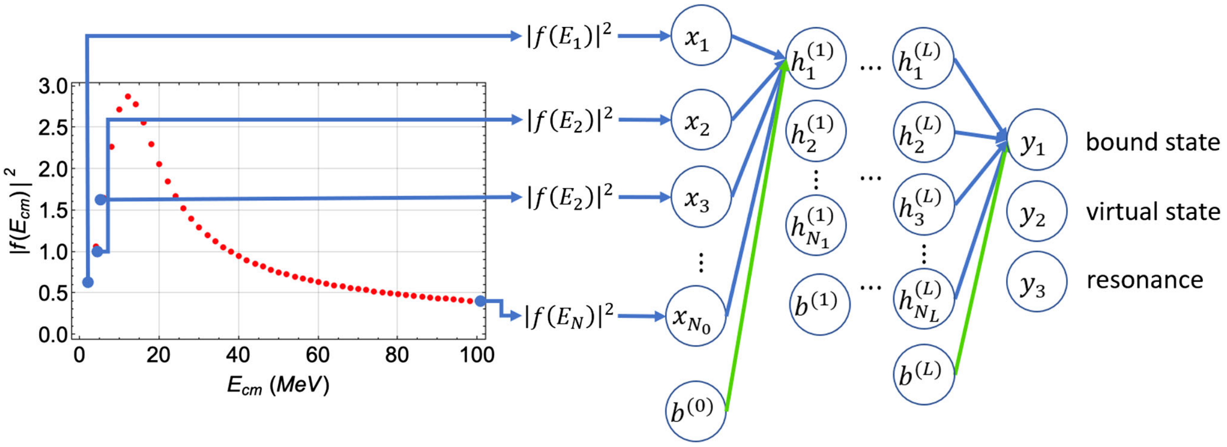

FIG. 1. Schematic of deep neural network for S-matrix pole classification.

applicability of our trained network using a separable potential model to generate a validation dataset. We also use the partial wave and potential models of Nijmegen group as a final test in the same section. Finally, we formulate our conclusion in Sec. VI.

II. DEEP NEURAL NETWORK FOR POLE CLASSIFICATION

We briefly review the basic operation in deep learning [15] and discuss how it can be applied to pole classification problem. A neural network consists of an input, hidden layers and an output layer where each layer contains certain number of nodes. We use the term deep neural network for architectures having more than one hidden layers. Figure 1 shows the deep neural network set-up that we used in this study. The nodes x_i's in the input layer contain numerical values describing certain features of the input data while nodes that are not in the input layer are equipped with activation functions with range (0,1) or $(0,\infty)$, whichever is applicable. The nodes in $(\ell-1)$th layer are sent to each ℓth layer node by putting them in a linear combination

$$z_i^{(\ell)} = \Sigma_j w_{ij}^{(\ell)} h_j^{(\ell-1)} + b^{(\ell-1)} \tag{1}$$

where $z_i^{(\ell)}$ is the ith node preactivation value in the ℓth layer, $h_j^{(\ell-1)}$ is the jth node post-activation value of the $(\ell-1)$th layer, w_{ij}^{ℓ} is the weight connecting jth node of $(\ell-1)$th layer to ith node of ℓth layer and $b^{(\ell-1)}$ is the bias in $(\ell-1)$th layer. In this notation, input nodes are represented as $x_i = h_i^{(0)}$. The pre-activation value $z_i^{(\ell)}$ is fed to the activation function to get the node's postactivation value:

$$h_i^{(\ell)} = \sigma(z_i^{(\ell)}). \tag{2}$$

This arrangement of layers and nodes together with the choice of activation functions allows the neural network to build a nonlinear mapping of input vector **x** to output vector **y**.

The goal of deep learning is to find an optimal mapping between **x** and **y**. To do this, one has to prepare a training dataset containing inputs with known outputs. Initially, some random weights and biases are assigned to the neural network. Then we perform a forward pass, i.e., we feed all the training inputs and let the network calculate all the outputs. Now, the average difference between true output and the network's output define the cost function $C(\hat{w},\vec{b})$ where $\hat{w}$ and $\vec{b}$ are the initial weight matrix and bias vector, respectively. The weights and biases are updated using the gradient descent method via backpropagation [16]. One forward pass together with one backpropagation of the entire training dataset comprise one epoch of training. Several epochs are normally executed to update the weights and biases until the cost function reached its global minimum. The neural network architecture with its updated weights and biases correspond to the optimal map that we seek.

In this study, we construct a deep neural network with the cross section of two-body scattering, $|f(E_{cm})|^2$, on a discretized center-of-mass energy axis [0, 100 MeV] with 0.5 MeV spacing as input and a vector with three elements as output. One can use smaller spacing but this will increase the number of input nodes and may result in slow cost function convergence. Similarly, taking larger spacing with smaller number of input nodes will most likely converge fast to a higher cost function minimum. The chosen 0.5 MeV provides an optimal spacing for the current study. Now, the output nodes correspond to three distinct pole classifications, i.e., bound state, virtual state or resonance as shown in Fig. 1. The classification of pole is described as follows. Suppose p_0 represents the pole position on the complex momentum plane $\mathbb{C}$, then we say that it is a bound state pole if p_0 is positive pure imaginary. If $\mathrm{Im}p_0 < 0$ and $|\mathrm{Im}p_0| > |\mathrm{Re}p_0|$, then p_0 is a virtual state pole. Otherwise, if $|\mathrm{Im}p_0| < |\mathrm{Re}p_0|$ we call it resonance (see the Appendix of [17] for detailed explanation).

To obtain the optimal values of weights and biases, the network must be trained using a dataset of cross section with known enhancement origin. This will be explained in the next section.

III. DATASET TO TRAIN DEEP NEURAL NETWORK

A. General properties of S-matrix

Ideally, a reliable neural network model that can distinguish the nature of the pole responsible for the cross section enhancement must be optimized using a training dataset generated from an exact S-matrix. However, such an S-matrix cannot be derived from the fundamental theory of strong interaction QCD for hadrons due to its nonperturbative nature. In such a situation, we can still deduce the general form of S-matrix using the analyticity and unitarity requirements [18–20].

Consider the s-wave scattering of two particles with mass m_1 and m_2, reduced mass $\mu = m_1 m_2/(m_1 + m_2)$ and relative momentum magnitude of p. The S-matrix can be parametrized as

$$S(p) = \frac{1 - i\mu p K(p)}{1 + i\mu p K(p)} \tag{3}$$

satisfying unitarity provided that $K(p)$ is the real-valued K-matrix [21–23]. At energies near the location of the K-matrix pole M', we can write $K = g'^2/(E - M') + c$ where $E = E_1 + E_2$ with E_i as the energy of particle m_i and g', c are reals. Analyticity and $K(-p) = K(p)$ are satisfied in the nonrelativistic case, i.e., $E = p^2/(2\mu)$, by the parametrization

$$K(p) = \frac{g^2}{p^2 - M} + c \tag{4}$$

where $g^2 = 2\mu g'^2$ and $M = 2\mu(M' - m_1 - m_2)$. From the S-matrix in (3), one can obtain the partial wave amplitude using the relation

$$S(p) = 1 + 2ipf(p). \tag{5}$$

Consider now how the K-matrix parameters dictate the singularities of S-matrix in (3). If we substitute $K(p)$ into $S(p)$, we get

$$S(p) = -\frac{i\mu c p^3 - p^2 + i\mu(g^2 - cM)p + M}{i\mu c p^3 + p^2 + i\mu(g^2 - cM)p - M} \tag{6}$$

and the pole position is obtained from

$$i\mu c p^3 + p^2 + i\mu(g^2 - cM)p - M = 0. \tag{7}$$

Taking the complex conjugate of (7) and knowing that μ, g^2 and M are reals, we can recover the same equation as that for p, i.e., $-p^*$ satisfies the same cubic equation. This means that the denominator of $S(p)$ in (6) contains a factor $(p + i\beta)^2 - \alpha^2$ which gives a conjugate pair of poles with real α, β. The third unpaired solution to (7) must have the property $p = -p^*$. This can only be true if p is pure imaginary. In fact it is possible that all the solutions of (7) are pure imaginary. It follows that we can write (6) in factored form as

$$S(p) = \left(-\frac{p + i\gamma}{p - i\gamma}\right)\left[\frac{(p - i\beta)^2 - \alpha^2}{(p + i\beta)^2 - \alpha^2}\right] \tag{8}$$

where α, β, γ are real numbers that are related to g^2, M and c parameters.

For $c = 0$ we only have a pair of conjugate poles given by

$$p_0 = -i\frac{\mu g^2}{2} \pm \sqrt{M - \left(\frac{\mu g^2}{2}\right)^2} \tag{9}$$

and we readily identified $\beta = \mu g^2/2$ and $\alpha = \sqrt{M - (\mu g^2/2)^2}$. Note that $\beta > 0$ is required to avoid having S-matrix poles on the upper half momentum plane (other than the imaginary axis), otherwise causality is violated [18,24]. For $c \neq 0$, a third imaginary pole $i\gamma$ appears and α, β are modified according to:

$$\begin{aligned} \alpha^2 &= \xi M - \beta^2 \\ \beta &= \frac{\mu g^2}{2}\xi\left[1 + (1 - \xi)\frac{Mc}{g^2}\right] \\ \gamma &= \frac{1}{\xi\mu c}. \end{aligned} \tag{10}$$

These are obtained by comparing the expansion in the denominator of (8) with that of (7). A dimensionless quantity ξ is introduced to facilitate the comparison and for given values of μ, g^2 and M, ξ is an implicit function of c given by

$$1 = \xi + c\mu^2 g^2 \xi^2\left[1 - (1 - \xi)\frac{cM}{g^2}\right] \tag{11}$$

with $\xi \to 1$ as $c \to 0$ or $c \to \pm\infty$ (see Fig. 2).

The bounded $\xi = \xi(c)$ implies that the third pole $i\gamma$ will originate from $\pm\infty i$ as c becomes nonzero. For $c > 0$, we can generate a simple pole at $p_0 = i\gamma$ on the upper half momentum plane and if we let $c \to +\infty$, this pole gets very close to the threshold. This corresponds to a bound state in accordance to the completeness relation in [25]. Now, as we vary c from zero to some negative value, the poles redistribute themselves as shown in Fig. 3. Here, we see an instance when all the three poles are pure imaginary and at some finite values of c, two of the poles will merge and

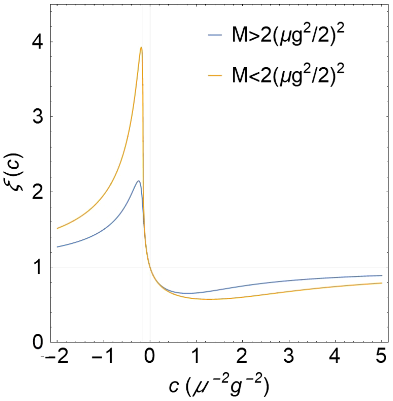

FIG. 2. Behavior of $\xi = \xi(c)$.

turn into conjugate pair as seen in Fig. 3(a) and (b). The merging of poles happens at some $c < 0$ when the slope of ξ becomes infinite as shown in Fig. 2. This demonstrates that the constant term in (4) is capable of generating S-matrix pole and should not be treated as background (see also [26]).

The conjugate pair of poles in (8) will always have $\beta > 0$ for all values of c. For $c \to 0$, ξ approaches unity and we recover (9) with $\beta > 0$. Also, as $c \to +\infty$, (10) gives a positive β since $0 < \xi < 1$. Finally, if $c < 0$ we see from Fig. 2 that $(1 - \xi) < 0$ and this still gives a positive β demonstrating that causality is not violated for all values of c.

The form of S-matrix in (8) and its relation to K-matrix in (4) allows us to identify the parenthetical factor as the generator of pure imaginary momentum pole and the square-bracket factor as the generator of conjugate poles. To avoid ambiguity in the classification it is more plausible to separate these two factors. That is, the first factor will only be used to generate the bound-virtual dataset while the second factor will be used to generate conjugate virtual-resonance dataset. The two datasets will be combined as a single classification dataset before we use it to optimize the parameters of our neural network. This will suffice to assign three distinct outputs in our neural network, i.e., bound, virtual and resonance. Note that one can also use the combined form in (8) but a "bound with resonance" must be added to the output entry. This additional category is not yet relevant in the current study.

B. Bound state and virtual state

Let us first consider the threshold-enhancement caused by a shallow bound state or a virtual state in s-wave amplitude. From previous discussion, we learned that the first factor in (8) can be used to generate a near-threshold bound or virtual state pole. A closer look will reveal that this gives an identical cross section whatever the sign of γ. That is, with $S(p) = -(p + i\gamma)/(p - i\gamma)$ we get $|f(p)|^2 = 1/(p^2 + \gamma^2)$ and there is no way to distinguish between virtual and bound state enhancements. In general, there is background contribution in addition to the pole part of the S-matrix, making it possible to distinguish the two enhancements. Thus, it is imperative to include a background to the S-matrix parametrization for the bound-virtual dataset, i.e.,

$$S(p) = e^{2i\delta_{bg}(p)}\left(-\frac{p + i\gamma}{p - i\gamma}\right), \tag{12}$$

where $\delta_{bg}(p)$ is the background phase.

The form of $\delta_{bg}(p)$ is restricted by unitarity and analyticity requirements. First, unitarity dictates that $\delta_{bg}(p)$ be a real-valued function for real momentum p. Second, analyticity requires that there be no poles in the analytically continued $e^{2i\delta_{bg}(p)}$ on the upper-half momentum plane and that the reflection principle be satisfied. Here, we introduce the background phase shift given by

$$\delta_{bg}(p) = \eta \tan^{-1}\left(\frac{p}{\Lambda_{bg}}\right). \tag{13}$$

where η is a real parameter and $\Lambda_{bg} > 0$ is the training S-matrix cutoff parameter. If we let $\eta < 0$, (13) reduces to a repulsive hard-core type background used in [27] with $-\eta/\Lambda_{bg}$ as the core radius if p is near the threshold. Also, (13) can simulate the left-hand cut both in the physical

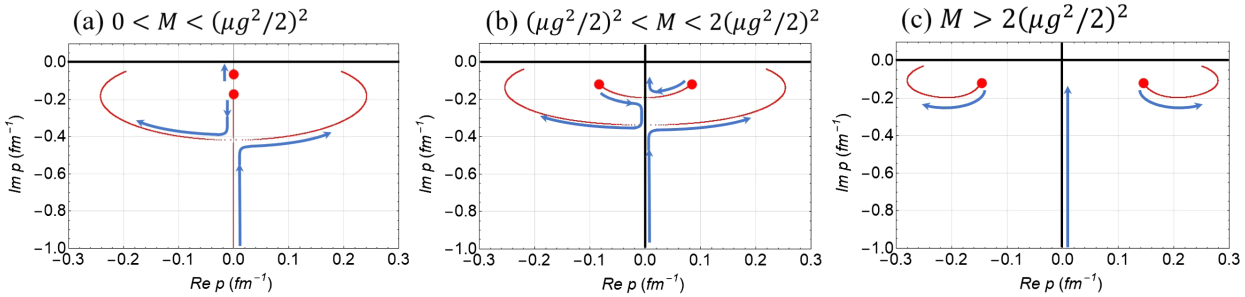

FIG. 3. Configuration of S-matrix pole as c is varied from zero to some negative value starting with (a) a pair of virtual states, (b) virtual state with widths and (c) resonance. The red dots represents the pole positions when $c = 0$, the red line shows the trajectory and the blue line shows the direction of pole motion as c becomes negative.

TABLE I. Dataset generated in this study.

Dataset	η	Λ_{bg} in (MeV)	Size
Set 0	$[-4,-3,-2,-1]$	(100,1100)	2×10^6
Set 1	$[-4,-3,-2,-1]$	(200,1200)	4×10^6
Set 2	$[-4,-3,-1,0]$	(200,1200)	4×10^6

and unphysical sheet even in the nonrelativistic case since the analytically continued $\tan^{-1}(p/\Lambda_{bg})$ has branch cuts in $\mathbb{C}$ along the strip $(-i\infty,-i\Lambda_{bg})\cup(i\Lambda_{bg},i\infty)$ [28].

Using the parameters of background phase in (13), we prepared three training datasets that will be used in the subsequent numerical experiments. These are shown in Table I. The purpose of each dataset is described as follows: set 0 is used to experiment with different neural network architecture in Sec. IV while set 1 and set 2 are used to train two deep neural network models for numerical experiments in Sec. V. For each dataset, we choose negative values for η to mimic a repulsive background since the attractive case is already taken care of by the pole factor in (12). It suffices to use the integer values shown in the second column of Table I for the purpose of this study. Then, for each η we generate 500 random values of Λ_{bg} in the range specified in third column of Table I. The size of each dataset is determined by the parameters of the pole part.

The parameters for pole part of bound-virtual in (12) is generated as follows. For each η and Λ_{bg} in Table I, we choose 1,000 random values of γ in the interval $(-0.9\Lambda_{bg},-10\text{ MeV})\cup(10\text{ MeV},200\text{ MeV})$. This choice gives a range of bound state binding energy from 0.106 MeV to 42.55 MeV. We ensure that the range of γ is cut so that equal numbers of near-threshold virtual and bound state poles are generated. With the values of η, Λ_{bg} and γ specified, the S-matrix in (12) can now be used to calculate the input partial wave $|f(E)|^2$ in (5). For each input, we assign an output label based on the sign of γ, i.e., label 0 if $\gamma>0$ (bound state) and 1 if $\gamma<0$ (virtual state). The number of parameters used results into a total of $4\times500\times1000=2,\ 000,\ 000$ input-output samples for bound and virtual state.

C. Virtual state and resonance

Using the same background phase in (13) and the second factor of (8), the S-matrix with conjugate pair of poles is written as

$$S(p)=e^{2i\delta_{bg}(p)}\frac{(p-i\beta)^2-\alpha^2}{(p+i\beta)^2-\alpha^2}. \tag{14}$$

The values of η and Λ_{bg} are again chosen from Table I but this time we only choose 50 random values for Λ_{bg}. For the pole parameters, 100 values of β is chosen in the interval (50 MeV, 200 MeV) and 100 values of α in (1 MeV, 300 MeV). These choice can give us resonance-peaks with width ranging from 0.12 MeV to 64 MeV. We calculate the input amplitude $|f(E)|^2$ using the above parameters and assign an output label of 1 for virtual state pole ($\beta>\alpha$) and 2 for resonance ($\beta<\alpha$). This is just a continuation of the output assignment in the previous subsection. We have a total of $4\times50\times100\times100=2,\ 000,\ 000$ input-output samples for resonance-virtual classification.

It is interesting to point out that enhancement due to a resonance pole is not completely distinguishable from that of a virtual state pole. Both of these singularities are capable of producing near-threshold peak structures in the scattering region as shown in Fig. 4(d). This is true if we include a background phase in the S-matrix as in (14). A virtual state pole ($\beta>\alpha$) that is far from threshold but close to the imaginary axis of unphysical sheet, as shown in Fig. 4(c), will produce a peak above the threshold due to the distortion caused by the branch point. Normally, if there is no S-matrix background, the conjugate partner of virtual state with width is sufficient to suppress the appearance of peak even if the poles are far from threshold [29]. This is no longer the case in the presence of background and the conjugate pole must be near the threshold to suppress the peak appearance as demonstrated in Fig. 4(b).

A slightly different scenario happens for resonance pole and its conjugate. If it is close to the threshold, a peak structure appears close to the real part of the pole. Here, the conjugate partner is already blocked by the branch cut and can no longer modify the line shape of amplitude. If the resonance pole is moved away from threshold but close to the imaginary axis, the branch point causes the peak structure to appear farther from the pole's real part, resulting in an almost identical line shape as that of the virtual pole [see Fig. 4(d)]. It is therefore crucial to have a

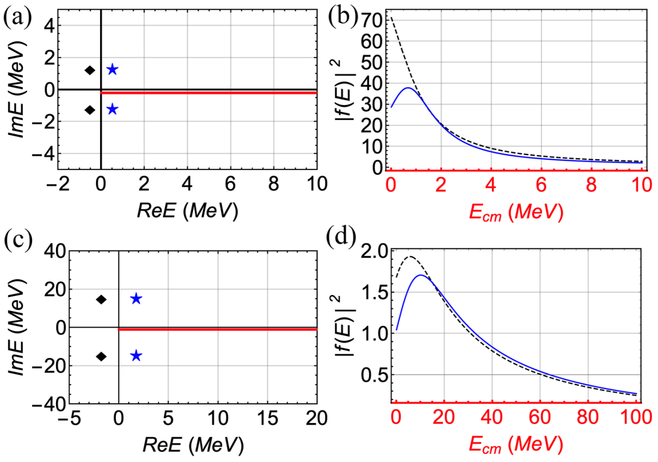

FIG. 4. Virtual (◇) and resonance (⋆) poles near threshold (a) and the corresponding line-shape (b). Poles far from the threshold but close to the imaginary axis of unphysical sheet (c) and the corresponding line-shape (d).

neural network trained to distinguish between these two almost-identical peak structures.

IV. ARCHITECTURE AND TRAINING

Now that we have the classification dataset ready, we proceed with the construction of neural network. To determine the optimal architecture for our task, we experiment with different architectures. Chainer framework [30] is used to build the neural network and to carry out the training. Here, we only use the set 0 of Table I which consists only of bound-virtual samples. This dataset is chosen to deliberately make the classification difficult by putting some of the relevant pole in the branch cut of background. We further split the classification data set into two such that 80% is used for training, which optimizes the weights and biases, and the remaining 20% for testing.

Four neural network architectures are used in this experiment. We describe them using the notation

$$[N_0 + 1, \ldots, N_\ell + 1, \ldots, N_L + 1, 3] \qquad (15)$$

where N_ℓ is the number of nodes in the ℓth layer ($\ell = 0, 1, \ldots, L$), with L as the total number of hidden layers and (+1) denotes the added bias. For all architectures, we have $N_0 = 200$ nodes for the input layer and three nodes for the output. We assign the ReLU as activation function for hidden-layer nodes

$$\mathrm{ReLU}(z_i^{(L+1)}) = \max\,(0, z_i^{(L+1)}) \qquad (16)$$

and use softmax for output nodes

$$\mathrm{softmax}(z_i^{(L+1)}) = \frac{\exp(z_i^{(L+1)})}{\Sigma_j^{N_{L+1}} \exp(z_j^{(L+1)})}. \qquad (17)$$

In the classification problem, the cost-function to be minimized is the softmax cross entropy given by

$$C(\hat{w}, \vec{b}) = \frac{1}{X} \sum_{\vec{x}} \vec{a}(\vec{x}) \cdot \log\,[\vec{y}_{\hat{w},\vec{b}}(\vec{x})] \qquad (18)$$

where $\hat{w}$ is the weight matrix, $\vec{b}$ is the bias vector, $\vec{x}$ is one of the training input with $\vec{a}(\vec{x})$ as the correct answer, X is the size of training sample, and $\vec{y}_{w,b}(\vec{x})$ is the network's output. We use the standard stochastic gradient descent [31,32] to optimize the weights and biases with learning rate of 0.01 and batch size of 1600.

The performance of each network architecture is measured by feeding the testing input to the network and comparing the network's output to the correct label. Then, we count the number of correct predictions. The test accuracy of each architecture is shown in Fig. 5. The vertical axis gives the accuracy of neural network's predictions using the testing set and the horizontal axis is the training epoch. Generally, the testing accuracy shows large fluctuation due to the stochasticity introduced in the calculation of cost-function. It is interesting to find that the performance of $L = 1$ architectures shown in Fig. 5(a) and Fig. 5(b) did not improve much even if we added more nodes. After 1000 epochs, the testing accuracies are 94.4% for the $N_1 = 100$ architecture and 94.5% for the $N_1 = 150$. This is just a 0.1% improvement in accuracy. However, we get a significant increase when the additional 50 nodes are placed in the second hidden layer. For a deep neural network with $L = 2$, $N_1 = 100$ and $N_2 = 50$, the performance is shown in Fig. 5(c). Here, we get a 97.2% testing accuracy after 1000 epochs, a significant improvement compared to $L = 1$ architecture with the same number of nodes. We also check if increasing L, while keeping the total number of nodes fixed, will further improve the performance. The result of $L=3$ with $N_1=N_2=N_3=50$ is shown Fig. 5(d) giving a testing accuracy of 97.3% after 1000 epochs. The result is almost comparable with the $L = 2$ architecture. However, the $L = 2$ architecture is more practical to use since it is much faster to train compared to $L = 3$. Specifically, for the same number of epochs, the total elapsed time of training for the two $L = 1$ architectures are 2.0×10^5 sec and 2.3×10^5 sec, respectively. While for $L = 2$ and $L = 3$, we have 2.6×10^5 sec and 3.1×10^5 sec, respectively. Thus, for the rest of this study we will use a two-hidden layer neural network described in Table II.

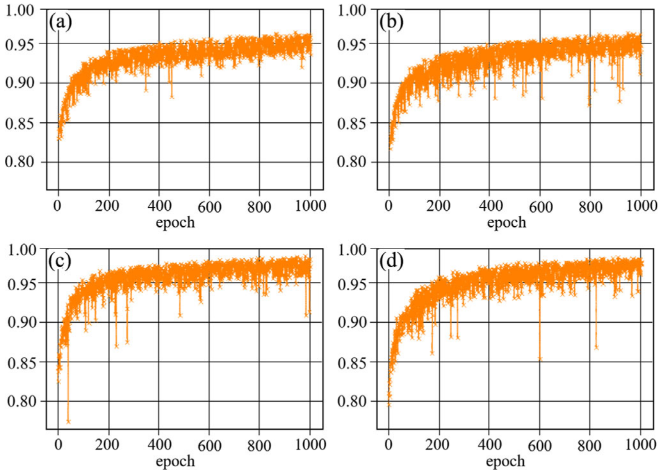

FIG. 5. Testing accuracy of different neural networks with architecture (a) $[200 + 1, 100 + 1, 3]$, (b) $[200 + 1, 150 + 1, 3]$, (c) $[200 + 1, 100 + 1, 50 + 1, 3]$ and (d) $[200 + 1, 50 + 1, 50+ 1, 50 + 1, 3]$

TABLE II. Our deep neural network architecture.

Layer	Number of nodes	Activation Function
Input	200 + 1	
1st	100 + 1	ReLU
2nd	50 + 1	ReLU
Output	3	Softmax

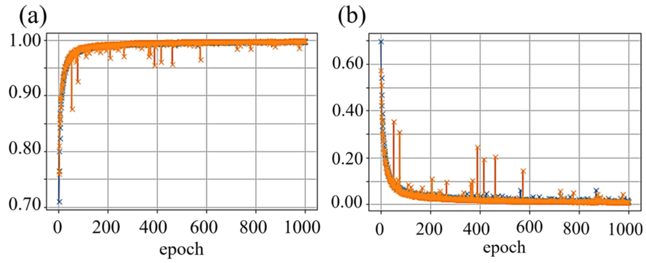

FIG. 6. Testing accuracy of neural network model trained using set 1 (a) and the cost-function profile (b) in each training epoch.

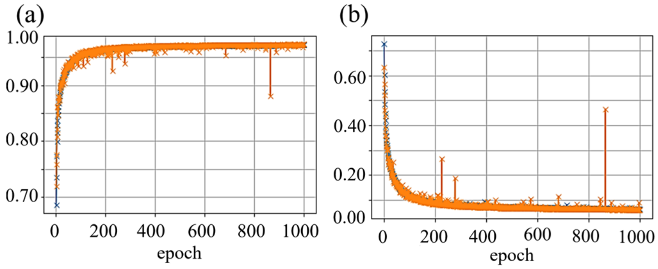

FIG. 7. Testing accuracy of neural network model trained using set 2 (a) and the cost-function profile (b) in each training epoch.

We now proceed to train our chosen network architecture using the classification set 1 and set 2 datasets in Table I. Each of these datasets contains 4,000,000 training input-output tuples for bound-virtual and resonance-virtual cases. The network's performance with set 1 and set 2 datasets are shown in Fig. 6 and Fig. 7, respectively. Optimization using set 1 shows that the accuracy saturates as early as 400 epochs, indicating that the global minimum of the cost-function is already reached. The network's accuracy is 99.7% for the testing of set 1 dataset after 1,000 epochs. The same saturation behavior is observed for set 2. However, the accuracy after 1,000 epochs is only 97.3% for testing. The lower accuracy is due to the inclusion of $\eta = 0$ which corresponds to the no-background case. This gives rise to identical enhancements at threshold whether the pole is a bound or virtual state. Despite its lower accuracy, this dataset is still useful in our subsequent numerical experiment.

We now have two deep neural network models with the same architecture but trained by two slightly different datasets, i.e., set 1 and set 2. In the next section we will study the applicability of these models using an exact solvable separable potential and then apply this to the nucleon-nucleon scattering data.

V. VALIDATION OF NEURAL NETWORK MODEL

We now explore if the trained neural network has the ability to generalize beyond the training dataset. It is important that the validation set be different to that of the training set to make a valid conclusion on the network's ability to generalize. This is done by generating a validation data using an exactly solvable model.

A. Separable potential

The simplest model that can give us an exact solution to the Lippmann-Schwinger equation is a separable potential [21,22]. Here, we consider the s-wave potential given by $V(p,p') = \lambda g(p')g(p)$ with Yamaguchi form factor $g(p) = \Lambda^2/(p^2+\Lambda^2)$ where λ is an energy-independent coupling strength and Λ is a cutoff parameter [33]. The single-channel S-matrix for this model is given by

$$S(p) = \left(\frac{p+i\Lambda}{p-i\Lambda}\right)^2 \left[\frac{2(p-i\Lambda)^2 - \lambda\pi\mu\Lambda^3}{2(p+i\Lambda)^2 - \lambda\pi\mu\Lambda^3}\right]. \quad (19)$$

We can introduce a dimensionless parameter $\zeta = \pi\mu\lambda\Lambda/2$ to rescale the momentum plane with the cutoff Λ as scaling parameter. Figure 8(a) shows the trajectory of pole along the imaginary momentum axis as ζ is varied. At $\zeta = 0$, the pole starts at $p = -i\Lambda$ and as ζ increases in negative value, the pole splits into two. One of the pole moves beyond the cutoff limit while the other one gets closer to threshold. If $-1 < \zeta < 0$, the near-threshold pole $p_0 = i\Lambda(-1+\sqrt{\zeta})$ is a virtual state. If we further make the potential attractive by letting $\zeta < -1$, the near-threshold pole crosses the threshold and becomes a bound state pole. The adjustable parameter ζ can be used to produce different amplitudes to estimate the network's prediction.

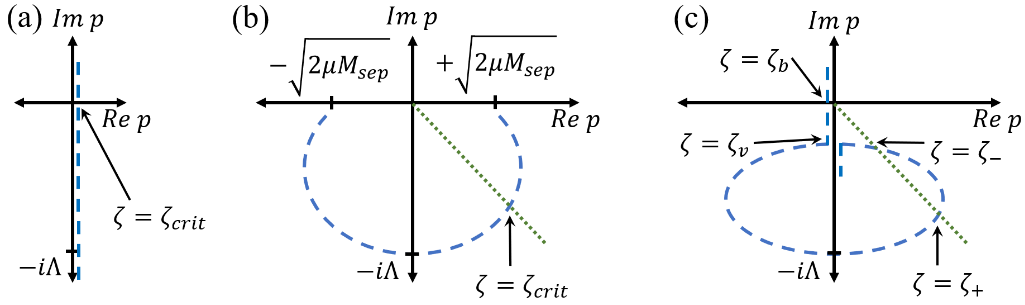

FIG. 8. Pole trajectory of separable potential with energy-independent coupling (a), energy-dependent coupling with $M_{sep} > 0$ (b) and with $M_{sep} < 0$ (c). The dashed line shows the pole's trajectory and the dotted line separates resonances with virtual states.

S-wave bound and virtual enhancement at the threshold are possible for separable potential with energy-independent coupling λ. The absence of a centrifugal barrier makes it impossible to produce resonances with attractive interaction [7]. This can be modified, however, by allowing the coupling to be energy dependent [34]. Minimal number of conjugate poles are produced if we let the energy dependence be

$$\lambda \to (E - M_{sep})\lambda \tag{20}$$

where $E = p^2/(2\mu)$ with threshold at $E = 0$. The parameter M_{sep} is the zero of partial wave amplitude such that when $E = M_{sep}$ there is no scattering. The energy-dependent coupling gives an S-matrix

$$S(p) = \left(\frac{p + i\Lambda}{p - i\Lambda}\right)^2 \left[\frac{2(p - i\Lambda)^2 - \lambda\pi\mu\Lambda^3(E - M_{sep})}{2(p + i\Lambda)^2 - \lambda\pi\mu\Lambda^3(E - M_{sep})}\right] \tag{21}$$

with the pole position at

$$\frac{p}{\Lambda} = \frac{1}{1 - \zeta}\left[-i \pm \sqrt{\zeta(\epsilon\zeta - 1 - \epsilon)}\right] \tag{22}$$

where we introduce a new set of dimensionless parameters $\zeta = \pi\Lambda^3\lambda/4$ and $\epsilon = 2\mu M_{sep}/\Lambda^2$.

Consider the case when the zero of amplitude is on the scattering region, i.e., $M_{sep} > 0$ or $\epsilon > 0$. We get a conjugate pair of poles provided that $\zeta(\epsilon\zeta - 1 - \epsilon) > 0$. This is true for the case of attractive potential, i.e., $\lambda < 0$ or $\zeta < 0$ and repulsive case when $\zeta > (1 + \epsilon)/\epsilon > 0$. We consider only the attractive case which is physically meaningful for the discussion of resonance. Figure 8(b) shows the trajectory of poles as ζ is varied. The conjugate poles start at $p = -i\Lambda$ when $\zeta = 0$ and moves in the opposite direction as ζ becomes negative. The pole remains below the line $|\mathrm{Re}p| = |\mathrm{Im}p|$ when $\zeta > \zeta_{\mathrm{crit}}$ where

$$\zeta_{\mathrm{crit}} = \frac{1}{2}\left[\frac{1 + \epsilon}{\epsilon} - \sqrt{\left(\frac{1 + \epsilon}{\epsilon}\right)^2 + \frac{4}{\epsilon}}\right] < 0. \tag{23}$$

Here, we only have virtual state with width. If we further make ζ negative, such that $\zeta < \zeta_{\mathrm{crit}}$, the pole will move above the line and turns into a resonance pole. As $\zeta \to -\infty$, the pole approaches the point $p = \pm\sqrt{2\mu M_{sep}}$ on the real axis. To ensure that the zero will appear in the cross section, we let the values of M_{sep} to be within [0, 100 MeV].

The pole trajectory for $M_{sep} < 0$ is more involved compared to the previous case. Here, the resonance pole can only be produced provided that $-(3 - \sqrt{8}) < \epsilon < 0$, otherwise ζ will have to be complex. From Fig. 8(c), we start producing virtual state with widths when $\zeta_+ < \zeta < 0$ and then resonance when $\zeta_- < \zeta < \zeta_+$ where

$$\zeta_\pm = \frac{1}{2}\left[\frac{1 + \epsilon}{\epsilon} \pm \sqrt{\left(\frac{1 + \epsilon}{\epsilon}\right)^2 + \frac{4}{\epsilon}}\right] < 0. \tag{24}$$

As ζ becomes more negative, i.e., $\zeta_v < \zeta < \zeta_-$ where $\zeta_v = (1 + \epsilon)/\epsilon$, the resonance pole will again cross the equal-line and turn into virtual state with width. The two poles will then merge on the zero of amplitude at $p = -i\sqrt{2\mu M_{sep}}$ and then split, producing one near-threshold virtual state pole. This near-threshold pole can turn into a bound state pole if $\zeta < \zeta_b$ where

$$\zeta_b = \frac{1}{2}\left[\frac{1 + \epsilon}{\epsilon} - \sqrt{\left(\frac{1 + \epsilon}{\epsilon}\right)^2 - \frac{4}{\epsilon}}\right]. \tag{25}$$

We separate the validation dataset into three, the first one is generated using the energy-independent coupling which gives amplitude enhancement at threshold. The second and third datasets are generated using the energy-dependent coupling, one with $M_{sep} > 0$ and other one with $M_{sep} < 0$. The last two datasets are capable of producing peak structures above the threshold. Also, for convenience, we restrict the third dataset, i.e., with $M_{sep} < 0$, to produce conjugate poles only. In each set, we choose a range of cutoff parameter $(\Lambda_{\mathrm{min}}, \Lambda_{\mathrm{max}})$ and generated 100,000 amplitudes using different combinations of parameters.

We must point out that (19) and (21) have no background branch cuts along the imaginary axis compared to S-matrix of training data in (12) and (14). Instead, the validation data has isolated second order pole at $p = i\Lambda$. This might have some repercussions on the predictive power of the trained neural network when applied to the separable potential.

B. Validation of neural network model trained using set 1

We now proceed to test our trained neural network using the validation dataset. In particular, we want to investigate if the network can generalize beyond the training set, i.e., we still get accurate predictions even if the validation set is different from the training dataset. Note that if the validation set is just a subset of the training dataset, then we expect that the accuracy of prediction should be high. We also want to explore the region of applicability of the trained neural network. We can asses both the ability of the network to generalize and its applicability by changing the value of cutoff Λ since this parameter controls the position of the background singularity.

Consider first the accuracy of prediction with respect to the energy-independent coupling set. From Fig. 9(a), we obtain optimal accuracy in the cutoff region between 400–1000 MeV despite that the background singularity of the validation set is different to that of the training set. We can say that, within this region, the neural network generalizes beyond the training data in distinguishing bound and virtual

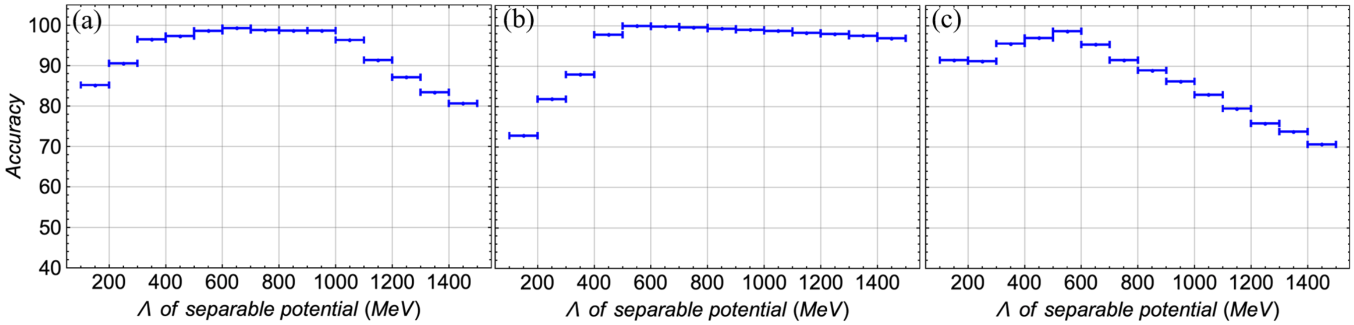

FIG. 9. Accuracy of neural network model trained using set 1 with separable potential amplitude as input for different cutoff parameters. (a) Performance with energy-independent coupling, (b) for energy-dependent coupling with $M_{sep} > 0$ and (c) for energy-dependent coupling with $M_{sep} < 0$. Each horizontal bar correspond to 100,000 input s-wave cross section.

state enhancements. Below 400 MeV, the difference between the training and the validation background starts to manifest as seen from the decrease in accuracy as the cutoff is decreased. We also observe a decrease in accuracy in the cutoff region above 1000 MeV. Here, increasing the cutoff pushes the background far from the scattering region; consequently, a bound or virtual near-threshold pole enhancement becomes identical as we have discussed in Sec. III.

It is interesting to find that the accuracy of prediction is different in energy-dependent set as shown in Fig. 9(b) and Fig. 9(c) even if the neural network is just distinguishing resonance and virtual state with width enhancements for both cases. This difference is probably due to the position of the amplitudes zero, M_{sep}. For the case of $M_{sep} > 0$, i.e., the zero is above the threshold, the second order pole background in (21) can produce a boundlike enhancement at the threshold. This is the reason why we get lower accuracy in Fig. 9(b) below 400 MeV. In fact, the network gives a bound state prediction even if there is no bound state in the validation set. This is, however, suppressed in the $M_{sep} < 0$ case in Fig. 9(c) where the zero below the threshold cancels the effect of the isolated background pole. The absence of extra structure near the threshold allows the network to distinguish a resonance with that of virtual state with width.

The situation is reversed as we go to higher cutoff region. This time, the $M_{sep} > 0$ gives high accuracy in $\Lambda > 600$ MeV as shown in Fig. 9(b) compared to $M_{sep} < 0$ in Fig. 9(c). If Λ is large, the resonance peak can go beyond the center-of-mass energy range. For $M_{sep} < 0$, the zero below the threshold causes the cross section to monotonically rise from some small value to some maximum at $E_{cm} = 100$ MeV. In the absence of peak, the structure for resonance and virtual state with width becomes almost identical. This is the reason why we have decreasing accuracy in Fig. 9(c) as the cutoff increases. On the other hand, for $M_{sep} > 0$, the large Λ means that no boundlike enhancement will appear at the threshold. The structure between the threshold and the zero at $E = M_{sep}$ can still be used by the network to distinguish a resonance with a virtual state with width even if the relevant peak goes beyond the range of center-of-mass energy. This is the reason why we have high accuracy in $M_{sep} > 0$ validation set in high Λ region.

C. Validation of neural network model trained using set 2

For certain values of parameters, the training and validation backgrounds can have similar forms. That is, if we set $\eta = -2$, the training background $e^{2i\delta_{bg}}$ reduces to $(p + i\Lambda_{bg})^2/(p - i\Lambda_{bg})^2$ but with domain $\mathbb{C}/(-i\infty, -i\Lambda_{bg}) \cup (i\Lambda_{bg}, i\infty)$. One may attribute the good performance of our neural network to this similarity. We can test this assumption by using the training set 2 in Table I where $\eta = -2$ is replaced with $\eta = 0$. The accuracy of the network trained using set 2 is shown in Fig. 10. Notice that above 600 MeV, the results are all similar to the performance of network trained using set 1 in Fig. 9. This demonstrates that even if the validation dataset is not in the training set, the neural network can still give high accuracy of predictions. This also illustrates that the decrease in accuracy as the cutoff increases as shown in Fig. 10(a) and Fig. 10(c) is an intrinsic part of pole classification problem.

We pointed out in the previous subsection that the difference in training and validation background manifests in the low cutoff region. The presence of second order pole in the background of validation dataset and the absence of $\eta = -2$ in the training parameter aggravate the situation. This is seen as a drastic drop in accuracy in Fig. 10(b) and Fig. 10(c) below 200 MeV. This means that in this region, the accuracy of the networks prediction is sensitive to the nature of background singularity.

We give a short comment on the network's performance on the shallow bound and virtual state produced by energy-dependent set with $M_{sep} < 0$. From the trajectory of poles in Fig. 8(c), a near-threshold bound state or virtual pole is always accompanied by another virtual pole. The latter pole is much closer to the scattering region compared to the

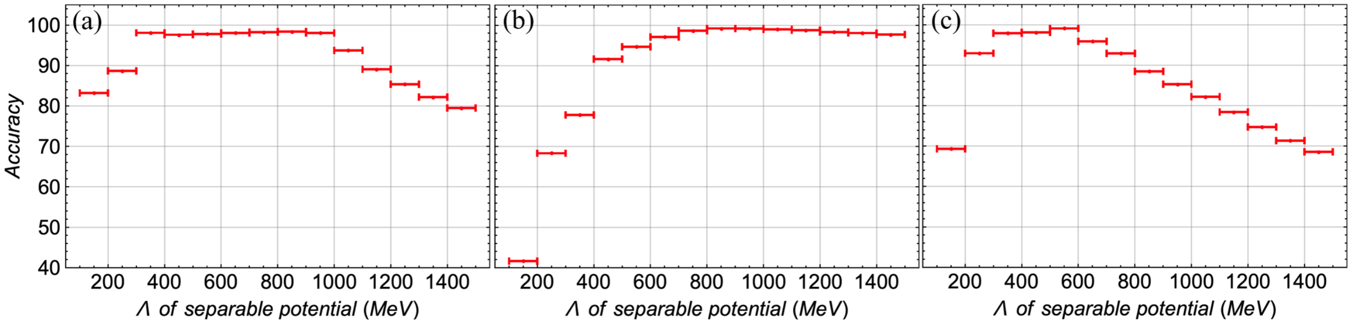

FIG. 10. Accuracy of neural network model trained using set 2 with separable potential amplitude as input for different cutoff parameters. (a) Performance with energy-independent coupling, (b) for energy-dependent coupling with $M_{sep} > 0$ and (c) for energy-dependent coupling with $M_{sep} < 0$. Each horizontal bar correspond to 100, 000 input s-wave cross section.

accompanying virtual pole of (19) in Fig. 8(a). This makes the classification difficult, i.e., accuracy is less than 50%, because the training S-matrix in (12) educates the network only with a single near-threshold pole. This can be improved by inserting an extra pole factor in (12) to simulate this background virtual pole.

D. Application to nucleon-nucleon system

As a final validation, we use the partial wave analyses and potential models of the Nijmegen group [35–38] as input to our neural network. These models are fitted to the nucleon-nucleon scattering data published between 1955 to 1992. They give the correct phase shifts at any laboratory kinetic energy below 350 MeV. The fitting results are summarized in Table III. Here, PWA93 corresponds to the analyses of multienergy partial wave on the pp data, the np data and on the combined pp and np database [36]. All three analyses give an excellent fit of $\chi^2/N \sim 1$ where N denotes the number of scattering data. Nijm93 is the Nijmegen soft-core potential model introduced in [37] with NijmI as the nonlocal Reid-like and NijmII is the local version. In the same paper, Reid93 is also introduced which is a regularized Reid soft-core potential. All of these contain the charge-dependent one-pion exchange tail. Lastly, two meson-exchange is included in the extended soft-core ECS96 model of [38].

Now, using the $^1S_0$ and $^3S_1$ phase-shifts of the mentioned models, we generate the input amplitude on a center-of-mass energy interval [0, 100 MeV]. We can say that within the cutoff range from 400 MeV to 1, 000 MeV, our neural network model can classify a bound-virtual enhancement with 98% accuracy based on our analysis with separable potential model. The resulting amplitude is then fed to the neural network and the results are shown in Table IV. All the predictions are correct, i.e., the network was able to identify that the $^1S_0$ partial wave threshold enhancement is due to the presence of a virtual state pole while that of $^3S_1$ is due to a bound state pole. It is interesting to point out that the small differences among the models do not affect the network's prediction. This means that if the input data falls within some error band, the neural network can still give consistent classification.

TABLE III. χ^2/N for the Nijmegen partial wave and potential models in the 0–350 MeV laboratory frame energy interval. Data for PWA93, NijmI, NijmII, Nijm93, and Reid93 are taken from [37] and ESC96 is from[38].

	PWA93	ECS96	NijmI	NijmII	Nijm93	Reid93
N_{par}	39	14	41	47	15	50
χ^2/N	0.99	1.26	1.03	1.03	1.87	1.03

TABLE IV. Neural network's prediction with Nijmegen model's amplitude as input. We get the same result whether we use the network trained using either set 1 or set 2.

	PWA93	ECS96	NijmI	NijmII	Nijm93	Reid93
$^1S_0$	virtual	virtual	virtual	virtual	virtual	virtual
$^3S_1$	bound	bound	bound	bound	bound	bound

VI. CONCLUSION

This study set out to demonstrate how deep learning can be applied in classifying the nature of pole causing a cross section enhancement. The method is straightforward in a sense that we can use a simple S-matrix parametrization to generate all the possible line shape that can emerge in the scattering region. We have shown that our neural network model gives high accuracy of more than 90% in the acceptable range of cutoff parameter (400–800 MeV). This suffices to have an accurate prediction on the nucleon-nucleon scattering data. Also, the study shows that a neural network trained using a simple S-matrix parametrization is able to generalize beyond the training set. This is demonstrated when we validated our neural network using separable potential models and the nucleon-nucleon Nijmegen models. However, there are limitations in the applicability of deep learning for enhancement classification. One example is the noticeable decrease in accuracy if the cutoff parameter is too large. For the bound-virtual classification, the effect of background is important to distinguish the two structures. While for virtual-resonance classification, the peak structure tend to appear beyond the center-of-energy range if the cutoff is very large, making the classification difficult.

It is important to extend our approach to coupled-channel case since most of the exotic phenomena are believed to be generated from coupled-channel interactions. Although the current study deals with single-channel scattering, the findings can still be used in coupled-channel analysis. In particular, we found that if the validation cutoff is too small, then the neural network's prediction becomes sensitive to the nature of background singularity. This observation should extend to the coupled-channel case and it is appropriate to explore other possible background parametrization such as the one used in [39,40]. This will be done elsewhere.

ACKNOWLEDGMENTS

This study is supported in part by JSPS KAKENHI Grant No. JP17K14287, and by MEXT as "Priority Issue on Post-K computer" (Elucidation of the Fundamental Laws and Evolution of the Universe) and SPIRE (Strategic Program for Innovative Research). A. H. is supported in part by JSPS KAKENHI No. JP17K05441 (C) and Grants-in-Aid for Scientific Research on Innovative Areas, No. 18H05407, No. 19H05104. D. L. B. S. is supported by the UP OVPAA FRASDP and DOST-PCIEERD postdoctoral research grant.

[1] S. K. Choi *et al.* (Belle Collaboration), Phys. Rev. Lett. **91**, 262001 (2003).
[2] S. L. Olsen, T. Skwarnicki, and D. Zieminska, Rev. Mod. Phys. **90**, 015003 (2018).
[3] F.-K. Guo, C. Hanhart, U.-G. Meißner, Q. Wang, Q. Zhao, and B.-S. Zou, Rev. Mod. Phys. **90**, 015004 (2018).
[4] Y. Yamaguchi, A. Hosaka, S. Takeuchi, and M. Takizawa, J. Phys. G **47**, 053001 (2020).
[5] F.-K. Guo, X.-H. Liu, and S. Sakai, Prog. Part. Nucl. Phys. **112**, 103757 (2020).
[6] W. R. Frazer and A. W. Hendry, Phys. Rev. **134**, B1307 (1964).
[7] B. C. Pearce and B. F. Gibson, Phys. Rev. C **40**, 902 (1989).
[8] F.-K. Guo, C. Hanhart, Q. Wang, and Q. Zhao, Phys. Rev. D **91**, 051504(R) (2015).
[9] E. S. Swanson, Phys. Rev. D **91**, 034009 (2015).
[10] L. D. Landau, Nucl. Phys. **13**, 181 (1959).
[11] S. X. Nakamura and K. Tsushima, Phys. Rev. D **100**, 051502 (2019).
[12] S. X. Nakamura, Phys. Rev. D **100**, 011504 (2019).
[13] E. Alpaydin, *Introduction to Machine Learning* (The MIT Press, Cambridge, 2010).
[14] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova, Rev. Mod. Phys. **91**, 045002 (2019).
[15] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015), http://neuralnetworksanddeeplearning.com/.
[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature (London) **323**, 533 (1986).
[17] Y. Ikeda, T. Hyodo, D. Jido, H. Kamano, T. Sato, and K. Yazaki, Prog. Theor. Phys. **125**, 1205 (2011).
[18] R. J. Eden, P. V. Landshoff, D. I. Olive, and J. C. Polkinghorne, *The Analytic S-Matrix* (Cambridge University Press, London, England, 1966).
[19] G. F. Chew, *S-Matrix Theory of Strong Interactions* (W. A. Benjamin Inc., New York, 1961).
[20] G. F. Chew, Rev. Mod. Phys. **34**, 394 (1962).
[21] J. R. Taylor, *Scattering Theory: The Quantum Theory of Nonrelativistic Collisions* (John and Wiley & Sons, New York, 1972).
[22] R. G. Newton, *Scattering Theory of Waves and Particles*, 2nd ed. (Springer-Verlag, Berlin Heidelberg, 1982).
[23] N. Suzuki, T. Sato, and T.-S. H. Lee, Phys. Rev. C **79**, 025205 (2009).
[24] N. G. van Kampen, Phys. Rev. **91**, 1267 (1953).
[25] N. Hu, Phys. Rev. **74**, 131 (1948).
[26] R. L. Workman, R. A. Arndt, W. J. Briscoe, M. W. Paris, and I. I. Strakovsky, Phys. Rev. C **86**, 035202 (2012).
[27] S. Ishida, M. Ishida, T. Ishida, K. Takamatsu, and T. Tsuru, Prog. Theor. Phys. **98**, 621 (1997).
[28] T. W. Gamelin, *Complex Analysis* (Springer-Verlag, New York, 2001).
[29] M. Döring, C. Hanhart, F. Huang, S. Krewald, and U.-G. Meißner, Nuc. Phys. **A829**, 170 (2009).
[30] https://github.com/pfnet/chainer#reference.
[31] L. Bottou, F. E. Curtis, and J. Nocedal, arXiv:1606.04838v3.
[32] R. Ge, F. Huang, C. Jin, and Y. Yuan, arXiv:1503.02101v1.
[33] Y. Yamaguchi, Phys. Rev. **95**, 1628 (1954).
[34] Y. Ikeda, H. Kamano, and T. Sato, Prog. Theor. Phys. **124**, 533 (2010).
[35] http://nn-online.org/NN/?page=nnphs2.
[36] V. G. Stoks, R. A. M. Klomp, M. C. M. Rentmeester, and J. J. de Swart, Phys. Rev. C **48**, 792 (1993).
[37] V. G. Stoks, R. A. M. Klomp, C. P. F. Terheggen, and J. J. de Swart, Phys. Rev. C **49**, 2950 (1994).
[38] Th. A. Rijken and V. G. Stoks, Phys. Rev. C **54**, 2869 (1996).
[39] D. M. Manley, Int. J. Mod. Phys. A **18**, 441 (2003).
[40] D. M. Manley and E. M. Saleski, Phys. Rev. D **45**, 4002 (1992).