Model independent analysis of coupled-channel scattering: A deep learning approach

Denny Lane B. Sombillo,[1,2,*] Yoichi Ikeda,[3] Toru Sato,[2] and Atsushi Hosaka[2,4]

[1]*National Institute of Physics, University of the Philippines Diliman, Quezon City 1101, Philippines*
[2]*Research Center for Nuclear Physics (RCNP), Osaka University, Ibaraki, Osaka 567-0047, Japan*
[3]*Department of Physics, Kyushu University, Fukuoka 819-0395, Japan*
[4]*Advanced Science Research Center, Japan Atomic Energy Agency, Tokai, Ibaraki 319-1195, Japan*

(Received 13 May 2021; accepted 12 July 2021; published 5 August 2021)

We develop a robust method to extract the pole configuration of a given partial-wave amplitude. In our approach, a deep neural network is constructed where the statistical errors of the experimental data are taken into account. The teaching dataset is constructed using a generic S-matrix parametrization, ensuring that all the poles produced are independent of each other. The inclusion of statistical error results into a noisy classification dataset which we should solve using the curriculum method. As an application, we use the elastic πN amplitude in the $I(J^P) = 1/2(1/2^-)$ sector where 10^6 amplitudes are produced by combining points in each error bar of the experimental data. We fed the amplitudes to the trained deep neural network and find that the enhancements in the πN amplitude are caused by one pole in each nearby unphysical sheet and at most two poles in the distant sheet. Finally, we show that the extracted pole configurations are independent of the way points in each error bar are drawn and combined, demonstrating the statistical robustness of our method.

DOI: 10.1103/PhysRevD.104.036001

I. INTRODUCTION

Identifying the nature of observed enhancements in hadron-hadron scatterings is one of the central goals of hadron spectroscopy [1–7]. The primary concern is to associate the enhancements with nearby S-matrix poles [8–12]. Ideally, we make a complete measurement of observables, such as elastic and reaction cross sections with spin dependence if possible. For low energy processes, the measurement is followed by partial wave analysis of the experimental data to obtain the amplitudes. The poles, Riemann sheets, and other resonance parameters are extracted from the amplitudes. To achieve this program, conventionally, one often uses a simple parametrization of the amplitude such as Breit-Wigner or Flatte parametrizations, or some more rigorous formulation like the K-matrix formalism [13,14]. Alternatively, one can use a dynamical model using suitable hadron degrees of freedom and their interactions. In these methods, some parameters are determined to fit the experimental data. In general, however, the applicability of these methods is somewhat limited by the assumptions of the model.

In this study, we propose an alternative method to the above parameter-fitting approach using deep learning [15–18]. Eventually, we would aim at the program that determines detailed properties of poles of the amplitude, including their positions, residues, among others. However, this is extremely difficult at this moment, and therefore, we limit our program to identify the pole configuration, that is, the number of poles in each Riemann sheet associated with the structures in the amplitude. The information we obtain is limited but valuable enough to draw some physical insights (see Refs. [19–23]). Furthermore, the obtained pole configuration is useful in designing an appropriate parametrization with minimal number of parameters to extract the pole positions. In this way, the information obtained using deep learning in a model-independent manner is complemented by a more practical model-dependent parameter fitting method.

We design and develop a deep neural network (DNN) to detect the pole configuration using only the partial-wave amplitude. The idea is to produce a general set of simulated amplitudes with known pole configuration to build the training dataset. In doing so, the general properties of the S-matrix (analyticity, unitarity, and Hermiticity) are satisfied. Furthermore, the poles that we generate must be independent of each other to ensure that the detected poles by the DNN are free from any bias. The next step is to

*sombillo@rcnp.osaka-u.ac.jp, dbsombillo@up.edu.ph

construct a DNN program. In the optimization of the DNN, we use the training dataset to teach the DNN to recognize physically realizable amplitudes from the experimental data and extract the pole configuration. As anticipated, our method is a classification program that gives a general description of the data. Therefore, to extract the physical properties of the enhancement structures in the amplitudes, a dynamical model approach must be used while maintaining the result obtained by our program. We emphasize that once the DNN model are constructed, the optimized parameters in DNN can be reused to analyze different processes. This treatment is possible since our formulation does not rely on a specific functional form of amplitudes.

We consider the two-hadron scatterings with two coupled channels. For a more specific demonstration, we take the elastic πN partial-wave amplitude, collected and analyzed by the GW-SAID in Refs. [24,25], as our experimental data. We choose this amplitude because of the interesting role of threshold effects to the noticeable structures seen just above the ηN and below the $K\Sigma$ thresholds as shown in Fig. 1. We use the coupled πN and ηN channels treatment for the present study. In addition to the deep learning approach, we also introduce an alternative way to utilize the error bars to interpret experimental data. Instead of generating several parameters of a model that fits within the experimental result, we do the reverse treatment. We produce several amplitudes directly from the experimental results by combining points in the error bars then feed them to the trained DNN. Since there are infinitely different ways to combine points in each error bar, we expect to get different pole configurations. We can interpret the DNN inference with the highest count as the best pole configuration associated with the experimental data. Thus, we are extracting the information directly from the experimental data and not just constraining the model.

The structure is as follows. First, we discuss in Sec. II the general properties of the S-matrix used to generate the training dataset. Then, we proceed with the construction and optimization of our DNN in Sec. III. The treatment of experimental data for the DNN inference is done in Sec. IV. Finally, we give our conclusion in Sec. V.

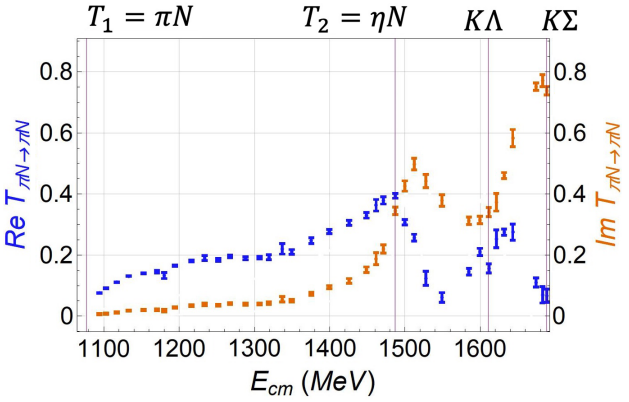

FIG. 1. The elastic πN amplitude of the GW-SAID in [26]. The two-hadron thresholds are shown as vertical thin lines. Only the πN and ηN channels are considered in the present study. The $K\Lambda$ and $K\Sigma$ thresholds are shown for reference purposes only.

II. GENERAL PROPERTIES OF S-MATRIX

In this section, we review the general properties of the S-matrix that we will use in the generation of the training dataset. For the present purpose, it will suffice to consider the two-hadron scatterings with two channels. Here, we take into account the πN as channel 1 and ηN as channel 2. The $K\Lambda$ channel can be ignored due to its weak coupling to the πN channel [27]. We can also ignore the $K\Sigma$ channel by restricting the energy region of analysis up to the $K\Sigma$ threshold energy. The proximity of the peak structures to ηN threshold allows us to utilize the nonrelativistic relation,

$$E = \frac{p_i^2}{2\mu_i} + T_i. \tag{1}$$

where E is the center-of-mass scattering energy, p_i is the relative momentum of the $i$the channel ($i \in \{1, 2\}$) with two-particle reduced mass μ_i and threshold energy T_i. Here, the smooth variation of the lower-channel is approximated using $p_1 = \sqrt{2\mu_1(E - T_1)}$.

To reveal the full analytic properties of the S-matrix, we let the energy and momentum take complex values. With the relation in Eq. (1), the complex energy is a two-valued function of complex momenta with the positive or negative imaginary part. We call the energy plane associated with the positive (negative) imaginary part of the ith momentum channel as the top [t] (bottom [b]) Riemann sheet. For the coupled two-channels, the four Riemann sheets are conveniently labeled using the intuitive notation in Ref. [28] as $[s_1 s_2]$ where $s_i = \{t, b\}$ is the sheet of the ith channel. The scattering region, the energy region where we plot the cross section, lies along the real axis on the upper half of the physical $[tt]$ sheet. The other Riemann sheets are collectively called unphysical sheets, and they are connected, together with the physical sheet, in a nontrivial way due to the presence of branch cut. Figure 2 shows the connection of the relevant regions of all the Riemann sheets in two perspectives. First, the upper-half of $[tt]$ ($[tb]$) is connected to lower-half of $[bt]$ ($[bb]$) between the interval T_1 to T_2. Figure 2(a) shows that the upper-half of $[tt]$ is connected to the lower half of $[bb]$ above T_2. Second, Fig. 2(b) shows that the lower half of $[bt]$ is connected to the upper-half of $[tb]$ above T_2. Note that all the Riemann sheets are disconnected from each other below the lowest threshold T_1. Generally, poles near the scattering region, thus on the $[bb]$ and $[bt]$ sheets, may generate a prominent peak, but some poles on the $[tb]$ sheet can still give a noticeable effect, especially if close to the threshold energy.

The properties of the S-matrix govern the collision of particles with short-range interaction. The most important

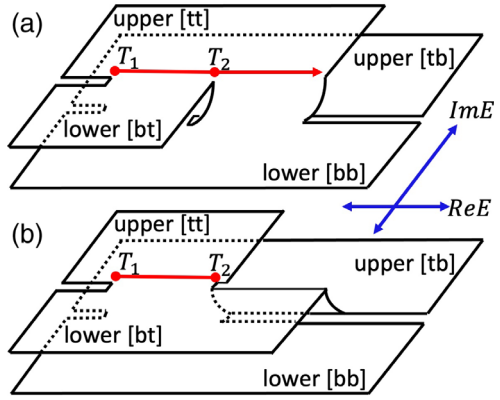

FIG. 2. Relevant regions of two-channel Riemann sheets. The red ray, with two dots, represents the scattering region lying on the upper $[tt]$. The curve surface in (a) shows the $[tt]-[bb]$ connection, while the curve in (b) shows the $[bt]-[tb]$ connection.

property is causality, which means that no scattered wave will reach the detector before the incident wave reaches the scattering target. Causality imposes that the scattering amplitudes take the real-boundary values of some analytic functions of complex energy on the physical sheet [10,29]. It follows that there should be no pole on the physical sheet except possibly some bound state poles on the real axis below the lowest threshold. Next, we also expect that the total probability is conserved, which requires that the S-matrix S be unitary, i.e., $SS^{\dagger}=1$. This property allows us to restrict the form of the full S-matrix and its elements in the corresponding interval of scattering energy. Finally, below the lowest threshold, the S-matrix should be real-valued or Hermitian [30]. The well-known consequence of Hermiticity is the reflection principle [8,9], i.e., the energy poles come in conjugate pairs. By combining analyticity, unitarity, and hermiticity on the relevant energy region, we can construct a general parametrization for our S-matrix.

Using Hermiticity and unitarity, we can perform analytic continuation [31,32] and obtain the following S-matrix elements:

$$\begin{aligned} S_{11}(p_1,p_2) &= \frac{D(-p_1,p_2)}{D(p_1,p_2)} \\ S_{22}(p_1,p_2) &= \frac{D(p_1,-p_2)}{D(p_1,p_2)} \\ S_{11}S_{22}-(S_{12})^2 &= \frac{D(-p_1,-p_2)}{D(p_1,p_2)}, \end{aligned} \tag{2}$$

where the subscripts refer to the channels. Here, the function $D(p_1,p_2)$ determines the pole positions of the S-matrix. We can calculate the scattering amplitudes using the relation $S_{ii'}=\delta_{ii'}+2iT_{ii'}$ where $\delta_{ii'}$ is the Kronecker delta and $i,i'\in\{1,2\}$. If we can control the singularities using a specified form of $D(p_1,p_2)$, then we can generate a set of simulated amplitudes $T_{ii'}(p_1,p_2)$ with known pole configuration.

A. Controlled poles and Riemann sheets

To accommodate a large pole-configuration space, we have to generate a set of simulated amplitudes with arbitrary number of independent poles. We look for $D_j(p_1,p_2)=0$ that can give exactly one pole $E^{(j)}_{\rm pole}$ occupying one of the unphysical Riemann sheet and then form the product,

$$D(p_1,p_2)=\prod_j D_j(p_1,p_2). \tag{3}$$

This prescription ensures that all $E^{(j)}_{\rm pole}$s are produced independently of each other. Now, from the reflection principle, we know that if $E^{(j)}_{\rm pole}$ is a pole, then the complex conjugate $E^{(j)*}_{\rm pole}$ must also be a pole. To fix the Riemann sheet of $E^{(j)}_{\rm pole}$ and accommodate its conjugate partner, we use

$$\begin{aligned} D_j(p_1,p_2) &= [(p_1-i\beta_1^{(j)})^2-\alpha_1^{(j)2}] \\ &\quad +\lambda[(p_2-i\beta_2^{(j)})^2-\alpha_2^{(j)2}]. \end{aligned} \tag{4}$$

The absolute values of the real parameters $\alpha_i^{(j)},\beta_i^{(j)}$ are already determined by the real and imaginary parts of $E^{(j)}_{\rm pole}$ through the relation in Eq. (1). To specify the Riemann sheet of $E^{(j)}_{\rm pole}$, we choose the signs of $\beta_1^{(j)}$ and $\beta_2^{(j)}$. This feature allows us to uphold the analyticity requirement by not choosing simultaneous positive $\beta_1^{(j)}$ and $\beta_2^{(j)}$. The importance of the extra parameter λ is discussed in the following.

A quick inspection shows that, if Eq. (1) is imposed on $D_j(p_1,p_2)=0$ with $D_j(p_1,p_2)$ given by Eq. (4), we get four solutions in either p_1 or p_2. Two of these are the assigned pole $E^{(j)}_{\rm pole}$ and its conjugate partner while the other two are known as the shadow and its conjugate partner [33–36]. The position and Riemann sheet of the shadow pole is obtained by expressing $D_j(p_1,p_2)=0$ in terms of p_1 (or p_2) and factoring out $(p_1-i\beta_1^{(j)})^2-\alpha_1^{(j)2}$ [or $(p_2-i\beta_2^{(j)})^2-\alpha_2^{(j)2}$]; these are

$$\left[p_1-i\beta_1^{(j)}\left(\frac{\mu_1-\mu_2\lambda}{\mu_1+\mu_2\lambda}\right)\right]^2-\left[\alpha_1^{(j)2}+\frac{4\lambda\mu_1^2\beta_2^{(j)2}}{(\mu_1+\mu_2\lambda)^2}\right]=0 \tag{5}$$

and

$$\left[p_2+i\beta_2^{(j)}\left(\frac{\mu_1-\mu_2\lambda}{\mu_1+\mu_2\lambda}\right)\right]^2-\left[\alpha_2^{(j)2}+\frac{4\lambda\mu_2^2\beta_1^{(j)2}}{(\mu_1+\mu_2\lambda)^2}\right]=0. \tag{6}$$

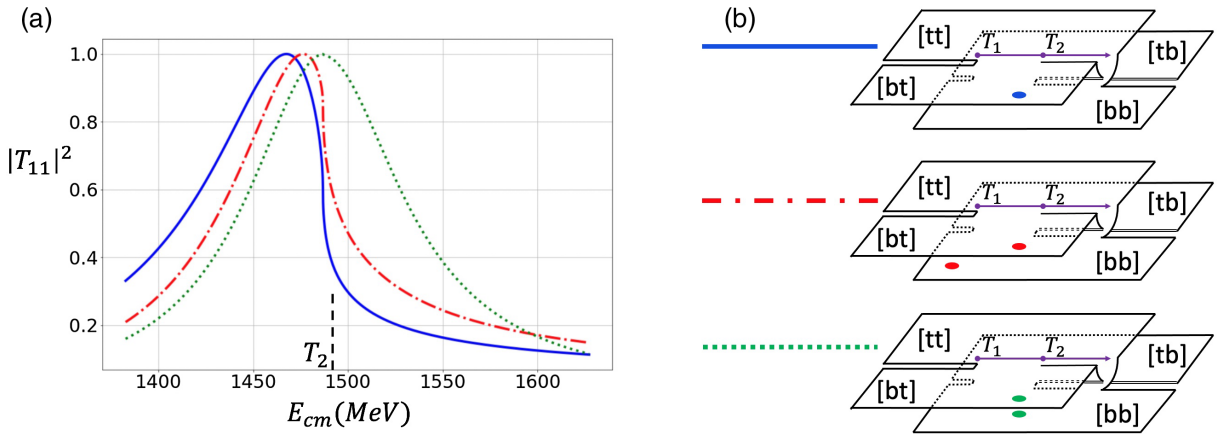

FIG. 3. The partial cross section (a) and the corresponding pole configuration (b). The pole in $[bt]$ sheet is fixed and is the same for all cases with real part equal to T_2. The $[bb]$ pole is added without assuming any trajectory.

The reversed sign of $\mathrm{Im}p_2$ tells us that the Riemann sheet of the shadow is different to that of the assigned pole. This means that $E^{(j)}_{\mathrm{pole}}$ can respect analyticity but the shadow may not. To avoid the possible violation of analyticity, we throw the shadow far from the relevant scattering region. In particular, we choose a sufficient negative λ that can put the shadow (and its conjugate partner) on the real axis below the lowest threshold. By doing this, each $E^{(j)}_{\mathrm{pole}}$ will have its own distant background pole. Thus, we can guarantee that each $D_j(p_1, p_2)$ will produce exactly one isolated pole on a specified Riemann sheet. In this way, the poles are produced independent of each other while respecting analyticity, giving us a general parametrization.

B. Observable effects of shadow

We push the shadow pole solution of $D_j(p_1, p_2) = 0$ far from the relevant scattering region to arrive at a general parametrization. However, there are situations where a shadow pole exists and may have observable consequences in the scattering region around the relevant threshold. Consider, for example; the familiar two-channel Breit-Wigner model [33,34] with the pole-position condition taking the form,

$$D(p_1, p_2) = E - E_{\mathrm{BW}} + i\gamma_1 p_1 + i\gamma_2 p_2 = 0, \quad (7)$$

where E_{BW} is the Breit-Wigner mass and $\gamma_i \geq 0$ is the coupling to the ith channel. For $\gamma_2 < \gamma_1$, we can have a pole in $[bb]$ and a shadow in $[bt]$. If we slowly turn off γ_2, the pole and shadow will move towards a common energy point (the same real and imaginary part) but in different Riemann sheets. The cusp at T_2, due to $p_2 \propto \sqrt{E - T_2}$, disappears on the scattering amplitude indicating that the situation is now reduced to a single-channel scattering.

It turns out that putting another independent pole in our parametrization can mimic the effect of shadow. To demonstrate this effect, let us consider the extreme case where we fixed the position of an isolated pole on the $[bt]$ sheet such that its real part is equal to T_2. Figure 3 shows the resulting $|T_{11}|^2$ (solid blue line) where a peak structure below T_2 with a noticeable cusp (infinite slope) at the threshold is observed. The addition of an arbitrary $[bb]$ pole below T_2 push the peak structure slightly closer to T_2 (dashed red line) and slightly diminish the cusp structure. Finally, if the $[bb]$ pole is placed exactly at the same position as the fixed $[bt]$ pole, the peak occurs at the actual real part of the pole (which is at T_2), and the cusp disappeared (dotted green line). This behavior is an indication that the pole decouples to the second channel, similar to turning off γ_2 in the two-channel Breit-Wigner model. It follows that, we can turn off the channel coupling by using an arbitrary trajectory for two independent poles on different sheets. This behavior shows that the arbitrary pole we placed on a different Riemann sheet effectively functions as a shadow of the fixed pole. In our formulation, we can generate the pole and shadow independently. The decoupling effect of a pair of independent poles on different Riemann sheets can be proven in a general way as discussed in the Appendix.

We now have a parametrization that can produce as many poles as we want with an arbitrary way of controlling the strength of channel coupling. In the next section, we use our general S-matrix parametrization to produce a set of simulated amplitudes for the teaching dataset.

III. CONSTRUCTION OF DEEP NEURAL NETWORK MODEL

A. Generation of training dataset

The design of our DNN generally depends on the features of input data, which we take to be the real and imaginary parts of amplitude, and the number of possible classifications in the teaching dataset. In the final stage of analysis, we will use the experimental data to make DNN inferences. It is, therefore, essential to include the limited

energy resolution in the design of DNN. To do this, we generate a set of amplitudes on randomly spaced energy points. Specifically, we divide the region of interest in the experimental data into some number of bins, say B bins. One can think of each bin as the energy resolution of a particle detector. It is safe to assume that some probability distribution determines the energy of a particle entering a detector. We can use this distribution to pick one representative energy in each bin to calculate the real and imaginary parts of the amplitude. For our specific task, we divide the πN center-of-mass energy range, from πN to $K\Sigma$ thresholds, into 37 bins since there are 37 data points in the GW-SAID data [24,25]. Also, for simplicity, we use a uniform distribution to choose a point in each energy bin.

The number of possible pole-configuration classifications depends on how many poles we are willing to count within some specified Riemann sheet region. The number of configurations will also determine the number of output nodes assigned to our DNN model. Appealing once again to the GW-SAID data, there are two prominent structures—one is around the ηN threshold, and the other is between $K\Lambda$ and $K\Sigma$ thresholds. The number of poles that we have to count should not be less than two. In our study, it suffices to consider a maximum of four distributed poles in any combination of Riemann sheets. With this restriction, we identified 35 possible configurations. Each of these configurations corresponds to one output node in our DNN. The possible configurations and output-node assignment are shown in Table I.

The enhancements in the GW-SAID πN amplitude occur only on a limited portion of the scattering region. Thus, it is practical to limit the region where we produce and count our poles in each Riemann sheet. We define the counting region as

$$\begin{cases} T_2 - 50 \le \mathrm{Re}\, E_{\mathrm{pole}} \le T_2 + 200 & \text{all RS} \\ -200 \le \mathrm{Im}\, E_{\mathrm{pole}} < 0 & [bt]\ \&\ [bb] \\ 0 < \mathrm{Im}\, E_{\mathrm{pole}} \le 200 & [tb], \end{cases}$$

where the energies are in units of MeV. Here, we do not have to count the conjugate poles on the upper half of $[bt]$ or $[bb]$ and the lower half of $[tb]$ since they do not correspond to different independent states. In addition to the counted poles, we also produce distant background poles far from the scattering region, but these are not counted. By randomly generating at-most four poles inside the counting region, we constructed 1.8×10^6 labeled amplitudes divided uniformly into 35 configurations for the training set. Our task is to find a map between the input amplitude to the corresponding output pole configuration in the form of a DNN model. To monitor the performance of DNN, we also produce an independent 3.5×10^4 labeled amplitudes for the testing set.

TABLE I. Classification output-node label.

Label	S-matrix pole configuration
0	No nearby pole
1	1 pole in $[bt]$
2	2 poles in $[bt]$
3	3 poles in $[bt]$
4	4 poles in $[bt]$
5	3 poles in $[bt]$ and 1 pole in $[bb]$
6	2 poles in $[bt]$ and 1 pole in $[bb]$
7	2 poles in $[bt]$ and 2 poles in $[bb]$
8	1 pole in $[bt]$ and 2 poles in $[bb]$
9	1 pole in $[bt]$ and 3 poles in $[bb]$
10	1 pole in $[bt]$ and 1 pole in $[bb]$
11	1 pole in $[bb]$
12	2 poles in $[bb]$
13	3 poles in $[bb]$
14	4 poles in $[bb]$
15	3 poles in $[bb]$ and 1 pole in $[tb]$
16	2 poles in $[bb]$ and 1 pole in $[tb]$
17	2 poles in $[bb]$ and 2 poles in $[tb]$
18	1 pole in $[bb]$ and 2 poles in $[tb]$
19	1 pole in $[bb]$ and 3 poles in $[tb]$
20	1 pole in $[bb]$ and 1 pole in $[tb]$
21	1 pole in $[tb]$
22	2 poles in $[tb]$
23	3 poles in $[tb]$
24	4 poles in $[tb]$
25	3 poles in $[tb]$ and 1 pole in $[bt]$
26	2 poles in $[tb]$ and 1 pole in $[bt]$
27	2 poles in $[tb]$ and 2 poles in $[bt]$
28	1 pole in $[tb]$ and 2 poles in $[bt]$
29	1 pole in $[tb]$ and 3 poles in $[bt]$
30	1 pole in $[tb]$ and 1 pole in $[bt]$
31	2 poles in $[bt]$, 1 pole in $[bb]$ and 1 pole in $[tb]$
32	1 pole in $[bt]$, 2 poles in $[bb]$ and 1 pole in $[tb]$
33	1 pole in $[bt]$, 1 pole in $[bb]$ and 2 poles in $[tb]$
34	1 pole in $[bt]$, 1 pole in $[bb]$ and 1 pole in $[tb]$

B. DNN architecture

Figure 4 shows the basic architecture of our DNN model. The first 37 nodes in the input layer are for the random energy points chosen in each bin, and the last two 37 nodes are for real and imaginary parts of the amplitude. For the output layer, we use 35 nodes such that each one corresponds to the pole configuration listed in Table I. There are no general guidelines into how many hidden layers and nodes in each layer we should assign to obtain an optimal result. It is, therefore, useful to test some architectures. In this study, we experiment with six different architectures with hidden layer designs shown in Table II. Here, we use the notation [XXX-$\cdots$-XXX] to denote the number of layers and number of nodes in each hidden layer. For example, [200-200] means that the architecture has two hidden layers with 200 nodes in each layer while [100-100-

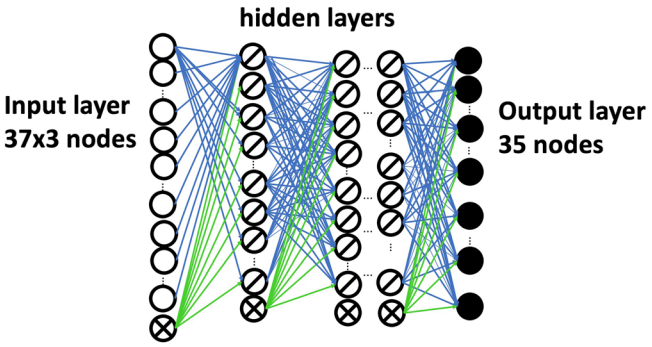

FIG. 4. DNN architecture. The circles are for input nodes, crossed circles are for bias nodes, single cross circles are for hidden layer nodes, and shaded circles are for output nodes. The lines are the weights and biases.

100] has three hidden layers with 100 nodes in each hidden layer. Note that one can design a DNN with different number of nodes in each hidden layer. We use the uniform number of nodes used in Table I to make the analysis tractable.

Except for the input layer nodes and the biases, all the other nodes are equipped with an activation function. For the hidden layer nodes, we use the rectified linear unit (ReLU) such that for the nth node,

$$\mathrm{ReLU}(z_n^{(N+1)}) = \max(0, z_n^{(N+1)}), \tag{8}$$

where $z_n^{(N+1)}$ is the linear combination of all the Nth layer node values multiplied by the appropriate weights and shifted by the bias of the same layer. For the output layer, it is optimal to use the softmax given by

$$\mathrm{softmax}(z_n^{(L+1)}) = \frac{\exp(z_n^{(L+1)})}{\sum_m^{N_{L+1}} \exp(z_m^{(L+1)})}, \tag{9}$$

where L is the last hidden layer. The softmax cross entropy is the appropriate cost function for a general classification problem. This is given by

TABLE II. Hidden layer architectures of DNN models considered in this study. The input and output layer nodes are identical for all models.

DNN model label	Hidden layer architecture
1	[200-200]
2	[200-200-200]
3	[200-200-200-200]
4	[100-100]
5	[100-100-100]
6	[100-100-100-100]

$$C(\mathbf{w}, \mathbf{b}) = \frac{1}{X} \sum_{\vec{x}} \vec{a}(\vec{x}) \cdot \log[\vec{y}_{\mathbf{w},\mathbf{b}}(\vec{x})], \tag{10}$$

where X is the total number of items in training set, $\vec{x}$ is an array containing the input-node values, $\vec{a}(\vec{x})$ is an array that corresponds to the true label of input $\vec{x}$ and $\vec{y}_{\mathbf{w},\mathbf{b}}(\vec{x})$ is the output of the DNN with weights and biases $(\mathbf{w}, \mathbf{b})$. The goal of training is to find the optimal $(\mathbf{w}, \mathbf{b})$ that minimizes the cost $C(\mathbf{w}, \mathbf{b})$. The construction of the DNN models and the execution of training loop are all done in Chainer [37–40].

In a typical training loop, we feed the teaching dataset using some mini-batch procedure to add stochasticity in estimating the cost-function. Then, we execute the optimization of weights and biases using some variant of stochastic gradient descent [41]. We perform these procedures on all our models and found that none is learning the classification problem. In particular, the training and testing accuracies remain at around 2.86% no matter how many training epochs we use. The observed accuracy is, in fact, the accuracy that we will get if we make a random guess out of 35 possibilities. We tried different optimizers and different combinations of hyperparameters shown in Table III but still obtained a nonimproving performance.

The difficulty in learning the classification problem is due to the noise introduced in the generation of the training dataset. This noise includes the random choice of energy points in evaluating amplitudes and the randomly added unitary background poles. It is often advisable to use a dataset with less noise to improve the training performance [42], but this defeats the purpose of simulating the energy resolution in the experimental data. The inclusion of energy resolution in the design of DNN inevitably results in a noisy dataset. Thus, we resort to a different approach to initiate the learning process without tampering with our teaching dataset. In the following, we introduce the idea of the curriculum method and how it can start the learning of a complicated classification problem.

C. Curriculum method

The curriculum method was first developed in Ref [43] for simple cases. Depending on the difficulties of classification problems, a more rigorous treatment requires a

TABLE III. List of optimizers and hyperparameters used in the noncurriculum training. For further descriptions see Ref. [40].

Optimizers	Adam, AdaDelta, AdaGrad AMSGrad, AdaBound, AMSBound
Minibatch sizes	32, 64, 512, 1024, 1536 2048, 2560, 4096
Weight initializers	Normal, HeNormal, Uniform HeUniform, Orthogonal
Others	with dropout, without dropout

well-organized training dataset [44–46]. For our purpose, it is sufficient to adopt a more heuristic approach where we subjectively identify the simplest dataset and introduce new classification until we present all the training sets. With the 35 pole-configuration classifications, four of which corresponds to at-most-one-pole configuration (labels 0, 1, 11, and 21 of Table I). We treat these four classes as the simplest examples and call them curriculum 1. Then we add one of the two-pole classifications and call the new set curriculum 2. We do this until we have included all the 35 classifications in the final curriculum 32. Table IV shows the incremental progression of the curriculum training. Note that by the end of curriculum 7, we have introduced all the two-pole configurations. Similarly, all the three-pole configurations are presented at curriculum 17 and all four-pole configurations at curriculum 32. The code used to build each curriculum is in the public repository [47].

TABLE IV. Scheme that we used in the curriculum learning stage. One new classification label is added in the subsequent curriculum. The label descriptions are given in Table I.

Curriculum label	Dataset addition scheme	Configuration presented
1	Labels 0,1,11,21	At-most one pole
2	Curriculum 1 + label 2	
3	Curriculum 2 + label 12	
4	Curriculum 3 + label 22	
5	Curriculum 4 + label 10	
6	Curriculum 5 + label 20	
7	Curriculum 6 + label 30	At-most two poles
8	Curriculum 7 + label 3	
9	Curriculum 8 + label 13	
10	Curriculum 9 + label 23	
11	Curriculum 10 + label 6	
12	Curriculum 11 + label 8	
13	Curriculum 12 + label 16	
14	Curriculum 13 + label 18	
15	Curriculum 14 + label 26	
16	Curriculum 15 + label 28	
17	Curriculum 16 + label 34	At-most three poles
18	Curriculum 17 + label 4	
19	Curriculum 18 + label 14	
20	Curriculum 19 + label 24	
21	Curriculum 20 + label 5	
22	Curriculum 21 + label 7	
23	Curriculum 22 + label 9	
24	Curriculum 23 + label 15	
25	Curriculum 24 + label 17	
26	Curriculum 25 + label 19	
27	Curriculum 26 + label 25	
28	Curriculum 27 + label 27	
29	Curriculum 28 + label 29	
30	Curriculum 29 + label 31	
31	Curriculum 30 + label 32	
32	Curriculum 31 + label 33	At-most four poles

We trained all the six models using curriculum 1 for the first 300 epochs, curriculum 2 for the subsequent 300 epochs, and then curriculum 3 until epoch 650. The goal here is to find which architecture performs best and then devote the rest of the training to the chosen DNN. Figure 5 shows the performance of the six DNNs. All models give decent accuracies at the start of curriculum 1, indicating that the chosen simple classification is indeed learnable. There is a noticeable accuracy drop at the start of curriculum 2 and curriculum 3 due to the introduction of a new classification set. Nevertheless, we can perform a few more training epochs within the curriculum to improve accuracy. The essential point here is that all our models start to learn the classification problem using the curriculum approach. Of all the six models, the architecture with a hidden layer [200-200-200] shows a promising performance, especially at the onset of curriculum 3. We, therefore, devote the rest of our computing resources to DNN model 2.

Note that it is not practical to perform more epoch per curriculum since the accuracy will definitely drop at the start of a new curriculum. It will suffice to have some decent accuracy for each curriculum and continue the training after introducing all the classifications. Hence, we restart the training of the chosen DNN and, to accelerate the process, we only use 100 epochs per curriculum which are then continued after all the pole configurations are introduced. The training performance is shown in Fig. 6. We also vary the minibatch size from 512 in the early part of the curriculum epoch, so we can take advantage of large stochasticity to 4,608 in the later part to stabilize the accuracy. At the end of epoch 3,200 (end of curriculum 32), we now have a DNN model that can detect up to four poles in any Riemann sheet with training and testing accuracies of 63.5% and 68.3%, respectively. We continue until epoch 31,050 using the typical training loop and obtained a final training and testing accuracies of 76.5% and 80.4%, respectively.

Recall that in the construction and training of our DNN model, only the energy resolution is included and not the fluctuation of the amplitudes, which correspond to statistical errors in experimental data. The goal of using a generic S-matrix is to teach the DNN to recognize only those amplitudes that satisfy the unitarity, analyticity, and Hermiticity requirements. This restriction is justified because we expect the actual experimental data to conform to the mentioned general properties. Inserting a random error or offset in the training amplitudes will violate such requirements. In the inference stage, the trained DNN will reinterpret the input experimental amplitudes as if they satisfy the expected properties even if there are some offsets. This feature is the advantage of having a model with many parameters (weights and biases) where the trained DNN automatically discards the irrelevant peculiarities of the input data. We further describe the inference stage in the next section.

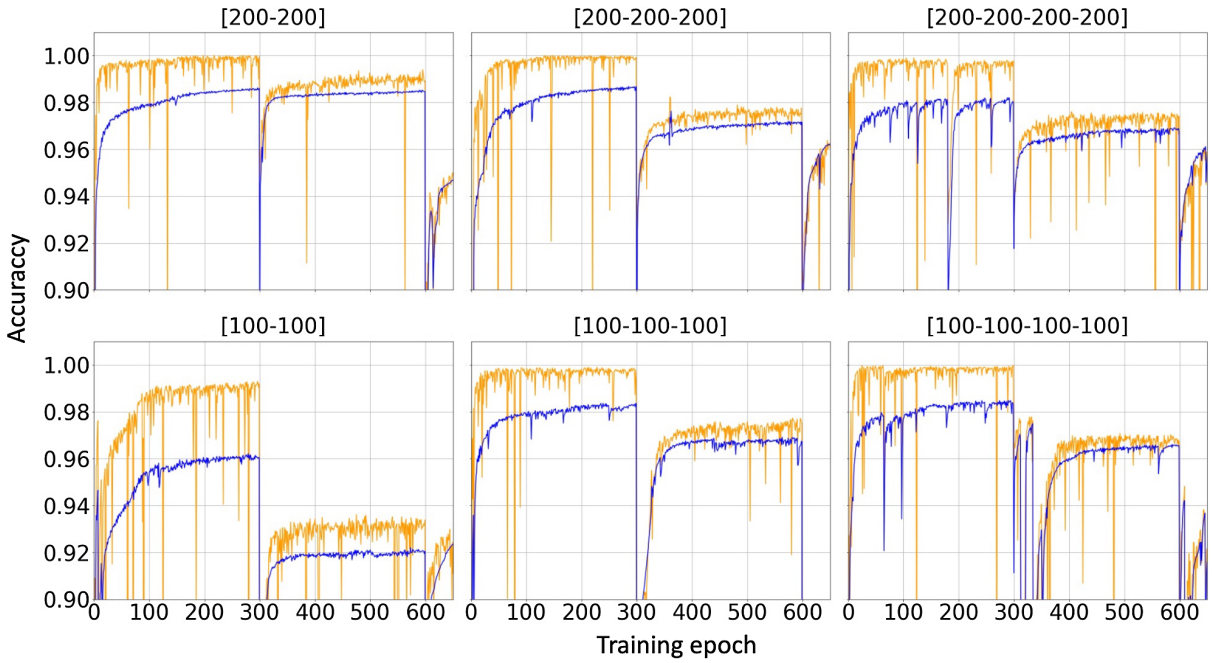

FIG. 5. Performance of the DNN models for the first three curricula. The blue lines are for the training performance while the orange lines are for the testing performance.

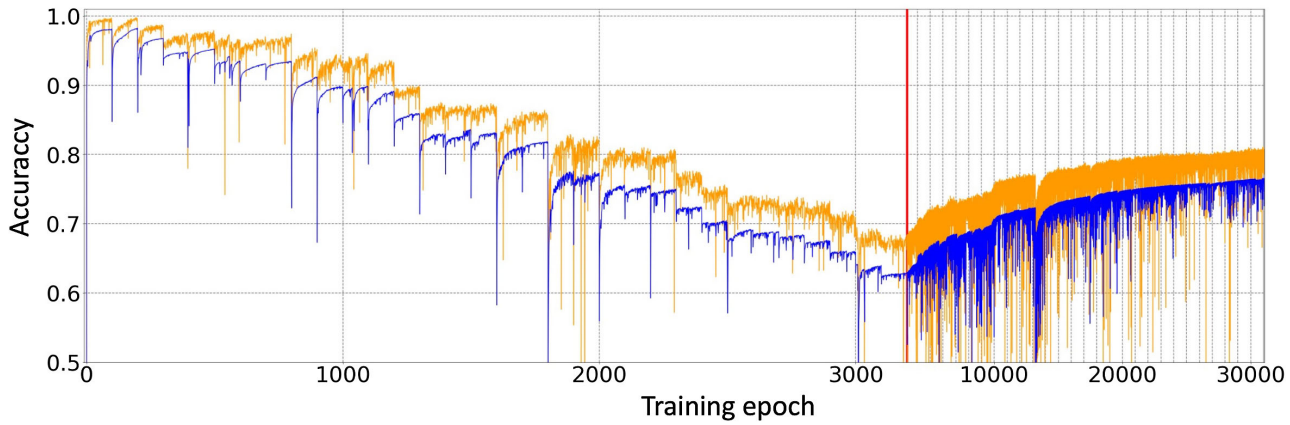

FIG. 6. Training (blue) and testing (orange) performance of our DNN model. Curriculum learning is used until epoch 3200 (red vertical solid line). The scale of horizontal axis is changed to epoch/20 after the curriculum learning at the vertical red line.

IV. APPLICATION TO πN SCATTERING

We can now use our trained DNN to make inferences on the experimental data. Due to the presence of error bars of the data, we can expect that there are multiple possible interpretations associated with the amplitude of interest. The result of DNN inferences must reflect the uncertainty due to the error bars in the experimental data. We can accomplish this by combining points in each error bar to produce several amplitudes. Here, we can further assume that some probability distribution weights the points in each error bar. Except for the assumed probability distribution, the generated amplitudes for the DNN inference comes directly from the experimental data without imposing anything.

A. Interpreting the error bar

We usually interpret each error bar to be 1 standard deviation σ of the Gaussian distribution around the central value. Therefore, we can generate 10^6 amplitudes directly from the experimental data by drawing points in each error bar using the Gaussian distribution. Note that the way we interpret the error bar corresponds to the confidence level that the points used to generate the amplitudes are within the error bar of the data. For example, increasing the σ to 2σ

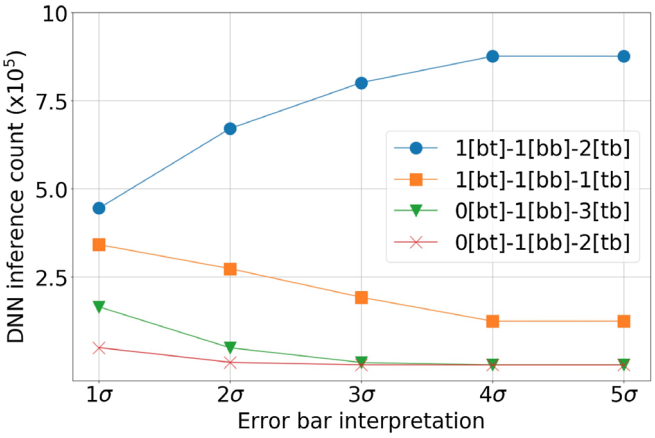

FIG. 7. DNN inference on amplitudes generated from experimental data with points drawn from Gaussian distribution in each error bar. The legend shows the identified pole configurations by the DNN.

means that we increase the confidence level from 68% to 95%, and so on. In Fig. 7 we use different interpretations of the error bar in the generation of 10^6 amplitudes and then count the number of DNN inference output. The result shows that out of 35 possibilities, only four-pole configurations are identified by the DNN. These detected configurations are shown in the legend of Fig. 6. Notice that as we increase the confidence level of the error bar, the most likely configuration emerges. Specifically, we find that the structures in the elastic πN amplitude are caused by one pole in $[bt]$, one pole in $[bb]$, and two poles in $[tb]$. The other three configurations are suppressed as the confidence level is increased, implying that these three configurations correspond to amplitudes produced by points outside the error bars.

One advantage of our approach is that we can go beyond the typical Gaussian distribution of points in each error bar and use other distributions. In particular, all the points in each error bar may be equally important, and a uniform distribution might be appropriate. In this way, all points used to generate the amplitudes are guaranteed to remain within the error bar. The DNN inference result, with the uniformly distributed points in each error bar, is shown in Table V. It is interesting to note that changing the weights at which the points are combined to produce the amplitudes does not affect the DNN inference very much. That is, we still get the same conclusion that the πN amplitude is best described by one pole in each adjacent Riemann sheet, one at most two poles in the distant sheet. Our approach contrasts with the conventional model-fitting scheme, where only a selected region of each error bar is used to describe the data. Thus, our deep learning approach is statistically robust compared to the conventional model-fitting scheme.

TABLE V. Result of the DNN inferences on the GW-SAID πN amplitude. The points used to generate the 10^6 experimental amplitudes are drawn in each error bar using a uniform distribution.

Percentage	bt	bb	tb
60.3%	1	1	2
30.9%	1	1	1
7.5%	0	1	3
1.3%	0	1	2

B. Discussion of results

Note that we made no assumptions on the poles detected by the DNN. At this point, one can now use a model to interpret the origin of the detected poles. Here, we give some general statements based on the expected effects of Riemann sheet poles on the scattering amplitude. First, the most prominent structure in Fig. 1 is the enhancement between the $K\Lambda$ and $K\Sigma$ thresholds. For the two-channel analysis, such enhancement can only be produced by at least one $[bb]$ pole above the second threshold. Meaning, we can associate this enhancement to the $[bb]$ pole that our trained DNN consistently identifies.

The next noticeable structure is around the ηN threshold. As an intuitive interpretation, one may associate the detected $[bt]$ pole to this near-ηN structure. However, a near-threshold $[bt]$ pole is expected to give rise to an amplitude peak below the threshold with an imaginary part close to unity, i.e., reaching the unitarity limit. This description is not the case with the near-ηN enhancement. Also, notice that the DNN sometimes identifies the $[bt]$ pole as a $[tb]$ pole (compare the first and the third rows of Table V). The comparison suggests that the $[bt]$ pole is near the $[bt]-[tb]$ interface which is located above the ηN threshold [see Fig. 2 (b)]. The detected $[bt]$ pole may be associated with the enhancement between the $K\Lambda$ and $K\Sigma$ thresholds in the form of a shadow pole. Thus, we can only attribute the ηN threshold enhancement to a non-$[bt]$ sheet pole.

The enhancement around ηN threshold could be due to the threshold cusp effect. However, to make the structure prominent, some nearby poles must be associated with it. Notice that our trained DNN consistently detected $[tb]$ poles in all of its inferences. With the $[bb]$ pole already allocated to the other enhancement and with $[bt]$ pole ruled out, we are only left with at most two poles in the $[tb]$ sheet to account for the ηN threshold enhancement. The limitation imposed on our counting region guarantees that at least one of the detected $[tb]$ poles is close to the ηN threshold. The peak structure slightly above the ηN threshold suggests that one of the $[tb]$ poles is close to the $[bb]-[tb]$ interface.

We can only make further statements about the origin of the detected poles by appealing to some dynamical model. Nevertheless, we can give some general speculations on the nature of the detected poles. It might be the case that in the zero-coupling limit of πN and ηN channels, the detected $[bb]$ and the $[bt]$ poles move towards a common position

resulting in a typical Breit-Wigner resonance of the lower channel. Such is a typical feature of a pole-shadow pair. On the other hand, in the zero coupling limit, one of the $[tb]$ poles go to the $[bb]$ sheet and approaches the same position as the other $[tb]$ pole. In this way, the pole decouples the lower channel and results in either a near-threshold resonance or just a virtual state of the higher channel. Note that a dynamical model is needed to verify the above speculations.

V. CONCLUSION AND OUTLOOK

We have demonstrated how to design a DNN that can extract the pole configuration of a given scattering amplitude. Using a generic S-matrix, we can produce a sizable teaching dataset to train our model to detect the number of poles in each Riemann sheet. Pole generation using Eqs. (2)–(4) gives us the advantage of performing deep learning analysis on any elements of the S-matrix. This prescription allows us to use the method on any scattering processes such as $2 \to 2$ or $1 \leftrightarrow 2$. Our approach can be extended to higher channels by putting an extra $\lambda[(p - i\beta)^2 - \alpha^2]$ term. Here, the new parameter λ can be adjusted to control the new shadow produced by the new channel. Additionally, the uniformization method introduced in Refs. [48,49] also provides an alternative way to control the poles and generate the training dataset. One possible advantage of uniformization is that it can be extended to relativistic scattering; this will be considered elsewhere.

Also, we have shown the effectiveness of the curriculum method in initiating the learning process with a noisy dataset. The curriculum method allows us to accommodate the limited energy resolution in the design of DNN. We also provided a method of utilizing the error bars in the experimental data in a statistically robust way. By generating several experimental amplitudes, we can make a reasonable interpretation of the data based on the most consistent inference of the DNN.

Finally, our proposed S-matrix treats the experimental results in a model-independent way. The poles produced for the training dataset are independent of each other and are not constrained by any *a priori* trajectory. The above implies that the trained DNN makes no assumptions in detecting the poles associated with the experimental data. It is now up to some dynamical model to interpret which set of poles are supposed to be paired as pole-shadow partners or which one is independent of the other. Upon obtaining the pole-configuration using a model-independent analysis, one can now design an appropriate parametrization to extract the relevant pole parameters.

ACKNOWLEDGMENTS

This study was supported in part by MEXT as Program for Promoting Researches on the Supercomputer Fugaku (Simulation for basic science: from fundamental laws of particles to creation of nuclei). D. L. B. S. is supported in part by the DOST-SEI ASTHRDP postdoctoral research fellowship. Y. I. is partly supported by JSPS KAKENHI No. JP17K14287 (B) and No. 21K03555 (C). A. H. is supported in part by JSPS KAKENHI No. JP17K05441 (C) and Grants-in-Aid for Scientific Research on Innovative Areas, No. 18H05407 and No. 19H05104.

APPENDIX: ARBITRARY TRAJECTORY TO DECOUPLE TWO CHANNELS

We show that an irrelevant independent pole can remove the cusp at the higher threshold of a coupled channel scattering. Note that the origin of a threshold cusp is purely kinematical. This will occur at $E = T_2$ when the amplitude has an explicit dependence on p_2, where $p_2 \propto \sqrt{E - T_2}$. If we can manage to remove the p_2 (like turning off γ_2 in two-channel Breit-Wigner) or turn p_2 into some analytic function of E in the amplitude, then the cusp singularity should disappear.

Let the relevant pole E_{pole} in, say, $[bb]$ sheet (i.e., $\text{Re}E_{\text{pole}} > T_2$) satisfies the condition,

$$F(E_{\text{pole}}) + ig_1(E_{\text{pole}})p_1 + ig_2(E_{\text{pole}})p_2 = 0,$$

where F, g_1, g_2 are analytic functions of E. We can always find F, g_1, g_2 such that the nearest singularity to the scattering region is an isolated simple pole. Since E_{pole} is in $[bb]$, then we can express the momentum poles (p_1, p_2) as

$$p_1 = -i\beta_1 \pm \alpha_1$$
$$p_2 = -i\beta_2 \pm \alpha_2,$$

where we set α_1, α_2, β_1 and β_2 to be positive. It is understood that p_1 and p_2 are related to the E_{pole} via the energy constraint in Eq. (1). It follows that, the modified equation,

$$F(E_{\text{pole}}) + ig_1(E_{\text{pole}})p_1 - ig_2(E_{\text{pole}})p_2 = 0.$$

is satisfied by the same $E = E_{\text{pole}}$ but in the $[bt]$ sheet, i.e., the momentum poles are

$$p_1 = -i\beta_1 \pm \alpha_1$$
$$p_2 = i\beta_2 \mp \alpha_2.$$

Now, consider the S-matrix element $S_{11}(p_1, p_2)$ given by

$$S_{11}(p_1, p_2) = \left(\frac{F - ig_1p_1 + ig_2p_2}{F + ig_1p_1 + ig_2p_2}\right) \times \left(\frac{F - i\bar{g}_1p_1 - i\bar{g}_2p_2}{F + i\bar{g}_1p_1 - i\bar{g}_2p_2}\right).$$

The above form is consistent with $S_{11}(p_1, p_2) = D(-p_1, p_2)/D(p_1, p_2)$ in Eq. (2). From the previous discussion, we know that the first parenthetical factor will generate a relevant pole E_{pole} in $[bb]$ sheet.

Deform $\bar{g}_1$ and $\bar{g}_2$ arbitrarily such that $\bar{g}_1 \to g_1$ and $\bar{g}_2 \to g_2$, respectively. The second factor will produce an irrelevant pole at $E = E_{\text{pole}}$ in $[bt]$ sheet. Using this limiting procedure, the S-matrix element becomes

$$S_{11}(p_1, p_2) = \frac{(F - ig_1p_1)^2 + (g_2p_2)^2}{(F + ig_1p_1)^2 + (g_2p_2)^2}.$$

The explicit dependence of $S_{11}(p_1, p_2)$ on p_2 is now replaced with p_2^2. We still have a peak at around $E = \text{Re}E_{\text{pole}}$ due to the relevant pole since the denominator did not cancel out as we perform the limiting procedure. However, the amplitude will no longer have a branch cut or threshold cusp at $E = T_2$ due to the absence of explicit p_2 dependence. Furthermore, from Eq. (2), the limit $\bar{g}_1 \to g_1$ and $\bar{g}_2 \to g_2$, will give us $S_{22} \to 1$ and $S_{12}^2 \to 0$. This means that putting an arbitrary irrelevant pole in the same position as the main pole, but on a different Riemann sheet, effectively decouples the two channels.

[1] S. L. Olsen, T. Skwarnicki, and D. Zieminska, Nonstandard heavy mesons and baryons: Experimental evidence, Rev. Mod. Phys. **90**, 015003 (2018).

[2] F.-K. Guo, C. Hanhart, U.-G. Meißner, Q. Wang, Q. Zhao, and B.-S. Zou, Hadronic molecules, Rev. Mod. Phys. **90**, 015004 (2018).

[3] R. Aaij *et al.* (LHCb Collaboration), Observation of a Narrow Pentaquark State, $P_c(4312)^+$, and of Two-Peak Structure of the $P_c(4450)^+$, Phys. Rev. Lett. **122**, 222001 (2019).

[4] R. Aaij *et al.* (LHCb Collaboration), Observation of structure in the J/ψ -pair mass spectrum, Sci. Bull. **65**, 1983 (2020).

[5] E. Wang, W.-H. Liang, and E. Oset, Analysis of the $e^+e^- \to J/\psi D\bar{D}$ reaction close to the threshold concerning claims of a $\chi_{c0}(2P)$ state, Eur. Phys. J. A **57**, 38 (2021).

[6] J. Haidenbauer and U. G. Meißner, On the structure in the ΛN cross section at the ΣN threshold, arXiv:2105.00836.

[7] B.-S. Zou, Building up the spectrum of pentaquark states as hadronic molecules, Sci. Bull. **66**, 1258 (2021).

[8] J. Taylor, *Scattering Theory: Quantum Theory on Nonrelativistic Collisions* (Wiley, New York, 1972).

[9] R. G. Newton, *Scattering Theory of Waves and Particles*, Theoretical and Mathematical Physics (Springer-Verlag, Berlin Heidelberg, 1982).

[10] R. Eden, P. V. Landshoff, D. I. Olve, and J. C. Polkinghorne, *The Analytic S-Matrix*, 2nd ed. (Cambridge University Press, London, 1966).

[11] C. Fernández-Ramírez, A. Pilloni, M. Albaladejo, A. Jackura, V. Mathieu, M. Mikhasenko, J. A. Silva-Castro, and A. P. Szczepaniak (JPAC Collaboration), Interpretation of the LHCb $P_c(4312)^+$ Signal, Phys. Rev. Lett. **123**, 092001 (2019).

[12] X.-K. Dong, V. Baru, F.-K. Guo, C. Hanhart, and A. Nefediev, Coupled-Channel Interpretation of the LHCb Double- J/ψ Spectrum and Hints of a New State Near the $J/\psi J/\psi$ Threshold, Phys. Rev. Lett. **126**, 132001 (2021).

[13] V. D. Burkert and T. S. H. Lee, Electromagnetic meson production in the nucleon resonance region, Int. J. Mod. Phys. E **13**, 1035 (2004).

[14] S.-Q. Kuang, L.-Y. Dai, X.-W. Kang, and D.-L. Yao, Pole analysis on the hadron spectroscopy of $\Lambda_b \to J/\Psi pK^-$, Eur. Phys. J. C **80**, 433 (2020).

[15] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[16] D. L. B. Sombillo, Y. Ikeda, T. Sato, and A. Hosaka, Classifying the pole of an amplitude using a deep neural network, Phys. Rev. D **102**, 016024 (2020).

[17] D. L. B. Sombillo, Y. Ikeda, T. Sato, and A. Hosaka, Classifying near-threshold enhancement using deep neural network, Few-Body Systems **62**, 52 (2021).

[18] D. Bourilkov, Machine and deep learning applications in particle physics, Int. J. Mod. Phys. A **34**, 1930019 (2019).

[19] D. Morgan, Pole counting and resonance classification, Nucl. Phys. **A543**, 632 (1992).

[20] D. Morgan and M. R. Pennington, New data on the $K\bar{K}$ threshold region and the nature of the $f_0(S^*)$, Phys. Rev. D **48**, 1185 (1993).

[21] T. Hyodo, S. I. Nam, D. Jido, and A. Hosaka, Flavor SU(3) breaking effects in the chiral unitary model for meson baryon scatterings, Phys. Rev. C **68**, 018201 (2003).

[22] V. K. Magas, E. Oset, and A. Ramos, Evidence for the Two Pole Structure of the $\Lambda(1405)$ Resonance, Phys. Rev. Lett. **95**, 052301 (2005).

[23] Z.-Y. Wang, H. A. Ahmed, and C. W. Xiao, Is two-poles' $\Lambda(1405)$ one state or two?, arXiv:2106.10511.

[24] R. L. Workman, R. A. Arndt, W. J. Briscoe, M. W. Paris, and I. I. Strakovsky, Parameterization dependence of T matrix poles and eigenphases from a fit to πN elastic scattering data, Phys. Rev. C **86**, 035202 (2012).

[25] R. A. Arndt, W. J. Briscoe, I. I. Strakovsky, and R. L. Workman, Extended partial-wave analysis of πN scattering data, Phys. Rev. C **74**, 045205 (2006).

[26] http://gwdac.phys.gwu.edu/analysis/pin_analysis.html.

[27] H. Kamano, S. X. Nakamura, T. S. H. Lee, and T. Sato, Nucleon resonances within a dynamical coupled-channels model of πN and γN reactions, Phys. Rev. C **88**, 035209 (2013).

[28] B. C. Pearce and B. F. Gibson, Observable effects of poles and shadow poles in coupled-channel systems, Phys. Rev. C **40**, 902 (1989).
[29] N. G. van Kampen, *S* matrix and causality condition. II. Nonrelativistic particles, Phys. Rev. **91**, 1267 (1953).
[30] L. D. Landau and E. M. Lifshitz, *Quantum Mechanics Non-Relativistic Theory*, 3rd ed. (Pergamon, New York, 1977).
[31] R. G. Newton, Structure of the many channel S-matrix, J. Math. Phys. (N.Y.) **2**, 188 (1961).
[32] K. J. L. Couteur and R. E. Peierls, The structure of a non-relativistic S-matrix, Proc. R. Soc. A. **256**, 115 (1960).
[33] N. Suzuki, T. Sato, and T. S. H. Lee, Extraction of resonances from meson-nucleon reactions, Phys. Rev. C **79**, 025205 (2009).
[34] A. M. Badalyan, L. P. Kok, M. I. Polikarpov, and Y. A. Simonov, Resonances in coupled channels in nuclear and particle physics, Phys. Rep. **82**, 31 (1982).
[35] R. J. Eden and J. R. Taylor, Poles and shadow poles in the many-channel *S* matrix, Phys. Rev. **133**, B1575 (1964).
[36] W. R. Frazer and A. W. Hendry, *S*-matrix poles close to threshold, Phys. Rev. **134**, B1307 (1964).
[37] S. Tokui, K. Oono, S. Hido, and J. Clayton, Chainer: A next-generation open source framework for deep learning, in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)* (2015).
[38] T. Akiba, K. Fukuda, and S. Suzuki, ChainerMN: Scalable distributed deep learning framework, in *Proceedings of Workshop on ML Systems in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)* (2017).
[39] S. Tokui, R. Okuta, T. Akiba, Y. Niitani, T. Ogawa, S. Saito, S. Suzuki, K. Uenishi, B. Vogel, and H. Yamazaki Vincent, Chainer: A deep learning framework for accelerating the research cycle, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, 2019), pp. 2002–2011.
[40] https://github.com/chainer/chainer.
[41] D. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[42] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook* (Springer International Publishing AG, Springer Nature, 2018).
[43] J. L. Elman, Learning and development in neural networks: The importance of starting small, Cognition **48**, 71 (1993).
[44] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, Curriculum learning, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09 (Association for Computing Machinery, New York, NY, USA, 2009), p. 4148.
[45] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, Automated curriculum learning for neural networks, in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 70 (PMLR, International Convention Centre, Sydney, Australia, 2017), pp. 1311–1320.
[46] G. Hacohen and D. Weinshall, On the power of curriculum learning in training deep networks, arXiv:1904.03626.
[47] https://github.com/sombillo/DNN-pole-configuration.
[48] W. Yamada and O. Morimatsu, New method to extract information of near-threshold resonances: Uniformized Mittag-Leffler expansion of Green's function and *T* matrix, Phys. Rev. C **102**, 055201 (2020).
[49] W. A. Yamada and O. Morimatsu, Application of the uniformized Mittag-Leffler expansion to $\Lambda(1405)$, Phys. Rev. C **103**, 045201 (2021).