MIT Libraries | DSpace@MIT

MIT Open Access Articles

Visualizing a neural network that develops quantum perturbation theory

The MIT Faculty has made this article openly available. ***Please share*** how this access benefits you. Your story matters.

Citation: Wu, Yadong, Pengfei Zhang, Huitao Shen and Hui Zhai. "Visualizing a neural network that develops quantum perturbation theory." Physucial Review A 98 (2018), 010701.

As Published: http://dx.doi.org/10.1103/PhysRevA.98.010701

Publisher: American Physical Society

Persistent URL: http://hdl.handle.net/1721.1/117207

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

Massachusetts Institute of Technology

Rapid Communications

Visualizing a neural network that develops quantum perturbation theory

Yadong Wu,[1] Pengfei Zhang,[1] Huitao Shen,[2] and Hui Zhai[1,3]
[1]*Institute for Advanced Study, Tsinghua University, Beijing 100084, China*
[2]*Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[3]*Collaborative Innovation Center of Quantum Matter, Beijing 100084, China*

(Received 14 March 2018; published 30 July 2018)

Motivated by the question whether the empirical fitting of data by neural networks can yield the same structure of physical laws, we apply neural networks to a quantum-mechanical two-body scattering problem with short-range potentials—a problem that by itself plays an important role in many branches of physics. After training, the neural network can accurately predict s-wave scattering length, which governs the low-energy scattering physics. By visualizing the neural network, we show that it develops perturbation theory order by order when the potential depth increases, without solving the Schrödinger equation or obtaining the wave function explicitly. The result provides an important benchmark to the machine-assisted physics research or even automated machine learning physics laws.

DOI: 10.1103/PhysRevA.98.010701

Introduction. Human physicists have made great achievements in discovering laws of physics during the last several centuries. Based on experimental observations, they interpret data and build theories using logical inference and rigorous mathematical reasoning. Artificial intelligence, which has recently been extensively applied to physics research [1–37], interprets data empirically by, for example, fitting on statistical models.

While physicists hope that the artificial intelligence could help discover new physics in unexplored areas, there are certainly challenges to overcome due to the different nature of "knowledge" among human physicists and artificial intelligence. First, the great complexity of machine learning models, such as deep neural networks with more than millions of fitting parameters, makes it difficult for humans to understand whether machines have successfully captured the underlying laws of physics. Second, despite the great success of artificial intelligence in many disciplines outside physics [38], its weakness [39,40] leads people to believe that it lacks the ability of sophisticated reasoning [41]. Therefore, it is important to ask the question whether these two approaches of distilling knowledge from data will yield the same answer when facing the same problem. The positive answer will mark an important step forward to the machine-assisted physics research or even automated machine learning physics laws.

Motivated by this question, we let the machine supervisedly learn a quantum-mechanical scattering problem. We show that the machine could not only correctly produce the scattering length for a given interaction potential, but also give a human tractable decision-making process. More concretely, the machine will develop perturbation theory order by order with increasing the potential strength.

Two-body scattering problem. The scattering problem plays an important role in many branches of physics, including atomic and molecular physics, nuclear physics, and particle physics [42]. In this work, we focus on the two-body scattering problems in three dimensions with spherical potentials. Consider the Schrödinger equation in the relative frame

$$\left(-\frac{\hbar^2}{2\bar{m}}\nabla^2 + V(r)\right)\Psi = E\Psi, \tag{1}$$

where $\bar{m}$ is the reduced mass of two particles and $V(r)$ is the interaction potential. The wave function can be expanded through partial wave decomposition [43] as

$$\Psi(\mathbf{r}) = \sum_{l=0}^{+\infty} \frac{\chi_{kl}(r)}{kr}\mathcal{P}_l(\cos\theta), \tag{2}$$

where k is the wave vector and l labels different angular momentum. χ_{kl} is the radial wave function and $\mathcal{P}_l(\cos\theta)$ represents the angular part. For s-wave scattering, one can show that the asymptotic behavior of the radial wave function $\chi_{k,l=0} \propto \sin(kr + \delta_k)$. We can therefore determine the important quantity of the s-wave scattering length a_s as $\tan\delta_k = -ka_s + o(k^2)$ [43].

In this way, we establish a mapping between $V(r)$ and a_s, which is to be learned by the neural network. In the following, we first train neural networks with $V(r)$ as the input and a_s as the output. Here a_s is obtained by numerically solving the Schrödinger equation (see Appendices). In practice, a_s can be extracted from experimental data of low-energy collision. After training, the neural network can predict a_s directly from $V(r)$.

Perturbation theory. Before presenting the results of neural networks, let us first review a textbook method for computing the scattering length. One can compute the scattering amplitude $\mathcal{T}$ analytically through perturbative Born expansion [43]. In fact, the perturbation theory is by far the most well-established approach to compute observables in interacting quantum mechanics or quantum field theory. Particularly in our two-body scattering problem, $\mathcal{T}$ is given by $\mathcal{T} = V + VG_0V + VG_0VG_0V + \cdots$, where G_0 is the Green's function for free Hamiltonian with $V(r) = 0$. Keeping only the s-wave component and taking the low-energy limit, we obtain $a_s = 2\pi^2\langle 0|\mathcal{T}|0\rangle$, where $|0\rangle$ denotes the zero-momentum state.

 ©2018 American Physical Society

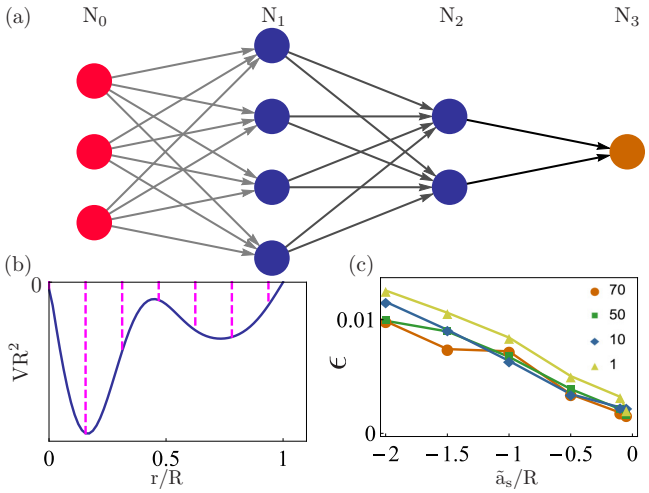

FIG. 1. (a) Schematic of the fully connected neural network. (b) Schematic of the input data, which is a finite-range potential discretized at N_0 different points. (c) The error ϵ of the neural network after training as a function of maximum scattering length $\tilde{a}_s$, with different number of neurons N_1 in the first hidden layer. See main text for the definition of ϵ and $\tilde{a}_s$.

Here and throughout the rest of this work, we take $m = \hbar = 1$ for simplicity. Combining these equations, we get $a_s = a_s^{(1)} + a_s^{(2)} + \cdots$, with

$$a_s^{(1)} = 2\pi^2 \langle 0|V|0\rangle = \int V(r) r^2 dr \tag{3}$$

as the first-order Born approximation and

$$a_s^{(2)} = 2\pi^2 \langle 0|V G_0 V|0\rangle = -\frac{1}{2}\int V(r)V(r')\mathcal{K}(r,r')dr\,dr' \tag{4}$$

as the second-order Born approximation. Here

$$\mathcal{K}(r,r') = rr'(r + r' - |r - r'|). \tag{5}$$

The Born expansion is based on the potential strength, and is a very good approximation for weak potentials.

Neural network. In order to keep the size of the input data finite, we consider short-range attraction potentials $V(r)$, i.e., $V(r) < 0$ for $r < R$ and $V(r) = 0$ for $r \geqslant R$. For simplicity, we restrict the depth of $V(r)$ to be before the first resonance so that $a_s < 0$. We first generate the potential randomly (see Appendices) and then discretize it uniformly along $r = 0$ to $r = R$ into N_0 pieces, as shown in Fig. 1(b). The input data is thus an N_0-dimensional vector

$$\mathbf{V} = \left[V\left(\frac{R}{N_0}\right), V\left(\frac{2R}{N_0}\right), \ldots, V(R)\right]^{\mathrm{T}}. \tag{6}$$

Before the first resonance, the larger the potential depth, the larger the absolute value of scattering lengths in general.

We use a fully connected neural network as our machine learning model, whose structure is schematically shown in Fig. 1(a). It is composed of three fully connected layers. The effect of the first (hidden) layer can be mathematically summarized as $\mathbf{Y} = f(W\mathbf{V} + \mathbf{B})$, where $\mathbf{V}$ is the N_0-dimensional input vector defined previously, $\mathbf{Y}$ is an N_1-dimensional output vector, W is an $N_1 \times N_0$ weight matrix, and $\mathbf{B}$ is an N_1-dimensional bias vector. f is the activation function that is applied elementwisely to the vector and is chosen to be rectified linear units $f(x) = \max\{0, x\}$. A similar hidden layer taking $\mathbf{Y}$ as its input comes after the first hidden layer. Its weight matrix is of size $N_2 \times N_1$, and its output is linearly mapped to the final single value output, which is interpreted as the scattering length a_s. In our work, we set $N_0 = 64$, $N_2 = 32$ fixed and change N_1 of the neural network.

We train the neural network by minimizing the mean squared error on 3×10^4 randomly generated $\{\mathbf{V}, a_s^{\mathrm{T}}(\mathbf{V})\}$ pairs in units of $\{1/R^2, R\}$, where $a_s^{\mathrm{T}}(\mathbf{V})$ is the scattering length obtained from solving the Schrödinger equation given discretized potential $\mathbf{V}$. In the training data set, $a_s^{\mathrm{T}}(\mathbf{V})$ is uniformly distributed in the range of $(\tilde{a}_s, 0)$. We set $L2$ regularization strength to be 0.1 to prove weight matrix visualization. After training, the neural network should be able to make a prediction of the scattering length, denoted as $a_s^{\mathrm{P}}(\mathbf{V})$, directly from the input potential $\mathbf{V}$. This prediction may deviate from the true scattering length $a_s^{\mathrm{T}}(\mathbf{V})$. We characterize the performance of the trained neural network by calculating the averaged error ϵ defined as

$$\epsilon = \frac{1}{N_t}\sum_l \left|\frac{a_s^{\mathrm{P}}(\mathbf{V}^l) - a_s^{\mathrm{T}}(\mathbf{V}^l)}{a_s^{\mathrm{T}}(\mathbf{V}^l)}\right|, \tag{7}$$

on a test set of similar $\{\mathbf{V}, a_s^{\mathrm{T}}(\mathbf{V})\}$ pairs. Here $\mathbf{V}^l$ denotes the lth potential in the test set and $N_t = 10^3$ is the size of the testing set. As shown in Fig. 1(c), the network can always predict the scattering length a_s with high accuracy for different $\tilde{a}_s$ and with different number of neurons in the first layer (denoted by N_1).

To understand how the neural network computes the scattering length, it is very informative to visualize the weight matrix W in the first hidden layer that maps the input data $\mathbf{V}$ to the first intermediate vector $\mathbf{Y}$. We plot W in Fig. 2 for

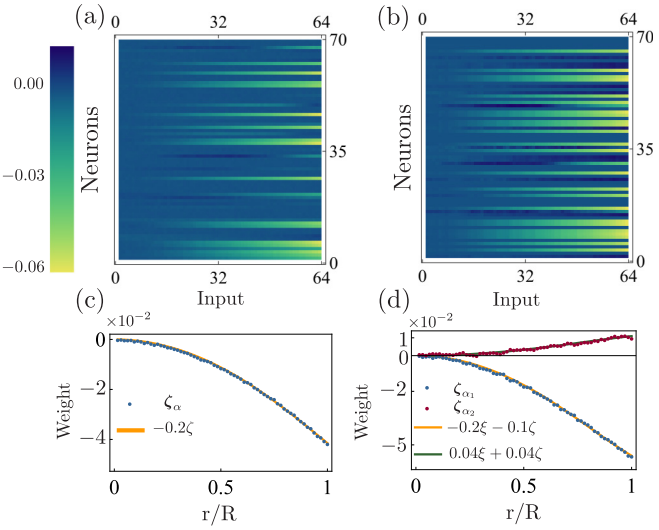

FIG. 2. Visualization of the matrix W [(a) and (b)] and a typical N_0-dimensional vector $\boldsymbol{\zeta}_\alpha = \{W_{\alpha i}, i = 1, \ldots, N_0\}$ as a function of $r/R = i/N_0$ [(c) and (d)], for $\tilde{a}_s/R = -0.1$ [(a) and (c)] and for $\tilde{a}_s/R = -0.5$ [(b) and (d)]. The solid line in (c) is a fitting of $\boldsymbol{\zeta}_{\alpha=46}$. The orange (lower) and green (upper) solid lines in (d) are fittings of $\boldsymbol{\zeta}_{\alpha_1=46}$ and $\boldsymbol{\zeta}_{\alpha_2=54}$ as linear combinations of $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$.

two neural networks trained on a very small magnitude of $\tilde{a}_s/R = -0.1$ and on a relatively large magnitude of $\tilde{a}_s/R = -0.5$, respectively. For both networks, most of their matrix elements automatically vanish and are thus not responsible for producing the scattering length. In the following, we analyze the remaining matrix elements for these two scenarios.

First-order Born approximation. Let us first focus on the case of shallow potentials. For convenience, we represent W using N_0-dimensional row vectors, known as "neurons": $W = (\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_{N_1})^{\mathrm{T}}$. The effect of the first (hidden) layer can then be represented as $Y_i = f(\boldsymbol{\zeta}_i^{\mathrm{T}}\mathbf{V} + B_i)$. The jth component of the ith neuron is denoted as $\zeta_{i,j}$. From Fig. 2(a), one can see that nearly all nonvanishing $\boldsymbol{\zeta}_\alpha$ behave similarly. A typical $\boldsymbol{\zeta}_\alpha$ is plotted in Fig. 2(c).

Importantly, we find that typical $\zeta_{\alpha,j}$ can be quite accurately described by $\sim j^2$. As shown in Fig. 2(c), $\boldsymbol{\zeta}_\alpha$ and normalized $\boldsymbol{\zeta} \propto (1, 4, \ldots, N_0^2)$ correspond very well. This means that the αth neuron in the first hidden layer performs the calculation

$$Y_\alpha = \sum_j \zeta_{\alpha,j} V_j \propto \sum_j j^2 V\left(\frac{jR}{N_0}\right). \tag{8}$$

This is precisely the discrete version of the first-order Born approximation shown in Eq. (3). If one uses a similar neural network to study scattering volume or supervolume for p-wave or d-wave scattering, one obtains that typical $\boldsymbol{\zeta}_\alpha$ for the shallow potential behaves as $\sim j^4$ or $\sim j^6$, respectively, which are both consistent with the first-order Born approximation (see Appendices).

Second-order Born approximation. As the potential depth increases, more features begin to emerge in W. There are many neurons whose behavior cannot be fitted by j^2 discussed in the previous section, shown in Figs. 2(b) and 2(d). It is natural to suspect that this new neuron pattern comes from the second-order Born approximation. To verify this, we first notice that the discretized version of Eq. (4) is

$$a_s^{(2)} = -\sum_{ij} V\left(\frac{iR}{N_0}\right) V\left(\frac{jR}{N_0}\right) K_{ij}, \tag{9}$$

where K is a $N_0 \times N_0$ matrix constructed as

$$K_{ij} = \frac{1}{N_0^3} ij(i + j - |i - j|). \tag{10}$$

K matrix can be diagonalized, and let us denote $\boldsymbol{\xi}_\alpha$ as its eigenvector associated with eigenvalue Λ_α. In this way, we can rewrite Eq. (9) as

$$a_s^{(2)} = -\mathbf{V}^{\mathrm{T}} K \mathbf{V} = -\sum_\alpha \Lambda_\alpha \left(\boldsymbol{\xi}_\alpha^{\mathrm{T}}\mathbf{V}\right)^2. \tag{11}$$

Moreover, since the largest eigenvalue Λ_1 is an order of magnitude larger than the second largest eigenvalue Λ_2, i.e., $\Lambda_1/\Lambda_2 \sim 16.38$, we can approximate Eq. (11) further by only keeping the contribution of the largest eigenvalue as

$$a_s^{(2)} \propto \left(\boldsymbol{\xi}_1^{\mathrm{T}}\mathbf{V}\right)^2. \tag{12}$$

Indeed, if we train the neural network with the same architecture as before, but using $a_s^{(2)}$ calculated using Eq. (4) as the output instead of the exact a_s, the neural network could compute $a_s^{(2)}$ accurately. In this new network, neurons in the first hidden layer have only one pattern, and can be well fitted by $\boldsymbol{\xi}_1$ defined above, as illustrated in Fig. 3. In this way, neurons in the first hidden layer compute $\boldsymbol{\xi}^{\mathrm{T}}\mathbf{V}$ with $\boldsymbol{\xi} \equiv \boldsymbol{\xi}_1$ and the second hidden layer simply fits a function x^2 in order to compute $a_s^{(2)}$.

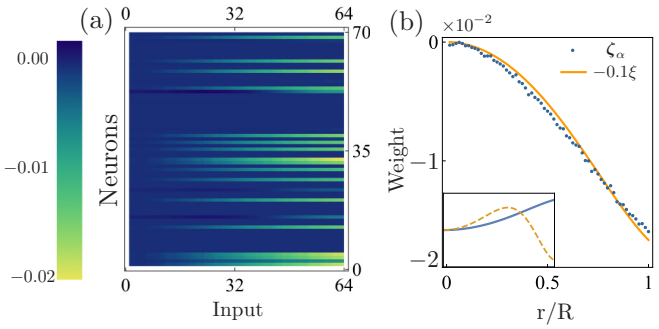

FIG. 3. Neural network with the same architecture as that in Fig. 2, but trained with $a_s^{(2)}$ as the output instead of a_s. (a) Visualization of the matrix W. (b) A typical vector $\boldsymbol{\zeta}_{\alpha=32}$ as a function of $r/R = i/N_0$ for $\tilde{a}_s/R = -0.5$. It is compared with the eigenvector of K matrix with the largest eigenvalue $\sim\boldsymbol{\xi}$ shown by the solid line. The inset of (b) is the eigenvectors associated with first two largest eigenvalues. For $N_0 = 64$, the blue solid curve shows $\sim\boldsymbol{\xi}_1$ with the largest eigenvalue $\Lambda_1 = 29.5$, and the orange dashed curve shows $\sim\boldsymbol{\xi}_2$ with the second largest eigenvalue $\Lambda_2 = 1.8$.

Equipped with the vector $\boldsymbol{\zeta}$ introduced for the first-order Born approximation and the vector $\boldsymbol{\xi}$ introduced for the second-order Born approximation, we find neurons in W with new features for a deeper potential can be well fitted by $\alpha\boldsymbol{\zeta} - b\boldsymbol{\xi}$, as shown in Fig. 2(d). In this way, in the first hidden layer, all ingredients for the first-order and the second-order Born approximation are already computed. With a proper polynomial function implemented in the second hidden layer, the neural network produces $a_s = a_s^{(1)} + a_s^{(2)}$ as its final output.

The key observations so far can be summarized as follows: (i) When the potential is shallow, the neural network only captures the first-order perturbation as it is already accurate enough to reach a certain level of accuracy. (ii) As the potential becomes deeper, the neural network gradually develops the structure to capture at least the second-order perturbation. It is quite remarkable that, by empirical fitting, the neural network develops the same kind of perturbation theory as that developed by human physicists. At least in this example, we understand how the neural network works and provides a positive answer to the question raised in the Introduction, namely, human physicists and neural networks can yield the same answer when facing the same problem.

An empirical formula. Last but not least, we bring out a different aspect: If one can tolerate a certain amount of error, the neural network can also inspire an empirical formula that works quite well even beyond the perturbative regime, although the formula lacks rigorous mathematical reasoning.

Here the formula is inspired by the fact that even if we reduce the first hidden layer to have only one neuron, the neural network can still predict scattering lengths quite accurately with errors only increasing by a small amount [Fig. 1(c)]. In Fig. 4(a), we compare W with $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$. After proper scaling, they actually behave similarly. This implies that they

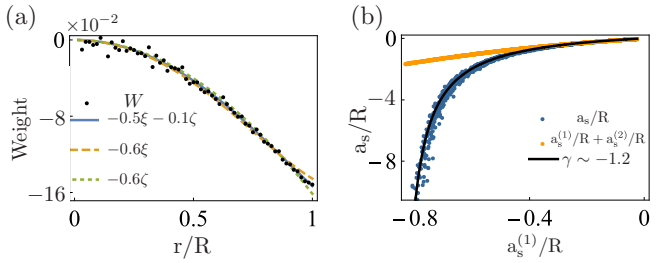

FIG. 4. (a) When the first layer only contains one neuron, W becomes an N_0-dimensional vector. Here we compare W and different linear combinations of $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$. (b) The relation between the actual scattering length a_s and $a_s^{(1)}$ from the first-order Born approximation. The black line is the empirical formula, Eq. (13), and the orange (light-gray) dots are scattering lengths computed from first two orders of Born approximation.

can be regarded as the same without introducing much error. Therefore, (i) if we replace W with $\boldsymbol{\zeta}$, this simple neural network produces nothing but $a_s^{(1)}$, and (ii) if we replace $\boldsymbol{\xi}$ also with $\boldsymbol{\zeta}$, by Eq. (12) $a_s^{(2)}$ can be represented by $a_s^{(1)}$ as $a_s^{(2)} \propto (a_s^{(1)})^2/R$. With these we have

$$a_s = a_s^{(1)} + \gamma\left(a_s^{(1)}\right)^2/R + O\left(\left(a_s^{(1)}/R\right)^3\right) \approx \frac{a_s^{(1)}}{1-\gamma a_s^{(1)}/R}, \tag{13}$$

where γ is a fitting parameter.

Equation (13) is our empirical formula for scattering lengths. It takes $a_s^{(1)}$ as the input and produces a_s as the output. To verify this formula, we first generate many interaction potentials randomly, then compute the corresponding a_s exactly from solving the Schrödinger equation, and $a_s^{(1)}$ from Eq. (3). If Eq. (13) is a good approximation, these data will collapse into a single curve that can be fitted by Eq. (13). This is indeed the case as is shown in Fig. 4(b), where the fitting parameter is taken to be $\gamma = -1.2$. Interestingly, Equation (13) works quite well even when a_s is an order of magnitude larger than $a_s^{(1)}$, that is to say, when the potential is deep enough to be beyond the weakly interacting regime. To compare, we also plot $\{a_s^{(1)} + a_s^{(2)}, a_s^{(1)}\}$ from the same set of potentials in Fig. 4(b). Clearly, the validity of the empirical formula, Eq. (13), goes significantly beyond the second-order Born approximation.

Outlook. In summary, we have trained a neural network to predict the s-wave scattering length directly from the interaction potential. By visualizing its weight matrix, we demonstrate that the neural network develops perturbation theory order by order as the interaction strength increases. Moreover, the performance of the neural network also inspires us to derive a simple approximate formula for scattering lengths that works well beyond the perturbative regime.

Guaranteed by the great expressibility of neural networks [44,45], our formalism can be generalized to study more complicated quantum few-body and many-body problems. The former is important in atomic and molecular physics, as well as quantum chemistry. The latter is related to machine learning phases of matter, which has already become a topic of intensive interest [1–20]. Our understanding of how neural network works in this simple problem can shed light on understanding more sophisticated problems better.

Acknowledgments. We thank Ning Sun, Ce Wang, and Jinmin Yi for helpful discussions. This work is supported by MOST under Grant No. 2016YFA0301600 and NSFC Grant No. 11734010. H.S. thanks IASTU for hosting his visit to Beijing, where the key parts of this work were done.

APPENDIX A: METHOD FOR GENERATING TRAINING DATA

In this Appendix we demonstrate in detail the method for generating our training data $\{\mathbf{V}, a_s(\mathbf{V})\}$.

Generating potentials. To extract the universal method for solving a physical problem, the shape of the potentials in the training data should be diverse enough, so that the machine will not learn features specific to some potential shape. Therefore, we construct random potentials using random walk.

Formally, $\tilde{V}(r)$, understood as a three-dimensional spherical potential, is constructed as

$$\tilde{V}(r) = \begin{cases} V_i, & r_i \leqslant r \leqslant \min\{r_{i+1}, R\} \\ 0, & r \geqslant R, \end{cases} \tag{A1}$$

where V_i and r_i are generated by recurrence relations $V_i = V_{i-1} + \delta V_i$ and $r_i = r_{i-1} + \delta r_i$, with initial value V_0 uniformly distributed from $[-\tilde{V}_0/2, \tilde{V}_0/2]$ for some positive $\tilde{V}_0$ and $r_0 = 0$. δr_i and δV_i are all random variables; each is identically and independently distributed as $\theta(\delta r_i)N(\delta r_i, 4R/N_0, \sqrt{2}R/N_0)$ and $N(\delta V_i, 0, \tilde{V}_0/10)$. Here $N(x, \mu, \sigma)$ is the normal distribution with expectation μ and standard deviation σ. This procedure, on the one hand, introduces enough randomness, and on the other hand, gives a very smooth potential. Finally we get our purely attractive potential by $V(r) = -|\tilde{V}(r)|$. A schematic of the generation of potential is shown in Fig. 5.

To feed the neural network with generated potentials, we discretize the potential $V(r)$ uniformly from $r = 0$ to $r = R$. As a result the input data are N_0-dimensional vectors of

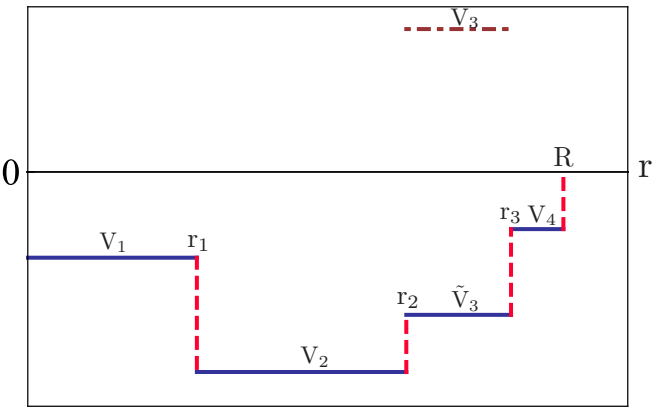

FIG. 5. Random square well potential generated by two kinds of Gaussian distributions.

the form

$$\mathbf{V} = \left[V\left(\frac{R}{N_0}\right), V\left(\frac{2R}{N_0}\right), \ldots, V(R)\right]^T. \tag{A2}$$

We take $N_0 = 64$ throughout this work.

Generating scattering lengths. After generating a potential, we calculate the corresponding scattering length by solving the Schrödinger equation using transfer matrix method [46] for the discretized potential $\mathbf{V}$ [using the original potential $V(r)$ will introduce systematic error].

Within each square well $r \in (r_{i-1}, r_i)$, the wave function is $\phi_i = a_i j_l(k_i r) + b_i n_l(k_i r)$. For zero-energy scattering, we have $k \sim \sqrt{-V}$ ($m = \hbar = 1$). j_l, n_l are two spherical Bessel functions, and l labels the angular momentum and thus the partial wave. To avoid divergence at $r = 0$, $\phi_0 = j_l(k_0 r)$. At the boundary of each square well $r = r_i$, the wave function obeys continuity condition $\phi(r_i^-) = \phi(r_i^+)$, $\phi'(r_i^-) = \phi'(r_i^+)$. By matching the boundary condition, we can determine all ϕ_i for $r < R$.

Outside the short-range potential $r \to R^+$ the wave function is determined by scattering parameters:

$$s \text{ wave} : \psi_0 \propto 1 - \frac{a_s}{r} \equiv \phi_{o0}, \tag{A3}$$

$$p \text{ wave} : \psi_1 \propto \frac{x}{3v} - \frac{1}{x^2} \equiv \phi_{o1}, \tag{A4}$$

$$d \text{ wave} : \psi_2 \propto \frac{x^2}{15D} - \frac{3}{x^3} \equiv \phi_{o2}. \tag{A5}$$

Finally, by matching the boundary condition at $r = R$,

$$\frac{\psi_l'}{\psi_l} = \frac{\phi_n'}{\phi_n} = \frac{\phi_{ol}'}{\phi_{ol}}, \tag{A6}$$

we obtain the scattering length for different partial waves.

Data postselection. Before the first resonance, $a_s < 0$ becomes more negative when the potential becomes deeper. We generate potentials with different $\tilde{V}_0$, which determines the overall depth of generated potential. We then postselect preferred data to make our training set a uniform distribution $a_s \in (\tilde{a}_s, 0)$.

APPENDIX B: TRAINING DETAILS

We use Wolfram *Mathematica* [47] to train and evaluate neural networks. All the weights are randomly initialized to a normal distribution with Xavier method [48]. While optimizing parameters, we minimize the loss function

$$L = \frac{1}{N}\sum_l \left[a_s^{\mathrm{P}}(\mathbf{V}^l) - a_s^{\mathrm{T}}(\mathbf{V}^l)\right]^2 \tag{B1}$$

using Adam algorithm with batch size 32. Here $a_s^{\mathrm{T}}(\mathbf{V})$ is the scattering length obtained using the method introduced in the last section. $a_s^{\mathrm{P}}(\mathbf{V})$ is the prediction made by the network and N is the size of the testing set. Considering there are thousands of parameters in this network, the training set contains $N = 3 \times 10^4$ $\{\mathbf{V}, a_s(\mathbf{V})\}$ pairs. In order to prove weight matrix visualization, we set L_2 regularization strength be 0.1, although there is no noise in the training set. We check that there is no overfitting by using a validation set of size $N_v = 3 \times 10^3$. It typically takes five epochs for the network to converge, meaning the training is very easy. This is actually a hint that the network may fit to a very simple function of $a_s(\mathbf{V})$.

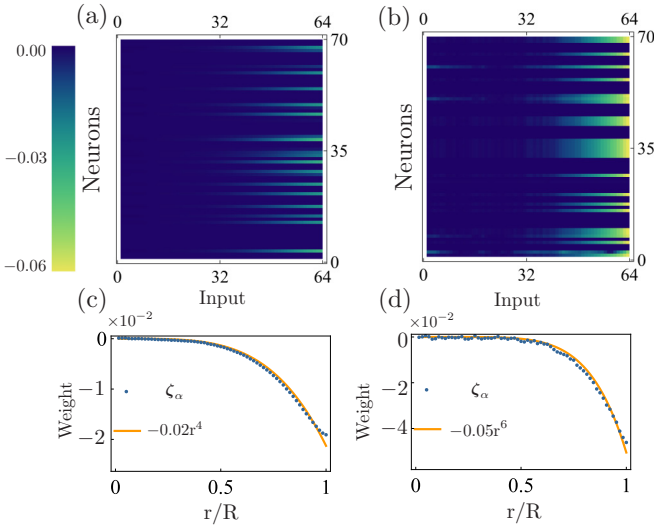

FIG. 6. Visualization of the matrix W [(a) and (b)] and a typical N_0-dimensional vector $\boldsymbol{\zeta}_\alpha$ as a function of $r/R = i/N_0$ [(c) and (d)], for p-wave [(a) and (c)] and for d-wave scattering [(b) and (d)] in the first Born approximation regime. The solid line in (c) is $\boldsymbol{\zeta}_{\alpha=38}$, fitted by r^4 and the solid line in (d) is $\boldsymbol{\zeta}_{\alpha=64}$ fitted by r^6. For p-wave training set, the scattering volumes v distribute uniformly within $v/R^3 \in (-0.02, 0)$. For d-wave training set, the scattering supervolumes D distribute uniformly with $D/R^5 \in (-0.01, 0)$.

APPENDIX C: FIRST-ORDER BORN APPROXIMATION FOR HIGH PARTIAL WAVES

In this Appendix, we report that the neural network could capture first-order Born approximation for p-wave and d-wave wave functions.

In the main text the first Born approximation is shown in Eq. (3) for s partial wave. For p-wave and d-wave wave functions, the approximation becomes

$$v = \int V(r) r^4 dr, \tag{C1}$$

$$D = \int V(r) r^6 dr, \tag{C2}$$

where v and D are scattering volume and supervolume, respectively. We generate training data in the same way, and train neural networks with architecture $N_0 = 64$, $N_1 = 70$, $N_2 = 32$, $N_3 = 1$. Figure 6 shows weight matrix W and the fitting result of a typical neuron in the first hidden layer. As one can see, neurons trained on p-wave data behave as r^4, and neurons trained on d-wave data behave as r^6, as expected from Eqs. (C1) and (C2).

[1] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[2] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Sci. Rep. **7**, 8823 (2017).
[3] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[4] Y. Zhang and E.-A. Kim, Phys. Rev. Lett. **118**, 216401 (2017).
[5] Y. Zhang, R. G. Melko, and E.-A. Kim, Phys. Rev. B **96**, 245119 (2017).
[6] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).
[7] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **86**, 44708 (2017).
[8] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).
[9] P. Ponte and R. G. Melko, Phys. Rev. B **96**, 205146 (2017).
[10] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[11] A. Tanaka and A. Tomiya, J. Phys. Soc. Jpn. **86**, 063001 (2017).
[12] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).
[13] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).
[14] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[15] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).
[16] N. C. Costa, W. Hu, Z. J. Bai, R. T. Scalettar, and R. R. P. Singh, Phys. Rev. B **96**, 195138 (2017).
[17] C. Wang and H. Zhai, Phys. Rev. B **96**, 144432 (2017).
[18] P. Broecker, F. F. Assaad, and S. Trebst, arXiv:1707.00663.
[19] K. Ch'ng, N. Vazquez, and E. Khatami, Phys. Rev. E **97**, 013306 (2018).
[20] P. Zhang, H. Shen, and H. Zhai, Phys. Rev. Lett. **120**, 066401 (2018).
[21] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101 (2017).
[22] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 241104 (2017).
[23] X. Y. Xu, Y. Qi, J. Liu, L. Fu, and Z. Y. Meng, Phys. Rev. B **96**, 041119 (2017).
[24] Y. Nagai, H. Shen, Y. Qi, J. Liu, and L. Fu, Phys. Rev. B **96**, 161102 (2017).
[25] H. Shen, J. Liu, and L. Fu, Phys. Rev. B **97**, 205140 (2018).
[26] L. Huang, Y.-F. Yang, and L. Wang, Phys. Rev. E **95**, 031301 (2017).
[27] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[28] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).
[29] L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.
[30] K. Mills and I. Tamblyn, Phys. Rev. E **97**, 032119 (2018).
[31] K. Mills, M. Spanner, and I. Tamblyn, Phys. Rev. A **96**, 042113 (2017).
[32] Y.-Z. You, Z. Yang, and X.-L. Qi, Phys. Rev. B **97**, 045153 (2018).
[33] T. Mano and T. Ohtsuki, J. Phys. Soc. Jpn. **86**, 113704 (2017).
[34] V. Dunjko and H. J. Briegel, arXiv:1709.02779.
[35] W.-J. Rao, Z. Li, Q. Zhu, M. Luo, and X. Wan, Phys. Rev. B **97**, 094207 (2018).
[36] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Phys. Rev. B **96**, 205152 (2017).
[37] N. Yoshioka, Y. Akagi, and H. Katsura, Phys. Rev. B **97**, 205110 (2018).
[38] Y. LeCun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).
[39] A. Nguyen, J. Yosinski, and J. Clune, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015* (IEEE, Boston, MA, 2015).
[40] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, Proceedings of the International Conference on Learning Representations (ICLR), 2017 (unpublished).
[41] E. Brynjolfsson and T. Mitchell, Science **358**, 1530 (2017).
[42] E. Braaten and H.-W. Hammer, Phys. Rep. **428**, 259 (2006).
[43] L. D. Landau and E. M. Lifshitz, *Quantum Mehanics: Non-Relativistic Theory* (Pergamon, London, 1981).
[44] G. Cybenko, Math. Control Signals Syst. **2**, 303 (1989).
[45] K. Hornik, Neural Netw. **4**, 251 (1991).
[46] B. Jonsson and S. T. Eng, IEEE J. Quantum Electron. **26**, 2025 (1990).
[47] *Mathematica, Version 11.1* (Wolfram Research, Champaign, IL, 2017).
[48] X. Glorot and Y. Bengio, in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, edited by Yee Whye Teh and Mike Titterington (Proceedings of Machine Learning Research, Sardinia, Italy, 2010), Vol. 9, pp. 249–256.