

Semileptonic decays of heavy mesons with artificial neural networks

Cody M. Grant,¹ Ayesha Gunawardana¹,² and Alexey A. Petrov^{1,2}

¹*Department of Physics and Astronomy Wayne State University, Detroit, Michigan 48201, USA*

²*Leinweber Center for Theoretical Physics University of Michigan, Ann Arbor, Michigan 48196, USA*



(Received 21 April 2020; accepted 21 July 2020; published 5 August 2020)

Experimental checks of the second row unitarity of the Cabibbo-Kobayashi-Maskawa (CKM) matrix involve extractions of the matrix element V_{cd} , which may be obtained from semileptonic decay rates of D to π . These decay rates are proportional to hadronic form factors which parametrize how the quark $c \rightarrow d$ transition is realized in $D \rightarrow \pi$ meson decays. The form factors cannot yet be analytically computed over the whole range of available momentum transfer q^2 , but can be parametrized with a varying degree of model dependency. We propose analysis of the form factor shapes using a system of artificial neural networks trained from experimental pseudodata and averaged together to predict their shapes with a prescribed uncertainty. We comment on the parameters of several commonly-used model parametrizations of semileptonic form factors. We extract shape parameters and use unitarity to bound the form factor at a given q^2 , which then allows us to bound the CKM matrix element $|V_{cd}|$.

DOI: 10.1103/PhysRevD.102.034003

I. INTRODUCTION

Studies of exclusive semileptonic decays of heavy mesons play an important role in understanding the dynamics of the strong interaction. They may also provide additional constraints on physics beyond the standard model (SM) [1]. Such searches, recently performed in B decays yielded tantalizing results in measurements related by lepton universality requirements, i.e., by the requirement that couplings of leptons to gauge bosons be independent of the lepton flavor. It is interesting to see if similar anomalies exist in semileptonic decays of charmed particles if higher precision data are available [2–5].

Accurate theoretical description of such transitions is also needed for the extraction of relevant Cabibbo-Kobayashi-Maskawa (CKM) matrix elements. In particular, decays of charmed D^0 , D^+ , or D_s mesons provide the simplest way to determine the magnitudes of quark mixing parameters V_{cs} or V_{cd} [5,6]. Extractions of these CKM matrix elements from experimentally measured semileptonic decay rates are done with the knowledge of matrix elements of quark currents that describe strong interaction effects. This implies that accurate description of semileptonic transitions is also needed for improvement of our understanding of quark hadronization mechanisms in quantum chromodynamics (QCD). A hadronic transition

between two mesons in exclusive semileptonic decays makes it a suitable system to theoretically analyze matrix elements of flavor changing currents, which are usually parametrized in terms of momentum-dependent form factors. In semileptonic decays of charmed mesons, the form factors that describe the hadronic part of the decay amplitudes are conventionally introduced as

$$\begin{aligned} &\langle K(\pi)(p_{K(\pi)})|\bar{q}\gamma_\mu c|D(p_D)\rangle \\ &= F_+(q^2)\left(P_\mu - \frac{m_D^2 - m_{K(\pi)}^2}{q^2}q_\mu\right) + F_0(q^2)\frac{m_D^2 - m_{K(\pi)}^2}{q^2}q_\mu, \end{aligned} \quad (1)$$

where $P = p_D + p_{K(\pi)}$ and $q = p_D - p_{K(\pi)}$. Experimental studies of these form factors are performed through the analysis of the differential decay rate $d\Gamma/dq^2$. In the simplest cases where the mass of the final state lepton can be neglected, the differential decay rate can be written as

$$\frac{d\Gamma(D \rightarrow K(\pi)\ell\nu_\ell)}{dq^2} = \frac{G_F^2|V_{cq}|^2}{24\pi^3}|\mathbf{p}_{K(\pi)}|^3|F_+(q^2)|^2, \quad (2)$$

where $|\mathbf{p}_{K(\pi)}|$ is the magnitude of the $K(\pi)$ 3-momentum vector in the D -meson rest frame. As can be seen from Eq. (2), only a single form factor, $F_+(q^2)$, contributes.

Accurate calculations of the nonperturbative form factors $F_{+j0}(q^2)$ in the whole momentum range are very challenging. Aside from lattice QCD [7] and/or QCD sum rule (QCDSR) [8] calculations of matrix elements of hadronic

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

currents in exclusive decays, we are currently lacking a complete nonperturbative description of hadronic form factors. While both lattice QCD and QCDSR computations of form factors are improving, at the moment they only provide model-independent predictions for $F_+(q^2)$ at limited regions of q^2 .

Rather general arguments based on analyticity of $F_+(q^2)$ have been used to place general constraints on the shapes of the form factors. A popular approach that rigorously employs analyticity requirement involves the so-called z -expansion, where a series expansion of the form factor around some point $t = q^2$ is improved by making a conformal transformation to the parameter z [9],

$$z(q^2) = \frac{\sqrt{t_+ - t_0} - \sqrt{t_+ - q^2}}{\sqrt{t_+ - t_0} + \sqrt{t_+ - q^2}}, \quad (3)$$

which maps the interval $-\infty < q^2 < t_+$ onto the line segment $-1 < z < 1$. Here t_0 is a free parameter that corresponds to the values of q^2 that maps onto $z = 0$, and $t_{\pm} = (m_D \pm m_{\pi})^2$. The form factor can be expanded in z as

$$F_+(q^2) = \frac{1}{\Phi(q^2, t_0)} \sum_{k=0}^{\infty} a_k(t_0) z^k(q^2, t_0) \quad (4)$$

where $\Phi(q^2, t_0)$ is an arbitrary function that is analytic anywhere but the unitarity cut [9,10]. Note that $\Phi(q^2, t_0)$ is often written as $\Phi(q^2, t_0) = P(q^2)\phi(q^2, t_0)$, with the Blaschke factor $P(q^2) = z(q^2, m_V^2)$ if there are poles present in between $q^2 = 0$ and the beginning of the unitarity cut, as in $B \rightarrow \pi$ transitions where $m_V = m_{B^*}$ [11,12]. Note that $P(q^2) = 1$ for the $D \rightarrow \pi \ell \nu$ transition. The expansion in Eq. (4) is converging rapidly, so only a few terms in the expansion are really needed.¹ Lattice QCD [7] or QCD sum rule [8] results can be used to constrain the coefficients a_k to provide a model-independent parametrization of the form factor.

As it stands, phenomenological parametrizations of the form factors are also often used [14]. The most common parametrization is a ‘‘single pole’’ shape, where the pole refers to the lowest mass vector resonance formed in the t -channel with quantum numbers of the quark current. For example, in the decay $D \rightarrow \pi e \bar{\nu}_e$ the dominant pole is the D^* , a vector state with 1^- quantum numbers,

$$F_+^{\text{pole}}(q^2) = \frac{F_+(0)}{1 - \hat{q}^2}, \quad (5)$$

where $F_+(0)$ is the value of the form factor at zero momentum recoil that has to be fixed either from the

¹See however [13] for a discussion of possible shortcomings of this approach.

lattice QCD or from other arguments, and $\hat{q}^2 = q^2/m_{D^*}^2$. While physical masses of the states $D^*(2010)$ (for $D \rightarrow \pi$ transition) or $D_s^*(2112)$ (for $D \rightarrow K$ transition) could be used, the mass m_{D^*} is often taken as a fit parameter, as there is no reason to believe that the lowest-lying pole would saturate the form factor over the whole available kinematical range. More complicated shapes, with more effective poles, are also available [1],

$$F_+(q^2) = \frac{F_+(0)}{(1 - \alpha)} \frac{1}{1 - q^2/m_V^2} + \sum_{k=1}^N \frac{\rho_k}{1 - \frac{1}{\gamma_k} \frac{q^2}{m_V^2}}, \quad (6)$$

where α determines the strength of the dominant pole, ρ_k gives the strength of the k th term in the expansion, and $\gamma_k = m_{V_k}^2/m_V^2$, with m_{V_k} representing masses of the higher mass states with vector quantum numbers. In principle, a form factor can be approximated to any desired accuracy by introducing a large number of effective poles. Keeping the number of terms in this expansion manageable, a popular parametrization due to Becirevic and Kaidalov (BK) [15] is often used, representing the $N = 1$ truncation of the expansion in Eq. (6),

$$F_+^{\text{BK}}(q^2) = \frac{F_+(0)}{(1 - \hat{q}^2)(1 - a_{\text{BK}}\hat{q}^2)}, \quad (7)$$

where a_{BK} is a fit parameter. As with the case of a single pole shape in Eq. (5), a good fit to experimental distribution can be obtained if m_V is regarded as a fit parameter as well. While there is only one extra parameter between Eq. (5) and (7), the BK parametrization starts off with more parameters. A number of parameters is reduced by applying relationships obtained in the heavy-quark limit to decrease the number of parameters displayed in the above equation. A further extension of the BK parametrization was proposed by Ball and Zwicky (BZ) [16,17],

$$F_+^{\text{BZ}}(q^2) = \frac{F_+(0)}{1 - \hat{q}^2} \left(1 + \frac{r_{\text{BZ}}\hat{q}^2}{1 - a_{\text{BZ}}\hat{q}^2} \right), \quad (8)$$

where r_{BZ} and a_{BZ} are the shape parameters. Note that the parametrization of the form factor in BZ model employed in Eq. (8) [17] can be related to the original BZ-parametrization [16]

$$F_+^{\text{BZ}}(q^2) = \frac{r_1}{1 - \hat{q}^2} + \frac{r_2}{1 - a_{\text{BZ}}\hat{q}^2}, \quad (9)$$

by the identification $F_+(0) = r_1 + r_2$ and $r_{\text{BZ}} = (a_{\text{BZ}} - 1)r_2/(r_1 + r_2)$. Note that a_{BZ} represents parametrization of the continuum states above D^* and therefore $a_{\text{BZ}} < 1$.

All form factor parametrization discussed above represent physically-motivated ways to describe hadronic

input. Yet, a question might be asked then what uncertainty should be assigned to the *choice* of a particular shape of the fit function. In other words, we will be interested if choosing a specific functional form for the form factor induces a bias in the interpretation of results of an experimental analysis.

This question may be addressed in the framework of machine learning (ML) approach, in particular, it can be investigated with the help of artificial neural networks (ANN). Based on the Kolmogorov-Arnold representation theorem [18], it has been shown that ANN can be used as an unbiased estimator of data [19,20]. This fact has been used by the NNPDF collaboration to parametrize nucleon's parton distribution functions [21–23], and in form factor analysis of nucleon data [24,25]. In this paper we shall build a statistical interpolating model based on a system of ANNs that contains information on experimental uncertainties and correlations, but does not introduce theoretical bias. Following [21,22], we employ an approach based on multilayer feed-forward neural networks trained using the back-propagation learning algorithm.

II. NEURAL NETWORKS

A. Basic facts about neural networks

With the recent explosion of interest in machine learning, artificial neural networks are now widely employed in analyses in experimental particle physics. Their use in jet-finding algorithms and other applications are well known [26]. Roughly speaking, a neural network is represented by a certain nonlinear function that connects input and output data. This leads to another feature of ANNs which we

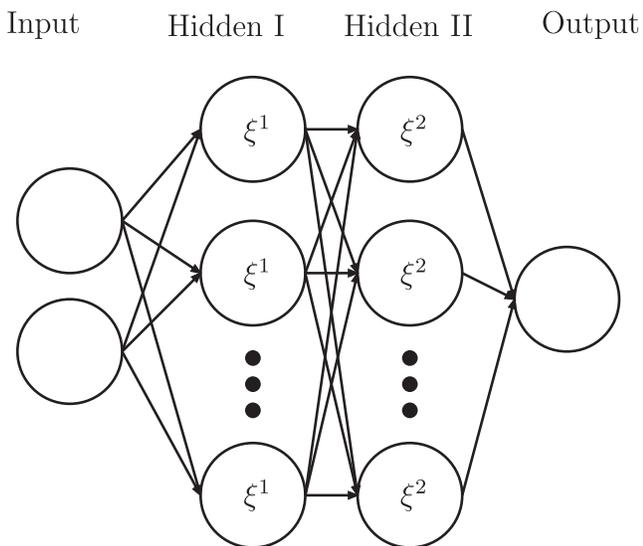


FIG. 1. A sample structure of an artificial neural network with two hidden layers. In this work we used the ANN structure (2,3,4,1), i.e., a network with two nodes in the input layer, three (four) nodes in the first (second) hidden layer, and one node in the output layer. The function ξ^ℓ is defined in Eq. (13).

explore in this paper: their ability to provide unbiased universal approximants to incomplete data [19,20].

An ANN is built to mimic the structure of human neurons and consists of a set of interconnected units (see Fig. 1) called *neurons* or *nodes*. The activation state of a neuron is determined as a function of the activation states of the i neurons connected to it. Each pair of neurons is connected by a synapsis, characterized by a weight, which we call ω_i . We also introduce a set of θ_i , representing thresholds for each neuron to “fire.” Each ANN contains several groups of neurons called *layers*. The first layer is called an input layer. It provides input information that is to be approximated. In this paper the input information is the value of q^2 for each bin in q^2 distribution of the CKM matrix element times the semileptonic form factor. We find it convenient in this work to work with an input layer that contains two nodes, as we shall explain later. The final layer is the output layer. It gives the value of form factor for each q^2 along with its uncertainty. Layers between the input and output are conventionally called *hidden*. In this work we employ ANNs with two hidden layers containing 3 nodes in first hidden layer and 4 in the second one. The ANN is *trained* when optimal sets of weights and thresholds are determined such that ANN reproduces the training data within a given uncertainty. This is achieved by minimizing the error function,

$$E[\omega, \theta] \equiv \frac{1}{2} \sum_{A=1}^{n_p} (o(q_A^2) - y_A)^2, \quad (10)$$

where n_p is the number of pseudodata used to train an ANN, $o(q_A^2)$ is the output, which is given by the ANN's fit for a given input data q_A^2 . The target data point for our paper, y_A , is obtained from the magnitude of the CKM matrix element times the semileptonic form factor, $|V_{cd}F_+(q^2)|$. The differential distribution of Eq. (2) is proportional to its square.

The $o(q_A^2)$ is obtained using forward propagation. In order to achieve this we pass the input through a network of hidden nodes. The output from the first hidden layer with n_1 number of nodes is

$$\xi^{[1]} = g\left(\sum_{i=1}^{n_1} \omega_i^{[1]} q^2 - \theta^{[1]}\right). \quad (11)$$

In this equation the response of each neuron is given by

$$g(x) \equiv \frac{1}{1 + e^{-x}}, \quad (12)$$

which is the *sigmoid* activation function, and the summation over the q^2 data points is implied.

The $\xi^{[1]}$ is then used as an input for the second hidden layer with n_2 number of hidden nodes, and so on. The

process is continued until the output layer of ANN is reached. In general, we can construct the output from ℓ th hidden layer with n_ℓ number of nodes as

$$\xi^{[\ell]} = g\left(\sum_{i=1}^{n_\ell} \omega_i^{[\ell]} \xi_i^{[\ell-1]} - \theta^{[\ell]}\right). \quad (13)$$

where $\xi^{[\ell-1]}$ is the output from the $(\ell - 1)$ th layer. The fit of the L layer ANN $o(q^2)$ is then defined as

$$o(q^2) = \xi^{[L]}. \quad (14)$$

In the training process the thresholds and weights need to be adjusted so the output represented the training data with a set precision, so the error function in Eq. (10) need to be minimized. It is common to use the method of steepest descent for this purpose. Instead, we decided to use the nonlinear conjugate gradient (NLCG) method [27,28] to minimize Eq. (10). In each iteration the ω_i and the θ_i update as

$$\delta\omega^{[\ell]} = -\eta \frac{\partial E}{\partial \omega^{[\ell]}}, \quad \delta\theta^{[\ell]} = -\eta \frac{\partial E}{\partial \theta_i^{[\ell]}}, \quad (15)$$

where η is the learning rate at a given iteration. The NLCG method employed here does not require a pre-defined learning rate. The learning rate is initially determined by using line search algorithms [27], and then iteratively updated based on the gradients that are in a conjugate direction to original gradient used in the line search algorithm. As it turns out, the NLCG method converges much faster than steepest descent method for the fits employed in this paper. For more details on the NLCG method, see Ref. [28]. The gradients of the error function are obtained by using the method of back propagation [29]. Back propagation can be thought of as a consecutive application of the chain rule. By applying the chain rule we find the gradient of the L th layer to be

$$\Delta^{[L]} = g'(h^{[L]})[o(q^2) - y] \quad (16)$$

where $g'(h^{[L]})$ is the derivative of the activation function with respect to $h^{[L]}$ and

$$h^{[L]} = \sum_{i=1}^{n_{L-1}} \omega_i^{[L]} \xi_i^{[L-1]} - \theta^{[L]}. \quad (17)$$

The derivatives with respect to ω_i and θ_i for layer L are given by

$$\begin{aligned} \frac{\partial E}{\partial \omega_i^{[L]}} &= \Delta^{[L]} \xi_i^{[L-1]}; & i = 1, \dots, n_{L-1}, \\ \frac{\partial E}{\partial \theta_i^{[L]}} &= -\Delta^{[L]}. \end{aligned} \quad (18)$$

The output of Eq. (16) is used to obtain the derivatives of the $(L - 1)$ th layer, $\Delta_j^{[L-1]}$,

$$\Delta_j^{[L-1]} = g'(h^{[L-1]}) \Delta_i^{[L]} \omega_i^{[L]}. \quad (19)$$

The procedure is repeated for the hidden layers to find derivatives of error function with respect to ω_i and θ_i in each layer,

$$\begin{aligned} \frac{\partial E}{\partial \omega_i^{[\ell]}} &= \Delta_i^{[\ell]} \xi_j^{[\ell-1]}; & i = 1, \dots, n_\ell, & \quad j = 1, \dots, n_{\ell-1}, \\ \frac{\partial E}{\partial \theta_i^{[\ell]}} &= -\Delta_i^{[\ell]}; & i = 1, \dots, n_\ell. \end{aligned} \quad (20)$$

Using these we can obtain the numerical gradient of the error function and find the corrections to the weights and thresholds.

B. Neural network training

Training of ANNs described in the previous section must be performed either on real or artificial data (pseudodata). The pseudodata is generated using as much experimental information as possible. It can be constructed with uncorrelated data, correlated data, normalized data, or some combination of all three. In this work we elected to follow [23] and generate pseudodata from the BES III experimental data set of [30] employing Monte Carlo techniques. We chose to select only this experimental data set and not to include earlier experimental data because the BES III data set includes complete correlation matrices for statistical and systematic uncertainties. It would indeed be interesting to apply our methodology to upcoming Belle II data. The artificial data is generated as

$$\begin{aligned} |V_{cd}F_+(q^2)|_i^{(\text{art}), (k)} &= |V_{cd}F_+(q^2)|_i^{(\text{exp})} \\ &+ \sum_{j=1}^{N_{\text{sys}}} r_{\text{sys}, j}^{(k)} \sigma_{\text{sys}, ji} + \sum_{m=1}^{N_{\text{stat}}} r_{\text{stat}, m}^{(k)} \sigma_{\text{stat}, mi} \end{aligned} \quad (21)$$

where $i = 1, \dots, N_{\text{data}}$ is the number of experimental data entries considered, which is equal to the number of q^2 bins. These entries are used to generate $k = 1, \dots, N_{\text{rep}}$ of Monte Carlo “replicas.” These replicas are generated following the recipe of [23]. The first term on the right-hand side of Eq. (21) is the central value from the experimental data point for a given q^2 bin. The data points in the replicas are created from it by using the remaining terms on the right-hand side of Eq. (21), which provide variation in pseudodata samples. They represent experimental uncertainties (correlated systematic, and correlated statistical, respectively) obtained from the experimental

data. Each ‘‘uncertainty term’’ is multiplied by a Gaussian random number $r_{\text{sys},j}^{(k)}$ or $r_{\text{stat},m}^{(k)}$, which have a mean of zero and a standard deviation of one. The correlation matrix elements, $\text{corr}(j, i)$, found in Ref. [30] is a normalized correlation matrix. We use the relative uncertainty found under the correlation matrix in the same table to compute $\sigma_{\text{sys},ji}$ and $\sigma_{\text{stat},li}$:

$$\sigma_{j,i} = \sqrt{\tilde{\sigma}_i \tilde{\sigma}_j \text{corr}(j, i)}, \quad (22)$$

where $\tilde{\sigma}_i$ is the relative uncertainty for the i th bin of data. Every value of $|V_{cd}F_+(q^2)|^{\text{(art)}}$ has a different and unique q^2 input generated for it.

There are three possible ways to generate pseudodata for each given value of q^2 inside each bin: a random choice of q^2 generated with a flat prior across the entire q^2 bin, the central q^2 value for each bin, or a random choice of q^2 assuming some non-flat distribution of q^2 values inside the bin. We chose the first option, as introducing a distribution of q^2 values, be it a delta function at the middle of the bin or some other continuous random distribution, would introduce a bias in the ANN curve related to the choice of the q^2 distribution. Therefore, we feel that generating a flat prior for q^2 inside each bin is the best option with the least number of assumptions.

We generate a new set of pseudodata when we initialize a new neural network. Each batch has an average q^2 and a standard deviation relating to the q^2 values, which are used to scale the each value of q^2 which we have generated. Using the scaled q^2 data as a secondary input is recommended to improve the stability and the performance of ANNs [31]. In particular, data standardization is a popular data scaling choice, and it is defined as $\tilde{q}_{ip}^2 = (q_{ip}^2 - \bar{q}_p^2)/\sigma_p$, where p is the batch number and i is a single q^2 value in the batch. With this transformed data, each of our ANNs has the structure (2,3,4,1), as the two hidden layers provide the most efficient structure without compromising the performance or accuracy. We terminate ANN training when the difference in the error functions from the current and last epochs reaches a value of 10^{-8} or less. We found that using a larger bound results in a poor fit, while using a smaller bound exponentially increases the computation time. With a higher number of nodes, the fit for the system of ANN’s would have a smaller uncertainty, but the training speed would also be reduced. Finally, to address overfitting concerns, we studied various configurations of ANNs. Please see Appendix for more information regarding various choices of hyperparameters.

III. FORM FACTOR PARAMETRIZATION WITH NEURAL NETWORKS

We generated 18,000 pseudodata points for q^2 bins for each of the ten networks. After training all networks

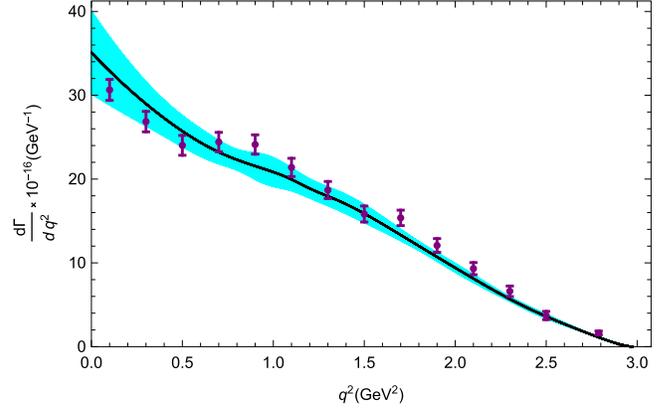


FIG. 2. Our averaged ANN result for the differential decay rate plotted against the experimental measurement. The purple data points are the experimental data from [30]. The black and cyan curves are the average value and one standard deviation, respectively, from the output of our averaged ANN.

individually, we found the average ANN-predicted curve for $|V_{cd}F_+(q^2)|$ with uncertainty calculated for every q^2 value. The differential decay rate, $d\Gamma/dq^2$, can then be easily computed. The q^2 dependences of the rate and the $|V_{cd}F_+(q^2)|$ curves are shown in Figs. 2 and 3 respectively. Further results of the ANN training and relevant graphs are available at the URL <https://s.wayne.edu/hepmachinelearning/>.

We would like to point out that ANNs are worse at fitting the values near the edges of their training ranges when compared to fitting values near the middle of that range, which is common for ANN fits [32]. Yet, considering the ANN-predicted $|V_{cd}F_+(q^2)|$ curve in Fig. 3, it appears that there is only a slight ‘‘flaring’’ of the uncertainty at the

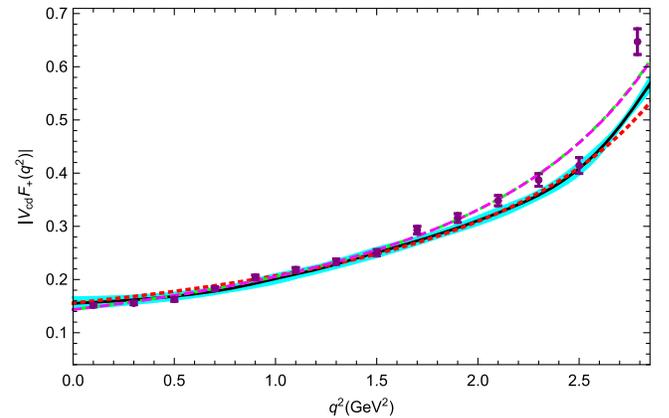


FIG. 3. ANN fits for $|V_{cd}F_+(q^2)|$ plotted against the three models described in the text. The black and cyan curves are the average value and one standard deviation, respectively, from the output of our neural network. The dotted red curve is the simple pole model. The dot-dashed green curve is the modified pole model. The dashed magenta curve is the BZ model. The purple data points are calculated from the experimental data in Ref. [30].

TABLE I. Form factor parameters at $q^2 = 0$ for this work and three common model choices. Note the unreasonably tiny uncertainty of the parameters $F_{1,2}$ of the pole form factor, which is related to the rigidity of the chosen functional form.

Form factor	$ V_{cd}F_+(0) \times 10^{-2}$	$F_1 \times 10^{-1} \text{ GeV}^{-1}$	$F_2 \times 10^{-1} \text{ GeV}^{-1}$
ANN (this work)	15.52 ± 1.07	1.049 ± 0.148	1.322 ± 0.343
$F_+^{\text{pole}}(q^2)$	15.57 ± 0.10	2.4830 ± 0.0001	1.2330 ± 0.0001
$F_+^{\text{BK}}(q^2)$	14.37 ± 0.16	3.170 ± 0.072	1.666 ± 0.055
$F_+^{\text{BZ}}(q^2)$	14.35 ± 0.25	3.215 ± 1.378	1.666 ± 0.887

kinematical boundaries. The uncertainties of $d\Gamma/dq^2$ curve in Fig. 2 are then roughly doubled, as it is a square of $|V_{cd}F_+(q^2)|$ times some kinematical factors. The fractional uncertainty at the $q^2 = 0$ boundary is visually more pronounced due to larger experimental uncertainties of data toward $q^2 = 0$. We would like to compare our results with some common form factor models: simple pole, the BK model (or modified pole), and the BZ model [15–17]. Since the ANN fits a product of V_{cd} and $F_+(q^2)$, a direct comparison will be affected by the value of V_{cd} that would have to be taken as an external parameter. With that in mind, we can compare $|V_{cd}F_+(0)|$ obtained from the model fits and our ANN analysis of the semileptonic decay data.

A further insight into how well model-inspired parametrizations describe hadronic dynamics is possible if we expand the form factor around $q^2 = 0$,

$$|V_{cd}F_+(q^2)| = |V_{cd}F_+(0)|(1 + F_1q^2 + F_2q^4 + \dots), \quad (23)$$

and compare the coefficients F_n of the higher order terms for the models to our averaged ANN output. We looked at the ratios of the n th derivative of the form factor divided by the form factor at $q^2 = 0$,

$$F_n = \frac{1}{n!F_+(0)} \left. \frac{d^n F_+(q^2)}{d(q^2)^n} \right|_{q^2=0}. \quad (24)$$

We note that the first and second terms in Eq. (23), which are independent of the value of V_{cd} , are quite sensitive to the quark hadronization dynamics. In particular, drawing parallels to the discussion of the charge radius of the proton [33], the slope of $F_+(q^2)$ at $q^2 = 0$, denoted F_1 , encodes the information about the effective size of the volume where the quark transition takes place. We shall call the coefficient F_1 a transitional charge radius.

In order to compare $F_{1,2}$ of a particular model-inspired parametrizations to our ANN fits, we need to determine shape parameters for each form factor model. Other than the simple pole model, where we take the mass of the $D^*(2010)$ resonance as m_{D^*} , the parameters that need to be

fit include a_{BK} for the BK model, and a_{BZ} and r_{BZ} for the BZ model. We obtain these shape parameters by fitting the model to the experimental data. Using this procedure we find $a_{\text{BK}} = 0.277 \pm 0.029$ for the BK model and $r_{\text{BZ}} = 0.295 \pm 0.123$ and $a_{\text{BZ}} = 0.192 \pm 0.555$ for the BZ model. We note that for each of these parametrizations the combination $|V_{cd}F_+(0)|$ is also treated as a fit parameter.

The resulting values for $|V_{cd}F_+(0)|$, F_1 , and F_2 for the neural network parametrization and the model-inspired parametrizations can be found in Table I. Please note that ANN output is a discrete set of values for various values of q^2 . Regardless of how fine the values of q^2 are spaced, numerical calculations of derivatives for F_1 and F_2 [see Eq. (24)] need to be done very carefully. Since we only use data close to $q^2 = 0$, i.e., the data in the range $0 \leq q^2 \leq 0.54 \text{ GeV}^2$, we found it easier to fit the averaged ANN curve directly to a Taylor expansion around $q^2 = 0$.

As we can see from the first column of Table I, the values of $|V_{cd}F_+(0)|$ and F_2 are consistent throughout the popular form factor models and are roughly consistent with the results of our ANN study. The agreement is much worse for the parameter F_1 : the ANN fits are consistently smaller for the transitional charge radius F_1 . It is likely that this happens due to rather rigid parametrizations of the model-inspired form factors, which artificially decrease possible uncertainties associated with them. This is particularly true for the simple pole parametrization $F_+^{\text{pole}}(q^2)$: the uncertainty of $F_{1,2}$ is unreasonably small because once $|V_{cd}F_+(0)|$ is fixed, the only uncertainties that can cause the spread in $F_{1,2}$ are the experimental uncertainties in the value of m_{D^*} , which are rather small. We conclude that it is possible that more effective poles need to be taken into account if model-inspired form factors are used for parametrizations of future experimental data. We compare ANN and model fits in Fig. 3.

IV. FORM FACTOR BOUNDS AND THEIR DERIVATIVES

We can use our ANN fits to obtain separate bounds on the CKM matrix element V_{cd} if we combine our fits with

model-independent bounds on the hadronic form factor imposed by analyticity and unitarity requirements [11]. In order to do so and place an upper bound on $|F_+(0)|$, we would need to calculate moments of the heavy-light invariant amplitude $\Pi_+(q^2)$, which we denote by $\chi_+^{(n)}$. They are defined by the relation,

$$\chi_+^{(n)} = \frac{1}{\pi} \int_{t_+}^{\infty} dt \frac{\text{Im}\Pi_+(t+i\epsilon)}{t^{n+1}}, \quad (25)$$

where n denotes a specific moment and $t_+ = (m_D + m_\pi)^2$. These moments can be computed in QCD. In addition, an inequality for the imaginary part of $\Pi_+(q^2)$, which holds for $t > t_+$,

$$\text{Im}\Pi_+(t+i\epsilon) \geq \frac{3}{2} \frac{1}{48\pi} \frac{[(t-t_+)(t-t_-)]^{3/2}}{t^3} |F_+(t)|^2 \quad (26)$$

can be found via the unitarity sum of the $D\pi$ state spectral function in the isospin limit [11]. This result leads to an inequality with respect to the moment,

$$\chi_+^{(n)} \geq \frac{1}{\pi} \int_{t_+}^{\infty} dt \rho_+^{(n)}(t) |F_+(t)|^2. \quad (27)$$

The functions $\rho_+^{(n)}(t)$ are defined as a product of the denominator from Eq. (25) and the phase space factors from Eq. (26),

$$\rho_+^{(n)}(t) = \frac{3}{2} \frac{1}{48\pi} \frac{[(t-t_+)(t-t_-)]^{3/2}}{t^{n+4}}. \quad (28)$$

The form factor is an analytic function in the cut complex t -plane, so we can apply the standard techniques to derive the constraint on the form factor [11]. We can bring Eq. (27) to a canonical form by mapping it to the interior of a unit disk using the transformation in Eq. (3). In this mapping $z(t_+) = 1$ and $z(\infty) = -1$. In terms of z , the inequality is

$$\frac{1}{2\pi} \int_0^{2\pi} d\phi |g_+^{(n)}(e^{i\phi})|^2 \leq \chi_+^{(n)}. \quad (29)$$

On one hand the analytic function $g_+^{(n)}(z)$ can be written as

$$g_+^{(n)}(z) = F_+(\tilde{t}(z)) \omega_+^{(n)}(z), \quad (30)$$

where $\omega_+^{(n)}(z)$ is called the outer function. The outer function is analytic and has no zeroes in the support domain, such that $|\omega_+^{(n)}|^2$ is equal to $\rho_+^{(n)}(\tilde{t}(e^{i\phi}))$ times the Jacobian of Eq. (3),

$$\omega_+^{(n)}(z) = \left(\frac{1}{4\sqrt{2\pi}} \right) \frac{(t_+ - \tilde{t}(z))(\sqrt{t_+ - t_-} + \sqrt{t_+ - \tilde{t}(z)})^{\frac{3}{2}}(\sqrt{t_+ - t_0} + \sqrt{t_+ - \tilde{t}(z)})}{\sqrt[4]{t_+ - t_0}(\sqrt{t_+} + \sqrt{t_+ - \tilde{t}(z)})^{(n+4)}}, \quad (31)$$

where we have set $t_0 = 0$ for our calculation. The outer function and the form factor can be expressed in terms of the variable z using $\tilde{t}(z) = t_+(1 - \frac{(1-z)^2}{(1+z)^2})$, which is the inverse transform of Eq. (3). On the other hand, $g_+^{(n)}(z)$ can be expanded in z ,

$$g_+^{(n)}(z) = g_{+,0}^{(n)} + g_{+,1}^{(n)}z + g_{+,2}^{(n)}z^2 + \dots \quad (32)$$

This expansion is convergent for $|z| < 1$. It follows from Eq. (29) that the coefficients must satisfy the inequality

$$\chi_+^{(n)} \geq \sum_{j=0}^{\infty} (g_{+,j}^{(n)})^2. \quad (33)$$

The left side of the above inequality is always positive, which leads to a maximum number of g -coefficients being nonzero. Expanding Eq. (30) in a Taylor series

around $z = 0$ and setting it equal to Eq. (32) results in each g -coefficients being a function of $F_+(0)$, $F_{1,2}$, and meson masses. Substituting $g_{+,j}^{(n)}(F_+(0), F_{1,2}, \dots)$ into Eq. (33) and solving for $F_+(0)$ leads to a bound on the form factor (at $q^2 = 0$) in terms of F_1 , F_2 , and $\chi_+^{(n)}$,

$$|F_+(0)| \leq \frac{16t_+ \sqrt{2\pi\chi_+^{(n)}(t_+ + \sqrt{t_+\Delta t_{\pm}})}}{\sqrt{h^{(n)}(F_1, F_2, t_+, t_-)}}, \quad (34)$$

where we defined $\Delta t_{\pm} = (t_+ - t_-)$, and $h^{(n)}(F_1, F_2, t_+, t_-)$ is the denominator for this bound, which depends on the moment that is being used to calculate the bound. Changing the moment also means we have to change which outer function we have to use via Eq. (31). Explicitly, the denominator of the $|F(0)|$ bound from the first moment and outer function can be written as

$$\begin{aligned}
h^{(1)}(F_1, F_2, t_{\pm}) = & \frac{5}{16}t_-^2 - \frac{1}{2}t_- \sqrt{t_+ \Delta t_{\pm}} + \frac{25}{16} \sqrt{t_+^3 \Delta t_{\pm}} - t_- t_+ + \frac{109}{64}t_+^2 \\
& + F_1 t_+ \left(2t_-^2 - \frac{13}{2}t_- \sqrt{t_+ \Delta t_{\pm}} - \frac{17}{4} \sqrt{t_+^3 \Delta t_{\pm}} - \frac{11}{2}t_- t_+ - \frac{17}{4}t_+^2 \right) \\
& + F_1^2 t_+^2 (5t_-^2 - 32t_- \sqrt{t_+ \Delta t_{\pm}} + 106 \sqrt{t_+^3 \Delta t_{\pm}} - 85t_- t_+ + 106t_+^2) \\
& + F_1 F_2 t_+^3 (-32t_-^2 + 88t_- \sqrt{t_+ \Delta t_{\pm}} - 224 \sqrt{t_+^3 \Delta t_{\pm}} + 200t_- t_+ - 224t_+^2) \\
& + F_2 t_+^2 (-4t_-^2 + 10t_- \sqrt{t_+ \Delta t_{\pm}} - 2 \sqrt{t_+^3 \Delta t_{\pm}} + 11t_- t_+ - 2t_+^2) \\
& + F_2^2 t_+^4 (32t_-^2 - 64t_- \sqrt{t_+ \Delta t_{\pm}} + 128 \sqrt{t_+^3 \Delta t_{\pm}} - 128t_- t_+ + 128t_+^2). \tag{35}
\end{aligned}$$

The moments $\chi_+^{(n)}$ in Eq. (34) have been calculated in OPE as the sum of the perturbative and nonperturbative contributions. We calculated updated numbers for the moments based on new values for the condensates and masses at a scale of $\mu = 2$ GeV. The parameter values at a scale of $\mu = 2$ GeV were calculated in [34–37] and can be found in Table II. The perturbative pieces for the heavy to light correlators were calculated for up to two loops in Ref. [38]. Using Eqs. (34), (35), and the Appendix of Ref. [38], we can find the updated values for $\chi_+^{(n)PT}$, for which we include

TABLE II. Perturbative and nonperturbative parameters used in the calculation of moments χ_+ .

Quantity	Value
α_S (2 GeV)	0.38 ± 0.03
$m_{c,\text{pole}}$	1.67 ± 0.07 GeV
$\langle \bar{u}u \rangle$ (2 GeV)	$(-0.276_{-0.010}^{+0.012}) \text{ GeV}^3$
$\langle \alpha G^2 \rangle$	$(7.0 \pm 1.3) \times 10^{-2} \text{ GeV}^4$
\bar{m}_c (2 GeV)	1.10 ± 0.03 GeV

TABLE III. Perturbative and nonperturbative parts of the moments $\chi_+^{(n)}$ for $n = 1, 2, 3$.

Moment, n	$\chi_+^{(n)NP} \times 10^{-3}$	$\chi_+^{(n)PT} \times 10^{-3}$	$\chi_+^{(n)} \times 10^{-3}$
1 (in GeV^{-2})	0.98 ± 0.25	6.37 ± 0.67	7.35 ± 0.89
2 (in GeV^{-4})	0.35 ± 0.12	0.80 ± 0.15	1.15 ± 0.26
3 (in GeV^{-6})	0.13 ± 0.05	0.14 ± 0.04	0.27 ± 0.09

TABLE IV. Upper bound for $|F_+(0)|$ and lower bound for $|V_{cd}|$ calculated for each moment.

Moment, n	$ F_+(0) $, upper bound	$ V_{cd} $, lower bound
1	0.529 ± 0.171	0.293 ± 0.097
2	0.847 ± 0.271	0.183 ± 0.060
3	1.593 ± 0.494	0.097 ± 0.030

uncertainties. The results are in the Table III. The non-perturbative piece can be written as

$$\chi_+^{(n)NP} = -\frac{1}{m_{c,\text{pole}}^{2(n+2)}} \left[\bar{m}_c \langle \bar{u}u \rangle + \frac{\langle \alpha G^2 \rangle}{12\pi} \right] \tag{36}$$

where $\langle \bar{u}u \rangle$ and $\langle \alpha G^2 \rangle$ are the quark and gluon condensates, respectively. These parameter values have been taken from Table II, and the updated values for the moments can be found in Table III.

As shown in Table I, we found F_1 and F_2 for our averaged system of neural networks to be $F_1 = (1.049 \pm 0.148) \times 10^{-1} \text{ GeV}^{-1}$ and $F_2 = (1.322 \pm 0.343) \times 10^{-1} \text{ GeV}^{-2}$. Using these two values with $|V_{cd}F_+(0)| = (15.52 \pm 1.07) \times 10^{-2}$, and plugging it into the inequality we obtained with Eq. (34), we can find an upper bound for the form factor at $q^2 = 0$ for each moment that has been calculated. The results can be found in Table IV. The results quoted in the table are consistent with the result $|V_{cd}| = 0.218 \pm 0.004$ quoted by the Particle Data Group (PDG) [37].

V. CONCLUSIONS

Accurate theoretical description of semileptonic form factor $F_+(q^2)$ are needed for accurate extraction of the CKM matrix elements V_{cd} and for studies of possible new physics contributions. While lattice QCD and QCD sum rules' calculations provide model-independent results for various portions of available q^2 range, extrapolations of $F_+(q^2)$ are often needed to extend the predictions to other values of q^2 , for which a particular shape of the q^2 -dependence is often used. What systematic uncertainty does choosing a particular function to describe a q^2 dependence of the form factors brings to such extrapolation? We performed the fit of the available experimental data to an artificial neural net, which was used in a capacity of a universal unbiased approximant. We found that the optimal configuration of each perceptron neural net to be (2,3,4,1), i.e., a net with two hidden layers with 3 (4) nodes in the first (second) hidden layer. The results of the ANN

training and relevant graphs are available at <https://s.wayne.edu/hepmachinelearning/>. While the simple ANNs employed in this paper do not provide spectacular extrapolation to $q^2 = 0$, the obtained results, displayed in Table I, can be used to test existing models of q^2 -dependence of the $F_+(q^2)$ form factor. Based on our fits, we conclude that it is possible that more effective poles need to be taken into account if model-inspired form factors are used for parametrizations of future experimental data. Finally, we used the resulting ANN fit to improve unitarity constraints on the form factor, which allowed for the model-independent bounds on V_{cd} .

We would like to note that while our system of ANNs was only trained with one set of data, we believe a completely trained system of ANNs, such as ours, can in principle be used for analyses of the same form factor in other datasets of exclusive semileptonic D decays.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Department of Energy under Contract No. de-sc0007983. We thank Gil Paz for reading the manuscript and helpful comments. A. A. P. thanks Roy Briere for useful and oftentimes sobering conversations. A. A. P. thanks the Institute for Nuclear Theory at the University of Washington for its kind hospitality and stimulating research environment. This research was also supported in part by the INT's U.S. Department of Energy Grant No. DE-FG02-00ER41132.

APPENDIX: HYPERPARAMETER TUNING OF ARTIFICIAL NEURAL NETWORKS

In this Appendix we discuss the optimization of artificial neural network in terms of its *hyperparameters*. The hyperparameters of a multilayer neural networks are

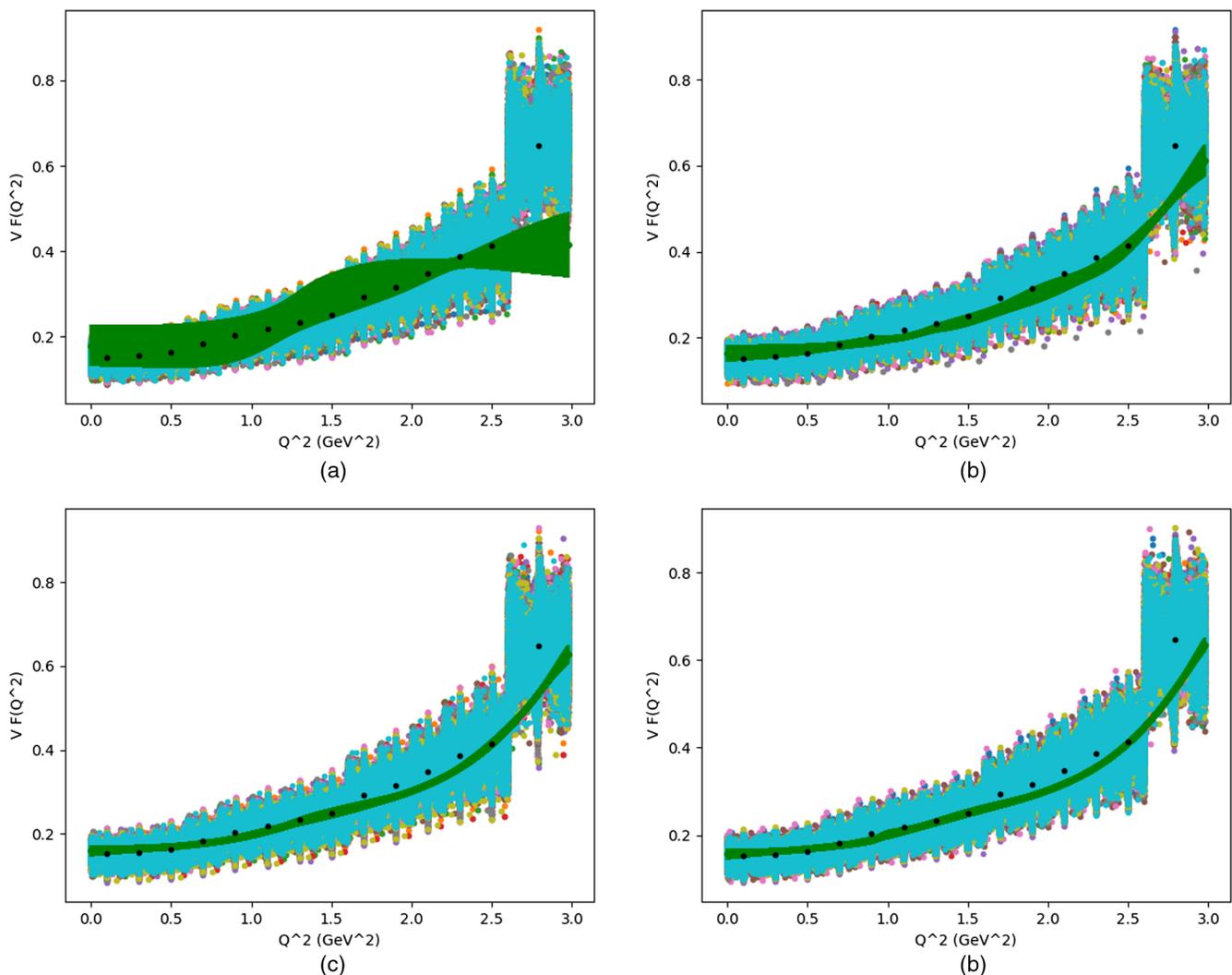


FIG. 4. Impact of the number of epochs per neural network training. ANN structure (2,3,4,1): (a) ANN fit for 20 epochs, (b) ANN fit for 30 epochs, (c) ANN fit for 40 epochs, and (d) ANN fit for 50 epochs.

defined as the number of hidden nodes in each layer, the number of training epochs per neural network, and the size of the training set per neural network. Here we concentrate on the case of ANN with two hidden layers.

In literature, the grid method and random search are two most popular ways to search for the optimal set of hyperparameter values. However, the grid method is considered to be a somewhat brute force method [39], so we decided to employ a randomly generated hyperparameter space for the following analysis. Specifically, we randomly generated the number of nodes in each layer and provided a representation of different configurations in the parameter space. In the following, the ANN fits were obtained by averaging over ten

ANNs, and we used 18,000 pseudodata points per ANN q^2 bin.

1. Impact of the number of epochs

To analyze the impact of the number of epochs per neural network, we generated the following figures. Starting from 20 epochs, we increased the amount up to 50 with ten epoch intervals. The structure of the neural networks is fixed at (2,3,4,1). Figure 4 shows the neural network output for different epochs.

The optimization algorithm iteratively updates the weights (ω) and cutoff (θ) in each epoch. A large number of epochs increase the probability of finding the global minimum of the error function. As a result, ANN fit gradually improved as the number of epochs increased.

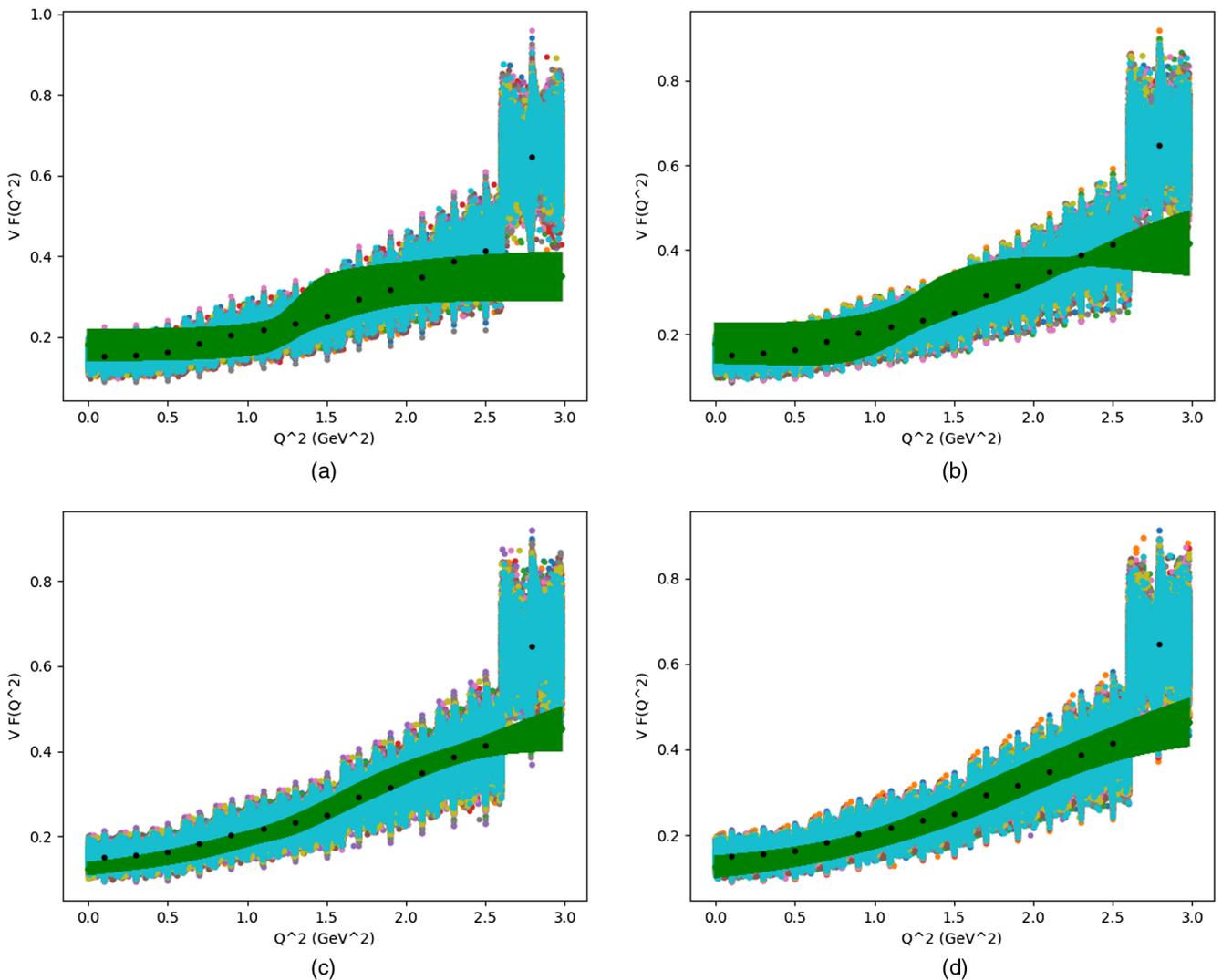


FIG. 5. Impact of the number of hidden nodes per neural network for 20 epochs: (a) ANN fit for (2,1,2,1), (b) ANN fit for (2,3,4,1), (c) ANN fit for (2,6,8,1), and (d) ANN fit for (2,15,12,1).

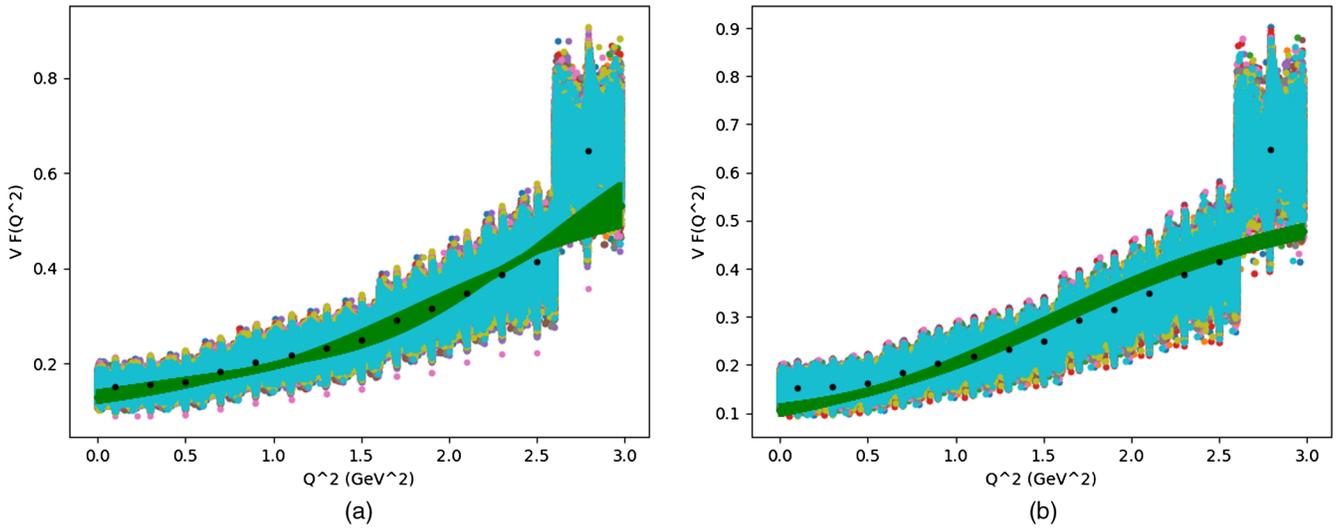


FIG. 6. Impact of the number of hidden nodes per neural network for 20 epochs: (a) ANN fit for (2,20,23,1), (b) ANN fit for (2,30,37,1).

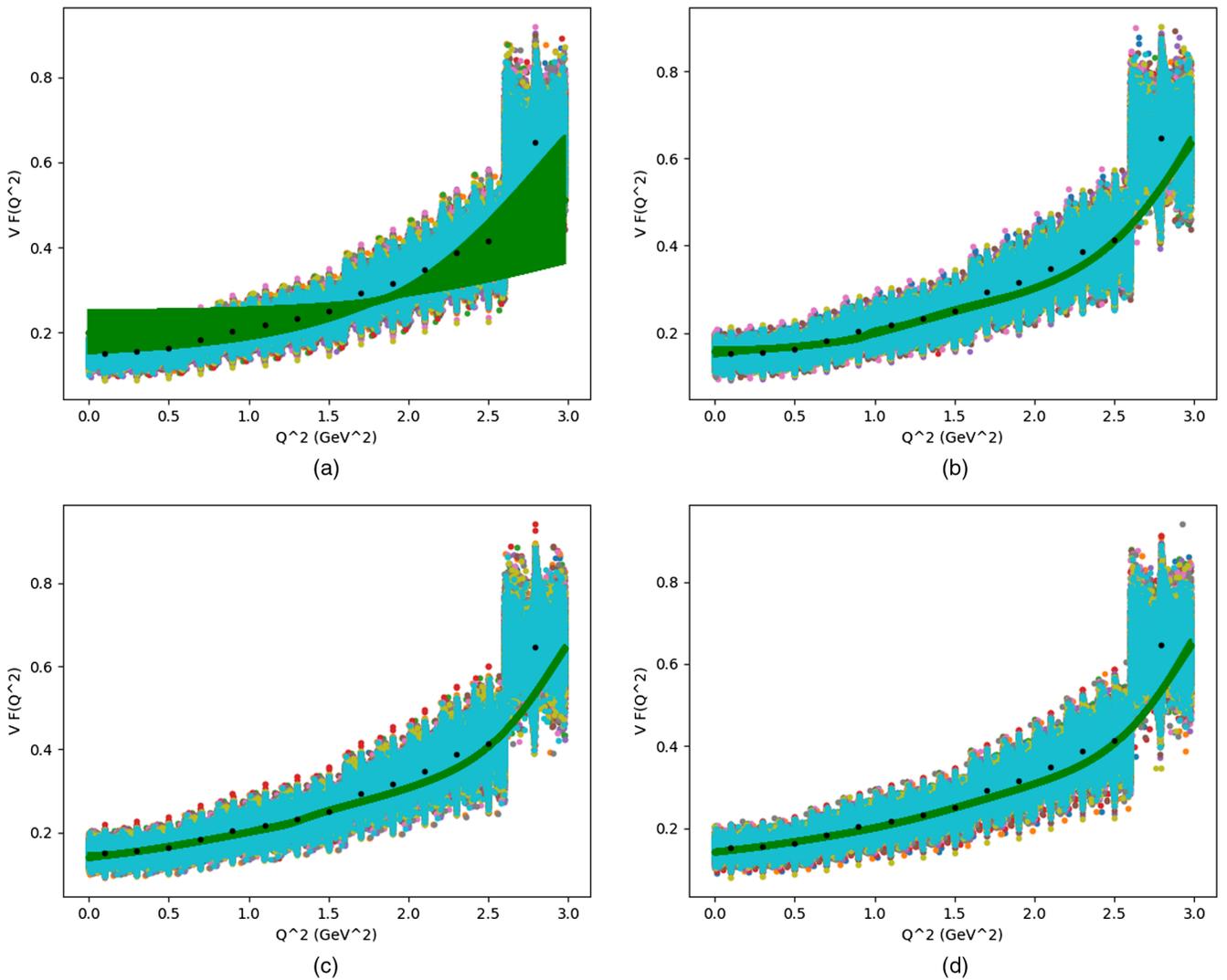


FIG. 7. Impact of the number of hidden nodes per neural network for 50 epochs: (a) ANN fit for (2,1,2,1), (b) ANN fit for (2,3,4,1), (c) ANN fit for (2,6,8,1), and (d) ANN fit for (2,15,12,1).

2. Impact of number of hidden nodes for small number of epochs

The number of epochs were fixed to 20. Starting from ANN structure (2,1,2,1) we increased the number of hidden nodes as (2,3,4,1), (2,6,8,2), (2,15,12,1), (2,20,23,1), and (2,30,37,1). Figures 5 and 6 show the impact of number of hidden nodes on ANN fits at the lower end of epoch space.

As expected, the neural networks fit gradually improved with the increasing number of hidden nodes.

3. Impact of the hidden nodes for large number of epochs

Similarly, we analyzed the impact of a large number of epochs. Starting from the ANN structure (2,1,2,1), we increased the number of hidden nodes per ANN while keeping the number of epochs fixed at 50. Figures 7 and 8 show the ANN fit obtained for (2,1,2,1), (2,3,4,1), (2,6,8,1), (2,15,12,1), (2,20,23,1) and (2,30,37,1).

At 50 epochs, we obtain a robust ANN fit with a relatively small number of hidden nodes. For example,

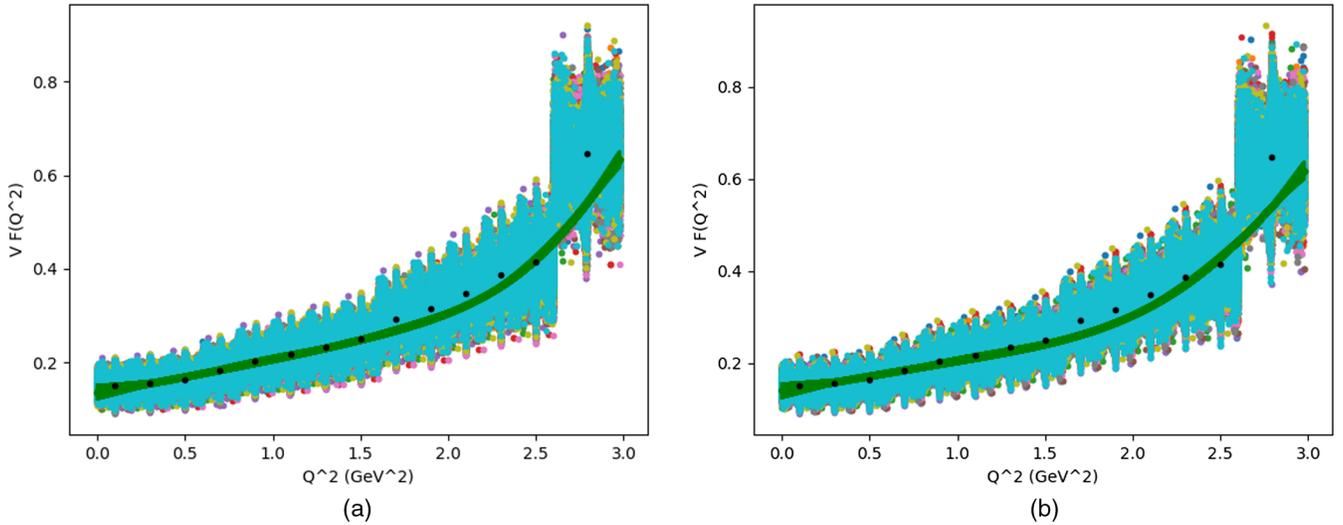


FIG. 8. Impact of the number of hidden nodes per neural network for 50 epochs: (a) ANN fit for (2,20,23,1), (b) ANN fit for (2,30,37,1).

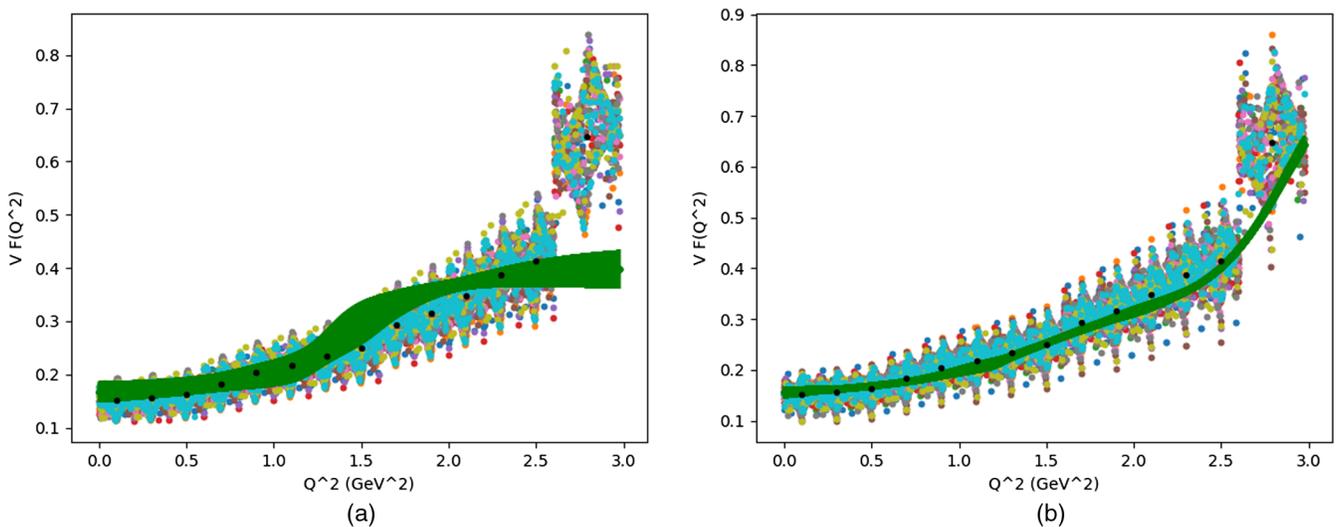


FIG. 9. Impact of the 180 pseudodata points per q^2 bin at 20 epochs and 50 epochs for ANN structure (2,3,4,1): (a) ANN fit for 20 epochs, and (b) ANN fit for 50 epochs.

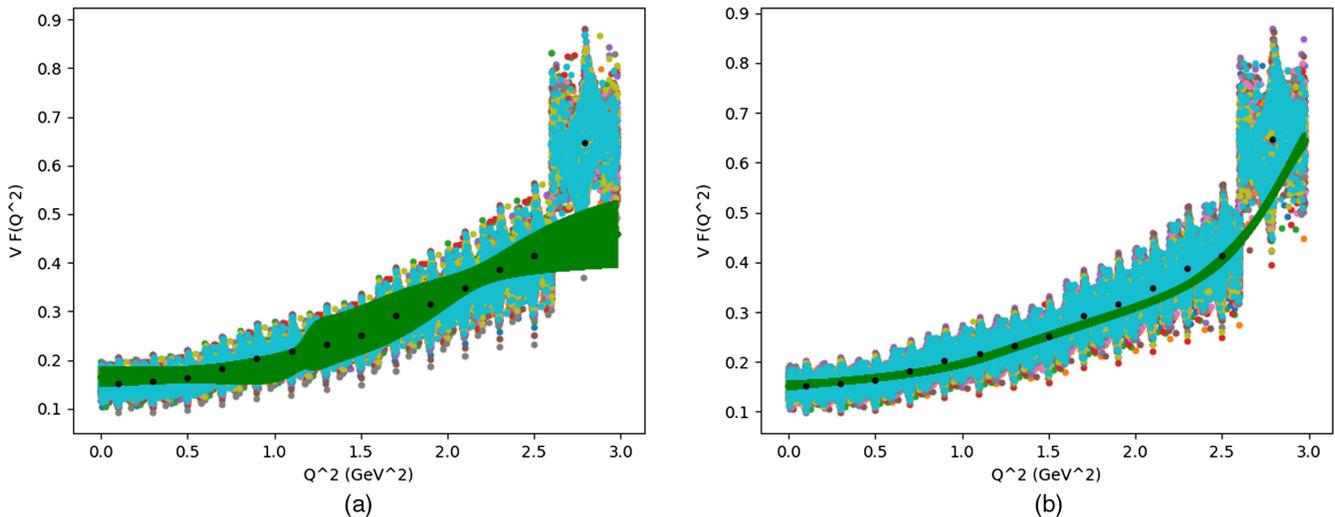


FIG. 10. Impact of the 1800 pseudo data per q^2 bin at 20 epochs and 50 epochs for ANN structure (2,3,4,1): (a) ANN fit for 20 epochs, and (b) ANN fit for 50 epochs.

at 50 epochs, our ANN fit starts to capture the experimental data set features for the structure (2,3,4,1). In comparison, at 20 epochs, a similar performance was observed only at the structure (2,15,12,1). On the other hand, from Figs. 7 and 8, we infer that at 50 epochs, the ANN's with a large number of hidden nodes started to overfit the data set.

4. Impact of the number of pseudodata points per q^2 bin

The number of pseudodata per q^2 bin was changed to 180. We start our analysis with 20 epochs and then increased to 50 while keeping the structure (2,3,4,1). Figure 9 provide the ANN fits for 20 epochs and 50 epochs.

Similarly, Fig. 10 provides the ANN fits for 1800 pseudodata points per q^2 bin. We trained the neural networks with 20 and 50 epochs while keeping the structure fixed at (2,3,4,1).

Figures 9 and 10 should be compared with the Figs. 5(b) and 7(b). We found that the number of pseudodata per bin does not play a significant role than the number of epochs. However, if we trained a large number of neural networks, then the number of pseudodata points per bin could have been an important parameter. Finally, we observed that 18,000 pseudo data per bin is adequate to train ten ANNs.

-
- [1] M. Artuso, B. Meadows, and A. A. Petrov, *Annu. Rev. Nucl. Part. Sci.* **58**, 249 (2008).
 - [2] M. Ablikim *et al.* (BESIII Collaboration), *Phys. Rev. Lett.* **122**, 011804 (2019).
 - [3] M. Ablikim *et al.* (BESIII Collaboration), *Phys. Rev. Lett.* **121**, 171803 (2018).
 - [4] C. Z. Yuan and S. L. Olsen, *Nat. Rev. Phys.* **1**, 480 (2019).
 - [5] L. Riggio, G. Salerno, and S. Simula, *Eur. Phys. J. C* **78**, 501 (2018).
 - [6] Y. S. Amhis *et al.* (HFLAV Collaboration), arXiv: 1909.12524.
 - [7] S. Aoki *et al.* (Flavour Lattice Averaging Group), *Eur. Phys. J. C* **80**, 113 (2020).
 - [8] A. Khodjamirian, C. Klein, T. Mannel, and N. Offen, *Phys. Rev. D* **80**, 114005 (2009).
 - [9] C. G. Boyd, B. Grinstein, and R. F. Lebed, *Phys. Rev. Lett.* **74**, 4603 (1995).
 - [10] T. Becher and R. J. Hill, *Phys. Lett. B* **633**, 61 (2006).
 - [11] B. Ananthanarayan, I. Caprini, and I. Sentitemsu Imsong, *Eur. Phys. J. A* **47**, 147 (2011).
 - [12] B. Grinstein and R. F. Lebed, *Phys. Rev. D* **92**, 116001 (2015).
 - [13] S. Descotes-Genon and A. Le Yaouanc, *J. Phys. G* **35**, 115005 (2008).
 - [14] See, e. g., R. N. Faustov, V. O. Galkin, and X. W. Kang, *Phys. Rev. D* **101**, 013004 (2020).
 - [15] D. Becirevic and A. B. Kaidalov, *Phys. Lett. B* **478**, 417 (2000).
 - [16] P. Ball and R. Zwicky, *Phys. Rev. D* **71**, 014015 (2005).
 - [17] F. Su and Y. Yang, *Int. J. Mod. Phys. A* **26**, 3185 (2011).
 - [18] A. N. Kolmogorov, *Dokl. Akad. Nauk SSSR* **114**, 953 (1957).
 - [19] K. Hornik, M. Stinchcombe, and H. White, *Neural Netw.* **2**, 359 (1989).
 - [20] G. Cybenko, *Math. Control Signals Syst.* **2**, 303 (1989).

- [21] S. Forte, L. Garrido, J. I. Latorre, and A. Piccione, *J. High Energy Phys.* **05** (2002) 062.
- [22] R. D. Ball *et al.* (NNPDF Collaboration), *J. High Energy Phys.* **04** (2015) 040.
- [23] J. C. Rojo, [arXiv:hep-ph/0607122](https://arxiv.org/abs/hep-ph/0607122).
- [24] K. M. Graczyk, P. Plonski, and R. Sulej, *J. High Energy Phys.* **09** (2010) 053.
- [25] L. Alvarez-Ruso, K. M. Graczyk, and E. Saul-Sala, *Phys. Rev. C* **99**, 025204 (2019).
- [26] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [27] J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer, New York, 1999).
- [28] M. F. Moller, *Neural Netw.* **6**, 525 (1993).
- [29] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design* (PWS Publishing, Boston, 1996).
- [30] M. Ablikim *et al.* (BESIII Collaboration), *Phys. Rev. D* **92**, 072012 (2015).
- [31] S. G. Krishna and K. K. Sahu, [arXiv:1503.06462](https://arxiv.org/abs/1503.06462).
- [32] F. S. Wong, *Neurocomputing; Variable Star Bulletin* **2**, 147 (1991).
- [33] G. Paz, [arXiv:1909.08108](https://arxiv.org/abs/1909.08108).
- [34] A. Gunawardana and G. Paz, *J. High Energy Phys.* **11** (2019) 141.
- [35] A. Khodjamirian and A. A. Petrov, *Phys. Lett. B* **774**, 235 (2017).
- [36] S. Narison, *Phys. Lett. B* **706**, 412 (2012).
- [37] M. Tanabashi *et al.* (Particle Data Group), *Phys. Rev. D* **98**, 030001 (2018).
- [38] K. G. Chetyrkin and M. Steinhauser, *Eur. Phys. J. C* **21**, 319 (2001).
- [39] Analytics Vidhya, <https://medium.com/analytics-vidhya/hyperparameter-search-part-1-2b67fd7a71d8>.