





Empirical fits to inclusive electron-carbon scattering data obtained by deep-learning methodsBeata E. Kowal ^{*}, Krzysztof M. Graczyk [†], Artur M. Ankowski ,
Rwik Dharmapal Banerjee , Hemant Prasad, and Jan T. Sobczyk*Institute of Theoretical Physics, University of Wrocław, Plac Maxa Borna 9, 50-204 Wrocław, Poland*

(Received 3 January 2024; revised 4 June 2024; accepted 15 July 2024; published 6 August 2024)

Employing the neural network framework, we obtain empirical fits to the electron-scattering cross sections for carbon over a broad kinematic region, extending from the quasielastic peak through resonance excitation to the onset of deep-inelastic scattering. We consider two different methods of obtaining such model-independent parametrizations and the corresponding uncertainties: based on the bootstrap approach and the Monte Carlo dropout approach. In our analysis, the χ^2 defines the loss function, including point-to-point and normalization uncertainties for each independent set of measurements. Our statistical approaches lead to fits of comparable quality and similar uncertainties of the order of 7%. To test these models, we compare their predictions to test datasets excluded from the training process and theoretical predictions obtained within the spectral function approach. The predictions of both models agree with experimental measurements and theoretical calculations. We also perform a comparison to a dataset lying beyond the covered kinematic region, and find that the bootstrap approach shows better interpolation and extrapolation abilities than the one based on the dropout algorithm.

DOI: [10.1103/PhysRevC.110.025501](https://doi.org/10.1103/PhysRevC.110.025501)**I. INTRODUCTION**

The stringent precision requirements of the long-baseline neutrino-oscillation program create an urgent need to significantly improve the accuracy of available estimates of the cross sections for neutrino interactions with atomic nuclei [1,2]. The complexity of this problem stems from multiple factors: the process of scattering involves multicomponent vector and axial currents, different interaction mechanisms can produce the same final states, experimental data with the desired precision are only available for nuclear cross sections, and flux averaging greatly diminishes differences between different interaction channels.

A widely recognized way to address those difficulties is to leverage the similarities of electron and neutrino interactions [3]. In electron scattering, only electromagnetic currents contribute, cross sections are higher by several orders of magnitude, beams are monoenergetic and adjustable, and precise data can be collected not only for complex nuclei, but also for deuterium and hydrogen. These advantages allow nuclear effects to be studied in detail using electron-scattering data.

Provided that the cross sections for neutrinos and electrons are calculated consistently and are sufficiently constrained by precise electron data, the only significant source of uncertainties in the neutrino predictions would be related to the

axial contributions, to be determined from the near-detector measurements, possibly making use of the guidance provided by the lattice quantum chromodynamics computations.

In parallel to the efforts of developing models of nuclear response, it would be useful to explore novel techniques for obtaining model-independent estimates of nuclear cross sections. Deep neural networks (DNNs) are ideally equipped to provide such predictions. Indeed, DNNs are known for their excellent adaptive abilities [4], which enabled obtaining unbiased parametrizations of nuclear cross sections [5], parton distribution functions [6,7], vector [8–11] and axial [12] nucleon form factors, deeply virtual Compton form factors [13], and nuclear charge radii [14].

In this article, we explore methods of obtaining such parametrizations of electron-scattering cross sections, independent of nuclear model assumptions, and pay special attention to estimating their uncertainties. For an analysis of this type, the abundance of available experimental data is critical.

We consider the carbon target, for which experimental cross sections have been collected in various laboratories worldwide over the last five decades and span the broadest kinematic region [15]. These features are of paramount importance to maximize the predictive abilities of the resulting DNN models in the broad kinematic region of relevance for neutrino-oscillation experiments.

Using DNNs to parametrize electron-scattering data, we obtain both the best fits and an estimate of the associated uncertainties. In this analysis, we consider two different methods.

In the first one, we adopt an ensemble approach, bootstrap-like model [16,17], similar to the one developed by the Neural Network Parton Distribution (NNPDF) group [6]. The experimental data are used to generate bootstrap datasets. For each

^{*}Contact author: beata.kowal@uw.edu.pl[†]Contact author: krzysztof.graczyk@uw.edu.pl

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. 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.

of the datasets, a DNN fit is obtained. The model's prediction is obtained by averaging all the fits, and the square root of the variance gives the uncertainty.

In the second approach, we employ the Monte Carlo (MC) dropout technique [18], in which DNN units are temporarily removed at random to prevent overfitting. This procedure is repeatedly used during the training and inference stages, resulting in random responses of the DNN. These responses are then averaged over to obtain the DNN prediction, and their distribution provides the uncertainty estimate.

Our analysis pays special attention to systematic uncertainties related to data normalization, as previously done in Refs. [9–11]. We introduce them as a penalty term in the loss function. We use a DNN architecture with ten hidden layers and address the vanishing gradient problem (VGP) by implementing batch normalization layers [19]. Our models are validated by comparisons to the test dataset and with predictions of the spectral function approach [20]. Eventually, we compare the model's prediction with higher energetic data from Ref. [21].

A similar analysis of electron-scattering data was recently performed by Al Hammal *et al.* [5], who considered nuclear targets ranging from ^4He to ^{59}Ni . Their loss function did not account for experimental uncertainties and was given as a relative absolute error. However, an effort was made to estimate the uncertainties due to the model parameter variations. The authors of Ref. [5] adopted a network with 35 hidden layers and relied on the residual mechanism for the VGP. To test their model, Al Hammal *et al.* compared their results with the predictions of the code GIBUU [22] and the SuSAv2 approach [23].

The DNN approaches do not make any assumptions on whether the interaction takes place at the nucleus or nucleon level and do not separate the longitudinal and transverse response functions. As a consequence, the DNN approaches are conceptually different from the phenomenological approaches that were successfully developed in the literature, e.g. for electron scattering on the proton [24], deuteron [25], and carbon and oxygen [26,27] targets.

Our paper is organized as follows. In Sec. II, we discuss the experimental data underlying our analysis and provide the rationale for the imposed kinematic cuts. In Sec. III, we introduce our statistical model, providing details of the loss function, the methods for estimating uncertainties, and the network architectures. In Sec. IV, we report and analyze our numerical results. Finally, we state our conclusions in Sec. V.

II. ELECTRON-SCATTERING DATA

For a DNN to successfully learn electron-scattering cross sections, it is essential to provide it with a sufficient amount of experimental data. In this analysis, we consider the carbon target, which has been studied extensively in past experiments; see Ref. [15] for a recent review.

The available data [28–39] are in the form of double-differential cross sections $d^2\sigma/d\omega d\Omega$ for inclusive electron scattering—in which only the final kinematics of the electron is measured—at various beam energies E and scattering

TABLE I. Summary of the data used in this analysis. The numbers of points refer to the data surviving the cut (1).

Reference	Abbrev.	Norm. uncert.	Number of points
Arrington <i>et al.</i> [28]	Arri1995	4.0%	56
Arrington <i>et al.</i> [29]	Arri1998	4.0%	398
Bagdasaryan <i>et al.</i> [30]	Bagd1988	10.0%	125
Baran <i>et al.</i> [31]	Bara1988	3.7%	259
Barreau <i>et al.</i> [32]	Barr1983	2.0%	1243
Dai <i>et al.</i> [33]	Dai2018	2.2%	177
Day <i>et al.</i> [34]	Day1993	3.4%	316
Fomin <i>et al.</i> [35]	Fomi2010	4.0%	359
O'Connell <i>et al.</i> [36]	O'Con1987	5.0%	51
Sealock <i>et al.</i> [37]	Seal1989	2.5%	250
Whitney <i>et al.</i> [38]	Whit1974	3.0%	31
Total			3265

angles θ , given as a function of the energy ω that the interaction transferred to the nucleus.

We consider a broad kinematic region, in which different interaction mechanisms are known to play an important role, starting from quasielastic scattering at low energy transfers, to pion production through resonance excitation in the intermediate regime, and to the onset of deep-inelastic scattering at the highest ω values.

At very low values of energy transfer, elastic scattering off the whole nucleus may occur, and inelastic interactions with the nucleus may lead to an excitation of the giant dipole resonance or a discrete nuclear state. Due to the scarcity of data in this region, accounting for these processes is not currently feasible in our approach. We remove their contributions by applying a cut,

$$\omega > \omega_{\min} = \frac{M_A E_x + 0.5E_x^2 + E^2(1 - \cos\theta)}{M_A + E(1 - \cos\theta)}, \quad (1)$$

where ω_{\min} is the energy transfer required to excite the nucleus of the mass M_A with the energy E_x , and θ is the scattering angle. Typically, $\omega_{\min} \approx E_x + t_A$, with t_A being the nuclear recoil energy. In our analysis, the E_x value is set to 26 MeV [20].

We do not include the measurements reported by Gomez *et al.* [21], who studied how the EMC effect depends on the nuclear mass number. Those seven points, each collected for a different $(E, \cos\theta)$ pair, provide useful constraints on theoretical models, but they cannot be readily accommodated within our machine-learning approach. We make use of those data to investigate the extrapolation properties of our models.

The kinematic region spanned by the data used in our analysis is presented in Fig. 1, projected in the (ω, Q^2) and $(\omega, \cos\theta)$ planes, Q^2 being the four-momentum transfer. While at low Q^2 and at low θ there is an abundance of the experimental data, this is not the case in general. Even for carbon, the best-studied nuclear target, there are still large swaths with very scarce information on the cross sections. This issue is challenging to machine-learning models based on data alone.

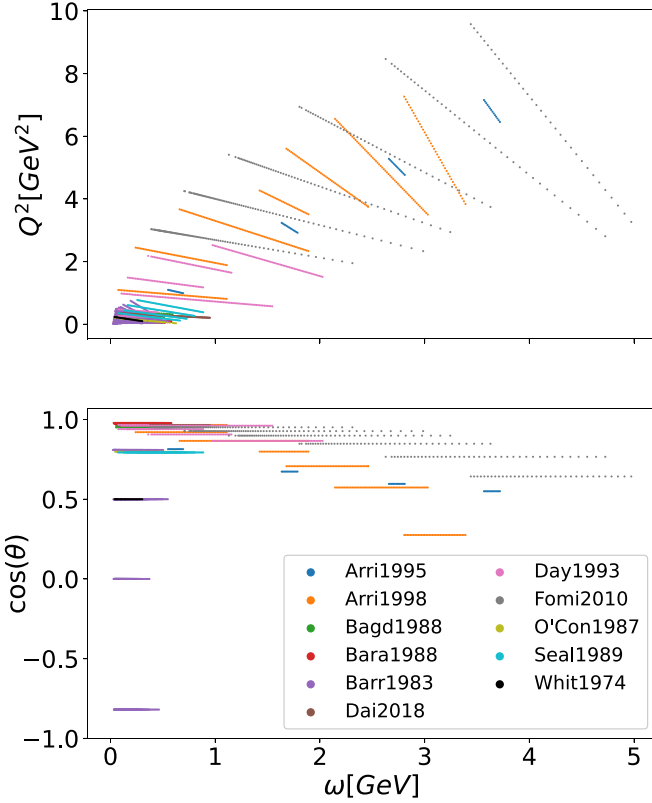


FIG. 1. Kinematic domain covered by the experimental data [28–38] considered in our analysis, shown in the planes of (top) energy transfer and four-momentum transfer, and (bottom) energy transfer and cosine of the scattering angle.

In Table I, we summarize the experimental data and provide the normalization uncertainties (“Norm. uncert.”) assumed in our analysis.

III. NEURAL NETWORK FRAMEWORK

A. Likelihood analysis

As discussed in the previous section, in this analysis, we consider 11 independent datasets. We denote the k th dataset containing N_k points as

$$\mathcal{D}_k = \{(E_k^i, \theta_k^i, \omega_k^i, d\sigma_k^i, \Delta d\sigma_k^i) : i = 1, \dots, N_k\}, \quad (2)$$

where $d\sigma_k^i$ and $\Delta d\sigma_k^i$ are the i th measurement in k th dataset and its corresponding uncertainty, and N_k is number of data points in the k th dataset. The full dataset reads $\mathcal{D} = \mathcal{D}_1 \cup \dots \cup \mathcal{D}_{11}$. The total uncertainty $\Delta d\sigma_k^i$ combines the statistical and systematic uncertainties, characteristic to a given experiment.

The global analyses of data for electron-proton scattering [40,41], charged current single pion production [42], and quasielastic neutrino-deuteron scattering [12] found that including normalization uncertainties is essential to obtain consistent fits.

In the present paper, we follow the same philosophy. To fit the data, we introduce the following loss function [9]:

$$\chi_{\text{tot}} = \sum_{k=1}^{11} \left[\chi_k^2(\lambda_k) + \frac{1}{2} \left(\frac{1 - \lambda_k}{\Delta \lambda_k} \right)^2 \right], \quad (3)$$

where

$$\chi_k^2(\lambda_k) = \frac{1}{2} \sum_{i=1}^{N_k} \left(\frac{d\sigma_k^i - \lambda_k d\sigma_k^{\text{fit}}(E_k^i, \theta_k^i)}{\Delta d\sigma_k^i} \right)^2, \quad (4)$$

with $\Delta \lambda_k$ denoting the overall systematic uncertainty for k th dataset, and λ_k being a subject of optimization. In order to optimize the λ_k 's, we follow the algorithm proposed in Refs. [9–11]. The obtained normalization parameters are given in Table II.

B. Neural network models

Our analysis aims to obtain a function that returns the value of differential cross section for electron-carbon scattering, given an input value of energy, scattering angle, and energy transfer. We use a DNN parametrization for this purpose.

While DNNs are known for their adaptive ability, they are also known for their limitations [43,44]. For example, when the input or output data values are too large (in absolute value), the network is unable to relate the input to the output, or the optimization algorithms work ineffectively.

In our problem, the output is the cross section, which ranges from 10^{-5} to 10^6 nb/(sr GeV). Therefore, to perform the optimization, we rescale the output as

$$d\sigma \rightarrow \left(\frac{10^9}{137^2 E \cos(\theta/2)} \frac{\cos^2(\theta/2)}{4E^2 \sin^4(\theta/2)} \right)^{-1} d\sigma, \quad (5)$$

where the beam energy, E , is given in GeV and the double differential cross section, $d\sigma$, in nb/(sr GeV).

To accelerate the optimization process and help the DNN capture the proper normalization, we extend the number of input variables from three to five: $(E, \omega, \theta, \cos \theta, Q^2)$. The last two are the functions of the first three, but keeping such input accelerates the optimization. The same procedure was employed by NNPDF group [6] to fit parton distribution functions, and by Al Hammal *et al.* [5] to fit electron-nucleus scattering data.

We consider two types of DNN architecture. The first one, statistical model A, consists of the input layer, followed by ten blocks and an output layer. The network structure is shown in Fig. 2. The input is five-dimensional, and the output is one-dimensional. Each block has a hidden layer with 300 units fully connected with the previous module and batch-normalization layer. We keep rectified linear unit (ReLU) activation functions in each module. The output activation function is given by a sigmoid function. The batch-normalization layer [19] is essential to obtain successful fits. This type of layer helps to maintain a proper normalization of the outputs that, as a result, prevents the gradients from vanishing and the optimization process from stopping, which is a problem typical for DNNs. Additionally, the batch-normalization layer naturally regularizes the model



FIG. 2. The neural network architecture for model A. The light gray box corresponds to the hidden layer of fully connected neurons with ReLU activation function. The dark gray box denotes the batch normalization layer. In the output, there is the sigmoid function. In the architecture for model B, not depicted, each fully connected layer is followed by the dropout layer. In both models A and B, the network has ten layers of hidden units, consisting of 300 units each.

parameters. Hence, such a model should not tend to overfit the data.

The second network, statistical model B, has a similar structure, but each module consists of a fully connected layer followed by a batch-normalization layer and the dropout layer. The dropout layer was introduced by Srivastava *et al.* [18] to regularize the network parameters, prevent the model from overfitting and, as a consequence, increase the predictive abilities of the networks.

C. Uncertainty estimation

1. Uncertainties in deep learning

One of the challenges in modeling data using DNNs is obtaining systems with good generalization abilities. Many adaptive units define the DNNs and can easily overfit the data, resulting in poor generalization. Another challenge is estimating how uncertain the networks are in their predictions.

Evaluation of uncertainties in deep neural network predictions is difficult. DNNs are defined by many parameters, and standard statistical techniques either do not work or are inefficient. There are two primary sources of (predictive) uncertainty: epistemic (model’s uncertainties due to its parameter dependence and structure) and aleatoric (data uncertainty) [45].

In recent years, an effort has been made to develop methods for estimating the predictive uncertainties [46]. However, an optimal method has not been established yet, as each of the developed techniques is known to have some limitations and problems.

There are two classes of the most popular methodologies for estimating uncertainties [46–48]: the ensemble and Bayesian methods. In the first one, some number (from dozens to thousands) of the networks are trained. The model’s prediction is given by the mean of the model’s responses, and the uncertainty is determined from their standard deviation. The second class of methods has roots in Bayesian statistics, which offers tools that capture both epistemic and aleatoric uncertainties.

An example of an ensemble method is the bootstrap approach, which originates from frequentist statistics [49] and

can be considered in nonparametric and parametric forms. In the first case, the idea is to create B bootstrap sets by sampling with repetition. In the other, the data are drawn from the generating probability distribution. It is crucial to notice that the bootstrapping leads to the *poor man’s Bayes posterior* [50]. Hence, one should expect that bootstrap and Bayesian approaches lead to similar density distributions.

The bootstrap technique for neural networks was adapted independently by Tibshirani [16] and Breiman [17]. This method is known in the literature as bootstrap aggregation or bagging (see Refs. [17,43]), and it is commonly used in the neural network community. One reason for its popularity is its low computational requirements compared to Bayesian techniques. In fact, the recommended number of neural models trained on the bootstrap datasets varies from 25 to 200.

The bootstrap approach estimates both types of predictive uncertainties due to the uncertainties in measurements and the variation of network parameters. Moreover, as we remarked above, bootstrapping leads to a similar “posterior” distribution as sampling in parameter space in Bayesian Monte Carlo approaches. In the first, the prior probabilities are modeled by the generated bootstrap datasets; in the other, they are postulated based on objective Bayesian assumptions.

The Bayesian methods are considered state-of-the-art for estimating uncertainties. The main goal is to evaluate the set of posterior density probabilities. The Bayesian approach allows us to predict both aleatoric and epistemic uncertainties. Moreover, within the Bayesian approach, it is possible to compare the various neural network models. The main difficulty is evaluating the posterior densities.

One popular method with roots in Bayesian statistics but is simple in implementation is the Monte Carlo dropout technique [51]. It is based on the observation that data analysis using a one-hidden neural network with an infinite number of neurons corresponds to the Gaussian process [52], one of the well-formulated methods of Bayesian statistics [53]. A single-fit network model is obtained as the result of the MC dropout.

2. Models A and B

We consider two models to estimate the uncertainties of the network predictions. In model A, we adapt the bootstrap method as Refs. [6,7]. Its idea is to use experimental measurements to produce the so-called bootstrap datasets and fit every set by the corresponding network model. The number of data points in the bootstrap dataset is the same as in the original one. A bootstrap point is obtained by sampling from a normal distribution. If the original data point is characterized by central value $d\sigma_k^i$ and uncertainty $\Delta d\sigma_k^i$, the bootstrap datum reads

$$d\sigma_k^{\text{clone},i} = d\sigma_k^i + r\Delta d\sigma_k^i, \quad (6)$$

where r is drawn from the standard normal distribution. We treat the normalization uncertainties as described in the previous sections and introduce χ^2 as a loss function to optimize the network’s parameters.

We split the full dataset into the training and test datasets in proportions 9 to 1. For the training data, 50 bootstrap

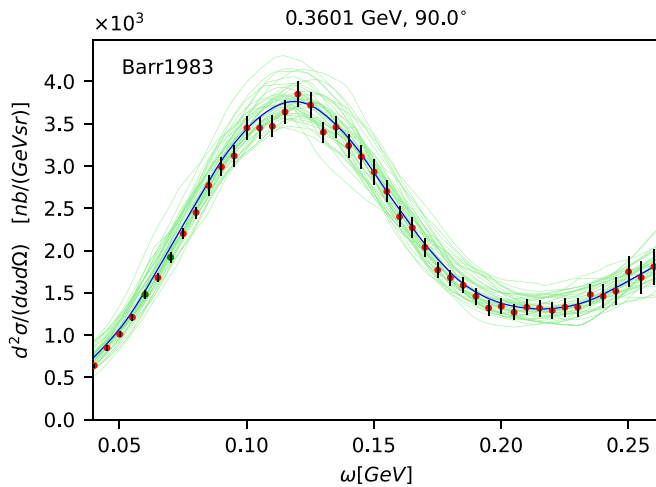


FIG. 3. Illustration of the algorithm used in model A to estimate the fit's uncertainty. Bootstrap datasets are generated from the original data according to the normal distribution (not depicted), and fits are performed 50 times (green lines). The cross section's estimate corresponds to the mean of the fits (black line), and the uncertainty corresponds to their standard deviation (not shown). The data are taken from Ref. [32].

datasets are created. Then, we obtain the network fit for every bootstrap dataset. The mean gives the model's response over predictions of all 50 fits, while the corresponding square root of variance defines uncertainty. In Fig. 3 we illustrate how the algorithm works. The figure shows the experimental data together with 50 fits to the bootstrap datasets.

In model B, we use the Monte Carlo dropout technique [18]. Its idea is to keep the dropout layer after every layer of units. This layer switches off each unit in the network with user-defined probability p . Initially, this method was introduced to regularize the training process to prevent overfitting and improve the model's generalization ability. However, when one keeps the dropout layer active in the inference mode, the network's prediction becomes random. To obtain the network prediction, one computes the network's response dozens of times (in our case, 50 times) and takes the average [51]. The uncertainty is given by the corresponding variance. In this method, the predictive uncertainty and the model's performance depend on the hyperparameter p , the dropout rate, which must be fine tuned.

We will employ two criteria to fine-tune the value of p . The first criterion is to yield the model with the lowest generalization error on the test dataset. The second criterion is for the model to reproduce the results of model A. As discussed in Sec. IV, we determine the dropout rate $p = 0.01$ to give the best MC dropout model.

3. Overfitting

Neural network models can have problems with generalization. If a model has a low capacity for describing analyzed data, it underfits it. On the other hand, when a model is too complex (that is, it has too many adaptive units), it may overfit the data. In both cases, the predictive abilities of the model are low. This problem is known as a bias-variance dilemma [54].

TABLE II. Values of the normalization parameters λ obtained in models A and B.

Abbrev.	Norm. uncert.	Model A	Model B
Arri1995	4.0%	1.01	1.02
Arri1998	4.0%	1.00	0.96
Bagd1988	10.0%	1.03	1.06
Bara1988	3.7%	1.01	0.98
Barr1983	2.0%	0.99	1.02
Dai2018	2.2%	1.00	0.97
Day1993	3.4%	0.99	0.98
Fomi2010	4.0%	1.01	0.96
O'Con1987	5.0%	1.02	1.01
Seal1989	2.5%	1.02	1.04
Whit1974	3.0%	0.93	0.93

The idea is to find the optimal neural network that describes the analyzed data well and has good predictive abilities.

There are various methods developed to prevent models from data overfitting. The simplest solution is to add a penalty term (regularization) to the loss, so that the algorithm does not reach the local minimum of the loss but converges to the configuration of weights in the neighborhood of the minimum.

In our analyses, we adopt that method. We optimize the model parameters using the AdamW algorithm with the weight decay of 0.004, determined during the pre-analyses. Having non-zero weight decay corresponds to considering the loss of the form

$$\chi_{\text{tot}}^2(\text{mini batch}) + \frac{\text{weight decay}}{2} \sum_i w_i^2, \quad (7)$$

where w_i 's are the weights (parameters) of the network.

Analyzing model A, we average over an ensemble of models. Consequently, even if a single model overfits the bootstrap data, this is not the case for the average prediction. In model B, the dropout layers regularize the model. Both analyses consider DNNs with batch normalization layers, which also naturally regularize the model.

Finally, we would like to mention that there is a significant difference between the generalization abilities of the small models (low number of adaptive units), such as one-hidden layer neural networks, and DNNs employed in this article. When the number of data constraints (in our case, data points) is much smaller than the number of model parameters, the DNNs naturally tend to generalize well, even without regularization procedures [55]. Here, the neural network is defined by more than 800 000 parameters, while the number of data points is only about 3000.

TABLE III. Generalization error on the test dataset for model B, with the total number of data points N .

p	0.005	0.01	0.02	0.03	0.04	0.10	0.16	0.20
$\chi^2/N(\text{test})$	1.60	1.15	2.05	2.32	2.79	2.91	3.77	5.23

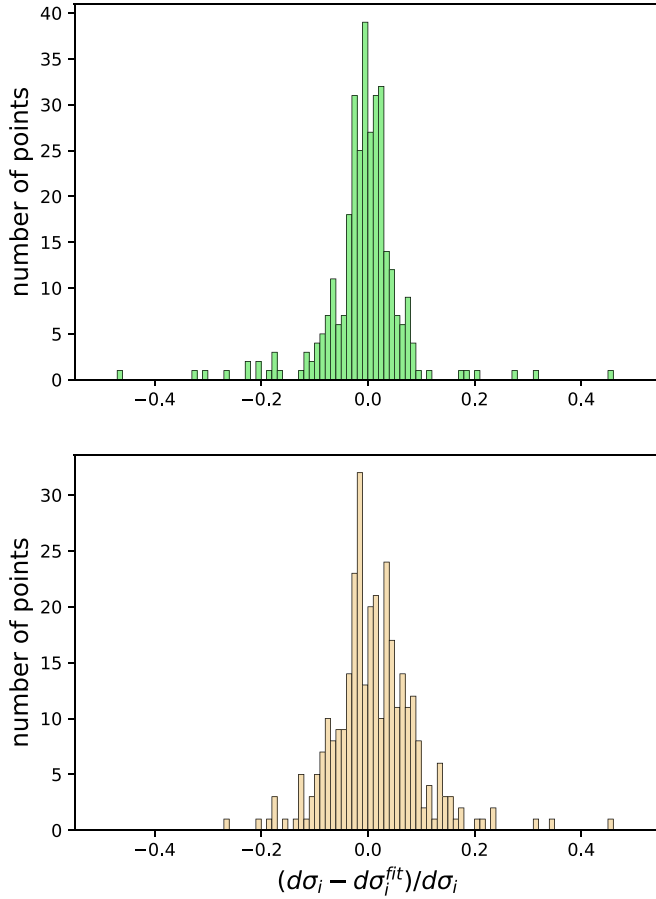


FIG. 4. Histograms of the normalized residuals, $(d\sigma_i - d\sigma_i^{\text{fit}})/d\sigma_i$, for (top) model A and (bottom) model B. Test data are included.

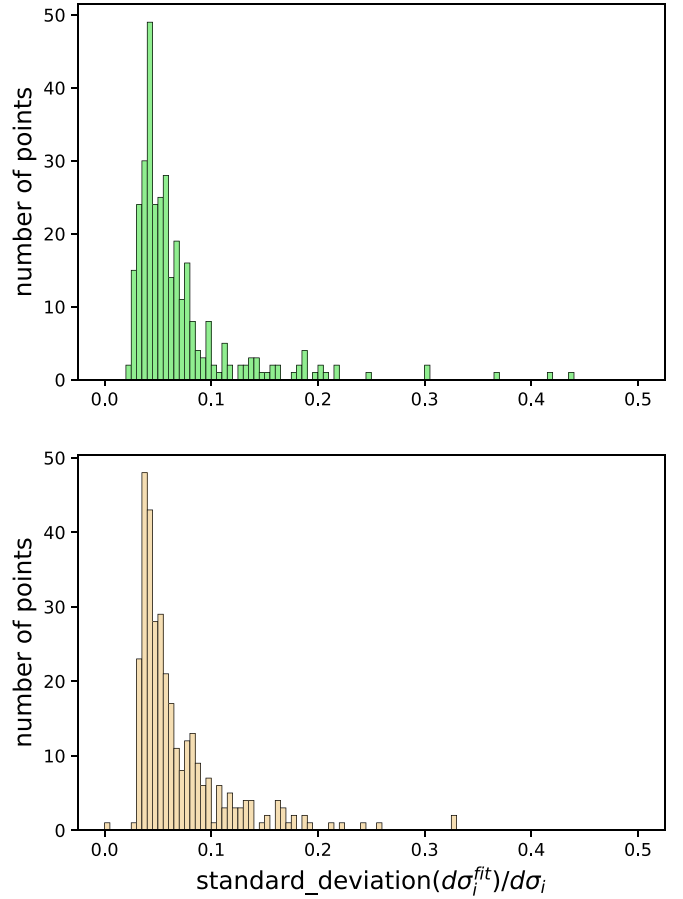


FIG. 5. Histograms of the standard deviation normalized by the experimental central value of (top) model A and (bottom) model B. Test data are included.

IV. NUMERICAL RESULTS

We performed both analyses employing the JAX package [56] and cross-checked the obtained results by using the KERAS package [57]. We ran optimization for no more than about 60 000 epochs. To optimize the models, we used the AdamW algorithm [58]. The models were regularized by keeping the value of weight decay as 0.004. The training was done in the minibatch configuration with five batches.

As discussed in the previous section, model B must be calibrated. We considered the network with several values of dropout rate p . The model with $p = 0.01$ was characterized by the lowest value of χ^2 computed on the test dataset; see Table III.

We also found that when $p = 0.01$ in model B, the obtained distributions of normalized residuals and normalized standard deviations were similar to those for model A. As an example, in Figs. 4 and 5, we show the histograms computed for the

test dataset. Additionally, in Table IV, we present the mean of normalized standard deviations obtained for model B with various p rates. In the considered range, the lower the value of p , the lower the mean uncertainty. As the mean in model A is about 7%, the consistency requirement favors the value $p = 0.01$ in model B. As both the χ^2 and model-consistency arguments support the dropout rate $p = 0.01$, we employ this value in our calculations with model B.

In Figs. 6 and 7, we present comparisons of the predictions of models A and B to six datasets [29,32,34,37]. The data span a broad kinematic region, corresponding to energy transfers between ≈ 30 MeV and ≈ 1.9 GeV, and momentum transfers $|\mathbf{q}|$ between ≈ 0.14 and ≈ 2.7 GeV.

The first six panels of Figs. 6 and 7 sample the $|\mathbf{q}|$ values below 1 GeV, where the cross section is dominated by quasielastic scattering and the Δ -resonance excitation, and nuclear effects are the most pronounced. Model A

TABLE IV. The mean of normalized standard deviations (Norm. uncert.) obtained for model B with various p rates.

p	0.005	0.01	0.02	0.03	0.04	0.10	0.16	0.20
Mean (Norm. uncert.)	0.05	0.07	0.08	0.10	0.12	0.14	0.21	0.27

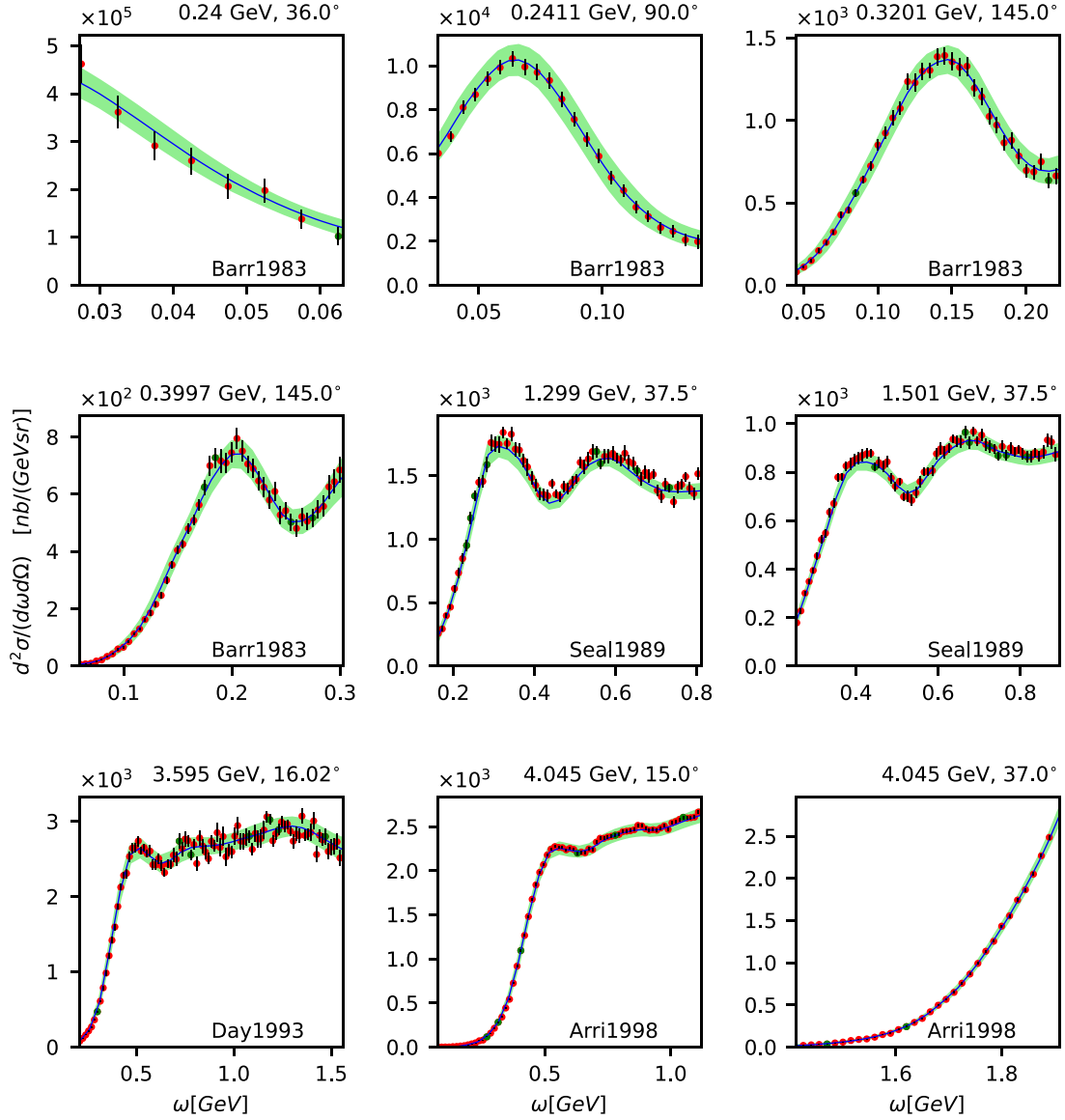


FIG. 6. Double-differential cross section $d^2\sigma/d\omega d\Omega$ for inclusive electron scattering on carbon. We compare the predictions of model A to the experimental data from Refs. [29,32,34,37]. The shaded areas denote the 1σ uncertainties. The panels are labeled with the beam energy and scattering angle values. The red (blue) points represent the training (test) dataset. The predictions are not rescaled according to the determined normalization parameters.

reproduces the data with excellent accuracy, both at low and high scattering angles, although contributions of the mechanism of interactions involving two-body currents are expected to be very different in these regimes. It outperforms model B, for which the agreement is very good nevertheless.

The last three panels of Figs. 6 and 7 probe the transition of the cross section from the region where excitation of higher resonances plays an important role to where it can be described in terms of deep-inelastic scattering. Despite of the complexity of the interaction mechanisms at these kinematics, both models are able to reproduce the data with remarkable accuracy. It is important to keep in mind that the normalization parameters are *not* included in the results presented in Figs. 6 and 7.

As discussed in Sec. III, the normalization parameters are incorporated in our fits to the data, giving additional freedom to the parameter space. Table II shows that their values typically agree with the unity to about 2% for model A and 4% for model B.

Interestingly, the normalization parameters for the data reported by Whitney *et al.* [38] obtained in both models are low but consistent. It suggests that there might be tension at the level of $\approx 7\%$ between the normalization of this dataset and those of the other measurements, with the results of Ref. [38] being overestimated.

We refer the reader to the Supplemental Material [59] for comparisons of the predictions of both the models to all the measurements considered in this analysis, including the dataset of Whitney *et al.*

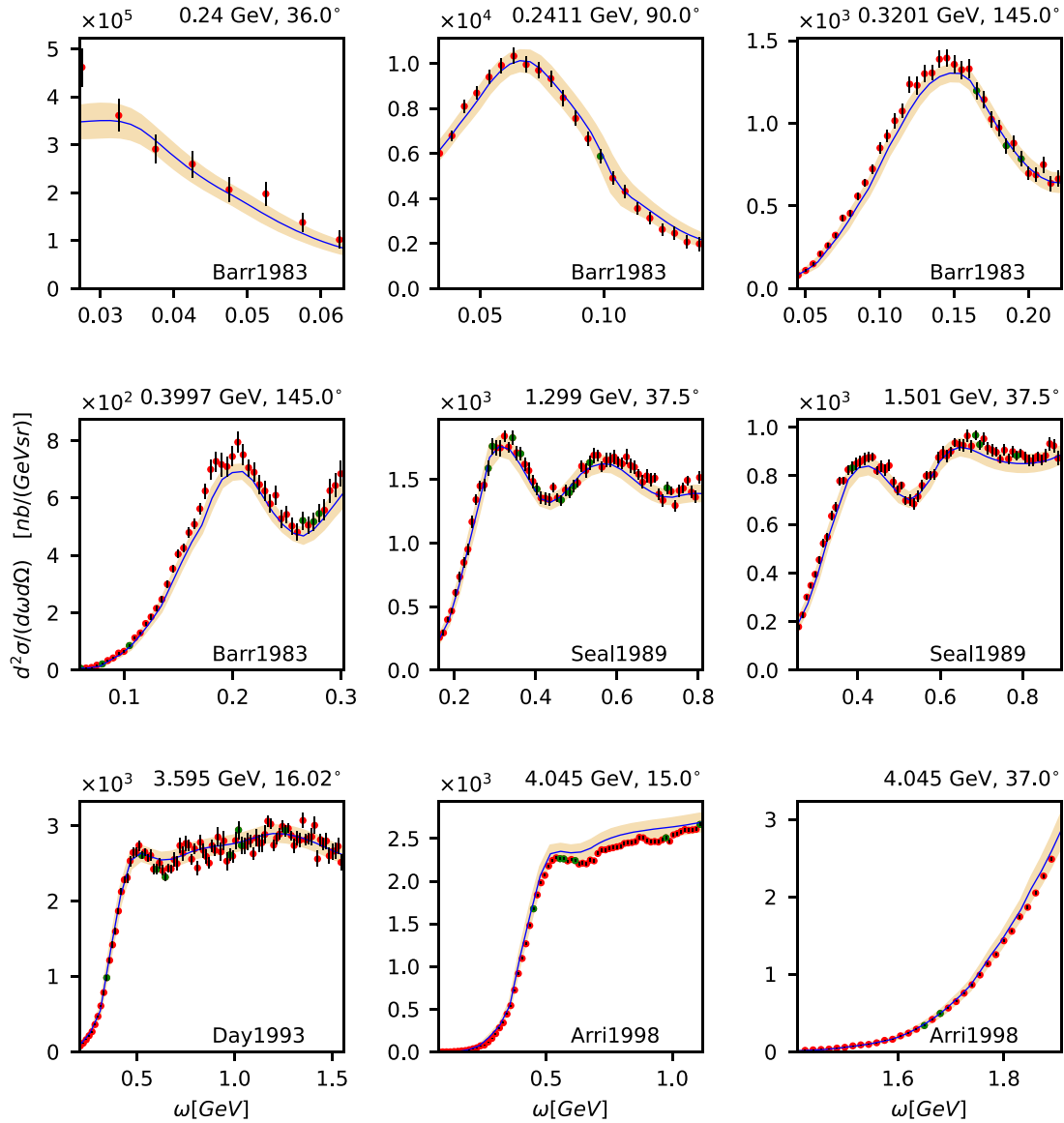


FIG. 7. Same as Fig. 6 but for model B.

So far, we have discussed the performance of the obtained fits in describing the available data. It is interesting to check how well they interpolate the cross section within the domain covered by the training data, and extrapolate beyond it.

To explore the models' abilities to interpolate, in Fig. 8, we compare their predictions with the spectral function calculations of Ref. [20], which include only quasielastic scattering. For these tests, we select the beam energy of 600 MeV—relevant for neutrino-oscillation experiments such as T2K and the Short Baseline Neutrino program—and the scattering angles of 30° , 45° , and 60° .

The amount of data collected at similar kinematic settings is the largest for $\theta = 60^\circ$, and the smallest for $\theta = 30^\circ$. Consequently, the predictions of the two models and their uncertainties are the most similar for $\theta = 60^\circ$. For lower scattering angles, the uncertainties apparently increase in model A but not in model B. At 45° ; one can clearly see a sudden jump

in the uncertainties of model A at $\omega \simeq 0.55$ GeV, reflecting the behavior of the underlying data [32].

The predicted positions of the quasielastic peak turn out to agree very well between the predictions of model A and the spectral function approach. This is also the case for model B when $\theta = 60^\circ$ and $\theta = 45^\circ$. However, when we lower the scattering angle, an increasing difference can be observed.

The magnitude of the cross section predicted by the fits at $\theta = 60^\circ$ seems consistent with the theoretical model, having in mind a large contribution of interaction mechanisms other than quasielastic scattering induced by one-body currents. At $\theta = 45^\circ$, the theoretical calculation agrees with model A (model B) at the 1σ ($2-3\sigma$) level. The differences increase at 30° , showing both the limitations of the obtained fits, and the urgent need for more experimental data to model electroweak interactions with the precision required by the neutrino-oscillation program.

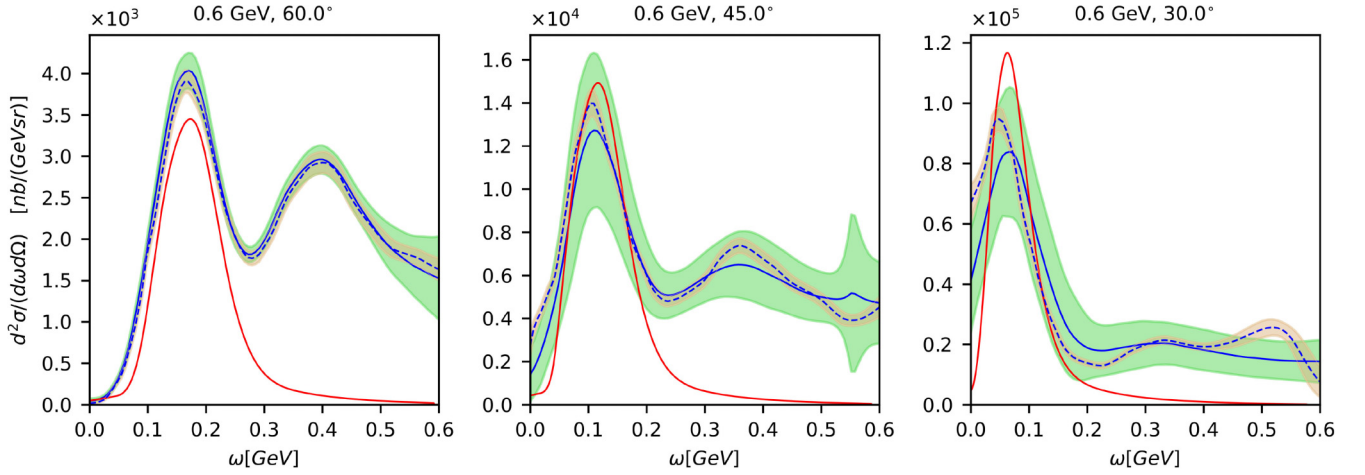


FIG. 8. Interpolation tests of the obtained fits for beam energy 0.6 GeV and scattering angles 30°, 45°, and 60°. We compare the spectral function calculations [20] of the quasielastic cross section, depicted by the solid red lines, with the predictions of model A (model B), represented by the solid (dashed) blue lines. The green (orange) areas correspond to the 1σ uncertainties.

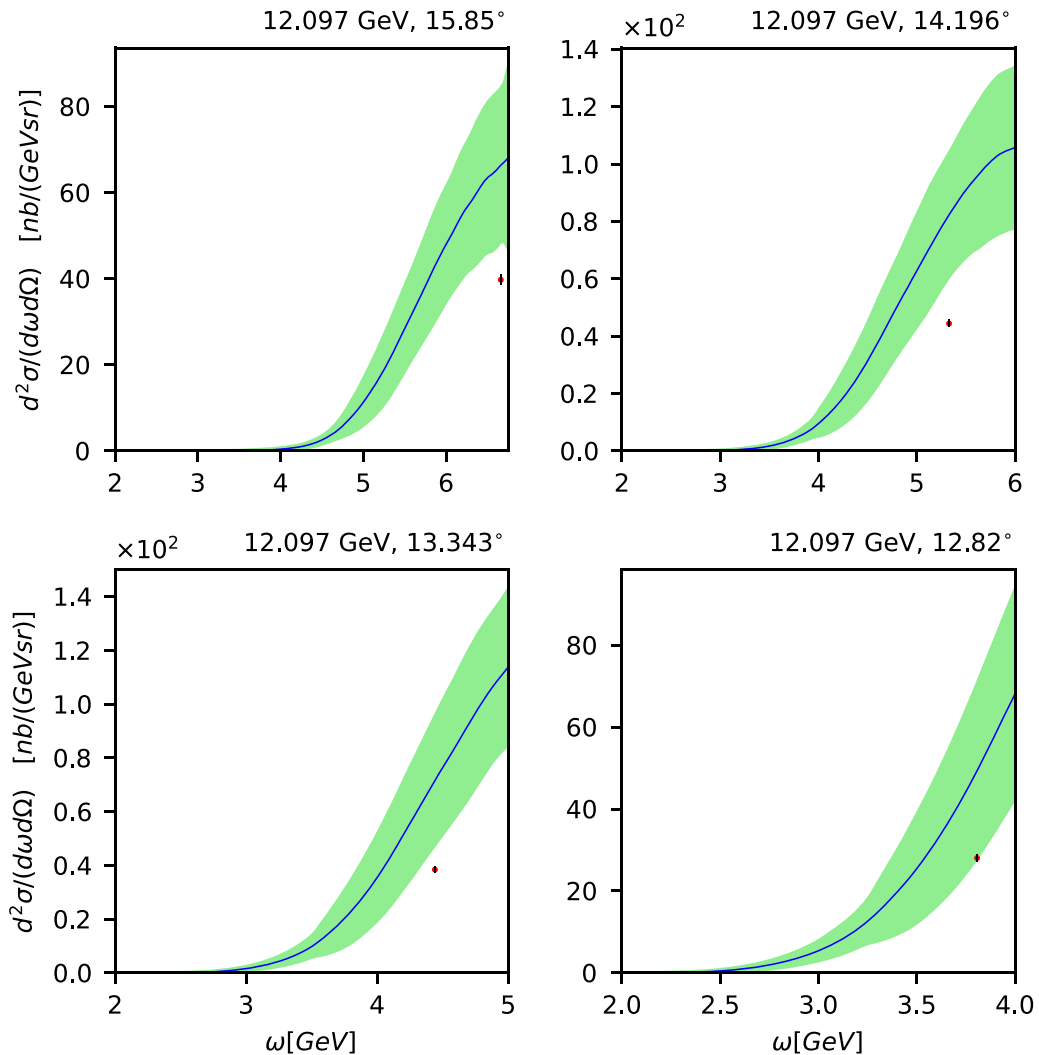


FIG. 9. Extrapolation tests of model A against the measurements by Gomez *et al.* [21] in the deep-inelastic regime, not included in the training. The highlighted areas represent the 1σ uncertainties.

To test the extrapolation properties of both models, we compare their predictions for the cross section in the deep-inelastic regime, at the kinematic settings of the measurements by Gomez *et al.* [21]. As shown in Fig. 9, the predictions of model A manage to reproduce these data at the 1σ – 2σ level. For model B (not shown in the picture), we find that it underestimates them.

In Fig. 9, the beam energy is fixed to 12.1 GeV, and scattering angles vary between 12.82° and 15.85° , with the corresponding momentum transfers $4.4 \leq |\mathbf{q}| \leq 7.0$ GeV. Only two datasets used in the training probe similar kinematics: the measurements performed by Fomin *et al.* [35] for beam energy 5.8 GeV, and scattering angles 40° and 50° , with $3.9 \leq |\mathbf{q}| \leq 5.3$ GeV.

We expect that the accuracy of extrapolation would improve, should more data be available for training.

In view of the obtained results, we conclude that model A is better at propagating the uncertainties of the underlying experimental data than model B.

V. SUMMARY

In this paper, we develop a deep neural network framework to parametrize the electron-scattering cross sections for carbon over a broad kinematic region, extending from the quasielastic peak, through resonance excitation, to the onset of deep-inelastic scattering. Our results do not depend on theoretical assumptions but rely exclusively on experimental

measurements. We pay special attention to estimating the uncertainties of the network's predictions, which reflect experimental uncertainties and are a measure of consistency between different datasets. The analysis of the impact of various types (epistemic and aleatoric) of uncertainties on predictive uncertainty is complex [46] and will be a focus of our future studies.

We discuss two statistical models. Model A is based on an ensemble of 50 neural networks, which fit bootstrap datasets. Model B uses a single neural network with dropout layers. Both approaches reproduce the training data accurately. Additionally, model A turns out to be able to estimate the cross sections outside the kinematic domain of the training data.

In addition to their importance for the studies of inclusive electron scattering on nuclei, our empirical fits can be readily included in the NUWRO Monte Carlo generator and pave the way toward the development of a more accurate description of electroweak cross sections. In future analyses, we plan to adapt and extend the methods developed in this article to the modeling of neutrino interactions.

To facilitate their usage, the resulting fits are released to the public through the GitHub repository [60].

ACKNOWLEDGMENT

This research is partly (K.M.G., A.M.A., J.T.S.) or fully (B.E.K., R.D.B., H.P.) supported by the National Science Centre under Grant No. UMO-2021/41/B/ST2/02778.

-
- [1] B. Abi *et al.* (DUNE Collaboration), *J. Instrum.* **15**, T08008 (2020).
 - [2] K. Abe *et al.* (Hyper-Kamiokande Proto-Collaboration), *Prog. Theor. Exp. Phys.* **2015**, 053C02 (2015).
 - [3] A. M. Ankowski *et al.*, *J. Phys. G* **50**, 120501 (2023).
 - [4] G. Cybenko, *Math. Control Signal Systems* **2**, 303 (1989).
 - [5] O. Al Hammal, M. Martini, J. Frontera-Pons, T. H. Nguyen, and R. Pérez-Ramos, *Phys. Rev. C* **107**, 065501 (2023).
 - [6] S. Forte, L. Garrido, J. I. Latorre, and A. Piccione, *J. High Energy Phys.* **05** (2002) 062.
 - [7] L. Del Debbio, S. Forte, J. I. Latorre, A. Piccione, and J. Rojo (NNPDF Collaboration), *J. High Energy Phys.* **03** (2005) 080.
 - [8] K. M. Graczyk, P. Plonski, and R. Sulej, *J. High Energy Phys.* **09** (2010) 053.
 - [9] K. M. Graczyk, *Phys. Rev. C* **84**, 034314 (2011).
 - [10] K. M. Graczyk and C. Juszczak, *Phys. Rev. C* **90**, 054334 (2014).
 - [11] K. M. Graczyk and C. Juszczak, *J. Phys. G* **42**, 034019 (2015).
 - [12] L. Alvarez-Ruso, K. M. Graczyk, and E. Saul-Sala, *Phys. Rev. C* **99**, 025204 (2019).
 - [13] K. Kumericki, D. Mueller, and A. Schafer, *J. High Energy Phys.* **07** (2011) 073.
 - [14] X.-X. Dong, R. An, J.-X. Lu, and L.-S. Geng, *Phys. Rev. C* **105**, 014308 (2022).
 - [15] A. M. Ankowski and A. Friedland, *Phys. Rev. D* **102**, 053001 (2020).
 - [16] R. Tibshirani, *Neural Comput.* **8**, 152 (1996).
 - [17] L. Breiman, *Mach. Learn.* **24**, 123 (1996).
 - [18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *J. Mach. Learn. Res.* **15**, 1929 (2014).
 - [19] S. Ioffe and C. Szegedy, in *ICML'15: Proceedings of the 32nd International Conference on Machine Learning* (Proceedings of Machine Learning Research, 2015), Vol. 37, pp. 448–456.
 - [20] A. M. Ankowski, O. Benhar, and M. Sakuda, *Phys. Rev. D* **91**, 033005 (2015).
 - [21] J. Gomez *et al.*, *Phys. Rev. D* **49**, 4348 (1994).
 - [22] U. Mosel and K. Gallmeister, *Phys. Rev. C* **99**, 064605 (2019).
 - [23] J. E. Amaro, M. B. Barbaro, J. A. Caballero, R. González-Jiménez, G. D. Megias, and I. Ruiz Simo, *J. Phys. G* **47**, 124001 (2020).
 - [24] M. E. Christy and P. E. Bosted, *Phys. Rev. C* **81**, 055213 (2010).
 - [25] P. E. Bosted and M. E. Christy, *Phys. Rev. C* **77**, 065206 (2008).
 - [26] A. Bodek and M. E. Christy, *Phys. Rev. C* **106**, L061305 (2022).
 - [27] A. Bodek and M. E. Christy, *Phys. Rev. C* **107**, 054309 (2023).
 - [28] J. Arrington *et al.*, *Phys. Rev. C* **53**, 2248 (1996).
 - [29] J. Arrington *et al.*, *Phys. Rev. Lett.* **82**, 2056 (1999).
 - [30] D. S. Bagdasaryan, M. K. Boyadzian, G. B. Kazarian, K. P. A. Kechian, E. R. Markarian, G. G. Mkrtrchian, O. P. Petrosian, I. A. Troshenkova, and V. O. Tatevosian (1988) (unpublished).
 - [31] D. Baran *et al.*, *Phys. Rev. Lett.* **61**, 400 (1988).
 - [32] P. Barreau *et al.*, *Nucl. Phys. A* **402**, 515 (1983).
 - [33] H. Dai *et al.*, *Phys. Rev. C* **98**, 014617 (2018).
 - [34] D. B. Day *et al.*, *Phys. Rev. C* **48**, 1849 (1993).
 - [35] N. Fomin *et al.*, *Phys. Rev. Lett.* **105**, 212502 (2010).
 - [36] J. S. O'Connell *et al.*, *Phys. Rev. C* **35**, 1063 (1987).
 - [37] R. M. Sealock *et al.*, *Phys. Rev. Lett.* **62**, 1350 (1989).

- [38] R. R. Whitney, I. Sick, J. R. Ficenec, R. D. Kephart, and W. P. Trower, *Phys. Rev. C* **9**, 2230 (1974).
- [39] O. Benhar, D. Day, and I. Sick, *Rev. Mod. Phys.* **80**, 189 (2008).
- [40] J. Arrington, *Phys. Rev. C* **68**, 034325 (2003).
- [41] W. M. Alberico, S. M. Bilenky, C. Giunti, and K. M. Graczyk, *Phys. Rev. C* **79**, 065204 (2009).
- [42] K. M. Graczyk, D. Kielczewska, P. Przewlocki, and J. T. Sobczyk, *Phys. Rev. D* **80**, 093001 (2009).
- [43] C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, 1995).
- [44] F. Chollet, *Deep Learning with Python* (Manning, Shelter Island, NY, 2017).
- [45] E. Hüllermeier and W. Waegeman, *Mach. Learn.* **110**, 457 (2021).
- [46] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, *Artif. Intell. Rev.* **56**, 1513 (2023).
- [47] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, *Infor. Fusion* **76**, 243 (2021).
- [48] L. Smith and Y. Gal, [arXiv:1803.08533](https://arxiv.org/abs/1803.08533).
- [49] B. Efron, *Ann. Stat.* **7**, 1 (1979).
- [50] B. Efron, *Ann. Appl. Stat.* **6**, 1971 (2012).
- [51] Y. Gal and Z. Ghahramani, [arXiv:1506.02142](https://arxiv.org/abs/1506.02142).
- [52] R. M. Neal, Bayesian learning for neural networks, Ph.D. thesis, University of Toronto, 1995.
- [53] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning (MIT Press, Cambridge, 2006), p. 248.
- [54] S. Geman, E. Bienenstock, and R. Doursat, *Neural Comput.* **4**, 1 (1992).
- [55] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, [arXiv:1611.03530](https://arxiv.org/abs/1611.03530).
- [56] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python+NumPy programs, GitHub, <https://github.com/google/jax> (2018).
- [57] F. Chollet *et al.*, Keras, <https://keras.io> (2015).
- [58] I. Loshchilov and F. Hutter, [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [59] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevC.110.025501> for full comparison of both models predictions and all experimental data in the analysis.
- [60] <https://github.com/bekowal/CarbonElectronNeuralNetwork>.