Empirical fits to inclusive electron-carbon scattering data obtained by deep-learning methods

Beata Kowal, Krzysztof Graczyk

Institute for Theoretical Physics
University of Wrocław
Poland

June 28, 2024

Motivations

NuWro is a Monte Carlo neutrino event generator, created at the University of Wroclaw.

https://github.com/NuWro

It simulates neutrino-nucleon and neutrino-nucleus reactions

- elastic and quasi-elastic scattering
- single pion production through $\Delta(1232)$ resonance including non-resonant background
- deep inelastic scattering
- nuclear effects, two-body current contribution to (quasi-) elastic scattering coherent pion production

Motivations

With the help of Neural Network techniques, it is planned to accelerate and optimize the NuWro generator.

There are some similarities between electron and neutrino scattering.

The first step towards this goal is to create a neural network describing electron-nucleon scattering.

Based on:

arxiv:2312.17298
B. Kowal, K.Graczyk,
A. Ankowski, R. Banerjee, H. Prasad and J. Sobczyk (NuWro team)

The goal

- Making the model independent way to predict inclusive electron-carbon scattering cross-sections (based on the experimental measurements):

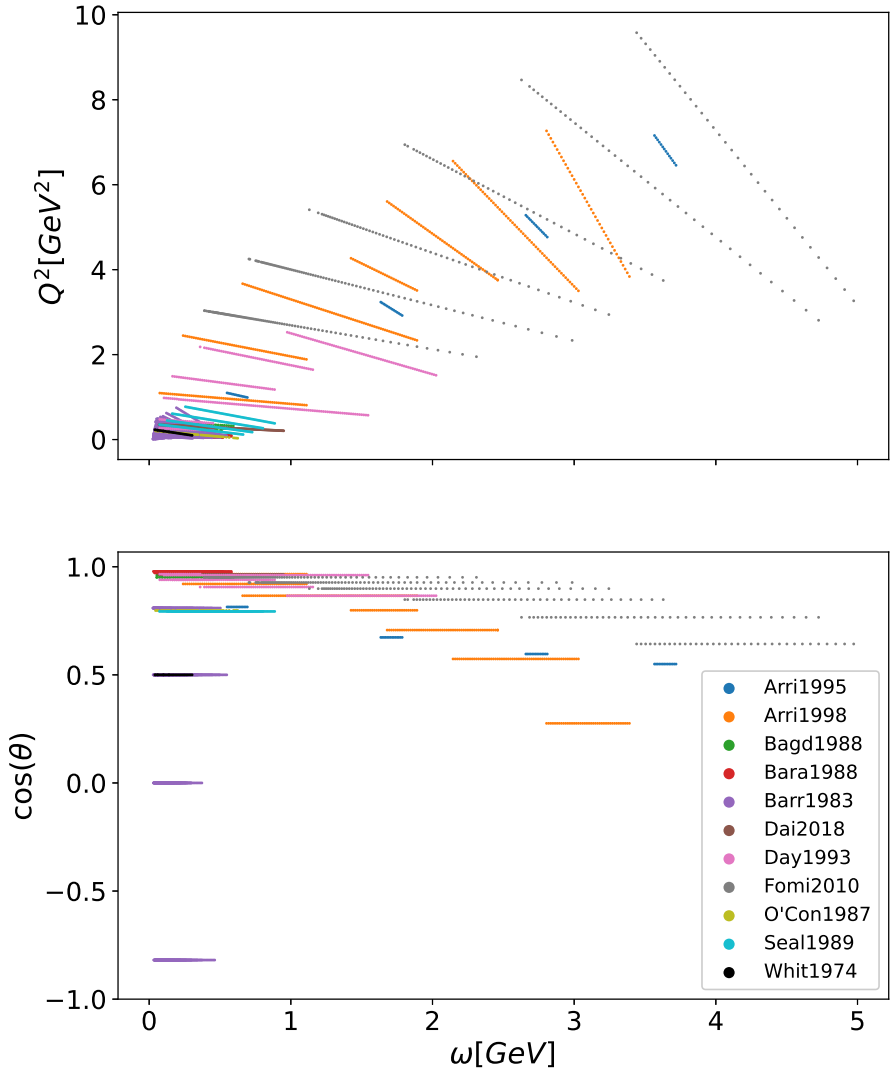
$$DNN(E,\theta,\omega) \rightarrow \frac{d^2\sigma}{d\cos(\theta)d\omega} \tag{1}$$

DNN - Deep Neural Network prediction,
E = Energy, θ = scattering angle, ω =transfer of energy

- Development of techniques that allow us estimate the uncertainty of the predictions of DNN.

Data

- Data from the Electron Nucleus Scattering Archive: `http://discovery.phys.virginia.edu/research/groups/qes-archive/notes.html`
- we concentrate on electron-carbon data
- a broad kinematic region: **quasielastic scattering**, **pion production**, and the onset of **deep-inelastic scattering**
- **We remove data with the lowest ω by applying an appropriate cut**, because they correspond to elastic scattering and discrete nuclear state



Data

- 11 independent datasets.
- the k-th dataset containing N_k points

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

  where $d\sigma_k^i$ and $\Delta d\sigma_k^i$ are the i-th measurement in k-th dataset and corresponding uncertainty.
- $\Delta d\sigma_k^i$ includes statistical and systematic uncertainties.
- The normalization λ, systematic uncertainty, is taken into consideration.

Abbrev.	Number of points
Arri1995	56
Arri1998	398
Bagd1988	125
Bara1988	259
Barr1983	1243
Dai2018	177
Day1993	316
Fomi2010	359
O'Con1987	51
Seal1989	250
Whit1974	31
Total	3265

Potential problems

- Cross-section values in ineffective, for DNN, range domain

$\rightarrow$ re-scale the cross-sections (rescaled data < 1)

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

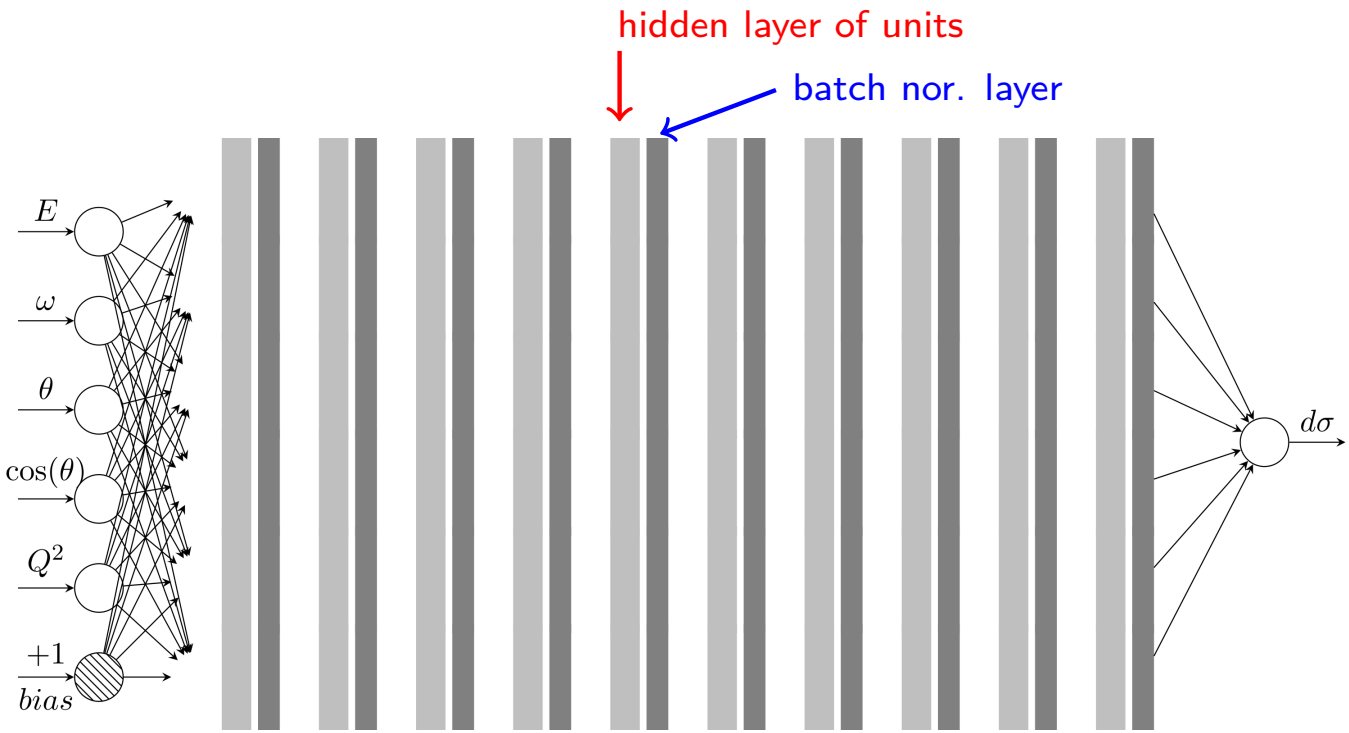
- Additional imput parameters, that depend on the others, to achieve fitting :

$$(E, \omega, \theta) \rightarrow (E, \omega, \theta, \cos(\theta), Q^2)$$

- DNN may over-fit the data
- How to estimate the uncertainty of the predictions?

DNN: Model A (Bootstrap model)

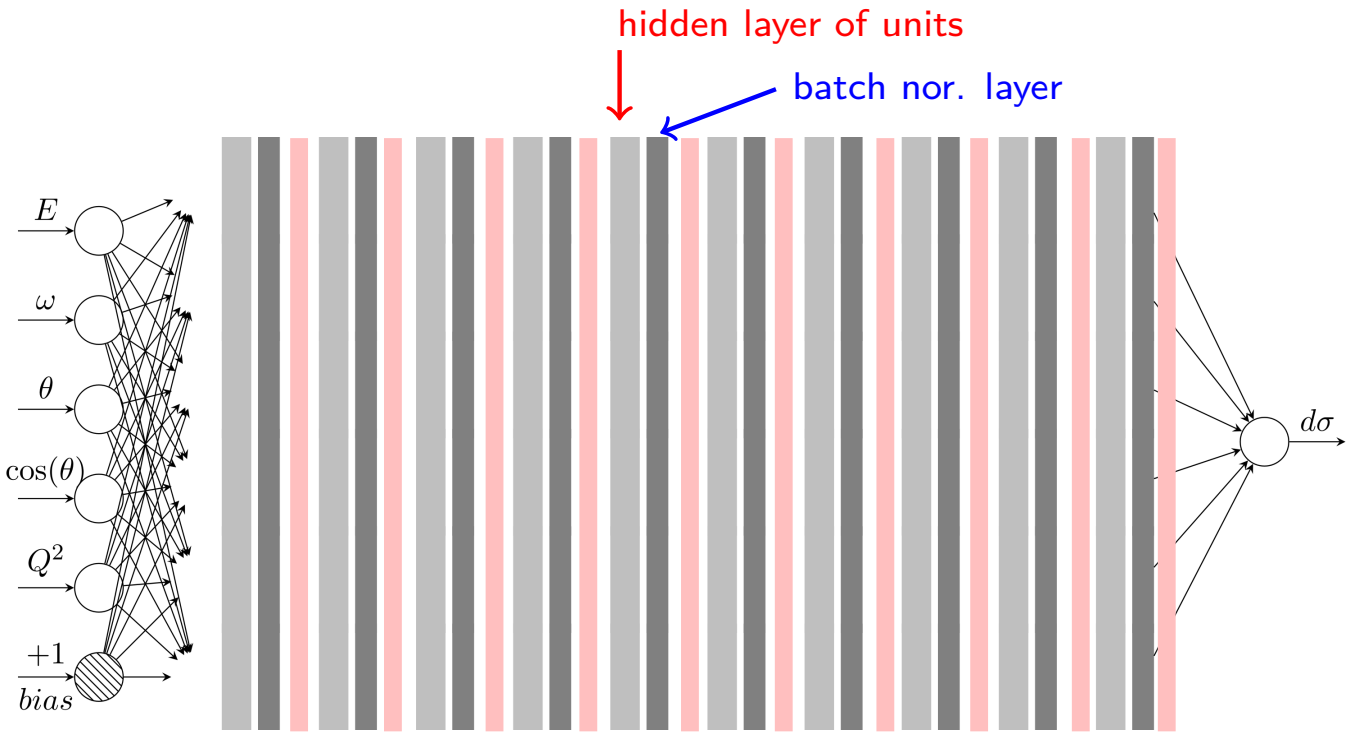
- 10 blocks with 300 fully connected neurons and following batch normalization layer



- Batch Normalization (Ioffe and Szegedy, arxiv:1502.03167): re-centering and re-scaling inputs of the leyers, improves optimization and regularize the model

DNN: Model B (MC Dropout model)

- 10 blocks with 300 fully connected neurons and following batch normalization layer and dropout leyer



- **Dropout layer**: In every layer, some hidden units are dropped from the processing the signal (forward and backward), with a probability $\mathbf{p}$ [Hinton, et al., arXiv:1207.0580.,]
- improves generalization
- prevents overfitting

Model A: Bootstrap approach

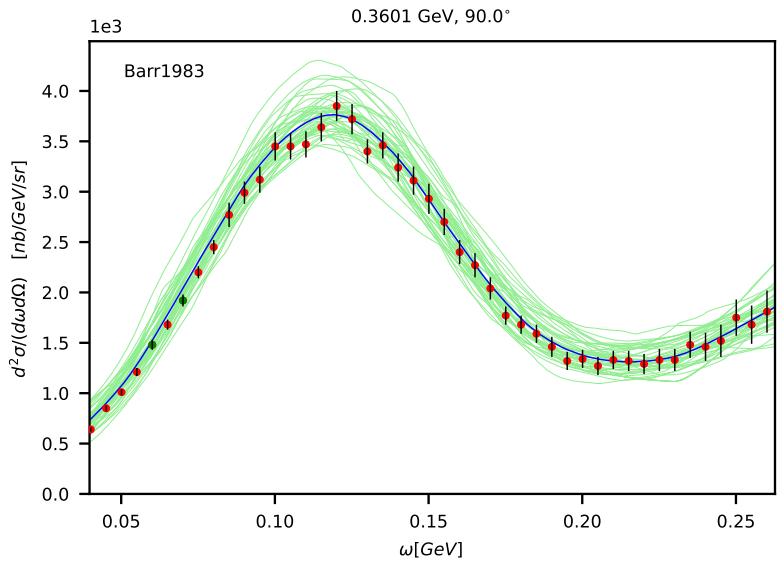
- Efron (1979): Bootstrap methods
- Adapted for neural networks by Tibshirani (1996) and Breiman (1996).

i For each data sample, we have a Gaussian distribution with
- mean = data point σ_k^i
- std. deviation = uncertainty $\Delta\sigma_k^i$

ii we draw $M = 50$ bootstrap (clone) datasets from Gaussian distributions

iii For each bootstrap (clone) data set, obtain DNN fit.

iv Averaging over the models prevents overfitting. Get the mean and standard deviation



Model B: MC Dropout approach

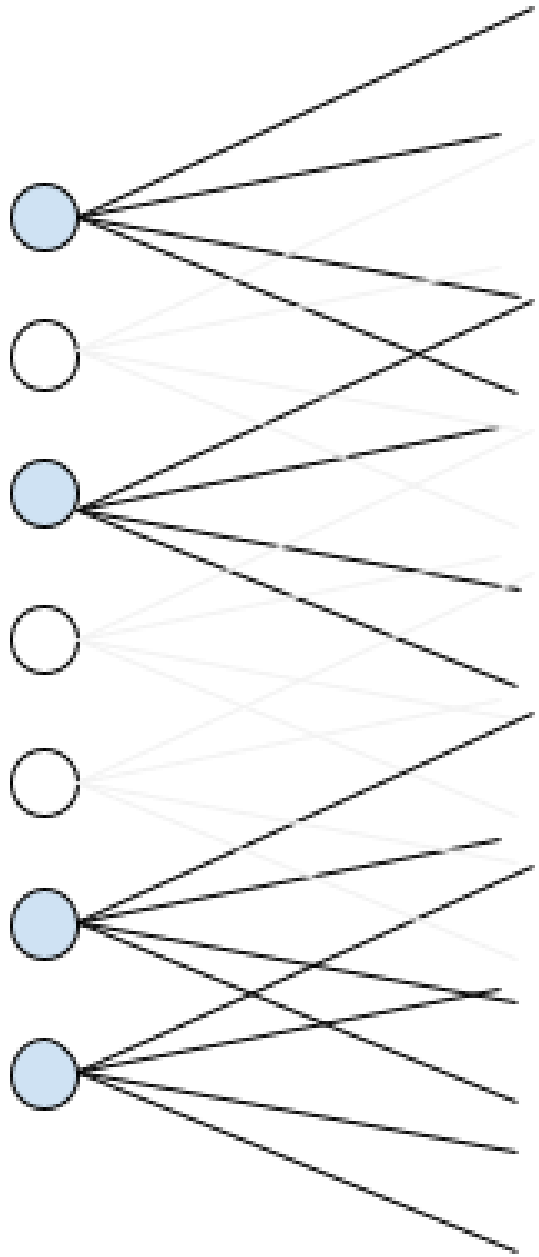
- We keep dropout layers active in training mode
- We obtain one DNN fit

$\rightarrow$ To make prediction:

- We keep dropout layers active in inference mode

i compute $M = 50$ times the response of the network for a given input.

ii average over the predictions, get the mean and standard deviation



Systematic Normalization

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

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

Abbrev.	$\Delta\lambda_k$
Arri1995	4.0%
Arri1998	4.0%
Bagd1988	10.0%
Bara1988	3.7%
Barr1983	2.0%
Dai2018	2.2%
Day1993	3.4%
Fomi2010	4.0%
O'Con1987	5.0%
Seal1989	2.5%
Whit1974	3.0%

- λ_k's are hyperparameters

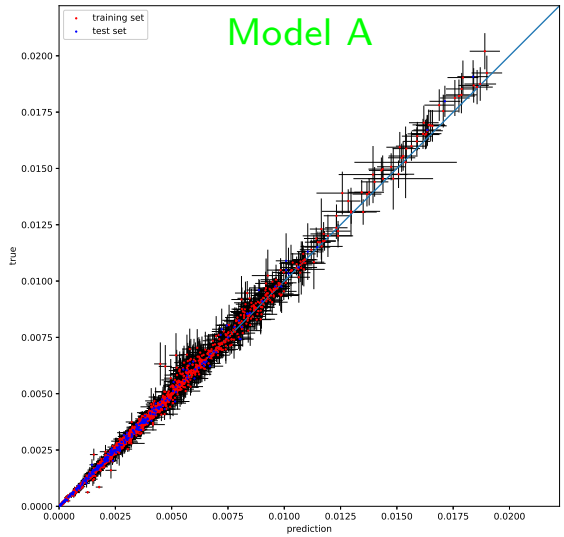
Normalization and data consistency

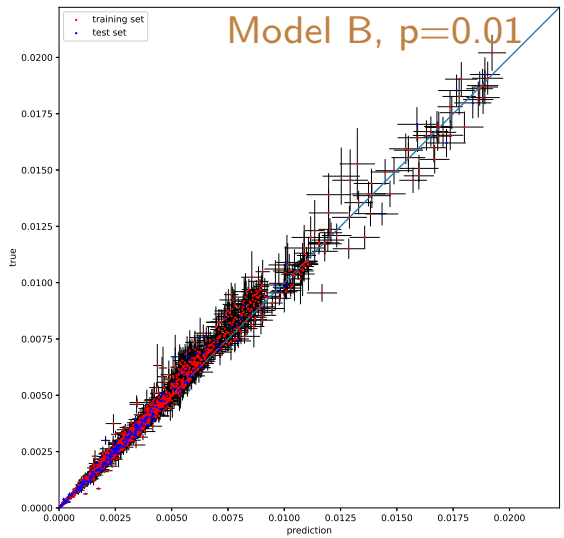
Abbrev.	Norm. uncert.	model A λ_k	model B $\lambda_k(p = 0.01)$
Arri1995	4.0%	1.01	1.02
Arri1998	4.0%	1.00	0.96
Bagd1988	10.0%	1.03	1.06
Bara1988	3.7%	1.01	0.98
Barr1983	2.0%	0.99	1.02
Dai2018	2.2%	1.00	0.97
Day1993	3.4%	0.99	0.98
Fomi2010	4.0%	1.01	0.96
O'Con1987	5.0%	1.02	1.01
Seal1989	2.5%	1.02	1.04
Whit1974	3.0%	0.93	0.93

- A tension between Whit1974 and the rest of datasets?

Numerical Analysis

- Jax package
- HDF5 to store and organize data
- AdamW algorithm with decay width 0.004
- Minibatch configuration with five batches
- We split the dataset into training and test datasets, with a proportion of 9:1



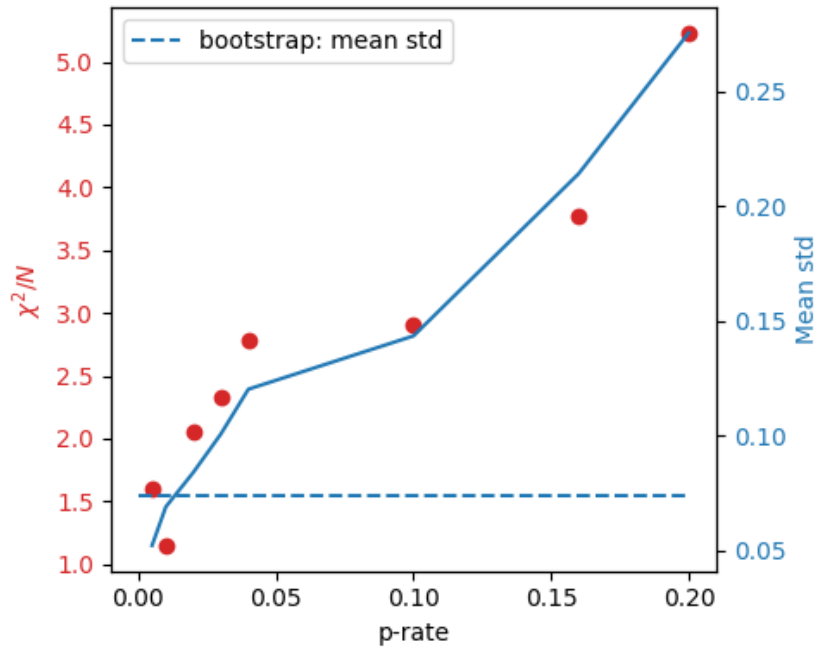


Calibration of MC Dropout

* we may expect similar results between bootstrap and Bayesian approaches

  Efron, Bayesian inference and the parametric bootstrap, (2012)

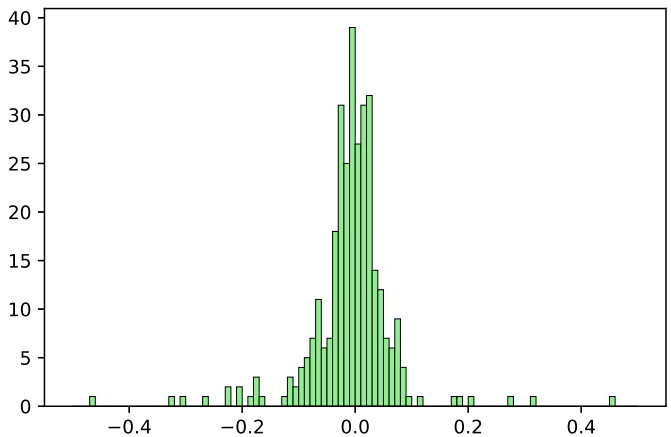
- Run model A (dropout) for several p values
- Compute $\chi^2(test)$
- After calibration, we choose Dropout Model with $p = 0.01$

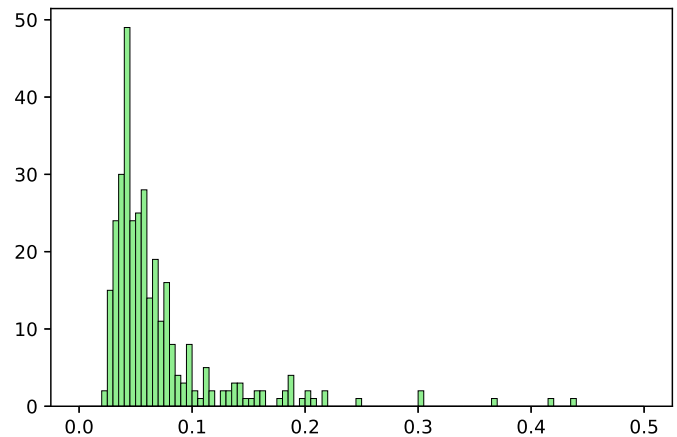


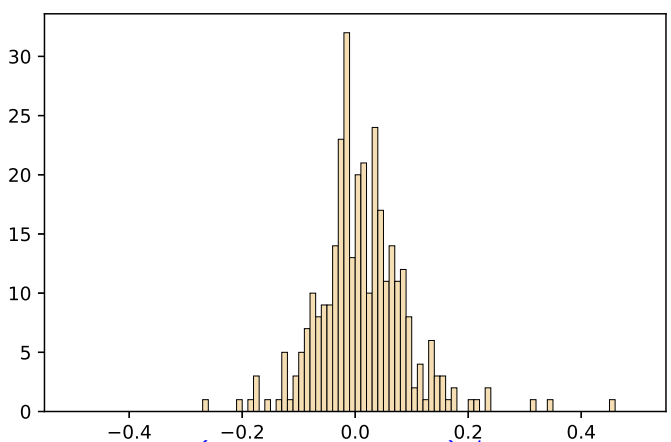
- Mean[uncertainty/true value](test data)

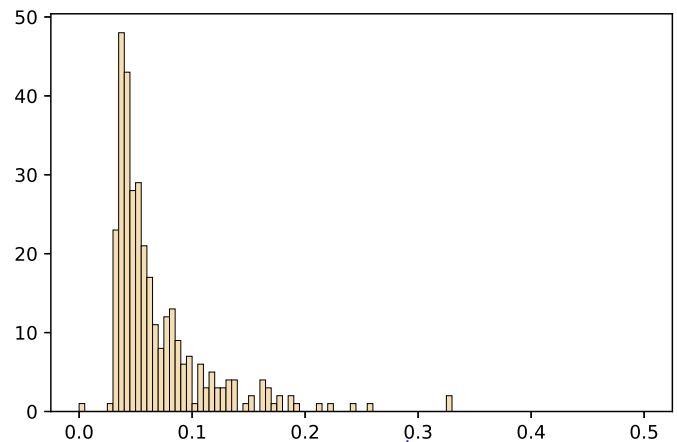
Histograms: Model A, Bootstrap (top) and Model B, MC Dropout (bottom)

On the test data set, dropout p=0.01

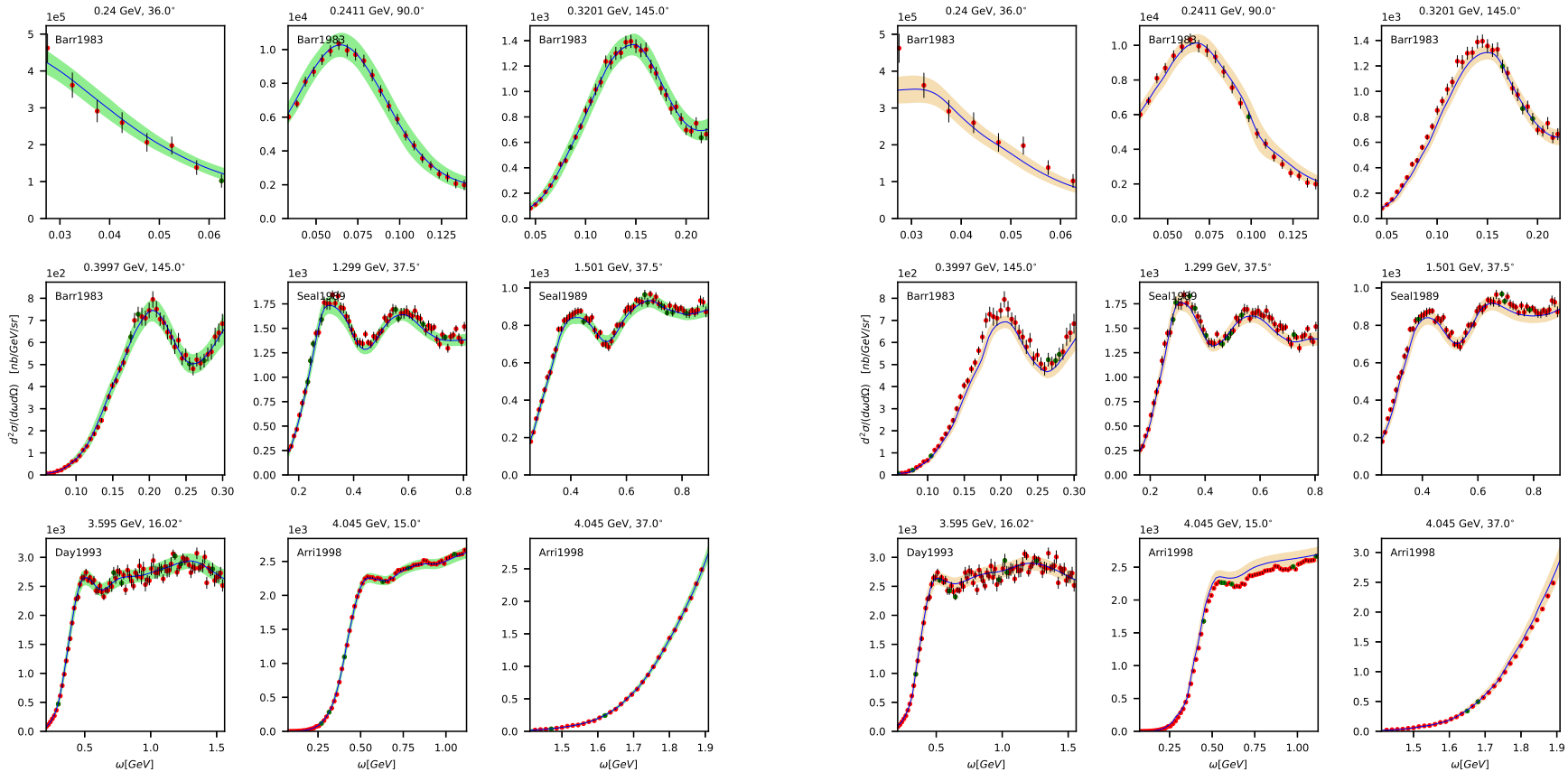






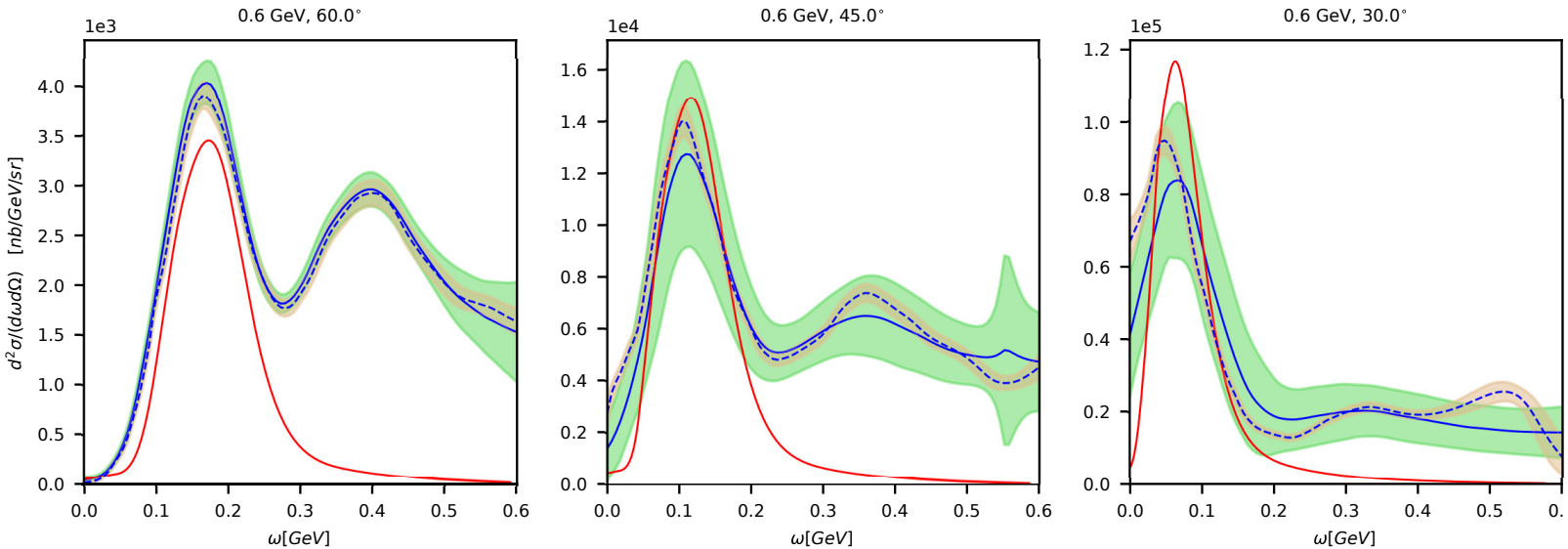


Results: Model A (bootstrap) and Model B (dropout)



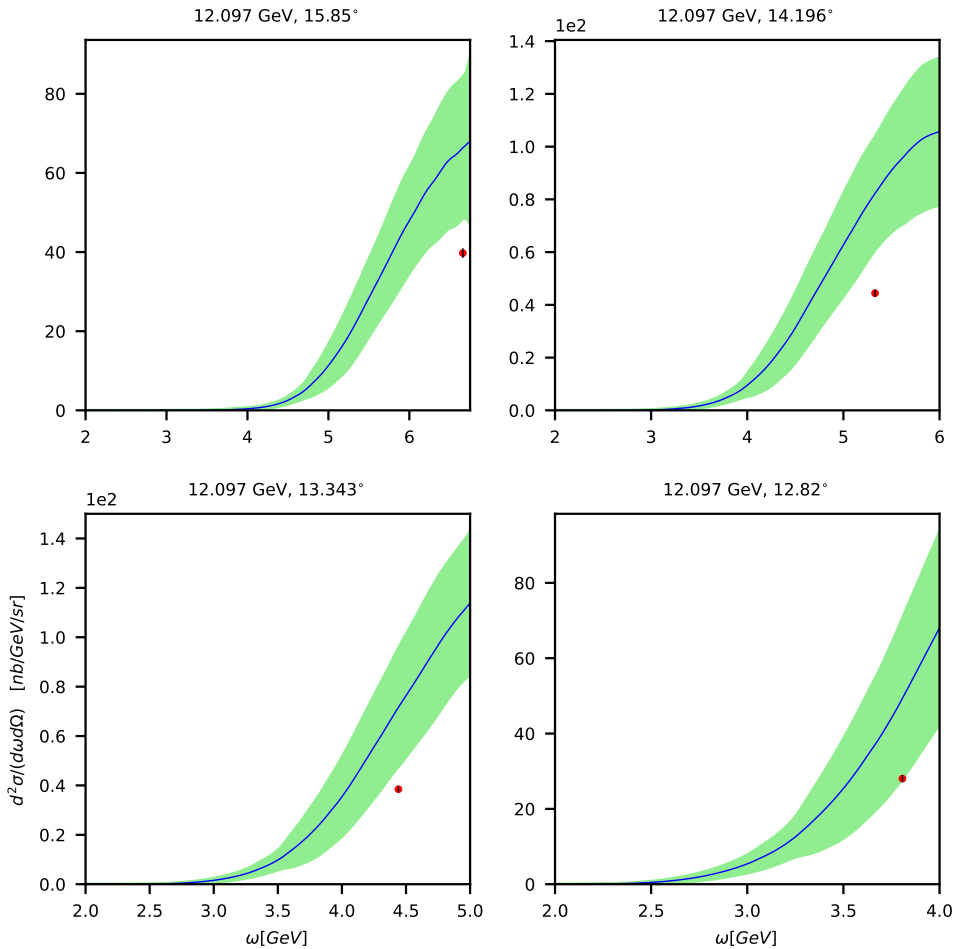
- training and test points

DNN vs. Spectral function



- Model A Bootstrap and Model B MC Dropout (p=0.01)
- Spectral function QE scattering, Ankowski, Benhar, Sakuda, PRD 91, (2015) 03300
- Energy of 600 MeV relevant for neutrino-oscillation experiments such as T2K and the Short Baseline Neutrino program
- Bootstrap model generalizes better

Model A Bootstrap vs. Gomez *et al.*



- data: Gomez *et al.*, PRD 49, (1994) 4348. (deep inelastic scattering data)

Summary

- we obtained fits to electron-carbon data using to methods: dropout and bootstrap
- we compared their predictions to
  - a test dataset,
  - a dataset lying beyond the covered kinematic region,
  - a theoretical predictions obtained within the spectral function approach
- Models reproduce the data well but Model A generalizes better than Model B
- Both methods take into account a statistical and systematic uncertainties

* available from `https://github.com/bekowal/CarbonElectronNeuralNetwork`