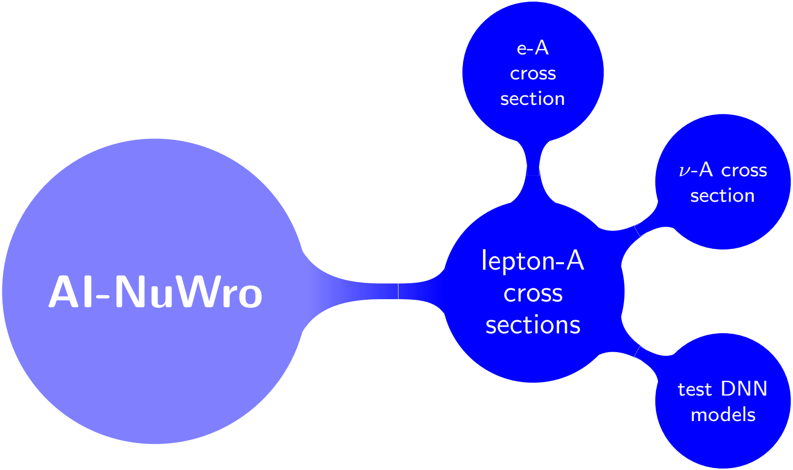
Inclusive electron-carbon scattering cross sections from deep-learning analysis

Krzysztof M. Graczyk

Institute for Theoretical Physics
University of Wrocław
Poland

April 15, 2024

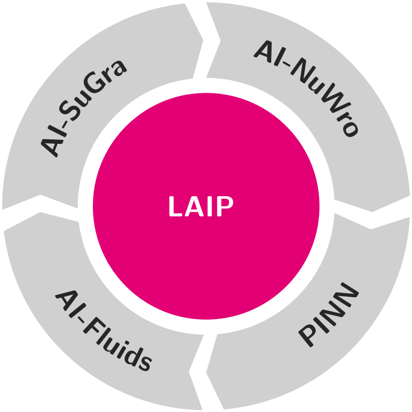


- inclusive electron-nucleus scattering cross sections from deep-learning analysis

based: Beata Kowal, Graczyk, Ankowski, Banerjee, Prasad, and Sobczyk, arxiv:2312.17298

$\rightarrow$ case study of the DNN techniques

Laboratory of AI for Physics (LAIP)



https://kgraczyk.github.io/laip/

- AI-NuWro
  - with Jan, Artur, Beata, Luis, Rwik, Hemant: financed by National Science Centre, Poland
- PINN - Physics Informed Neural Network
  - solving PDFs, Bayesian approach, with Juszczak and Witkowski
- AI-Fluids:
  - mostly fluid flow in porous media, with local CFD group lead by Maciek Matyka
- AI-SuGra: Searches for algebra structures for SuperGravity
  - with Remik Durka

PAST Bayesian Neural Networks: elastic ep scattering, TPE, E-M FFs, Proton Radius, Axial FF (from 2009 to 2018).

Today's Goal

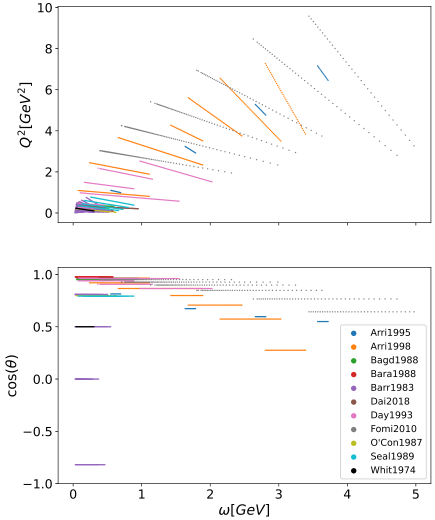
- Model independent way of predictive inclusive electron-carbon cross-sections.
- * Based on the experimental measurements only:

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

  Deep Neural Network (DNN), E = Energy, θ = scattering angle, ω =transfer of energy
- Development techniques that allow us to assess how uncertain are the predictions of DNN.
- Similar work by Al Hammal *et al.*, PRC 107, 065501 (2023)

Data

- Data from `http://discovery.phys.virginia.edu/research/groups/qes-archive/notes.html`
- we concentrate on electron-carbon data (the most informative)
- a broad kinematic region: **quasielastic scattering**, **pion production**, and the onset of **deep-inelastic scattering**
- At the lowest ω, elastic scattering, and inelastic interactions (with an excitation of the giant dipole resonance or a discrete nuclear state)
  $\rightarrow$ the scarcity of data in this region
  $\rightarrow$ **we remove their contributions by applying an appropriate cut**.



Data

- 11 independent datasets.
- the kth dataset containing N_k points

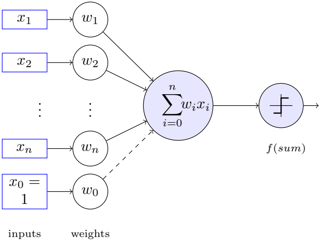
  $$\mathcal{D}_k = \{(E_k^i, \theta_k^i, \omega_k^i, d\sigma_k^i, \Delta d\sigma_k^i) : \ i = 1, \ldots, N_k\},$$

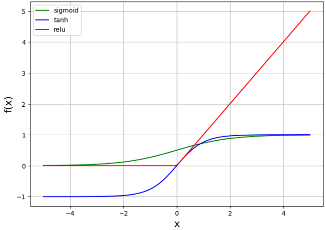
  where $d\sigma_k^i$ and $\Delta d\sigma_k^i$ are the i-th measurement in k-th dataset and corresponding uncertainty.
- $\Delta d\sigma_k^i$ is symmetric: includes statistical and point-to-point systematic uncertainties.
- The normalization, systematic uncertainty, is taken into consideration.

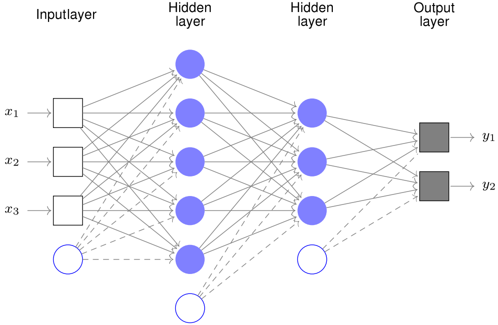
Abbrev.	Number of points
Arri1995	56
Arri1998	398
Bagd1988	125
Bara1988	259
Barr1983	1243
Dai2018	177
Day1993	316
Fomi2010	359
O'Con1987	51
Seal1989	250
Whit1974	31
Total	3265

Basics of Neural Networks

Neuron (Perceptron)



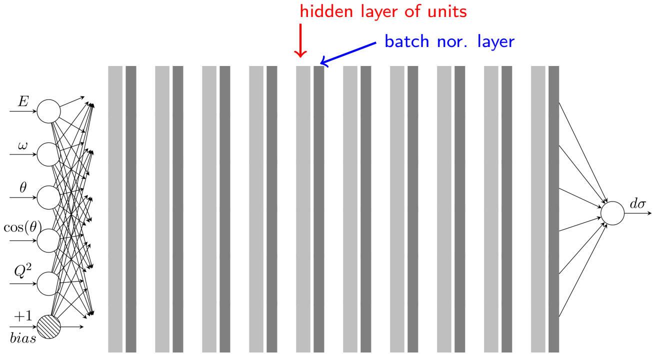




- Shallow neural network: one, two, hidden layers,
- Deep Neural Networks: representation learning!?

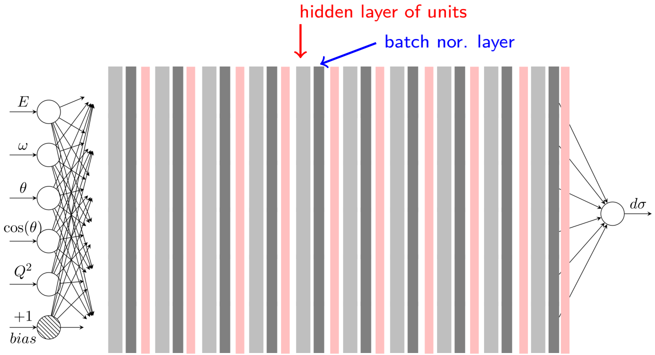
$\rightarrow$ usually many layers neural networks

DNN: Model A



- 10 blocks, each consists of 300 fully connected units and following batch normalization layer
- Batch Normalization (Ioffe and Szegedy, arxiv:1502.03167): solve (partially) vanishing gradient problem, improve optimization, regularize "naturally" the model

DNN: Model B



- 10 blocks, each consists of 300 fully connected units and following batch normalization layer
- Dropout layer: In every layer, hidden units are dropped from the processing the signal (forward and backward), with probability **p** [Hinton, et al., arXiv:1207.0580.,]
- lowers the error on the data test, so it improves generalization
- prevents overfitting

Potential problems

- Data in ineffective, for DNN, range domain

→ re-scale the cross-sections

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

- to improve:

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

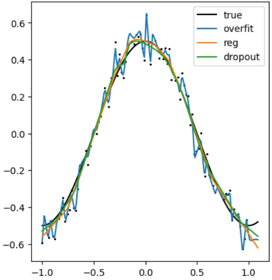
- DNN may over-fit the data
- How to get a model with good predictive ability to generalize well?

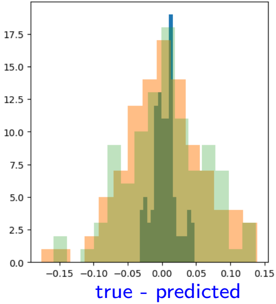
* Open problem in DL, see Zhang, *et al.*, arXiv:1611.03530, *Understanding deep learning requires rethinking generalization*

- How uncertain are the predictions?

* Open problem in DL, see Gawlikowski *et al.*, *A Survey of Uncertainty in Deep Neural Networks*, arXiv:2107.03342

generalization: bias-variance trade-off





Yes Shrinking DNN weights

$$Loss \rightarrow Loss + \frac{\alpha}{2}\sum_i w_i^2$$

Yes Batch normalization and training in mini-batch configuration (5 batches)

Yes Dropout (model B)

Yes Data augmentation (model A)

Yes Check the model performance on test data

No Cross-validation techniques: did not work effectively, lack of data

$\rightarrow$ we prefer to have more data (in wide kinematical range) in the training dataset

Uncertainties in DNN predictions

- DNN: models with a large number of parameters
- conventional methods might not work

→ not designed for DNNs

→ numerically inefficient

We follow

- Bagging or bootstrap approach (model A)
  - → Ensemble methods
- MC Dropout (model B)

→ Variational inference

- Bayesian methods

see Gawlikowski *et al.*, arXiv:2107.03342

Model A: bootstrap aggregation or bagging

- Efron (1979): bootstrap parametric and non-parametric
- Adapted for neural networks by Tibshirani (1996) and Breiman (1996).

→ We consider parametric-like:

i For each data sample, we have a Gaussian distribution with mean σ_k^i and variance $\Delta\sigma_k^i$

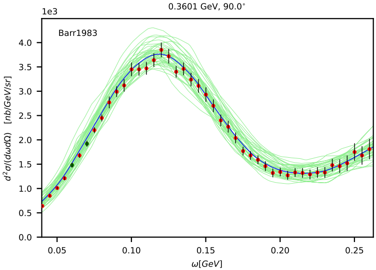
ii Collect $M = 50$ bootstrap (clone) datasets (Tibshirani: M from 25 to 200)

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

iv Average over the ensemble of models

* Augmentation-like technique

** Averaging over the models prevents overfitting



Model B: MC Dropout

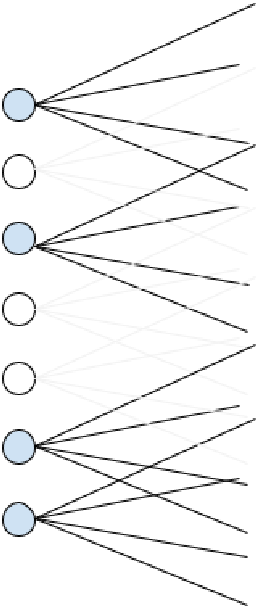
Gal and Ghahramani, arXiv:1506.02142 $\rightarrow$ *approximate Bayesian inference in deep Gaussian processes*

- Keep dropout layers active in training and inference modes!

$\rightarrow$ To make prediction:

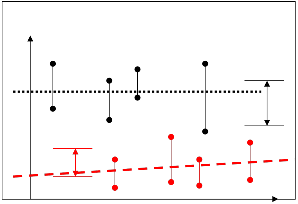
i compute $M = 50$ times the response of the network for a given input.

ii average over the predictions, get the mean and variance



Likelihood and Systematic Normalization

$$\chi_{\text{tot}} = \sum_{k=1}^{11} \left[\chi_k^2(\lambda_k) + \frac{1}{2} \left(\frac{1-\lambda_k}{\Delta\lambda_k} \right)^2 \right], \quad \chi_k^2(\lambda_k) = \frac{1}{2} \sum_{i=1}^{N_k} \left(\frac{d\sigma_k^i - \lambda_k d\sigma_i^{\text{fit}}(E_k^i, \theta_k^i)}{\Delta d\sigma_k^i} \right)^2$$



see D'Agostini, NIMPR A 346 (1994) 306

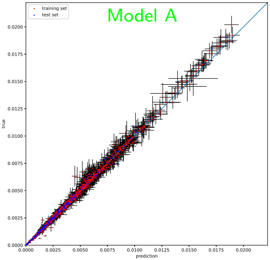
- elastic ep scattering, see e.g. PRC79 (2009) 065204
- C_5^A-axial form factor and consistency of ANL and BNL data: PRD80 (2009) 093001
- DNN tends to lose proper normalization, Graczyk *et al.* Self-Normalized Density Map (SNDM) for Counting Microbiological Obejcts, Sci Rep 12, 10583 (2022)

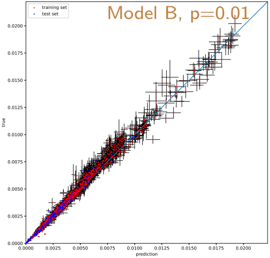
Abbrev.	$\Delta\lambda_k$
Arri1995	4.0%
Arri1998	4.0%
Bagd1988	10.0%
Bara1988	3.7%
Barr1983	2.0%
Dai2018	2.2%
Day1993	3.4%
Fomi2010	4.0%
O'Con1987	5.0%
Seal1989	2.5%
Whit1974	3.0%

- λ_k's are hyperparameters

Numerical Analysis

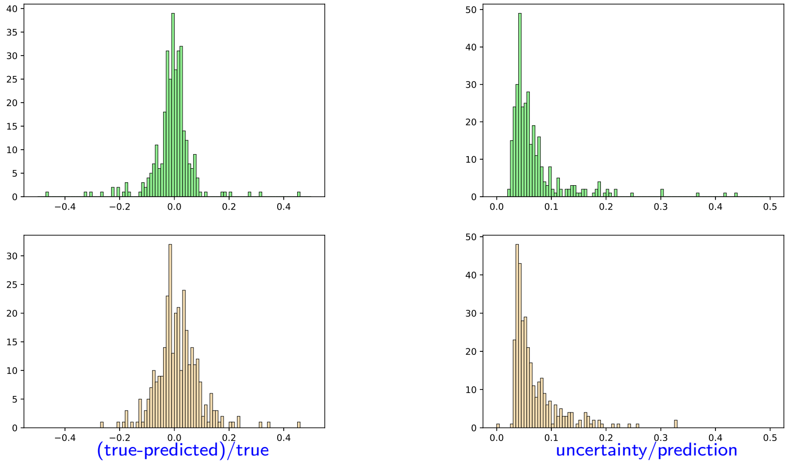
- Jax package (in pre-analysis also Keras@TensorFlow)
- AdamW algorithm with decay width 0.004
- Minibatch configuration with five batches
- We split the dataset into training and test datasets, with a proportion of 9:1.
- Run MC dropout for several p values





Histograms: Model A, bagging (top) and Model B, MC dropout (bottom)

On the test data set, dropout p=0.01

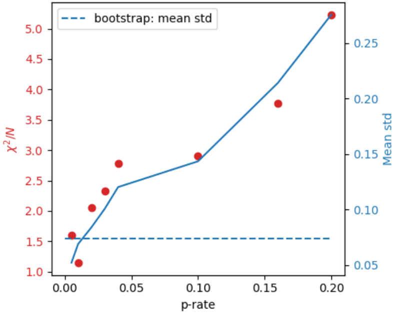


Calibration of MC dropout

Standard Consider MC dropout for various p value and compute $\chi^2(test)$!

- Bootstrapping leads to the ***"poor man's Bayes posterior"***.
- we may expect similar results between bootstrap and Bayesian approaches

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



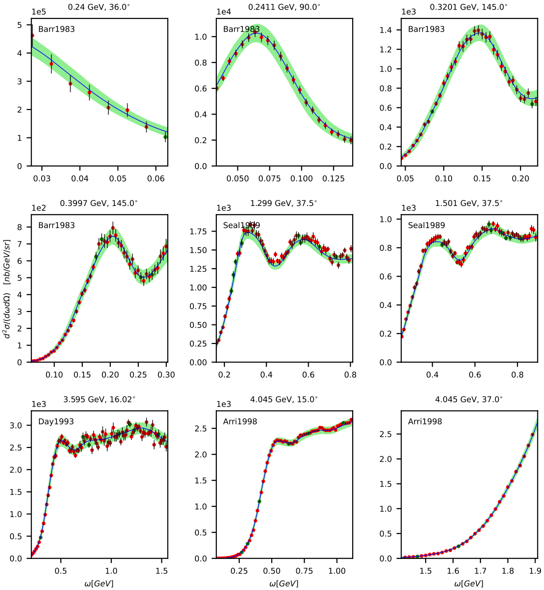
- Mean[uncertainty/prediction](test data)
- After calibration, we choose MC dropout with $p = 0.01$

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

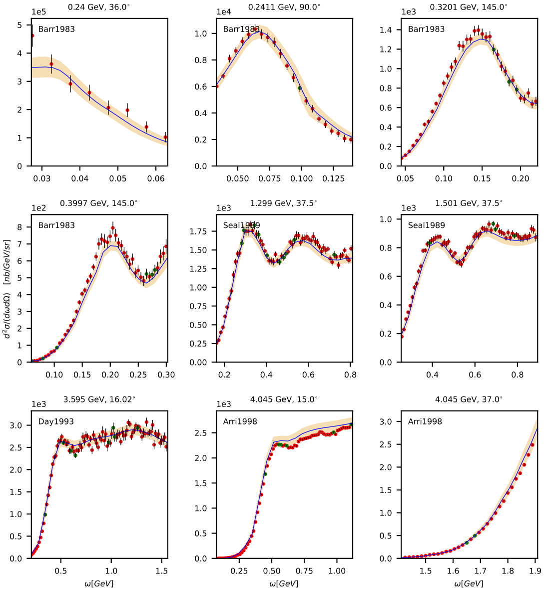
- A tension between Whit1974 and the rest of datasets?

Results: Model A (bootstrap)



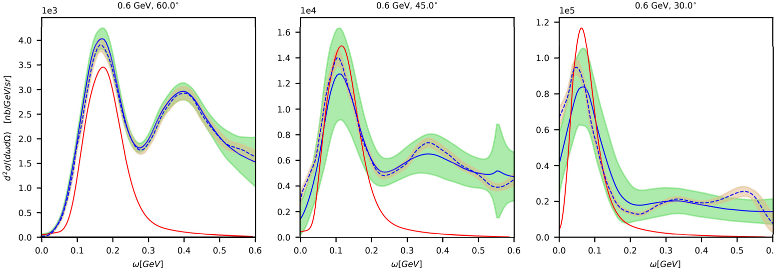
- training and test points

Results: Model B (MC dropout)



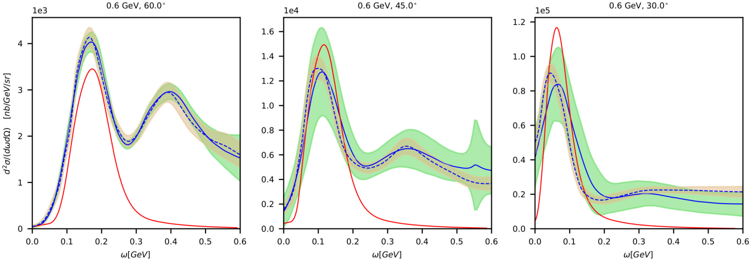
- training and test points

DNN vs. Spectral function



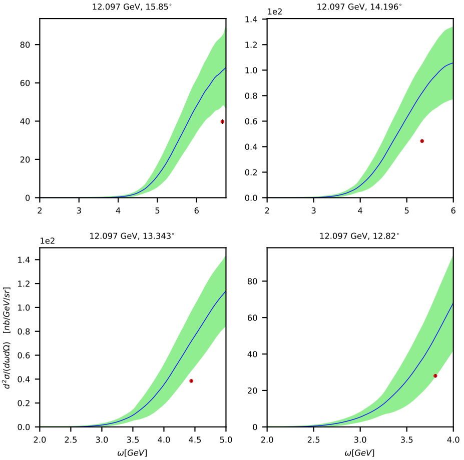
- Model A and model B (p=0.01)
- Spectral function QE scattering, Ankowski, Benhar, Sakuda, PRD 91, (2015) 03300
- Energy of 600 MeV relevant for neutrino-oscillation experiments such as T2K and the Short Baseline Neutrino program

DNN vs. Spectral function



- Model A and model B **(p=0.04)**
- 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

Model A vs. Gomez *et al.*



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

Summary

- Models reproduce the data well but Model A generalizes better than Model B
- Both methods take into account aleatoric (data) and epistemic (model) uncertainties
- When new data arrives then model can be easily tuned!

→ **Longitudinal and Transverse** components and consider other target

→ DNN model of νA cross sections

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

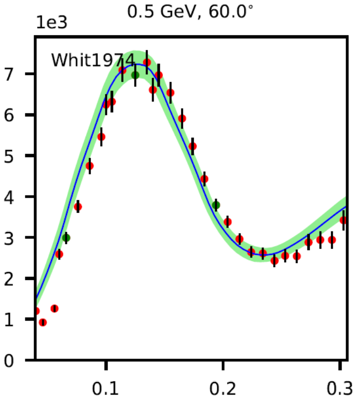
AI for NuWro

- Starting from data and objective DL tools (not dedicated particularly to the problem) → cross section model → (the first small step ...)

→ An example of **Physics guided Neural Network (PgNN)** approach

- The theoretical input can be included too: towards **Physics encoded Neural Network (PeNN)**

* for review of PgNN, PiNN, PeNN see, *Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks in Scientific Computing*, arXiv:2211.07377


0.5 GeV, 60.0°
1e3
Whit1974
7
6
5
4
3
2
1
0
0.1
0.2
0.3