



HAL
open science

Using Neural Networks for Inelastic Neutron Scattering Cross Sections Interpretation

Greg Henning, M. Kerveno, Philippe Dessagne, Marie Vanstalle, Levana
Gesson, Lucie Petit, Thomas Raymond

► **To cite this version:**

Greg Henning, M. Kerveno, Philippe Dessagne, Marie Vanstalle, Levana Gesson, et al.. Using Neural Networks for Inelastic Neutron Scattering Cross Sections Interpretation. 2024. hal-04653772

HAL Id: hal-04653772

<https://hal.science/hal-04653772>

Preprint submitted on 19 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Using Neural Networks for Inelastic Neutron Scattering Cross-Sections Interpretation

Greg Henning, M. Kerveno, Ph. Dessagne (Université de Strasbourg, Centre National de la Recherche Scientifique, IPHC UMR 7178, F-67000 Strasbourg, France)

In collaboration with L. Gesson and M. Vanstalle, (Université de Strasbourg, CNRS, IPHC UMR 7178, F-67000 Strasbourg, France)

With the help of L. Petit and T. Raymond (Master Students at Université de Strasbourg).

July 2024.

Abstract: The necessary improvement of evaluated nuclear data for nuclear applications development is possible through new and high-quality experimental measurements. In particular, improving (n, n') cross-section evaluations for faster neutrons than those involved in current reactors is a goal of interest for new reactor fuels.

Our group at CNRS-IPHC has been running an experimental program to measure $(n, n' \gamma)$ cross-section using prompt γ -ray spectroscopy and neutron energy determination by time-of-flight, recording and analyzing data for $^{182,184,186}\text{W}$, ^{232}Th , $^{233,235,238}\text{U}$ [1, 2].

From the partial γ -transition measurements, the total (n, n') cross-section has to be inferred, either by summing individual contributions [3,4] (a method usually valid only up to a certain neutron energy), or by constraining reaction models with measured exclusive $(n, n' \gamma)$ cross sections [5,6]. This interpretation work is made difficult in (the usual) cases when not all the transitions going to the ground state could be measured. If that happens, one has to rely on *filling* the missing information by models or guess, reducing the accuracy of the final computed cross-section.

Here we propose a new method, involving training a Neural Network on a calculated data set and using it to predict the (n, n') cross-section from the experimental $(n, n' \gamma)$ ones. This allows a quick combination of models and experimental data. After detailing the method and checks performed for consistency, some test cases will be presented. Potential benefits, as well as the identified weakness, and future application will be discussed.

Context.

The advancement of nuclear reactor technologies heavily relies on the use of evaluated databases to perform numerical simulations. Nevertheless, the current limitations in these databases lead to uncertainties, preventing the calculations from reaching the requisite level of computational precision [7].

To improve the quality of evaluated databases and enhance the accuracy of numerical simulations, new measurements as well as updated theoretical models are needed. Furthermore, as innovative fuel cycles will use fast neutrons and novel fissile isotopes, the existing body of knowledge, predominantly focused on the $^{235}\text{U}(n_{\text{thermal}}, *)$ reactions, falls short in characterizing Generation IV reactor designs. Therefore, it is essential to conduct comprehensive investigations into isotopes such as O, F, Na, Pb, ^{232}Th , ^{233}U , ^{239}Pu , ... with a specific emphasis on reactions induced by fast neutrons.

Inelastic neutron scattering ((n, n'), as well as (n, xn) with $x \geq 2$) has some significance in the context of reactor operation. This is because they change the neutron spectrum, modify the neutron population, and create new isotopes in the fuel. However, due to a scarcity of experimental data, evaluations and models are not able today to predict (n, n') cross sections with an accuracy high enough to meet the requirements of new reactors developments. That warrants new measurements of high precision. To enhance the reliability of nuclear reaction codes, and constrain the models using precise measurement of exclusive (n, xn γ) cross sections is a good option [2].

Improving (n, n') modeling with exclusive (n, n' γ) measurements.

To infer total (n, n') cross section from exclusive (n, n' γ) ones, one can either use the many (n, n' γ) cross sections measured to constrain models' parameters (such as level density, spin distribution, gamma strength function) and calculate the total cross section with the deduced parameters [5].

Another way, that does not rely on models, is to sum a specific set of transitions. Indeed, by definition, $\sigma_{(n,n')} = \sum_{\gamma \rightarrow g.s.} \sigma_{(n,n'\gamma)} \times (1 + \alpha_{\gamma})$, where α_{γ} is the internal conversion coefficient for the given transitions. In other words, one just needs to determine the cross section for all the transitions going directly to the ground state and sum them to get the total (n, n') cross section. Figure 1 shows a schematic level scheme, with the transitions of interest highlighted.

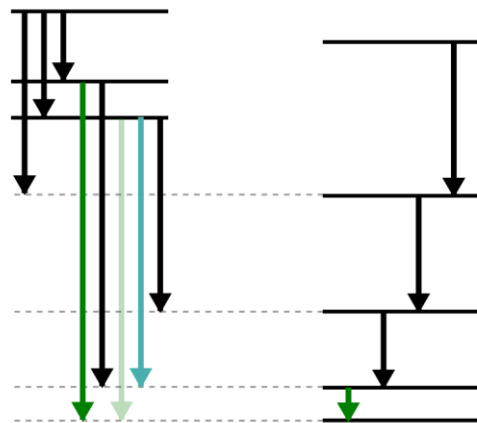


Fig. 1: Schematic nuclear level scheme, with the transitions going to the ground state highlighted in green. One transition to the ground state is not accessible experimentally is in light green, the blue transition decaying from the same level can be used as a proxy (see text).

Unfortunately, in general, it is not possible to measure all the transitions going to the ground state, either because they are too weak, outside the sensitivity range of the detector, or simply unknown. In even-even nuclei, for example, the transition from the first excited state to the ground state usually carries most of the $(n, n' \gamma)$ cross section. If it is missed experimentally, the reconstruction of the total (n, n') cross section with this method becomes almost impossible.

It is possible to go around some of these issues by using *proxy* transitions, i.e. observed transitions that decay from the same state as one of interest (i.e. going to the g.s.) that is not available experimentally, and apply a branching ratio (BR) coefficient to *reconstruct* the transition cross section we wanted originally (see the blue arrow in Figure 1).

However, this method's accuracy is highly dependent on the knowledge of the nucleus of interest. In particular, a poorly known level structure will make the processing difficult. (Additionally, the correct conversion coefficient can only be known if the transition multipolarity is well-defined.) It also introduces new uncertainties in the calculations. Furthermore, the obtained cross section is only exact up to a given incident neutron energy, above which only a lower limit can be computed (The limit is set by the excitation energy of the highest state observed in our analysis) [4].

To sum up, the weighed summing method can be quite time-consuming for limited output, as one has to find all the available transitions or their proxies, compute the branching ratio coefficients, add the correction for electronic conversion, ... all that just to obtain a limit on the total (n, n') cross section. And we also leave out

potentially a lot of transitions that have been measured, but are not used because they don't go to the ground state, or are not proxies of ones that do. This is especially true for heavy nuclei like actinides, the lighter ones (such as Oxygen and around) have a *sparse* level scheme, allowing the sum methods to be carried out accurately up to a much higher incident neutron energy.

In order to simplify and automatize the process, and maximize the use of all available $(n, n' \gamma)$ cross sections, one can try to use a large numerical model, fit it on an appropriate set of reference data, and then use this model to compute the total (n, n') cross section from the experimental $(n, n' \gamma)$ ones (as well as, possibly, other parameters).

Using Neural Networks

A Disclaimer

The authors [have no specific training](#) in *machine learning* or *neural network* and the work presented here has been performed by following tools manuals, How-to instructions and online examples.

What's a Neural Network

A *Neural Network* (NN) is a many parameters model that performs a complex (as in a lot of) series of linear combinations, with activation functions between nodes, as shown in Figure 2.

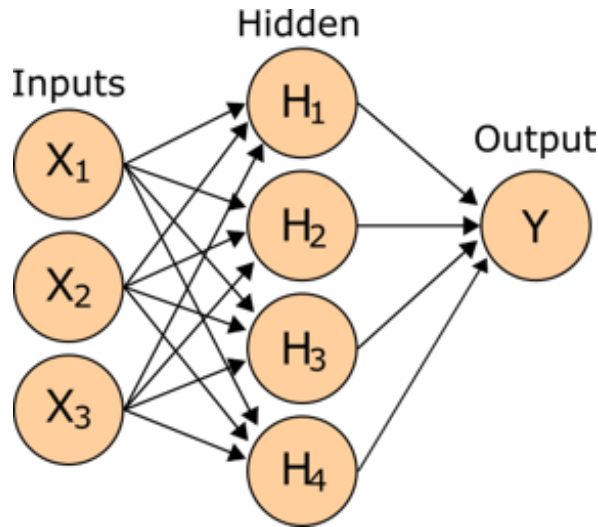


Fig. 2: Neural network schematic (image adapted from wikimedia under license CC-BY-SA-3).

Each node is fed by the previous layer values with coefficients and function to produce the output: $H_1 = \sum w_i f_{i,1}(X_i)$, where the activation function f can be $f(x) = x$, $f(x) = \max(0, x)$, $f(x) = \tanh(x)$, $f(x) = 1/(1 + e^{-x})$, ... (the last two are mostly use for categorisation problems).

Just as for a function, the many parameters may be adjusted on a set of data (i.e. *fitted*). This *training* of the neural network is done in a different way than the minimization of a χ^2 when fitting a function, but the general idea is the same.

The NN is trained and tested on a data set, and once that is done, it can be used with *new* input to compute the corresponding output. Once all the coefficients are determined, computing the final result is simple math and quick to compute; it is the training of the model that is the long and technical part.

The input can be any variable or element that can be represented by a number (organized either in vectors or matrices). In our cases, it will be $(n, n' \gamma)$ cross

sections, but that could be expanded by other variables (like incident neutron energy, isotope mass and or charge, ...). The same holds for output (it can even be generalized to outputting several values); here we will focus on computing cross sections, but any other quantities could be used.

Using a Neural Network with neutron inelastic scattering cross sections.

The objective of this study is to train a Neural Network (NN) using model calculations to take $(n, n' \gamma)$ cross sections as inputs and compute either total (n, n') cross section, or level production ones.

The test cases presented later will study the different way to use Neural Networks with $(n, n' \gamma)$ cross sections as inputs. This study will focus on the ^{184}W isotope, because a good number of experimental $(n, n' \gamma)$ cross sections are available to feed the model.

Tools

The work presented here uses the following tools, codes and libraries:

- The code `Talys` 1.8 [8] is used to compute cross sections, including total (n, n') , exclusive $(n, n' \gamma)$ and level production.
- Python 3.8 [9] is used for data handling, calculations, ... within Jupyterlab 3.1 [10] environment running on WSL2 [11] Ubuntu [12] 20.04 with the following dependencies:
 - Numpy 1.21 [13] for data handling,
 - Matplotlib 3.4 [14] for visualization,

- and Scikit-learn 1.3 [15] for the *machine learning* part.

The data used will be the cross sections for total inelastic neutron scattering ($\sigma_{184W(n,n\prime)}$), exclusive γ channels ($\sigma_{184W(n,n\prime\gamma)}$), and level productions ($\sigma_{184W(n,n\prime J^\pi)}$) computed by Talys. The Talys data is split in two subsets, one for training the NN, the other for testing/validation of the

Inputs and comparison data will be experimental $\sigma_{184W(n,n\prime\gamma)}$ [16, 17] and level production cross section from *Guenther et al., 1982* [18].

Test case 1: from $(n, n' \gamma)$ s to $(n, n' 2_1^+)$.

The $(n, n' \gamma)$ cross sections measured with Grapheme [16,17] are the only data of this kind. Therefore, no comparison to similar data is possible to check that the experimental results are not *off* because of mistake or issues in the analysis.

The NN approach will first be used to perform $(n, n' 2_1^+)$ interpolation from $(n, n' \gamma)$ and compare with existing data from *Guenther, et al.* [18]. To that end, the $(n, n' \gamma)$ cross section obtained experimentally are collected, as well as the matching ones computed by Talys, to constitute the training set.

The *input* $(n, n' \gamma)$ are scaled down each to 2/3 of the maximum valued obtained in Talys computation, so that the NN input (both in training and use) are between 0 and 1 (each input transition has its own scaling factor). This ensures that no specific input is much larger than the other, and helps *balance* the Neural

Network. With the way we use the Neural Network, using the rectified linear unit activation function - $f(x) = \max(0, x)$ - if any input is larger than 1, this will not be an issue. The experimental data is also interpolated so that all $(n, n' \gamma)$ input are at the same neutron energy (as, depending on statistics and thresholds, the experimental point may not be all extracted at the same incident neutron energy). The prepping of input data and the reading of the output allows the scaling to be completely automatic and transparent to the user.

The workflow of the NN training, validation and use is shown in Figure 3.

Once the NN is trained on Talys data, it is validated by running it on the *testing* part of the Talys set. Since the expected results is known, we can verify that the Neural Network prediction matches the target.

We then perform a series of tests to verify that the model has been trained correctly, without *stupidly* outputting whatever it has been trained on whatever the input may be.

1. Garbage in / garbage out

For this test, the input is randomly generated between 0 and 1 for all input channels. The output of the NN for this *garbage* input is completely random (the colored dots in Figure 4), confirming that the calculation of the NN is sensitive to the input (i.e the NN is not over-trained to reproduce the training Talys data whatever the input may be).

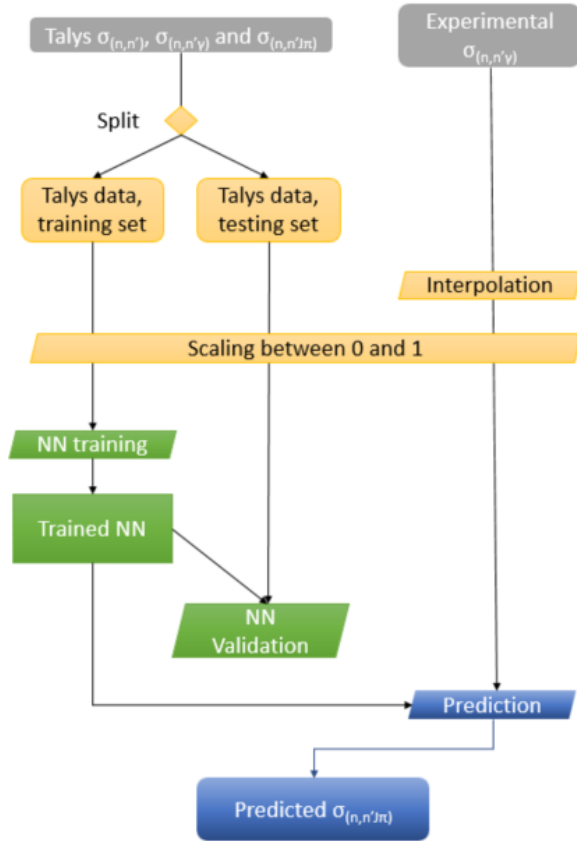


Fig. 3: Workflow of the NN data preparation (yellow), training and validation (green) and use for the prediction (blue) of a $(n, n' J^\pi)$ cross section.

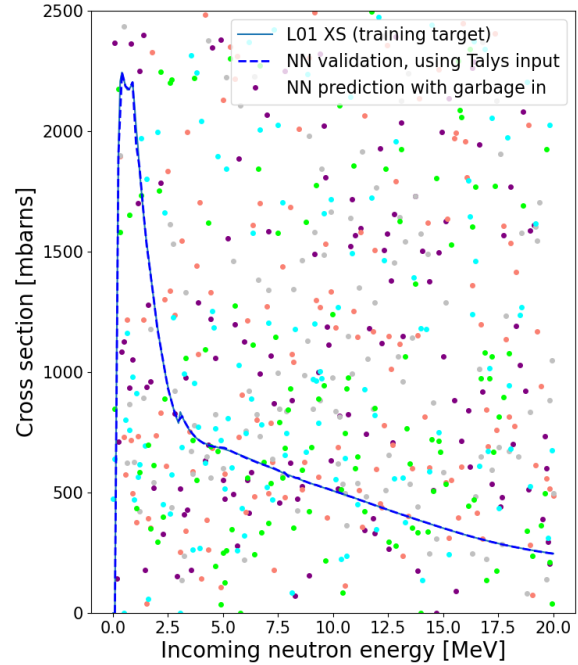


Fig 4: NN predictions with randomized input (using 5 different series of random input). The Full line is the Training target. The dashed line the NN validation prediction. (See text.)

2. Input variations around nominal value

Next, the validation input is randomly varied around its nominal value by 5, 10, and 20 %, we can verify that the output varies around the target central value (i.e. prediction from the unperturbed input). Figure 5 shows the results of these different computations. This confirms that the model is stable around the central value.

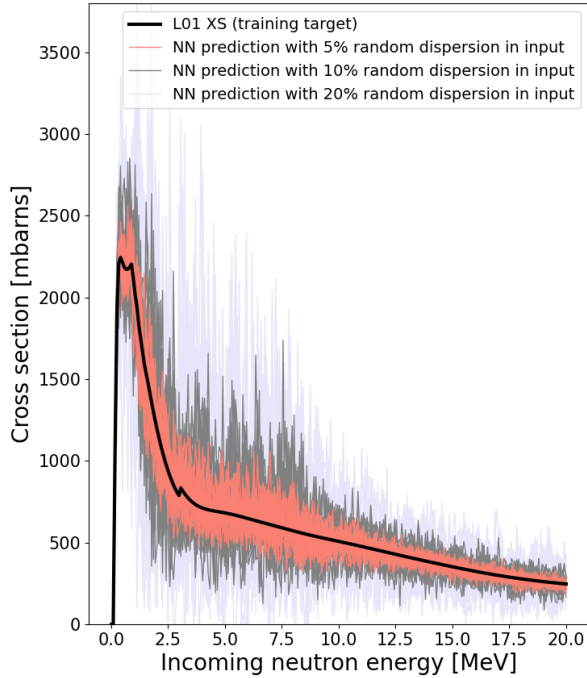


Fig. 5: NN predictions with input values randomized around their nominal values using a normal distribution of 5 (red), 10 (dark gray) and 20 % (light gray).

3. Randomize hidden layers size and test the stability of prediction

Finally, we run different NNs that have been trained using different size (and number) of hidden layers, around the values that give a correct output. We could check that the prediction of the model is stable. With 35 different NNs (in Figure 6), there is a clear spread in results, but one can see that by using a large series of NN, a central value and uncertainty can be derived, that's compatible with Guenther et al. [18].

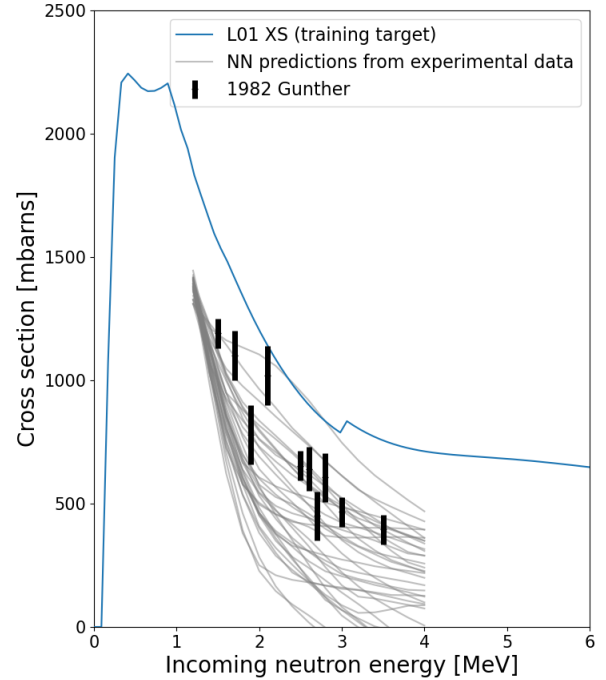


Fig. 6: NN predictions from different NNs with randomized number and size of hidden layers. Each gray line is the prediction from one NN (with experimental data as input), and compared to the reference experimental values from Guenther et al. [18] in black points.

Results

After confirming that the NN works as expected with the previous tests, we used 3 different NNs (different number of nodes in the hidden layers), with several iterations of randomization of the input values (experimental $\sigma_{^{184}W(n,n'\gamma)}$) according to their uncertainty. From these calculations, we extract 3 central values (one for each NN) for the prediction of the $(n, n' 2_1^+)$ cross section, with an associated standard deviation. The results are shown in Figure 7.

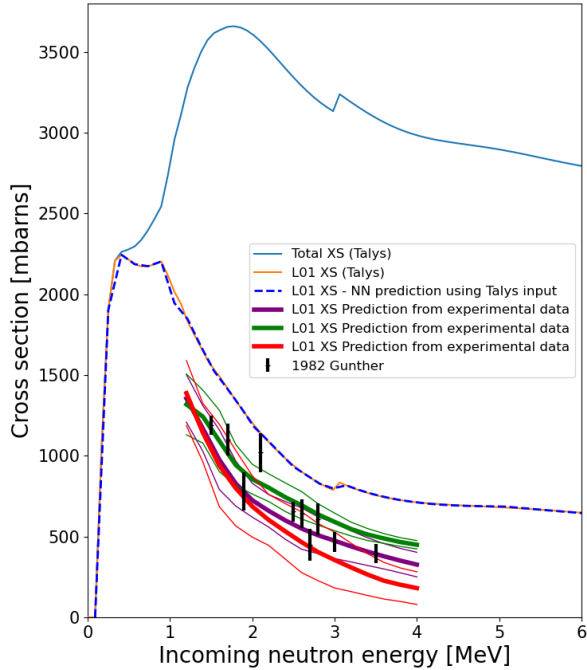


Fig. 7: NN predictions from three different NNs with variation of the inputs around their central value according to their uncertainty. The total (n, n') cross section calculated by Talys is given (for reference) in full blue line. The training target is the orange line and the dashed blue line is the NN validation prediction. The prediction from the NNs are the color lines (thick line is the central values; thin lines delineate the ± 1 standard deviation range). The experimental values from Guenther et al. [18] are the black points.

The σ_{L01} derived from experimental $(n, n' \gamma)$ with NN trained on Talys is compatible with previous reference [18].

This result is particularly important, as it shows that the NN approach can help us *transform* the experimental data in another form, so that it can be compared to previous results.

This first case gives us confidence in the experimental data and encourages us to continue investigating the use of NN.

Test case 2 : from $(n, n' \gamma)$ cross sections to (n, n') .

The main focus of our work is $(n, n' \gamma)$ cross sections measured with Grapheme [2, 4, 16, 17]. Having shown in the previous section that NNs can predict, after training on model calculations and from experimental data as input, a reasonable output, we can try to perform the $\{\sigma_{(n, n' \gamma)}\}_{\{\gamma\}} \rightarrow \sigma_{(n, n')}$ transformation.

As previously, we will use input scaling, multiple NNs with different hidden layers size, and variation of the input data around their central value, following their uncertainty.

Here, we have two data sets to train the NNs: one is a default Talys calculation performed by the authors, the other is a Talys calculation done by P. Romain [19]. The two different calculations use different model parameters (optical model potential, γ strength functions, level densities, ...). The default ones use a general *all-purpose* parameter set, while P. Romain's one uses parameters optimized for deformed nuclei like Tungsten isotopes. This will help us study the dependence of the prediction to the training set (both training sets represent the same reaction).

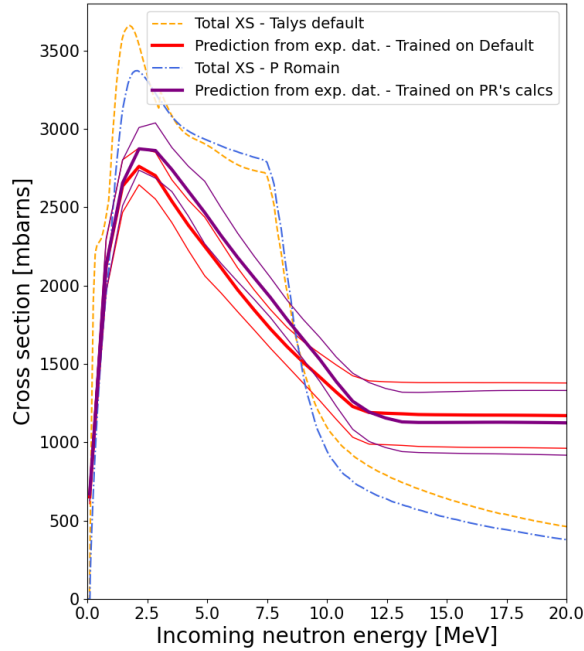


Fig. 8: NN predictions from two different NNs trained on different model calculation sets (see text), with variation of the experimental inputs around their central value according to their uncertainty and the means. The resulting red and purple lines show the average NN predicted values for the cross section with the default Talys training (red) and P. Romain's computation for training (purple).

Results are shown in Figure 8. The two NN models predict very similar (n, n') cross section from the experimental $(n, n' \gamma)$ ones (well within the uncertainties limit of each other), suggesting only a very small dependence to the training set.

Still, the NNs derived total (n, n') cross section predictions are below Talys calculations in amplitude, they are consistent with what we see in the individual $(n, n' \gamma)$, where the observed

$L01L00^1$ transition from the first excited state to the ground state is also significantly below model calculations [16, 20].

Test case 3: from $(n, n' \gamma)$ s to $(n, n' \gamma_{L01L00})$.

This leads us to our third investigation: checking the consistency of our experimental $L01L00$ cross section against all the others.

Indeed, the experimental cross section for the $^{184}\text{W}(n, n' \gamma_{L01L00})$ is below the values computed using reaction models, as seen in Figure 9.

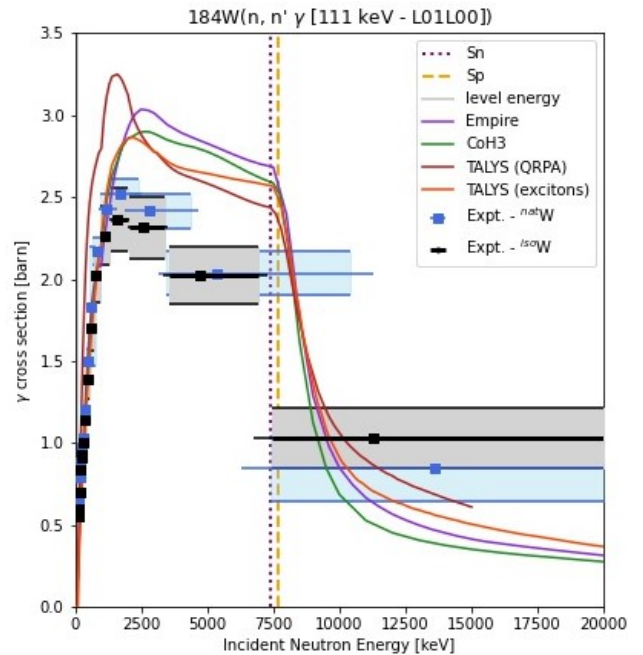


Fig. 9: Experimental $^{184}\text{W}(n, n' \gamma_{L01L00})$ cross section compared to reaction code calculations. (from [16]).

¹ Talys labels the transitions $L_{ii}L_{ff}$ with ii and ff the indices of the initial and

final levels, in order of increasing excitation energy.

The goal is now to test with the Neural Network the consistency of the $^{184}\text{W}(n, n' \gamma_{L01L00})$ experimental data with the other γ -ray cross sections obtained in the same experiment.

To that end, following the same steps as before, NNs are trained with the experimental $(n, n' \gamma)$ cross sections as input, with the specific $(n, n' \gamma_{L01L00})$ one excluded from the input, and used as target. The inputs are interpolated over a range of incident neutron energy, in order to have a meaningful number and position of points.

The processed is repeated with multiple NNs, with different hidden layers size, and with variation of the inputs around their central value according to their uncertainty. The resulting central value and standard uncertainty is displayed in Figure 10.

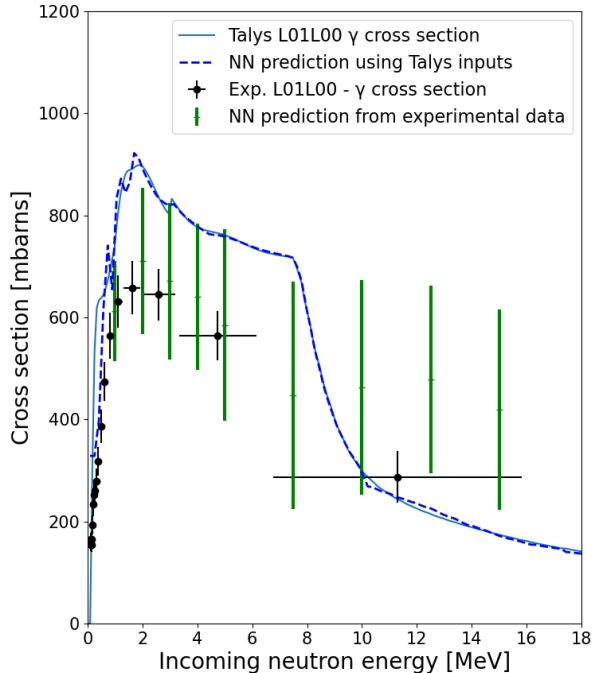


Fig. 10: Predictions averaged from many different NNs fitted on the experimental $(n, n' \gamma)$ cross sections as input, with the

specific $(n, n' \gamma_{L01L00})$ one used as target. The Talys training and validation curve are in full and dashed blue lines respectively. The experimental values from [17] in black points. And the predictions from NNs in green points.

The output is not very conclusive: in average, the NN predictions are compatible with the experimental $^{184}\text{W}(n, n' \gamma_{L01L00})$ without being too far off from the Talys values. Therefore, it is not possible to directly conclude that the experimental $^{184}\text{W}(n, n' \gamma_{L01L00})$ cross section is not consistent with the other experimental $(n, n' \gamma)$ cross sections.

However, the possibility to check the consistency of experimental value is a good application of the NN computation, and could be applied to other results, in particular when possible contamination may be suspected.

Conclusion.

The objective was to try using Neural Networks to help interpret experimental $(n, n' \gamma)$ cross sections. In this paper, we present preliminary results, using NNs *naïvely*.

The NNs have been trained on Talys calculations and were able to *transform* $\sigma_{(n,n')\gamma}$ into level excitation cross section, which allowed a comparison with existing experimental data. The NNs were tested for stability, consistency, ... using various tests and variations in the hidden layers size, training set, ...

The interest of the method is that it can be used to check data consistency, and validate experimental data when no direct comparison is possible. It provides a way to constrain model calculations with experimental input very quickly, in

particular, when the amount of available experimental data is too low to allow the use of a more direct method.

This first tries being successful (or at least not too disastrous), we will, in the future, improve on the NN hyper parameters with a cross validation search for the optimum ones [21], and generalize the process to incorporate better inputs and streamline the testing and validation.

Finally, we started a collaboration with the ion therapy team in IPHC, in order to evaluate the possible use of Neural Networks for large scale nuclear reaction cross-section inference using models and experimental values.

References

1. "What can we learn from (n,xny) cross sections about reaction mechanism and nuclear structure?" Maëlle Kerveno, Marc Dupuis, Catalin Borcea, Marian Boromiza, Roberto Capote, Philippe Dessagne, Greg Henning, Stéphane Hilaire, Toshihiko Kawano, Alexandra Negret, Markus Nyman, Adina Olacel, Eliot Party, Arjan Plompen, Pascal Romain and Mihaela Sin. EPJ Web Conf., 239 (2020) 01023 [10.1051/epjconf/202023901023](https://doi.org/10.1051/epjconf/202023901023) <https://hal.archives-ouvertes.fr/hal-02957494>
2. "From γ emissions to (n, xn) cross sections of interest: The role of GAINS and GRAPHEME in nuclear reaction modeling", by Kerveno, M. and Bacquias, A. and Borcea, C. and Dessagne, Ph. and Henning, G. and Mihailescu, C. and Negret, A. and Nyman, M. and Olacel, A. and Plompen, M. and Rouki, C. and Rudolf, G. and Thiry, C. in European Physical Journal A 51, 12 (2015). [10.1140/epja/i2015-15167-y](https://doi.org/10.1140/epja/i2015-15167-y) <https://hal.archives-ouvertes.fr/hal-02154831>
3. "Neutron inelastic cross-section measurements for ^{24}Mg ". A. Olacel, C. Borcea, P. Dessagne, M. Kerveno, A. Negret, and A. J. M. Plompen. Phys. Rev. C 90, 034603 (2014). [doi:10.1103/physrevc.90.034603](https://doi.org/10.1103/physrevc.90.034603)
4. "From $^{232}\text{Th}(n, n'\gamma)$ cross sections to level production and total neutron inelastic scattering cross sections" N. Dari Bako, et al. EPJ Web of Conf., 284 ,08005 (2023) <https://doi.org/10.1051/epjconf/202328408005>
5. "How to produce accurate inelastic cross sections from an indirect measurement method?", by Kerveno, Maëlle and Henning, Greg and Borcea, Catalin and Dessagne, Philippe and Dupuis, Marc and Hilaire, Stéphane and Negret, Alexandru and Nyman, Markus and Olacel, Adina and Party, Eliot and Plompen, Arjan in EPJ N - Nuclear Sciences & Technologies 4, (2018). [10.1051/epjn/2018020](https://doi.org/10.1051/epjn/2018020) <https://hal.archives-ouvertes.fr/hal-02109918>
6. "Cross-section measurements for the $^{57}\text{Fe}(n,\text{n}\gamma)^{57}\text{Fe}$ and $^{57}\text{Fe}(n,2\text{n}\gamma)^{56}\text{Fe}$ reactions" A. Negret, M. Sin, C. Borcea, R. Capote, Ph. Dessagne, M. Kerveno, N. Nankov, A. Olacel, A. J. M. Plompen, and C. Rouki. Phys. Rev. C 96, 024620 (2017)[doi:10.1103/physrevc.96.024620](https://doi.org/10.1103/physrevc.96.024620)
7. NEA/WPEC-26, "Uncertainty and Target Accuracy Assessment for Innovative Systems Using Recent Covariance Data Evaluations" (2008).
8. A.J. Koning, S. Hilaire and M.C. Duijvestijn, "TALYS-1.0", Proceedings of the International Conference on Nuclear Data for Science and Technology, April 22-27, 2007, Nice, France, editors O.Bersillon, F.Gunsing, E.Bauge, R.Jacqmin, and S.Leray, EDP Sciences, 2008, p.211-214.
9. Van Rossum, G. & Drake, F.L., 2009. Python 3 Reference Manual, Scotts Valley, CA: CreateSpace.
10. Kluyver, T. et al., 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt, eds. Positioning and Power in Academic Publishing: Players, Agents and Agendas. pp. 87–90
11. Windows Subsystem for Linux

12. ubuntu.com
13. Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
14. J. D. Hunter, “Matplotlib: A 2D Graphics Environment”, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
15. F. Pedregosa, et al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research 12 (2011): 2825-2830.
16. G. Henning, Antoine Bacquias, Catalin Borcea, Mariam Boromiza, Roberto Capote, et al. Measurement of $^{182,184,186}\text{W}$ ($n, n' \gamma$) cross sections and what we can learn from it. PHYSOR 2020: Transition to a Scalable Nuclear Future, Mar 2020, Cambridge, United Kingdom. (hal-02956052)
17. G. Henning, “Experimental ($n, n' \gamma$) cross sections for isotopes 182,184 and ^{186}W ”, <https://doi.org/10.57745/JRCNEJ> (2024).
18. Fast Neutron Total and Scattering Cross Sections of ^{182}W , ^{184}W , ^{186}W . P.T. Guenther, A.B.Smith, J.F.Whalen <http://www-nds.iaea.org/EXFOR/10803.015>
19. Pascal Romain, CEA, DAM, DIF, France. Private communication.
20. Improving the accuracy of $^{182,184,186}\text{W}$ ($n, n' \gamma$) cross sections calculations. G. Henning et al. In preparation for publication.
21. Loïc Estève, "INRIA/scikit-learn-mooc". doi: 10.5281/zenodo.7220306.