

PAPER • OPEN ACCESS

Towards automated analysis for neutron reflectivity

To cite this article: Daniil Mironov *et al* 2021 *Mach. Learn.: Sci. Technol.* **2** 035006

View the [article online](#) for updates and enhancements.

You may also like

- [A general approach to maximise information density in neutron reflectometry analysis](#)
Andrew R McCluskey, Joshaniel F K Cooper, Tom Arnold et al.
- [Theoretical characterization of uncertainty in high-dimensional linear classification](#)
Lucas Clarté, Bruno Loureiro, Florent Krzakala et al.
- [InClass nets: independent classifier networks for nonparametric estimation of conditional independence mixture models and unsupervised classification](#)
Konstantin T Matchev and Prasanth Shyamsundar



PAPER

Towards automated analysis for neutron reflectivity

OPEN ACCESS

Daniil Mironov¹, James H Durant¹ , Rebecca Mackenzie² and Joshaniel F K Cooper¹ RECEIVED
25 June 2020¹ ISIS Neutron and Muon source, Rutherford Appleton Laboratory, Harwell Campus, Didcot OX11 0QX, United KingdomREVISED
8 January 2021² SciML, Scientific Computing Division, Rutherford Appleton Laboratory, Harwell Campus, Didcot OX11 0QX, United KingdomACCEPTED FOR PUBLICATION
18 February 2021E-mail: Jos.Cooper@stfc.ac.ukPUBLISHED
23 April 2021

Keywords: neutron, x-ray, reflectivity, machine learning, neural network

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Abstract

We describe a neural network-based tool for the automatic estimation of thin film thicknesses and scattering length densities from neutron reflectivity curves. The neural network sits within a data pipeline, that takes raw data from a neutron reflectometer, and outputs data and parameter estimates into a fitting program for end user analysis. Our tool deals with simple cases, predicting the number of layers and layer parameters up to three layers on a bulk substrate. This provides good accuracy in parameter estimation, while covering a large portion of the use case. By automating steps in data analysis that only require semi-expert knowledge, we lower the barrier to on-experiment data analysis, allowing better utility to be made from large scale facility experiments. Transfer learning showed that our tool works for x-ray reflectivity, and all code is freely available on GitHub (neutron-net 2020, available at: <https://github.com/xmironov/neutron-net>) (Accessed: 25 June 2020).

1. Introduction

In the operation of large-scale facilities, costs run into the hundreds of thousands of pounds per day, and the lead time for researchers applying for beamtime can be months. Therefore, any tool that increases the efficacy of the measuring period can have a huge impact. In this paper, we present a machine learning (ML) based data pipeline for neutron reflectivity model selection, which takes a reflectivity curve and uses it to predict thin film layer properties before sending these predictions and the data to a conventional fitting program for detailed analysis by the user.

In a traditional reflectometry pipeline, the output of an instrument is given in the format of a three column ASCII file, despite recent effort to standardise the inclusion of metadata (Open Reflectivity Standards Organisation (ORSO) 2021). A model for this data is constructed manually using knowledge of the experimental setup and underlying science. The model can then be fitted according to, for example, a chi-squared metric, using standard optimisation routines such as differential evolution, genetic algorithms or similar. This process is currently performed entirely by hand, and often cannot keep up with the pace of experiments.

In our tool, reflectivity data is taken from an instrument and passed directly to our ML models which predict the number of layers present as well as those layers' properties and the data itself are then fed into the reflectivity analysis Python package *refnx* (Nelson and Prescott 2019), for fine tuning and interpretation by the end users. This process is completed automatically in seconds, rather than minutes of repetitive manual work, and requires no user input. In effect, this lowers the barrier to analysis and reduces the time to obtain analysed data. We thereby allow users to spend the bulk of their time interpreting results. Modification for use in other fitting software (Björck and Andersson 2007, Kienzle 2020) is relatively straightforward.

Several other recent attempts have been made towards automation of reflectivity analysis (Greco *et al* 2019, Carmona-Loaiza and Raza 2020), however, the transformation of the data into layer properties is non-trivial and no general solution can exist. All approaches thus far have severe limitations, from fixed inputs or outputs, to extremely simplified systems. Our approach overcomes most of these limitations, allowing any reflectivity curve as an input and any physically measurable values for an output. We have limited the implementation demonstrated here to scattering length density (SLD) profiles consisting of a

maximum of three layers, since the accuracy of predictions will decrease as the complexity increases. However, the purpose of the tool is not to replace detailed analysis, it is to know the general film properties quickly and with no effort; for this we deemed three layers sufficient to cover most cases.

We do not remove the need for an expert's interpretation, we simply automate the tasks which do not need it. Our pipeline enables on-experiment analysis, informing choices about changing experiment conditions or samples *in operando*, saving time and effort.

2. Methods

In neutron reflectivity, the properties of thin films or thin film heterostructures are elucidated by measuring the reflectivity of a collimated neutron beam from the film surface. This is done as a function of momentum transfer, Q , which depends on the incident angle, θ , and neutron wavelength, λ , as

$$Q = 4\pi \frac{\sin \theta}{\lambda}.$$

The reflectivity is governed by the Fresnel reflectivity, decaying with Q^{-4} for bulk substrate (i.e. a single interface); see Sivia (2011) for an introduction. When thin films are present on the substrate, interference occurs and the reflectivity oscillates around the Fresnel curve. For single reflections, this tends to Bragg's law (Bragg and Bragg 1913), and the period of the oscillations, ΔQ , is just governed by the layer thickness, d :

$$\Delta Q = \frac{2\pi}{d}.$$

Interpretation of a reflectivity curve is complicated by the fact that multiple reflections can occur, in fact only a single layer can be solved analytically. The loss of phase information upon reflection makes determining the layer structure a classic inverse problem, so in practise, to transform from a SLD profile to a reflectivity curve, the reflectivity is calculated recursively using the Parratt formalism (Parratt 1954), or similar. As long as the instrument resolution is correctly applied, this results in a dataset which describes potential experimental data extremely well, and for simulated data gives a known ground truth. This process is also relatively fast, meaning that a very large amount of training data can be produced inexpensively for a neural network.

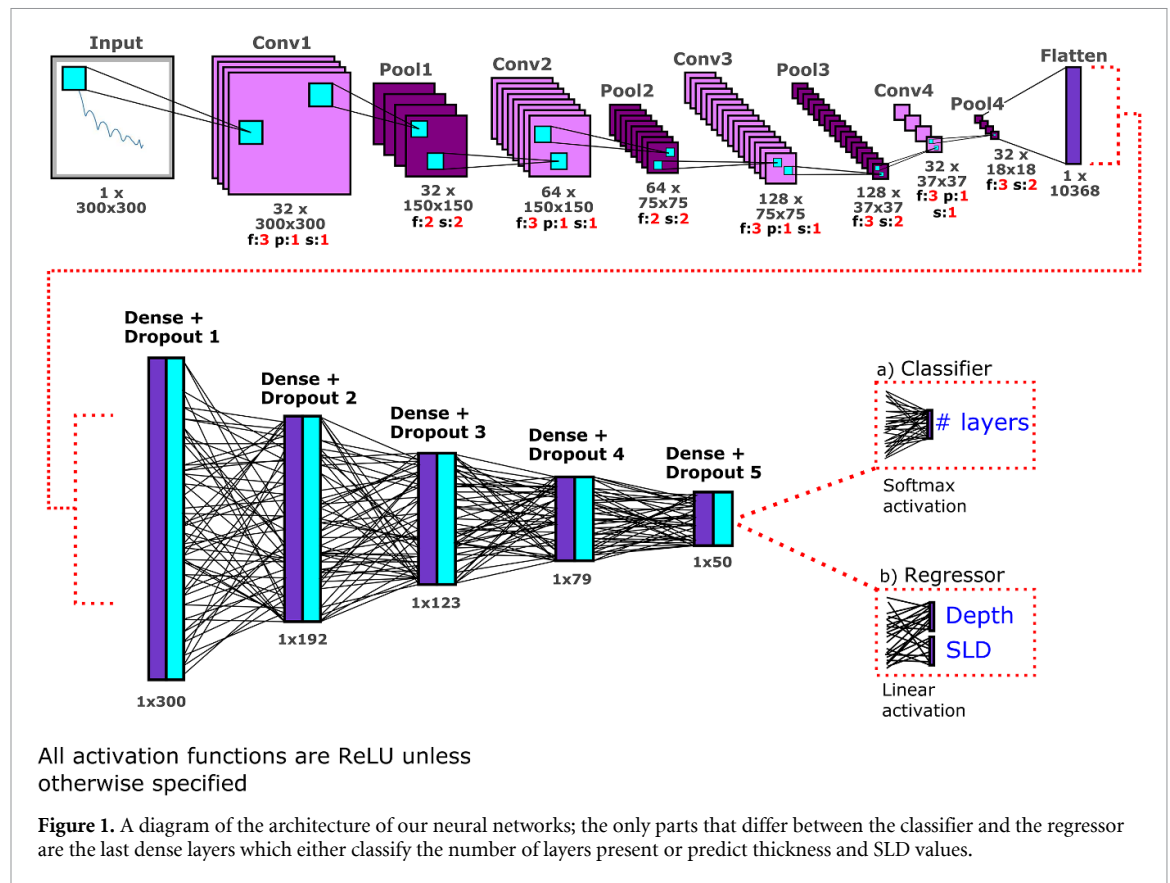
Such a set of idealised, noiseless layer structure data was generated in this manner using `refnx` (Nelson and Prescott 2019), with the simulation of 50 000 one-, two- and three-layer sample systems. Realistic artificial noise was then added to this noiseless data to produce a synthetic dataset as close to real data as possible. All layers used silicon as a substrate, applied a constant dQ/Q resolution of 2% and used a scale factor of 1. For each sample, a roughness was randomly sampled from a uniform distribution in the interval $[2, 8] \text{ \AA}$ and applied between all interfaces of that sample. For each system, the layer SLDs were randomly sampled from a uniform distribution in the interval $[-1, 10] \times 10^{-6} \text{ \AA}^{-2}$. This range encompasses most materials likely to be measured. Layer thicknesses were randomly sampled from a Gaussian-like distribution biased towards thinner layers within the interval $[20, 1000] \text{ \AA}$. These steps intentionally biased our dataset to see realistic reflectivity curves, rather than completely random ones. Given our intention to use neural networks to train on these data, the one-, two- and three-layer samples were shuffled together to remove any ordering bias that could potentially be introduced.

Each synthetic sample consisted of 500 momentum transfer and reflectivity coordinates (Q, R) spaced evenly on a logarithmic scale over $[0.005, 0.3] \text{ \AA}^{-1}$; this simulated the experimental difficulty in obtaining points in high Q . These 500 (Q, R) points were plotted to create a line graph represented as an RGB image or alternatively an array of dimensions $300 \times 300 \times 3$. This plot was then converted to greyscale and stored in a HDF5 file as a $300 \times 300 \times 1$ array of raw grayscale pixel values. The ground truths for the per-layer SLDs and thicknesses that we would use to train our networks were normalised between 0 and 1 (LeCun *et al* 2012).

Since noise in a real measurement is typically caused by counting statistics, we developed a novel approach to try and reproduce this. For a given Q point, i , the number of measured neutrons, N_i , will be the product of the true reflectivity, r_i , and the number of incident neutrons, μ_i at that momentum transfer:

$$N_i = r_i \mu_i.$$

To replicate this, we used a measured incident spectrum and converted it into a reflectivity curve in order to ascertain an estimate for the number of incident neutrons per Q point. Each reflectivity point is then randomly relocated somewhere in a normal distribution, centred on the ground truth and whose width depends on the incident flux and the true reflectivity, i.e. number of counts measured at that point. We can



modify this by some arbitrary scale factor to allow the noise to be increased or decreased, analogous to changing the experimental run time.

The core of our ML approach consisted of first designing a classification network that would indicate the number of thin film layers for a given dataset. This would be then fed into a regression network that would produce predictions on the SLDs and thicknesses of the layers within the sample system. A holistic approach was attempted, with an ‘all-in-one’ model that would perform both functions, but an explicit separation of classification and regression tasks saw better performance, albeit at the cost of increased complexity.

The architectures of both the classification and regression networks were largely the same. An initial four-layer convolutional neural network (CNN) filters across the greyscale images with max-pooling and uses rectified linear unit (ReLU) activation functions between the network layers. Such filtering reduces input images from a $300 \times 300 \times 1$ input to an $18 \times 18 \times 32$ representation. This representation is then flattened to a 1D array of size $10\,368 \times 1$ to be passed into a feed-forward dense neural network. This section includes six dense layers, coupled to dropout layers (Srivastava *et al* 2014). These layers, bar the last, also use ReLU activation functions. The sixth layer is where the networks differ, with a softmax activation function for the classifier, and linear activation functions for the regressors.

The hyperparameters for both the classifier and regressors were a learning rate of 3×10^{-5} using Nadam (Dozat 2016) with Keras default β_1 , β_2 , decay and epsilon, and a batch size of 40 images per sample. The classifier was trained for 80 epochs on 150 000 shuffled curves consisting of an equal number of one-, two- and three-layer samples (50 000 of each). The regressor for the one-layer samples was trained for 100 epochs (though this network achieves close to its final accuracy after ~ 10 epochs), and the regressors for the two- and three-layer samples were trained for 150 epochs. As seen in figure 1 the regression networks force the models’ distinct output nodes (thickness and SLD) to share parameters until the final layer. This shared-parameter nature for both types of outputs offered more robust models (Caruana 1993, Baxter 1997, Ruder 2017).

A generic data pipeline was then constructed in which the trained neural networks sit. The pipeline starts with Mantid (Arnold *et al* 2014) processing the raw instrument data (in this case from the Offspec neutron reflectometer) and outputting a three column CSV file. We take this file and normalise it such that its maximum reflectivity is 1 to ensure that the input real data is as close to the synthetic training dataset as possible, thus improving prediction accuracy. The reflectivity data is then plotted, the resulting image converted to greyscale, and passed through the classifier to determine the number of layers. The prediction is

then used to select an appropriate regressor that analyses the image to produce SLD and thickness predictions for all the thin film layers. These predictions are then fed into the reflectivity fitting software package *refnx* (Nelson and Prescott 2019) and are used as initial ‘guesses’ for its optimisation algorithms. The fitting software can then perform its optimisation, getting to a model that likely describes the given sample. With the Python script running in the background on the experiment directory, users can receive fitted data in ~ 10 s with no input.

When applying transfer learning, we took our networks trained on neutron reflectivity data and partially retrained them with x-ray reflectivity data. We found that the best accuracy was obtained by allowing the entire network to be trainable. This is in contrast to most transfer learning, where either a few layers are set to be trainable, or the entire network is fixed, and a few trainable layers are added to the network. We believe the reason that this approach worked so well, is because x-ray and neutron reflectivity are so extremely similar, that the differences amount to a small scaling of the predicted values.

To demonstrate one possible application of our tool, we simulated a batch of datasets, with two layers, one of which became thicker with ‘time’. All other parameters of the sample remained fixed; this effectively represented an experiment observing a thin film being grown *in situ*, e.g. in a sputter chamber. The data was simulated (so we had a known ground truth), and then had noise and background added to make it more representative of a real measurement. The top layer’s thickness varied in the range [100, 850] Å. The bottom layer had a fixed 100 Å thickness. The SLDs of the first and second layer were $2.5 \times 10^{-6} \text{ \AA}^{-2}$ and $5.0 \times 10^{-6} \text{ \AA}^{-2}$ respectively.

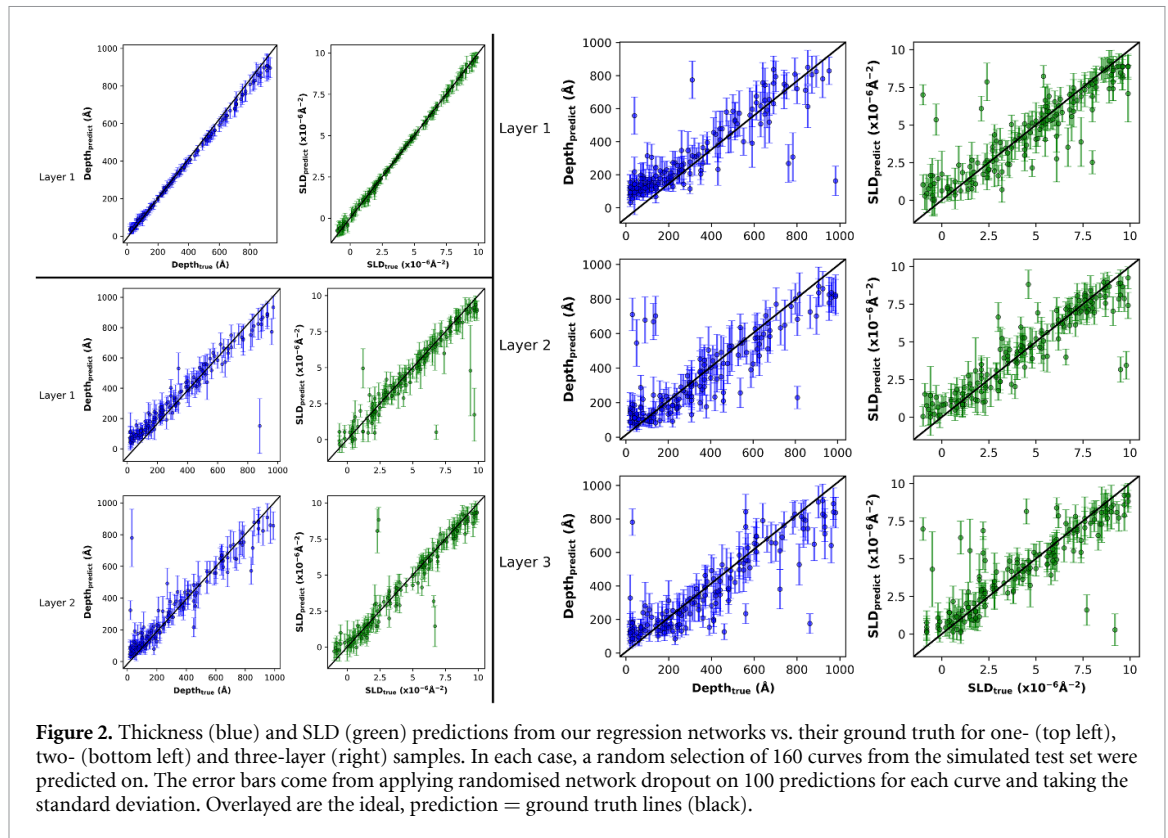
3. Results and discussion

Before discussing the results, we will make some comments on certain decisions made in the design of the networks. One of the most noticeable choices was to feed reflectivity data into our ML networks in the form of images. This was not the first, nor the most obvious approach, however there are several subtleties associated with our problem that rationalise our decision. Firstly, neutron instruments are not simple black boxes that give you a set answer. They are configured for each individual experiment and, as consequence, the output data size changes; we must be able to take account for this in our network input. Secondly, the data consists of co-ordinates, and what is more, these co-ordinates are in Fourier space. Therefore, the position of features in the data is important as they correspond to different real space lengthscales. These two points effectively ruled out the use of a solely densely connected network, as feeding in variable positions is not possible and so some form of 2D interpolation is necessary since we do need to have a fixed input size. As a result, we chose to use $300 \times 300 \times 1$ images as inputs to our network, plotting the data over a fixed Q and reflectivity range defined by the instrument’s minimum and maximum measurable settings. This has the benefit of easily allowing zeroes in the data and not causing dead units in the network.

As previously stated, the raw datasets are given as a series of (R, Q) coordinates typically between 50 and 500 points. These, when plotted and converted into a $300 \times 300 \times 1$ image yield a huge increase in data size, with seemingly no increase in information content. We found, however, that this data representation is flexible and generalisable as an input into the network and has tangible performance benefits. Using images, we found significant improvements in the networks’ accuracy and loss metrics, presumably due to the input being better suited for the network architecture; CNNs are well designed to extract spatial information, as is desirable in our case with co-ordinate inputs. After a variety of optimisations, including the use of data-loading generators, the training times for the networks on a single desktop graphics processing unit (GPU) were ~ 6 h for a one-layer system, ~ 9 h for two-layer system, ~ 12 for a three-layer system and ~ 18 h for the classification network, thus negating some of the drawbacks from using an artificially inflated data size.

The classifier was trained on one-, two- and three-layer samples (150 000 in total) and achieved an average accuracy of 95.6% after 80 epochs. This consisted of a 99.7%, 93.1% and 94.0% accuracy for one-, two- and three-layer samples respectively. The networks were initially trained on noiseless data, as a proof of the principle and given such perfect data, the classification network achieved an accuracy of 99.7%. Given the strengths of CNNs for image classification purposes, this result was not particularly surprising. However, the fact that it was then possible to use a CNN to predict the thicknesses and SLD’s of a reflectivity curve, from an image of that curve, was intriguing, and may warrant further investigation as an approach. The discrepancy between this noiseless accuracy and the former is likely due to the indistinguishability of some two- and three-layer samples in the presence of significant noise.

As one would expect, when realistic noise was added useable information available at high Q decreased. When deciding how many points to use for the training reflectivity arrays, we found that using many points to train did not seem to worsen network performance on real data which is made up of fewer points. This is likely due to the fact that the networks trained to ignore these high Q values. However, increasing the



number of points per sample did significantly improve training losses as more information was available elsewhere to base predictions on.

From the data shown in figure 2, it is clear that the SLD and thickness values for any single layer are almost trivially predicted for all thicknesses and SLDs. This may not necessarily have been expected, but is promising, since the single layer problem is solvable analytically (Sivia 2011). In fact, though not shown, even a single epoch of training gives results almost comparable to those obtained after 100 epochs. The implication here is that the network could potentially be trained on less data for these types of systems, and still achieve a suitable performance. As the number of layers present in a system are increased, the complexity of the problem naturally increases also. Due to this, we experience a decrease in the accuracy of predictions, as shown in figure 2, though the accuracy does remain high.

In addition to the prediction, our network also outputs an uncertainty associated with each data point. This represents the standard deviation of 100 predictions made by the network per data point, with randomised dropout applied during prediction. The uncertainty is therefore representing that which is inherent in the network, or model. In this application, this technique has proven well to describe the real error, since most points have error bars that overlap with the prediction equals ground truth line. We should note that this approach does not try to directly infer the uncertainty of the prediction, as other concepts do, e.g. Hafner *et al* (2018, 2019), but instead functions as a compact Bayesian approximation (Gal and Ghahramani 2015). A notable benefit of this implementation is that it can be used in any network already utilising dropout layers, with no change in network architecture required.

We used our network to predict on some real data, taken on the Offspec reflectometer at the ISIS Neutron and Muon source. We show the normalised data and the corresponding predictions from the network in figure 3. Also shown in figure 3 are the SLD profiles of the ground truth models obtained via fitting against the corresponding predicted profiles. The inputs to the pipeline were very much an untreated reflectivity file, consisting of three column ASCII files (Q , R , dR). For curve 1, this was 434 data points and for curve 2, this was 129 points; both included some points that were zero. They were both measured at differing Q points and over different Q range, demonstrating the flexibility of our approach. The predictions are not perfect, table 1. Comparison of predicted and fitted values for real data passed through the neural network, but are very usable and are fitted in several seconds by `refnx`, showing the utility of the tool.

The results of the time varying experiment are shown in figure 4, with an exemplar dataset inset to show the data being predicted upon. The raw outputs of the network without any subsequent fitting show excellent agreement with the ground truths. However, the deviation between ground truth and predicted layer 1 thicknesses appears to increase with time. With our training set intentionally biased towards thinner layers,

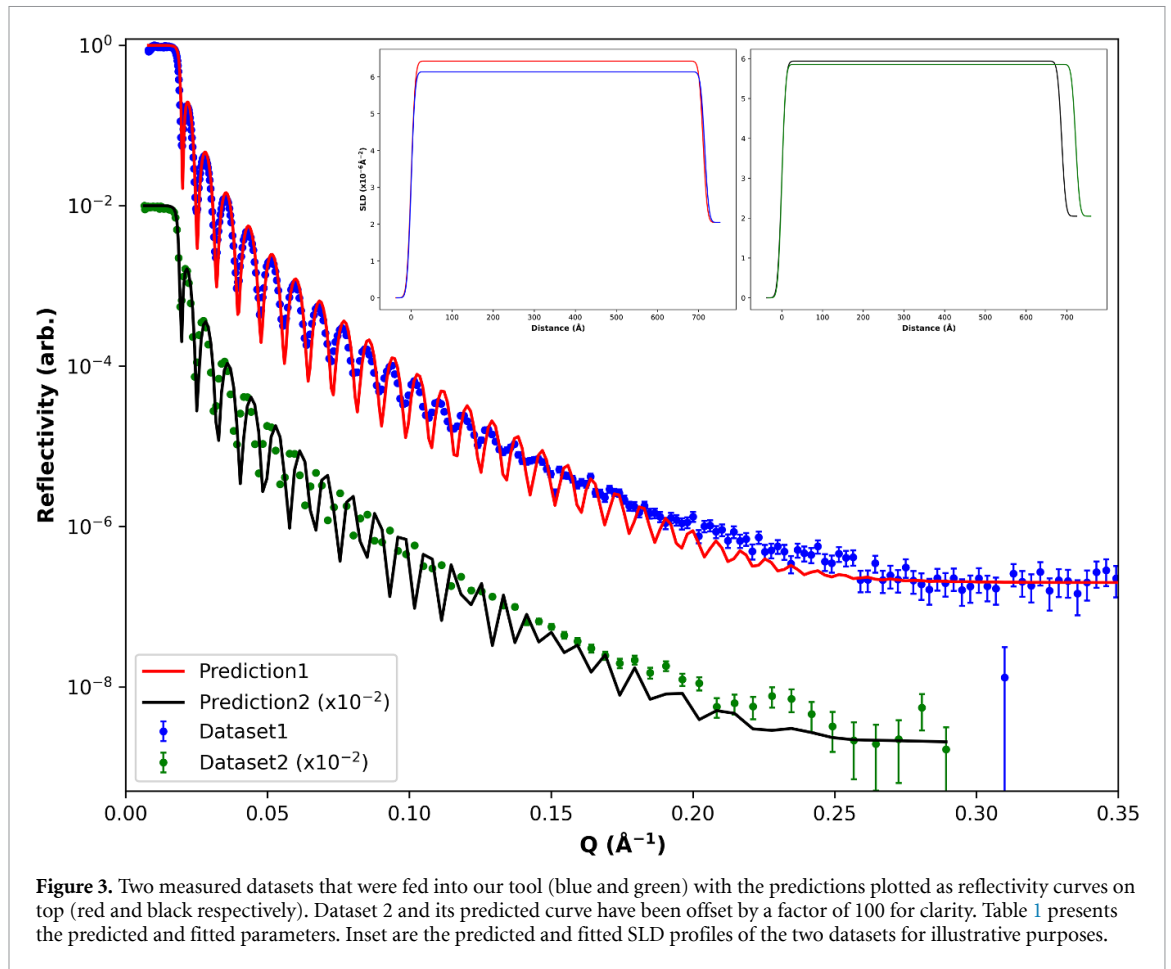


Figure 3. Two measured datasets that were fed into our tool (blue and green) with the predictions plotted as reflectivity curves on top (red and black respectively). Dataset 2 and its predicted curve have been offset by a factor of 100 for clarity. Table 1 presents the predicted and fitted parameters. Inset are the predicted and fitted SLD profiles of the two datasets for illustrative purposes.

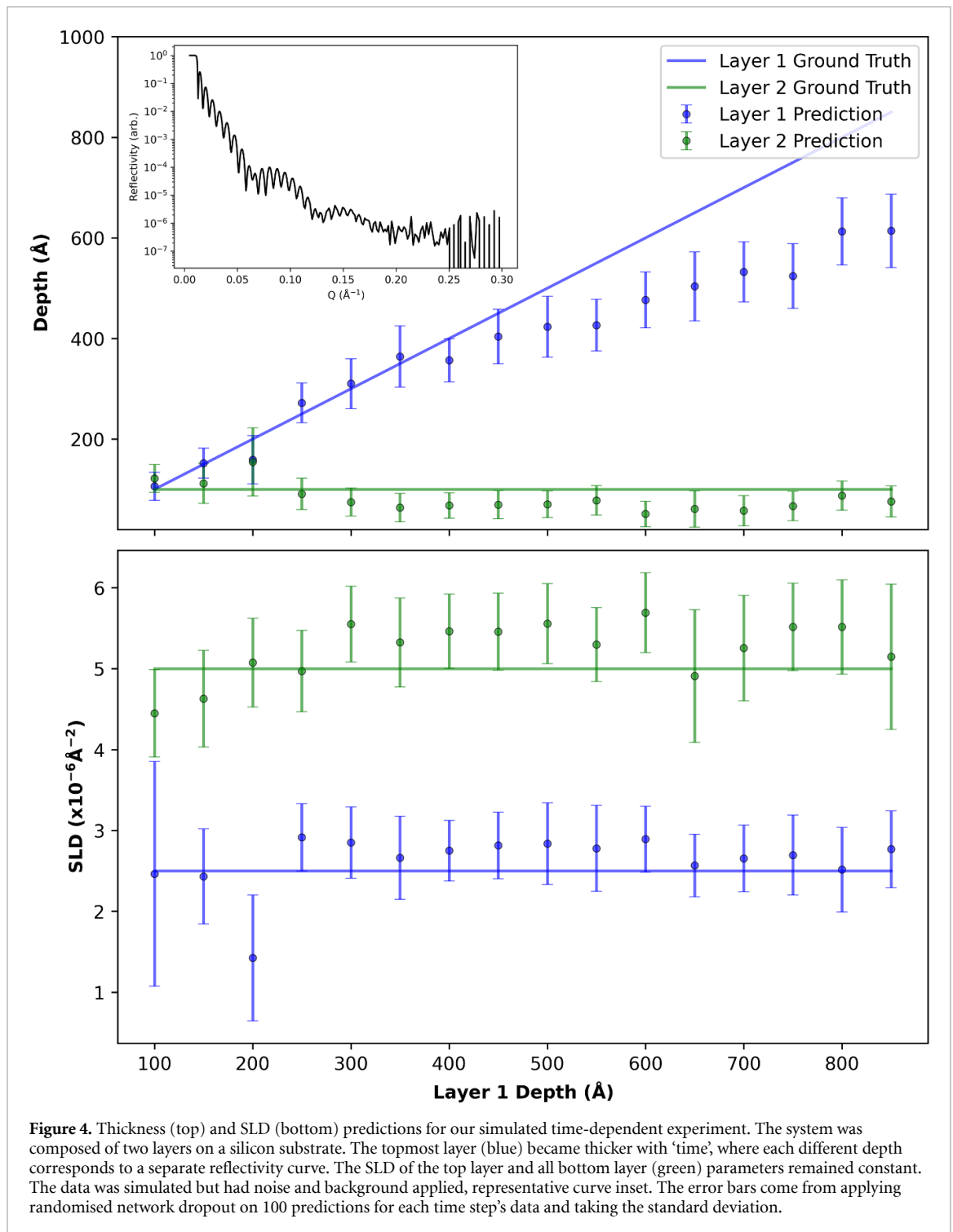
Table 1. Comparison of predicted and fitted values for real data passed through the neural network.

Dataset	SLD ($\times 10^{-6} \text{ \AA}^{-2}$)		Thickness (\AA)	
	Predicted	Fitted	Predicted	Fitted
(1)	6.4(3)	6.138(1)	712(45)	715.1(1)
(2)	5.9(3)	5.864(1)	690(37)	721.7(1)

this discrepancy at higher thicknesses is to be expected due to fewer thick layer samples encountered during training. Despite this, feeding these predictions into `refnx`'s fitting routine gives perfectly fitted data in a matter of seconds. This again demonstrates the utility of the approach, as in an experiment where large changes occur with time, it is often difficult for manual data analysis to keep up with the live data. Our tool allows the analysis to happen in very close to real time, and with no additional effort.

In addition to this work, we also used our trained network and applied transfer learning to see how well the ML networks would perform with x-ray reflectivity data. For the regressor trained on two-layer samples, the transfer learning took ~ 1 h on a desktop computer with GPU available, and resulted in an average mean squared error on normalised target values of 0.0068 after 20 epochs of retraining. This compares well with the neutron results, whose error was 0.0056, showing that our tool is well suited to other reflectivity applications, and is easily transferrable.

Finally, it is worth remarking on our choice to predict only up to three layers, and all on silicon. These choices were made for many of the reasons that we do not wish to replace the detailed analysis by the expert users. For an initial fit, if a reflectivity curve can be relatively well described by one, two or three layers, then there is little need to add more complexity to fit all the subtler details of the curve, even if the final answer will have more. For this application at least, we decided that predicating up to three layers with a good guess would have more utility than predicting four (or more) layers, with less information about those layers. The choice of a silicon substrate comes from this both acting to prove the method, as well as being the most common substrate choice for neutron reflectivity. Small amounts of retraining would allow any other substrate choice to be learnt.



4. Conclusions

We have created a neural network-based pipeline that is able to quickly predict the scattering lengths and thickness of up to three-layer systems measured using neutron reflectivity solely from being shown the reflectivity curves. These predictions are not fits but instead form the input to a reflectivity fitting program that is able to refine the initial prediction, using conventional search procedures. Our pipeline gives users semi-fitted data in a few seconds with no input, and massively lowers the barrier to entry for fitting the data immediately. Using a small amount of transfer learning, the models were also able to predict SLD profiles from x-ray reflectivity curves with similar accuracies. We view this tool as an extension of automatic data reduction on beamlines, taking data from an instrument and putting it into a form that is useful for the end users. The code for this work is open and available on GitHub ([neutron-net 2020](#)).

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

The authors would like to thank Johnathan Xue, whose initial work and results allowed the project to go ahead. We thank the Ada Lovelace Centre and ISIS neutron and muon source for year in industry placement student funding.

This work was partially supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1, particularly the 'AI for Science' theme within that grant and The Alan Turing Institute.

ORCID iDs

James H Durant  <https://orcid.org/0000-0001-8489-7860>

Joshaniel F K Cooper  <https://orcid.org/0000-0002-4635-2381>

References

- Arnold O et al 2014 Mantid—data analysis and visualization package for neutron scattering and μ SR experiments *Nucl. Instrum. Methods Phys. Res. A* **764** 156–66
- Baxter J 1997 A Bayesian/information theoretic model of learning to learn via multiple task sampling *Mach. Learn.* **28** 7–39
- Björck M and Andersson G 2007 GenX: an extensible x-ray reflectivity refinement program utilizing differential evolution *J. Appl. Crystallogr.* **40** 1174–8
- Bragg W H and Bragg W L 1913 The reflection of x-rays by crystals *Proc. R. Soc. A* **88** 428–38
- Carmona-Loaiza J M and Raza Z 2020 Towards reflectivity profile inversion through artificial neural networks (arXiv:2010.07634) (Accessed: 5 January 2021)
- Caruana R A 1993 Multitask learning: a knowledge-based source of inductive bias *Machine Learning Proc.* 1993 pp 41–48
- Dozat T 2016 Incorporating Nesterov Momentum into ADAM *Workshop Track-ICLR 2016* (Accessed: 25 February 2021)
- Gal Y and Ghahramani Z 2015 Dropout as a Bayesian approximation: representing model uncertainty in deep learning *33rd Int. Conf. Machine Learning, ICML 2016* (International Machine Learning Society (IMLS)) pp 1050–59 (available at: <http://proceedings.mlr.press/v48/gal16.html>) (Accessed: 25 February 2021)
- Greco A et al 2019 Fast fitting of reflectivity data of growing thin films using neural networks *J. Appl. Crystallogr.* **52** 1342–7
- Hafner D et al 2018 Noise contrastive priors for functional uncertainty (arXiv:1807.09289) (Accessed: 3 June 2020)
- Hafner D et al 2019 Reliable uncertainty estimates in neural networks using noise contrastive priors *35th Conf. Uncertainty in Artificial Intelligence, UAI 2019* (Accessed: 25 February 2021)
- Kienzle P et al 2020 Refl1D: Neutron and X-Ray Reflectivity Analysis Release 0.8.13 (Accessed: 5 January 2021)
- LeCun Y A et al 2012 Efficient BackProp *Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 9–48
- Mironov D, Durant J H and Cooper J F K 2020 neutron-net (available at: <https://github.com/xmironov/neutron-net>) (Accessed: 25 June 2020)
- Nelson A J and Prescott S W 2019 Refnx: neutron and x-ray reflectometry analysis in python *J. Appl. Crystallogr.* **52** 193–200
- Open Reflectivity Standards Organisation (ORSO) 2021 (available at: <https://www.reflectometry.org>) (Accessed: 7 January 2021)
- Parratt L G 1954 Surface studies of solids by total reflection of x-rays *Phys. Rev.* **95** 359
- Ruder S 2017 An overview of multi-task learning in deep neural networks (available at: <http://sebastianruder.com/multi-task/index>) (Accessed: 15 June 2020)
- Sivia D S 2011 *Elementary Scattering Theory: For X-ray And Neutron Users* 1st edn (Oxford: Oxford University Press)
- Srivastava N et al 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58