Eur. Phys. J. C (2020) 80:12
https://doi.org/10.1140/epjc/s10052-019-7562-1

THE EUROPEAN PHYSICAL JOURNAL C

Special Article - Tools for Experiment and Theory

DeepXS: fast approximation of MSSM electroweak cross sections at NLO

Sydney Otten[1,2,a], **Krzysztof Rolbiecki**[3], **Sascha Caron**[1,4], **Jong-Soo Kim**[5,6], **Roberto Ruiz de Austri**[7], **Jamie Tattersall**[8,9]

[1] Institute for Mathematics, Astro- and Particle Physics IMAPP, Radboud Universiteit, Nijmegen, The Netherlands
[2] GRAPPA, University of Amsterdam, Amsterdam, The Netherlands
[3] Faculty of Physics, University of Warsaw, Warsaw, Poland
[4] Nikhef, Amsterdam, The Netherlands
[5] Mandelstam Institute for Theoretical Physics, University of the Witwatersrand, Johannesburg, South Africa
[6] National Institute for Theoretical Physics, University of the Witwatersrand, Johannesburg, South Africa
[7] Instituto de Fisica Corpuscular, IFIC-UV/CSIC University of Valencia, Valencia, Spain
[8] Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University, Aachen, Germany
[9] ESR Labs, Munich, Germany

Received: 6 June 2019 / Accepted: 16 December 2019 / Published online: 7 January 2020

Abstract We present a deep learning solution to the prediction of particle production cross sections over a complicated, high-dimensional parameter space. We demonstrate the applicability by providing state-of-the-art predictions for the production of charginos and neutralinos at the Large Hadron Collider (LHC) at the next-to-leading order in the phenomenological MSSM-19 and explicitly demonstrate the performance for $pp \to \tilde{\chi}_1^+\tilde{\chi}_1^-$, $\tilde{\chi}_2^0\tilde{\chi}_2^0$ and $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$ as a proof of concept which will be extended to all SUSY electroweak pairs. We obtain errors that are lower than the uncertainty from scale and parton distribution functions with mean absolute percentage errors of well below 0.5 % allowing a safe inference at the next-to-leading order with inference times that improve the Monte Carlo integration procedures that have been available so far by a factor of $\mathcal{O}(10^7)$ from $\mathcal{O}(\text{min})$ to $\mathcal{O}(\mu\text{s})$ per evaluation.

1 Introduction

Dimensionality persists to be a curse for everyone that seeks the needle in a complex haystack. Despite all the achievements from data science so far, physicists often resort to simplified, lower-dimensional models to obtain a tractable problem [4,6]. This strategy prevents the scientific community from utilising all the available information to pin down the laws of nature. To overcome this very general issue, we investigate deep learning techniques as a potential solution motivated by the successful application of neural networks to cross sections with a four-dimensional parameter space [14]. We find positive results for cross sections that depend on a 19-dimensional parameter space with highly complex structures.

One of the most widely studied beyond the Standard Model (BSM) theories remains supersymmetry (SUSY) [20,25,31,32]. The ever increasing sophistication of experimental analyses requires that theoretical tools match the precision requirements set by experiments. One of the key requirements is performing cross section calculations of BSM processes at least at the next-to-leading order (NLO) accuracy. This goal has been gradually reached over many years, for particles produced both by strong and weak interactions, and the current state-of-the-art calculations also include resummed higher order corrections [11,22]. Currently, for most applications it is possible and sufficient to calculate the production cross section at the next-to-leading-log approximation. However, such calculations are typically time consuming, e.g. it takes about three minutes for the computer program `Prospino` [10] to calculate the chargino pair production cross section, $pp \to \tilde{\chi}_1^+\tilde{\chi}_1^-$, at NLO. The computational time for `Resummino` [23] is similar at NLO but taking into account higher order corrections increases the time consumption 20-fold.

Many applications, for example global scans of the multi-dimensional parameter space of the Minimal Supersymmetric Standard Model (MSSM), see e.g. [7–9,15,28,39], demand a much faster method for the computation of NLO

[a] e-mail: Sydney.Otten@ru.nl

cross sections. In the case of strongly produced SUSY particles, this problem is addressed by the computer program `NNLLfast` [12] that offers an approximation of relevant cross sections within a fraction of a second at the next-to-next-to-leading logarithmic accuracy.

In this paper we present a novel approach that enables a fast approximation of cross sections in a high-dimensional parameter space and as an example demonstrate the applicability for chargino and neutralino[1] production cross sections at NLO accuracy in the phenomenological MSSM-19 (pMSSM-19), where 19 denotes the number of model parameters. We employ a machine learning technique to approximate the result from the cross sections calculated using `Prospino`. While the task might appear straightforward, there are several challenges that one has to solve to obtain a tool that provides both speed and high accuracy. Firstly, the cross sections span over up to 13 orders of magnitude, depending on the electroweakino masses and couplings. Secondly, the electroweakino sector is parametrized by four independent parameters in the SUSY Lagrangian and, in addition, the cross sections depend on the other SUSY particles, either at the tree-level (squarks) or at the loop level (gluinos).

At the parton level, chargino and neutralino production occurs via s-channel exchange of gauge bosons and t-channel exchange of squarks. In case of the chargino production, this includes γ and Z exchange in the s-channel and left-handed (doublet) squark exchange in the t-channel. For neutralino pair production we have contributions from the Z boson exchange[2] and both left- and right-handed squarks. Finally, the associated production of a chargino and a neutralino occurs via exchange of the W boson in s-channel and left-handed squark exchange in the t-channel. At the loop level, when one considers SUSY-QCD contributions (i.e. the first order in the strong coupling α_s) there appear contributions involving gluinos. In Fig. 1 we show sample diagrams at the born and loop level. For more details see Ref. [10]. In the final step to calculate the production cross section in proton-proton collisions, the partonic cross section has to be convoluted with a parton distribution function (PDF) which parametrizes proton in terms of its constituents: quarks and gluons. Thus the final result cannot be given in the analytical form.

For the actual calculation of the cross section one needs to specify the final state particles and their physical masses, masses of the virtual particles (squarks and gluinos) and the mixing angles in the chargino and neutralino sectors. The

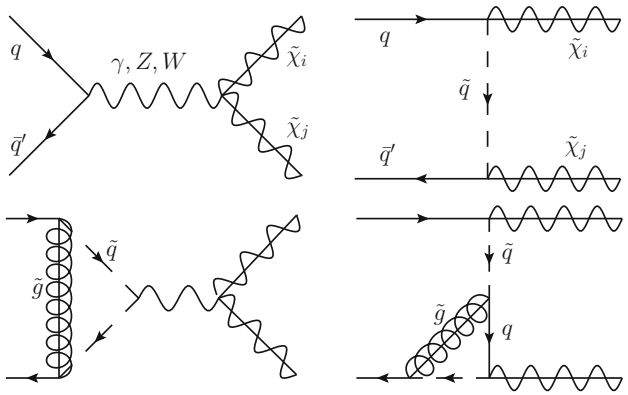


Fig. 1 Sample Feynman diagrams of electroweakino production at the tree level (upper row) and loop level (lower row)

Table 1 Summary of parameters required for each production process of charginos and neutralinos

Param.	$pp \to$		
	$\tilde{\chi}_1^+\tilde{\chi}_1^-$	$\tilde{\chi}_2^0\tilde{\chi}_2^0$	$\tilde{\chi}_1^\pm\tilde{\chi}_2^0$
Chargino mixing: U_{1i}, V_{1i}	4	–	4
Neutralino mixing: N_{2i}	–	4	4
Chargino mass: $m_{\tilde{\chi}_1^+}$	1	–	1
Neutralino mass: $m_{\tilde{\chi}_2^0}$	–	1	1
Doublet squark: $m_{\tilde{u}L}$	2	2	2
Singlet squark: $m_{\tilde{u}R}, m_{\tilde{d}R}$	–	2	–
Gluino: $m_{\tilde{g}}$	1	1	1
Total	8	10	13

chargino mixing is parametrized by two 2×2 unitary matrices U and V, while for the neutralino mixing it is a 4×4 unitary matrix N. For each process only one specific row of the matrices is required, corresponding to the respective final state particle. We summarize the number of required parameters for each process in Table 1. Note that in the following we do not explicitly impose the unitarity condition, which gives us a flexibility to extend the approach to other SUSY models beyond MSSM.

Thus, we need to construct representations for complicated functions whose effective parameter space that one has to cover can have up to 13 dimensions. A temperature plot showing the non-trivial K-factor landscape in only two of these dimensions is shown in Fig. 2 for $\tilde{\chi}_2^0\tilde{\chi}_1^+$. Here, we focus on the four most relevant processes at the LHC, i.e. the production of chargino pairs, $\tilde{\chi}_1^+\tilde{\chi}_1^-$, neutralino pairs, $\tilde{\chi}_2^0\tilde{\chi}_2^0$, and associated production of a chargino and a neutralino, $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$. The approach that we present here can be extended to other electroweak processes and models, e.g. the next-to-MSSM scenarios [21].

[1] Charginos and neutralinos are supersymmetric partners of the Standard Model gauge and Higgs bosons. In the following we will use the umbrella term "electroweakinos".

[2] At the loop level, photon exchange in the s-channel is also possible, however this electroweak correction is not included in `Prospino`.

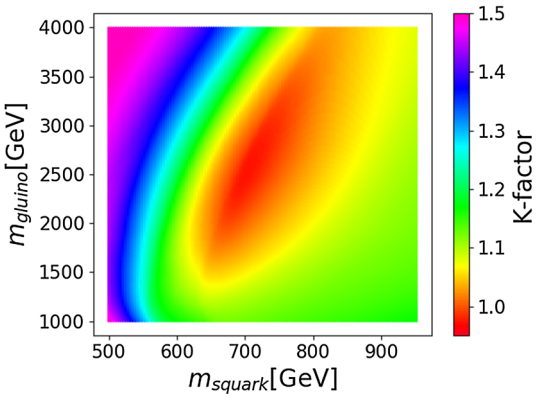


Fig. 2 A temperature plot for the K-factor in the wino scenario, predicted by a neural network, for $\tilde{\chi}_2^0\tilde{\chi}_1^+$ in the $m_{\tilde{q}}/m_{\tilde{g}}$ plane, already showing a non-trivial K-factor landscape for two free parameters. The electroweakino masses are set to 400 GeV

2 Methodology

In order to develop a code that can predict values of an otherwise computationally expensive function fast and reliably - the cross section in our case - we take the following approach. First we calculate values of the function at a large number of points, 10^7 samples at the leading order (LO) and $\mathcal{O}(10^5)$ samples at NLO. The points are sampled randomly in a high-dimensional parameter space in given ranges. The data is then used to train a customised artificial neural network (ANN), which is adopting deep learning techniques, stacking and an iterative ANN-based point selection procedure that picks points from a labeled pool of samples. The properly trained model is then able to provide accurate predictions of the cross section at a given parameter point. The performance of the resulting ANN is tested with 10^4 samples that the deep network has never seen during training.

2.1 Data generation

The pMSSM-19 parameters are sampled with a flat prior within the ranges given in Table 2, see also Ref. [2]. Since sleptons do not affect the actual calculation of the cross section at any stage they are assumed to be mass degenerate between left and right-handed states for the first and second generations. These parameter sets are then passed to `SPheno 3.38~` [36,37] to calculate the spectrum with default settings. For further processing we accept the points which have: no tachyonic degrees of freedom; the lightest neutralino as the LSP; the first two generations of squarks heavier than 500 GeV, cf. [1]; chargino $\tilde{\chi}_1^\pm$ heavier than 100 GeV, cf. [40]. They are then fed into `Prospino 2.1~`, which calculates the cross section using CTEQ6 parton distribution functions (PDFs) [33,38]. Note that even though the scan is performed in terms of the soft SUSY breaking parameters, the actual input for the cross section calculations will be defined via physical masses and mixing angles. Thus, the relevant masses and mixing angles from the spectrum with the corresponding LO cross section and/or K-factor are systematically collected so that they can be used to optimise an ANN implementation as training and validation data. For all LO cross sections, we have created 10^7 samples. For the K-factors, the number of generated samples varies between 1–6 $\times$ 10^5, for reasons explained in the following.

Table 2 Variable input parameters of the ATLAS pMSSM scan and the range over which these parameters are scanned

Parameter	Description	Scanned range
$m_{\tilde{L}_1} = m_{\tilde{E}_1}$	$1^{st}/2^{nd}$ gen. $SU(2)$ doublet/singlet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{L}_3}$	3^{rd} gen. $SU(2)$ doublet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{E}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking slepton mass	[90 GeV, 4 TeV]
$m_{\tilde{Q}_1}$	$1^{st}/2^{nd}$ gen. $SU(2)$ doublet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{U}_1}$	$1^{st}/2^{nd}$ gen. $SU(2)$ singlet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{D}_1}$	$1^{st}/2^{nd}$ gen. $SU(2)$ singlet soft breaking squark mass	[200 GeV, 4 TeV]
$m_{\tilde{Q}_3}$	3^{rd} gen. $SU(2)$ doublet soft breaking squark mass	[100 GeV, 4 TeV]
$m_{\tilde{U}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking squark mass	[100 GeV, 4 TeV]
$m_{\tilde{D}_3}$	3^{rd} gen. $SU(2)$ singlet soft breaking squark mass	[100 GeV, 4 TeV]
A_t	Stop trilinear coupling	[−8 TeV, 8 TeV]
A_b	Sbottom trilinear coupling	[−4 TeV, 4 TeV]
A_τ	Stau trilinear coupling	[−4 TeV, 4 TeV]
$\lvert\mu\rvert$	Higgsino mass parameter	[80 GeV, 4 TeV]
$\lvert M_1\rvert$	Bino mass parameter	[0 TeV, 4 TeV]
$\lvert M_2\rvert$	Wino mass parameter	[70 GeV, 4 TeV]
M_3	Gluino mass parameter	[200 GeV, 4 TeV]
M_A	Pseudoscalar Higgs mass	[100 GeV, 4 TeV]
$\tan\beta$	Ratio of vacuum expectation values	[1, 60]

The NLO cross section can be written as a product of the K-factor and the LO cross section:

$$\sigma_{\text{NLO}} = K \cdot \sigma_{\text{LO}}. \tag{1}$$

Since most difficulties in the structure already appear at the leading order and the K-factor is a slowly varying function of the input parameters, we construct the NLO prediction by multiplying the predictions of the LO and K-factor regressors. This significantly decreases the computational cost by

reducing the amount of necessary NLO data by two orders of magnitude.

Because the starting point of the data generation is the pMSSM-19 parameter space, we must cover it appropriately and therefore we are confronted with the curse of dimensionality. To tackle this, we manually restrict the parameter space by excluding all cross sections that are not relevant. By exploiting the fact that the number of events N at the LHC is equal to the product of the integrated luminosity and the cross section:

$$N = L_{\text{int}} \cdot \sigma, \tag{2}$$

and assuming the final integrated luminosity of the LHC to be $L_{\text{int}} = 3000\,\text{fb}^{-1}$, we can derive a lower bound for the cross section by demanding at least one event in the life-time of the LHC. The resulting lower bound is $\sigma_{\text{min}} = 3.3 \cdot 10^{-7}$ pb.

The generated data is then processed by a deep learning pipeline that utilises the ANN-based point selection (NNPS) and stacking, i.e. a manually implemented logical connection of different, eventually specialised predictors for the same task. Since we assume that most use-cases will run a spectrum calculator anyway and `SPheno` barely consumes computational capacity, we only create a neural network representation for the mapping from the masses and mixing angles to the LO cross section and K-factor.

2.2 Optimising the representations

The deep learning techniques used here are employed via artificial neural networks implemented with `Keras` [19] and a `Tensorflow` [3] backend that were trained on a GPU using `CUDA` [34] and `cuDNN` [18]. The pre- and post-processing of the input data together with the neural network architecture and the machine learning model parameters form the technical realization of the deep learning representation of the function $\sigma = \sigma(\text{pMSSM-19})$.

The input of the neural networks is taken from the `SPheno` output and consists of the electroweakino and squark masses for the LO cross sections, and gluino mass for the K-factor, as well as the relevant chargino and neutralino mixing matrix entries, and is preprocessed via the z-score normalisation: the inputs x_i are transformed into $x_i' = \frac{x_i - \mu(x)}{\sigma_{\text{sd}}(x)}$, where $\mu(x)$ and $\sigma_{\text{sd}}(x)$ are the mean and standard deviation of x. Whenever deemed useful, expert knowledge was applied and high-level features were formed, e.g. for the K-factor prediction, the mean of the squark masses was used, which corresponds to the calculation method employed in `Prospino`.

An ANN is a collection of artificial neurons, along whose connections an input is propagated. During the propagation the input is transformed depending on the network architecture and the machine learning model parameter set θ characterising the ANN. The output is an estimate of the function value for the given input parameters. The set θ is initially drawn from a random distribution and learned via updates from a stochastic gradient descent-like optimisation algorithm that minimises a loss function which measures the deviation between model predictions and true (known) cross sections. In our case, the loss function is the mean absolute percentage error

$$\text{MAPE} = \frac{1}{N} \sum_{i=0}^{N} \left| \frac{y_{\text{true},i} - y_{\text{pred},i}}{y_{\text{true},i}} \right|, \tag{3}$$

which is minimised. The chosen optimiser is `ADAM` [29] with default parameters except a learning rate scheduling with initial and final learning rates α_i and α_f combined with `EarlyStopping` [13]. When an iteration has ended, i.e. when either the pre-defined maximum number of epochs is reached or `EarlyStopping` terminates the process, the learning rate is divided by 2, the weights giving the best validation loss so far are loaded into the architecture and the optimisation continues until α_f is reached.

Due to the high computational cost of hyperparameter scans, i.e. many hours to days for a single hyperparameter point and the fact that the hyperparameter space is high-dimensional with a mixture of integer and continuous dimensions, we choose a heuristic approach to determine the hyperparameters. For different processes of electroweakino production we therefore adapt different techniques to achieve MAPEs below 0.5 % and maximum errors of below 10 %.

The σ_{LO} input is propagated through eight hidden layers with 100 neurons each and the `selu` [30] activation function, while for the K-factors only 32 neurons per layer are used. Note that the neural capacity of this network is low when compared to state of the art deep learning architectures [27]. However, an even lower capacity also delivered reasonable predictions but the drawbacks were that (a) training took much longer until its best performance was reached and (b) the best performance itself was worse. We chose the architecture following the suggestions for self-normalizing neural networks [30] that were specifically developed to obtain state of the art neural network models for regression and classification problems. One of its big advantages is that the self-normalisation allows for gradients in the deeper layers that have the same order of magnitude as in the first layers which enables the ML model to learn more abstract features. The inputs are labeled with the corresponding cross section, in most cases pre-processed with a shifted logarithm such that for the input x_i its label is given by

$$y_i' = -\min(\log(\sigma)) + \log(\sigma_i) \tag{4}$$

or for the K-factor divided by 2 or 4, depending on the pair. The loss function is the MAPE for the K-factors and

a modification thereof for the LO cross sections that takes into account the pre-processing. In the default setup, the MAPE would minimise the error on $\log(\sigma)$ and it would result in sub-optimal performance. Therefore, several custom loss functions have been implemented that are constructed such that the loss function explicitly minimises the MAPE of the original values. The final data for the LO cross sections has been trained for 150 epochs and 7–10 iterations with $\alpha_i = 0.0008$, a patience of 50 and a batch-size of 120. For the K-factors we used larger batch-sizes of between 512 and 1024 and a larger number of epochs of up to 250 per iteration. For $\tilde{\chi}_1^+\tilde{\chi}_1^-$ and $\tilde{\chi}_2^0\tilde{\chi}_2^0$ we take all of the randomly generated samples and train deep networks for the LO and the K-factors.

For $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$ we extend our setup to include NNPS because of a recurring problem of outliers with large errors in problematic, often underpopulated, regions of the parameter space even with 10^7 training samples. NNPS allowed us to have a much better performance with only a fraction of the random samples. The NNPS setup was as follows: the initial training begins with 10^6 ($\tilde{\chi}_2^0\tilde{\chi}_1^+$) or $1.5\cdot10^6$ ($\tilde{\chi}_2^0\tilde{\chi}_1^-$) samples and runs for a short amount of time, namely 40 epochs, 5 iterations and a batch-size of 1000. The resulting neural network is then evaluated on the pool of the remaining $9\cdot10^6$ samples. The 10^5 or $1.5\cdot10^5$ samples for which the neural network performed worst are iteratively added to the training set. This is done 10 and 15 times respectively. The actively sampled training set is then used for a more thorough training identical to the procedure used for $\tilde{\chi}_1^+\tilde{\chi}_1^-$ and $\tilde{\chi}_2^0\tilde{\chi}_2^0$. The evaluation of these networks is then investigated and in both cases the performance is further enhanced by training additional neural networks that are specialised on a fraction of the target value range. For a specialised network covering target values in the range 0.001 and 0.2 pb for $\tilde{\chi}_2^0\tilde{\chi}_1^-$, we also z-score transformed the target values without taking the logarithm. The general and specialised networks are then stacked together by using the general network to predict whether a point is predicted better by the specialised network: if that is the case, the prediction of the specialised network is returned, if not, the general network will return its prediction. The $\tilde{\chi}_2^0\tilde{\chi}_1^+$ K-factors have been treated similarly, while for the $\tilde{\chi}_2^0\tilde{\chi}_1^-$ we only used one neural network trained on random samples.

3 Results

In this section we present the accuracy of the tool, `DeepXS`, including statistical measures of its performance. We also discuss inference times and subtleties of its validity. The testing of `DeepXS` was performed using 10^4 pMSSM-19 points generated according to the same rules as the training samples.

Table 3 shows the performance of `DeepXS` for the cross sections σ_{NLO} that are larger than the threshold $\sigma_{\text{exp}} = 6.6\cdot10^{-5}$ pb. This threshold corresponds to the integrated luminosity 150 fb^{-1}, which is the data collected thus far by the LHC, and assuming 10 produced events. Therefore, for current applications this threshold provides a very conservative estimate of the observable electroweakino production. The entries for 1σ, 2σ and 3σ denote the maximum error for 68.27 %, 95.45 % and 99.73 % of the samples. We use the intervals as defined for the normal distribution motivated by the shape of the error distribution. However, we note that it has fatter tails.

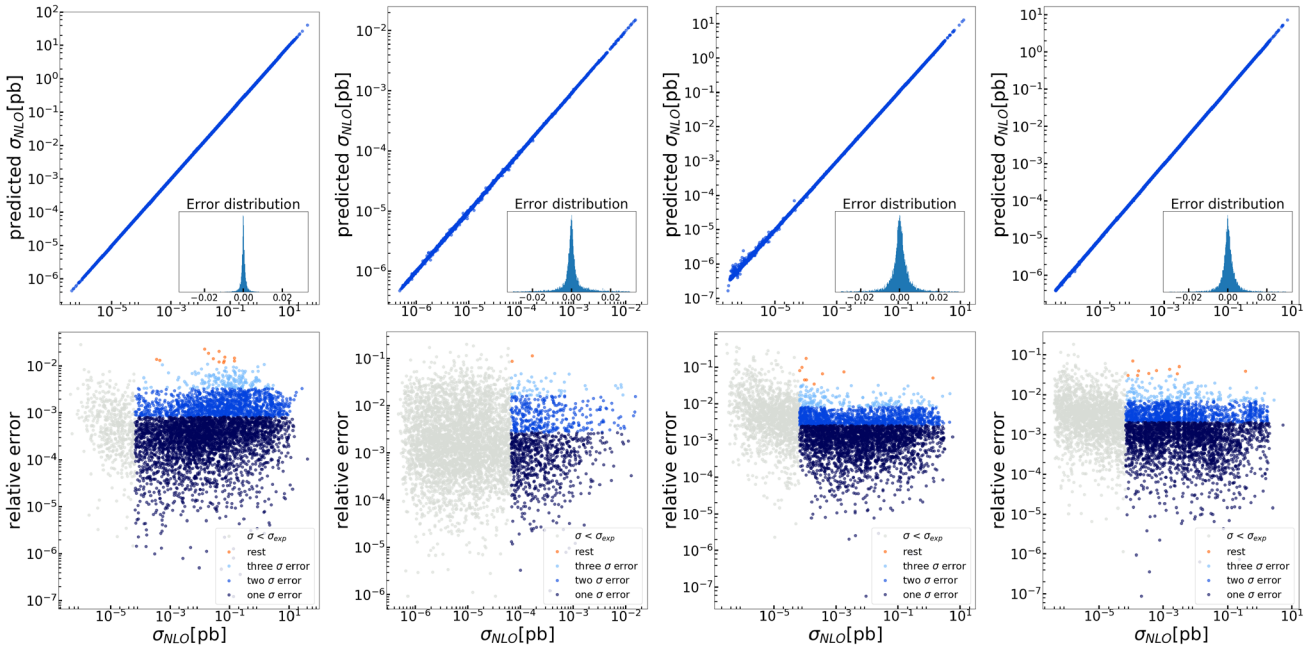
Table 3 Relative error bands and MAPE for the NLO predictions for $\sigma_{\text{NLO}} \geq \sigma_{\text{exp}} = 6.6\cdot10^{-5}$ pb

Pair	MAPE	1σ	2σ	3σ
$\tilde{\chi}_1^+\tilde{\chi}_1^-$	0.091 %	0.081 %	0.334 %	1.195 %
$\tilde{\chi}_2^0\tilde{\chi}_2^0$	0.384 %	0.274 %	1.652 %	5.773 %
$\tilde{\chi}_2^0\tilde{\chi}_1^+$	0.263 %	0.258 %	0.822 %	3.299 %
$\tilde{\chi}_2^0\tilde{\chi}_1^-$	0.214 %	0.206 %	0.701 %	3.035 %

With all MAPEs being well below 0.5 % and a maximum error of the 3σ bands of 5.773 %, the error of the cross sections clearly is sub-dominant relative to scale and PDF uncertainty for a large majority of the presented cases. Figure 3 demonstrates a large density of points around an error of 10^{-4}–10^{-2}, which matches the precision of Vegas integration, $5\cdot10^{-3}$, typically reported by `Prospino`. For $\tilde{\chi}_1^+\tilde{\chi}_1^-$, the maximum error on the 10^4 test samples is ≈ 3 % while for the other pairs it is $\mathcal{O}(10\,\%)$. We note that this size of uncertainty is otherwise expected to arise due to PDF and scale variation which starts with 3–4 % for high cross sections and rises to $\mathcal{O}(10\,\%)$ for high masses. The largest error is observed for two samples with an error of $\approx 10\%$ for $\tilde{\chi}_2^0\tilde{\chi}_2^0$ and $\tilde{\chi}_2^0\tilde{\chi}_1^+$.

The case of $\tilde{\chi}_2^0\tilde{\chi}_2^0$ will be improved with NNPS in the upcoming version of the tool. Note that the dimensionality of $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$, $d = 11$ at LO and $d = 12$ at NLO, is much higher than for $\tilde{\chi}_2^0\tilde{\chi}_2^0$, with dimensionality 6 and 7 respectively. When training $\tilde{\chi}_2^0\tilde{\chi}_1^\pm$ on 10^7 random samples, the predictions were much worse than they are currently for $\tilde{\chi}_2^0\tilde{\chi}_2^0$. We thus expect that NNPS will further improve the $\tilde{\chi}_2^0\tilde{\chi}_2^0$ to the same level of precision we have achieved for the other pairs. That the overall accuracy for $\tilde{\chi}_2^0\tilde{\chi}_1^-$ is better than for $\tilde{\chi}_2^0\tilde{\chi}_1^+$ is due to the more thoroughly performed NNPS, which will thus be the standard for future work.

Below the threshold of σ_{exp}, our predictions also have a MAPE of below 1% with a maximum of 0.81 % for $\tilde{\chi}_2^0\tilde{\chi}_2^0$, lower values of ≈ 0.3 % for the mixed pairs and ≈ 0.1 % for chargino pairs. Note however, that although errors above 10 % are more frequent for $\sigma \leq \sigma_{\text{exp}}$, the PDF uncertainty is typically also high in the corresponding region of the parameter space.



(a) $\tilde{\chi}_1^+\tilde{\chi}_1^-$ **(b)** $\tilde{\chi}_2^0\tilde{\chi}_2^0$ **(c)** $\tilde{\chi}_2^0\tilde{\chi}_1^+$ **(d)** $\tilde{\chi}_2^0\tilde{\chi}_1^-$

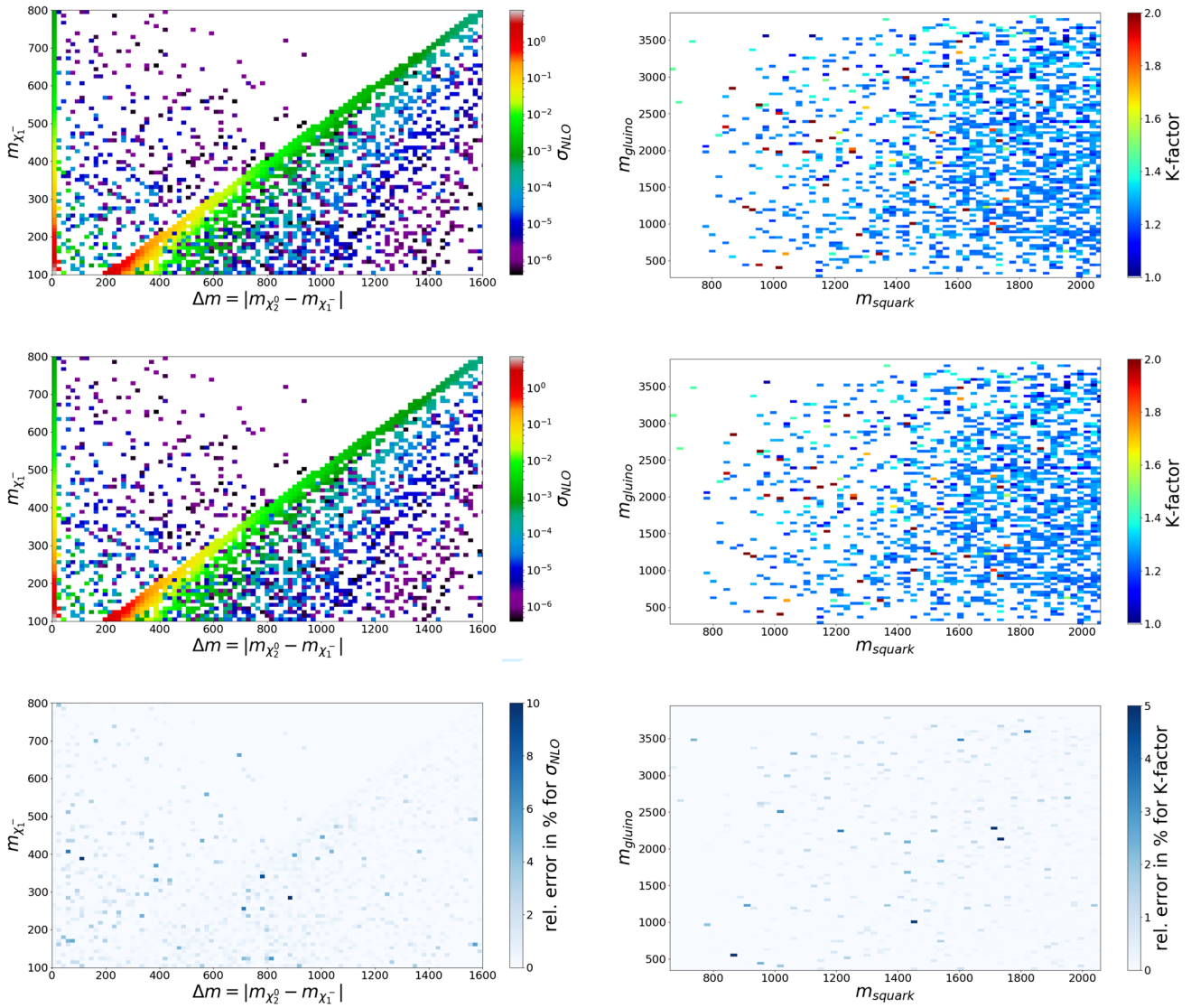
Fig. 3 The true vs. predicted NLO cross sections with histograms of the relative error (top) and the true NLO cross section vs. the relative error with confidence intervals (bottom), as defined in Sect. 3, for the test sample of 10^4 points

Figure 4 shows a comparison between the Prospino calculation and the DeepXS prediction including the relative errors for σ_{NLO} and the K-factor for $pp \to \tilde{\chi}_2^0\tilde{\chi}_1^-$. We observe that the neural networks predict the complicated cross section landscapes so well that the plots corresponding to the predictions and the Prospino calculations are indistinguishable by eye. Only the plot showing the relative errors reveal a handful of slight deviations of no more than $\mathcal{O}(10\%)$, consistent with Fig. 3. The plots were created using the same 10000 points as for Fig. 3.

DeepXS is interfaced to pySLHA [16] and can process SLHA2 files [5]. Additionally, a possibility to feed in the relevant parameters via .csv and .txt files has been implemented. When providing SLHA files, DeepXS needed 72.3 s to evaluate 10^4 samples or 7.23 ms per evaluation of $\tilde{\chi}_1^+\tilde{\chi}_1^-$ at LO and NLO, already making DeepXS $\mathcal{O}(10^4)$ faster than Prospino. When SLHA files are an input, DeepXS tests if $\tilde{\chi}_1^0$ is the LSP, if the light chargino mass is above 100 GeV and if the squark masses are above 500 GeV, and a warning is given if any of these conditions is not fulfilled. When text files with an array are provided, the inference of 10^7 $\tilde{\chi}_1^+\tilde{\chi}_1^-$ predictions both at LO and NLO took 261.51 seconds on an Intel i7-4790K CPU or $\approx 26\,\mu$s per evaluation, making it ≈ 6.9 million times faster than Prospino. When predicting mixed pairs, each evaluation takes slightly longer due to the stacking and the necessity to infer from more than two neural networks. In all cases, warnings are given when the predicted cross section is lower than σ_{exp}.

4 Conclusions

We presented a method that for the first time allows a fast and highly accurate approximation of cross sections that depend on a high-dimensional and complex parameter space. As the first application, we developed a novel tool, DeepXS, that enables a fast approximation of NLO cross sections for pMSSM-19 electroweakinos. Beside the incorporation of expert knowledge, it employs stacked artificial neural networks supplemented by ANN-based point selection techniques to provide fast predictions based on the full NLO calculation using Prospino. Compared to Prospino, DeepXS is more than 4 and up to 7 orders of magnitude faster, while ensuring an accuracy of 1 % for more than 95 % of the test points. Training the neural networks takes $\mathcal{O}(h)$ (≈ 1 for K-factors and ≈ 12 for the leading order). Note that modifications of the underlying physics model do not imply that one has to retrain starting from 0. Instead one can initialize the new ML model with the older optimum and minimize the loss function for the new case starting from there: in the machine learning literature this is a well-known and studied technique called transfer learning [41]. Should the precision requirements for supersymmetric cross sections at NLO evolve such that we need to eliminate the few remaining outliers with errors of $\mathcal{O}(10)\%$ we can do so by creating a larger pool of samples with an even more dedicated neural network point selection procedure. Additionally we can make use of ensemble techniques [26] to boost the performance



(a) $\sigma_{\mathrm{NLO}}(\Delta m, m_{\tilde{\chi}_1^-})$ **(b)** $K(m_{\tilde{q}}, m_{\tilde{g}})$

Fig. 4 `Prospino` (top), `DeepXS` (middle) and relative error (bottom) (**a**) for σ_{NLO} in the $\Delta m = |m_{\tilde{\chi}_2^0} - m_{\tilde{\chi}_1^-}|$ vs. $m_{\tilde{\chi}_1^-}$ plane; (**b**) for the K-factor in the $m_{\tilde{q}} = \frac{1}{8}\sum_{i=1}^{8} m_{\tilde{q}_i}$ vs. $m_{\tilde{g}}$ plane for $pp \to \tilde{\chi}_2^0 \tilde{\chi}_1^-$

and, although computationally very expensive, use Bayesian techniques to optimize the hyperparameters of our neural network models. To enable an uncertainty estimate for individual points, future architectures to regress cross sections should include Monte Carlo Dropout [24]: in each layer a fixed fraction of the neurons is randomly deactivated during training and inference. This procedure will lead to varying predictions for a fixed input, allowing to obtain a distribution with a mean and a standard deviation of the prediction per point. As is shown in [24], this procedure converges towards a Bayesian posterior, enabling a meaningful comparison of the uncertainty of the prediction with the PDF uncertainty. Until this exists, and although very conservative, one must rely on the error maps presented in Fig. 3 to estimate an uncertainty. The tool can be found in a GitHub repository [35] including examples that show the NNPS sampling strategy. Further development will include the completion of all electroweakino pairs, extensions of the MSSM, an estimation of scale and PDF uncertainties and a merge with `BSM-AI` [17].

Acknowledgements S. C. and S. O. thank the support by the Netherlands eScience Center under the project iDark: The intelligent Dark Matter Survey. R. RdA, thanks the support from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 674896, the "SOM Sabor y origen de la Materia" MEC projects and the Spanish MINECO Centro de Excelencia Severo Ochoa del IFIC program under grant SEV-

2014-0398. K. R. is supported by the National Science Centre, Poland, under grants DEC-2016/23/G/ST2/04301, 2015/18/M/ST2/00518 and 2015/19/D/ST2/03136.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The data is available on reasonable request.]

Funded by SCOAP[3].

References

1. Morad Aaboud et al., Search for squarks and gluinos in final states with jets and missing transverse momentum using 36 fb^{-1} of $\sqrt{s} =$ 13 TeV pp collision data with the ATLAS detector. Phys. Rev. D **97**(11), 112001 (2018)
2. Georges Aad et al., Summary of the ATLAS experiment's sensitivity to supersymmetry after LHC Run 1 - interpreted in the phenomenological MSSM. JHEP **10**, 134 (2015)
3. Martín Abadi et al. Tensorflow: A system for large-scale machine learning. in *Proceedings of the 12th USENIX conference on operating systems design and implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA. USENIX Association (2016)
4. Jalal Abdallah et al., Simplified models for dark matter searches at the LHC. Phys. Dark Univ. **9–10**, 8–23 (2015)
5. B.C. Allanach et al., SUSY les houches accord 2. Comput. Phys. Commun. **180**, 8–25 (2009)
6. Peter Athron et al., A global fit of the MSSM with GAMBIT. Eur. Phys. J. **C77**(12), 879 (2017)
7. Athron, Peter and Balázs, Csaba and Buckley, Andy and Cornell, Jonathan M. and Danninger, Matthias and Farmer, Ben and Fowlie, Andrew and Gonzalo, Tomás E. and Harz, Julia and et al. Combined collider constraints on neutralinos and charginos. Eur. Phys. J. C. https://doi.org/10.1140/epjc/s10052-019-6837-x (2019)
8. E. Bagnaschi et al., Likelihood analysis of the pMSSM11 in light of LHC 13-TeV data. Eur. Phys. J. **C78**(3), 256 (2018)
9. Philip Bechtle et al., Constrained supersymmetry after 2 years of LHC data: a global view with Fittino. JHEP **06**, 098 (2012)
10. W. Beenakker, M. Klasen, M. Kraemer, T. Plehn, M. Spira, P.M. Zerwas, The production of charginos / neutralinos and sleptons at hadron colliders. Phys. Rev. Lett. **83**, 3780–3783 (1999). (**Erratum: Phys. Rev. Lett.100,029901(2008)**)
11. Wim Beenakker, Christoph Borschensky, Michael Kraemer, Anna Kulesza, Eric Laenen, Vincent Theeuwes, Silja Thewes, NNLL resummation for squark and gluino production at the LHC. JHEP **12**, 023 (2014)
12. Wim Beenakker, Christoph Borschensky, Michael Krämer, Anna Kulesza, Eric Laenen, NNLL-fast: predictions for coloured supersymmetric particle production at the LHC with threshold and Coulomb resummation. JHEP **12**, 133 (2016)
13. Y. Bengio. Practical recommendations for gradient-based training of deep architectures. arXiv:1206.5533, (June 2012)
14. Gianfranco Bertone, Nassim Bozorgnia, Jong Soo Kim, Sebastian Liem, Christopher McCabe, Sydney Otten, Roberto Ruiz de Austri, Identifying WIMP dark matter from particle and astroparticle data. JCAP **1803**, 026 (2018)
15. Gianfranco Bertone, Francesca Calore, Sascha Caron, Roberto Ruiz, Jong Soo Kim, Roberto Trotta, Christoph Weniger, Global analysis of the pMSSM in light of the Fermi GeV excess: prospects for the LHC Run-II and astroparticle experiments. JCAP **1604**(04), 037 (2016)
16. Andy Buckley, PySLHA: a pythonic interface to SUSY les houches accord data. Eur. Phys. J. **C75**(10), 467 (2015)
17. Sascha Caron, Jong Soo Kim, Krzysztof Rolbiecki, Roberto Ruiz de Austri, Bob Stienen, The BSM-AI project: SUSY-AI–generalizing LHC limits on supersymmetry with machine learning. Eur. Phys. J. **C77**(4), 257 (2017)
18. S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer. cuDNN: efficient primitives for deep Learning. arXiv:1410.0759, (October 2014)
19. François Chollet et al. Keras. https://github.com/fchollet/keras
20. M. Drees, R. Godbole, P. Roy. Theory and phenomenology of sparticles: an account of four-dimensional $N = 1$ supersymmetry in high energy physics, (World Scientific, Hackensack, USA, 2004)
21. Ulrich Ellwanger, Cyril Hugonie, Ana M. Teixeira, The next-to-minimal supersymmetric standard model. Phys. Rept. **496**, 1–77 (2010)
22. Benjamin Fuks, Michael Klasen, David R. Lamprea, Marcel Rothering, Gaugino production in proton-proton collisions at a center-of-mass energy of 8 TeV. JHEP **10**, 081 (2012)
23. Benjamin Fuks, Michael Klasen, David R. Lamprea, Marcel Rothering, Precision predictions for electroweak superpartner production at hadron colliders with Resummino. Eur. Phys. J. **C73**, 2480 (2013)
24. Y. Gal, Z. Ghahramani. Dropout as a bayesian approximation: representing model uncertainty in deep learning, In: Proceedings of The 33rd International Conference on Machine Learning, vol. 48, pp 1050–1059 (2015)
25. Howard E. Haber, Gordon L. Kane, The search for supersymmetry: probing physics beyond the standard model. Phys. Rept. **117**, 75–263 (1985)
26. L.K. Hansen, P. Salamon, Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 993–1001 (1990)
27. K. He, X. Zhang, S. Ren, J. Sun. Identity mappings in deep residual networks, (2016)
28. Jong Soo Kim, Krzysztof Rolbiecki, Roberto Ruiz, Jamie Tattersall, Torsten Weber, Prospects for natural SUSY. Phys. Rev. **D94**(9), 095013 (2016)
29. D. P. Kingma, J. Ba. Adam, A Method for Stochastic Optimization. arXiv:1412.6980, (December 2014)
30. G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-Normalizing Neural Networks. arXiv:1706.02515, (June 2017)
31. Stephen P. Martin. A supersymmetry primer. Adv. Ser. Dir. High Energy Phys. https://doi.org/10.1142/9789812839657_0001
32. Stephen P. Martin, Adv. Ser. Direct. High Energy Phys. **18**, 1 (1998)
33. Pavel M Nadolsky, Hung-Liang Lai, Qing-Hong Cao, Joey Huston, Jon Pumplin, Daniel Stump, Wu-Ki Tung, C .P. Yuan, Implications of CTEQ global analysis for collider observables. Phys. Rev. **D78**, 013004 (2008)
34. John Nickolls, Ian Buck, Michael Garland, Kevin Skadron, Scalable parallel programming with cuda. Queue **6**(2), 40–53 (2008)
35. Sydney Otten et al. DeepXS. https://github.com/SydneyOtten/DeepXS. Accessed 3 Jan 2020
36. W. Porod, F. Staub, SPheno 3.1: extensions including flavour, CP-phases and models beyond the MSSM. Comput. Phys. Commun. **183**, 2458–2469 (2012)

37. Werner Porod, SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e+ e- colliders. Comput. Phys. Commun. **153**, 275–315 (2003)
38. J. Pumplin, D .R. Stump, J. Huston, H .L. Lai, Pavel M Nadolsky, W .K. Tung, New generation of parton distributions with uncertainties from global QCD analysis. JHEP **07**, 012 (2002)
39. C. Strege, G. Bertone, G.J. Besjes, S. Caron, R. Ruiz de Austri, A. Strubig, R. Trotta, Profile likelihood maps of a 15-dimensional MSSM. JHEP **09**, 081 (2014)
40. The LEP SUSY Working Group and the ALEPH, DELPHI, L3 and OPAL experiments note LEPSUSYWG/01-03.1. http://lepsusy.web.cern.ch/lepsusy. Accessed 3 Jan 2020
41. Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. How transferable are features in deep neural networks?, In: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (eds.), Advances in Neural Information Processing Systems 27, pp 3320–3328 (2014)