SCYNet: testing supersymmetric models at the LHC with neural networks

Philip Bechtle^{1,a}, Sebastian Belkner^{1,b}, Daniel Dercks^{2,c}, Matthias Hamer^{1,d}, Tim Keller^{3,e}, Michael Krämer^{3,f}, Björn Sarrazin^{3,g}, Jan Schütte-Engel^{3,h}, Jamie Tattersall^{3,i}

¹ Universität Bonn, Nussallee 12, Bonn, Germany

² Universität Hamburg, Luruper Chaussee 149, Hamburg, Germany

³ Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University, 52074 Aachen, Germany

Received: 14 March 2017 / Accepted: 16 September 2017 / Published online: 25 October 2017

© The Author(s) 2017. This article is an open access publication

Abstract SCYNet (SUSY Calculating Yield Net) is a tool for testing supersymmetric models against LHC data. It uses neural network regression for a fast evaluation of the profile likelihood ratio. Two neural network approaches have been developed: one network has been trained using the parameters of the 11-dimensional phenomenological Minimal Supersymmetric Standard Model (pMSSM-11) as an input and evaluates the corresponding profile likelihood ratio within milliseconds. It can thus be used in global pMSSM-11 fits without time penalty. In the second approach, the neural network has been trained using model-independent signature-related objects, such as energies and particle multiplicities, which were estimated from the parameters of a given new physics model.

1 Introduction

Direct searches for new particles at the LHC are among the most sensitive probes of beyond the Standard Model (BSM) physics and play a crucial role in global BSM fits. Calculating the profile likelihood ratio (referred to as χ^2 in the following) for a new physics model from LHC searches is straightforward in principle: for each point in the model parameter space, signal events are generated using a Monte Carlo sim-

ulation. χ^2 is then calculated from the number of expected signal events, the Standard Model background estimate and the number of observed events for a given experimental signal region. The computation time for such simulations, can be overwhelming however, especially when testing BSM scenarios with many model parameters. Global fits of supersymmetric (SUSY) models, for example, are typically based on the evaluation of $\mathcal{O}(10^9)$ parameter points, see e.g. [1–3], and the required Monte Carlo statistics for estimating the number of signal events for each parameter point requires up to several hours of CPU time. In this study we have attempted to provide a fast evaluation of the LHC χ^2 for generic SUSY models by utilizing neural network regression.

Global SUSY analyses which combine low-energy precision observables, like the anomalous magnetic moment of the muon, and LHC searches for new particles strongly disfavour minimal SUSY models, like the constrained Minimal Supersymmetric Standard Model (cMSSM) [1]. Thus, more general supersymmetric models have to be explored, including for example the phenomenological MSSM (pMSSM-11) [4], specified by 11 SUSY parameters defined at the electroweak scale. The pMSSM-11 allows one to accommodate the anomalous magnetic moment of the muon, the dark matter relic density and the LHC limits from direct searches. However, the large number of model parameters poses a significant challenge for global pMSSM-11 fits.

In this paper we introduce the SCYNet tool for the accurate and fast statistical evaluation of LHC search limits—and potential signals – within the pMSSM-11 and other SUSY models. SCYNet is based on the results of a simulation of new physics models that used CheckMATE 2.0 [5–7], and the subsequently calculated χ^2 from event counts in the signal regions of a large number of LHC searches. The χ^2 -estimate has been used as an input to train a neural network-based regression; the resulting neural network then provides a very

^a e-mail: bechtle@physik.uni-bonn.de

^b e-mail: sbelkner@th.physik.uni-bonn.de

^c e-mail: daniel.dercks@desy.de

^d e-mail: Matthias.Hamer@cern.ch

^e e-mail: tim.keller@rwth-aachen.de

^f e-mail: mkraemer@physik.rwth-aachen.de

^g e-mail: sarrazin@physik.rwth-aachen.de

^h e-mail: schuette@physik.rwth-aachen.de

ⁱ e-mail: tattersall@physik.rwth-aachen.de

Fig. 1 Flowchart indicating how we have calculated the LHC χ^2 via the direct (a) and reparametrized (b) neural network regression approach described in the text. Throughout the paper, the subscript “SN” denotes χ^2 as obtained via SCYNet

fast and reliable evaluation of the LHC search results, which can be used as an input to global fits.

Neural networks allow predictions of complex data by overlapping non-linear functions in an efficient manner, and they are straightforwardly implemented using open source libraries like Tensorflow [8], Theano [9] and Keras [10]. Previous studies have used neural networks to predict the LHC likelihood of the CMSSM [11] and show how the CMSSM parameters [12] may be derived. In addition random forests were used to train a classifier between allowed and excluded models [13] and recently Gaussian processes [14] were used to predict the number of signal events for a simple SUSY model [15].

Here, we have applied neural networks as a regression tool to predict χ^2 values for an ensemble of many signal regions. Compared to other regression methods, neural networks excel when dealing with sparse sample sets due to their non-linear nature. They can thus be valuable tools for predicting LHC χ^2 values of complex BSM models.

In order to train and validate the neural network regression we have considered the pMSSM-11 as an example of a complex and phenomenologically highly relevant SUSY model. Two different approaches have been explored: in the so-called direct approach we have simply used the 11 parameters of the pMSSM-11 as input to the regression; cf. Fig. 1a. The resulting neural network evaluates χ^2 of the pMSSM-11 within milliseconds and can thus be used in global pMSSM-11 fits without time penalty. A second, so-called reparametrized, approach uses SUSY particle masses, cross sections and branching ratios to first estimate signature-based objects, such as particle energies and particle multiplicities, which should relate more closely to the LHC χ^2 values (Fig. 1b). While the calculation of the particle energies and multiplicities requires extra computation time, the corresponding neural network should be more general and be able to predict χ^2

for a wider class of SUSY models. To explore this feature, we have examined how well a reparametrized neural network trained on the pMSSM-11 can predict χ^2 for two other popular SUSY models, namely the cMSSM and a model with anomaly mediated SUSY breaking (AMSB) [16, 17].

This paper is structured as follows: In Sect. 2 we provide details of the LHC event generation within the pMSSM-11, and the method we have used to calculate a global LHC χ^2 . The training and validation of the direct and reparametrized neural net approaches are presented in Sects. 3 and 4, respectively. The results are summarized and our conclusions are presented in Sect. 5. More details as regards the statistical approach are given in the appendices.

2 Event generation and LHC χ^2 calculation

For the training and testing of the neural networks we have calculated the LHC χ^2 for a set of pMSSM-11 parameter points using CheckMATE. This χ^2_{CM} calculation (the subscript “CM” denotes the CheckMATE result) required the generation of SUSY signal events, a detector simulation and the evaluation of the LHC analyses. In this section we describe the simulation chain, list the LHC analyses we include, and explain our calculation of χ^2_{CM} from the event counts in the various experimental signal regions. A graphical depiction of this process can be found in Fig. 2.

The pMSSM-11 is based on 11 SUSY parameters: the bino, wino and gluino mass parameters, M_1, M_2, M_3 , the scalar quark and lepton masses of the 1st/2nd and 3rd generation, respectively, $M(\tilde{Q}_{12}), M(\tilde{Q}_3), M(\tilde{L}_{12}), M(\tilde{L}_3)$, the mass of the pseudoscalar Higgs, $M(A)$, the trilinear coupling of the 3rd generation, A , the Higgsino mass parameter, μ , and the ratio of the two vacuum expectation values, $\tan(\beta)$ [4]. The ranges of the pMSSM-11 parameters we have considered are specified in Table 1.

Table 1 pMSSM-11 ranges we have used for the sample generation. The lower bounds on the mass parameters were motivated by the current collider bounds for SUSY particle production. The relatively large range for third generation trilinear coupling A were motivated by the observed Higgs mass. All the pMSSM-11 parameters were defined at a scale of 1 TeV [18]

pMSSM-11 parameter	Description	Range
M_1	Bino mass	$[-4000, 4000]$ GeV
M_2	Wino mass	$[100, 4000]$ GeV
M_3	Gluino Mass	$[-4000, -400] \cup [400, 4000]$ GeV
$M(\tilde{Q}_{12})$	1st and 2nd gen. scalar squark mass	$[300, 5000]$ GeV
$M(\tilde{Q}_3)$	3rd gen. scalar squark mass	$[100, 5000]$ GeV
$M(\tilde{L}_{12})$	1st and 2nd gen. scalar slepton mass	$[100, 3000]$ GeV
$M(\tilde{L}_3)$	3rd gen. scalar slepton mass	$[100, 4000]$ GeV
$M(A)$	Pseudoscalar Higgs pole mass	$[0, 4000]$ GeV
A	Third generation trilinear coupling	$[-5000, 5000]$ GeV
μ	Higgsino mass parameter	$[-5000, -100] \cup [100, 5000]$ GeV
$\tan(\beta)$	Higgs doublet vacuum expectation value	$[1, 60]$ GeV

Fig. 2 Flow chart of the simulation chain we have used in the sample generation. The subscript “CM” in χ^2_{CM} denotes χ^2 -values obtained using CheckMATE

We have sampled pMSSM-11 points using both a uniform and a Gaussian random probability distribution. The maximum and standard deviation of the Gaussian distribution have been chosen to be one fourth and one half of the parameter range, respectively.¹ Consequently we have generated less points near the edges of the parameter range, and in the decoupling regime of large pMSSM-11 parameters, where the SUSY spectra are beyond the LHC reach.

After we have generated a pMSSM-11 parameter point we have used SPheno-3.3.8 [19] to calculate the spectrum. Following the SPA convention [18], all 11 parameters were defined at a scale of 1 TeV. We have then only proceeded

in the simulation chain if the following pre-selection criteria have been fulfilled:

- There were no tachyons in the spectrum.
- The χ^0_1 was the LSP, such that it was the dark matter candidate.
- Both Higgs bosons (h^0, H^0) had a mass above 110 GeV.
- $m_{\chi^0_1} > 103.5$ GeV [20].
- The experimental value and the predicted value for the electroweak precision observables m_W , $|\Delta(M_{B_s})$, $\text{BR}(B_s \rightarrow \mu\mu)$, $\text{BR}(b \rightarrow s\gamma)$, $\text{BR}(B_u \rightarrow \tau\nu)$ differed by less than 5 times the total uncertainty given in Table 2.

The pre-selection criteria were applied in order to restrict the SUSY parameter space to phenomenologically viable regions. Consequently, the neural networks that were trained on that parameter space will only be valid if the pre-selection criteria are fulfilled. Note that the anomalous magnetic moment of the muon, $(g-2)_\mu$, has not been included in the pre-selection criteria, because in a global fit one might not wish to include this observable.

2.1 Event generation

If all pre-selection criteria were fulfilled we then generated LHC signal events for 8 and 13 TeV. At 8 TeV we simulated the production of both electroweak and strongly interacting SUSY particles, including all possible $2 \rightarrow 2$ scattering processes. The 8 TeV analyses were included since no 13 TeV electroweak searches were available at the time the event generation for this work was started. For the simulation of the electroweak events, which include all combinations of slepton, chargino and neutralino final states, we used MadGraph5_aMC@NLO 2.4 [45] with the CTEQ6L1 PDF [46] and showered the events with Pythia 6.428 [47]. Since the LHC is only sensitive to electroweak production if $m_{\chi^0_1} < 500$ GeV [48], we have only simulated the elec-

¹ For ranges with allowed negative values, this distribution is mirrored around zero.

Table 2 Precision observables used to constrain the parameter space. Note that a newer and slightly more precise measurement $\text{BR}(B_s \rightarrow \mu^+ \mu^-) = (2.8^{+0.7}_{-0.6}) \times 10^{-9}$ is now available but was not included in the pre-selection [25]

Precision observable	Experimental value	Theoretical uncertainty (%)	Refs.
m_W	$(80385 \pm 15) \text{ MeV}$	0.1	[21]
$\text{BR}(b \rightarrow s \gamma)$	$(3.43 \pm 0.21 \pm 0.07) \times 10^{-4}$	14	[22]
$\text{BR}(B_s \rightarrow \mu^+ \mu^-)$	$(2.9 \pm 0.7) \times 10^{-9}$	26	[23]
$\text{BR}(B_u \rightarrow \tau \nu)$	$(1.05 \pm 0.25) \times 10^{-4}$	20	[24]
$ \Delta(M_{B_s}) $	$(17.719 \pm 0.036 \pm 0.023) \text{ ps}^{-1}$	24	[24]

troweak processes if this was the case. The processes which contain final states with strongly interacting SUSY particles, i.e. squarks and gluinos, were simulated using Pythia 8.2² [49,50], with cross sections normalized to NLO using NLL-Fast 2.1 [51–57].

At 13 TeV, only processes with strongly interacting SUSY particles were simulated, since all analyses which were available within CheckMATE target these final states. We have again used MadGraph5_aMC@NLO 5.2.4 [45] to generate events and Pythia 6.248 [47] was used to decay and shower the final state. The cross sections were normalized to NLO accuracy as obtained from NLL-Fast 2.1 [51–57]. One additional parton was generated at the matrix element level and then matched to the parton shower if the mass gap between either the lightest squark or the gluino and the LSP was below 300 GeV. Such a procedure allowed for the accurate determination of the signal acceptance in compressed spectra where initial state radiation is crucial to pass the experimental cuts [58,59].

For 8 TeV we calculated χ^2_{CM} for a total of 210,000 pMSSM-11 parameter points and at 13 TeV we simulated 140,000 parameter points. To guarantee sufficient statistical accuracy, we produced 5 times the expected number of LHC signal events for each particle production process³ up to a maximum of 45,000 events per process due to limited computational resources. In addition we generated a minimum of 1000 events to improve the χ^2 stability in regions of parameter space with small statistics. For each pMSSM-11 parameter point we have simulated on average 20,000 events, which required approximately 380,000 CPU hours in total.

The number of events generated per point ensures that the mean statistical error in the $3\text{-}\sigma$ region around the global minimum is almost an order of magnitude smaller than the trained neural network accuracy (see Sect. 3). As the global χ^2 rises, the associated error also increases but remains a

subdominant source of the neural network error until we are over $4\text{-}\sigma$ (and therefore the models are heavily disfavoured) from the minimum. For future work, the number of event generated will be directed by the associated statistical error.

Once the event generation was completed, the event files were passed through CheckMATE⁴ [5–7] which contains a tuned Delphes-3 [65] detector simulation with separate setups for 8 and 13 TeV. The analyses which have been used to train the SCYNet neural network regression are listed in Table 3.

2.2 LHC χ^2 calculation with CheckMATE

We have implemented a calculation of χ^2_{CM} that approximates a likelihood, based on the CheckMATE output of the event count in each signal region (SR). In the following we will use $\chi^2_{\text{CM},jk}$ to denote χ^2 of analysis j and SR k from CheckMATE. The exact statistical prescription we have used to calculate $\chi^2_{\text{CM},jk}$ for each *single* SR is described in Appendix A.

The individual signal regions were combined to give a global χ^2_{CM} with a procedure that combined the most sensitive (expected) orthogonal SRs. Our algorithm chose these by first dividing the SRs in each analysis into orthogonal disjoint groups. Each disjoint group can contain several SRs and the SRs of one disjoint group are disjoint to all SRs in all other disjoint groups, see Fig. 3. If one disjoint group contains more than one SR, the SR with the largest signal/ S_{exp}^{95} ratio in this group was selected. For one analysis we then added the χ^2 of all selected SRs, $\chi^2_{\text{CM},j} := \sum_{\text{selected } k} \chi^2_{\text{CM},jk}$. In the last step all the $\chi^2_{\text{CM},j}$ from the individual analyses have been added to give the overall $\chi^2_{\text{CM}} := \sum_j \chi^2_{\text{CM},j}$.

All the analyses we have combined for the same center-of-mass energy target different final state topologies and contain explicit loose lepton vetoes so they are guaranteed to be orthogonal. In addition, if a signal region contains an inclusive jet selection, this is considered to be overlapping with all signal regions demanding a higher jet multiplicity and these regions are not combined. The one exception where we allow potentially overlapping signal regions are those

² At 8 TeV, strong production was not generated with additional QCD radiation at the matrix element level to save computational resources. The reason is that, for strong production, the 13 TeV results provide the most stringent limits and therefore this additional accuracy is not required at 8 TeV.

³ We split the processes up into the following groups: *ew ew*, *ew strong*, $\tilde{g}\tilde{g}$, $\tilde{q}\tilde{q}$, $\tilde{b}_1\tilde{b}_1$, $\tilde{b}_2\tilde{b}_2$, $\tilde{t}_1\tilde{t}_1$, $\tilde{t}_2\tilde{t}_2$. *ew* refers to all SUSY particles without QCD charge and *strong* refers to all SUSY particles with QCD charge.

⁴ CheckMATE uses the FastJet library [60,61] and in particular the anti-kt algorithm [62] for jet reconstruction. In the course of this work we have used analyses originally developed for SUSY searches in the NMSSM [63] and the Super-Razor observable [64].

Table 3 Analyses considered at 8 and 13 TeV. The names correspond to the internal CheckMATE nomenclature. $\mathcal{L}$ denotes the luminosity and N_{SR} the number of SRs for each analysis

Nr.	Name	Description	$\mathcal{L}[\text{fb}^{-1}]$	N_{SR}	Refs.
<i>Analyses @ 8 TeV</i>					
1	atlas_1402_7029	3 leptons + $\cancel{E}_T$	20.3	24	[26]
2	atlas_1403_5294	2 leptons + $\cancel{E}_T$	20.3	9	[27]
3	atlas_conf_2013_036	4 or more leptons	20.7	5	[28]
4	atlas_1308_2631	2 b-jets + $\cancel{E}_T$	20.1	6	[29]
5	atlas_1403_4853	2 leptons(opposite sign)	20.3	12	[30]
6	atlas_1404_2500	2 leptons(same sign) + jets	20.3	5	[31]
7	atlas_1405_7875	Multiple jets + $\cancel{E}_T$	20.3	15	[32]
8	atlas_1407_0583	1 isolated lepton + jets + $\cancel{E}_T$	20.3	27	[33]
9	atlas_1407_0608	2 stop quarks	20.3	3	[34]
10	atlas_1502_01518	1 jet + $\cancel{E}_T$	20.3	9	[35]
11	atlas_1503_03290	2 leptons(same flavor, opposite sign) + $\cancel{E}_T$	20.3	1	[36]
<i>Analyses @ 13 TeV</i>					
12	atlas_conf_2015_076	1 isolated lepton + jets + $\cancel{E}_T$	13.3	6	[37]
13	atlas_1602_09058	Jets + 2 leptons(same sign) or 3 leptons	3.2	4	[38]
14	atlas_1605_03814	Multiple jets + $\cancel{E}_T$	3.2	7	[39]
15	atlas_conf_2015_082	Leptonically decaying Z + jets + $\cancel{E}_T$	3.2	1	[40]
16	atlas_1604_07773	1 jet + $\cancel{E}_T$	3.2	13	[41]
17	atlas_conf_2016_013	Multiple leptons + Multiple Jets	3.2	10	[42]
18	atlas_conf_2015_067	b-jets + $\cancel{E}_T$	3.2	3	[43]
19	cms_pas_sus_15_011	2 leptons(same flavor, opposite sign) + $\cancel{E}_T$	2.2	47	[44]

Fig. 3 Example for a disjoint group selection for one analysis. The analysis has four SRs (A,B,C,D). Because SR A overlaps with SR B and SR B overlaps with SR C, we grouped SR A,B and C into one disjoint group G1. Because SR D does not overlap with any of the other SR of the analysis, the disjoint group G2 only contains the SR D

that require b-jets with those that focus on light jets. The reason is that very few signal regions demand b-jet vetoes and consequently if we did not allow these potential overlaps, we would be unable to combine the b-jet analyses with any of the other searches. However, we think that this is an acceptable approximation, since the b-jet analyses are normally far more sensitive than normal jet searches once the signal con-

tains b-jets, since they allow for a substantial reduction in the SM background. Consequently the χ^2 from third generation production will be dominated by the specialized searches and the correlated signal regions will not cause problems in practice. In the case of the 8 (13) TeV analyses we found 47 (65) disjoint SRs for the group with the largest sensitivity.

Since the experimental collaborations do not provide information on the correlation of the systematic errors, we had to assume the uncertainties were Gaussian distributed and uncorrelated.

Figure 4a, b show the distributions of χ^2_{CM} for all sampled points. Two peaks can be observed in both plots: the first peak, called the *zero signal peak*, is located at around $\chi^2_{\text{CM}} \approx 40$ (8 TeV, left plot) and $\chi^2_{\text{CM}} \approx 54$ (13 TeV, right plot), respectively. All χ^2_{CM} values in the zero signal peaks belong to pMSSM-11 parameter points with very heavy SUSY spectra and thus very little or no expected signal in any of the SRs. A second peak occurs at $\chi^2 = 100$ because we set all $\chi^2_{\text{CM}} > 100$ to this value. This was done to avoid the neural networks learning any structures at large $\chi^2_{\text{CM}} > 100$, where the model is already clearly excluded by the LHC data.

The three regions labeled I, II and III outside of the zero signal peak and the peak at $\chi^2_{\text{CM}} = 100$ in Fig. 4a, b are called *rare target ranges*. We differentiated between regions below (I) and above (II and III) the zero signal peak and between

Fig. 4 Distributions of χ_{CM}^2 for 8 TeV (a) and 13 TeV (b). The dashed lines indicate ranges for which we evaluated the performance of the SCYNet neural network separately as described in the text

regions with χ_{CM}^2 closer to the minimum (I and II), since these are much more important for global fits than those far away, and thus need better modeling. We have evaluated the performance of the SCYNet neural network separately for these different regions.

3 Neural network regression: direct approach

In this section the direct approach for neural network regression used in SCYNet has been described. The inputs to the network were the 11 parameters of the pMSSM-11 and the output, χ_{SN}^2 , is an estimate of χ_{CM}^2 for the LHC searches described in Sect. 2. We have trained separate networks for the 8 and 13 TeV analyses, with outputs $\chi_{\text{SN},8\text{TeV}}^2$ and $\chi_{\text{SN},13\text{TeV}}^2$, respectively. In this section, the setup and performance of both networks has been discussed, but we have focused on the results obtained for $\chi_{\text{SN},8\text{TeV}}^2$ for illustration.

3.1 Parameters of the neural networks

All neural networks used in this section were of the simple feed forward type⁵ and have been designed using the Tensorflow 0.11 [8] library. We performed scans of the neural network parameters (so-called hyperparameters) to find the best configuration.

The hyperparameters which we were trying to optimize, were the number of hidden layers (varied between 2 and 5), the number of neurons in the hidden layers (varied between 50 and 450), the batch size (varied between 80 and 3000), the

learning rate of the Adam minimization algorithm [66] (varied between 10^{-1} and 10^{-4}) and the regularization parameter (varied between 10^{-3} and 10^{-6}). We performed a directed grid scan of the hyperparameters and the networks discussed in this section always refer to the configuration that was found to function best.⁶ The results of the hyperparameter optimization are independent on the χ_{CM}^2 range considered. To prevent overfitting of the hyperparameters, for each scanned hyperparameter we choose a random validation set. A brief overview of neural network terminology is given in Appendix C.

Our 8 TeV network had four hidden layers $l = 1, \dots, 4$, each with $N_l = 300$ neurons and all neurons have hyperbolic tangent activation functions. The weights in layer l were initialized with a Gaussian distribution with standard deviation $1/\sqrt{N_{l-1}}$ and mean zero, while the biases were initialized with a Gaussian distribution with standard deviation equal to one and mean equal to zero. In order to train a network with a hyperbolic tangent output activation function, the targets were transformed to the range between -1 and 1 with a modified Z-score normalization (see details in Appendix B).

During the training we used a quadratic cost function and trained with a batch size of 750. We used the Adam minimization algorithm with a learning rate of 0.001 and all other parameters of the Adam optimizer were set to the default values from [66].

The complete cost function C consists of the quadratic cost function and a quadratic regularization:

$$C = \text{quadratic cost} + \frac{\lambda}{2N_{\text{train}}} \sum_{jkl} (w_{jk}^l)^2, \quad (1)$$

⁵ A feed forward neural network is ‘simple’ if neurons from the l th layer are only connected with neurons in the $l+1$ th layer.

⁶ The optimal hyperparameter configuration is the one which produces the smallest average error between χ_{SN}^2 and χ_{CM}^2 on the validation set.

Fig. 5 Evolution of the mean error on points in the validation set during training on the 8 TeV LHC χ^2_{CM} compared to a nearest neighbor interpolator for all points (a), divided into different χ^2_{CM} ranges (b)

where N_{train} were the number of points in the training set, λ is the regularization parameter and w_{jk}^l is the weight connecting the k th neuron in the $(l-1)$ th layer with the j th neuron in the l th layer. We used 10,000 validation points ($N_{\text{val}} = 10,000$) while the rest of the sampled points were used for training.⁷ In the hyperparameter scan we found that $\lambda = 10^{-5}$ gave the best network performance. In addition to the above regularization term to avoid overfitting we also always checked that error on the validation and training dataset were comparable.

The 13 TeV network had the same structure as the 8 TeV network but the batch size during training was slightly adjusted to 500, while all other hyperparameters were the same as in the 8 TeV case. The training of a single neural network took approximately 1 hour using 4 CPU cores.

3.2 Training the neural networks

The training phase of the 8 TeV neural network has been visualized in Fig. 5a, b where we have compared the mean error on points in the validation data during the training phase to that of a nearest neighbor interpolator. The following parameter has been used to quantify the network performance,

$$\text{Mean error on valid. points} := \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} |\chi_{\text{CM}_i}^2 - \chi_{\text{NN}_i}^2|. \quad (2)$$

After the last training epoch the mean error on points in the validation set is 1.68. As shown in Fig. 5a this performance

Table 4 Mean errors of the neural networks for the 8 TeV and 13 TeV LHC χ^2 prediction in different ranges. The results for the direct and reparameterized approach (Sect. 4) are compared

8 TeV					
χ^2 range	0–38	38–42	42–70	70–95	95–100
Direct	1.9	0.7	3.7	6.7	1.0
Reparameterized	1.6	0.5	3.6	7.3	1.0
13 TeV					
χ^2 range	0–53.5	53.5–56	56–70	70–95	95–100
Direct	1.8	0.9	2.4	4.0	0.5
Reparameterized	2.0	0.7	2.2	4.7	0.3

is a significant improvement over a nearest neighbor interpolator [67].

Figure 5b shows how the mean errors for χ^2 values in the different ranges defined in Fig. 4a, b behaved during the training phase. The corresponding mean errors after the last training epoch can be found in Table 4. The mean errors in the rare target ranges I ($0 < \chi^2 \leq 38$), II ($42 < \chi^2 \leq 70$), and III ($70 < \chi^2 \leq 95$) were larger than the mean errors in the zero signal range and in the range around 100. This behaviour will be called *rare target learning problem* (RTLTP) in the following and can be understood from the χ^2_{CM} distribution in Fig. 4a and the corresponding discussion in Sect. 2.2: the majority of the scanned pMSSM-11 points led to χ^2_{CM} values in the range $38 < \chi^2 \leq 42$ and $95 < \chi^2$, which were thus described more accurately by the neural network than those in regions I, II and III.

In future work the RTLTP will be addressed with two strategies: the 11-dimensional probability density function (pdf) in

⁷ No test set was used because in the hyperparameter scan the validation set was chosen randomly for each scanned hyperparameter.

Fig. 6 Error histogram for the neural network compared to the nearest neighbor interpolator for all validation points (a), divided by χ^2 ranges (b)

the parameter space of the points in the range $0 < \chi^2 \leq 38$ and $42 < \chi^2 \leq 95$ will be used to randomly sample new pMSSM-11 points which will be added to the training and validation samples. In the second approach, each point in the RTLP region will be used to seed new random points using a narrow 11-dimensional pdf centered around each point in the above-mentioned χ^2 range. New points can then be generated, simulated and used to improve the network especially in the rare target ranges. Motivated by the profile likelihood requirement, which sets an estimate of $\Delta\chi^2 = 1$ for the 1σ range, we plan to improve the training and validation set size by subsequent application of these procedures until a mean error of $\Delta\chi^2$ well under 1 is reached.

For the 13 TeV neural network we found similar results to those from the 8 TeV network that has already been displayed in Fig. 5a, b. The mean errors for the 13 TeV network at the end of the training phase have been given in Table 4. The histogram in Fig. 6a shows the difference between the CheckMATE and SCYNet results $\chi^2_{SN} - \chi^2_{CM}$. Again the comparison to the nearest neighbor interpolator shows that the neural network provides a much more powerful tool to predict the LHC χ^2 . The mean error for the neural network on points in the validation set is 1.45.

The performance of the 13 TeV network is also different for the different χ^2 ranges; see Fig. 6b. However, the difference between the mean error in the zero signal range and in the rare target ranges is less pronounced than for the 8 TeV network. This can be understood from comparing Fig. 4a, b: the 13 TeV LHC analyses are sensitive to a wider range of pMSSM-11 parameters, and thus fewer of the sampled points result in no signal expectation.

3.3 Testing the neural networks

In this section we have used an additional, statistically independent validation set of 60,000 pMSSM-11 points that passed the pre-selection criteria to compare the SCYNet prediction to the CheckMATE result. For illustration we have focused on the projection of the 11-dimensional pMSSM parameter space onto the masses of the gluino $\tilde{g}$ and the neutralino $\tilde{\chi}_1^0$, which are particularly relevant for χ^2 of the LHC searches. All plots in this section have been given for the 8 TeV case, while similar results were obtained for the 13 TeV case.

In Fig. 7a we have presented the minimal χ^2_{CM} obtained by CheckMATE for the validation set of pMSSM-11 points in bins of the gluino and neutralino masses. The minimum χ^2 in each $(m(\tilde{g}), m(\tilde{\chi}_1^0))$ -bin typically corresponds to scenarios, where all other SUSY particles, and in particular squarks of the first two generations, were heavy and essentially decoupled from the LHC phenomenology.

In Fig. 7b the corresponding result obtained from the SCYNet neural network regression has been given. We found that the neural network reproduces the main features of the χ^2 distribution. We emphasize here that each bin represents a single pMSSM-11 parameter point and consequently the results show that the network successfully reproduces the LHC results across the whole plane.

Figure 8a, b again show the χ^2_{CM} and χ^2_{SN} values as a function of the gluino and neutralino masses, but now have we displayed the maximum χ^2 in each bin. Comparing the two figures proves that the neural network reproduces the main features of the χ^2 distribution well.

Fig. 7 Minimum pMSSM-11 χ^2 in the gluino-neutralino mass plane for the 8 TeV validation set for the CheckMATE results (a) and the predictions of SCYNet (b)

Fig. 8 Maximum pMSSM-11 χ^2 in the gluino-neutralino mass plane for the 8 TeV validation set for the CheckMATE results (a) and the predictions of SCYNet (b)

Fig. 9 Difference between the CheckMATE and SCYNet χ^2 predictions in the gluino-neutralino mass plane for all 8 TeV validation points. In each bin the mean difference has been calculated for all validation points

The difference between the CheckMATE and SCYNet result, $\chi^2_{\text{CM},8\text{TeV}} - \chi^2_{\text{SN},8\text{TeV}}$ has been presented in Fig. 9. In this plot we have taken the mean difference between both results for all validation points that lie in the respective histogram bin in order to demonstrate in which regions of parameter space the network performs best. We have found overall very good agreement between the CheckMATE and SCYNet result. However, sizable differences were visible in the compressed regions where $m(\tilde{g})$ is close to $m(\tilde{\chi}_1^0)$. This particular region has been probed by monojet searches [35, 41] which were sensitive only for very degenerate spectra. Thus the χ^2 contribution peaks suddenly as the mass splitting between the SUSY states is reduced. Unfortunately, the rapid change in χ^2 makes this region difficult for the neural network to learn and will be targeted specifically in future work by generating more training data in this area.

Fig. 10 Using validation points from Fig. 9, we show the difference between the SCYNet parametrization and the CheckMATE result against $m(\tilde{\chi}_1^0)$ for gluino masses between 850 and 900 GeV (a) and against $m(\tilde{g})$ for neutralino masses between 400 and 450 GeV (b)

Fig. 11 All validation points with $m(\tilde{g}) > 1.5 \text{ TeV}$ and $m(\tilde{q}_{1,2}) > 1.5 \text{ TeV}$ are used to produce the two plots. In a we show the difference between the SCYNet parametrization and the CheckMATE result

against $m(\tilde{\chi}_1^+)$ for $50 \text{ GeV} < m(\tilde{\chi}_1^0) < 250 \text{ GeV}$. In b we show the difference between the SCYNet parametrization and the CheckMATE result against $m(\tilde{t}_1)$ for $150 \text{ GeV} < m(\tilde{\chi}_1^0) < 350 \text{ GeV}$

As obvious from Fig. 9, the parametrization of χ^2_{CM} through SCYNet worked very well on average. However, the pMSSM-11 points still have a rather broad distribution of $|\chi^2_{CM} - \chi^2_{SN}|$, especially near the crucial transition from the non-sensitive to the sensitive region. After all, this is exactly the region where the *rare target learning problem* (RTLTP) alluded to in Sect. 3.2 occurs. In order to illustrate this, in Fig. 10a we have displayed the distribution of $\chi^2_{CM} - \chi^2_{SN}$ in bins of $m(\tilde{\chi}_1^0)$ for gluinos with a mass between 850 and 900 GeV. In Fig. 10b the equivalent result for neutralino masses between 400 and 450 GeV has been given. The mass ranges were chosen such that we include the transition regions from a low to a high χ^2_{CM} in the minimum profile plot 7a. We can clearly say that in both cases there is a narrow peak around $\chi^2_{CM} - \chi^2_{SN} = 0$ and this is also true in

the crucial transition regions. However, we can also see that the RTLTP causes few, but significant outliers, which will be subject to future improvements with targeted training.

We also examined the performance of the neural network to parameter points where the first and second generation squarks and gluinos are decoupled ($m > 1.5 \text{ TeV}$). For these parameter points the phenomenology is dominated either by electroweak (mainly chargino and neutralino) or stop production. In Fig. 11a we again display $\chi^2_{CM} - \chi^2_{SN}$ but this time versus the lightest chargino mass, $m(\tilde{\chi}_1^+)$ in parameter points where $50 \text{ GeV} < m(\tilde{\chi}_1^0) < 250 \text{ GeV}$. We see that the accuracy is comparable to strong production with the majority of the points having small errors but again some outliers are present across the mass range. In Fig. 11b we show the corresponding plot for light stop quarks and we see that for

$m(\tilde{t}_1) \gtrsim 600$ GeV the accuracy is comparable. Unfortunately for $m(\tilde{t}_1) \lesssim 600$ GeV we found very few points in our training or validation dataset due to the various b -physics limits we place on the pre-selection. Consequently the accuracy of the network degrades in this region but we plan to target parameter point generation here in future work.

We finally note that our results for the pMSSM-11 χ^2 cannot be compared in a straightforward way to exclusions published by the LHC experimental collaborations. ATLAS and CMS have not presented any specific analyses for the pMSSM-11, but typically interpret their searches in terms of SUSY simplified models. The simplified models assume 100% branching ratios into a specific decay mode, which does not hold in the pMSSM-11. Instead, in the pMSSM-11 there are in general a number of competing decay chains that result in a large variation in the final state produced in different events. As a result, the events do not predominantly fall into the signal region of one particular analysis, but are instead shared between many different analyses. Thus the constraints on the pMSSM-11 are in general weaker than those on simplified SUSY models.

4 Neural network regression: reparametrized approach

The direct approach of SCYNet discussed in Sect. 3 allows for a successful representation of the pMSSM-11 χ^2 from the LHC searches. However, despite the successful modeling, there is motivation to explore an alternative ansatz. Foremost, the model parameters used as input to the direct approach do not necessarily correlate with the χ^2_{CM} behavior. For example, parameters such as A and $\tan \beta$ are not directly linked to any single observable in the signal regions considered here. This separation implies a complex function that has to be learned and modeled by the neural network itself. Another downside of the direct approach is that a neural network trained on the model parameters is inherently model dependent. For every model considered, new training data is required and a new neural network must be trained.

In this section, a different set of phenomenologically motivated parameters was proposed as an input to a new network: The *reparametrized network*. These input parameters are in principle observables, and they are more closely related to the χ^2_{CM} values. Most importantly they are model independent. It was the aim of this ansatz to reach a performance which was at least comparable or better to the performance of the SCYNet direct approach. However, this comes at the cost of an increase in computation time, since for each evaluation, branching fractions and cross sections have to be calculated. The methods used for building and training the neural network were similar to the ones used in the direct approach but here we found that a deeper network of 9 layers performed better.

Fig. 12 Flowchart of the reparametrization procedure

4.1 Reparametrization procedure

A new model is excluded if in addition to the expected Standard Model background a statistically significant excess of events was predicted for a particular signal region which, however, is not observed in data. These observables generally correspond to a combination of final state particle multiplicity and their corresponding kinematics. For example, if two distinct models produce the same observable final states at the LHC they would both be allowed/excluded independently of the mechanism producing the events. The reparametrized approach aims to calculate neural network inputs that relate more closely to these observables. As displayed in Fig. 12 all the allowed 2-to-2 production processes were considered. For each produced particle, the tree of all possible decays with a branching ratio larger than 1% was traversed. At the run time, each occurrence of a final state jet, b-jet, $e^\pm$, $\mu^\pm$, $\tau^\pm$, $\cancel{E}_T$ as well as intermediate state on-shell $W^\pm$, Z and $t/\bar{t}$ were counted. Note that the algorithm does not differentiate between the type of particle producing the missing energy, since these are always invisible to the detector. Therefore, the set of all invisible particle types (including for example neutrinos or a SUSY LSP) was considered as a single final state type. The charge conjugated partners of all final state particles and resonances (excluding jets, b-jets, Z and $\cancel{E}_T$) were considered separately which adds to 9 separate

final state categories and 5 different parameters for the resonances. After all decay trees are constructed, the weighted mean and standard deviation of the number and the maximal kinematically allowed energy of observed final state particles and resonances were also calculated (see Sect. 4.2 for more details). The weights were the individual occurrence probability of each respective decay tree. Finally these quantities were averaged across all 2-to-2 production processes weighted by their individual cross sections.

The aforementioned quantities complete the set of inputs in the reparametrized approach which adds up to a total of 56 parameters. The branching ratios were analytically calculated using SPheno-3.3.8 [19] and read in with PYSLHA. 3.1.1 [68]. For strong processes NLL-Fast [69] was used for both 8 and 13 TeV, since the cross sections are quickly obtained from interpolations on a pre-calculated grid. For electroweak production processes Prospino2.1 [51–57, 70–74] was used. In the standard configuration, the cross section evaluation of Prospino requires the most computing time in our complete calculation. Consequently we only calculate the cross sections to leading order and we reduce the number of iterations performed by the VEGAS [75] numerical integration routine, resulting in an increase on the statistical uncertainty from $< 1\%$ to a few percent. Obviously the induced error slightly damages the mean accuracy of the reparametrized neural net. Nevertheless, the final accuracy of the reparametrized approach was comparable to the direct approach, as demonstrated below.

The total computation time of the reparametrization requires 3–11 seconds and the Prospino run time was the dominant factor despite the reduced integration accuracy. This was still a significant improvement over $\mathcal{O}(\text{hours})$ needed without a parametrization like SCYNet. However, compared to the order of millisecond evaluation of the direct approach this makes the reparametrized approach less attractive for applications in global fits.

4.2 Motivation and calculation of the phenomenological parameters

Almost all analyses that search for new physics define the final state multiplicity (this can also be a range) in the relevant signal regions. Consequently the mean number of final state particles aims to help the neural net to find areas in parameter space where individual signal regions dominate. In addition, many analyses differentiate themselves by a kinematical selection of an on-shell W , Z or t in the decay chain, in order to isolate and exclude certain SM backgrounds. For this reason we have also included the mean number of resonant SM states in the decay chains as another parameter.

When we examined the missing energy SUSY searches used in this study, we found that the most prevalent cuts were those based on the energy of the final state particles. This is

also true of a missing energy cut which can be considered as the (vector) summation in the transverse plane of the energy of particles that go undetected. Consequently we sought to introduce a parameter that is correlated to the energy of the particles that would be observed. However, we had to keep in mind that the aim of our study was to be able to calculate the LHC χ^2 in as short a time as possible and any calculation that consumes too much CPU, most importantly those which would require an integration over possible phase space configurations, disqualifies the reparametrization from being practical.

Since we already used the mean multiplicity of the various final state (and resonant) particles, the obvious choice would have been to calculate the mean energy for each of these states. However, for a cascade decay that may have included a number of $1 \rightarrow 3$ and even $1 \rightarrow 4$ decays and such a calculation requires a phase space integration, which unfortunately cannot be performed in the time frame available for each call to SCYNet. As a replacement, we instead chose to use the maximum kinematically allowed energy a particle can have as a weighted average over all production processes and possible decay chains.

In the center of mass frame, this maximum was calculated with the term,

$$E_i^{\max} = \sqrt{\frac{(m_k^2 - (\sum_j m_j + m_i)^2)(m_k^2 - (\sum_j m_j - m_i)^2)}{(2m_k)^2}} + m_i^2. \quad (3)$$

$E_i^{\max}$ gives the maximum energy of daughter particle i with mass m_i in the decay $k \rightarrow i + \{j\}$ in the center-of-mass of the mother k . m_k is the mass of the mother particle while m_j represents the mass of the other daughter particle(s) j . This term is fast to calculate and the same for any number of daughter particles.

In the next step, the energy of i must be boosted from the center of mass frame of the mother k into the lab frame:

$$E_i^{\max, \text{Lab}} = \gamma_k E_i^{\max}. \quad (4)$$

Here, the boost γ_k depends on the energy of the mother k as measured in the lab-frame which we have estimated at an earlier stage in the decay chain, $\gamma_k = E_k^{\text{Lab}}/m_k \approx E_k^{\max, \text{Lab}}/m_k$. For the initial mother (heaviest) particle in the decay chain, we assumed that this was produced at rest and consequently, $E_{\text{initial}}^{\max, \text{Lab}}$ was fixed to m_{initial} .

During averaging across all decay trees, not only the mean of the parameters but also the standard deviation was calculated and used as an input. The motivation for such a parameter can easily be understood if for example we consider two new physics scenarios, one which always produces a single lepton in the final state, while the other produces four leptons but only in 25% of decay chains. Both of these scenarios

Fig. 13 Progression of the mean absolute error the direct and reparametrized neural networks achieve on the validation set after each training iteration for 8 TeV (a) and 13 TeV (b)

Table 5 Mean errors on the LHC χ^2 prediction for both approaches in SCYNet

Energy in TeV	Direct approach	Reparameterized approach
8	1.7 (3.1%)	1.5 (2.7%)
13	1.5 (2.0%)	1.3 (1.8%)

have a mean number of leptons equal to one but if we have a signal region that requires 4-leptons, clearly only the second scenario will satisfy such an analysis.

The set of inputs in the reparametrized approach added 56 parameters but it cannot be trivially assumed that the function mapping the presented parameters to χ^2_{SN} is injective, nor that they are uncorrelated. In a way, the parameters can be seen as a replacement for a first layer of the neural net preparing the inputs to facilitate the modeling process. Their viability should be evaluated based on the results they produce. In theory the reparametrization could be performed for any model, but in practice the cross sections and branching ratios must be obtainable in a timely manner which is why we have restricted ourselves to supersymmetric models where such tools were already available.

4.3 Architecture and training of the reparametrized neural network

For the reparametrized approach, we again used Tensorflow 0.11 [8] but also made use of the convenient Keras 1.1 [10] frontend. Here we found that a deeper architecture was preferred with a first layer built from 500 hidden nodes followed by eight layers of 200 hidden nodes. The added depth, creates additional challenges and required us to alter the hyperparameter setup that we used in the direct

approach. Here, rectified linear units (relu) [76] were used as activation functions due to them being less prone to the vanishing gradient problem. Due to the larger architecture, overfitting was expected to play a larger role and to counteract this, the regularization term was increased by a factor of 100 to be $\lambda = 0.001$; see Eq. (1). In addition the batch size was reduced to 120 (32 for 13TeV) to improve convergence.⁸ As in the direct approach, before starting the training, the inputs were again forced to undergo a Z-score normalization. Since the ReLU activation function is not restricted, the modification described in Appendix B was not necessary.

In contrast to the model parameters, the reparametrized inputs were heavily correlated and consequently the inputs were decorrelated by projecting the data into the eigenvectors of the covariance matrix. These mappings were calculated based on the training dataset and then stored to be able to apply the same mappings when evaluating the network. For the training itself, the nAdam [66,77,78] optimizer was used. Compared to the Adam optimizer used in the previous sections, the nAdam optimizer adds Nesterov momentum.⁹ The learning rate was initialized a higher value of 0.01 for 8TeV and 0.002 for 13 TeV. Similar to the batch size, the larger learning rate improved the convergence behaviour in the early stages of the training and was later removed, since, as in the direct approach, the learning rate was reduced if the mean error has stopped improving. The training of a single neural network took approximately 8 hours using 4 CPU cores.

⁸ The added random element supports the minimizer in overcoming flat areas and local minima.

⁹ Instead of applying the momentum with the gradient, the momentum is applied first and the gradient is calculated after the update for the updated weights.

Fig. 14 Performance of the neural network trained on the reparametrized pMSSM-11 points at 8 TeV (a) and 13 TeV (b)

Fig. 15 Difference between the CheckMATE and SCYNets χ^2 predictions in the gluino-neutralino mass plane for all 8 TeV validation points for the reparameterized network. In each bin the mean difference has been calculated for all validation points

4.4 Results of the reparametrized approach to LHC neural nets

During the calculation of the phenomenological parameters, cross sections and branching ratios were calculated. These quantities are related to the expected signal events in each signal region in a much more direct way than the model parameters. Consequently the target quantity of the parametrization, χ^2 as derived from the signal counts in each non-overlapping signal region should be more directly dependent on the input parameters of the reparametrized network. This hypothesis was supported by Fig. 13 which showed that a nearest neighbor interpolator improves (in both the 8 and the 13 TeV case) when moving from the direct to the reparametrized approach. We interpreted this result as showing that the function mapping the reparameterized inputs to the outputs was flatter.

This improvement directly relates to the performance of neural nets which is given in Table 5 as the total mean error of all networks while the distribution of the mean error has been given in Fig. 14. In total, the reparameterized approach on average displayed a lower error. Consequently, as we have shown in Table 4, the reparameterized approach outperformed the direct approach in several χ^2 ranges. This is offset by the significantly longer computation time required for each call of the reparametrized net in SCYNets. The dominant component of this computation is the time required by Prospino to calculate the electroweak cross sections. Additional research is now being performed to see if this calculation can be done by a separate neural network. That may allow one to bring down the computation time as to be competitive with the direct approach.

In Fig. 15 we have again shown the average accuracy of the neural network in the gluino-neutralino mass plane. The plot displays the same general features as seen in Fig. 9 with the best accuracy for low masses where the model is effectively ruled out and in the decoupling regime as a result tends towards the Standard Model χ^2 . Between these two regions we can again see that the accuracy of the network suffers and this is due to the same RTLP discussed in Sect. 3.2. Interestingly however, we do not see a reduction in the accuracy of the network in the compressed mass region like we did in the direct approach. We believe that this is due to the fact that the input parameters to the network are the maximum final state energies of the particles. All the energies reduce significantly in the compressed region and such a direct correlation should be easy for the reparameterized neural network to learn. In contrast, in the direct approach, the network also has to learn the precise combination of input parameters that lead to a compressed spectra.

Fig. 16 Performance of the neural network trained on the reparametrized pMSSM-11 points on the cMSSM (a) and AMSB (b). In each bin the mean difference was calculated for all validation points

The most useful upside of the reparametrized approach was that the used input parameters were chosen in a way which should make them (in principle) model independent. This could allow a network which was trained with one model to be used as prediction tool for a wide variety of models. In Fig. 16 we have displayed the performance of the 13 TeV reparametrized network trained with the pMSSM-11 samples when fed with points from the cMSSM and the AMSB. We emphasize here that due to Renormalization Group Equation (RGE) running, both the cMSSM and the AMSB models are *not* subsets of the pMSSM-11, since in general all scalar masses become non-degenerate. In large regions of the parameter space, the network has predicted the model χ^2 directly calculated from CheckMATE correctly. However, along the transition region from clearly excluded (in the bottom of the frame) to clearly not excluded (in the top of the frame) a few regions of parameter space display discrepancies.

In particular a visible anomaly can be seen in the top left of Fig. 16 for the AMSB and to a lesser degree the cMSSM. These wrongly predicted points arise because in the pMSSM-11 model, all the stop and the sbottom states (both left- and right-handed) were assumed to be mass degenerate. However, as already alluded to, RGE effects mean that in general this was not the case in the AMSB and cMSSM. In fact, the model points in the anomaly region exhibited a light gluino decaying exclusively into a stop-top pair,

$$\tilde{g} \rightarrow \tilde{t} t, \quad (5)$$

since the sbottoms are heavier than the gluino here. The stops then further decay, producing either another top quark and a neutralino or a b-quark and a chargino depending on the exact details of the mass spectra via,

$$\tilde{t} \rightarrow t \tilde{\chi}^0 \quad \text{or} \quad \tilde{t} \rightarrow b \tilde{\chi}^\pm. \quad (6)$$

Further decays of the top/chargino always produce a W -boson, leading to two W -bosons per decay chain and four in the complete event. In the considered pMSSM-11 model however, one always additionally observes the gluino decay via an on-shell sbottom and these often do not decay via W -bosons. The average number (and standard deviation) of resonant W -bosons is part of the neural network inputs in the reparametrized approach. Since all decays were averaged during the reparametrization procedure, the pMSSM-11 samples will never contain such a large number of intermediate W bosons. Thus, when we apply the neural network to the AMSB or cMSSM, we no longer interpolate between known parameter points but instead extrapolate to regions of parameter space that the network has never sampled as they were not represented in the training data.

A simple solution to counteract the lack of training data that represents parameter points that only contain light gluinos and stops was to generate additional points in the pMSSM-19 [4]. For these points, the particles charged under $SU(3)$ were forced to obey the following mass relation:

$$m_{\tilde{t}} < m_{\tilde{g}} < m_{\tilde{u}, \tilde{d}, \tilde{s}, \tilde{c}, \tilde{b}} \quad (7)$$

in order to generate training data in the region the reparametrized network fails.

The results of a network trained with the additional pMSSM-19 samples have been displayed in Table 6. The network performs slightly worse on the pMSSM-11 set which was to be expected because the net had to focus on regions of the parameter space which were not present in the pMSSM-11. On the other hand, the mean error of the network applied to the AMSB and cMSSM was significantly reduced. The

Table 6 Performance of the 13 TeV neural network in the reparametrized approach applied to models different from the pMSSM-11 used for training

Model	Only pMSSM-11 samples	Additional pMSSM-19 samples
pMSSM-11	1.32	1.50
CMSSM	2.43	1.92
AMSB	2.44	1.63

Fig. 17 Performance of the neural network trained on the reparametrized pMSSM-11 points with added samples containing non-degenerate stop sbottom pairs on the CMSSM (a) and AMSB (b). In each bin the mean difference was calculated for all validation points

effects of this can be observed in Fig. 17 which has significantly reduced errors compared to Fig. 16.

One may notice that anomalous areas still exist with larger errors. This was due to the fact that we did not sample the additional pMSSM-19 points with a high enough density for the neural network to learn the parameter space properly. The points with the worst reconstruction again contain a spectrum with a lighter stop and can be expected to be improved if the pMSSM-19 points were sampled correctly from the beginning.

Nevertheless, the result does call into question whether our reparameterized neural network can really be called ‘model independent’, since it fails when faced with input parameters outside of the range that were used to train the net. Consequently, each published network will come with data that defines the range of applicability of the network and the user will be warned if this limits are exceeded. To go outside of this range will then require new parameters points to be simulated and the neural network to be retrained.

5 Conclusions

We have developed a neural network regression approach, called SCYNet, for calculating χ^2 of a given SUSY model from a large set of 8 TeV and 13 TeV LHC searches. Previously, such a χ^2 calculation would require computational intensive and time consuming Monte Carlo simulations. The

SCYNet neural network regression, on the other hand, allows for a fast χ^2 evaluation and is thus well suited for global fits of SUSY models.

We have explored two different approaches: in the first, so-called direct, method we simply used the pMSSM-11 parameters as input to the neural network. Within this method the χ^2 evaluation for an individual pMSSM-11 parameter point only takes few milliseconds and can thus be used for global pMSSM-11 fits without time penalty. However, the neural network based on the pMSSM-11 parameters cannot be used for any other model. In the second approach, we reparametrized the input to the neural network. Specifically, we used the SUSY masses, cross sections and branching ratios of the pMSSM-11 to estimate signature-based objects, such as particle energies and multiplicities, which provide a more model-independent input for the neural network regression. Although calculating the particle energies and multiplicities for any given parameter point requires $\mathcal{O}(\text{seconds})$ of computational time, the reparametrized network trained on a particular model can in principle be applied to BSM scenarios the network has not encountered before.

The mean error of both neural network approaches lies in the range of $\Delta\chi^2 = 1.3\text{--}1.7$, corresponding to a relative precision between 2% and 3%. This is already very close to the accuracy required for a global fit, where the profile likelihood equivalence between 1 sigma and $\Delta\chi^2 = 1$ implies $\Delta\chi^2 < 1$ as an appropriate goal in precision. For such a complex application with $\mathcal{O}(50)$ non-overlapping signal regions,

the SCYNet implementation represents the most advanced neural network regression for the pMSSM-11 to date.

Applying the reparametrized pMSSM-11 neural network to the cMSSM and AMSB models, we found a few regions of parameter space where the network would fail to predict the correct χ^2 . As the cMSSM and AMSB models are not subsets of the pMSSM-11, there are cMSSM and AMSB parameter configurations where the network has to extrapolate to regions that were not represented in the training data. We have, however, demonstrated that such problems can be addressed systematically by additional specific training of the network.

Our results motivate the continuation and further improvement of the neural network regression approach to calculating the LHC χ^2 for BSM theories. A more accurate approximation to the true LHC likelihood can, for example, be obtained by modeling the probability density functions used for the event generation of the training sample to increase the sampling density in specific regions of parameter space. Furthermore, the treatment of systematic correlations between signal regions should be studied in more detail. The direct approach should be extended to other models, such as the pMSSM-19, so that the neural networks can be used in the corresponding global fits, and the reparameterized neural network regression should be studied further in the context of additional models.

With these improvements in mind, it is possible to decrease the mean error of the SCYNet neural network approach significantly below $\Delta\chi^2 \approx -2 \ln \mathcal{L} < 1$, thus making it a powerful tool for a large variety of global BSM fits.

Acknowledgements The work has been supported by the German Research Foundation (DFG) through the Forschergruppe *New Physics at the Large Hadron Collider* (FOR 2239), by the BMBF-FSP 101 and in part by the Helmholtz Alliance “Physics at the Terascale”. We also would like to thank Jong Soo Kim, Sebastian Liem and Roberto Ruiz de Austri for discussions in the early stages of this project. In addition, we would like to thank Ferdinand Eiteneuer for help with calculating the statistical error on the training dataset.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. Funded by SCOAP³.

Appendix A: χ^2 calculation for one single SR

In this section we explain how to calculate the profile likelihood ratio (PLR) for one SR which can be interpreted as the ‘ χ^2 ’ value used in text.

For a given signal region, the following results are available:

- Number of predicted signal events N_S ,
- statistical and total systematic error on signal events $\sigma_{N_S}^{\text{stat}}$, $\sigma_{N_S}^{\text{sys}}$,
- number of predicted SM events N_{SM} , summed over all background sources,
- statistical and total systematic error on SM events $\sigma_{N_{SM}}^{\text{stat}}$, $\sigma_{N_{SM}}^{\text{sys}}$ and
- number of experimentally observed events N_E .

At first one constructs the likelihood as follows.

$$\mathcal{L}(N_E | \mu, \nu_S, \nu_{SM}) := \frac{e^{-\lambda} \lambda^{N_E}}{N_E!} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{\nu_S^2}{2}} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{\nu_{SM}^2}{2}}. \quad (\text{A.1})$$

The first term originates from the Poisson distribution which describes the compatibility of observing N_E events if λ are expected. Here, λ —which itself is a function of the parameters μ, ν_S, ν_{SM} explained below—is given as follows:

$$\lambda(\nu_S, \nu_{SM}, \mu) := \mu N_S e^{\frac{\sigma_{N_S}}{\nu_S} \nu_S} + N_{SM} e^{\frac{\sigma_{N_{SM}}}{\nu_{SM}} \nu_{SM}}. \quad (\text{A.2})$$

The total uncertainties on signal and background

$$\sigma_{N_S} := \sqrt{(\sigma_{N_S}^{\text{stat}})^2 + (\sigma_{N_S}^{\text{sys}})^2}, \quad (\text{A.3})$$

$$\sigma_{N_{SM}} := \sqrt{(\sigma_{N_{SM}}^{\text{stat}})^2 + (\sigma_{N_{SM}}^{\text{sys}})^2}, \quad (\text{A.4})$$

are reparameterized in terms of dimensionless nuisance parameters ν_S, ν_{SM} , in a Gauss-like distribution, in Eq. (A.1). Equation (A.2) then describes lognormal¹⁰ distributions of N_S, N_{SM} with respective widths $\sigma_{N_S}, \sigma_{N_{SM}}$.

In a stochastic picture, we now wish to compare the null hypothesis H_0 and the alternative hypothesis H_1 :

- H_0 : The predicted number of signal events, i.e. $\mu = 1$, coincides well with the experimental observation,
- H_1 : Any number of signal events, i.e. any μ , coincides well with the experimental observation. (A.5)

To decide optimally between these two hypotheses, the Neyman–Pearson lemma [79] proposes the profile likelihood ratio,

$$PLR := \frac{\mathcal{L}_C}{\mathcal{L}_G}, \quad (\text{A.6})$$

¹⁰ A lognormally distributed variable is asymptotically Gaussian in the limit $\Delta X \ll X$ but forbids unphysical negative values for $\Delta X/X = \mathcal{O}(1)$.

with the respective profile likelihoods defined as

$$\mathcal{L}_C := \max_{\nu_S, \nu_{SM} \in \mathbb{R}} \mathcal{L}(\mu = 1, \nu_{SM}, \nu_S), \quad (\text{A.7})$$

$$\mathcal{L}_G := \max_{\mu, \nu_S, \nu_{SM} \in \mathbb{R}} \mathcal{L}(\mu, \nu_{SM}, \nu_S). \quad (\text{A.8})$$

Here, the constrained likelihood $\mathcal{L}_C$ maximizes $\mathcal{L}$ with respect to the nuisance parameters for the null hypothesis with fixed $\mu = 1$. In contrast, the global likelihood $\mathcal{L}_G$ also varies μ to find the global maximum of $\mathcal{L}$. Note that the range of allowed μ is not restricted here and both negative as well as values beyond unity are allowed.¹¹

According to Wilk's theorem [80], the variable

$$q_\mu := -2 \log PLR \quad (\text{A.9})$$

is asymptotically χ^2 distributed in the limit of large N_E . We can validate this statement for our setup in the simplified case of negligible uncertainties $\sigma_{N_S}, \sigma_{N_{SM}}$. Then, following the prescription above, one finds $q_\mu = (N_S + N_{SM} - N_E)^2 / (N_S + N_{SM})$, which is indeed the χ^2 distribution for one degree of freedom with observation N_E and expectation $N_S + N_{SM}$.

We therefore refer the value of q_μ whenever we use the expression ' χ^2 ' within this work.

Appendix B: Modified Z-score normalization

The χ^2 s which are our targets are called

$$y_1, \dots, y_N \in \mathbb{R}^+ \quad (\text{B.11})$$

in this appendix. We apply the Z-score normalization to them

$$\hat{y}_i = \frac{y_i - \mu}{\sigma}. \quad (\text{B.12})$$

Furthermore we define

$$\hat{y}_{\max} := \max \hat{y}_i, \quad y_{\max} := \max y_i \quad (\text{B.13})$$

and finally we normalize again

$$\hat{\hat{y}}_i = \frac{\hat{y}_i}{\hat{y}_{\max}} \in (-1, 1). \quad (\text{B.14})$$

All targets are therefore in $(-1, 1)$.

Usually one chooses $\mu = \frac{1}{N} \sum y_i$ and $\sigma = \frac{1}{N-1} \sum (y_i - \mu)^2$ but this would cause problems with the back-transformed outputs of the network. We want the back-transformed outputs of the network to be between the maximum $z_1 = 100$ and the minimum $z_0 = \text{minimum possible LHC } \chi^2$.

With the choice

$$\mu = \frac{z_0 + y_{\max}}{2}, \quad (\text{B.15})$$

z_0 corresponds to an output value of $\hat{\hat{y}} = -1$.

z_1 corresponds to an output value of $\hat{\hat{y}} = +1$.

Therefore we choose,

$$\sigma = \frac{1}{N-1} \sum_i \left(y_i - \left(\frac{1}{N} \sum_j y_j \right) \right)^2. \quad (\text{B.16})$$

Appendix C: Neural network terminology

This section aims to compactly explain the common neural network vocabulary.

- **Neuron:** Neurons are the basic building blocks of a neural network. A neuron uses the outputs of the previous layer through the activation function to generate the value that is fed forward through the network.
- **Network depth and width:** Basic architecture of the network. The width corresponds to the number of hidden neurons per layer while the depth gives the total number of layers. Note that the width can vary across the individual layers.
- **Activation function and weights/biases:** Non-linear function that is the basis of each neuron. The activation can be altered either by changing the fundamental mathematical function or the weights and biases that are the parameters of the function.
- **Cost function/Loss function:** Error function which is used to evaluate the quality of the neural network.
- **Minimizer:** To fit the data, the weights and biases need to be fixed to the values corresponding to the minimum of the loss function. The minimizer is the algorithm to search for this minimum. This is typically achieved by a calculating the gradient of the loss function for a set of given data samples and then moving down this slope.
- **Epoch:** During training, an epoch corresponds to one complete run over every sample in the training dataset.
- **Batch size:** Corresponds to the number of samples the minimizer uses before updating the weights of the network.
- **Learning rate:** The learning rate defines the step size of each update of the network.
- **Weight initialization:** Before the net can be trained, the weights and biases need to be set randomly. The range and shape of this random distribution can be altered.
- **Overfitting:** A large enough network is able to achieve zero error on any dataset. However, this incorporates fitting statistical fluctuations which allow the net to predict the training samples but not generalize to samples that are not part of the training set.
- **Dropout:** A common method used to counteract overfitting. Dropout disables a percentage of the neurons during each training iteration but the final network will have all

¹¹ Note that this is different from e.g. limit setting procedures where μ is restricted to values ≤ 1 .

neurons enabled. This results in an average of the different trained subsets of the network and such an average can cancel out the effect of overfitting.

- **Regularization:** Another common method of reducing overfitting issues. A term is added to the loss function corresponding to the sum of the squares of all weights and/or biases in the network. Consequently, the network prioritizes smoother functions and the strength of this effect is controlled by the regularization parameter.

References

1. P. Bechtel et al., Eur. Phys. J. C **76**(2), 96 (2016). doi:[10.1140/epjc/s10052-015-3864-0](https://doi.org/10.1140/epjc/s10052-015-3864-0)
2. K.J. de Vries et al., Eur. Phys. J. C **75**(9), 422 (2015). doi:[10.1140/epjc/s10052-015-3599-y](https://doi.org/10.1140/epjc/s10052-015-3599-y)
3. C. Strege, G. Bertone, G.J. Besjes, S. Caron, R. Ruiz de Austri, A. Strubig, R. Trotta, JHEP **09**, 081 (2014). doi:[10.1007/JHEP09\(2014\)081](https://doi.org/10.1007/JHEP09(2014)081)
4. A. Djouadi et al. (1998). https://inspirehep.net/record/481987/files/arXiv:hep-ph_9901246.pdf
5. M. Drees, H. Dreiner, D. Schmeier, J. Tattersall, J.S. Kim, Comput. Phys. Commun. **187**, 227 (2015). doi:[10.1016/j.cpc.2014.10.018](https://doi.org/10.1016/j.cpc.2014.10.018)
6. J.S. Kim, D. Schmeier, J. Tattersall, K. Rolbiecki, Comput. Phys. Commun. **196**, 535 (2015). doi:[10.1016/j.cpc.2015.06.002](https://doi.org/10.1016/j.cpc.2015.06.002)
7. D. Dercks, N. Desai, J.S. Kim, K. Rolbiecki, J. Tattersall, T. Weber, Comput. Phys. Commun. **221**, 383–418 (2017). doi:[10.1016/j.cpc.2017.08.021](https://doi.org/10.1016/j.cpc.2017.08.021)
8. M. Abadi et al., TensorFlow: large-scale machine learning on heterogeneous systems (2015). <http://tensorflow.org/>. Software available from tensorflow.org
9. Theano Development Team, arXiv e-prints abs/1605.02688 (2016). arXiv:[1605.02688](https://arxiv.org/abs/1605.02688)
10. F. Chollet, Keras (2015). <https://github.com/fchollet/keras>
11. A. Buckley, A. Shilton, M.J. White, Comput. Phys. Commun. **183**, 960 (2012). doi:[10.1016/j.cpc.2011.12.026](https://doi.org/10.1016/j.cpc.2011.12.026)
12. N. Bornhauser, M. Drees, Phys. Rev. D **88**, 075016 (2013). doi:[10.1103/PhysRevD.88.075016](https://doi.org/10.1103/PhysRevD.88.075016)
13. S. Caron, J. S. Kim, K. Rolbiecki, R. Ruiz de Austri, B. Stienen, Eur. Phys. J. C **77**(4), 257 (2017). arXiv:[1605.02797](https://arxiv.org/abs/1605.02797). doi:[10.1140/epjc/s10052-017-4814-9](https://doi.org/10.1140/epjc/s10052-017-4814-9)
14. C.E. Rasmussen, *Gaussian Processes in Machine Learning* (Springer, Berlin, 2004), pp. 63–71. doi:[10.1007/978-3-540-28650-9_4](https://doi.org/10.1007/978-3-540-28650-9_4)
15. G. Bertone, M. P. Deisenroth, J. S. Kim, S. Liem, R. Ruiz de Austri, M. Welling. (2016). arXiv:[1611.02704](https://arxiv.org/abs/1611.02704)
16. L. Randall, R. Sundrum, Nucl. Phys. B **557**, 79 (1999). doi:[10.1016/S0550-3213\(99\)00359-4](https://doi.org/10.1016/S0550-3213(99)00359-4)
17. G.F. Giudice, M.A. Luty, H. Murayama, R. Rattazzi, JHEP **12**, 027 (1998). doi:[10.1088/1126-6708/1998/12/027](https://doi.org/10.1088/1126-6708/1998/12/027)
18. J.A. Aguilar-Saavedra et al., Eur. Phys. J. C **46**, 43 (2006). doi:[10.1140/epjc/s2005-02460-1](https://doi.org/10.1140/epjc/s2005-02460-1)
19. W. Porod, Comput. Phys. Commun. **153**, 275 (2003). doi:[10.1016/S0010-4655\(03\)00222-4](https://doi.org/10.1016/S0010-4655(03)00222-4)
20. K.A. Olive et al., Chin. Phys. C **38**, 090001 (2014). doi:[10.1088/1674-1137/38/9/090001](https://doi.org/10.1088/1674-1137/38/9/090001)
21. TEW Group (2012). arXiv:[1204.0042](https://arxiv.org/abs/1204.0042)
22. Y. Amhis et al. (2012). arXiv:[1207.1158](https://arxiv.org/abs/1207.1158)
23. CMS, LHCb Collaborations (2013) CMS-PAS-BPH-13-007, LHCb-CONF-2013-012, CERN-LHCb-CONF-2013-012
24. J. Beringer et al., Phys. Rev. D **86**, 010001 (2012). doi:[10.1103/PhysRevD.86.010001](https://doi.org/10.1103/PhysRevD.86.010001)
25. V. Khachatryan et al., Nature **522**, 68 (2015). doi:[10.1038/nature14474](https://doi.org/10.1038/nature14474)
26. G. Aad et al., JHEP **04**, 169 (2014). doi:[10.1007/JHEP04\(2014\)169](https://doi.org/10.1007/JHEP04(2014)169)
27. G. Aad et al., JHEP **05**, 071 (2014). doi:[10.1007/JHEP05\(2014\)071](https://doi.org/10.1007/JHEP05(2014)071)
28. Search for supersymmetry in events with four or more leptons in 21 fb⁻¹ of pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector. Technical Report. ATLAS-CONF-2013-036, CERN, Geneva (2013). <https://cds.cern.ch/record/1532429>
29. G. Aad et al., JHEP **10**, 189 (2013). doi:[10.1007/JHEP10\(2013\)189](https://doi.org/10.1007/JHEP10(2013)189)
30. G. Aad et al., JHEP **06**, 124 (2014). doi:[10.1007/JHEP06\(2014\)124](https://doi.org/10.1007/JHEP06(2014)124)
31. G. Aad et al., JHEP **06**, 035 (2014). doi:[10.1007/JHEP06\(2014\)035](https://doi.org/10.1007/JHEP06(2014)035)
32. G. Aad et al., JHEP **09**, 176 (2014). doi:[10.1007/JHEP09\(2014\)176](https://doi.org/10.1007/JHEP09(2014)176)
33. G. Aad et al., JHEP **11**, 118 (2014). doi:[10.1007/JHEP11\(2014\)118](https://doi.org/10.1007/JHEP11(2014)118)
34. G. Aad et al., Phys. Rev. D **90**(5), 052008 (2014). doi:[10.1103/PhysRevD.90.052008](https://doi.org/10.1103/PhysRevD.90.052008)
35. G. Aad et al., Eur. Phys. J. C **75**(7), 299 (2015). doi:[10.1140/epjc/s10052-015-3517-3](https://doi.org/10.1140/epjc/s10052-015-3517-3) [Erratum: Eur. Phys. J. C **75**(9), 408 (2015)]
36. G. Aad et al., Eur. Phys. J. C **75**(7), 318 (2015). doi:[10.1140/epjc/s10052-015-3661-9](https://doi.org/10.1140/epjc/s10052-015-3661-9) [Erratum: Eur. Phys. J. C **75**(10), 463 (2015)]
37. Search for gluinos in events with an isolated lepton, jets and missing transverse momentum at $\sqrt{s} = 13$ with the ATLAS detector. Technical Report. ATLAS-CONF-2015-076, CERN, Geneva (2015). <https://cds.cern.ch/record/2114848>
38. G. Aad et al., Eur. Phys. J. C **76**(5), 259 (2016). doi:[10.1140/epjc/s10052-016-4095-8](https://doi.org/10.1140/epjc/s10052-016-4095-8)
39. M. Aaboud et al., Eur. Phys. J. C **76**(7), 392 (2016). doi:[10.1140/epjc/s10052-016-4184-8](https://doi.org/10.1140/epjc/s10052-016-4184-8)
40. A search for Supersymmetry in events containing a leptonically decaying Z boson, jets and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector. Technical Report. ATLAS-CONF-2015-082, CERN, Geneva (2015). <http://cds.cern.ch/record/2114854>
41. M. Aaboud et al., Phys. Rev. D **94**(3), 032005 (2016). doi:[10.1103/PhysRevD.94.032005](https://doi.org/10.1103/PhysRevD.94.032005)
42. Search for production of vector-like top quark pairs and of four top quarks in the lepton-plus-jets final state in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. Technical Report. ATLAS-CONF-2016-013, CERN, Geneva (2016). <http://cds.cern.ch/record/2140998>
43. Search for pair-production of gluinos decaying via stop and sbottom in events with b-jets and large missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector. Technical Report. ATLAS-CONF-2015-067, CERN, Geneva (2015). <http://cds.cern.ch/record/2114839>
44. V. Khachatryan et al., JHEP **12**, 013 (2016). arXiv:[1607.00915](https://arxiv.org/abs/1607.00915). doi:[10.1007/JHEP12\(2016\)013](https://doi.org/10.1007/JHEP12(2016)013)
45. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, JHEP **07**, 079 (2014). doi:[10.1007/JHEP07\(2014\)079](https://doi.org/10.1007/JHEP07(2014)079). arXiv:[1405.0301](https://arxiv.org/abs/1405.0301)
46. P.M. Nadolsky, H.L. Lai, Q.H. Cao, J. Huston, J. Pumplin, D. Stump, W.K. Tung, C.P. Yuan, Phys. Rev. D **78**, 013004 (2008). doi:[10.1103/PhysRevD.78.013004](https://doi.org/10.1103/PhysRevD.78.013004)
47. T. Sjöstrand, S. Mrenna, P.Z. Skands, JHEP **05**, 026 (2006). doi:[10.1088/1126-6708/2006/05/026](https://doi.org/10.1088/1126-6708/2006/05/026)
48. G. Aad et al., Phys. Rev. D **93**(5), 052002 (2016). doi:[10.1103/PhysRevD.93.052002](https://doi.org/10.1103/PhysRevD.93.052002)
49. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, Comput. Phys. Commun. **191**, 159 (2015). doi:[10.1016/j.cpc.2015.01.024](https://doi.org/10.1016/j.cpc.2015.01.024)
50. N. Desai, P.Z. Skands, Eur. Phys. J. C **72**, 2238 (2012). doi:[10.1140/epjc/s10052-012-2238-0](https://doi.org/10.1140/epjc/s10052-012-2238-0)
51. W. Beenakker, R. Hopker, M. Spira, P.M. Zerwas, Nucl. Phys. B **492**, 51 (1997). doi:[10.1016/S0550-3213\(97\)80027-2](https://doi.org/10.1016/S0550-3213(97)80027-2)
52. A. Kulesza, L. Motyka, Phys. Rev. Lett. **102**, 111802 (2009). doi:[10.1103/PhysRevLett.102.111802](https://doi.org/10.1103/PhysRevLett.102.111802)

53. A. Kulesza, L. Motyka, Phys. Rev. D **80**, 095004 (2009). doi:[10.1103/PhysRevD.80.095004](https://doi.org/10.1103/PhysRevD.80.095004)
54. W. Beenakker, S. Brensing, M. Krämer, A. Kulesza, E. Laenen, I. Niessen, JHEP **12**, 041 (2009). doi:[10.1088/1126-6708/2009/12/041](https://doi.org/10.1088/1126-6708/2009/12/041)
55. W. Beenakker, S. Brensing, M. Krämer, A. Kulesza, E. Laenen, L. Motyka, I. Niessen, Int. J. Mod. Phys. A **26**, 2637 (2011). doi:[10.1142/S0217751X11053560](https://doi.org/10.1142/S0217751X11053560)
56. W. Beenakker, M. Krämer, T. Plehn, M. Spira, P.M. Zerwas, Nucl. Phys. B **515**, 3 (1998). doi:[10.1016/S0550-3213\(98\)00014-5](https://doi.org/10.1016/S0550-3213(98)00014-5)
57. W. Beenakker, S. Brensing, M. Krämer, A. Kulesza, E. Laenen, I. Niessen, JHEP **08**, 098 (2010). doi:[10.1007/JHEP08\(2010\)098](https://doi.org/10.1007/JHEP08(2010)098)
58. H.K. Dreiner, M. Krämer, J. Tattersall, Europhys. Lett. **99**, 61001 (2012). doi:[10.1209/0295-5075/99/61001](https://doi.org/10.1209/0295-5075/99/61001)
59. H. Dreiner, M. Krämer, J. Tattersall, Phys. Rev. D **87**(3), 035006 (2013). doi:[10.1103/PhysRevD.87.035006](https://doi.org/10.1103/PhysRevD.87.035006)
60. M. Cacciari, G.P. Salam, Phys. Lett. B **641**, 57 (2006). doi:[10.1016/j.physletb.2006.08.037](https://doi.org/10.1016/j.physletb.2006.08.037)
61. M. Cacciari, G.P. Salam, G. Soyez, Eur. Phys. J. C **72**, 1896 (2012). doi:[10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2)
62. M. Cacciari, G.P. Salam, G. Soyez, JHEP **04**, 063 (2008). doi:[10.1088/1126-6708/2008/04/063](https://doi.org/10.1088/1126-6708/2008/04/063)
63. J. Cao, L. Shang, J.M. Yang, Y. Zhang, JHEP **06**, 152 (2015). doi:[10.1007/JHEP06\(2015\)152](https://doi.org/10.1007/JHEP06(2015)152)
64. M.R. Buckley, J.D. Lykken, C. Rogan, M. Spiropulu, Phys. Rev. D **89**(5), 055020 (2014). doi:[10.1103/PhysRevD.89.055020](https://doi.org/10.1103/PhysRevD.89.055020)
65. J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, M. Selvaggi, JHEP **02**, 057 (2014). doi:[10.1007/JHEP02\(2014\)057](https://doi.org/10.1007/JHEP02(2014)057)
66. D.P. Kingma, J. Ba, CoRR. **abs/1412.6980** (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
67. E. Jones, T. Oliphant, P. Peterson et al. SciPy: open source scientific tools for Python (2001). <http://www.scipy.org/> [Online; accessed <today>]
68. A. Buckley, Eur. Phys. J. C **75**(10), 467 (2015). doi:[10.1140/epjc/s10052-015-3638-8](https://doi.org/10.1140/epjc/s10052-015-3638-8)
69. W. Beenakker, C. Borschensky, M. Krämer, A. Kulesza, E. Laenen, S. Marzani, J. Rojo, Eur. Phys. J. C **76**(2), 53 (2016). doi:[10.1140/epjc/s10052-016-3892-4](https://doi.org/10.1140/epjc/s10052-016-3892-4)
70. M. Krämer, T. Plehn, M. Spira, P.M. Zerwas, Phys. Rev. Lett. **79**, 341 (1997). doi:[10.1103/PhysRevLett.79.341](https://doi.org/10.1103/PhysRevLett.79.341)
71. M. Krämer, T. Plehn, M. Spira, P.M. Zerwas, Phys. Rev. D **71**, 057503 (2005). doi:[10.1103/PhysRevD.71.057503](https://doi.org/10.1103/PhysRevD.71.057503)
72. A. Alves, O. Eboli, T. Plehn, Phys. Lett. B **558**, 165 (2003). doi:[10.1016/S0370-2693\(03\)00266-1](https://doi.org/10.1016/S0370-2693(03)00266-1)
73. T. Plehn, Phys. Rev. D **67**, 014018 (2003). doi:[10.1103/PhysRevD.67.014018](https://doi.org/10.1103/PhysRevD.67.014018)
74. A. Alves, T. Plehn, Phys. Rev. D **71**, 115014 (2005). doi:[10.1103/PhysRevD.71.115014](https://doi.org/10.1103/PhysRevD.71.115014)
75. G.P. Lepage, J. Comput. Phys. **27**, 192 (1978). doi:[10.1016/0021-9991\(78\)90004-9](https://doi.org/10.1016/0021-9991(78)90004-9)
76. V. Nair, G.E. Hinton, in *Proceedings of the 27th International Conference on Machine Learning* (2010), pp. 807–814
77. I. Sutskever, J. Martens, G.E. Dahl, G.E. Hinton, in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 (2013), pp. 1139–1147
78. T. Dozat (2015). http://cs229.stanford.edu/proj2015/054_report.pdf
79. J. Neyman, E.S. Pearson, *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, p. 289 (1933). <http://www.jstor.org/stable/91247>
80. S.S. Wilks, Ann. Math. Stat. **9**(1), 60 (1938). doi:[10.1214/aoms/1177732360](https://doi.org/10.1214/aoms/1177732360)