

Searching for physics beyond the Standard Model at the LHC in the framework of the pMSSM-11 by using a neural network approach

by Jan Schütte-Engel

supervised by Prof. Dr. Michael Krämer

September 2016

Master thesis in the Department of Theoretical Physics
at the
RWTH Aachen University

Contents

1	Introduction	1
2	The Standard Model	3
2.1	The parameters of the SM	6
2.2	Problems of the SM	6
2.2.1	Anomalous magnetic dipole moment of muon	6
2.2.2	Dark Matter	7
2.2.3	Hierarchy problem	7
3	Supersymmetry and the Minimal Supersymmetric extension of the Standard Model (MSSM)	9
3.1	The MSSM	14
3.2	The constrained MSSM	19
3.3	The phenomenological MSSM	20
3.3.1	pMSSM-19	20
3.3.2	pMSSM-11	20
4	Investigating the pMSSM-11	22
4.1	Fittino	22
4.2	The strategy to make the LHC predictions faster	25
5	Simulating LHC events	27
5.1	Simulating 8 TeV LHC events	29
5.2	Simulating 13 TeV LHC events	31
5.3	Checkmate	33
5.4	Obtaining an overall LHC χ^2	35
5.5	The sampling algorithms	37
6	Multivariate interpolation	39
6.1	Nearest neighbour interpolation	40
6.2	Neural networks	40
6.2.1	Human neural networks	40
6.2.2	Artificial neural networks	41
6.2.2.1	A model for a neuron	42
6.2.2.2	A model for a neural network	43
7	LHC neural networks	56
7.1	Tests with a 2 parameter cMSSM	56
7.2	Neural nets for the pMSSM-11	61
7.2.1	LHC 8 TeV χ^2 neural net	61
7.2.1.1	χ_{tot}^2	61

7.2.1.2	χ^2_{disjoint}	69
7.2.2	LHC 13 TeV χ^2 neural net	70
7.2.2.1	χ^2_{tot}	70
7.2.2.2	χ^2_{disjoint}	72
7.2.3	Neural net for the number of events in one SR	74
7.2.4	χ^2 s for pMSSM-19 from pMSSM-11 neural net	77
7.3	Rule of thumbs for neural networks with one output	78
7.4	Outlook	79
8	Global fit results	81
9	Conclusion	84
10	Acknowledgement	86
Appendix		87
A.1	Conventions	87
A.2	Calculation of the profile likelihood ratio for one SR	87
A.3	Groups of disjoint SRs	88
A.4	Production of 8 TeV events (second calss) in detail	93
A.5	Production of 13 TeV events in detail	96
A.6	Implementation of 13 TeV analyses into Checkmate	98
A.6.1	Search for gluinos in events with an isolated lepton, jets and missing transverse momentum at 13 TeV	99
A.6.1.1	hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1105, 865, 625) \text{GeV}$ simplified model	99
A.6.1.2	hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1385, 705, 25) \text{GeV}$ simplified model	100
A.6.1.3	hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1200, 160, 60) \text{GeV}$ simplified model	101
A.6.1.4	soft SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1000, 110, 60) \text{GeV}$ simplified model	103
A.6.2	Search for supersymmetry at 13 TeV in final states with jets and two same-sign leptons or three leptons	103
A.6.2.1	SR0b3j	103
A.6.2.2	SR1b	104
A.6.2.3	SR3b with events from model in figure 1d	104
List of literature		113

1 Introduction

Supersymmetry (SUSY) is an extension of the Standard Model (SM) symmetries which introduces fermionic symmetry generators[1]. The minimal supersymmetric extension of the SM (MSSM) introduces for each Standard Model particle a superpartner whose spin differs by 1/2. In the MSSM SUSY has to be broken in order to get electroweak symmetry breaking (EWSB). The breaking mechanism is not known and therefore one introduces in the MSSM Lagrangian explicit SUSY breaking terms. Along with these explicit SUSY breaking terms one introduces 105 [2] new parameters. A lot of these parameters can be set to zero due to resulting flavour changing neutral currents (FCNC) or large CP-violations which cannot be observed in experiments [2]. In this thesis the phenomenological MSSM with 11 parameters (pMSSM-11) will be investigated. The prefix phenomenological refers to the fact that the parameters are specified at the electroweak scale.

The general interest in SUSY is so large because the MSSM solves several problems of the SM. For example SUSY gives a solution to the hierarchy problem[3], provides a dark matter candidate and can also solve experimental and theoretical discrepancies for low energy observables. Although there are big efforts for finding supersymmetry signatures at the LHC no significant SUSY signatures have been observed yet. Nevertheless due to the large amount of problems which could be solved if SUSY is realized in nature there is still work in progress to find SUSY signatures at the LHC.

A big problem in the SUSY signature searches is that the parameter space of the pMSSM-11 is huge which makes the direct search at the LHC for SUSY more complicated. A common approach is to constrain the parameter space of the pMSSM-11 such that LHC searches can concentrate on the allowed parameter ranges for their searches. To constrain the pMSSM-11 parameter space we compare for as many as possible observables the experimental measurements and the theoretical predictions of the pMSSM-11. One can compare low energy electroweak precision observables (like $g - 2$ of the muon) or Higgs observables. Furthermore one can include LHC observables which are the numbers of events in signal regions (SR) of LHC analyses. At last one can compare the theoretical predictions with the experimental measurements for cosmological observables. The comparison of theory predictions with the experimental values for all observables constrain the parameter space. But due to the fact that the parameter space is large we have to scan a lot of parameter points and therefore the theory values for all observables have to be predictable in order of seconds. This is the case for all observables except for the LHC observables where it takes order of hours to get the theoretical predictions for one pMSSM-11 parameter point. For the LHC observables one has to simulate LHC events for each pMSSM-11 parameter point which is a time consuming task. In a typical parameter scan one wants to scan billions of points which is not possible even on the largest computer cluster if the predictions for the LHC observables take order of hours. In this master thesis we develop a neural network which compares the LHC theory predictions and experimental values in order of milliseconds. The quantity which gives indication about the degree of agreement between theory prediction and experimental measurement will be a χ^2 . Therefore the neural network will output the LHC χ^2 . In order to train the neural network one has to simulate LHC events. This is of course also time consuming, but can be done in a reasonable time frame.

This thesis is structured as follows: chapter 2 will give a short overview over the SM and its problems. In chapter 3 the basic ideas of SUSY will be explained. Furthermore the parameters of the MSSM and pMSSM-11 are given. The next chapter (4) goes into the details of the investigation strategy for the pMSSM-11. In chapter 5 the details about the LHC event generation are given. The chapter is followed by a general chapter (6) about multivariate interpolation which focuses on the neural network interpolation. In chapter 7 the LHC χ^2 neural network results will be presented. In chapter 8 we give first preliminary results for the preferred parameter regions in the pMSSM-11 by taking into account many observables.

2 The Standard Model

The Standard Model (SM) is the most powerful theory in particle physics. The SM Lagrangian is based on Poincare symmetry and the internal symmetry group $SU(3) \times SU(2) \times U(1)$. The elementary particles of the SM, which have all been confirmed experimentally, are listed in table 2.1.

elementary particles	electrical	weak	strong
Leptons (e^- , τ^- , μ^-)	✓	✓	✗
Neutrinos (ν_{e^-} , ν_{τ^-} , ν_{μ^-})	✗	✓	✗
Quarks (u,d,c,s,t,b)	✓	✓	✓

Table 2.1: Elementary particles of the SM. The table also shows which of the three fundamental forces (that the SM can describe) act on which particle.

The SM describes only the weak, electrical and strong force. The description of the gravitational force is beyond the SM.

Left handed leptons are grouped into $SU(2)$ doublets

$$L_L^1 = \begin{pmatrix} \nu_{e,L} \\ e_L \end{pmatrix}, L_L^2 = \begin{pmatrix} \nu_{\mu,L} \\ \mu_L \end{pmatrix}, L_L^3 = \begin{pmatrix} \nu_{\tau,L} \\ \tau_L \end{pmatrix}, \quad (2.1)$$

where every doublet contains two fields which both are 4 component Dirac spinor. The right handed leptons are not grouped into doublets, because the weak force is chiral, e.g. it distinguishes between left and right handed leptons. With the notation

$$l = (e, \mu, \tau), \quad \nu = (\nu_e, \nu_\mu, \nu_\tau) \quad (2.2)$$

we get for example a right handed electron with l_R^1 . Left handed quarks are also grouped into $SU(2)$ doublets:

$$Q_L^1 = \begin{pmatrix} u_L \\ d_L \end{pmatrix}, Q_L^2 = \begin{pmatrix} c_L \\ s_L \end{pmatrix}, Q_L^3 = \begin{pmatrix} t_L \\ b_L \end{pmatrix}, \quad (2.3)$$

where actually each object in the $SU(2)$ doublet of course has a $SU(3)$ color index which is dropped here. Furthermore we introduce the up-type and down-type quarks

$$u := (u, c, t), \quad d := (d, s, b). \quad (2.4)$$

Besides the known electric charge the objects grouped in doublets will have a Isospin quantum number. The upper component of a doublet has $I^3 = \frac{1}{2}$ Isospin and the lower component has $I^3 = -\frac{1}{2}$. E.g. $I^3(\nu_{e,L}) = \frac{1}{2}$ and $I^3(e_L) = -\frac{1}{2}$. The right handed elementary particles have no Isospin quantum number, because they transform trivially under $SU(2)$.

In table 2.2 we will specify the representations of $SU(3) \times SU(2) \times U(1)$ under which the elementary particle fields transform.

As a consequence of the gauge symmetry we have to introduce gauge fields G_μ^a , $a = 1..8 = 3^2 - 1$; W_μ^a , $a = 1...3 = 2^2 - 1$ and B_μ . There are 8 Gluon gauge fields G_μ^a corresponding to

elementary particle	$U(1)$	$SU(2)$	$SU(3)$
left handed leptons L_L^j	-1	2	1
right handed l_R^j	-2	1	1
left handed Quarks Q_L^j	$\frac{1}{3}$	2	3
right handed up-type Quarks u_R^j	$\frac{4}{3}$	1	3
right handed down-type Quarks d_R^j	$-\frac{2}{3}$	1	3

Table 2.2: Representations under which the elementary particle fields transform. We have used the relation $Q = I^3 + \frac{Y}{2}$. Furthermore in the $SU(2)$ and $SU(3)$ column we have set a 1 if the corresponding fields transform according to the trivial representation. We have set a 2/3 in the $SU(2)/SU(3)$ column if the particles transform with the fundamental representation. In the $U(1)$ column we have written down the corresponding Y parameter.

$SU(3)$, 3 W boson gauge fields W_μ^a corresponding to $SU(2)$ and one gauge field corresponding to $U(1)$ namely B_μ .

The Lagrangian containing the gauge fields is:

$$\mathcal{L}_G = -\frac{1}{4}F_{1,\mu\nu}F_1^{\mu\nu} - \frac{1}{4}F_{2,\mu\nu}^a F_2^{a,\mu\nu} - \frac{1}{4}F_{3,\mu\nu}^a F_3^{a,\mu\nu}, \quad (2.5)$$

where $F_{N,\mu\nu}^a$, $N = 1, 2, 3$ is the generalized field strength tensor, whose exact form can be looked up in any basic SM text book.

One could include more terms here which respect all symmetries of the SM, but these terms would cause CP violations which have not been measured. This is also related to the *strong CP problem*.

In the following the kinetic fermion terms will be written down:

$$\mathcal{L}_{F,kin} = \sum_{i=1}^3 \bar{L}_L^i iD^\mu L_L^i + \bar{Q}_L^i iD^\mu Q_L^i + \bar{l}_R^i iD^\mu l_R^i + \bar{d}_R^i iD^\mu d_R^i + \bar{u}_R^i iD^\mu u_R^i, \quad (2.6)$$

where D_μ is the covariant derivative, whose exact form can be looked up in any SM textbook. The Higgs boson is introduced in order to give the SM particles masses after electroweak symmetry breaking. The Higgs will be a doublet, whereas we choose $Y = 1$. Then due to the relation $Q = I^3 + \frac{Y}{2}$ we label the elements of the Higgs Boson as follows:

$$\Phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix}. \quad (2.7)$$

The Higgs will transfrom trivial under $SU(3)$. The most general gauge and Poincare invariant Lagrangian which leads to a renomalizable theory is:

$$\mathcal{L}_H = (D_\mu \Phi)^\dagger D^\mu \Phi - V(\Phi), \quad (2.8)$$

with

$$V(\Phi) = \frac{\lambda}{4}(\Phi^\dagger \Phi)^2 - \mu^2 \Phi^\dagger \Phi, \quad (2.9)$$

where μ, λ have to be larger than zero such that the Higgs field can acquire a nonzero vacuum expectation value (vev)¹. With the help of the Higgs field we can now introduce new possible

¹This is necessary such that spontaneous symmetry breaking (SSB) can happen.

gauge invariant terms. They are called Yukawa coupling terms, because we couple the Higgs to fermions.

$$\mathcal{L}_Y = \sum_{i,j=1}^3 g_{ij} \bar{L}_L^i \cdot \Phi l_R^j + h_{ij} \bar{Q}_L^i \cdot \Phi d_R^j + \tilde{h}_{ij} \bar{Q}_L^i \cdot \Phi^c u_R^j + \text{h.c.} \quad (2.10)$$

where Φ^c is the charge conjugated field with $Y = -1$

$$\Phi^c = i\sigma_2 \Phi^*. \quad (2.11)$$

In formula 2.6 and 2.10 we have no terms with right handed neutrinos, because in the original SM, as described here, neutrinos are massless.

The Higgs doublet can be reparameterized:

$$\Phi = \begin{pmatrix} \phi^+ \\ \frac{1}{\sqrt{2}}(v + H + i\chi) \end{pmatrix}. \quad (2.12)$$

Where H and χ are two real fields. H is the Higgs field and v is the vacuum expectation value (vev), $\frac{4\mu^2}{\lambda} =: v^2 \neq 0$.

The nonzero vev makes electroweak symmetry breaking possible. The Lagrangian is actually $SU(3) \times SU(2) \times U(1)$ invariant, but after inserting 2.12 and writing out all terms the remaining symmetry is $SU(3) \times U(1)$. In other cases one has explicit symmetry breaking. This is the case if one has right from the beginning terms in the Lagrangian which do not respect a certain symmetry. We will encounter this phenomenon in supersymmetric Lagrangians.

Inserting the reparameterized Higgs field into the Yukawa Lagrangian gives mass terms for all fermions (after a biunitary transformation). If we now plug (2.12) into the Higgs Lagrangian we will obtain a lot of terms, but one of these is:

$$\mathcal{L}_H = -\frac{1}{2}m_H^2 H^2 + \dots, \quad (2.13)$$

where $m_H = \sqrt{2}\mu$ is the tree level Higgs mass.

Gauge bosons transform according to the adjoint representation², which is real. Therefore the gauge fields are real fields.

field	$U(1)$	$SU(2)$	$SU(3)$
Higgs Boson H	1	2	1
B_μ	0	1	1
W_μ^a	0	3	1
G_μ^a	0	1	8

Table 2.3: Representations under which the Higgs and the gauge bosons transform. The adjoint representation of $SU(3)$ is labeled with 8 and the adjoint representation of $SU(2)$ with 3.

We have now introduced all parts of the SM. The whole SM Lagrangian is

$$\mathcal{L}_{SM} = \mathcal{L}_G + \mathcal{L}_{F,kin} + \mathcal{L}_H + \mathcal{L}_Y. \quad (2.14)$$

²This is actually only true for global transformations, but not for local transformations. The internal symmetry transformations in the SM are local transformations, nevertheless one says that the gauge bosons transform according to the adjoint representation.

Of course the mass terms have not been diagonalized yet, but this can be looked up in any text book about the SM. The main information of this section was taken from lecture notes of Michael Czakon [4].

2.1 The parameters of the SM

There are 19 free parameters in the original SM without neutrino masses.

- 3 coupling constants
- 4 parameters of Cabibbo–Kobayashi–Maskawa (CKM) matrix(3 mixing angles and one phase)³.
- μ and λ .⁴
- 3 lepton tree level masses : m_e, m_μ, m_τ
- 6 quark tree level masses : $m_u, m_d, m_c, m_s, m_t, m_b$ ⁵
- The additional possible terms in (2.5) would introduce a parameter often called θ . Experiments suggest that this parameter is zero (strong CP problem). Therefore one often says that the SM has only 18 parameters under the assumption that this parameter is zero from the beginning.

2.2 Problems of the SM

This section gives some examples of experimental and theoretical problems of the SM. Subsection 2.2.1 gives an overview about the discrepancies between the experimental an theoretical predicted values for the anomalous magnetic dipole moment of the muon. In subsection 2.2.2 the ability of the SM to give a dark matter candidate is discussed. Finally in subsection 2.2.3 the rather technical Hierarchy problem of the SM will be examined.

2.2.1 Anomalous magnetic dipole moment of muon

Currently the experimental measurement⁶ and the theoretical prediction differ by 3-4 standard deviations [5]. Therefore the observable anomalous magnetic moment of the muon gives strong evidence for physics beyond the SM.

At the present status people are still skeptical about this big deviation and therefore a new experiment at Fermilab [6] is about to confirm the experimental result.

Martin and Wells [7] showed that supersymmetry can give large contributions to the SM anomalous magnetic dipole moment in some parameter regions, such that the predictions agree with the experimental values from Brookhaven E821.

³The CKM matrix appears after applying the bi-unitary transformations on the Lepton and Quark fields (which diagonalize the mass terms) in the kinetic Lagrangian.

⁴The two parameters can be replaced by physical Higgs mass and the vev.

⁵The tree level lepton and quark masses have not been introduced yet, because we did not diagonalize the mass terms after EWSB in the Yukawa Lagrangian. One can look up how this works in any SM textbook

⁶from Brookhaven E821

2.2.2 Dark Matter

From cosmological observations we have strong evidence for some kind of matter which does not interact via the electromagnetic and strong force, but only via the weak and gravitational force. Particles with this behavior are not part of the traditional SM⁷.

There are several possibilities to extend the SM with a 'dark matter candidate'. One possibility is to extend the SM with right handed neutrinos [8] like suggested for the neutrino mass problem. The supersymmetric extension of the SM also provides a dark matter candidate. The lightest neutralino χ_1^0 has no electric and color charge and interacts only via the weak and gravitational force.

2.2.3 Hierarchy problem

In this section we briefly describe the hierarchy problem for the Higgs mass in the SM. The bare mass parameter for the Higgs mass in the SM is m_H and the corresponding term in the SM Lagrangian is:

$$-\frac{1}{2}m_H^2 H^2. \quad (2.15)$$

The tree level Higgs mass receives large loop corrections from fermion loops (see figure 2.1 a)[3]

$$\Delta m_H^2 = -\frac{|\lambda_f|^2}{8\pi^2} \Lambda_{UV}^2 + \dots \quad (2.16)$$

where Λ_{UV} is the momentum cutoff which regulates the one loop integral and λ_f is the coupling constant for the Higgs-fermion-fermion vertex $-\lambda_f H \bar{f} f$. All fermions of the SM can play the role of the fermion f .

Typically Λ_{UV} is of the order of the Planck scale that is 30 orders of magnitude larger than the electroweak scale [3, p. 4]. Therefore large tuning is necessary which can be seen as unnatural. Supersymmetry provides for every fermion a super partner which is a boson. The introduction of the scalar super partners leads to diagrams of the form 2.1 (b) with the scalar superpartner in the loop. If the Higgs-Higgs-scalar-scalar vertex ist $-\lambda_S |H|^2 |S|^2$ these scalar loop diagrams give corrections to the tree level Higgs mass[3, p. 4]:

$$\Delta m_H^2 = \frac{\lambda_S}{16\pi^2} [\Lambda_{UV}^2 - 2m_S^2 \ln \Lambda_{UV}/m_S + \dots], \quad (2.17)$$

where m_S is the mass of the scalar superpartner of the fermion f .



Figure 2.1: Loop diagrams for the Hierarchy problem. The graphic was taken from [3].

⁷The only particles of the traditional SM which come into consideration are the neutrinos. But due to the fact that they are massless in the traditional SM they cannot constitute a dark matter candidate which interact also gravitationally.

In the framework of the minimal supersymmetric standard model (MSSM) one can show that the quadratic terms in (2.16) and (2.17) cancel. Therefore we need only small finetuning depending on the mass of the scalar super partners of the fermions. How much finetuning one thinks is acceptable is always a matter of taste.

3 Supersymmetry and the Minimal Supersymmetric extension of the Standard Model (MSSM)

The SM Lagrangian was construed on the basis two symmetries:

- Poincare symmetry

The Poincare symmetry generators $P_\mu, J_{\mu\nu}$ fulfil the following commutation relations [1]

$$\begin{aligned} [P^\mu, P^\nu] &= 0 \\ [J^{\mu\nu}, P^\sigma] &= i(P^\mu \eta^{\nu\sigma} - P^\nu \eta^{\mu\sigma}) \\ [J^{\mu\nu}, J^{\rho\sigma}] &= i(J^{\mu\sigma} \eta^{\nu\rho} + J^{\nu\rho} \eta^{\mu\sigma} - J^{\mu\rho} \eta^{\nu\sigma} - J^{\nu\sigma} \eta^{\mu\rho}). \end{aligned} \quad (3.1)$$

- Internal symmetry

The SM contains internal symmetries of the group $SU(3) \otimes SU(2) \otimes U(1)$, whose generators fulfil the following commutation relations

$$[T_N^a, T_N^b] = if_N^{abc} T_N^c, \quad N = 2, 3 \text{ and } a, b, c = 1..N^2 - 1. \quad (3.2)$$

A central role in the development of supersymmetry was the **Coleman and Mandula theorem**. It states that 'the only possible Lie algebra of symmetry generators (for a non-trivial theory) consists of the generators P^μ and $J^{\mu\nu}$ of translations and homogeneous Lorentz transformations, together with possible internal symmetry generators (T), which commute with P^μ and $J^{\mu\nu}$ ', [9, p.12].

$$[P^\mu, T] = 0, \quad [J^{\mu\nu}, T] = 0 \quad (3.3)$$

For example in the SM we have:

$$\begin{aligned} [P^\mu, P^\nu] &= 0 \\ [J^{\mu\nu}, P^\sigma] &= i(P^\mu \eta^{\nu\sigma} - P^\nu \eta^{\mu\sigma}) \\ [J^{\mu\nu}, J^{\rho\sigma}] &= i(J^{\mu\sigma} \eta^{\nu\rho} + J^{\nu\rho} \eta^{\mu\sigma} - J^{\mu\rho} \eta^{\nu\sigma} - J^{\nu\sigma} \eta^{\mu\rho}) \\ [T_N^a, T_N^b] &= if_N^{abc} T_N^c, \quad N = 2, 3 \text{ and } a, b, c = 1..N^2 - 1 \\ [P^\mu, T_N^a] &= 0, \quad [J^{\mu\nu}, T_N^a] = 0. \end{aligned} \quad (3.4)$$

From the Coleman Mandula theorem it follows that we cannot extend the Poincare algebra with generators which do not commute with the P^μ and $J^{\mu\nu}$ generators. But this is exactly what we want in supersymmetry. In order to achieve this we have to elude the Coleman Mandula theorem. The Haag–Lopuszanski–Sohnius theorem [10] (which can be seen as an extension of the Coleman Mandula theorem) states that the Poincare algebra can be extended by generators which do not commute with P^μ or $J^{\mu\nu}$ by introducing fermionic generators $Q_r, r = 1, \dots, \mathcal{N}$ (which fulfil anticommutation relations). The Coleman Mandula theorem is eluded in this case,

because we do not have a Lie algebra anymore.¹

Before we will consider the extended Poincare algebra with the new fermionic generators we will take a look at left and right handed spinors.

Left and right handed spinors transform according to the two basic representations of $SL(2, \mathbb{C})$, which is homeomorphic to the Lorentz group[1].

$$\chi = \begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} \text{ Left handed spinor} \quad (3.5)$$

$$\bar{\psi} = \begin{pmatrix} \bar{\psi}^1 \\ \bar{\psi}^2 \end{pmatrix} \text{ Right handed spinor} \quad (3.6)$$

The left handed spinors transform according to the $(1/2, 0)$ representation and the right handed spinors according to the $(0, 1/2)$ representation. In the formulation of the SM the basic objects have been 4 component Dirac spinors, but one can write the SM Lagrangian also in terms of left and right handed two component spinors. **In supersymmetry one typically works with two component left and right handed spinors rather than with four component Dirac spinors.**

The components of a left handed spinor are always labeled with subscripts and the components of a right handed spinor are always labeled by a superscript. This choice is pure convention. Furthermore in order to label a spinor as right handed we add a bar on the variable which describes a right handed spinor.

Of course one can transform a right handed spinor into a left handed spinor and vice versa (later we will see how).

One can show that the components of the left/right- handed spinors transform under a finite Lorentz transformation as[1]:

$$\begin{aligned} \chi_a &\rightarrow \exp\left(-\frac{i}{2}\omega_{\mu\nu}\sigma^{\mu\nu}\right)_a{}^b \chi_b, \\ \bar{\psi}^{\dot{a}} &\rightarrow \exp\left(-\frac{i}{2}\omega_{\mu\nu}\bar{\sigma}^{\mu\nu}\right)^{\dot{a}}{}_b \bar{\psi}^b. \end{aligned} \quad (3.7)$$

with

$$\begin{aligned} (\sigma^{\mu\nu})_a{}^b &:= \frac{i}{4}(\sigma^\mu\bar{\sigma}^\nu - \sigma^\nu\bar{\sigma}^\mu)_a{}^b, \\ (\bar{\sigma}^{\mu\nu})_{\dot{b}}{}^{\dot{a}} &:= \frac{i}{4}(\bar{\sigma}^\mu\sigma^\nu - \bar{\sigma}^\nu\sigma^\mu)_{\dot{b}}{}^{\dot{a}}, \end{aligned} \quad (3.8)$$

and

$$\sigma^\mu = (1, \sigma^i), \quad \bar{\sigma}^\mu = (1, -\sigma^i). \quad (3.9)$$

In the transformation formulas (3.7) another convention becomes obvious. When we sum over the indices of a right handed spinor we use dotted indices. The introduction of the dotted indices is actually redundant if one marks right handed spinors with a bar, but nevertheless one uses dotted indices for right handed spinors to point out that they belong to another representation. Now we want to form Lorentz invariants out of left and right handed spinors.

The Lorentz invariant spinor product of two L-type spinors (ϕ, χ) is:

$$\phi \cdot \chi = \phi^a \chi_a = \epsilon^{ab} \phi_b \chi_a, \quad (3.10)$$

¹One algebra has one Lie bracket which is in the case of the Poincare algebra just the commutator. But in the case when we extend the algebra by anticommuting generators we have two Lie brackets (commutator and anticommutator) and therefore not a typical algebra anymore. In this case one has a graded Lie algebra [1]

where on the right side everything is expressed with the components of the L-type spinors. Out of two R-type spinors $(\bar{\phi}, \bar{\psi})$ we can form the following Lorentz invariant:

$$\bar{\phi} \cdot \bar{\psi} = \bar{\phi}_{\dot{a}} \bar{\psi}^{\dot{a}} = \epsilon_{\dot{a}\dot{b}} \bar{\phi}^{\dot{b}} \bar{\psi}^{\dot{a}}, \quad (3.11)$$

where again everything on the right side is expressed through components of the right handed spinors. Also we have introduced a 2×2 matrix

$$\epsilon_{12} = -1, \quad \epsilon_{21} = +1, \quad \epsilon_{11} = \epsilon_{22} = 0, \quad (3.12)$$

$$\epsilon^{12} = +1, \quad \epsilon^{21} = -1, \quad \epsilon^{11} = \epsilon^{22} = 0. \quad (3.13)$$

One can show, that

$$i\sigma^2 \chi^* \quad (3.14)$$

transforms like an R-type spinor, if χ transforms like an L-type spinor and

$$i\sigma^2 \psi^* \quad (3.15)$$

like an L-type spinor if ψ transforms like an R-type spinor.

Therefore it is consistent to define:

$$\bar{\chi}_{\dot{a}} = \chi_a^*, \quad (3.16)$$

because

$$i\sigma^2 \chi^* = \begin{pmatrix} \chi_2^* \\ -\chi_1^* \end{pmatrix} \quad (3.17)$$

transforms like R-type if χ transforms like L-type. Then we define a new spinor $\bar{\chi}$ which transforms like an R-type

$$\begin{pmatrix} \chi_2^* \\ -\chi_1^* \end{pmatrix} =: \begin{pmatrix} \bar{\chi}^1 \\ \bar{\chi}^2 \end{pmatrix} = \begin{pmatrix} \bar{\chi}_2 \\ -\bar{\chi}_1 \end{pmatrix}. \quad (3.18)$$

From this we get relation (3.16).

In the following the extended Poincare algebra will be presented for the case of a $\mathcal{N} = 1$ supersymmetry. In this case we have the four symmetry generators :

$$P_\mu, \quad J_{\mu\nu}, \quad Q_a, \quad \bar{Q}^{\dot{a}}, \quad (3.19)$$

where Q_a transforms like a left handed spinor and $\bar{Q}^{\dot{a}}$ like a right handed one. The commutation and anticommutation relations of all generators are now [2]:

$$\begin{aligned} \{Q_a, Q_b\} &= 0 \\ \{Q_a^*, Q_b^*\} &= 0 \Leftrightarrow \{\bar{Q}_{\dot{a}}, \bar{Q}_{\dot{b}}\} = 0 \\ \{Q_a, Q_b^\dagger\} &= 2(\sigma^\mu)_{ab} P_\mu \Leftrightarrow \{Q_a, \bar{Q}_{\dot{b}}\} = 2(\sigma^\mu)_{a\dot{b}} P_\mu \\ [Q_a, J^{\mu\nu}] &= (\sigma^{\mu\nu})_a^b Q_b \\ [Q_a, P^\mu] &= 0 = [\bar{Q}^{\dot{a}}, P^\mu]. \end{aligned} \quad (3.20)$$

One can show that the application of the fermionic generators changes the spin of a one particle state

$$Q |s\rangle = |s \pm \frac{1}{2}\rangle. \quad (3.21)$$

As we will see in section 3.1 we will always pair a SM field with a new introduced *superpartner*. The superpartner for the SM fermions will be a spin zero field. The SM fermion and its superpartner together are called a *chiral supermultiplet*.

But we will also have to pair the spin one SM fields with spin 1/2 superpartners. These two fields together will be called *Vector/ gauge supermultiplets*.

The SM field and its superpartner transform always according to the same representation of $SU(3) \times SU(2) \times U(1)$.

Now that the main ideas of supersymmetry are worked out, we will consider the simplest supersymmetric Lagrangian which contains a complex scalar field ϕ , an L-type spinor χ^2 and an auxiliary field F :

$$\mathcal{L} = \partial_\mu \phi^\dagger \partial^\mu \phi + \chi^\dagger i\bar{\sigma}^\mu \partial_\mu \chi + F^\dagger F. \quad (3.22)$$

The field F was introduced for technical reasons. The transformation rules for ϕ , χ and F are not written out here explicitly, but can be looked up in [2].

The *Wess-Zumino model* generalizes the Lagrangian (3.22) and introduces the most general interaction terms[2]

$$\mathcal{L}_{WZ} = \mathcal{L}_{WZ,free} + \mathcal{L}_{WZ,int}, \quad (3.23)$$

$$\mathcal{L}_{WZ,free} = \partial_\mu \phi_i^\dagger \partial^\mu \phi_i + \chi_i^\dagger i\bar{\sigma}^\mu \partial_\mu \chi_i + F_i^\dagger F_i, \quad (3.24)$$

$$\mathcal{L}_{WZ,int} = W_i F_i - \frac{1}{2} W_{ij} \chi_i \cdot \chi_j + \text{h.c.}, \quad (3.25)$$

with

$$W = \frac{1}{2} M_{ij} \phi_i \phi_j + \frac{1}{6} y_{ijk} \phi_i \phi_j \phi_k, \quad M_{ij} \text{ symmetric in } i, j \text{ and } y_{ijk} \text{ symmetric in } i, j, k \quad (3.26)$$

and

$$W_{ij} = \frac{\partial^2 W}{\partial \phi_i \partial \phi_j} = M_{ij} + y_{ijk} \phi_k, \quad W_i = \frac{\partial W}{\partial \phi_i} = M_{ij} \phi_j + \frac{1}{2} y_{ijk} \phi_j \phi_k. \quad (3.27)$$

In the Wess Zumino Lagrangian we have introduced directly N supermultiplets which are labeled with $i = 1, \dots, N^3$.

If one considers the Euler-Lagrange equations for F_i and F_i^\dagger for the whole Wess Zumino Lagrangian one obtains

$$F_i = -W_i^\dagger, \quad F_i^\dagger = -W_i. \quad (3.28)$$

Therefore one can express the F_i fields in the Wess Zumino Lagrangian through the other fields:

$$\begin{aligned} \mathcal{L}_{WZ} &= \partial_\mu \phi_i^\dagger \partial^\mu \phi_i + \chi_i^\dagger i\bar{\sigma}^\mu \partial_\mu \chi_i + |W_i|^2 + (-|W_i|^2 - \frac{1}{2} W_{ij} \chi_i \cdot \chi_j + \text{h.c.}) \\ &= \partial_\mu \phi_i^\dagger \partial^\mu \phi_i + \chi_i^\dagger i\bar{\sigma}^\mu \partial_\mu \chi_i - |W_i|^2 + (-\frac{1}{2} W_{ij} \chi_i \cdot \chi_j + \text{h.c.}). \end{aligned} \quad (3.29)$$

Next we will introduce the *superfield formalism* which can also be used to get the interaction terms of the Wess Zumino. A left *left chiral superfield* is defined as:

$$\Phi(x, \theta) = \phi(x) + \theta \cdot \chi(x) + \frac{1}{2} \theta \cdot \theta F(x). \quad (3.30)$$

²If χ is the SM field ϕ will be its superpartner

³There is always a summation over i in the WZ Lagrangian.

It combines the complex scalar field ϕ , the left handed spinor χ and the field F .

One can show that if one takes the F-Term⁴ of the *Superpotential of the Wess Zumino model*

$$W_{WZ} = \frac{1}{2} M_{ij} \Phi_i \Phi_j + \frac{1}{6} y_{ijk} \Phi_i \Phi_j \Phi_k \quad (3.31)$$

one obtaines

$$(W_{WZ})|_F = M_{ij} \phi_i F_j - \frac{1}{2} M_{ij} \chi_i \cdot \chi_j + \frac{1}{2} y_{ijk} \phi_i \phi_j F_k - \frac{1}{2} y_{ijk} \chi_i \cdot \chi_j \phi_k. \quad (3.32)$$

The terms in (3.32) are exactly the terms one can find in (3.25)⁵.

$$\mathcal{L}_{WZ,int} = (W_{WZ})|_F + \text{h.c.} \quad (3.33)$$

Taking the F-term of the superpotential is also a more general strategy, because one can show that the F-term always leads to a SUSY invariant action[2].

Furthermore it is not a coincidence that the superpotential W_{WZ} expressed in terms of supermultiplets (3.31) has the same form as the W in (3.26) where all supermultiplets are replaced by the scalar fields of the supermultiplet.

The SUSY and gauge-invariant Lagrangian for the gauge group $SU(N)$ with $N = 1, 2, 3$ is [2]:

$$\mathcal{L}_{gauge}^N = -\frac{1}{4} F_{N,\mu\nu}^a F_N^{a\mu\nu} + i(\lambda_N)^{a\dagger} \bar{\sigma}^\mu (D_\mu \lambda_N)^a + \frac{1}{2} D_N^a D_N^a, \quad (3.34)$$

where $F_{N,\mu\nu}^a, N = 2, 3, a = 1, \dots, N^2 - 1$ ⁶ is the generalized Field strength tensor, whose exact form can be looked up in [2]. $(\lambda^N)^a$ is the L-type superpartner for the gauge field $(A_\mu^N)^a$ and $D_N^a, a = 1, \dots, N^2 - 1$ are real scalar fields which have been introduced in order to balance out the off shell degree of freedom for the gauge fields[2]. The D_μ 's are the covariant derivatives which are acting on the spinor fields. Details are in [2, p. 83].

The two fields A_μ^N, λ_N will then constitute a gauge supermultiplet.

Now that we have constructed Lagrangians for left chiral multiplets and gauge multiplets we want to construct Lagrangians which take into account interactions between one gauge supermultiplet (A_μ, λ) and chiral supermultiplets (ϕ_i, χ_i) . The coupling of the gauge bosons to the matter fields of the chiral supermultiplet will arise through covariant derivatives. The interaction between the D and gaugino fields with elements of the chiral supermultiplet can be determined in [2, p.85-94]. All in all we obtain in the $U(1)$ case:

$$\begin{aligned} \mathcal{L}_{gauge+chiral}^1 &= \mathcal{L}_{WZ}(\text{gauge invariant}, \partial_\mu \rightarrow D_\mu) + \mathcal{L}_{gauge}^1 \\ &\quad \underbrace{-\sqrt{2}g^1[(\phi_i^\dagger \chi_i) \cdot \lambda_1 + \lambda_1 \cdot (\chi_i^\dagger \phi_i) - g^1(\phi_i^\dagger \phi_i) D_1]}_{\text{interaction terms}} \end{aligned} \quad (3.35)$$

and in the case $N = 2, 3$:

$$\begin{aligned} \mathcal{L}_{gauge+chiral}^N &= \mathcal{L}_{WZ}(\text{gauge invariant}, \partial_\mu \rightarrow D_\mu) + \mathcal{L}_{gauge}^N \\ &\quad \underbrace{-\sqrt{2}g^N[(\phi_i^\dagger T_N^a \chi_i) \cdot (\lambda_N)^a + (\lambda_N)^{a\dagger} \cdot (\chi_i^\dagger T_N^a \phi_i)] - g^N(\phi_i^\dagger T_N^a \phi_i) D_N^a}_{\text{interaction terms}} \end{aligned} \quad (3.36)$$

⁴The F-Term is the factor which comes with the $\frac{1}{2}\theta \cdot \theta$ term if one multiplies out the products of all superfields in the superpotential.

⁵Up to the hermitian conjugate which is in (3.25).

⁶In the case $N = 1$ the field strength tensor has only one a component, therefore one does not write an index a in the case $N = 1$

where $g^N, N = 1, 2, 3$ are the three coupling constants. The D fields in the Lagrangians (3.35) and (3.36) can be eliminated with the Euler Lagrange equations:

$$D_N^a = g^N \sum_i \phi_i^\dagger T_N^a \phi_i. \quad (3.37)$$

Now that we have worked out the main ideas of supersymmetry we will explore a straight forward supersymmetric extension of the SM Lagrangian (The minimal supersymmetric extension of the Standard Model (MSSM))

3.1 The MSSM

The MSSM is the minimal extension of the SM, because it introduces the minimal amount of extra particles which are needed to make the theory supersymmetric. For each SM particle a superpartner will be introduced⁷. The SM field and its superpartner will constitute a supermultiplet and have the same internal quantum numbers. A list of the chiral supermultiplets [2, p. 95] can be seen in table 3.1. The gauge supermultiplets are in table 3.2.

Names	spin 0	spin 1/2	$SU(3)_c$	$SU(2)_L$	$U(1)_Y$
Q	$\begin{pmatrix} \tilde{u}_L \\ \tilde{d}_L \end{pmatrix}$	$\begin{pmatrix} u_L \\ d_L \end{pmatrix}$	3	2	1/3
\bar{u}	$\tilde{\bar{u}}_L (\tilde{u}_R)$	$\bar{u}_L \sim (u_R)^c$	$\bar{3}$	1	-4/3
\bar{d}	$\tilde{\bar{d}}_L (\tilde{d}_R)$	$\bar{d}_L \sim (d_R)^c$	$\bar{3}$	1	2/3
L	$\begin{pmatrix} \tilde{\nu}_{l,L} \\ \tilde{l}_L \end{pmatrix}$	$\begin{pmatrix} \nu_{l,L} \\ l_L \end{pmatrix}$	1	2	-1
\bar{l}	$\tilde{\bar{l}}_L (\tilde{l}_R)$	$\bar{l}_L \sim (l_R)^c$	1	1	2
H_u	$\begin{pmatrix} H_u^+ \\ H_u^0 \end{pmatrix}$	$\begin{pmatrix} \tilde{H}_u^+ \\ \tilde{H}_u^0 \end{pmatrix}$	1	2	1
H_d	$\begin{pmatrix} H_d^0 \\ H_d^- \end{pmatrix}$	$\begin{pmatrix} \tilde{H}_d^0 \\ \tilde{H}_d^- \end{pmatrix}$	1	2	-1

Table 3.1: Chiral supermultiplets in the MSSM. All objects listed in the spin 1/2 column are 2 component objects and should not be mixed up with the 4 component objects in table 2.2. Furthermore the bar here means not Dirac conjugation, but antiparticle. There are always 3 supermultiplets for the Quarks and Leptons respectively which will be labeled with $Q_i, \bar{u}_i, \bar{d}_i, L_i, \bar{l}_i, i = 1, 2, 3$. We just have dropped the index i in the table. The superpartners of the ordinary SM fields are labeled with a tilde. The superpartner for the SM fermions are named with a 's' plus their SM name. For example the superpartner for the top quark is called stop. The names of the superpartners of the Higgs bosons are called Higgsinos. The $U(1)_y$ quantum number is determined with the relation $Y = 2(Q - I_3)$.

For fermions the basic objects in the MSSM are L-type spinors⁸. L-type spinors will appear

⁷Except for the Higgs. We will have two Higgs doublets in the SM each with a superpartner.

⁸This is different from the SM. In the SM we have Dirac spinors which are made out of a left and right handed spinor. One can of course express the SM Lagrangian completely out of L-type spinors.

in $SU(2)$ doublets. The R-type spinors will be expressed in the MSSM through the L-type spinor of the antiparticle. For example for the electron we have $e_R^- = (e_L^+)^c$.

The supermultiplet Q is an $SU(2)$ doublet containing the L-type quark fields and its superpartners. The corresponding superfield will also be labeled with Q ⁹:

$$Q = \begin{pmatrix} \tilde{u}_L \\ \tilde{d}_L \end{pmatrix} + \begin{pmatrix} u_L \cdot \theta \\ d_L \cdot \theta \end{pmatrix} + \frac{1}{2} \theta \cdot \theta \begin{pmatrix} F_u \\ F_d \end{pmatrix}. \quad (3.38)$$

Similar statements are also valid for the chiral supermultiplet L .

The left handed supermultiplet \bar{u} consists of the left handed spinor \bar{u}_L (which is defined to be the charge conjugate of the right handed up quark spinor) and its superpartner.

In table 3.1 the left handed field \bar{u}_L and its superpartner are $SU(2)$ singlets which is curious, because left handed fields should be not $SU(2)$ singlets. But this fact can be justified, because one has to see the left handed field as a charge conjugation of a right handed field which is a $SU(2)$ singlet. In table 2.2 the representation for right handed up type quarks was $(3, 1, 4/3)$ and because in table 3.1 we look at the charge conjugated fields we have for the up type quark supermultiplet the representation $(\bar{3}, 1, -4/3)$. In supersymmetry one expresses all R-type spinors with L-type spinors. Nevertheless in the end one has to construct Lagrangians which respect all symmetries and the objects in table 3.1 have to be arranged such that this is the case.

The superfield for \bar{u} which is also labeled as \bar{u} is

$$\bar{u} = \tilde{\bar{u}}_L + \theta \cdot \bar{u}_L + \frac{1}{2} \theta \cdot \theta. \quad (3.39)$$

Similar statements are also valid for the chiral supermultiplet \bar{d} and \bar{l} .

In table 3.1 we find two Higgs doublets with spin zero fields, where in the SM there is only one of these doublets. For both Higgs doublets we introduce superpartners. The second introduced Higgs doublet which is not present in the SM is the only exception where the SM particle content is extended. This extension is due to the fact that in the SM we need the charge conjugation of the SM Higgs doublet to form all Yukawa terms. The charge conjugation of the SM higgs doublet is not possible in the MSSM, because this would involve hermitian conjugates in the superpotential and they would violate supersymmetry [2, p. 58]. Therefore we have to introduce a second SM like Higgs doublet with opposite hypercharge to form all possible Yukawa terms like in the SM.

Names	spin 1/2	spin 1	$SU(3)_c$	$SU(2)_L$	$U(1)_y$
gluinos, gluons	\tilde{g}^a	g^a	8	1	0
winos, W-bosons	\tilde{W}^a	W^a	1	3	0
bino, B boson	\tilde{B}	B	1	1	0

Table 3.2: Gauge supermultiplets in the MSSM. The gauge bosons and its superpartners transform according to the adjoint representation. For the adjoint representation of $SU(3)$ for example we write an 8, because there are $3^2 - 1 = 8$ gauge bosons. The superpartners of the gauge bosons are always called with an 'ino' at the end. For example the superpartners for the gluons are called gluinos.

The introduced chiral and gauge supermultiplets here define the particle content of the minimal supersymmetric extension of the SM (MSSM).

⁹Of course we always drop the family index i of Q

The superpotential of the MSSM is [2]

$$W_{MSSM} = \sum_{i,j} y_u^{ij} \bar{u}_i Q_j \cdot H_u - y_d^{ij} \bar{d}_i Q_j \cdot H_d - y_l^{ij} \bar{l}_i L_j \cdot H_d + \mu H_u \cdot H_d. \quad (3.40)$$

where the \cdot in this case is the SU(2) invariant dot product. The same symbol was also used before for a spinor product, but it should be clear from the context what is meant.

The MSSM superpotential is the simplest Wess Zumino Superpotential which respects all gauge symmetries and takes into account all introduced chiral supermultiplets. If we choose the superfield formalism we have to put all superfields in (3.40). One obtains then the interaction terms by taking the F-Term of the superpotential.

The y^{ij} 's are exactly the same Yukawa couplings which also occur in the SM¹⁰.

We have now constructed a supersymmetric Lagrangian, which respects all gauge symmetries and which does not include non renormalizable terms. In summary the whole Lagrangian is:

$$\mathcal{L}_{MSSM} = \mathcal{L}_{WZ,MSSM} + \sum_{i=1}^3 \mathcal{L}_{\text{gauge,MSSM}}^i + \mathcal{L}_{\text{gauge-chiral-interaction}}^i \quad (3.41)$$

with

$$\mathcal{L}_{WZ,MSSM} = \mathcal{L}_{WZ,\text{free,MSSM}} + \mathcal{L}_{WZ,\text{int,MSSM}}. \quad (3.42)$$

$\mathcal{L}_{WZ,\text{free,MSSM}}$ includes all chiral supermultiplets arranged such that all symmetries are respected. And $\mathcal{L}_{WZ,\text{int,MSSM}}$ is constructed from the given superpotential W_{MSSM} .

$$\mathcal{L}_{\text{gauge,MSSM}}^1 = -\frac{1}{4} F_{1,\mu\nu} F_1^{\mu\nu} + i(\tilde{B})^\dagger \bar{\sigma}^\mu (D_\mu \tilde{B}) + \frac{1}{2} D_1 D_1, \quad (3.43)$$

$$\mathcal{L}_{\text{gauge,MSSM}}^2 = -\frac{1}{4} F_{2,\mu\nu}^a F_2^{a\mu\nu} + i(\tilde{W})^{a\dagger} \bar{\sigma}^\mu (D_\mu \tilde{W})^a + \frac{1}{2} D_2^a D_2^a, \quad (3.44)$$

$$\mathcal{L}_{\text{gauge,MSSM}}^3 = -\frac{1}{4} F_{3,\mu\nu}^a F_3^{a\mu\nu} + i(\tilde{g})^{a\dagger} \bar{\sigma}^\mu (D_\mu \tilde{g})^a + \frac{1}{2} D_3^a D_3^a, \quad (3.45)$$

$$\mathcal{L}_{\text{gauge-chiral-interaction}}^1 = -\sqrt{2} g^1 [(\phi_i^\dagger \chi_i) \cdot \tilde{B} + \tilde{B} \cdot (\chi_i^\dagger \phi_i) - g^1 (\phi_i^\dagger \phi_i) D_1], \quad (3.46)$$

$$\mathcal{L}_{\text{gauge-chiral-interaction}}^2 = -\sqrt{2} g^2 [(\phi_i^\dagger T_2^a \chi_i) \cdot (\tilde{W})^a + (\tilde{W})^{a\dagger} \cdot (\chi_i^\dagger T_2^a \phi_i)] - g^2 (\phi_i^\dagger T_2^a \phi_i) D_2^a, \quad (3.47)$$

$$\mathcal{L}_{\text{gauge-chiral-interaction}}^3 = -\sqrt{2} g^3 [(\phi_i^\dagger T_3^a \chi_i) \cdot (\tilde{g})^a + (\tilde{g})^{a\dagger} \cdot (\chi_i^\dagger T_3^a \phi_i)] - g^3 (\phi_i^\dagger T_3^a \phi_i) D_3^a, \quad (3.48)$$

where the summation over i runs over all chiral supermultiplets.

The μ parameter in the superpotential is the only new parameter which has been introduced with respect to the SM. But the SUSY invariant Lagrangian we have constructed is not a realistic one, because one can show that we cannot accommodate electroweak symmetry breaking[2, p. 98], which is necessary to give the ordinary SM particles a mass. Therefore we have to introduce SUSY breaking (EWSB) terms to allow electroweak symmetry breaking.

Now lets come to the SUSY breaking terms in the MSSM Lagrangian. We expect that supersymmetry is an exact symmetry and that it will be spontaneously broken [3]. There are a lot of models which introduce spontaneously supersymmetry breaking, but because there is still no consensus about the correct way of supersymmetry breaking we will simply introduce terms

¹⁰One can see this if one writes out the Yukawa terms of the SM with two component spinors[2, p. 96]. Here we use a more suitable naming as in (2.10) and furthermore another sign convention.

which break supersymmetry explicitly. The supersymmetry breaking terms are called soft, because the couplings of these terms have positive mass dimension [3, p. 48].

The most general soft SUSY breaking terms which respect the gauge symmetries and do not spoil renormalizability are[2, p. 115-116].

- Mass terms for the Gauginos

$$-\frac{1}{2}(M_3\tilde{g}^a \cdot \tilde{g}^a + M_2\tilde{W}^a \cdot \tilde{W}^a + M_1\tilde{B}^a \cdot \tilde{B}^a + \text{h.c.}) \quad (3.49)$$

- Masses for the squarks and sleptons

$$-m_{\tilde{Q}_{ij}}^2\tilde{Q}_i^\dagger \cdot \tilde{Q}_j - m_{\tilde{u},ij}^2\tilde{\bar{u}}_{L,i}^\dagger \tilde{u}_{L,i} - m_{\tilde{d},ij}^2\tilde{\bar{d}}_{L,i}^\dagger \tilde{d}_{L,i} \quad (3.50)$$

$$-m_{\tilde{L}_{ij}}^2\tilde{L}_i^\dagger \cdot \tilde{L}_j - m_{\tilde{l},ij}^2\tilde{\bar{l}}_{L,i}^\dagger \tilde{l}_{L,i} \quad (3.51)$$

where \tilde{Q}_i , \tilde{L}_i are no superfield, but the 'scalar component' of the chiral superfield

$$\tilde{Q}_i = \begin{pmatrix} \tilde{u}_L \\ \tilde{d}_L \end{pmatrix}_i, \quad \tilde{L}_i = \begin{pmatrix} \tilde{\nu}_{l,L} \\ \tilde{l}_L \end{pmatrix}_i \quad (3.52)$$

- Higgs mass terms:

$$-m_{H_u}^2H_u^\dagger \cdot H_u - m_{H_d}^2H_d^\dagger \cdot H_d - (bH_u \cdot H_d + \text{h.c.}) \quad (3.53)$$

The Higgs doublets in this case are also no chiral superfields, although the notation is the same for a chiral superfield (see table 3.1), but here we mean:

$$H_u = \begin{pmatrix} H_u^+ \\ H_u^0 \end{pmatrix}, \quad H_d = \begin{pmatrix} H_d^0 \\ H_d^- \end{pmatrix} \quad (3.54)$$

- triple scalar coupling

$$-a_u^{ij}\tilde{\bar{u}}\tilde{Q}_j \cdot H_u + a_d^{ij}\tilde{\bar{d}}_i\tilde{W}_j \cdot H_d + a_l^{ij}\tilde{\bar{l}}_i\tilde{L}_j \cdot H_d + \text{h.c.} \quad (3.55)$$

In this case the fields are also no chiral superfields.

In the following there will be a few comments on the soft SUSY breaking terms.

- The mass matrices which appear in the soft SUSY breaking terms must be hermitian.
- All the newly introduced terms break supersymmetry explicitly, but still respect all gauge symmetries.
- All introduced soft SUSY breaking parameters are scale depended. One has to define them at a specific scale and use renormalization group equations (RGE) to determine them at another scale. More about RGE can be found in [2].
- The introduction of the soft SUSY breaking parameters implicates 105 new parameters [2, p. 117], which can lead to large flavor changing neutral currents (FCNC) and new sources of CP-violation. Now that we do not observe large FCNC's and CP-violations in experiments, a lot of the newly introduced parameters will be close to zero. In sections 3.2 and 3.3 we will discuss models which can be seen as a subset of the whole MSSM model and do not have these FCNCs and CP violation terms.

- In the construction of the MSSM superpotential Lagrangian there would have been more possible terms which have not been written out, because they do not respect the R-parity symmetry [11].

$$(-1)^R = (-1)^{3B+2S-L} \quad (3.56)$$

where B is the baryon number, L the lepton number and S the spin. For the superparticles¹¹ we always have -1 and for the ordinary SM particles we have $+1$. This means that a SM particle can only decay to two superparticles and not into a SM particle and one superparticle. A superparticle can only decay into one SM particle and one superparticle, but not into two superparticles. One important consequence of R-parity conservation is that the lightest superparticle will be totally stable. If this stable particle is electrically charged this would be excluded cosmologically [11], but if the particle is electrically neutral it can constitute a dark matter candidate.

Electroweak symmetry breaking happens if we give H_u^0 and H_d^0 a nonvanishing vacuum expectation value called v_u and v_d respectively. One can then derive two equations which have to be satisfied such that electroweak symmetry breaking can happen[2, p. 123 ff.]:

$$(|\mu|^2 + m_{H_u}^2) = b \cot \beta + m_Z^2 / 2 \cos 2\beta, \quad (3.57)$$

$$(|\mu|^2 + m_{H_d}^2) = b \tan \beta + m_Z^2 / 2 \cos 2\beta, \quad (3.58)$$

with

$$\tan \beta = \frac{v_u}{v_d}, \quad (3.59)$$

$$(v_u^2 + v_d^2)^{1/2} = \left(\frac{2m_W^2}{g^2} \right). \quad (3.60)$$

One now uses equations (3.57) and (3.58) to eliminate the parameters $m_{H_u}^2$ and $m_{H_d}^2$ in favor of the parameters $\tan \beta$, b , μ . The Z Boson mass is an experimental input.

$$m_{H_u}^2(b, \tan \beta, \mu), \quad m_{H_d}^2(b, \tan \beta, \mu) \quad (3.61)$$

The ratio of the two vacuum expectation values is now taken as a parameter of the Lagrangian. The respective values of v_u and v_d can be calculated with (3.60). One can show that v_u and v_d can be chosen real and positive.

Before we go on it should be mentioned that μ is often choosen to be real in order to avoid large CP-violation effects[2, p. 140].

Next we will consider the tree level masses of the scalar Higgses in the MSSM[2, p.128 ff.]. If we consider the fields $(\text{Im}(H_u^0), \text{Im}(H_d^0))$ in the MSSM Lagrangian there will be quadratic mixing¹² and therefore we have to diagonalize these mixing terms in order to have mass eigenstates in the Lagrangian which are a superposition of $(\text{Im}(H_u^0), \text{Im}(H_d^0))$. The corresponding mass eigenstates have tree level masses zero¹³ and

$$m_A = \sqrt{\frac{2b}{\sin \beta}}. \quad (3.62)$$

¹¹With superparticles all particles of the MSSM which are not SM particles are meant.

¹²When we talk about mixing in this context we mean that there are not only 'pure' quadratic terms $\text{Im}(H_u^0)^2$ and $\text{Im}(H_d^0)^2$, but also terms which are mixed $\text{Im}(H_u^0)\text{Im}(H_d^0)$

¹³Goldstone boson

m_A is the Tree level pseudoscalar Higgs mass parameter. The relation can be used to replace the parameters $(b, \tan \beta)$ by $(m_A, \tan \beta)$ in the whole MSSM Lagrangian.

Therefore the MSSM parameters

$$m_{H_u}^2, \quad m_{H_d}^2, \quad b, \quad (3.63)$$

can be replaced by

$$\mu, \quad m_A, \quad \tan \beta. \quad (3.64)$$

In [12] one can find that the tree level mass m_A can as a parameter also be replaced by the pseudoscalar Higgs pole mass m_{A^0} .

Next we consider the fields (H_u^+, H_d^-) which also have quadratic mixing. After diagonalization we end up with the massless field G^+ , which is the longitudinal mode of the W^+ boson and the massive field H^+ , $m_{H^+} = \sqrt{m_W^2 + m_{A^0}^2}$.

The diagonalization of the quadratic mixing for the fields $(\text{Re}(H_u^0 - v_u), \text{Re}(H_d^0 - v_d))$ gives the two mass eigenstates H^0 and h^0 .

If we write out the terms in the MSSM Lagrangian for the Bino(\tilde{B}), Wino(\tilde{W}^0) and the neutral Higgsions($\tilde{H}_u^0, \tilde{H}_d^0$) we see that there is also quadratic mixing between these four states. The diagonalization of the mixing matrixes will lead to new mass eigenstates called neutralinos $\tilde{\chi}_i^0$, $i = 1, 2, 3, 4$. They are defined in order that their masses satisfy the relations $m_{\tilde{\chi}_1^0} < m_{\tilde{\chi}_2^0} < m_{\tilde{\chi}_3^0} < m_{\tilde{\chi}_4^0}$. $\tilde{\chi}_1^0$ provides a non-baryonic dark matter candidate[2, p. 140] if it is the lightest superparticle.

In analogy to the neutralinos there is also quadratic mixing between \tilde{W}^+ and \tilde{H}_u^+ . The diagonalization of these terms leads to mass eigenstates which are called positive charginos $\tilde{\chi}_i^+$, $i = 1, 2$. The quadratic mixing between \tilde{W}^- and \tilde{H}_d^- leads to mass eigenstates called negative charginos $\tilde{\chi}_i^+$, $i = 1, 2$.

The squark mass terms also contain quadratic mixing and therefore have to be diagonalized as well. It turns out that the off diagonal components are small except for the stop and sbottom quarks. The diagonalization for the sbottom and stop mass matrixes yields mass eigenstates called $\tilde{b}_{1/2}$ and $\tilde{t}_{1/2}$ which are linear combinations of $\tilde{b}_{L/R}$ and $\tilde{t}_{L/R}$. The mass matrices for the sup, sdown, scharm, sstrange are not diagonalized because in a good approximation they are already mass eigenstates.

3.2 The constrained MSSM

The constrained MSSM (cMSSM)[13] assumes that at a very high scale¹⁴ the parameters of the MSSM are unified

$$M_1 = M_2 = M_3 = M_{1/2}, \quad (3.65)$$

$$m_{\tilde{Q},ij}^2 = m_{\tilde{L},ij}^2 = m_{\tilde{u},ij}^2 = m_{\tilde{d},ij}^2 = m_{\tilde{l},ij}^2 = \delta_{ij} M_0^2, \quad (3.66)$$

$$a_u^{ij} = A_0 y_u^{ij}, \quad a_d^{ij} = A_0 y_d^{ij}, \quad a_l^{ij} = A_0 y_l^{ij}, \quad (3.67)$$

$$m_{H_u}^2 = m_{H_d}^2 = M_0^2. \quad (3.68)$$

To complete the parameters of the Higgs sector we have to specify the parameters $\tan \beta$ and $\text{sign}(\mu)$ ¹⁵. If one wants the MSSM parameters at a lower scale one has to apply RGE. After

¹⁴Typically the plank or grand unified theory GUT scale.

¹⁵We only have to specify the sign of μ because with (3.57) and (3.58) the absolute value of μ is already determined.

Furthermore also the b parameter will be determined with (3.57) and (3.58) if $\tan \beta$, m_{H_u} and m_{H_d} are given.

bringing the MSSM parameters down to a lower scale by using the RGE, the relations (3.65)-(3.68) will not hold anymore. An exception are the parameters $\tan\beta$ and $\text{sign}(\mu)$ which have to be defined at the ew scale[13]. Typically one chooses the Z mass as the ew scale. In summary one has to specify 4 parameters to get a valid cMSSM model

$$M_0, \quad M_{1/2}, \quad A_0, \quad \tan\beta, \quad \text{sign}(\mu) \quad (3.69)$$

3.3 The phenomenological MSSM

In contrast to the cMSSM the phenomenological MSSM (pMSSM) does not need RGE because one specifies the soft SUSY breaking parameters directly at the electroweak scale. In section 3.3.1 the electroweak scale will be defined. In order to avoid FCNC and new sources of CP-violations one often sets in pMSSM models the tree level mass matrices and trilinear coupling matrices of the soft SUSY breaking terms diagonal.

Furthermore constraints between the remaining nonzero parameters are imposed. Depending on the relations between the soft SUSY parameters we can get different pMSSM models which are envisaged in the following.

All parameters for the pMSSM are defined at the ew scale which is in this thesis fixed to 1 TeV. RGE are only needed for the gauge and Yukawa couplings which are defined at m_Z .

3.3.1 pMSSM-19

The 19 dimensional phenomenological MSSM (pMSSM-19) defines the MSSM parameters [13] as follows:

$$M_1, M_2, M_3, \quad (3.70)$$

$$\mathbf{m}_{\tilde{Q}}^2 = \begin{pmatrix} m_{\tilde{Q}_{L,12}}^2 & 0 & 0 \\ 0 & m_{\tilde{Q}_{L,12}}^2 & 0 \\ 0 & 0 & m_{\tilde{Q}_{L,3}}^2 \end{pmatrix}, \quad \mathbf{m}_{\tilde{L}}^2 = \begin{pmatrix} m_{\tilde{L}_{L,12}}^2 & 0 & 0 \\ 0 & m_{\tilde{L}_{L,12}}^2 & 0 \\ 0 & 0 & m_{\tilde{L}_{L,3}}^2 \end{pmatrix}, \quad (3.71)$$

$$\mathbf{m}_{\tilde{d}}^2 = \begin{pmatrix} m_{\tilde{d}_{R,12}}^2 & 0 & 0 \\ 0 & m_{\tilde{d}_{R,12}}^2 & 0 \\ 0 & 0 & m_{\tilde{d}_{R,3}}^2 \end{pmatrix}, \quad \mathbf{m}_{\tilde{u}}^2 = \begin{pmatrix} m_{\tilde{u}_{R,12}}^2 & 0 & 0 \\ 0 & m_{\tilde{u}_{R,12}}^2 & 0 \\ 0 & 0 & m_{\tilde{u}_{R,3}}^2 \end{pmatrix}, \quad \mathbf{m}_{\tilde{l}}^2 = \begin{pmatrix} m_{\tilde{l}_{R,12}}^2 & 0 & 0 \\ 0 & m_{\tilde{l}_{R,12}}^2 & 0 \\ 0 & 0 & m_{\tilde{l}_{R,3}}^2 \end{pmatrix}, \quad (3.72)$$

$$\mathbf{a}^u = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_t \end{pmatrix}, \quad \mathbf{a}^d = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_b \end{pmatrix}, \quad \mathbf{a}^l = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_\tau \end{pmatrix} \quad (3.73)$$

The Higgs sector will be described with the following parameter choices in this thesis:

$$\mu, \quad m_{A^0}, \quad \tan\beta. \quad (3.74)$$

3.3.2 pMSSM-11

The main difference in the pMSSM-11 compared to the pMSSM-19 is that now the masses of the left and right sfermions are equal and one has only one trilinear coupling parameter. Therefore the parameters for the 11 parameter phenomenological MSSM (pMSSM-11) are:

$$M_1, M_2, M_3, \quad (3.75)$$

$$\mathbf{m}_{\tilde{Q}}^2 = \begin{pmatrix} m_{\tilde{Q}_{12}}^2 & 0 & 0 \\ 0 & m_{\tilde{Q}_{12}}^2 & 0 \\ 0 & 0 & m_{\tilde{Q}_3}^2 \end{pmatrix}, \quad \mathbf{m}_{\tilde{L}}^2 = \begin{pmatrix} m_{\tilde{L}_{12}}^2 & 0 & 0 \\ 0 & m_{\tilde{L}_{12}}^2 & 0 \\ 0 & 0 & m_{\tilde{L}_3}^2 \end{pmatrix}, \quad (3.76)$$

$$\mathbf{m}_{\tilde{d}}^2 = \mathbf{m}_{\tilde{u}}^2 = \mathbf{m}_{\tilde{Q}}^2, \quad \mathbf{m}_{\tilde{l}}^2 = \mathbf{m}_{\tilde{L}}^2, \quad (3.77)$$

$$\mathbf{a}^u = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_0 \end{pmatrix} = \mathbf{a}^d = \mathbf{a}^l, \quad (3.78)$$

$$\mu, \ m_{A^0}, \ \tan \beta. \quad (3.79)$$

This thesis will focus mostly on the investigation of the pMSSM-11 model.

4 Investigating the pMSSM-11

In section 3.3.2 the parameters of the pMSSM-11 model have been specified. Each 11 dimensional parameter point $\mathbf{p} = (M_1, M_2, \dots, \tan(\beta))$ will lead to a different pMSSM-11 model. There will be two investigation steps for the pMSSM-11:

1. we want to find the optimal 11-dimensional parameter point (which defines the optimal pMSSM-11 model). By definition the optimal pMSSM-11 model gives predictions which coincide best with the experimental observations. We therefore want to include as many as possible experimental observations in the comparison with the theoretical predictions. A variable which specifies the agreement between theoretical predictions and the experimental measurement is the χ^2 . If one compares two parameter points the experimental data is more likely if the one with the lower χ^2 is true. The Fittino framework combines the χ^2 's from all observables to a global χ^2 and searches for the χ^2 minimum in the 11 dimensional parameter space (more details are in 4.1).
2. when we found the 'optimal' pMSSM-11 model we want to calculate a p-value for the pMSSM-11, which gives information if the model is excluded or allowed by the experimental measurements.

In this thesis we will only work on the first investigation step (finding the parameter point which gives a χ^2 minimum) and therefore the p-value determination is not described in detail.

The following section will examine the Fittino framework which is used to achieve the two described investigation steps.

4.1 Fittino

Fittino [14], [15] is a general software framework which can be used to investigate supersymmetric models. Previously there have been for example studies for the cMSSM [16] with Fittino. In this thesis the supersymmetric model is the pMSSM-11.

For each parameter point Fittino uses different theory tools to calculate for chosen observables¹ the theoretical predictions and compares them all to the values which have been measured in experiment. All measurements and theory predictions are then combined to an overall χ^2 :

$$\chi^2 = (\mathbf{O}_{\text{meas}} - \mathbf{O}_{\text{pred}})^T \text{cov}^{-1} (\mathbf{O}_{\text{meas}} - \mathbf{O}_{\text{pred}}) + \chi^2_{\text{limits}}, \quad (4.1)$$

where \mathbf{O}_{meas} is the vector of measurements, \mathbf{O}_{pred} the vector of predictions, cov the covariance matrix including the theoretical uncertainties and χ^2_{limits} is the sum of the χ^2 for all observables where only bounds but no measurements are applied[16].

A higher order Monte Carlo Markov Cain (HOMCMC) then helps to find the global χ^2 minimum in the 11 dimensional parameter space.

¹A list with the chosen observables as well as the corresponding programs which will be used to calculate the theory predictions will follow.

Fittino will be needed in both of the two investigation steps, because in both cases we are searching for a χ^2 minimum².

There are basically four groups of observables which are included in the comparison of theory and experiment (see figure 4.1). The following list will give an overview which observable belongs

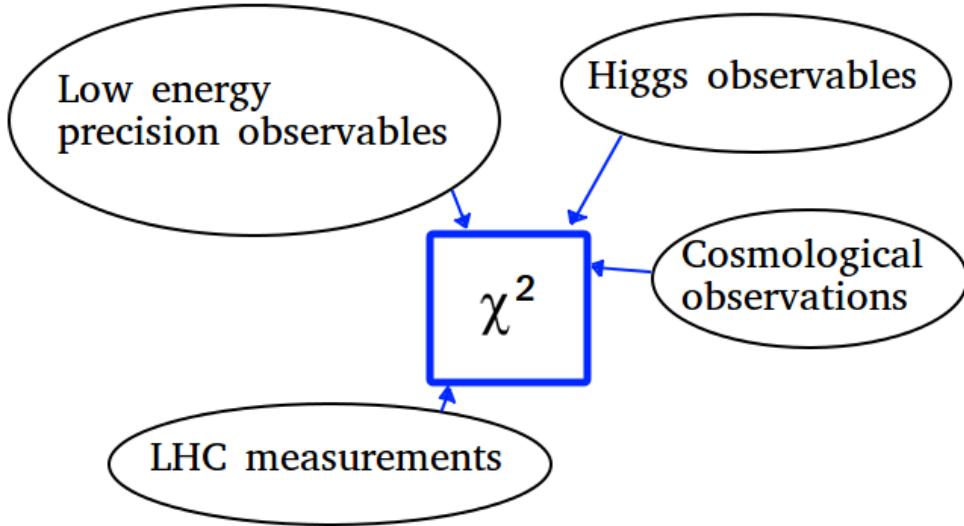


Figure 4.1: Overview of the four groups of observables which are included into the comparison between theory predictions and experimental measurements. All theory and experiment comparisons are combined to an overall χ^2 .

to which group.

1. Low energy precision observables³:

In table 4.1 we give all low energy precision observables, their corresponding experimental measured values, the assumed theory values and the tools which are used to calculate the theory predictions.

We use the anomalous magnetic moment of the muon $(g - 2)_\mu =: a_\mu$, the effective weak mixing angle $\sin^2 \Theta_{\text{eff}}$, the mass of the top and W boson, the B_s oscillation frequency and

²Actually for the first investigation step this is obvious with the previous description of the Fittino framework. But Fittino will also be used for the second investigation step, because there one searches several times for a χ^2 minimum. Nevertheless we will not go into detail here, because in this thesis we will only look at the first investigation step.

³Sometimes we call this class low energy electroweak precision observables

three branching ratios of rare decays: $\mathcal{B}(B_s \rightarrow \mu\mu)$, $\mathcal{B}(b \rightarrow s\gamma)$, $\mathcal{B}(B_u \rightarrow \tau\nu)$. The top pole mass parameter will be varied during the scan, because it can have effects on the other observables. Therefore we scan 12 parameters, the 11 pMSSM parameters and the top mass parameter⁴.

observable	experimental measured \pm error	theoretical error	tool
$a_\mu - a_\mu^{SM}$	$(28.7 \pm 8.0) \times 10^{-10}$ [17] [18]	7%	FeynHiggs [19–24]
$\sin^2 \Theta_{\text{eff}}$	0.23113 ± 0.00021 [25]	0.05 %	FeynHiggs [19–24]
m_t	173.34 ± 0.76 GeV [26]	1GeV	SPheno [27, 28]
m_W	80.385 ± 0.015 GeV [29]	0.01 %	FeynHiggs [19–24]
Δm_s	17.719 ± 0.043 ps $^{-1}$ [30]	24 %	FeynHiggs [19–24]
$\mathcal{B}(B_s \rightarrow \mu\mu)$	$(2.90 \pm 0.70) \times 10^{-9}$ [31]	26 %	SuperIso [32]
$\mathcal{B}(b \rightarrow s\gamma)$	$(3.43 \pm 0.22) \times 10^{-4}$ [33]	14%	SuperIso [32]
$\mathcal{B}(B_u \rightarrow \tau\nu)$	$(1.05 \pm 0.25) \times 10^{-4}$ [30]	20 %	SuperIso [32]

Table 4.1: Electroweak precision observables which are used in the global fit. The values in this table are overtaken from the previous Fittino cMSSM study [16].

2. Higgs Observables:

- The program HiggsBounds [34–38] applies constraints which come from Higgs searches at LEP, Tevatron and the LHC. Furthermore Higgsbounds also gives limits on heavier Higgs bosons. The χ^2 from Higgsbounds is one part of the χ^2_{limits} in (4.1).
- The program HiggsSignals [39] is used to incorporate signal rate measurements and the mass measurement of the 125 GeV Higgs boson. We chose the medium observable set, which includes 10 observables that are listed in [16].

3. Cosmological observables:

- The dark matter relic density has the experimental measured value $\Omega h^2 = 0.1187 \pm 0.0017$ [40]. The theoretical predictions are calculated with MicrOMEGAs [41, 42] and we require as the dark matter candidate the lightest neutralino. parameter points where the lightest neutralino is not the LSP are excluded. The error on the theoretical predictions is to be assumed a 10 % relative error.
- The bounds on the spin-independent WIMP-nucleon scattering cross section from the LUX experiment [43] are combined with the theoretical predictions from DarkSUSY [44] and AtroFit [45] to another χ^2 which also enters χ^2_{limits} .

4. LHC observables:

The LHC observables are the number of events in the signal regions (SR) of LHC analyses. For the comparison of theoretical values and experimental measurements one has to simulate events on the basis of a pMSSM-11 model and run several LHC (ATLAS or CMS) analysis⁵ over the simulated events. As an output one will obtain for each analysis the number of theoretical predicted events in the signal regions. One can compare these numbers to the numbers which have been obtained in the experiment. In chapter ... we describe how an overall LHC χ^2 can be obtained from all numbers. The overall LHC χ^2

⁴Of course the scan tries to find an optimal point where the top mass is close to the measured value.

⁵In section 5 we will give a list of all selected analyses.

is also included in the χ^2_{limits} in (4.1).

On top of all observables the restriction $m_{\chi_1^\pm} > 103.5 \text{ GeV}$ from LEP [46, p. 1582-1583] will be used in the future, but for the studies in this thesis it has not been considered yet.

4.2 The strategy to make the LHC predictions faster

Getting the theoretical predictions for the LHC observables is the most time consuming task. It takes $O(\text{hours})$ to simulate events and run several LHC analyses over them to obtain the number of events in the SRs⁶.

The comparison of theoretical and experimental values for all other observables takes $O(\text{seconds})$. If one now wants to scan large areas of the pMSSM-11 parameter space the LHC observables will be problematic, because the scan will run $O(\text{hours})$ to scan one pMSSM-11 parameter point. Therefore scanning large areas in the pMSSM-11 parameter space becomes technically impossible.

If one wants for example to scan 10 points in each parameter dimension this will take approximately 10^{11} cpu hours. Even on the largest cluster this seems not doable in a reasonable time frame.

One could come up with the idea that we just leave out the direct LHC comparison and only compare theoretical and experimental values for all other observables. But it will turn out that we need both, because the χ^2 coming from the LHC observables⁷ will give significant χ^2 contributions and is therefore necessary to find the global χ^2 minimum.

Therefore the challenging task is the creation of a tool which makes the LHC observables comparison much quicker ($O(\text{seconds})$).

In this thesis two different tools will be developed which achieve this goal:

- The first tool will take as an input the 11 parameters of the pMSSM. The output of this tool is the LHC χ^2 from all considered LHC observables/SRs.
- The second tool also takes 11 inputs but outputs the number of events in one SR.⁸

The development of these two tools is the main task of this master thesis. The tools will then be integrated into Fittino to solve the first investigation step in a reasonable time frame. Basically there are two different approaches in developing the two before mentioned tools:

1. The tool tries to tune the event generation processes in order to speed them up. If we could do the event generation in $O(\text{seconds})$ we could also generate the LHC χ^2 and the number of events in all SRs in $O(\text{seconds})$.
2. We generate events only for a few 11-dimensional parameter points ($0.5 - 1.0 \times 10^6$ is a doable number on a cluster). For these points we calculate the LHC χ^2 and number of events (# events) in the LHC SRs. Then the tool should **interpolate** from these known

⁶Depending how much effort one puts in the event generation this can be pushed to $O(\text{ minutes})$. Details about the event generation will be given later in chapter 5

⁷A detailed description of how this is calculated can be found in section 5.3

⁸Actually one needs a tool which outputs the number of events in all SRs and the corresponding errors. But for the first time the development of a tool which just outputs the number of events in one SR is enough, because if it is successful the development of a tool for all SRs is straight forward.

points the χ^2 's and # events for other 11 dimensional parameter points, where we have not generated events explicitly to calculate the χ^2 and #events in the SR.

In this thesis the second solution will be explored, because one cannot push the event generation to O(seconds) if one does not want to loose accuracy in the event generation.

This chapter was dedicated to pointing out the overall investigation strategy of the pMSSM-11. Furthermore it became clear that the main task of this master thesis will be to develop tools which speed up the theory predictions for the LHC observables. In order to achieve this goal we have to generate LHC events. In chapter 5 details about the event generation will be given. Chapter 6 describes the available interpolation tools and chapter 7 is devoted to the results using the different interpolation tools.

Finally chapter 8 will give first preliminary results of the global fit which use the developed tools of this master thesis. The preliminary global fit was done by Matthias Hamer but nevertheless the results are presented in this thesis because they are based on the developed tools of this thesis.

5 Simulating LHC events

This chapter gives a detailed overview of the event generation which was needed in order to get the χ^2 and #events in LHC SRs for pMSSM-11 parameter points.

Before going into detail it is explained in short terms what the LHC is and what physical processes we want to simulate. The Large Hadron Collider (LHC) is a particle accelerator located near Geneva which consists of a ring with 27 km circumference and seven different experiments/particle detectors[47],[48]. The two important ones in this thesis are:

- ATLAS (A Toroidal LHC ApparatuS)
- CMS (Compact Muon Solenoid)

In the LHC ring protons and lead-ions can be accelerated and collided, but in this thesis only proton-proton collisions are considered.

The protons are accelerated in opposite directions in the LHC ring and are collided in the center of the particle detectors. The particle detectors will give informations about the detected final state particles of a proton-proton collision. For example the transverse momentum of a measured object and its electrical charge.

One proton-proton collision and the measurement of the properties of its out coming particles is called an 'event'. During a run a lot of events are recorded by the detectors. Afterwards one investigates all events by running several analyses over the events. One analysis has several signal regions (SR). For example one analysis may have a SR where only events with two same sign leptons are counted.

The experiment measures numbers of events in all SRs. The question now is if the model we want to investigate will predict the same numbers. To check this we will have to simulate the event generation at the LHC within the pMSSM-11 framework and then run the LHC analyses over these simulated events. After that we can compare the number of events in the SR from simulation with the experimental measured numbers.

In principle one has to simulate all possible processes (with two protons in the initial state) which can occur in the given model (pMSSM-11) to get the predicted number of the events in the SRs. But usually the experimentalists already give the simulated number of Standard Model (SM) events in the SRs in their papers. Therefore we only need to simulate the processes which are additionally possible in the pMSSM-11¹. For example we do not have to simulate proton-proton \rightarrow gluon-gluon events, but proton-proton \rightarrow gluino-gluino events. The given number of SM simulated events in the SRs are then added to the number of events in the SRs when running the analyses only over the 'pure' pMSSM-11 events.

Now that it is clear what processes we actually have to simulate the whole simulation chain (which has to be run for each pMSSM-11 parameter point) will be explained. Figure 5.1 gives a

¹Due to R-parity conservation processes with two partons in the final state will always two SUSY particles in the final state, e.g. MSSM particles which are not present in the SM.

rough overview. At first we will chose a 11 dimensional parameter point which lies in the ranges which one can see in table 5.1.

parameter	scan range
M_1	[-4000,4000] GeV
M_2	[100,4000] GeV
M_3	[-4000,-400] \cup [400,4000] GeV
$m_{\tilde{Q}_{12}}$	[300,5000] GeV
$m_{\tilde{Q}_3}$	[100,5000] GeV
$m_{\tilde{L}_{12}}$	[100,3000] GeV
$m_{\tilde{L}_3}$	[100,4000] GeV
m_{A^0}	[0,4000] GeV
A^0	[-5000,5000] GeV
μ	[-5000,-100] \cup [100,5000] GeV
$\tan \beta$	[1,60]

Table 5.1: Ranges for the pMSSM-11 parameters where LHC events have been generated. The ranges have been motivated by a previous pMSSM-10 scan [49]. Different to the scan ranges in [49] is that we extend some ranges and avoid to scan low M_2 , $M_3, m_{\tilde{Q}_{12}}, m_{\tilde{Q}_3}, m_{\tilde{L}_{12}}, m_{\tilde{L}_3}$, μ parameters. The gluino pair production cross section will blow up in the case of small M_3 leading to a lot of signal in SRs which focus on strong production processes. Therefore we expect that parameter points with low M_3 are directly excluded. A similar argument applies for the low parameter ranges of $M_2, m_{\tilde{Q}_{12}}, m_{\tilde{Q}_3}, m_{\tilde{L}_{12}}, m_{\tilde{L}_3}$.

Then SPheno [27], [28] computes the complete SLHA file for the chosen pMSSM-11 parameter point. The SLHA file contains the masses of all particles and the decay tables of all particles. On top of that values for low energy precision observables are provided. After that we proceed only in the chain if several preselection criteria are fulfilled:

1. There are no tachyons in the spectrum.
2. SPheno runs without an error.
3. χ_1^0 is the LSP, because it then can serve as a dark matter candidate.
4. Both Higgses (h^0, H^0) have a mass above 110 GeV.
5. $m_{\chi_1^\pm} > 103.5$ GeV [46, p. 1582-1583]
6. The difference between the experimental value and predicted value for the electroweak precision observables

$$m_W, \Delta m_s, \mathcal{B}(B_s \rightarrow \mu\mu), \mathcal{B}(b \rightarrow s\gamma), \mathcal{B}(B_u \rightarrow \tau\nu) \quad (5.1)$$

must be less than 5 times the total error:

$$5 \times \sqrt{\sigma_{\text{experiment}}^2 + \sigma_{\text{theory}}^2} > |\text{theory value} - \text{experimental value}|. \quad (5.2)$$

The observable $g - 2$ of the muon is not part of the preselection because it would restrict the parameter space already at this stage to a very small allowed area. But due to the

fact that we want the interpolation tool in the end to give also predictions for parameter points where the $(g - 2)_\mu$ observable is not predicted well we do not include the observable in the preselection.

If all preselection criteria are fulfilled² then we generate signal SUSY events at a center of mass energy of 8 TeV and 13 TeV respectively.³ The details about the event generations can be found in section 5.1 and 5.2. After the simulation of the events we use Checkmate [50] which

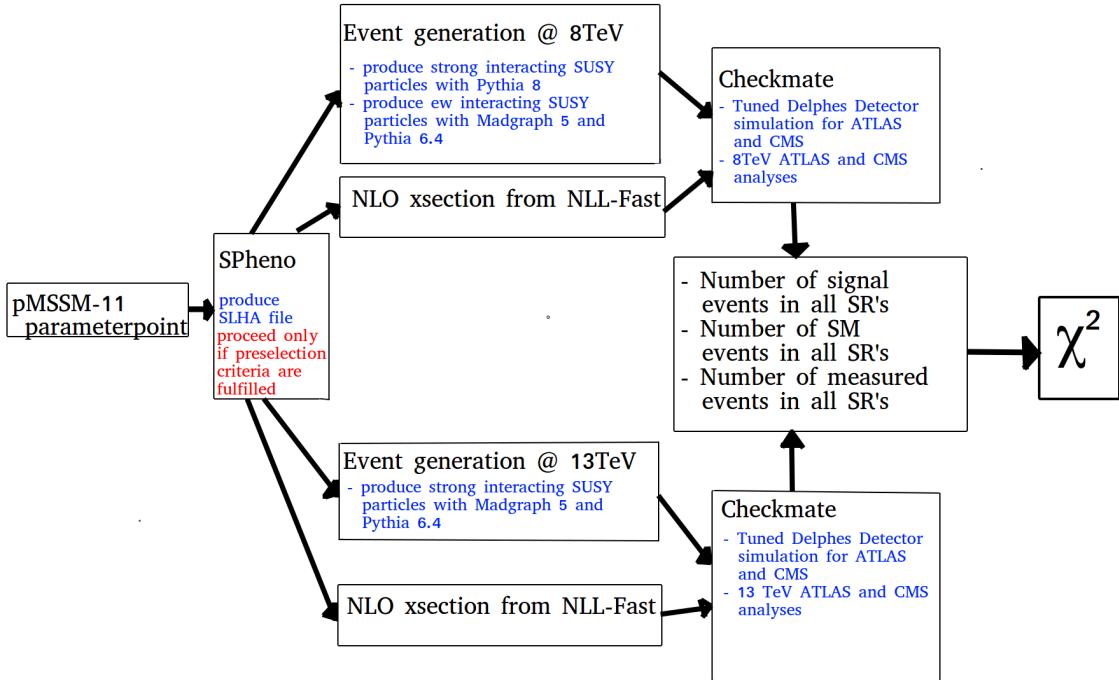


Figure 5.1: Simulation chain

includes a tuned Delphes detector simulation. In Checkmate ATLAS and CMS analyses are implemented which run over the events after the detector simulation. Checkmate already includes a lot of implemented analyses, but in this thesis two additional analyses have been implemented. Details can be found in Appendix A.6.

A more detailed description of the Checkmate output and how it is combined to a LHC χ^2 for 8 TeV and 13 TeV respectively can be found in 5.3.

5.1 Simulating 8 TeV LHC events

At 8 TeV we simulate two classes of processes. The first class of processes has final state particles which do not interact via the strong force (charginos, sleptons,...). The second class of processes includes only strongly interacting particles (gluinos, squarks) in the final states.

First class of processes:

² We saw that if one generates randomly points in the 11 dim parameter space approximately one sixth of all points survives the preselection cuts.

³This means that both protons have an energy of 4 TeV or 6.5 TeV. These energies are chosen, because the LHC carried out their measurements at these energies.

The events are simulated by using Madgraph 5.2.4 [51] and are showered with the pythia 6.428 [52], which is build into Madgraph.

The MadGraph commands are:

```
import model mssm
define ewmssm particles=sve svm svt el- mul- ta1- er-
mur- ta2- sve~ svm~ svt~ el+ mul+ ta1+ er+ mur+ ta2+
n1 n2 n3 n4 x1+ x2+ x1- x2-
generate p p > ewmssm particles ewmssm particles
```

We can generate all events just with one madgraph command, because we do the generation just at LO with no additional jet radiation. We always generate 5000 events and use the CTEQ 6L1 PDF.

The following setting in the madgraph run card are changed from default:

```
5000 = nevents
4000 = ebeam1
4000 = ebeam2
cteq6l1 = pdflabel
5 = asrwgtflavor
5 = maxjetflavor
```

After the madgraph run we read in the LO cross section and its error from the madgraph output. The cross section will be needed afterwards for normalization purposes (see section 5.3).

Furthermore we will estimate if it is necessary to generate more than the 5000 events. With the known cross sections we calculate 5 times the number of events one would expect to see at the LHC:

$$N_{\text{estimated}}^{\text{EW}} = \sigma \times 20 \text{fb}^{-1} \times 5. \quad (5.3)$$

If this number is larger than 5000 we generate $N_{\text{estimated}}^{\text{EW}} - 5000$ more events, but never more than 45000 events.

The first class of processes are only simulated if $m_{\chi_1^0} < 500$ GeV, because the LHC is only sensitive in this range. Being only sensitive to this range means that only in this mass range we expect to see events in the SRs which are designed for the detection of the electroweak events (first class of processes). Figure 5.2 clarifies this issue.

Second class of processes:

Events with strongly interacting SUSY final states are simulated using only pythia 8.2 [54]. We consider 5 different processes. The command files corresponding to all processes and the pythia 8 program can be seen in appendix A.4. The following processes are simulated

1. gluino pair production
2. gluino squark production/ gluino antisquark production
3. squark pair production / antisquark pair production
4. squark-antisquark production
5. strong-electroweak mix production: neutralino-squark, chargino-squark, neutralino-gluino, chargino-gluino production

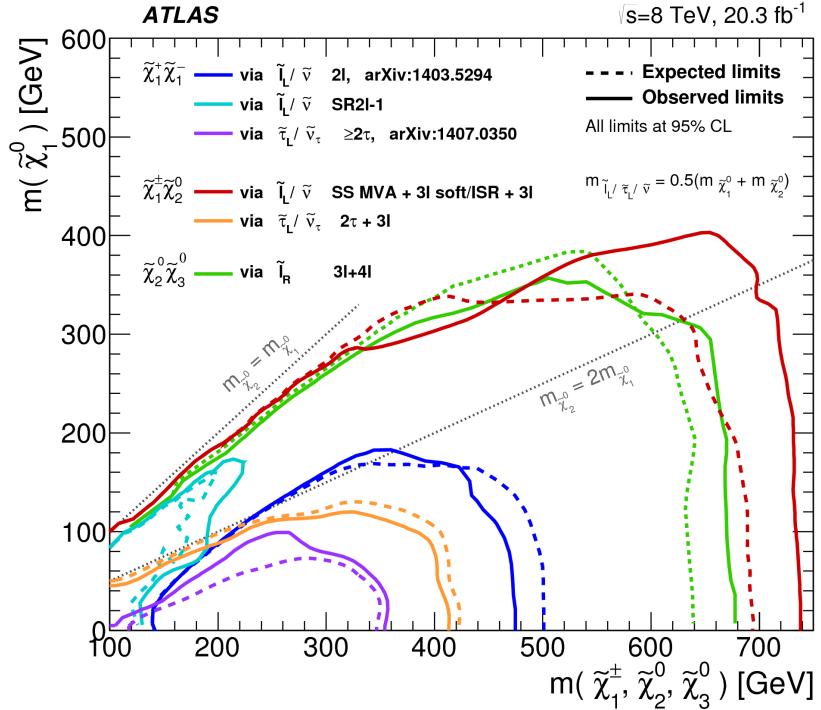


Figure 5.2: The areas below the curves are the 95 CL % excluded areas. This means that in these areas the LHC is sensitive, because one gets a lot of events in these parameter areas which leads to the exclusion. There are different exclusion lines in the plot, because ATLAS used different simplified models and different analyses to derive the different exclusion limits. The figure was taken from [53]

For each process (1-4) we use the NLO cross section to estimate the number of events that we will generate. The NLO cross sections are taken from NLL-Fast [55–62]. To cover the whole range of cross sections we need, the NLL-Fast grid was extended in this thesis. The extension of the NLL-Fast ranges was done with a slightly modified Prospino [55–62] version.

In the case of the first 4 processes we generate

$$\min(\max(1000, \sigma_{NLO} \times 20fb^{-1} \times 5), 5000) \quad (5.4)$$

events. In words we generate 5 times more events than one would expect to see at the LHC, but minimal 1000 events and maximal 5000 events per process. In the case of the last process we generate always 5000 events.

The reason why we generate the processes in the groups 1-4 is simply that we get xsections for these processes separately from NLL-Fast. If we would use LO cross sections for the normalization afterwards it would have been okay to generate all events at once.

5.2 Simulating 13 TeV LHC events

At 13 TeV we simulate only processes with strongly interacting partons in the final state of the process, because there are no analyses which focus on the production of weakly interacting SUSY particles, yet. The simulated processes are:

1. gluino-gluino production
2. squark-gluino/ antisquark-gluino production
3. squark-squark / antisquark-antisquark production
4. squark-antisquark production
5. stop1-antistop1 production
6. stop2-antistop2 production
7. sbottom1-antisbottom1 production
8. sbottom2-antisbottom2 production

The number of events we generate for each process is estimated in the following steps

- For all processes ($i=1\dots 8$) we estimate the number of events that we generate by using again the NLO cross sections from an extended NLL-Fast (for 13 TeV) [55–63] grid.

$$N_{\text{estimated}}^i = \sigma_{\text{NLO}}^i [fb] \times 4fb^{-1} \times 5 \quad (5.5)$$

⁴ If there are processes which have less than 10 estimated events then we do not generate any events for this process.

- Next we add all estimated number of events:

$$N_{\text{estimated}}^{\text{tot}} = \sum_{i=1}^8 N_{\text{estimated}}^i. \quad (5.6)$$

- If $N_{\text{estimated}}^{\text{tot}} > 50000$ we scale down the number of events for the considered processes

$$N_{\text{estimated}}^i \rightarrow N_{\text{estimated}}^i \frac{\sigma_{\text{NLO}}^i}{\sum_i \sigma_{\text{NLO}}^i}. \quad (5.7)$$

Of course the sum runs only over the processes which are still under consideration.

- In the last step we set the minimum of events for one process to 2000:

$$N_{\text{estimated}}^i \rightarrow \min(2000, N_{\text{estimated}}^i). \quad (5.8)$$

We use again Madgraph 5.2.4 [51] and decay and shower everything with pythia 6.248 [52]. If the mass gap between the LSP and one of the particles in the final state is less than 300 GeV we generate the processes with one extra jet in the final state and use MLM matching to merge the additional jet from the matrix element. Details about the Madgraph commands used for the event generation can be looked up in the appendix A.5.

⁴ As integrated Luminosity we choose $4fb^{-1}$, because the implemented 13 TeV analyses all have $3.2 fb^{-1}$ integrated luminosity

5.3 Checkmate

After generating all events for all processes we use Checkmate [50] to evaluate the event files. Checkmate first runs a tuned Delphes detector simulation over the events to account for example for mismeasurements of objects in the detector.

Next ATLAS and CMS analyses which have been re-implemented into Checkmate are run over the events. The following analyses have been used in this study:

- 8 TeV
 - Search for direct production of charginos and neutralinos in events with three leptons and missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector [64]
 - Search for direct production of charginos, neutralinos and sleptons in final states with two leptons and missing transverse momentum in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector [65]
 - 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 [66]
 - Search for direct third-generation squark pair production in final states with missing transverse momentum and two b-jets in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector [67]
 - Search for direct top-squark pair production in final states with two leptons in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector [68]
 - Search for supersymmetry at $\sqrt{s} = 8$ TeV in final states with jets and two same-sign leptons or three leptons with the ATLAS detector [69]
 - Search for squarks and gluinos with the ATLAS detector in final states with jets and missing transverse momentum using $\sqrt{s} = 8$ TeV proton–proton collision data [70],[71]
 - Search for top squark pair production in final states with one isolated lepton, jets, and missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector [72]
 - Search for pair-produced third-generation squarks decaying via charm quarks or in compressed supersymmetric scenarios in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector [73]
 - Search for new phenomena in final states with an energetic jet and large missing transverse momentum in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector [74]
 - Search for supersymmetry in events containing a same-flavour opposite-sign dilepton pair, jets, and large missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector [75] , [76]
- 13 TeV
 - Search for gluinos in events with an isolated lepton, jets and missing transverse momentum at $\sqrt{s} = 13$ TeV with the ATLAS detector [77]
 - Search for supersymmetry at $\sqrt{s} = 13$ TeV in final states with jets and two same-sign leptons or three leptons with the ATLAS detector [78]

- Search for squarks and gluinos in final states with jets and missing transverse momentum at $\sqrt{s} = 13$ TeV with the ATLAS detector [79]
- 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 [80]
- Search for new phenomena in final states with an energetic jet and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector [81]
- Search for new physics in final states with two opposite-sign, same-flavor leptons, jets, and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV [82]
- 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 [83]
- Search for pair production of gluinos decaying via stop and sbottom in events with b -jets and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector [84]

The first two analyses in the 13 TeV list have been implemented during the work on this thesis into Checkmate. The validation of these analyses can be found in the appendix A.6.

In the following the important Checkmate outputs are discussed. As mentioned before we have processes $i = 1, \dots, n^{\text{processes}}$. For example process $i = 1$ for gluino pair production and so on.

The cross sections and their errors for all processes have to be provided as Checkmate input: $\sigma_i, \sigma_{\sigma_i} \quad i = 1, \dots, n^{\text{processes}}$. For the 8 TeV electroweak processes we have one overall LO cross section and its error from AadGraph. The cross section for the mixed process at 8 TeV comes from Pythia 8 (The fifth process in the second class of processes). For all other processes we have NLO cross sections from NLL-Fast. The error on the cross sections from NLL-Fast comes only from the scale uncertainty. The pdf uncertainty is not included in the cross section error. The different analyses are labeled with a subscript $j = 1, \dots, n^{\text{Analyses}}$. Analysis j has signal regions $SR_{jk}, k = 1, \dots, n_j^{\text{SRs}}$, where n_j^{SRs} is the Number of signal regions of analysis j .

Checkmate will now calculate the following numbers

- $\hat{N}_{i,j,k}$ = is the number of signal events from process i in SR k of analysis j .
- $N_{i,j,k}$ is the normalized Number of events from process i in SR k of analysis j . Because the number $\hat{N}_{i,j,k}$ is based on the arbitrary number of generated events \hat{N}_i^{gen} we have to normalize $\hat{N}_{i,j,k}$ to the events we would observe in the experiment.

$$N_{i,j,k} = \hat{N}_{i,j,k} \frac{\mathcal{L}_{\text{int}}^j}{\hat{N}_i^{\text{gen}} / \sigma_i}, \quad (5.9)$$

where \mathcal{L}_{int} is the integrated luminosity.

- $N_{jk} = \sum_{i=1}^{n^{\text{processes}}} N_{i,j,k}$ is the normalized number of events in SR k of analysis j for all processes.

- $\sigma_{N_{jk}}^{\text{stat}} = \sum_{i=1}^{n^{\text{processes}}} \sigma_{\hat{N}_{i,j,k}} \frac{L_{\text{int}}^j}{\hat{N}_i^{\text{gen}}/\sigma_i}$ is the statistical error on N_{jk} , where $\sigma_{\hat{N}_{i,j,k}} = \sqrt{\hat{N}_{i,j,k}}$.
- $\sigma_{N_{jk}}^{\text{sys}} = \sum_{i=1}^{n^{\text{processes}}} \hat{N}_{i,j,k} \frac{L_{\text{int}}^j \sigma_{\sigma_i}}{\hat{N}_i^{\text{gen}}}$ is the systematical error on N_{jk} . The integrated luminosity has also an error, but this is not considered by Checkmate, because it is expected that the other error sources dominate.
- O_{jk} is the number of observed events in SR k of analysis j . These numbers come from the experimental collaborations and are the same for each Checkmate run.
- $B_{jk} \pm \sigma_{B_{jk}}$ is the number of simulated SM "background" events in signal region k of analysis j . These numbers come from the experimental collaborations and are the same for each Checkmate run.
- The logarithm of the profile likelihood ratio ($q_{\mu,jk}$) for each SR k of each analysis j . The detailed calculation of this quantity can be looked up in the appendix A.2. $q_{\mu,jk}$ is χ^2 distributed in the limit of large statistics. In the following we will assume that this is always fulfilled and therefore when we talk about the χ_{jk}^2 which is the χ^2 for SR k of analysis j we actually mean $q_{\mu,jk}$.
If we fix j and k and look at $q_{\mu,jk}$ for different pMSSM-11 parameter points the parameter point which leads to the lowest $q_{\mu,jk}$ coincides best with the experimental measured Number of events for SR k of analysis j .

5.4 Obtaining an overall LHC χ^2

It was already mentioned that we want for each pMSSM-11 parameter point the number of normalized events in all SRs and on top of that an overall LHC χ^2 . The number of normalized events in the SRs are in the Checkmate output, but we still have to combine all χ_{jk}^2 to an overall LHC χ^2 . In this thesis we will investigate two methods to obtain an overall LHC χ^2

1. An intuitive method is just to add all χ_{jk}^2 to an overall χ_{tot}^2

$$\chi_{tot}^2 = \sum_{j,k} \chi_{jk}^2. \quad (5.10)$$

For 8 TeV we add 120 χ^2 s because the 8 TeV analyses have **120** SRs in total. All 13 TeV analysis have **93** SRs in total.

The described method has a big disadvantage. To see this consider 4 SR($k = 1\dots 4$) and one analysis ($j = 1$) SR $k = 1$ will be disjoint form all other SRs and SR $k = 2, 3, 4$ are all overlapping. Now for one pMSSM-11 parameter point we obtain $\chi_{11}^2 = 0.2$ and $\chi_{12}^2 = 13.1$, $\chi_{13}^2 = 12.9$, $\chi_{14}^2 = 13.5$. In this case we have $\chi_{tot}^2 = 39.7$. For another pMSSM-11 parameter point one may obtain $\chi_{11}^2 = 2.5$ and $\chi_{12}^2 = 3.1$, $\chi_{13}^2 = 2.9$, $\chi_{14}^2 = 3.0$, leading to $\chi_{tot}^2 = 8.5$. So one would conclude that the second pMSSM-11 parameter point leads to a model which coincides better with the data than the first one. But one should be careful with this conclusion. SRs 2-4 are overlapping what means that the same events can enter all 3 SRs. Now if the selection criteria of SR 2-4 are very similar then this fact can sophisticate the conclusion.

To see this explicitly select only one of the SRs 2-4 in the calculation of the χ_{tot}^2 . For

the first pMSSM-11 parameter point we will get an χ^2_{tot} of 13.2 and for the second one $\chi^2_{tot} = 5.5$.⁵ In this case now the second parameter point seems to coincide better with the experimental situation, rather than the first one. Therefore the simple addition of all χ^2 form all SRs can lead to wrong conclusions.

2. Now that it is clear that something can go wrong if one adds all χ^2_{jk} , in this thesis an algorithm was implemented which selects the SRs whose χ^2 's can be added.

First in each analysis the SRs are divided into disjoint groups. All SRs of one group are disjoint to all SRs in all other groups. If one group has more than one element the SR with the largest $\frac{N}{S_{exp}^{95}}$ in this group will be selected⁶. For one analysis then one adds the χ^2 's of all selected SRs.

$$\chi_j^2 = \sum_{\text{selected } k} \chi_{jk}^2. \quad (5.11)$$

In the last step all χ_j^2 of all analyses will be added to the overall $\chi_{\text{disjoint}}^2 = \sum_j \chi_j^2$ ⁷. The information for each analysis which SR belongs to which disjoint group can be found in appendix A.3.

Consider an example where we have one analysis $j = 1$ with SRs $k = 1, 2, 3, 4, 5$ (figure 5.3). SR 1 overlaps with SR 2, SR 2 overlaps with SR 3 and SR 4 and 5 are disjoint from all other SRs. In this case the algorithm identifies the groups (1, 2, 3), (4), (5). Now if $\frac{N}{S_{exp}^{95}}$ for SR 1, 2 and 3 are for example (0.2, 0.3, 9.0) the algorithm selects SR 3 for the first group. Therefore the selected disjoint SRs are: (3), (4), (5). In total the algorithm now adds the χ^2 's for SR 3, 4 and 5 $\chi_1^2 = \sum_{k=3,4,5} \chi_{1,k}^2$. And because we just considered one analysis in the example we get the overall $\chi_{\text{disjoint}}^2 = \chi_1^2$.

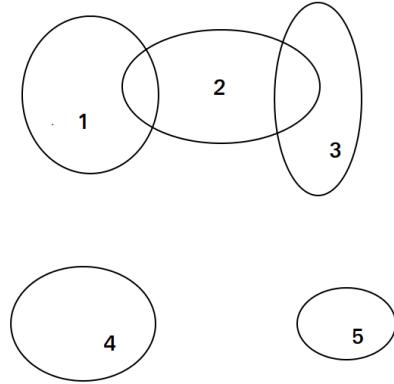


Figure 5.3: Illustration of overlapping SRs

⁵ The selected χ^2 for SRs 2-4 was approximately the χ^2 which was obtained for one SR in 2-4. A correct selection algorithm will be given below

⁶Here the signal for SR k of analysis j is N_{jk} . The S_{exp}^{95} value is provided by the experimental collaborations also for each analysis j and SR k .

⁷We assume that the SRs of different analyses are in a good approximation disjoint, because they all focus on different signatures.

Disrespecting all arguments for using χ^2_{disjoint} instead of χ^2_{tot} we will later on present tools which predict χ^2_{disjoint} and χ^2_{tot} , because the tool which predicts χ^2_{tot} has been used to compare the results in this thesis to other projects which predict χ^2_{tot} .

In the case of the 8 TeV analyses we have always a fixed number of **47** disjoint SRs. In the case of the 13 TeV analysis we have **65** disjoint SRs.

5.5 The sampling algorithms

In the last sections it was pointed out how the simulation for one pMSSM-11 parameter point works in detail and what outputs are produced. In this section it will be explained how sample the pMSSM-11 parameter points. First of all we sample only points in the ranges which are listed in table 5.1.

Basically we use four different sampling algorithms

1. **2048_flat**

This sampling algorithm generates in each parameterrange of table 5.1 two random numbers (with a flat probability distribution) and takes all combinations of these points to form $2^{11} = 2048$ 11-dimensional parameter points. Of course we applied this sampling algorithm not only once, but several times, such that $2048 \times x$ pMSSM-11 parameter points have been sampled.

2. **2048_pdf**

The algorithm works exactly like the **2048_flat** algorithm, but here the generation of the random numbers in the parameter ranges uses a gaussian probability distribution function (pdf) which is centered in the middle of the intervals of table 5.1.

3. **flat**

In this sampling algorithm for each pMSSM-11 parameter point 11 independent random numbers (with a flat pdf) according to the ranges in 5.1 are generated.

4. **pdf**

In this sampling algorithm for each pMSSM-11 parameter point 11 independent random numbers (with a gaussian pdf) according to the ranges in 5.1 are generated.

The reason for the sampling in 2048 packs is motivated by the chosen interpolation technique which will be described in chapter 6. In chapter 6 we will decide to chose a neural net as the interpolation method. Neural nets are motivated by the human brain. Now imagine a human should learn the χ^2 structure of the pMSSM-11 parameter space. You would start by one parameter point and then you would vary some components of this parameter point to see how this changes the χ^2 . With this method you would get a feeling of which directions of the parameter space are flat.

But one could also say that the **2048_flat** sampling algorithm has the disadvantage that it introduced less statistics, because it uses only 22 random numbers to generate 2^{11} pMSSM-11 parameter points. Therefore we did not use only one sampling algorithm but all of them (see tables below). In tables 5.2 and 5.3 we list the number of sampled points with the different sampling algorithms.

id	sampling type	number of sampled points	comment
1	pdf	198000	-
2	2048_flat	207000	-
3	flat	41000	-
4	2048_flat	53000	Areas where the smallest χ^2_{tot} values occur are identified. In these areas the 2048_flat sampling algorithm is applied.
5	flat	87000	Around existing points with $93 < \chi^2_{tot} < 150$ new points are sampled.
6	flat	15000	Around existing points with $\chi^2_{tot} < 87$ new points are sampled.
7	flat	22000	Around existing points with $150 < \chi^2_{tot} < 230$ new points are sampled.
8	flat	28000+26000	Around existing points with $\chi^2_{disjoint} < 31$ new points are sampled.
9	flat	15750+66000	Around existing points with $34 < \chi^2_{disjoint} < 200$ new points are sampled.

Table 5.2: Number of sampled points with the corresponding sampling method for the 8 TeV case. When it is mentioned in the comments that we have sampled around points then we define a small range around each parameter point which leads to a χ^2 in the mentioned range and apply the sampling algorithm in this range (there is no additional sampling around points with ids 4-9). In total for approximately 700000 pMSSM-11 points 8 TeV events have been generated. For 70000 points with id 1-3 there was no extra sampling because in the end we need some unbiased points to check the performance of the interpolation methods (these 700000 points will be used as validation set later on). The reason why we have sampled points in the special target areas (id's 4-9) will become obvious in chapter 7.

id	sampling type	number of sampled points	comment
1	pdf	12000	-
2	2048_flat	18000	-

Table 5.3: Number of sampled points for the 13 TeV case with the corresponding sampling algorithm. In the 13 TeV case there was no extra sampling around chosen points like in the 8 TeV case. In total for approximately 140000 pMSSM-11 parameter points 13 TeV events have been generated.

6 Multivariate interpolation

In the previous chapters 4 and 5 it is pointed out that there is need for an interpolation tool which interpolates the map from the 11-dim parameter space to the space of χ^2 's and # events in the SRs. This chapter starts with a general view on interpolations. In sections 6.1 and 6.2 we then look explicitly at two interpolation methods.

Consider an unknown function

$$\mathbf{f} : U_{\text{in}} \mapsto U_{\text{out}}, \quad \mathbf{x} \mapsto \mathbf{f}(\mathbf{x}), \quad U_{\text{in}} \subset \mathbb{R}^{n_{\text{in}}} \text{ and } U_{\text{out}} \subset \mathbb{R}^{n_{\text{out}}}, \quad (6.1)$$

from which we only assume that it is continuous. n_{out} is the dimension of the codomain and n_{in} is the dimensions of the domain of the function.

We only know the exact values for the unknown function at a finite set of points $\mathbf{x}^i \in U_{\text{in}}$, $i = 1, \dots, N_{\text{full}}$

$$\mathbf{f}(\mathbf{x}^i) = \mathbf{y}^i, \quad i = 1, \dots, N_{\text{full}} \quad (6.2)$$

$$, \quad (\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^{N_{\text{full}}}, \mathbf{y}^{N_{\text{full}}}) \quad \text{full information set.} \quad (6.3)$$

The \mathbf{y} 's are sometimes called targets.

The main task of an interpolation is to construct from the informations (6.3) a function

$$\hat{\mathbf{f}} : U_{\text{in}} \mapsto U_{\text{out}}, \quad \mathbf{x} \mapsto \hat{\mathbf{f}}(\mathbf{x}), \quad (6.4)$$

such that the disagreement between the interpolated function $\hat{\mathbf{f}}$ and the true unknown function \mathbf{f} is minimal. In oder to specify an agreement or disagreement between \mathbf{f} and $\hat{\mathbf{f}}$ we divide the full set into an 'interpolation set' and a 'validation set'.

$$\underbrace{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^{N_{\text{interpol}}}, \mathbf{y}^{N_{\text{interpol}}})}_{\text{interpolation set}}, \underbrace{\dots, (\mathbf{x}^{N_{\text{full}}}, \mathbf{y}^{N_{\text{full}}})}_{\text{validation set}}. \quad (6.5)$$

The information in the 'interpolation set' is used to construct $\hat{\mathbf{f}}$ and the information in the 'validation set' is used to check how big the disagreement between \mathbf{f} and $\hat{\mathbf{f}}$ is. It is not allowed to use the information of the 'interpolation set' for validating if the interpolation works well. For example, one way to check if the interpolation works well would be to evaluate the total error on the validation set

$$\sum_{i=N_{\text{interpol}}+1}^{N_{\text{full}}} \|\mathbf{f}(\mathbf{x}^i) - \hat{\mathbf{f}}(\mathbf{x}^i)\|. \quad (6.6)$$

If this quantity is small then the interpolation algorithm will be likely to work well. We will see in section 7 that it is sometimes not enough that the total error on the validation data is small to make a statement about the power of the interpolation algorithm for a specific problem.

In chapter 7 we will use the techniques presented in this chapter with $n_{\text{in}} = 2$ for the cMSSM with 2 parameters, but mostly with $n_{\text{in}} = 11$ for a pMSSM with 11 parameters. The dimension of the Codomain in our applications will be $n_{\text{out}} = 1$, because we want to interpolate the overall χ^2 and the number of events in one SR N_{jk} .

6.1 Nearest neighbour interpolation

The nearest neighbour interpolation is the simplest interpolation technique. In words the interpolator searches the element $\mathbf{x}^{i_{\min}}$ of the interpolation set which is closest to the point \mathbf{x} . Then the value for $\mathbf{f}(\mathbf{x})$ is interpolated as $\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{y}^{i_{\min}}$.

$$\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{y}^{i_{\min}} \quad \text{with} \quad \min_{i=1, \dots, N_{\text{interpol}}} (\|\mathbf{x} - \mathbf{x}^i\|^2) =: \|\mathbf{x} - \mathbf{x}^{i_{\min}}\|^2. \quad (6.7)$$

6.2 Neural networks

Artificial neural networks, which will be used in this thesis, can be viewed from two different perspectives. On the one side artificial neural networks can be motivated from 'human being neural networks' which one finds in the human body¹. On the other side an artificial neural network can be seen as a highly nonlinear interpolator.

The first section 6.2.1 of this chapter will address the motivation for the artificial neural nets from the 'human being' point of view. In the second section 6.2.2 an abstract overview of artificial neural nets as a nonlinear interpolator will be given.

6.2.1 Human neural networks

The information for this section was taken from [85–88].

Artificial neural nets like described in section 6.2.2 have not been developed out of the box. The idea to develop such artificial neural nets comes from the *nervous system* which every human being has.

The nervous system of a human consists of approximately 10^{11} neurons/nerve cells and can be divided mainly into the *central nervous system* and into the *peripheral nervous system*.

The *peripheral nervous system* consists of all nerves outside the brain and the spinal cord. The nerves inside the peripheral nervous system build a branched net through the whole body. The *central nervous system* consists of our brain and the spinal cord. It saves and handles information which come from the sensory systems. On top of that, it steers motoric actions of the body.

A more detailed differentiation of the human nervous system but can be found in [85]. In this section, we focus on the principal usage of the nervous system and its basic building block, the neurons/nerve cells.

The human nervous system contains all nerve cells/neurons of the human body. It gets its information from the sensory systems and processes these informations leading to a reaction of the human body.

For example, one sensor system of the human being are the eyes. Photoreceptor cells (for example rod cells or cone cells) gain information about the incoming light intensity and color spectrum. The information is forwarded by electrical impulses to other information-processing nerve cells/neurons. These nerve cells which have processed the informations about light intensity and the color spectrum can in turn again forward the processed information. It therefore gives many levels of information-processing with many connected neurons. A picture of two connected neurons can be seen in figure 6.1.

¹Actually there are not only 'human being neural nets', but also 'being neural nets' which can be found in nearly all creatures in this world. In the following we will only capture 'human being neural nets', because these structures have been investigated a lot and the principle of 'human being neural nets' and 'being neural nets' is the same.

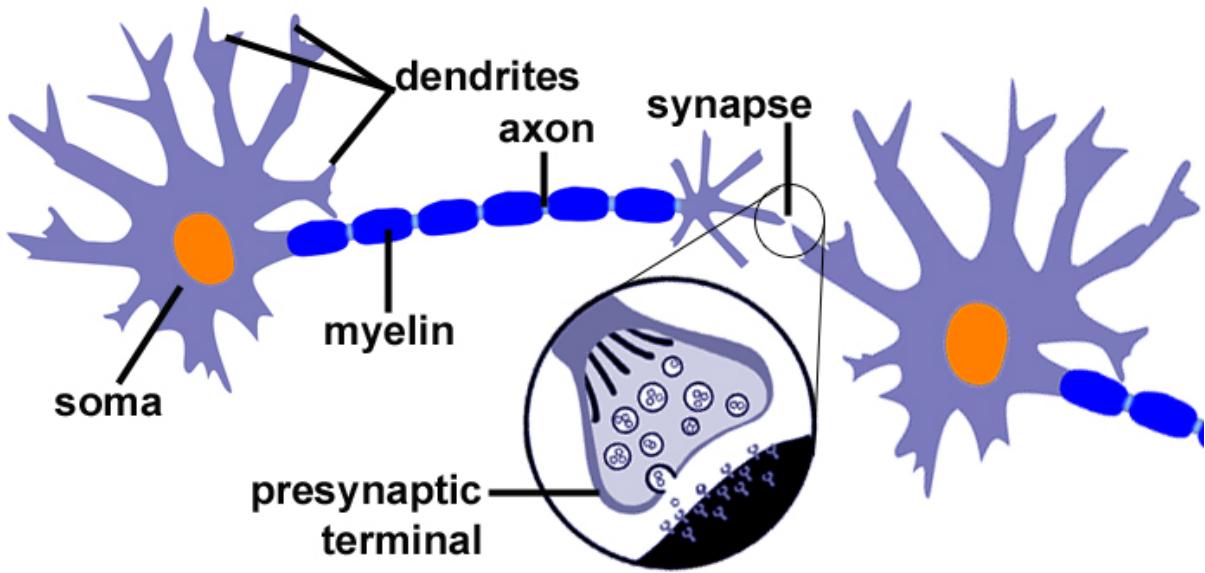


Figure 6.1: Two neurons are shown in this graphic [89]. The center of the neuron is called soma. Information from the neuron on the left to the neuron on the right flows through the axon. The transition between the left neuron and dendrite of the right neuron is called synapse.

The left neuron forwards an electrical signal through the axon. This signal can be transmitted to the dendrites of the right neuron through the synapses. There are two different forms of synapses: *electrical synapses* and *chemical synapses*.

An electrical signal which comes to the presynaptic terminal of an electric synapse will be transmitted directly through an *electrical synapse* to the dendrite of the right neuron.

The more common variant is the *chemical synapse*, shown in graphic 6.1. In this case there is no direct electrical coupling of the presynaptic terminal of the left neuron with the dendrite of the right neuron. The synaptic gap separates them. The electrical signal on the presynaptic side will be converted to a chemical signal and neurotransmitter are released. The neurotransmitter overcomes the synaptic gap and on the other side the neurotransmitter will be converted back into an electrical signal. The important advantage of the chemical transmission over the direct electrical transmission is that in the case of the chemical transmission a lot of different neurotransmitter are available such that the incoming signal can be regulated. One might want to harm or to strengthen the incoming signal.

The soma of the neuron sums up all the incoming signals and 'fires' again an electrical signal if an overall threshold is reached.

6.2.2 Artificial neural networks

This section points out the basic theory behind artificial neural networks and its learning process. The main information is taken from [90].

6.2.2.1 A model for a neuron

Like in human being neural networks the neuron the basic building block of an artificial neural network. The model of a neuron can be seen in figure 6.2 A neuron has n inputs:

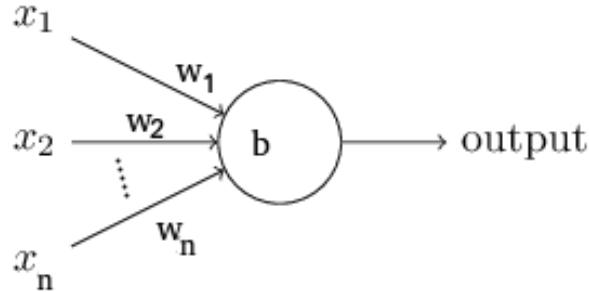


Figure 6.2: Basic building block of a neural network: Neuron [90]

$$x_i \in \mathbb{R}, \quad 1 \leq i \leq n \quad (6.8)$$

Each input to the neuron is multiplied by a **weight**

$$w_i \in \mathbb{R}, \quad 1 \leq i \leq n. \quad (6.9)$$

The total input of the neuron is defined to be:

$$\sum_{i=1}^n w_i x_i + b, \quad (6.10)$$

where a **bias** $b \in \mathbb{R}$ is added. The inputs to the neurons are the analogue to the electric signals which reach the synapses and $x_i w_i$ is the analogue to the transmitted signal in a chemical synapse. The bias is motivated from the threshold which has to be reached in each neuron to fire a signal.

The output of the neuron will be determined by an arbitrary scalar function, called **activation function**:

$$a : \mathbb{R} \mapsto \mathbb{R} \quad (6.11)$$

$$\text{output of Neuron} = a\left(\sum_{i=1}^n w_i x_i + b\right) \quad (6.12)$$

Some popular examples for activation functions are:

- linear activation function $a(x) = x$
- sigmoid activation function $a(x) = \frac{1}{1+e^{-x}}$
- rectified linear activation $a(x) = \max(0, x)$
- tanh activation function $a(x) = \tanh(x)$

- modified Heavyside function $a(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$

The reason why the bias is a threshold modelling of the human neurons can be seen ex-

plicitly in this case: $a(\sum_{i=1}^n w_i x_i + b) = \begin{cases} 0 & \sum_{i=1}^n w_i x_i \leq -b \\ 1 & \sum_{i=1}^n w_i x_i > -b \end{cases}$. If $\sum_{i=1}^n w_i x_i$ is large enough then

the neuron 'fires', otherwise it does not. Actually the modified Heavyside function is not recommended as an activation function in modeling neurons because it is not continuous. With non continuous activation functions the cost function, which will be introduced later on, will not be continuous and the minimization of a non continuous cost function is much more complex than the minimization of a continuous cost function. Instead one could take a smeared variant of the modified Heavyside function, the sigmoid activation function. The tanh activation function is also related to the sigmoid function and therefore also a popular activation function. The linear and rectified linear unit activation functions are motivated by electrical synapses which just forward the electrical signal and do not reweigh it.

An artificial neuron with the modified Heaviside activation function is also called Perceptron.

6.2.2.2 A model for a neural network

We can now build a neural network out of the artificial neurons from section 6.2.2.1. In general there are no rules for constructing a neural network. An example of a general neural network² can be seen in figure 6.3. In this example, the information flows from left to right through the network. Most of the neurons in this example forward their outputs only to neurons which lie further on the right (in the direction of the information flow). But in the example we can also find recurrent neurons, which forward their output information back into their input. Also loops can be found where the output of one neuron can go through some neurons back again to the input of the neuron.

Coming from this general view of a neural nets we define two specific classes of neural networks:

1. Feed forward neural networks: In this case the information flows only in one direction (from left to right). Recurrent neurons and loop structures are not allowed. The neural net is structured in layers (see figure 6.4). If the neurons of one layer are only connected with the neurons in the next layer then the net is called simple. If on top of that all neurons are connected with all neurons in the next layer the net is called in addition 'fully connected'. An example of a fully connected simple feed forward neural net with 4 layers (one input layer, one output layer and two hidden layer) is shown in figure 6.4.
2. Recurrent neural networks are feed forward neural nets which also allow for recurrent neurons and loop structures. These networks are much closer to the structures which one finds in the human nerve system, but it is much harder to train artificial recurrent neural nets because the training algorithms are less powerful. Therefore, we will consider simple feed forward neural nets in this thesis.

²If we talk about neural networks in the following we will always mean artificial neural networks.

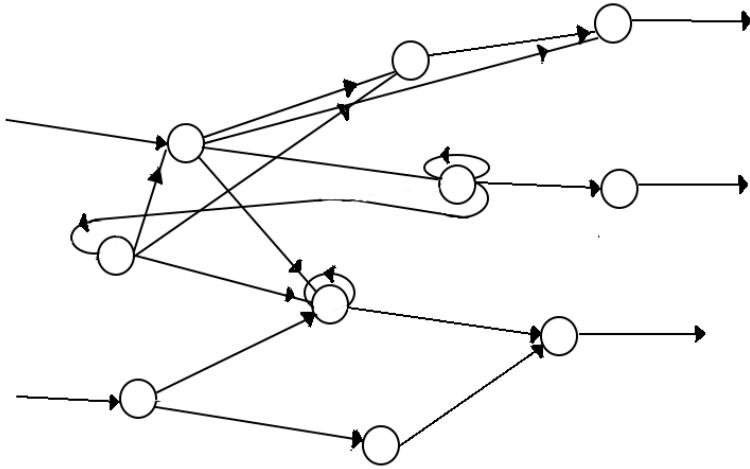


Figure 6.3: Example of a general neural network. The incoming informations enter the network through the two neurons on the left side. The outgoing information comes from the three neurons on the right side.

The nets which are considered **in this thesis will always be fully connected simple feed forward neural nets**. To describe them better we will introduce a convenient notation in the following³.

The net has L layers. Layer 1 is called the input layer and layer L is called the output layer. The layer $l \in \{1, \dots, L\}$ has N^l Neurons. The N^1 inputs of the net are labeled x_1, \dots, x_{N^1} ⁴ and the N^L outputs y_1, \dots, y_{N^L} . They are summarized to vectors:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_{N^1} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{N^L} \end{pmatrix}. \quad (6.13)$$

We call

$$w_{jk}^l, \quad l = 2, \dots, L, \quad j = 1, \dots, N^l, \quad k = 1, \dots, N^{l-1} \quad (6.14)$$

the weight which connects the k th neuron in the $l-1$ th layer to the j th neuron in the l th layer.

$$\mathbf{w}^l, \quad l = 2, \dots, L \quad (6.15)$$

is a $(N^l \times N^{l-1})$ matrix.

$$b_j^l, \quad l = 2, \dots, L \quad (6.16)$$

is the bias of the j th neuron in the l th layer⁵ and

$$o_j^l, \quad l = 1, \dots, L \quad (6.17)$$

³The conventions described in appendix A.1

⁴The x_i 's here are different from the x_i 's from section 6.2.2.1. In section 6.2.2.1 they have been the inputs to one single neuron, but here they are the inputs to the whole neural net.

⁵It starts at $l = 2$, because the input neurons just serve as placeholders which forward the inputs to the second layer.

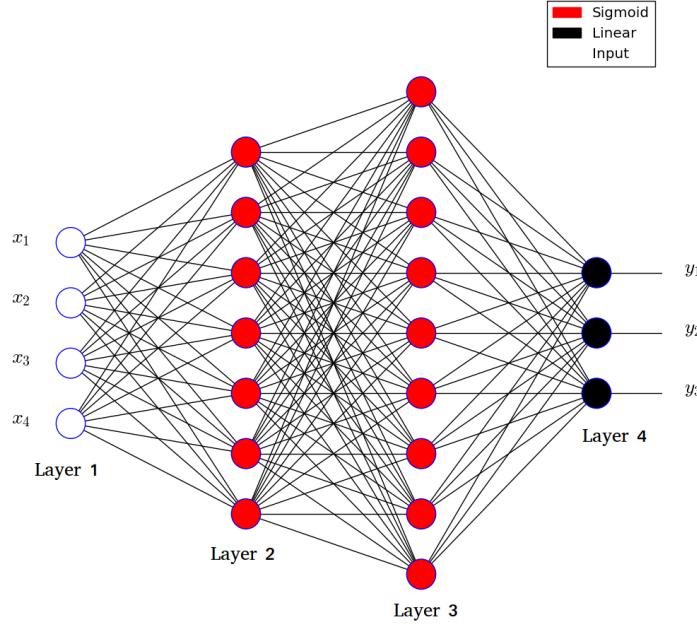


Figure 6.4: Example of a fully connected simple feed forward neural net. In the example all neurons in one layer are connected to all neurons in the next layer ('fully connected') and neurons in one layer are only connected to neurons in the next layer ('simple'). The activation function of one neuron is specified by its color. The net has 4 inputs x_1, \dots, x_4 and 3 outputs y_1, \dots, y_3 .

is called the output of the j th neuron in the l th layer.

The input to the j th neuron in the l th layer is:

$$i_j^l = \sum_{k=1}^{N^{l-1}} w_{jk}^l o_k^{l-1} + b_j^l, \quad l = 2, \dots, L. \quad (6.18)$$

Or in a more compact form:

$$\mathbf{i}^l = \mathbf{w}^l \cdot \mathbf{o}^{l-1} + \mathbf{b}^l, \quad l = 2, \dots, L. \quad (6.19)$$

The inputs $\mathbf{x} = \mathbf{i}^1 = \mathbf{o}^1$ are externally provided⁶.

With this convention we can write the output of the j th neuron in the l th layer as:

$$o_j^l = a^l(i_j^l), \quad l = 1, \dots, L, \quad (6.20)$$

where a^l is the activation function of the neurons in the l th layer. The activation function a^1 of the neurons in the first layer is always the identity, because the neurons in the first layer are

⁶Actually the neurons in the first layer do not have inputs, but they are the inputs (i.e. they serve as placeholders for the inputs). Therefore one has to set $\mathbf{i}^1 = \mathbf{o}^1$ if we want to use the convention which was introduced here.

just placeholders for the inputs of the net.

If we define a 'vectorized' activation function $\tilde{\mathbf{a}}^l$ which will be defined in the following equation:

$$\mathbf{o}^l = \tilde{\mathbf{a}}^l(\mathbf{i}^l) := \begin{pmatrix} a^l(i_1^l) \\ \vdots \\ a^l(i_{N_l}^l) \end{pmatrix}, \quad l = 1, \dots, L, \quad (6.21)$$

one can write for the outputs of the l th layer:

$$\mathbf{o}^l = \tilde{\mathbf{a}}^l(\mathbf{w}^l \mathbf{o}^{l-1} + \mathbf{b}^l). \quad (6.22)$$

For the input \mathbf{x} the overall output of the net is:

$$\mathbf{o}(w_{jk}^l, b_j^l, \mathbf{x}) := \mathbf{o}^L = \tilde{\mathbf{a}}^l(\mathbf{w}^L \mathbf{o}^{L-1} + \mathbf{b}^L) = \tilde{\mathbf{a}}^l(\mathbf{w}^L \tilde{\mathbf{a}}^l(\mathbf{w}^{L-1} \mathbf{o}^{L-2} + \mathbf{b}^{L-1}) + \mathbf{b}^L) = \dots \quad . \quad (6.23)$$

From 6.23 it becomes clear that the output depends on all weights and biases. Therefore the main task is to choose these free parameters such that the discrepancies between \mathbf{f} and $\hat{\mathbf{f}} := \mathbf{o}$ are minimal.

Like it was described at the beginning of this chapter we only use the information which is contained in the interpolation set to construct $\hat{\mathbf{f}}$. In the context of neural nets the construction is also called training phase and the interpolation set is called training set⁷. After the training phase it can be checked if the interpolation was successful by evaluating the total error on the validation set (compare 6.6).

In the following we will explain how the network parameters can be determined.

To judge if the parameters of the net are chosen well, we define a quantity called **cost/loss function**. There are many possibilities of choosing an appropriate cost function, but all must have in common that they have to be minimized with respect to the weights and biases in order to find the best parameters of the net.

One can choose between several cost functions.

- The most intuitive cost function is the *quadratic cost function*

$$C(w_{jk}^l, b_j^l) := \frac{1}{2N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \|\mathbf{y}^i - \mathbf{o}(w_{jk}^l, b_j^l, \mathbf{x}^i)\|^2 = \frac{1}{2N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N^L} C^q(y_j^i - o_j(w_{jk}^l, b_j^l, \mathbf{x}^i)). \quad (6.24)$$

The factor $\frac{1}{2N_{\text{train}}}$ could also be left out, because it does not matter in the minimization process⁸. In formula 6.24 we have set $N_{\text{train}} = N_{\text{interpol}}$ ⁹ and

$$C^q : \mathbb{R} \mapsto \mathbb{R}, \quad x \mapsto x^2. \quad (6.25)$$

⁷This naming is motivated by the fact that a human brain also needs training to learn features of some 'interpolation set'.

⁸Actually it has a reason why it is there. If one writes out the equations for mini batch stochastic gradient descent some equations will have a nicer form. More details in [90]

⁹One can also define a cost function for the validation set while summing over the elements of the validation set. But the cost function which is minimized in order to find the optimal weights and biases is the cost function for the training set because only information of the training set should enter in the weight and bias determination. The training set is another name for the interpolation set which is used often in the context of neural networks.

- If one is only interested in predicting the targets \mathbf{y}^i , $i = 1, \dots, N^{\text{full}}$ within some tolerance-range Δ^i around them¹⁰ one can use the new designed cost function which will be called the *error quadratic cost* in this thesis.

$$C(w_{jk}^l, b_j^l) := \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N^L} C^{\text{eq}}(y_j^i - o_j(w_{jk}^l, b_j^l, \mathbf{x}^i), \Delta_j^i, c), \quad (6.26)$$

where Δ_j^i is the j th component of the tolerance range of the i th training target. Furthermore we define:

$$C^{\text{eq}} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}, \quad (x, \Delta, c) \mapsto \begin{cases} (-x + \frac{c\Delta}{2} - \Delta)^2 + \Delta^2 c (1 - \frac{c}{4}) & x < -\Delta \\ -c x \Delta & -\Delta \leq x \leq 0 \\ c x \Delta & 0 < x \leq \Delta \\ (x + \frac{c\Delta}{2} - \Delta)^2 + \Delta^2 c (1 - \frac{c}{4}) & x > \Delta \end{cases}, \quad (6.27)$$

where $c \in (0, 1)$ is a small constant which has to be chosen by the user. Its meaning becomes obvious from figure 6.5. A comparison of the error quadratic cost and the simple quadratic cost can be seen in figure 6.5.

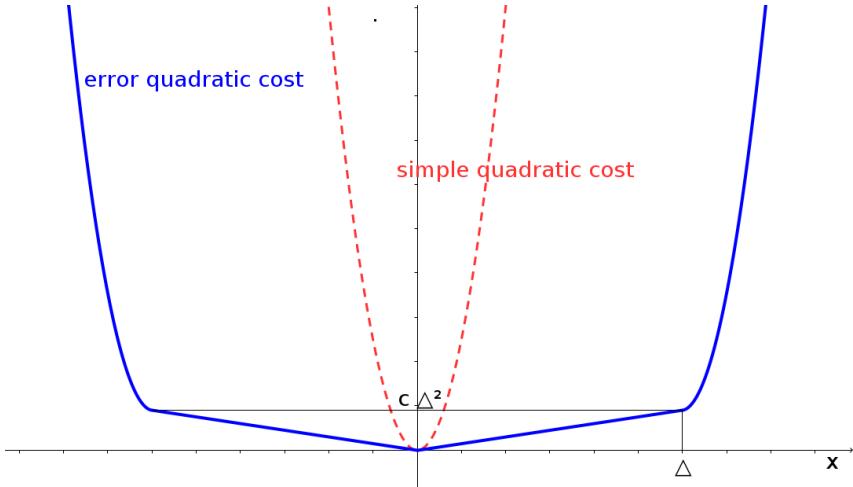


Figure 6.5: Comparison of simple quadratic cost 6.25 and error quadratic cost 6.27.

The error quadratic cost was designed such that the total value of the cost function and its derivative are less than the values one obtains with the simple quadratic cost in the tolerance range $(-\Delta, \Delta)$ ¹¹. This is also the reason why c must be between zero and one. The error quadratic cost function gives a larger error if the neural network predicts the targets outside of the tolerance range. Therefore the minimization algorithms¹² which try to reduce the cost function will spend more effort in reducing the error on the targets

¹⁰For example if the targets have errors σ^i and one is satisfied if the targets are predicted within the $1-\sigma$ range one could set $\Delta^i = \sigma^i$

¹¹Actually this is not true for the whole tolerance range. For $x \in (-c\Delta, c\Delta)$ the values for the simple quadratic cost are higher and the derivative is also larger. But due to the fact that one typically chooses $c \approx 10^{-4} \dots 10^{-2}$ this range is negligible small.

¹²The different algorithms are presented further below

which are outside the tolerance range than on targets which are already predicted within the tolerance range.

- The cross-entropy cost function is defined as

$$C(w_{jk}^l, b_j^l) = -\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \sum_{j=1}^{N^L} y_j^i \ln(o_j(w, b, \mathbf{x}^i)) + (1 - y_j^i) \ln(1 - o_j(w, b, \mathbf{x}^i)). \quad (6.28)$$

where y_j^i is the j th component of the desired output corresponding to the i th training element.

This cost function satisfies two basic conditions:

1. It is always positive. This is the case only when the outputs of the neurons in the last layer are between 0 and 1. In this case both terms in the sum are negative and the overall minus sign makes everything positive. The cross entropy cost function was designed especially for sigmoid neurons in the output layer¹³. In this case the condition is of course fulfilled that the outputs of the neurons in the last layer are between 0 and 1.
2. If outputs of the net are close to the desired outputs the cost function has a minimum.

Now that it is clear that we have to minimize the chosen cost function with respect to the weights and biases in order to find the optimal weights and biases the minimization algorithms which are used to achieve this goal will be presented. In the following consider a scalar function

$$g : V \mapsto \mathbb{R}, \quad \mathbf{z} \mapsto g(\mathbf{z}), \quad V \subset \mathbb{R}^n. \quad (6.29)$$

¹⁴.

- The simplest minimization algorithm is the *Gradient Descent Algorithm*. Expanding the function g yields:

$$g(\mathbf{z} + \Delta \mathbf{z}) \approx g(\mathbf{z}) + (\nabla g)(\mathbf{z}) \cdot \Delta \mathbf{z} + \frac{1}{2} (\Delta \mathbf{z})^T (\partial^2 g)(\mathbf{z})(\Delta \mathbf{z}) + \dots \quad (6.30)$$

with $(\partial^2 g)_{ij} = \partial_i \partial_j g$.

For now we will consider terms which are first order in $\Delta \mathbf{z}$

$$g(\mathbf{z} + \Delta \mathbf{z}) \approx g(\mathbf{z}) + (\nabla g)(\mathbf{z}) \cdot \Delta \mathbf{z}. \quad (6.31)$$

If we choose $\Delta \mathbf{z} = -\eta(\nabla g)(\mathbf{z})$, $\eta > 0$

$$g(\mathbf{z} + \Delta \mathbf{z}) \approx g(\mathbf{z}) - \eta((\nabla g)(\mathbf{z}))^2, \quad (6.32)$$

we have $g(\mathbf{z} + \Delta \mathbf{z}) < g(\mathbf{z})$.

This principle can be used to find the minimum of the function $g(\mathbf{z})$. After choosing an initial value \mathbf{z}_0 the algorithm could look as follows:

¹³ When one uses a quadratic cost function in combination with sigmoid neurons a learning slowdown problem occurs. The cross entropy cost function was designed especially to avoid this. More details can be found in [90]

¹⁴ Typically one has $n = n_{in}$ and $\mathbf{z}^T = (w_{jk}^2, \dots, w_{jk}^L, b_j^2, \dots, b_j^L)$

1. Calculate $(\nabla g)(\mathbf{z}_0)$ and update $\mathbf{z}_1 = \mathbf{z}_0 - \eta(\nabla g)(\mathbf{z}_0)$
2. Calculate $(\nabla g)(\mathbf{z}_1)$ and update $\mathbf{z}_2 = \mathbf{z}_1 - \eta(\nabla g)(\mathbf{z}_1)$
3. and so on

Of course the parameter η does not have to be constant in all steps. For example we could use a η_1 in step 1 and $\eta_2 \neq \eta_1$ in step 2. η is a so called **hyperparameter** which has to be set by the user such that the learning process works best. There is no general strategy how to set the hyperparameter. We will see in the following that much more hyperparameters will emerge.

The disadvantage about the gradient descent algorithm is that one can run into a local minimum (if the learning rate η is not chosen well). Another disadvantage could be that one gets $g(\mathbf{z}_i) < g(\mathbf{z}_{i+1})$ for some i , because the learning rate η is too large and we need more terms from the taylor expansion.

- A slight modification of the gradient descent algorithm is the *momentum based gradient descent* algorithm. In this case one starts with an initial value \mathbf{z}_0 and \mathbf{v}_0 and has the following update rules:
 1. Calculate $(\nabla g)(\mathbf{z}_0)$ and update $\mathbf{v}_1 = \mu\mathbf{v}_0 - \eta(\nabla g)(\mathbf{z}_0)$, and $\mathbf{z}_1 = \mathbf{z}_0 + \mathbf{v}_1$
 2. Calculate $(\nabla g)(\mathbf{z}_1)$ and update $\mathbf{v}_2 = \mu\mathbf{v}_1 - \eta(\nabla g)(\mathbf{z}_1)$, and $\mathbf{z}_2 = \mathbf{z}_1 + \mathbf{v}_2$
 3. and so on ...

The momentum based gradient descent algorithm introduced two hyperparameters μ and η .

- The *Adam Optimizer*[91] is also a minimization algorithm which relies only on first order information. The algorithm has four hyperparameters $(\alpha, \beta_1, \beta_2, \epsilon)$ which have to be chosen at the beginning but can be changed after each training epoch. Before we start with the algorithm we set start values: $\mathbf{z}_0, \mathbf{m}_0 = 0, \mathbf{v}_0 = 0$

1. Step $t = 1$:

$$\begin{aligned} - \mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)(\nabla g)(\mathbf{z}_{t-1}) \\ - \mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)((\nabla g)(\mathbf{z}_{t-1}))^2 \\ - \hat{\mathbf{m}}_t &= \frac{\mathbf{m}_t}{1 - \beta_1^t} \\ - \hat{\mathbf{v}}_t &= \frac{\mathbf{v}_t}{1 - \beta_2^t} \\ - \mathbf{z}_t &= \mathbf{z}_{t-1} - \alpha \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon) \end{aligned}$$

2. Step $t = 2$: analogous...

All operations on vectors are element-wise. Details about this algorithm can be found in the corresponding paper [91].

- Until now the learning algorithms only used 'first order information' what means that we expanded f only up to first order 6.30. One can also use the second order informations. One can then show [90] that we need the following update rule:

1. Calculate $(\nabla g)(\mathbf{z}_0)$ and $(\partial^2 g)(\mathbf{z}_0)$ and then update $\mathbf{z}_1 = \mathbf{z}_0 - \eta((\partial^2 g)(\mathbf{z}_0))^{-1}(\nabla g)(\mathbf{z}_0)$
2. Calculate $(\nabla g)(\mathbf{z}_1)$ and $(\partial^2 g)(\mathbf{z}_1)$ and then update $\mathbf{z}_2 = \mathbf{z}_1 - \eta((\partial^2 g)(\mathbf{z}_1))^{-1}(\nabla g)(\mathbf{z}_1)$

3. and so on...

In practice this strategy is computationally much more expensive because one also needs the second order derivatives where for the gradient descent approach one only needs first order derivatives('first order information').

Of course there are many other minimization algorithms, but the three first order algorithms listed above have been tested in this thesis. We found empirically (see chapter 7) that the Adam Optimizer does the best job.

It is important to note that all minimization algorithms use the backpropagation algorithm to compute the gradients which are always needed for the minimization algorithms. Of course the gradients could also be computed approximately with the difference quotient, but it turns out that the backpropagation algorithm is computationally more effective. A detailed description about the backpropagation algorithm can be found in [90].

In the following the difference between offline and online learning will be pointed out.

Online vs Offline learning:

To exemplify the difference between the two learning approaches consider a general cost function which has the following form

$$C(w_{jk}^l, b_j^l) = \sum_{i=1}^{N_{train}} C_i(w_{jk}^l, b_j^l), \quad (6.33)$$

where C_i depends on all weights and biases. Furthermore the index i should signalize that this quantity depends on the input x^i of the neural net.

The learning procedure called *offline learning*¹⁵ [85] works as follows. One starts with initial values for the weights and biases \mathbf{z}_0 . The vector of weights and biases is updated during the learning epochs as follows:

1. Training epoch:
Calculate $(\nabla C)(\mathbf{z}_0)$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_1$.
2. Training epoch:
Calculate $(\nabla C)(\mathbf{z}_1)$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_2$.
3. :

The gradient which is needed in each step by the minimization algorithm will be calculated using the backpropagation algorithm which gives the gradient for each C_i respectively. The overall gradient which is needed in each leaning epoch can then easily be calculated by adding them:

$$\nabla C = \sum_{i=1}^{N_{train}} \nabla C_i.$$

This strategy has the disadvantage that all elements of the training data are evaluated at once¹⁶ and maybe the whole data does not fit in computer memory.

¹⁵'Batch learning' is an alternative name.

¹⁶In the cost function one sums over all training samples.

Contrary to the batch learning strategy is *online learning* which uses every element of the training set separately to update the weights and biases. The algorithm starts again with choosing initial values for the weights and biases \mathbf{z}_0 .

1. Training epoch:

- a) Calculate $(\nabla C_1)(\mathbf{z}_0)$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_1$.
- b) Calculate $(\nabla C_2)(\mathbf{z}_1)$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_2$.
- c) \vdots
- d) Calculate $(\nabla C_{N_{train}})(\mathbf{z}_{N_{train}})$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_{N_{train}+1}$.

2. Training epoch: shuffle training set randomly and

- a) Calculate $(\nabla C_1)(\mathbf{z}_{N_{train}+1})$ and update weights and biases by using the rules of a minimization algorithm $\rightarrow \mathbf{z}_{N_{train}+1}$.
- b) \vdots

3. \vdots

The Gradients for C_i come directly from the backpropagation algorithm.

The disadvantage of this algorithm is that the cost function might fluctuate strongly during the learning phase.

Therefore the optimal learning algorithm is a compromise of batch learning and online learning and called *mini-batch learning*. The mini-batch algorithm introduces another hyperparameter the *batch size b* and works as follows:

1. training epoch:

- a) batch 1 = $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_b, \mathbf{y}_b)$, define the cost function only for this batch $C_{b1} = \sum_{\text{batch 1}} C_i$ and update weights and biases by using ∇C_{b1} .
- b) batch 2 = $(\mathbf{x}_{b+1}, \mathbf{y}_{b+1}), \dots, (\mathbf{x}_{2b}, \mathbf{y}_{2b})$, define cost function only for this batch $C_{b2} = \sum_{\text{batch 2}} C_i$ and update weights and biases by using ∇C_{b2} .
- c) \vdots

2. training epoch: shuffle training set randomly and

- a) batch 1 = $(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^b, \mathbf{y}^b)$, define the cost function only for this batch $C = \sum_{\text{batch 1}} C_i$ and update weights and biases.
- b) batch 2 = $(\mathbf{x}^{b+1}, \mathbf{y}^{b+1}), \dots, (\mathbf{x}^{2b}, \mathbf{y}^{2b})$, define cost function only for this batch $C = \sum_{\text{batch 2}} C_i$ and update weights and biases.
- c) \vdots

3. \vdots

Mini batch learning with $b = 1$ is equal to online learning and with $b = N_{train}$ to batch training. In chapter 7 we will do hyperparameter scans and to find out which b does the best job.

All mentioned minimization algorithms are already implemented in TensorFlow [92], which is a python library and is used to implement all artificial neural nets in this thesis.

After the discussion of online and offline learning we will come to a feature called *overfitting*.

Overfitting

In many cases one can observe that the minimization algorithm still reduces the cost function on the training data, but at the same time the total error on the validation data¹⁷ does not decrease anymore or even increases. The described situation is called overfitting and occurs, because the neural net still learns features of the training set, but these features are no general features one can also find in the validation set. In this case the parameters of the net are chosen such that the net represents especially the training data well. We want to avoid the scenario of overfitting, because we are looking for a neural net which is general also to data which didn't enter the learning process. There are several techniques to avoid overfitting.

- The simplest technique is *early stopping* which stops the whole learning procedure if the total error on the validation data saturates or increases.
- A more unintuitive method is to regularize the cost function by adding a term

$$\frac{\lambda}{2N_{train}} \sum_{l,j,k} (w_{jk}^l)^2, \quad \lambda > 0 \quad \text{L2 regularization} \quad (6.34)$$

or

$$\frac{\lambda}{2N_{train}} \sum_{l,j,k} |w_{jk}^l|, \quad \lambda > 0 \quad \text{L1 regularization.} \quad (6.35)$$

This method works well in many applications. A descriptive reason why it works can be found for example in chapter 3 of [90].

If one decides to use a regularization technique one will introduce a new hyperparameter λ .

If we use the quadratic and cross entropy cost function the regularization terms look like the terms in (6.35) and (6.34), but if we use the error quadratic cost we have only a factor λ in front of the sums in (6.35) and (6.34) to be consistent with the definitions of the cost functions.

- A technique which can also be used to avoid overfitting is *dropout*. Dropout is applied to selected layers of the network¹⁸ and introduces a new hyperparameter p for each layer where one decides to apply the dropout technique¹⁹.

For example we apply dropout with a dropout keep probability of $p = 0.5$ to the second layer of the network which has 20 neurons. The learning algorithm will then pick randomly in each learning epoch $10 = 20 \times 0.5$ neurons of the second layer and ignore them during the minimization step. In contrast the outputs of the 10 neurons in the second layer which have not been ignored will be scaled by $p^{-1} = 2$.

¹⁷Or equivalently the cost function evaluated with the validation data

¹⁸Of course one can decide to apply dropout to all layers of the network.

¹⁹One calls the parameter p the dropout keep probability. If one decides to apply dropout to multiple layers one can of course choose different dropout probabilities for the different layers p_1, p_2, \dots

This technique can also be motivated very well[90], but there is no general proof why it should help to avoid overfitting.

During this section the initialization of the weights and biases was mentioned often but it was never said how it is actually done. The next paragraph will clarify this issue.

Weight and bias initialization:

There are many possibilities to initialize the weights and biases of the network. For example one could initialize the weights and biases all to zero. Another possibility is to do a random initialization with a gaussian distribution (mean zero and standard deviation one). In the case of sigmoid activation functions it can be shown that due to the fact that the derivative of the sigmoid function is small for large values a learning slowdown occurs if we initialize the weights with a standard normal distribution [90]. The solution is to initialize the weights which connect layer l and $l - 1$ with a gaussian distribution which has mean zero and standard deviation $\frac{1}{N_{l-1}}$. The biases are still initialized with a standard normal distribution. In this thesis the weights and biases are always initialized this way, because we will use sigmoid and the closely related tanh activation a lot.

The next subtlety which should be discussed is called *Feature Scaling* and was also implicitly assumed in many steps before. The next section will clarify where it was assumed and what it does.

Transformations on the network input and outputs/Feature Scaling:

The information in this paragraph is mainly from [93].

Sometimes it is useful to transform the outputs \mathbf{y}^i and inputs \mathbf{x}^i and then train the net with these transformed values²⁰.

The transformation which one applies on the outputs has to be invertible in order to be able to back transform the outputted values of the neural net. The transformation on the inputs does not have to be invertible. One just has to make sure that one applies always the same transformation to every input of the net.

While a transformation of the inputs is not necessary²¹ it can however often be very helpful for the net to learn the data. For example if the net consists only of sigmoid neurons it turned out to be very helpful to normalize also the inputs to a range between zero and one²². In the following we will give an overview of the different transformations which have been tested in this thesis. Which one of the techniques will be applied depends on the specific problem and the choice of activation functions in the input and output layers.

For the following itemization of transformations consider a set $\mathbf{u}^1, \dots, \mathbf{u}^N$ ²³

- *min-max scaling:*

In order to perform the min-max scaling one first determines the maximum $u_{\max,j} =$

²⁰For example if the neural net has sigmoid activations in the output layer the net can clearly only output values between zero and one. But usually the desired outputs are not in this range and therefore one has to normalize the targets $\mathbf{y}^i \rightarrow \hat{\mathbf{y}}^i$. The net will then be trained on these normalized targets. If the net then should make a prediction one has to backtransform the outputted result.

²¹In the case of a sigmoid output the normalization of the outputs is necessary.

²²This is again due to the fact that without normalization of the inputs it can easily happen that $\sum x_i w_i$ is very large/low for a neuron in the first hidden layer. As a consequence the neurons will output either zero or one and then it is clear that the network will have problems to learn the data, because of the vanishing derivative of the activation function for large inputs.

²³ \mathbf{u} can be an input $\mathbf{u} = \mathbf{x}$ or an output $\mathbf{u} = \mathbf{y}$.

$\max_i(u_j^i)$ and minimum $u_{\min,j} = \min_i(u_j^i)$. The min-max normalization looks as follows:

$$\hat{u}_j^i = \frac{u_j^i - u_{\min,j}}{u_{\max,j} - u_{\min,j}}. \quad (6.36)$$

This normalization technique will normalize all \hat{u}_j^i such that they are in the range between zero and one and is therefore appropriate for a neural net with sigmoids in the output layer. Furthermore the transformation is invertible.

- *Z-score normalization*

For this normalization technique one determines first the mean $\mu_j = \sum_i(u_j^i)$ and the standard deviation $\sigma_j = \sqrt{\frac{1}{N-1} \sum_i (u_j^i - \mu_j)^2}$. The Z-score normalization then works as follows:

$$\hat{u}_j^i = \frac{u_j^i - \mu_j}{\sigma_j}. \quad (6.37)$$

If one for example has tanh activations in the output layer one could apply the further transform on \hat{u}_j^i

$$\hat{x}_j^i = \frac{\hat{u}_j^i}{\max_i |\hat{x}_j^i|}. \quad (6.38)$$

- *log-normalization*

Sometimes the dataset contains values which differ by orders of magnitude and the net struggles to learn these structures. The transformation

$$\hat{u}_j^i = \ln(u_j^i - \min_i u_j^i + 2) \quad (6.39)$$

will reduce the effect of large difference in the orders of magnitude.

Another technical aspect is the *learning slowdown* during the training which can be used to get a better convergence of the neural net.

Learning slowdown:

In this thesis we use the following learning slowdown: after 10 learning epochs we check if the slope of a line which has been fitted to the last 10 validation errors is larger than some threshold m_t ²⁴. If this is the case the learning rate of the minimization algorithm will be reduced by 1/2. This procedure is illustrated in figure 6.6.

²⁴Which will be set

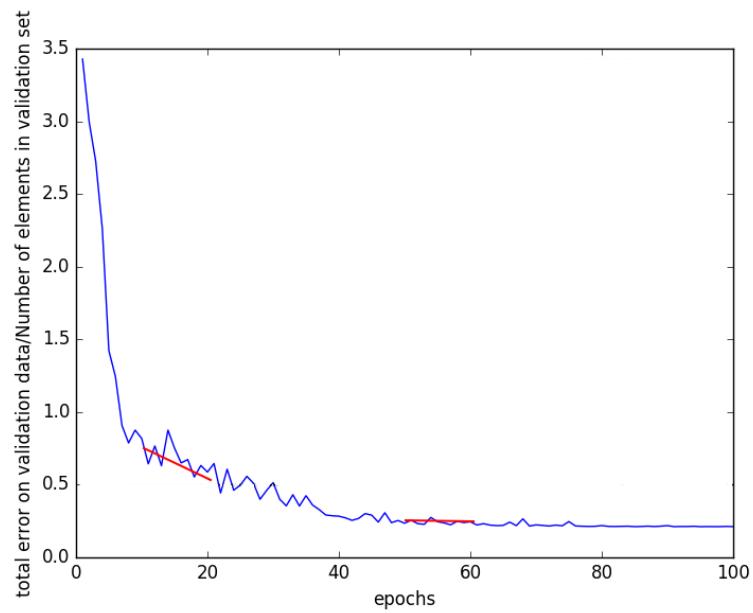


Figure 6.6: Illustration of the learning slowdown procedure. After epoch 20 the line which was fitted to the last 10 validation errors has a highly negative slope and therefore the learning rate will stay the same, but the slope of the line which is fitted after the 60th learning epoch is close to zero and therefore the learning rate will be reduced.

7 LHC neural networks

This chapter describes the development of the neural nets which help to make the comparisons of the LHC observables quicker. On the way to neural nets for the pMSSM-11 there have been tests with neural nets for a cMSSM. Therefore, this chapter starts with neural nets for a two parameter cMSSM (see section 7.1). In section 7.2 we explore neural nets for the pMSSM-11. During the investigation of neural nets for the cMSSM-2 and pMSSM-11 we carry out several hyperparameter scans to determine the best structure of the neural nets and the optimal values for the hyperparameters which have been introduced in section 6.2.2. The results of these hyperparameter scans will be summarized in section 7.3 by rule of thumbs. Finally in section 7.4 one can find an outlook.

7.1 Tests with a 2 parameter cMSSM

In the case of the cMSSM we use a dataset from previous Fittino studies [16]. In total we have $N_{full} = 55029$ scanned parameter points

$$(\mathbf{p}^1, (\chi^2)^1), \dots (\mathbf{p}^{N_{full}}, (\chi^2)^{N_{full}}), \quad \mathbf{p} = \begin{pmatrix} M_0 \\ M_{1/2} \end{pmatrix}. \quad (7.1)$$

The remaining parameters of the cMSSM are fixed to $\tan\beta = 10$, $A_0 = 0$, $\text{sign}(\mu) = +1$. The χ^2 in this study is slightly different from the one that we use for the pMSSM-11. The χ^2 which is used in the cMSSM case is a negative logarithmic profile likelihood (see equation (A.9)) but in the global Likelihood one restricts the signal strength μ to positive values. Furthermore one uses only one SR of the analysis [94] (details are in [16]).

The full set is split up into a validation set (10000 tuples) and the training set (45029 tuples). Before we do anything with the data we first apply a smooth cut on the full data set

$$\chi^2 \rightarrow \begin{cases} \chi^2 & \chi^2 \leq \chi_{max}^2 - 10 \\ -\frac{1}{40}(\chi^2)^2 + \frac{(\chi_{max}^2+10)\chi^2}{20} - \frac{(\chi_{max}^2)^2-20\chi_{max}^2+100}{40} & \chi_{max}^2 - 10 < \chi^2 \leq \chi_{max}^2 + 10 \\ \chi_{max}^2 & \chi^2 > \chi_{max}^2 + 10 \end{cases}, \quad (7.2)$$

where we have chosen $\chi_{max}^2 = 50$. The original data can be seen in figure 7.1a and the cutted data in figure 7.1b. The reason for the cutting is that we do not want that the neural net learns uninteresting details where the χ^2 is very high and the points are excluded anyway. Therefore all upcoming studies in this section will be done with the cutted data.

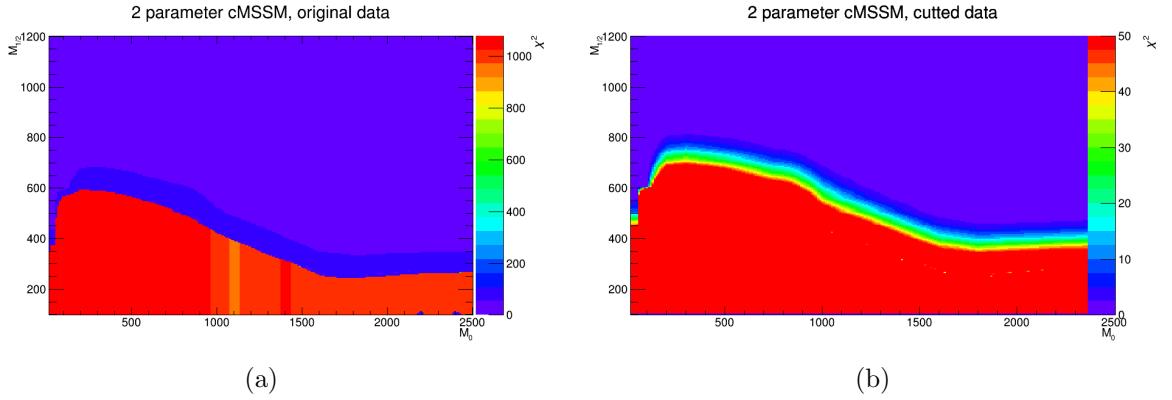


Figure 7.1: (a) shows the original data as obtained from the previous Fittino study. (b) shows the smoothly cutted data. Both histograms contain all 55029 points of the full set.

The neural network should learn the structure which one can see in figure 7.1b. As it was pointed out in section 6.2 there is no general strategy how to choose the hyperparameters of a neural net such that the neural net gives the best performance¹. In order to find the best hyperparameter for the given problem we do a hyperparameter scan. Table 7.1 shows all scanned hyperparameters. We test neural nets which always have three hidden layers (see figure 7.2). The number of neurons in the hidden layers will always be equal in this hyperparameter scan (30, 80, 150 neurons respectively). The batch size is the hyperparameter of the mini-batch learning strategy. λ is the L-2 regularization hyperparameter. Furthermore we vary the dropout keep probabilities in the different layers. The last three varied hyperparameters are three of the four hyperparameters from the Adam optimization algorithm (the parameter α which is not varied and set to its default value.).

hyperparameter	scanned values
number neurons in hidden layers	30, 80, 150
cost function	quadratic,cross
batch size	30, 80 ,1000 , 10000
λ	0.1, 0.01, 0.001 , 0.0001, 0.00001
dropout probability in layer 1	0.3,0.7, 1.0
dropout probability in layer 2	0.3,0.7, 1.0
dropout probability in layer 3	0.2,0.7, 1.0
learning rate	1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001
β_1	0.9, 0.3
β_2	0.999, 0.8

Table 7.1: Scanned hyperparameters.

¹As hyperparameter choice one can see for example the choice of the structure of the neural net, the choice of the cost function, the choice of the learning algorithm with its parameters, the choice of the regularization technique and so on.

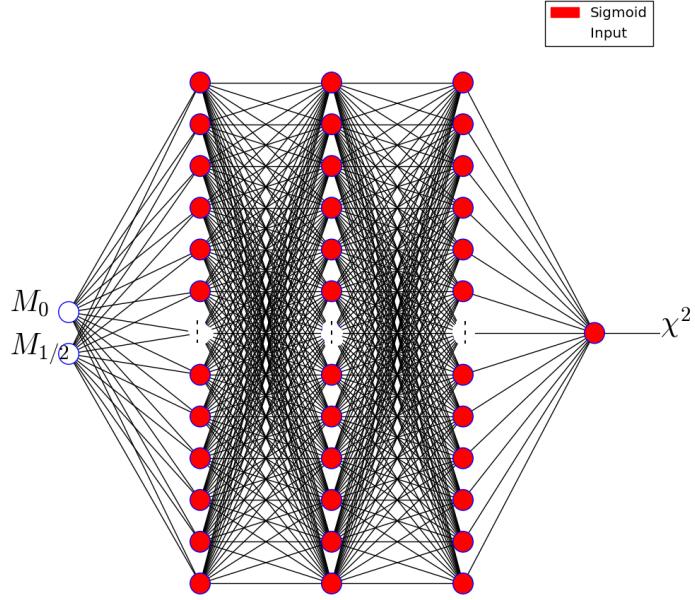


Figure 7.2: Neural network structure for the hyperparameter scan.

The activation functions in this problem are always sigmoid activation functions as can be seen in figure 7.2. Therefore a min-max scaling was applied to the inputs and outputs before the training. Furthermore we always used the Adam optimization algorithm, because initial tests before the actual hyperparameter scan proved that this minimization algorithm does the best job.

In total Total 77760 hyperparameter configurations have been scanned what means that we have trained 77760 different neural networks. Typically when one does a hyperparameter scan with a fixed validation set one has to introduce in addition to the validation set a test set in order not to overfit the hyperparameters to the validation set. But this was not necessary in this study because we do not use a fixed validation set (e.g. the points in the validation set are always the same). Before each scanned hyperparameter the validation set is chosen randomly out of the full set and therefore we do not need to introduce a test set. The hyperparameters cannot be overfitted to the validation set because we use a different validation set for each scanned hyperparameter.

The five best results for the scanned hyperparameters can be seen in table 7.2. The criterion which was used to determine the 'best' hyperparameter was the total error on the validation set averaged over the last 10 training epochs. The cross entropy cost function is clearly preferred over the quadratic cost function. The lowest allowed regularization parameter was 10^{-5} . Furthermore the scan shows that dropout is not helpful because all dropout keep probabilities are one. The optimal learning rate is at 0.01 but only the β_2 parameter is clearly preferred to be 0.999 which is also the default tensorflow value. On top of that the scan is not sensible to β_1 because there

is no clear preference for this hyperparameter. The optimal batch size is 80 because 4 of the 5 best hyperparameters have this batch size. The optimal number of neurons in the hidden layers is somewhere between 30 and 80.

#neurons in hid. layers	cost	batch size	λ	dropout 1/2/3	learning rate	β_1	β_2
80	cross	80	10^{-5}	1.0/1.0/1.0	0.01	0.3	0.999
30	cross	30	10^{-5}	1.0/1.0/1.0	0.01	0.9	0.999
30	cross	80	10^{-5}	1.0/1.0/1.0	0.01	0.9	0.999
30	cross	80	10^{-5}	1.0/1.0/1.0	0.01	0.3	0.999
80	cross	80	10^{-5}	1.0/1.0/1.0	0.01	0.9	0.999

Table 7.2: The five best hyperparameters from the hyperparameter scan (the best one is the first one in the table going down to the fifth best hyperparameter.).

The results of a neural net with the best found hyperparameter configuration are presented in the following². Figure 7.3 shows on the y-axis the total error on the validation data (e.g. equation (6.6)) divided by the number of points in the validation set. Therefore the y-axis of 7.3 shows the mean error on points in the validation set. At the end of the training phase the neural net performs better than the nearest neighbour interpolator. The mean error of the neural net is approximately 0.2 and the mean error of the nearest neighbour interpolator 0.5.

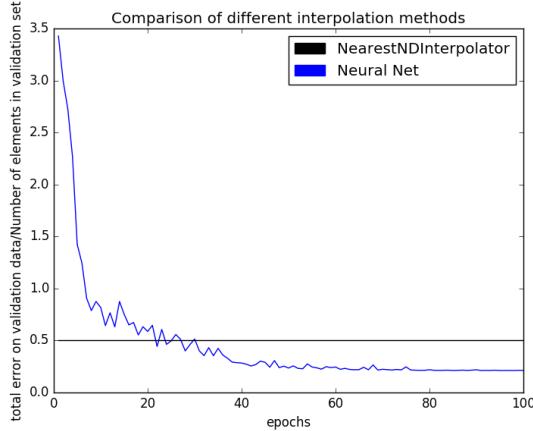


Figure 7.3: Mean error on points in the validation set. In blue one can see the results for the neural net after each learning epoch. The black line represents the result from a nearest neighbour interpolation.

In figure 7.4a one can see the error histogram of the neural net after the last training epoch. There is clearly a peak at zero which is a clear indicator that the net performs well. The same histogram which one sees in 7.4a can be seen in 7.4b, but here a second dimension with the desired χ^2 is added such that one can see for which desired χ^2 the neural net makes the biggest error. There are clearly some outliers at 250 and at zero, but most points in these ranges are predicted well. The χ^2 ranges which have a larger error are lying in the transition region between

²In the actual hyperparameter scan the learning slowdown algorithm was not used, but in the figures which will be shown in the following it has been applied. Nevertheless the difference is not too big, but in the case when one uses the learning slowdown the convergence of the total error on the validation data curve (see figure 7.3) is a bit better.

0 and 250. This can also be seen in figures 7.5a and 7.5b. Figure 7.5a shows the predictions of the neural net after the last training epoch for the full set. One can compare the predictions to the true data (figure 7.1b). Figure 7.5b then shows the absolute errors. Here one can clearly see that the regions with low and high χ^2 are predicted nearly perfectly, but some points in the transition region still have a larger error.

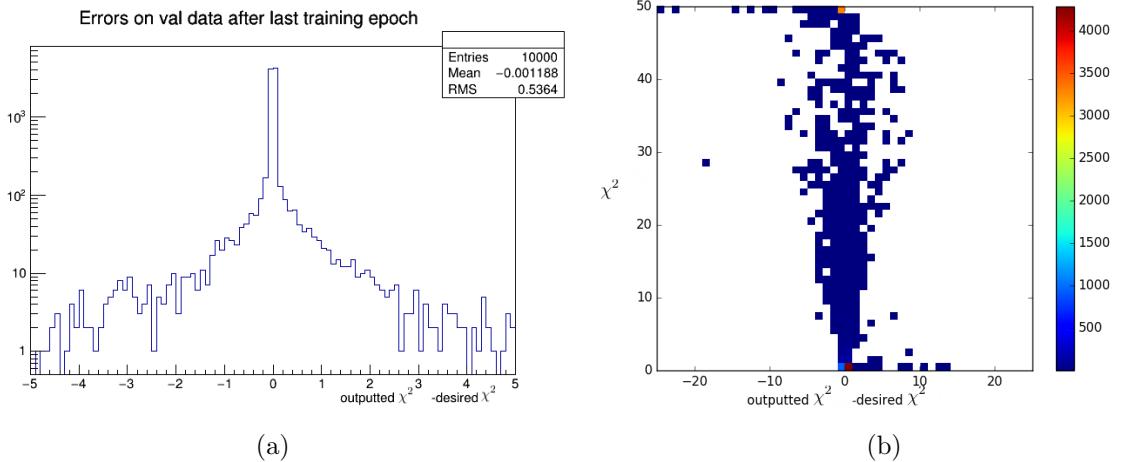


Figure 7.4: (a) shows the error histogram after the last training epoch and (b) shows exactly the same histogram with one additional dimension such that one can identify χ^2 areas where the neural net makes large errors.

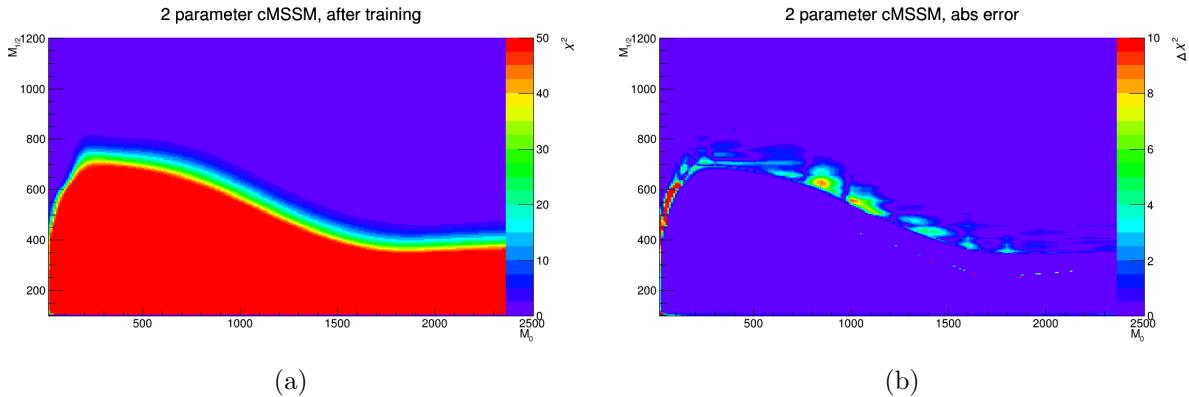


Figure 7.5: (a) shows the predictions of the neural net after the last training epoch and (b) the absolute error in relation to figure 7.1b

All in all it is very impressive that the neural net performs better than the nearest neighbour interpolator even though the points on the grid lay very dense to each other. This result encourages us to go to the pMSSM-11 where we will not have such a high point density due to the high dimension of the parameter space.

7.2 Neural nets for the pMSSM-11

During the work for this thesis we have generated LHC events at 8 and 13 TeV in the pMSSM-11 framework. For these events we have calculated the LHC χ^2 s and the number of events in all considered SR of all analyses (described in chapter 5). For 8 TeV (see section 7.2.1) and 13 TeV (see section 7.2.2) we will develop neural networks which predict χ_{tot}^2 and χ_{disjoint}^2 respectively for a pMSSM-11 parameter point. Only the neural nets for χ_{disjoint}^2 will be used later on in Fittino studies because in χ_{tot}^2 also χ^2 s from overlapping SRs will be added. Nevertheless we present also neural nets which output χ_{tot}^2 because these networks have been developed first and the χ_{disjoint}^2 networks are build up on the results for the χ_{tot}^2 networks.

Finally subsection 7.2.3 describes neural nets which do not output a χ^2 but the number of events in one SR of a LHC analysis.

In short we have the following data sets which the neural networks will learn:

$$(\mathbf{p}^1, y^1), \dots (\mathbf{p}^{N_{\text{full}}}, y^{N_{\text{full}}}), \quad \mathbf{p} = \begin{pmatrix} M_0 \\ M_{1/2} \end{pmatrix}. \quad (7.3)$$

with \mathbf{p} being a 11 dimensional pMSSM-parameter and the target y can be χ_{tot}^2 , χ_{disjoint}^2 or the number of events in one LHC SR. For the 8 TeV case we have approximately $N_{\text{full}} = 700000$ and for the 13 TeV case we have approximately $N_{\text{full}} = 140000$.

7.2.1 LHC 8 TeV χ^2 neural net

This subsection describes the development of neural nets which predict 8 TeV LHC χ^2 s. As it was pointed out before we will develop one neural net for χ_{tot}^2 and one for χ_{disjoint}^2 .

7.2.1.1 χ_{tot}^2

In this case we have 120 signal regions whose χ^2 s are all added. All χ^2 s will again be cut like in the cMSSM (e.g. (7.2)) smoothly with $\chi_{\text{max}}^2 = 250$.

Previously we saw in the case of the cMSSM that we have to do a hyperparameter scan to get a feeling of the optimal configured neural net. Because here we have a different problem (for example the neural nets here have 11 inputs instead of two), we will carry out a new hyperparameter scan. The scanned hyperparameters can be found in table 7.3. We allow 2,3,4,5 hidden layers which all have tanh activation functions (see figure 7.6). All layers can have 50, 150 or 450 neurons respectively. Furthermore we test if the quadratic cost or cross entropy cost works best. A hyperparameter which was newly introduced is the exponential damping hyperparameter. If exponential damping is on then the summand in the cost function corresponding to the desired χ_{des}^2 will be multiplied with $\exp(-\frac{5}{\chi_{\text{max}}^2}\chi_{\text{des}}^2)$. This has been done in order to give χ^2 s which are small a larger weight in the cost function. If they have a larger weight the minimization algorithm will spend more power on minimizing the terms with a lower χ^2 because we are interested in the end in a χ^2 minimum and want to predict the minimal χ^2 s with higher precision than the larger χ^2 s. The regularization parameter scan range now goes down to 10^{-6} due to the results for the cMSSM. Also the learning rate hyperparameter scan range is now a bit more restricted due to the cMSSM hyperparameter scan. The other Adam minimization algorithm hyperparameters are set to their default values, what is motivated by the cMSSM hyperparameter scan. Furthermore we allow only two dropout probabilities which are applied alternating to the hidden layers (e.g if we scan a neural net with four hidden layers

then the first layer has dropout probability 1 the second dropout probability 2 the third dropout probability 1 and the fourth dropout probability 2.). Finally the activation function in the last layer which contains only one neuron is varied between linear and tanh.

hyperparameter	scanned values
number of hidden layers	2,3,4,5
number neurons in hidden layers	50, 150, 450
cost function	quadratic,cross
exponential damping	on / of
batch size	80 ,500, 3000
λ	0.001 , 0.0001, 0.00001,0.000001
learning rate	0.1, 0.01, 0.001, 0.0001
dropout probability 1	0.9,0.95, 1.0
dropout probability 2	0.9,0.95, 1.0
activation in last layer	tanh / linear

Table 7.3: Scanned hyperparameter configurations.

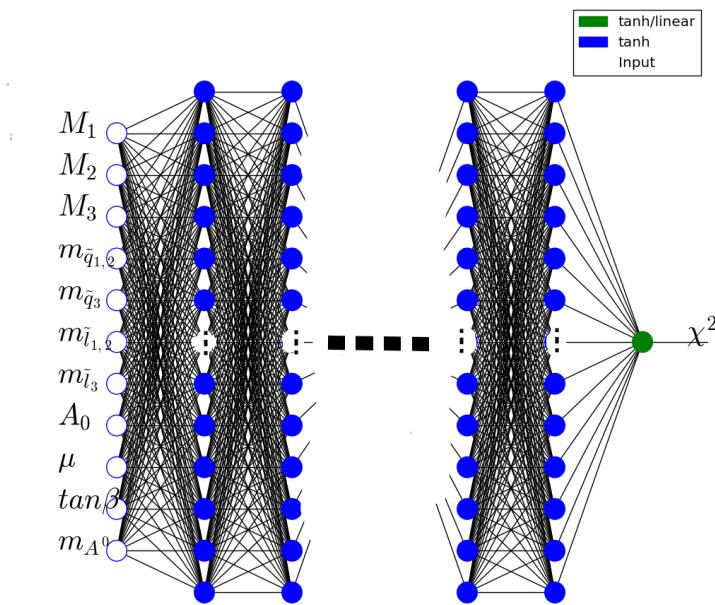


Figure 7.6: Network structure during the hyperparameter scan.

All hidden layers have tanh activation functions and we have used the Z-score normalization with a following normalization to values between -1 and 1 for the inputs and outputs of the neural net. Again the Adam minimization algorithm was used because it worked already well

in the case of the cMSSM. The choice of tanh activation functions and the Adam minimization algorithm will be done for all χ^2 neural networks from now on.

In total 31104 hyperparameter configurations have been scanned and the five best³ results can be seen in table 7.4.

Here contrary to the cMSSM hyperparameter scan the quadratic cost function is preferred over the cross entropy cost function which is due to the fact that we use tanh activations instead of sigmoid activations. Again dropout seems not to be helpful because all dropout keep probabilities are 1. Also one can say that a tanh activation function in the output neuron is more effectively than the linear activation function. The learning rate should be set to 0.001 which is less than the learning rate we found for the cMSSM. The found batch size is larger as in the cMSSM which is not surprising because we use a much larger training sample in this case. A regularization parameter between 10^{-5} and 10^{-6} seems appropriate from the scan results. For the number of hidden layers the statement of the scan is that four hidden layer is the best choice. The number of neurons in the hidden layer is not so clear from the five best hyperparameters.

# h. layer	#neurons	cost	damp.	batch size	λ	learn. rate	dropout 1/2	activation
4	150	quadratic	1.0	500	10^{-5}	0.001	1.0/1.0	tanh
4	150	quadratic	0.0	500	10^{-5}	0.001	1.0/1.0	tanh
4	50	quadratic	1.0	500	10^{-6}	0.001	1.0/1.0	tanh
5	450	quadratic	0.0	80	10^{-5}	0.0001	1.0/1.0	tanh
4	50	quadratic	0.0	500	10^{-4}	0.001	1.0/1.0	tanh

Table 7.4: The five best hyperparameters from the scan. The best one is the first one in the table going down to the fifth best hyperparameter.

A neural net with the best hyperparameter from table 7.4 will now be used in the following. During the hyperparameter scan there has not been a fixed validation set because we do not want to introduce a test set and until now only the sampled data with id's 1-3 (see table 5.2) has been used⁴.

Figure 7.7a shows that the mean error for the nearest neighbor interpolator is 25 and for the neural net we get a mean error of 3. This was expected because now we have in contrast to the cMSSM case not a dens grid of points.

Figure 7.7b shows the same as 7.7a but here in different χ^2 ranges. First of all the errors of the nearest neighbour interpolator are larger at the end of the learning process for all ranges. Secondly, now focusing on the neural net curves in 7.7b the mean error in the different ranges varies a lot. For example the error for χ^2 's which are actually in the range $87 < \chi^2 \leq 90$ is 0.6, but the error for χ^2 's which actually are in the range $0 < \chi^2 \leq 87$ is 9.1.

This feature can be understood if we look at the distribution of the scanned χ^2 's (see figure 7.8). One can see a peak in the χ^2 distribution at 88 and at 250. The peak at 250 is due to the fact that we cut very large χ^2 's and set them smoothly to 250. The peak at 88 corresponds to 'zero signal χ^2 's. One obtains the zero signal χ^2 when all analyses have nearly zero signal. This scenario happens very often if one just generates randomly parameter points in the parameter space. The ranges $0 < \chi^2 \leq 87$ and $90 < \chi^2 \leq 230$ will be called *rare target areas* in the following.

³To determine the best hyperparameters we have used the same criterion we have used before in the cMSSM.

⁴ It will become obvious why we also sampled the data with id 4-9 in a moment.

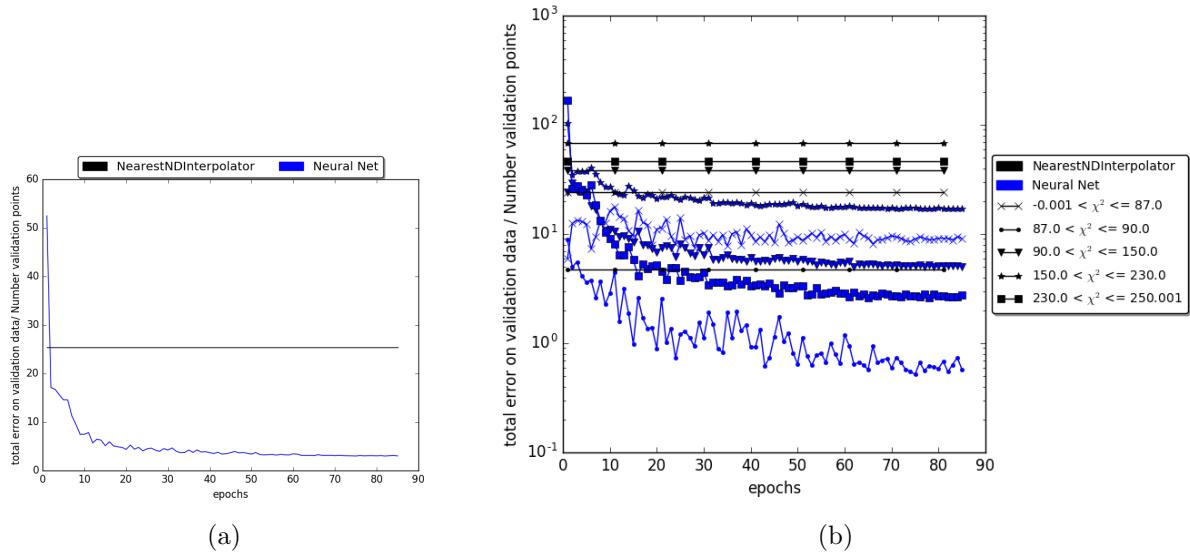


Figure 7.7: (a) shows the comparison of neural network and nearest neighbour interpolation. The mean error on points in the validation set is 8 times larger for the nearest neighbour interpolator. (b) shows the mean errors of the neural network and the nearest neighbour interpolator are in the chosen χ^2 ranges.

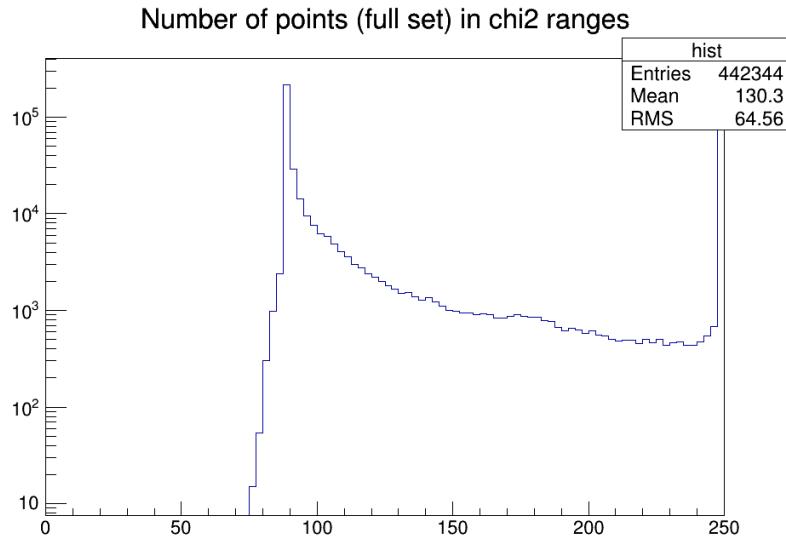


Figure 7.8: χ^2 distribution for all parameter points in the full set.

The mean errors in the rare target areas are large because the network learns the χ^2 s which appear very often better (these are the zero signal χ^2 s and the χ^2 around 250.). This problem will be called the rare target learning problem (RTLP) in this thesis. In the following we will present solution strategies to the RTLP:

1. sampling in rare target areas:

One can identify areas in the 11 dimensional parameter space which lead to χ^2 s which are

in rare target areas. Then one can sample especially new points in these areas. Another very similar approach is to sample around existing parameter points which lead to χ^2 's in the rare target areas. These two strategies have been applied for χ^2_{tot} and χ^2_{disjoint} targets (points with id's 4-9 in table 5.2).

2. artificial extension:

The artificial extension of points in rare target areas is a possible solution. For example one duplicates the pMSSM-11 parameter points which lead to χ^2 's in a rare target areas . The duplicated points get the same χ^2 as their original points, but one component of the 11-dimensional parameter point is slightly modified (one chooses randomly a component and adds 0.01)

3. sequence learning:

One trains the neural net not always with the full training set. For example for two epochs one trains the neural net with the full training set and then for one epoch one uses only the data in the training set which has target values in the rare areas. This will be repeated over and over. The training with the rare target data happens with a reduced learning rate.

We have now presented three solution strategies for the RTLP. To find the optimal solution we will do a second hyperparameter scan, but this time with all data (id's 1-9 in table 5.2)⁵ The hyperparameter scan will be based on the best hyperparameter we found in the previous hyperparameter scan (first line in table 7.4). Table 7.5 shows the scanned hyperparameters. The first hyperparameter determines if we use the data which was sampled in the rare target areas. Another hyperparameter determines if the data will be extended artificially and finally one hyperparameter is for the sequence learning technique. Because we use much more data here as in the previous hyperparameter scan we have introduced two other hyperparameters which should determine if it is helpful to multiply the number of neurons in the hidden layers (from previous best found hyperparameter) by a constant factor. The last hyperparameter in table 7.5 tests if the result improves if one multiplies the batch size of the previous best found hyperparameter by a factor 1.5.

hyperparameter	scanned values
use data sampled in rare target areas	yes/no
extend data artificially	yes/no
use sequence learning	yes/no
multiply number of neurons in first hidden layer by (x_1)	1.0 , 1.5, 2.0
multiply number of neurons in hidden layer 2,3,4 by (x_2)	1.0, 1.5, 2.0
multiply batch size by (x_3)	1.0, 1.5

Table 7.5: Scanned hyperparameter configurations.

The two best hyperparameters⁶ form the second hyperparameter scan are in table 7.6. Form table 7.6 one can see that it is useful to use extra sampling in rare target ranges but sequence

⁵One could say that in this case one should only use ids 1-7 because id's 8 and 9 are sampled points for χ^2_{disjoint} and here we deal with χ^2_{tot} . But we expect that the points which are rare target points for χ^2_{disjoint} are also rare target points for χ^2_{tot} .

⁶The best hyperparameters have been chosen such that the total error on the validation data is small and the error in the rare target ranges is minimal.

learning and artificial extension of the data in rare target areas does not help to solve the RTLP. Furthermore it is helpful to extend the number of neurons in the first hidden layer. The number of neurons in all other hidden layers remains the same while for the extension of the batch size no final statement can be made.

extra sampling	artificial extension	seq. learning	x_1	x_2	x_3
yes	no	no	2.0	1.0	1.0
yes	no	no	2.0	1.0	1.5

Table 7.6: Two best hyperparameters from second hyperparameter scan

In the following the results for a neural net with the best hyperparameter configuration (line one in table 7.6) are presented. In table 7.7 one can see the total mean error and the mean error in all ranges for the validation set⁷. The first line in the table gives the errors corresponding to a neural network with the best hyperparameter configuration which was found after the second hyperparameter scan and the second line gives errors for the best found hyperparameter after the first hyperparameter scan (without the extra sampled data in rare target areas). The conclusion is that the results with the hyperparameters from the second scan are slightly better. The total mean error is lower by 0.2 and also for most rare target areas the errors in the first line are lower.

$0 < \chi^2 \leq 250$	$0 < \chi^2 \leq 87$	$87 < \chi^2 \leq 90$	$90 < \chi^2 \leq 150$	$150 < \chi^2 \leq 230$	$230 < \chi^2 \leq 250$
2.8 (0.021)	8.9 (0.105)	0.54 (0.006)	4.8 (0.042)	17.5 (0.096)	2.3 (0.0095)
3.0 (0.022)	9.1 (0.107)	0.58 (0.006)	5.0 (0.044)	17.3 (0.094)	2.6 (0.011)

Table 7.7: The first column is for the whole validation set and the other columns are for the different χ^2 ranges. In parentheses we give the mean relative error. The first line corresponds to a neural net with the best hyperparameter configuration from the second scan (line one in table 7.6). The second line corresponds to a neural net where the best hyperparameter from the first scan (line one in table 7.4) determines the network properties. In parentheses we give the mean relative errors.

We showed that the RTLP can be attacked by sampling more points in the rare target areas. The other two approaches to solve the RTLP have not been used because the hyperparameter scan showed that they are not very helpful in solving the RTLP. Nevertheless the mean errors are still very different in different areas and there is still need for an improvement. In the outlook 7.4 we propose a method which could solve the RTLP completely.

The following plots correspond to the best hyperparameter configuration of the second scan (line one in table 7.6). Figures 7.9a and 7.9b show the mean errors for the range ($0 < \chi^2 \leq 250$) and for all ranges respectively.

⁷Again during the second hyperparameter scan there was no fixed validation set, but the results which will be presented here and in the following for 8 TeV all have the same unbiased validation set which was mentioned in the caption of table 5.2.

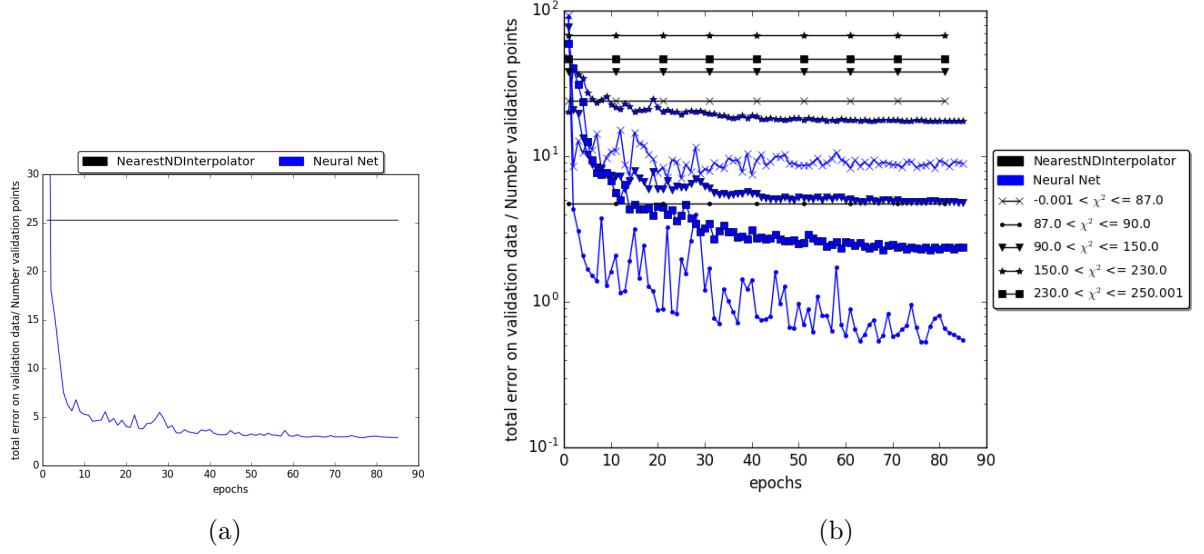


Figure 7.9: (a) shows the mean errors on validation set for the total χ^2 range and (b) shows the mean errors for all χ^2 ranges respectively.

In figure 7.10 we show the classification efficiency (e.g. the amount of points in the validation set which are predicted with an error of one or smaller.) The neural net has a classification efficiency of 70 % while the nearest neighbour interpolator has an efficiency of 52 %.

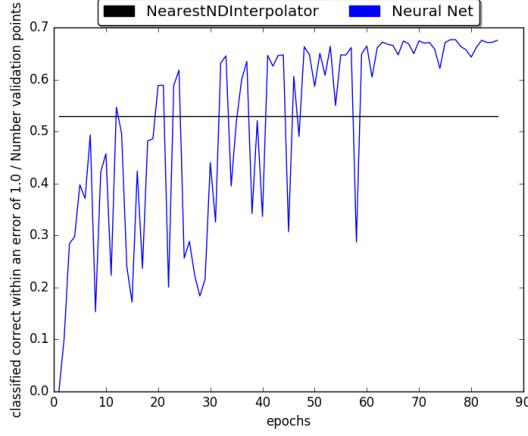


Figure 7.10: Classification efficiencies for all points in the validation set.

Finally in figure 7.11a one can see the error histogram comparison between the nearest neighbour interpolator and the neural net. As we expect from the other plots the nearest neighbour interpolator has much larger errors than the neural net. In figure 7.11b we show the error histograms of the neural net for all ranges. For the range $0 < \chi^2 \leq 87$ it is conspicuous that all entries in the histogram are in the positive range. This is due to the fact that the net tends to predict larger values for χ^2 's in this range than it actually should. The network learns the zero signal χ^2 's much better and tends to predict the zero signal χ^2 for χ^2 's which are actually smaller.

As expected the distribution for the zero signal χ^2 's (in red) has a clear peak at zero. Also the χ^2 's which are in the range $90 < \chi^2 \leq 150$ have a peak at zero, which is nevertheless not so clear like the zero signal peak. For the χ^2 's in the range $150 < \chi^2 \leq 230$ one cannot recognize a peak which is consistent with the large error in this range (see table 7.7). Finally the χ^2 in range $230 < \chi^2 \leq 250$ also have a clear peak at 250 which falls off sharply to the positive side of the histogram because there are many points in the validation set which have $\chi^2 = 250$, but the neural has a maximal predictable χ^2 due to the tanh activation function in the output neuron.

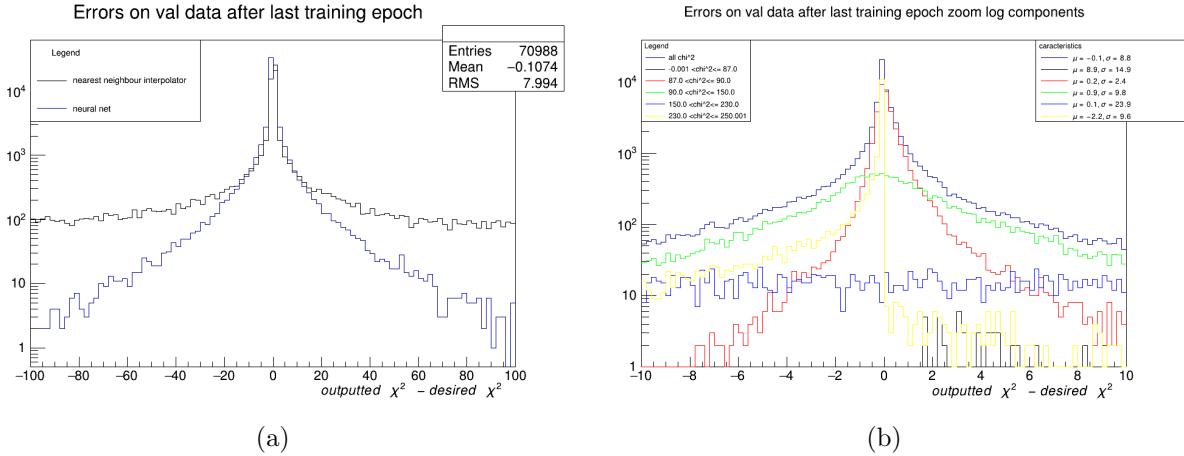


Figure 7.11: (a) shows the error histogram comparison of the nearest neighbour interpolator and the neural net. In (b) one can see the error histogram for the different ranges respectively.

In figure 7.12 one can see that in the transition region (between the zero signal χ^2 and 250) a clear peak is missing, where for the zero signal area and around 250 one can see clear peaks.

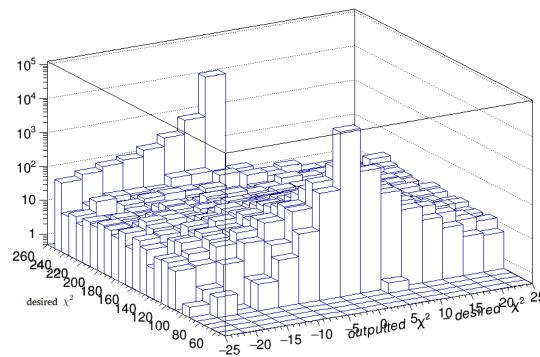


Figure 7.12: On the x axis we show the interpolation error while on the y-axis one can see the correct/desired χ^2 . The bars represent the number of points which fall in the respective histogram range.

7.2.1.2 χ^2_{disjoint}

For χ^2_{disjoint} we have chosen a lower cut as in the case of the χ^2_{tot} because there one has 120 SR which are added and in the case of the χ^2_{disjoint} one has only 47 nonoverlapping SR whose χ^2 s will be added.

In this case we use as basis a neural net with the best hyperparameters from the first hyperparameter scan for χ^2_{tot} (line one of table 7.4). On top of that we carry out a second hyperparameter scan where the same hyperparameters like in the second hyperparameter scan for χ^2_{tot} are varied (see table 7.5). The best two hyperparameters in this case can be found in table 7.8. Again the data set with the extra sampling in the rare target areas is preferred and the artificial extension of the data is not recommended. But this time sequence learning will be used in the best two hyperparameter configurations. The multiplication factor of the number of neurons in the first hidden layer is again 2.0 like in the previous second hyperparameter scan for χ^2_{tot} . The hyperparameters x_2 and x_3 do not show a clear tendency.

extra sampling	artificial extension	seq. learning	x_1	x_2	x_3
yes	no	yes	2.0	2.0	1.5
yes	no	yes	2.0	1.0	1.0

Table 7.8: The two best hyperparameters.

We do not show any result plots in this section, because they look basically the same as the result plots in subsection 7.2.1.1.

The mean errors of points in the validation set in all ranges are given in table 7.9. From the numbers one can see that there are again big differences in the mean error depending on the χ^2 range. The zero signal χ^2 's are in the range $31 < \chi^2 \leq 34$ and although we have applied techniques to avoid the RTLP, the chosen best hyperparameter in table 7.8 says that we should use sequence learning and data which was extra sampled in rare detail areas, it is still not completely solved.

$0 < \chi^2 \leq 200$	$0 < \chi^2 \leq 31$	$31 < \chi^2 \leq 34$	$34 < \chi^2 \leq 120$	$120 < \chi^2 \leq 200$
3.25 (0.049)	5.7 (0.18)	0.81 (0.024)	7.94 (0.13)	8.57 (0.052)

Table 7.9: Mean error on points in the validation set in the respective χ^2 ranges. The numbers in parentheses are mean relative errors. The mean errors in the rare target areas $0 < \chi^2 \leq 31$ and $34 < \chi^2 \leq 120$ is much larger than the error in the zero signal area ($31 < \chi^2 \leq 34$). The last range $120 < \chi^2 \leq 200$ includes a part of the rare detail area but also all χ^2 's which have been cut to 200.

Finally figure 7.13 shows the distribution of all scanned points (id's 1-9 in table 5.2, e.g. also the points which have been sampled in the rare target areas.). As one can see there are still peaks in the zero signal area and at 200. Therefore it is not surprising that the RTLP has not been solved completely.

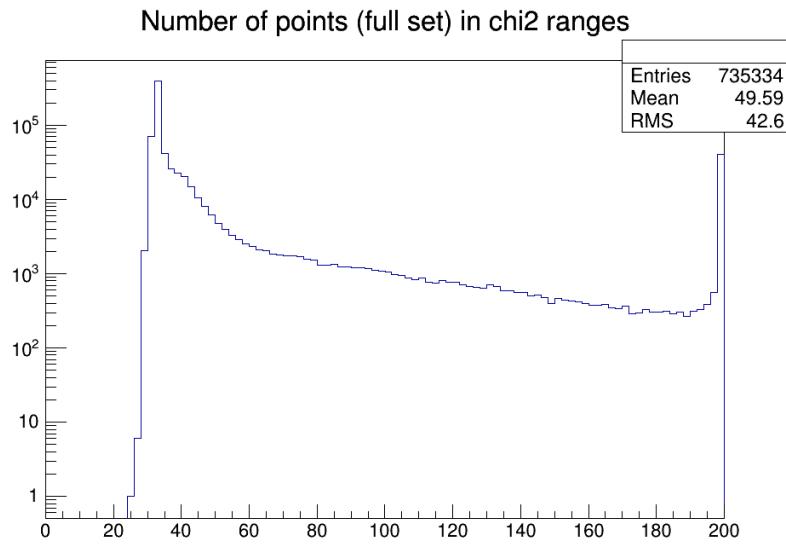


Figure 7.13: Distribution of all χ^2 s with id's 1-9 from table5.2

7.2.2 LHC 13 TeV χ^2 neural net

For 13 TeV we will develop again two neural nets. One network outputs χ_{tot}^2 and the other net χ_{disjoint}^2 . The validation set will include 10000 points in each case.

7.2.2.1 χ_{tot}^2

In figure 7.14 one can see the χ^2 distribution which shows again a peak at the zero signal χ^2 and a peak at 250. In contrast to the 8 TeV case the peaks are much less extreme what means that in the 13 TeV case the parameter space is much more sensitive.

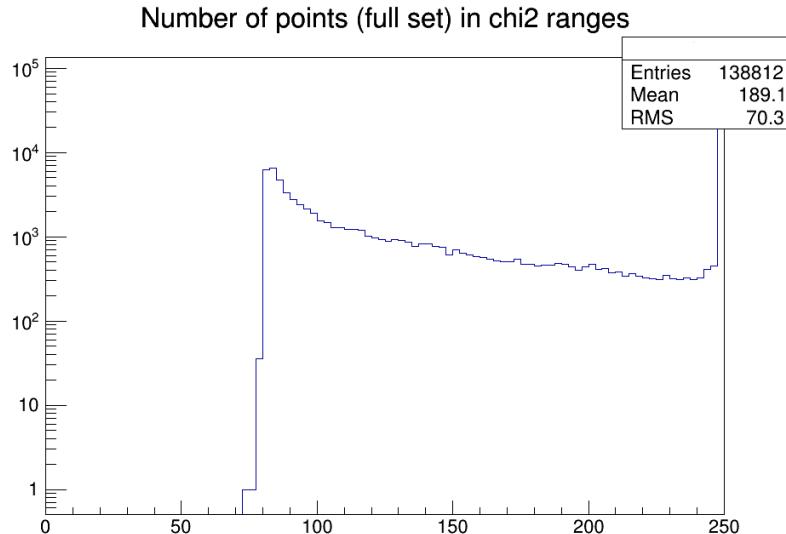


Figure 7.14: Distribution of all χ^2 from the data in table 5.3

Nevertheless we expect that also in this case the RTLP occurs. Therefore we will again choose a neural net with the best hyperparameters from the first 8 TeV χ^2 hyperparameter scan. On this basis we will carry out again a hyperparameter scan. The varied hyperparameters can be seen in table 7.10.

hyperparameter	scanned values
extend data artificially	yes/no
use sequence learning	yes/no
multiply number of neurons in first hidden layer by (x_1)	1.0 , 1.5, 2.0
multiply number of neurons in hidden layer 2,3,4 by (x_2)	1.0, 1.5, 2.0
multiply batch size by (x_3)	1.0, 1.5

Table 7.10: Scanned hyperparameter configurations. The varied hyperparameters agree with the varied hyperparameters from the second scans in the 8 TeV case (see table 7.5, but a hyperparameter which includes data which has been sampled in rare target areas is not present because we simply do not have this data.)

The best found hyperparameters can be found in table 7.11. The message from the scan results is that it is not helpful to extend the data artificially, which was also found in the 8 TeV hyperparameter scans. The statement about sequence learning is not clear, because the best found hyperparameter configuration⁸ does not use sequence learning, but the second and third best do. All in all the RTLP solution strategies are not favored by the best hyperparameter configuration, because the RTLP is not so extreme as in the 8 TeV case (the zero signal peak and the peak at 250 have been much more extreme in the 8 TeV case). Also the enlargement of the batch size is not useful.

artificial extension	seq. learning	x_1	x_2	x_3
no	no	1.5	2.0	1.0
no	yes	1.0	1.5	1.0
no	yes	2.0	2.0	1.0

Table 7.11: The three best hyperparameters. line 1 gives the best configuration and so on.

Table 7.12 gives the errors in the different χ^2 ranges for the neural net with the best found hyperparameter (line 1 in table 7.11). By looking at the errors in table 7.12 one can see that the RTLP does not seem to occur in every rare target area. For example in the area $0 < \chi^2 \leq 81$, but a closer look at the χ^2 distribution 7.14 clarifies this. In the range $0 < \chi^2 \leq 81$ most points lie around 80 (see figure 7.14). But due to the peak at the zero signal χ^2 which is at 81.6 the network learns these targets much better and tends to output the zero signal χ^2 also for parameter points which actually should lead to a χ^2 which lies in the range $0 < \chi^2 \leq 81$. The mean error of the χ^2 s in the range $0 < \chi^2 \leq 81$ will then be typically around 1. And this is exactly the case. Therefore the small mean error in the range $0 < \chi^2 \leq 81$ is not an indication for the fact that the RTLP does not occur in this range. Nevertheless it is not harming too much, because all points in this range are not too far away from the zero signal χ^2 .

⁸Again we have looked at the mean error in all ranges and at the mean error in the whole range to determine the best hyperparameter configurations.

$0 < \chi^2 \leq 250$	$0 < \chi^2 \leq 81$	$81 < \chi^2 \leq 83$	$83 < \chi^2 \leq 90$	$90 < \chi^2 \leq 230$	$230 < \chi^2 \leq 250$
5.7 (0.035)	1.21 (0.015)	1.39 (0.017)	2.5 (0.029)	11.0 (0.076)	3.5 (0.014)

Table 7.12: Mean errors in χ^2 ranges. In parentheses we show the mean relative errors.

7.2.2.2 χ^2_{disjoint}

For χ^2_{disjoint} we have chosen a lower cut as in the case of the χ^2_{tot} because there one has less SRs than in the case of χ^2_{tot} .

We carry out a hyperparameter scan with the hyperparameters shown in table 7.10. This is done again on the basis of the best found hyperparameterconfiguration which was found in the first scan for the 8 TeV χ^2_{tot} (see table 7.4). The best found hyperparameter configurations can be seen in table 7.13. The best hyperparameterconfiguration does not prefer an artificial extension in rare target areas but an extension of the number of neurons in the hidden layers. The batch size stays the same. Furthermore it is found that sequence learning is helpful.

artificial extension	seq. learning	x_1	x_2	x_3
no	yes	2.0	2.0	1.0
no	no	2.0	2.0	1.5
no	no	1.0	1.5	1.0

Table 7.13: The three best hyperparameters. The first line gives the best found hyperparameterconfiguration and so on.

The results corresponding to the best found hyperparameter configuration can be seen in table 7.14. We have a mean error of 4.9 and a mean relative error of 4.6%. The zero signal χ^2 is at 47 and therefore in the range $46.5 < \chi^2 \leq 49$. The RTLP occurs clearly in the ranges $49 < \chi^2 \leq 120$ and $120 < \chi^2 \leq 200$ ⁹, which becomes clear by looking at the χ^2 distribution. By looking at the mean errors in the range $0 < \chi^2 \leq 46.5$ one could conclude that the RTLP does not occur there because the mean error there is equal to the mean error in the zero signal χ^2 range. But this thinking is misleading in this case¹⁰. The RTLP also occurs in the range $0 < \chi^2 \leq 46.5$ which will be clarified by looking at the χ^2 distribution 7.15.

$0 < \chi^2 \leq 200$	$0 < \chi^2 \leq 46.5$	$46.5 < \chi^2 \leq 49$	$49 < \chi^2 \leq 120$	$120 < \chi^2 \leq 200$
4.9 (0.046)	1.53 (0.033)	1.55 (0.032)	5.87 (0.075)	4.94 (0.029)

Table 7.14: Mean errors in χ^2 ranges. In parentheses we show the mean relative errors.

The smallest χ^2 which occurs is around 45 (1.5 away from the zero signal χ^2). Even if we output all χ^2 s in the range with the zero signal χ^2 one ends up with a mean error of 1.5. And surprisingly the error in this range is 1.53 (see table 7.14). Therefore the RTLP also occurs in the range $0 < \chi^2 \leq 46.5$, but it is not that harming as in the 8TeV case for example, because the minimal χ^2 s are very close to the zero signal χ^2 s.

⁹Actually the range $190 < \chi^2 \leq 200$ is not a rare target range (see figure 7.15), but because we have defined the range $120 < \chi^2 \leq 200$ which includes the rare target range $120 < \chi^2 \leq 190$ and the range $190 < \chi^2 \leq 200$ we observe the RTLP also in the range $120 < \chi^2 \leq 200$.

¹⁰Exactly the same situation appeared in the χ^2_{tot} case.

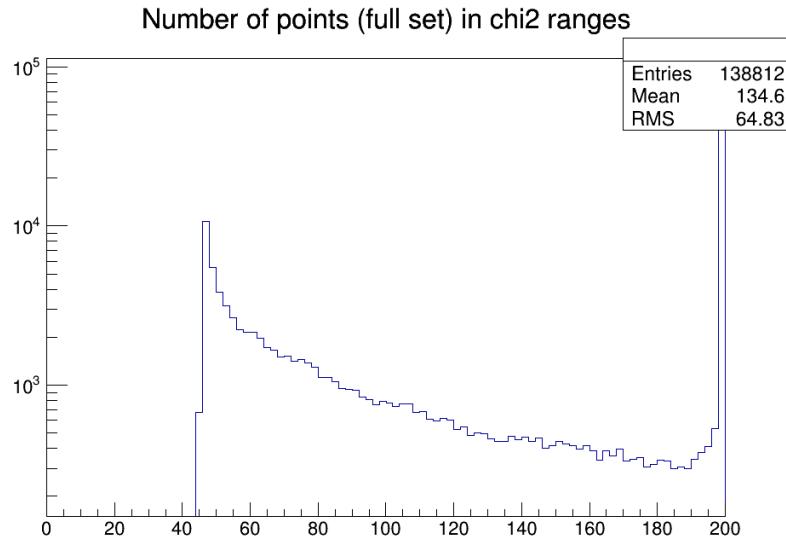


Figure 7.15: Distribution of all χ^2 from table5.3

Figure 7.16a shows the mean error on points in the validation set. The neural net performance is five times better than the performance of a nearest neighbor interpolator. In figure 7.16b one can see the mean errors in the different ranges. As it can also be seen from table 7.14 the mean errors in the ranges $0 < \chi^2 \leq 46.5$ and $46.5 < \chi^2 \leq 49$ are nearly identical.

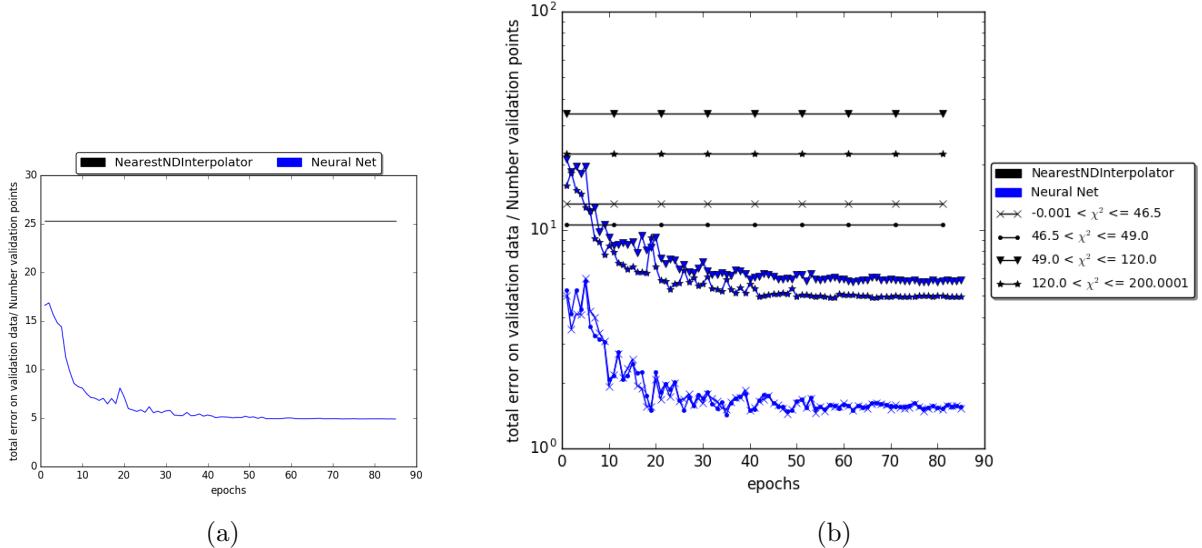


Figure 7.16: (a) shows the mean error for the range $0 < \chi^2 \leq 200$ and (b) shows the mean errors for the different χ^2 ranges.

In figure 7.17a we show again the error histograms for the neural net and the nearest neighbour interpolator. As expected from plots 7.16a and 7.16b the nearest neighbour interpolator has much bigger errors than the neural net. Finally figure 7.17b shows the error histograms for the neural net in the different χ^2 ranges. For the range $0 < \chi^2 \leq 46.5$ one can see that the

error is mostly positive which means that the output of the neural net is mostly larger than the desired values. This supports the before mentioned statement, that the χ^2 values which is actually located in the range $0 < \chi^2 \leq 46.5$ will be mapped by the net to the zero signal value, which is larger. The green and bright blue histograms in figure 7.17b have the largest standard deviations. This is expected because they also have the largest mean errors (see table 7.14).

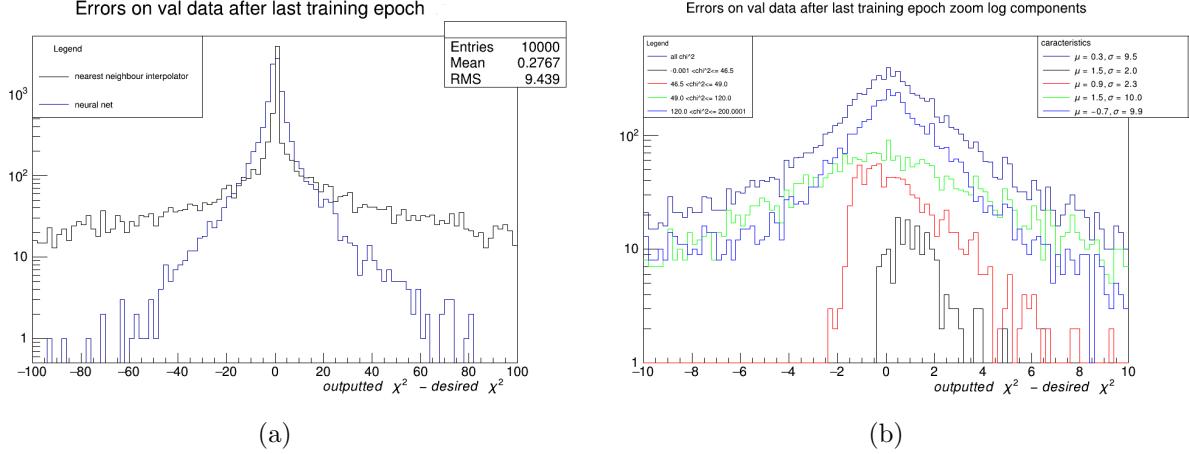


Figure 7.17: (a) shows the error histogram comparison between the nearest neighbour and the neural net. (b) shows the error histogram for the neural net in the different χ^2 ranges.

7.2.3 Neural net for the number of events in one SR

In this subsection we will develop a neural network which predicts the number of events in the SR 'SRA1' of the ATLAS analysis [67]. First we apply the smooth cuts again to the data (see equation 7.2). We cut smoothly at

$$S_{\max} = 1.5 \max(S_{\exp}^{95}, S_{\text{obs}}^{95}) = 57.0.$$

Furthermore for this specific SR we have:

$$N_{\text{obs}} = 102.0, \quad N_{\text{SM}} = 94.0 \pm 13$$

Independently of the previous hyperparameter scan we did a completely new scan because we face a completely new problem here. The scanned hyperparameter configurations can be found in table 7.15. The number of hidden neurons can be 2,3,4 or 5 (see figure 7.18) and all activation functions will be tanh because the first hyperparameter scan for the 8 TeV χ^2_{tot} neural net showed that it is also preferred that the output neuron should have a tanh activation function. We now also vary the number of neurons in the hidden layers independently. This has not been done so far in this extent. The result will reveal the optimal structure of a neural network with one output and 11 inputs.

Also new is that we now test the error quadratic cost. The error quadratic cost is suited for this problem because all signal events have an error. Here we set $\Delta = 2 \times \sigma_{\text{signal}}$, where σ_{signal} is the error on the signal.¹¹

¹¹One could also use the error quadratic cost for the χ^2 nets, although the χ^2 's have no errors. For example one could set $\Delta = 1$ if one uses the error quadratic cost for the χ^2 nets.

We also vary the batch size, regularization parameter and the learning rate parameter¹² in this scan. Nevertheless the scan ranges are strongly constrained due to the previous hyperparameter scans in this thesis.

hyperparameter	scanned values
number hidden layers	2, 3, 4, 5
# neurons hidden layer 1	50, 150, 450
# neurons hidden layer 2	50, 150, 450
# neurons hidden layer 3	50, 150, 450
# neurons hidden layer 4	50, 150, 450
# neurons hidden layer 5	50, 150, 450
cost function	quadratic, cross, clever quadratic
batch size	100, 500, 2000
λ	$10^{-4}, 10^{-5}, 10^{-6}$
learning rate	0.001, 0.0001, 0.00001

Table 7.15: Scanned hyperparameters

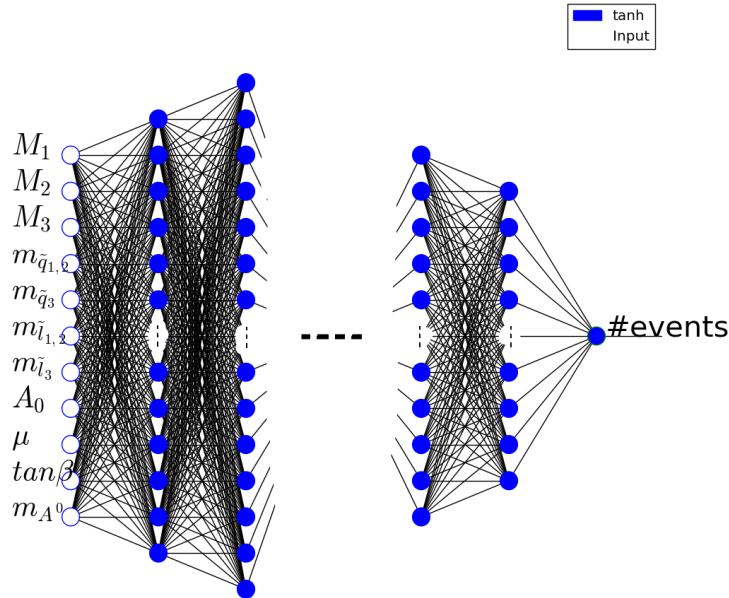


Figure 7.18: Network structure during the hyperparameter scan.

The best found hyperparameter configurations are listed in table 7.16. We list this time the

¹²The other hyperparameter of the Adam optimizer are set to their default values

17 best hyperparameter configurations, because we want to get a feeling of how to choose the best network structure.

hidden layers	cost	batch size	λ	learning	neurons in hidden layers rate
5	error quadratic	500	1e-05	0.001	450 450 150 50 150
5	error quadratic	100	1e-06	0.001	150 150 50 50 50
5	error quadratic	500	1e-05	0.001	450 450 150 450 150
5	error quadratic	500	1e-06	0.001	450 150 450 150 150
5	error quadratic	500	1e-06	0.001	450 450 450 450 150
4	error quadratic	500	1e-06	0.001	150 150 450 50
4	error quadratic	100	1e-05	0.001	450 450 150 50
5	error quadratic	100	1e-06	0.001	450 450 450 450 150
5	error quadratic	100	1e-06	0.001	50 450 450 50 50
5	error quadratic	100	1e-06	0.001	150 450 150 150 50
5	error quadratic	500	1e-05	0.001	50 150 150 150 150
5	error quadratic	500	1e-05	0.0001	150 450 150 450 50
5	error quadratic	500	1e-05	0.001	450 450 450 150 50
5	error quadratic	500	1e-06	0.001	450 450 450 150 150
4	error quadratic	500	1e-06	0.001	450 450 450 150
5	error quadratic	100	1e-06	0.0001	450 450 450 150 50
5	error quadratic	100	1e-06	0.001	150 150 450 50 150

Table 7.16: 17 best hyperparameter configurations. The best configuration can be found in line one and so on. The criterion to select the best hyperparameters is the mean error in all ranges averaged over the last 10 learning epochs.

First of all one can see in table 7.16 that the best network has 5 hidden layers and was trained with the error quadratic cost function. Furthermore during the training phase a batch size between 100 and 500 is recommended. The regularization parameter should be between 10^{-5} and 10^{-6} and the learning rate is preferred to be 0.001.

Interesting are the optimal hyperparameters for the number of neurons in the hidden layers. Summarized the result is that one should construct a neural net where one has many neurons in the first hidden layer such that the information which comes from the inputs can be distributed widely. The number of neurons in the last and maybe also in the second last layer should be smaller than the number of neurons in the first layer such that the information which was distributed at the beginning can be processed and brought down to one information which will be outputted in the end. We summarize these observations in the rule of thumbs for building a neural network in section 7.3.

The results corresponding to the best found hyperparameter configuration is presented in the following. Figure 7.19a shows the mean error which goes down to 0.6. Form this perspective one could conclude that the network does a very good job and predicts all signal events with a mean error of 0.6. But if one looks at figure 7.19b one can see that only the mean error in the range between 0 and 1 is small. The errors for number of events which are actually between 15 and 57 is over 20. This behaviour can again be understood if we check how the number of events are distributed in the 11 dimensional parameter space. Approximately 92 % of all points which have been sampled in the 11 dimensional parameter space yield number of events which

are between 0 and 1. Therefore the net learns the targets which lie in the range (0,1) much better. The pMSSM-11 parameter points which should lead to number of events larger than one will often have neural net output in the range (0,1) what leads to the large errors in the ranges (1,15) and (15,57).

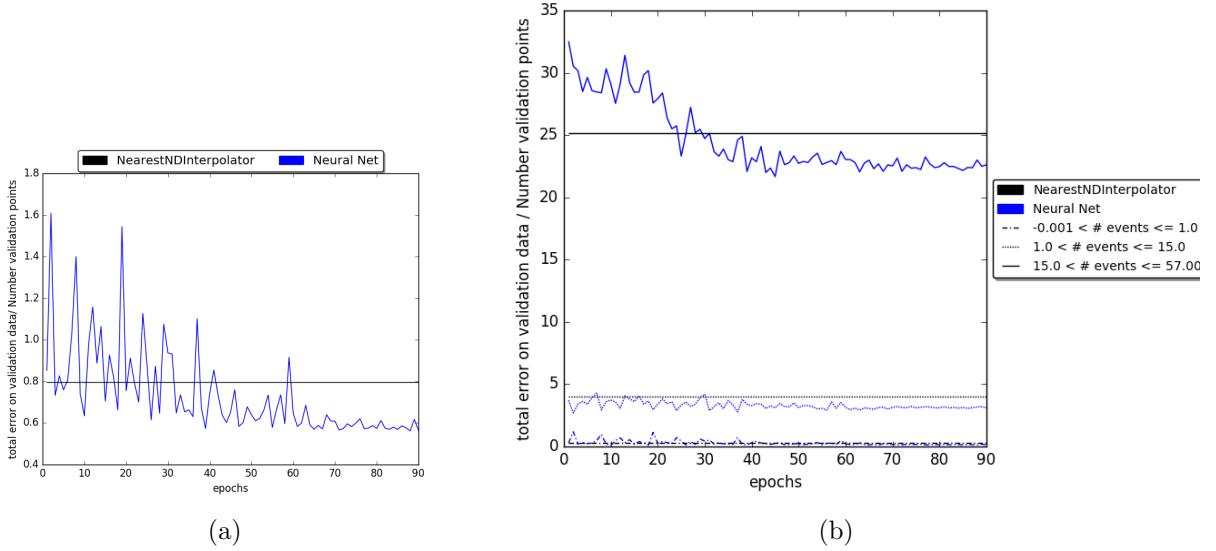


Figure 7.19: (a) shows the error histogram comparison between the nearest neighbour and the neural net. (b) shows the error histogram for the neural net in the different χ^2 ranges.

All in all one can summarize that the development of the neural net which should predict the number of events in a SR was not successful because the RTLP also occurs here but in a much stronger form than for the χ^2 neural networks.

However, the results from the hyperparameter scan that tell us how to construct the neural network are still valid, because even the best network would have the RTLP for this dataset. Therefore we expect that a neural network with the best found hyperparameter configuration will do a very good job on a dataset which has nor rare target areas.

Nevertheless, we also tried to develop a neural network with many outputs so that one predicts for all SR the number of signal events at once. This has not been successful as one could have already guessed from the previous results.

7.2.4 χ^2 s for pMSSM-19 from pMSSM-11 neural net

For this section we have generated 10000 pMSSM-19 χ^2_{tot} 's at 13 TeV with the described simulation chain (see chapter 5). We now try to predict these χ^2 s using the pMSSM-11 neural net from section 7.2.2.1. The pMSSM-19 parameters are reduced in the following way to pMSSM-11 parameters:

$$m_{\tilde{Q}_{12}} = \frac{m_{\tilde{Q}_{L,12}} + m_{\tilde{u}_{R,12}} + m_{\tilde{d}_{R,12}}}{3} \quad (7.4)$$

$$m_{\tilde{Q}_3} = \frac{m_{\tilde{Q}_{L,3}} + m_{\tilde{u}_{R,3}} + m_{\tilde{d}_{R,3}}}{3} \quad (7.5)$$

$$m_{\tilde{L}_{12}} = \frac{m_{\tilde{L}_{L,12}} + m_{\tilde{l}_{R,12}}}{2} \quad (7.6)$$

$$m_{\tilde{L}_3} = \frac{m_{\tilde{L}_{L,3}} + m_{\tilde{l}_{R,3}}}{2} \quad (7.7)$$

$$A_0 = \frac{A_t + A_b + A_\tau}{3} \quad (7.8)$$

In figure 7.20 one can see the mean errors in all chosen ranges. The ranges with $0 < \chi^2 \leq 81$ and $230 < \chi^2 \leq 250$ have a mean error of 3. χ^2 s which belong to the $81 < \chi^1 \leq 83$ range have a mean error of 8. Critical are the χ^2 s in ranges $83 < \chi^1 \leq 90$ and $90 < \chi^2 \leq 230$ with a mean error of 18 and 50 respectively.

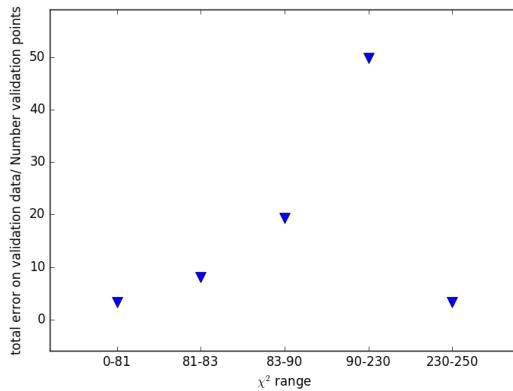


Figure 7.20: Mean errors in the χ^2 ranges.

The main message from this experiment is that the pMSSM-11 neural net can be used to give a rough feeling if the pMSSM-19 parameter point has a very large χ^2 or not. For precision studies where the χ^2 should be known very well the network should not be used.

7.3 Rule of thumbs for neural networks with one output

From all hyperparameter scans rule of thumbs of how to construct a network and choose its hyperparameters will be deduced.

- The learning rate for the Adam optimizer should be between 0.01 and 0.001. The other hyperparameters which are introduced by the Adam optimizer should be set to their default Tensorflow values.
- We try to give a relation between the batch size (b) and the size of the training set t .

$$b = t \times x \quad (7.9)$$

In the cMSSM scan we had 45000 training points and found an optimal batch size of $b = 30 - 80$. Therefore x lies approximately in the range 0.0006-0.002. For the pMSSM-11 χ^2 hyperparameter scans we get an optimal batch size between 500 and 750. With approximately 600000 training points. Therefore x should be between 0.0008 and 0.0013.

The pMSSM #events hyperparameter scan yield an optimal batch size between 100 and 500. Therefore we get with approximately 600000 training points a x in the ranges 0.0002 and 0.0008.

Combining all results we deduce that $x \in (0.0001, 0.002)$ should be a good choice.

- From all hyperparameter scans we conclude that $\lambda < 10^{-5}$ is an appropriate choice.
- The new designed error quadratic cost function is better than the quadratic and cross entropy cost functions¹³.
- For a network with many inputs and only one output one should start with many neurons in the first layer to distribute the information at the beginning widely. For a network with 11 inputs 100-500 neurons in the first layer are recommended and for a network with two inputs 30-80 neurons. Therefore as a rule of thumb one can say that the first hidden layer should contain approximately 10 to 50 times more neurons than there are input neurons. Going from the first hidden layer to the hidden layer which is in the middle of the network the number of neurons in the layers should stay equal or even increase. Beginning from the middle of the network the number of the neurons in the hidden layers can decrease or stay equal. But the number of neurons in the last hidden layer should be smaller than the number of neurons in the first hidden layer. Figures 7.21a - 7.21c show preferred network structures. In figure 7.21d one can see a network structure which should not be used.
- If a RTLP occurs it is not helpful to extend the data in rare areas artificially, but sequence learning can be helpful. On top of that it is always helpful to sample new data in rare target areas.

7.4 Outlook

In this section we give an outlook how one could improve the given results and what else can be done with the existing data

- We saw in all cases that the performance of the neural net can be very different in different target ranges. This was mainly due to the fact that the target values are not equally distributed. Some target values appear very often, others are very rare. A neural net which tries to cover the whole target range will therefore learn the targets in the rare target areas not very well. This problem was called rare target learning problem (RTLP) in this thesis. We expect that it would help if we trained a neural network for each range separately. A classifier then can then at the beginning select for a given parameter point the correct neural net.
- With the existing data one can train neural nets which have an allowed/excluded output¹⁴.
- One can interpolate a function which maps the parameters of the spectrum file¹⁵ on the pMSSM-11 parameters. With this function one can make χ^2 predictions for example for the cMSSM with an existing pMSSM-11 neural network.
- One could try if the error quadratic cost improves the χ^2 nets.

¹³Of course this was only tested for the #events hyperparameter scan and should also be tested for the χ^2 neural nets in the future.

¹⁴Checkmate provides the information if a scanned point is allowed or excluded.

¹⁵Particle masses

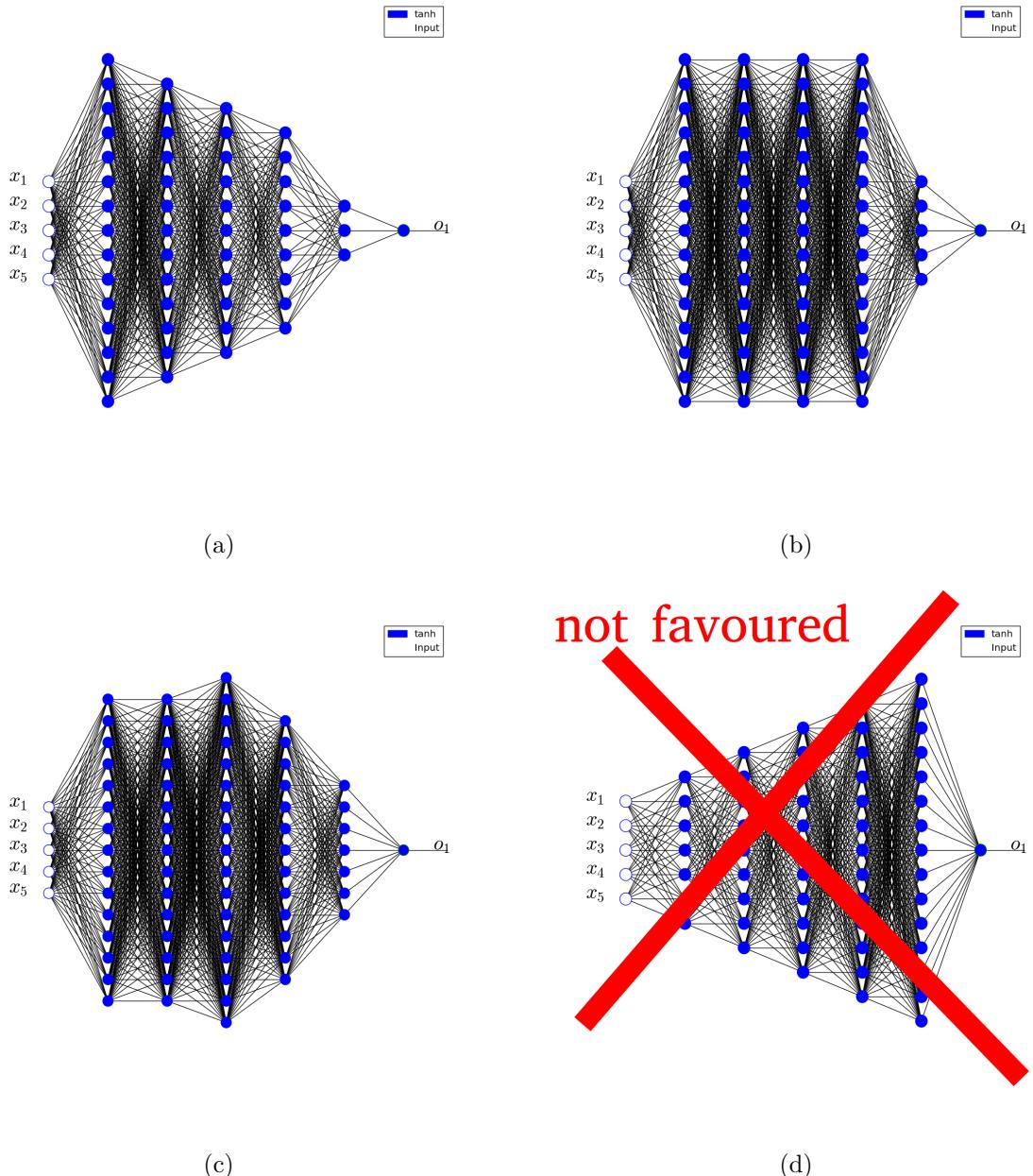


Figure 7.21: (a)-(c) shows examples for allowed network structures with 5 input neurons and one output. In (d) one can see a network structure which is not favored.

8 Global fit results

As it was already pointed out the main task of this master thesis is the development of the neural networks which predict the LHC χ^2_{disjoint} for 8 TeV and 13 TeV respectively. Nevertheless in this section the first preliminary global fit results which include the LHC χ^2 from the neural networks are presented. The global fit was done by Matthias Hamer with the Fittino framework. The best fit point from the fit with all observables which have been mentioned in chapter 4 is in table 8.1. But there are two reasons why the best fit point is very preliminary and will probably change in the future. First we have scanned only 2.6 million points yet and we expect that we have to scan many more points. Second the LHC χ^2 neural networks expect a pMSSM-11 parameter point where the parameters are defined at 1 TeV. But in the global scan the χ^2 for all other observables has been calculated by using pMSSM-11 parameters which are defined at 1.5 TeV. This inconsistency has to be resolved in the future and may lead to a different best fit point.

M_1	M_2	M_3	$m_{\tilde{Q}_{12}}$	$m_{\tilde{Q}_3}$	$m_{\tilde{L}_{12}}$	$m_{\tilde{L}_3}$	m_{A^0}	A_0	μ	$\tan \beta$
-196.1	3739.4	2396.4	2512.2	5082.9	360.5	708.7	3145.1	274.5	-4163.4	15.4

Table 8.1: Preliminary best fit point for the pMSSM-11. All quantities are given in GeV (except for $\tan \beta$).

Interesting about the best fit point is that the $m_{\tilde{L}_{12}}$ parameter is relatively low. $m_{\tilde{L}_{12}}$ is the tree level mass of the sleptons of the first two families. The low $m_{\tilde{L}_{12}}$ parameter therefore leads to low smuon masses which are needed to bring the predictions for $a_\mu - a_\mu^{SM}$ closer to the experimental value. In table 8.3 we list the predicted value for $a_\mu - a_\mu^{SM}$ with the best fit point. And indeed the predicted value of the best fit point for $a_\mu - a_\mu^{SM}$ differs from the experiment by 0.87 standard deviations. The SM prediction deviates 3.6 standard deviations from the experimental value.

Furthermore the parameters M_3 , $m_{\tilde{Q}_{12}}$ and $m_{\tilde{Q}_3}$, which determine the masses of the gluino and squarks at tree level, are very large. This is an artifact of the LHC observables which exclude gluinos and squarks with low masses, because their cross sections would be very large leading to much more events in the SUSY searches which focus on strong production processes. One can see this feature by looking at figure 8.1a and 8.1b. The χ^2 contribution from the LHC observables will blow up for small M_3 and $m_{\tilde{Q}_{12}}$ respectively. A detailed discussion of the plots 8.1a and 8.1b follows.

The SUSY particle spectrum corresponding to the best fit point is shown in table 8.2. As already mentioned we expect masses of approximately 2.5 TeV for squarks of the first two families and masses of approximately 5 TeV for squarks of the third family. The slepton masses are lower due to the low $m_{\tilde{L}_{12}}$, $m_{\tilde{L}_3}$ parameters. We observe a Higgs boson with mass 126(124) GeV from SPheno(FeynHiggs). The mass of the lightest neutralino which is the dark matter candidate is around 195 GeV.

mass in GeV	particle	mass in GeV	particle	mass in GeV	particle
2557	\tilde{d}_L	217	$\tilde{e}_L -$	126	h^0
2559	\tilde{d}_R	358	$\tilde{e}_R -$	3145	H^0
2556	\tilde{u}_L	203	$\tilde{\nu}_{e,L}$	3145	A^0
2560	\tilde{u}_R	216	$\tilde{\mu}_L$	3146	H^+
2557	\tilde{s}_L	359	$\tilde{\mu}_R$	195	$\tilde{\chi}_1^0$
2559	\tilde{s}_R	203	$\tilde{\nu}_{\mu,L}$	3738	$\tilde{\chi}_2^0$
2556	\tilde{c}_L	569	$\tilde{\tau}_1$	4202	$\tilde{\chi}_3^0$
2560	\tilde{c}_R	748	$\tilde{\tau}_2$	4207	$\tilde{\chi}_4^0$
5105	\tilde{b}_1	637	$\tilde{\nu}_{\tau,L}$	3738	$\tilde{\chi}_1^+$
5142	\tilde{b}_2			4207	$\tilde{\chi}_2^+$
5135	\tilde{t}_1				
5164	\tilde{t}_2				
2587	\tilde{g}				

Table 8.2: MSSM particle mass spectrum for best fit point. The input parameters for this spectrum are defined at 1.5 TeV. The h^0 mass obtained with FeynHiggs is 124.2 GeV.

Table 8.3 shows the predictions for all observables (except for the observables for HiggsSignals) at the best fit point. None of the observables has a bigger deviation than 1.3 sigma from the experimental measurements.

observable	predicted value	deviation in σ
$a_\mu - a_\mu^{SM}$	21.61×10^{-10}	0.87
$\sin^2 \Theta_{\text{eff}}$	0.2314427	1.3
m_t	174.69 GeV	1.1
m_W	80.375 GeV	0.58
Δm_s	21.469 ps $^{-1}$	0.73
$\mathcal{B}(B_s \rightarrow \mu\mu)$	3.549×10^{-9}	0.56
$\mathcal{B}(b \rightarrow s\gamma)$	3.192×10^{-4}	0.48
$\mathcal{B}(B_u \rightarrow \tau\nu)$	0.8002×10^{-4}	0.84
Ωh^2	0.1120	0.6

Table 8.3: Shows the observables and their correspondig predicted values at the best fit point. Furthermore the deviation to the experimental measured values is given.

Figures 8.1a and 8.1b show the LHC χ^2 for 8 TeV and 13 TeV which have been obtained by the neural networks. The plots are projections on the M_3 - $m_{\tilde{Q}_{12}}$ plane, which show all scanned points of the global fit. In the 8 and 13 TeV case we see that in the case of small M_3 masses and small $m_{\tilde{Q}_{12}}$ mass the LHC χ^2 is very high. This is expected because the parameter M_3 determines the tree level gluino mass and $m_{\tilde{Q}_{12}}$ the tree level squark mass of the first two families. The cross section for gluino and squark production will blow up for small gluino and squark masses. Therefore we expect high exclusion for small M_3 and $m_{\tilde{Q}_{12}}$. This is confirmed by the two plots 8.1a, 8.1b. The sensitivity of the 13 TeV analyses is larger than the sensitivity of the 8 TeV analyses. This feature can be seen in the plots because the high χ^2 areas in 8.1b is larger than in 8.1a.

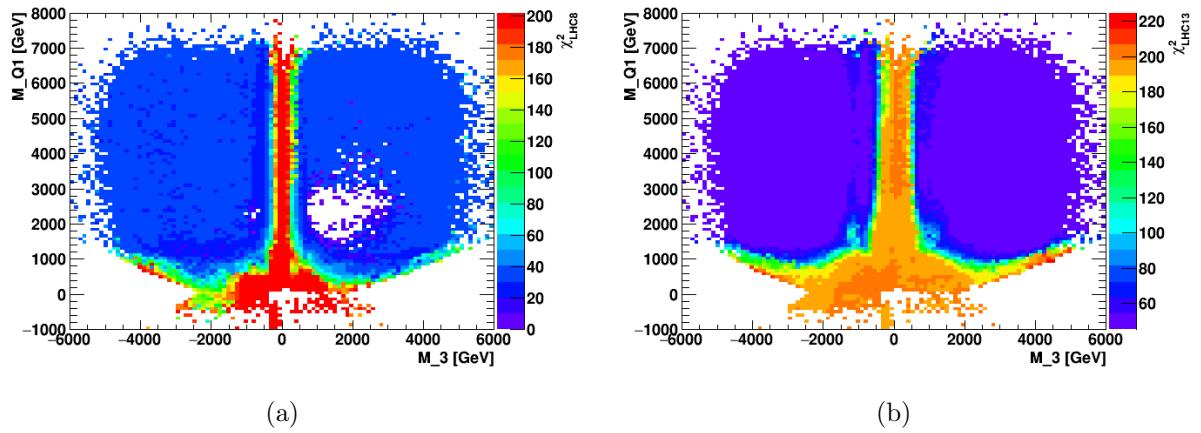


Figure 8.1: (a) shows the LHC χ^2 predictions of the 8 TeV neural network and (b) shows the 13 TeV neural network LHC χ^2 predictions.

9 Conclusion

Supersymmetry is the most popular studied theory beyond the Standard Model (SM) framework. Many models which are based on Supersymmetry have been proposed. In this thesis the supersymmetric model is the phenomenological minimal supersymmetric extension of the SM with 11 parameters (pMSSM-11). The main task is to check if the pMSSM-11 is a valid candidate for a fundamental model which is realized in nature. Two investigation steps will be needed. First we find the best pMSSM-11 parameter point (best pMSSM-11 model) by comparing the theoretical predictions to the experimental measurements. The best pMSSM-11 parameter point also called best fit point has the lowest χ^2 , which is the quantity we use to specify if the theoretical predictions coincide well or bad with the experimental results. Second a p-value has to be calculated, which reveals if the pMSSM-11 can explain all measurements consistently. In this thesis work on the first investigation step has been done.

In the comparison with the experimental measurements one includes electroweak precision observables, Higgs observables, cosmological observations and LHC observables, which are basically the number of events in LHC SRs. It turns out that the theoretical predictions for all observables can be done in order of seconds, except for the LHC observables, because one has to generate LHC events, what is a time consuming task (order of hours). Therefore scanning many points in the pMSSM-11 parameter space becomes technically problematic even on the largest computer cluster.

There are basically two options to bring the predictions for the LHC observables down to order of seconds. First one can modify the event generation, but this in contrast makes the results less accurate and may only push everything to order of minutes. The second solution strategy is the development of an interpolation tool which interpolates the unknown function from the 11 dimensional parameter space to the one dimensional space of LHC χ^2 / number of events in a LHC SR. If we develop a tool which predicts the number of events in one LHC SR, then the development of tools for all other SRs is straight forward. The LHC χ^2 can be calculated from the number of events in order of milliseconds.

In this thesis we decided to use the interpolation approach, which is extremely quick (order of milliseconds for one pMSSM-11 parameter point). We compared a nearest neighbour interpolator with a neural network interpolation. The power of the interpolation can be specified if one tests the interpolation method with pMSSM-11 'validation' points for which one knows the correct LHC χ^2 / number of events in the SR. A quantity which specifies the power of the interpolation method is the mean error on points in the validation set.

To train the neural networks we generated events at 8 and 13 TeV respectively. The software framework Checkmate was used to get the LHC χ^2 / number of events in SRs of 8 TeV and 13 TeV analyses. For the 8/13 TeV case we have generated for approximately 700000/140000 pMSSM-11 parameter points events.

A main result of this thesis is that the neural network interpolation method gives better results in all problems than the nearest neighbour interpolator. In the case of the χ^2 the mean error of the neural network interpolator is approximately 5 times smaller than the mean error with the nearest neighbour interpolator. The mean error for the neural network interpolator is 3.25 for 8 TeV χ^2 s and 4.9 for 13 TeV χ^2 s.

It is much more difficult to develop neural networks which predict the number of events in a LHC SR, because there are some SRs which have zero signals for many parameter points and in conclusion the neural networks learns the pMSSM-11 points which yield zero signal much better than the other points which do not yield zero signals. Therefore another main result is that it is much more comfortable to train neural networks which give directly the LHC χ^2 instead of training several neural networks which predict the number of events in several LHC SRs.

The developed neural networks which predict the LHC χ^2 for 8 TeV and 13 TeV respectively are incorporated into the Fittino framework. Furthermore the Fittino framework binds up all necessary tools which give the predictions for all other observables. In the end Fittino combines everything to an overall χ^2 . The pMSSM-11 parameter space is scanned by Fittino with a higher order Markov-Chain-Monte-Carlo technique to find the 11 dimensional best fit point which gives the smallest χ^2 . The first preliminary result for the best fit point is presented in this thesis. The best fit point has smuon masses around 200 GeV, what leads to a good agreement of the $(g-2)_\mu$ prediction with the experimental value. Squark and gluino masses are found to be above 2.5 TeV for the best fit point. The largest deviation from experiment and theory prediction with the best fit point for all electroweak observables and the relic dark matter density is 1.3 standard deviations.

During the thesis a lot of work on the optimal configuration of the neural networks has been done, which is summarized in the rule of thumbs for the best neural network configuration. The rule of thumbs can be used in further studies which want to use neural networks that target similar tasks.

Furthermore we tried to predict pMSSM-19 LHC χ^2 s with the pMSSM-11 neural networks, by reducing the 19 parameters with a convenient transformation to 11 parameters. The result from this study is that the pMSSM-11 neural network can be used to give a rough classification if a pMSSM-19 point is highly excluded or not.

All in all the results in this thesis show that LHC observables can also be incorporated into scans of high dimensional models with the help of neural networks. The nearest neighbour interpolation method is not suited to predict reliable LHC χ^2 s for a high dimensional parameter space with a rare point density.

10 Acknowledgement

First of all I thank my family for the big support during the last year. I also want to thank Michael Krämer, Jamie Tattersall and Jan Heisig for the helpful discussions about the physical side of my work. Furthermore I thank Philip Bechtle for the weekly discussions in the Fittino meetings. A big thanks goes to Mathias Hamer who did the first preliminary global fit. I also thank Björn Sarrazin who always took time and discussed a lot with me. In addition I thank Jory Sonneveld who presented the latest results on the SUSY conference in Melbourne. Christian Schwinn gave me some very helpful advices concerning the SM and SUSY theory part of this thesis. I like to thank Tim Keller for the helpful discussions about the event generation. Last but not least I thank Thorben Quast for helpful discussions about neural networks.

Appendix

A.1 Conventions

In this thesis vectors will be marked with bold symbols e.g. $\mathbf{v} \in \mathbb{R}^x$, $x > 1$.

Also matrices will be labeled with a bold symbol $\mathbf{m} \in \mathbb{R}^{x \times y}$, $x, y > 1$.

A matrix vector product will be labeled with a dot $\mathbf{m} \cdot \mathbf{v}$.

A product of two matrices will have no dot in between $\mathbf{m}\mathbf{n}$, $\mathbf{m} \in \mathbb{R}^{x_1 \times y_1}$, $\mathbf{n} \in \mathbb{R}^{x_2 \times y_2}$

A.2 Calculation of the profile likelihood ratio for one SR

In this section we calculate for one SR the profile likelihood ratio (PLR). In the end we will see that from this we can construct a variable which is χ^2 distributed and therefore can be interpreted as the χ^2 for the SR.

The following numbers are available

- Number of signal events in the SR N_S ,
statistical and systematical error on signal events $\sigma_{N_S}^{sys}$, $\sigma_{N_S}^{stat}$,
- number of the SM background events in SR N_{SM} ,
statistical and systematical error on SM events in SR $\sigma_{N_{SM}}^{sys}$, $\sigma_{N_{SM}}^{stat}$,
- number of measured events in the experiment in the SR N_E ,
statistical error on this number $\sigma_{N_E}^{stat}$

At first one constructs the Likelihood funktion

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

The first term is a Poisson distribution and the other two terms are gaussian probability distributions for the nuisance parameters. There is a nuisance parameter for the signal and background respectively. The nuisance parameters also appear in:

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

In formula (A.2) the signal S is still a free parameter. The motivation behind the exponential functions in (A.2) can be seen if we Taylor expand the parts in the formula:

$$S e^{\frac{\Delta S}{S} \nu_S} = S + \Delta S \nu_S + O((\frac{\Delta S}{S})^2) \quad (\text{A.3})$$

with $\Delta S = \sqrt{(\sigma_S^{stat})^2 + (\sigma_S^{sys})^2}$ and $\Delta N_{SM} = \sqrt{(\sigma_{N_{SM}}^{stat})^2 + (\sigma_{N_{SM}}^{sys})^2}$. With the expansion (A.3) it becomes clear that the nuisance parameter describes the deviation from the signal. With this in mind the normal distributions of the nuisance parameters in the Likelihood function also

make sense.

The null hypothesis is now

$$H_0 : \text{number of signal events } S = N_S \text{ coincides well with the experiment} (\mu = 1) \quad (\text{A.4})$$

and the alternative hypothesis

$$H_1 : \text{number of signal events } S \neq N_S \text{ coincides well with the experiment} (\mu \neq 1) \quad (\text{A.5})$$

For the null hypothesis one gets the constrained Likelihood

$$\mathcal{L}_C := \max_{\nu_S, \nu_{SM} \in \mathbb{R}} \mathcal{L}(\mu = 1, S = N_S, \nu_{SM}, \nu_S) \quad (\text{A.6})$$

where one sets $\mu = 1$ before maximizing. Furthermore for the alternative hypothesis one defines the global likelihood

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

Here one varies also over μ and negative μ are allowed. In the global Likelihood we set also $S = 1$. This is perfectly fine, because we vary over μ . The reason why we nevertheless fix S is that if S would be zero a variation over μ would be meaningless, but we want a 'signalvariation' in the global likelihood also if the signal is zero.

The profile Likelihood is defined as¹:

$$PLR := \frac{\mathcal{L}_C}{\mathcal{L}_G} \quad (\text{A.8})$$

Wilks theorem[96] says that enough large λ

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

is χ^2 distributed. In this thesis we always assume that this is the case and therefore when we talk about the χ^2 for a SR in this thesis we actually mean the negative logarithm of the profile likelihood ratio.

A.3 Groups of disjoint SRs

For each analysis we give the disjoint groups. Each SR of one group is disjoint to all SR of all other groups. The groups are named G1, G2, ... fore each analysis respectively.

8 TeV:

¹The Neyman Pearson Lemma [95] says that the profile likelihood ratio test is the most powerful hypothesis test. We are testing the hypothesis $\mu = 1$ against the hypothesis $\mu \neq 1$.

analysis	SR	group	analysis	SR	group
[67]	SRA1	G1	[68]	H160	G1
[67]	SRA2	G1	[68]	L100	-
[67]	SRA3	G1	[68]	L110	-
[67]	SRA4	G1	[68]	L120	-
[67]	SRA5	G1	[68]	L90	-
[67]	SRB	G1	[68]	SR1	G2
			[68]	SR2	G3
			[68]	SR3	G4
			[68]	SR4	G5
			[68]	SR5	G6
			[68]	SR6	G7
			[68]	SR7	G8

Table A.1: Disjoint groups

analysis	SR	group	analysis	SR	group
[65]	WWa_DF	G1	[70]	SR01_a.2jl	G1
[65]	WWa_SF	G2	[70]	SR01_b.2jm	G1
[65]	WWb_DF	G1	[70]	SR01_c.2jt	G1
[65]	WWb_SF	G2	[70]	SR01_d.2jW	G1
[65]	WWc_DF	G1	[70]	SR02_3j	G1
[65]	WWc_SF	G2	[70]	SR03_a.4jl-	G1
[65]	Zjets	G3	[70]	SR03_b.4jl	G1
[65]	mT2_120_DF	G1	[70]	SR03_c.4jm	G1
[65]	mT2_120_SF	G2	[70]	SR03_d.4jt	G1
[65]	mT2_150_DF	G1	[70]	SR03_e.4jW	G1
			[70]	SR04_5j	G1
			[70]	SR05_a.6jl	G1
			[70]	SR05_b.6jm	G1
			[70]	SR05_c.6jt	G1
			[70]	SR05_d.6jt+	G1

Table A.2: Disjoint groups

analysis	SR	group
[72]	bCa_low	G1
[72]	bCa_med	G1
[72]	bCb_high	G1
[72]	bCb_med1	G1
[72]	bCb_med2_a	G1
[72]	bCb_med2_b	G1
[72]	bCb_med2_c	G1
[72]	bCb_med2_d	G1
[72]	bCc_diag	G1
[72]	bCd_bulk_a	G1
[72]	bCd_bulk_b	G1
[72]	bCd_bulk_c	G1
[72]	bCd_bulk_d	G1
[72]	bCd_high1	G1
[72]	bCd_high2	G1
[72]	tN_boost	G1
[72]	tN_diag_a	G1
[72]	tN_diag_b	G1
[72]	tN_diag_c	G1
[72]	tN_diag_d	G1
[72]	tN_high	G1
[72]	tN_med	G1
[72]	tNbC_mix	G1
[72]	threeBody_a	G1
[72]	threeBody_b	G1
[72]	threeBody_c	G1
[72]	threeBody_d	G1

analysis	SR	group
[64]	SR0tauua01	G1
[64]	SR0tauua02	G2
[64]	SR0tauua03	G3
[64]	SR0tauua04	G4
[64]	SR0tauua05	G5
[64]	SR0tauua06	G6
[64]	SR0tauua07	G7
[64]	SR0tauua08	G8
[64]	SR0tauua09	G9
[64]	SR0tauua10	G10
[64]	SR0tauua11	G11
[64]	SR0tauua12	G12
[64]	SR0tauua13	G13
[64]	SR0tauua14	G14
[64]	SR0tauua15	G15
[64]	SR0tauua16	G16
[64]	SR0tauua17	G17
[64]	SR0tauua18	G18
[64]	SR0tauua19	G19
[64]	SR0tauua20	G20
[64]	SR0taub	G21
[64]	SR1tau	G22
[64]	SR2tauua	G23
[64]	SR2taub	G23

Table A.3: Disjoint groups

analysis	SR	group
[73]	M1	G1
[73]	M2	G1
[73]	M3	G1

analysis	SR	group	analysis	SR	group
			[74]	SR1	G1
[69]	SR0b	G1	[74]	SR2	G1
[69]	SR1b	G2	[74]	SR3	G1
[69]	SR3Lhigh	G3	[74]	SR4	G1
[69]	SR3Llow	G4	[74]	SR5	G1
[69]	SR3b	G5	[74]	SR6	G1
			[74]	SR7	G1
			[74]	SR8	G1
			[74]	SR9	G1

Table A.4: Disjoint groups

analysis	SR	group	analysis	SR	group
[66]	SR0Z	G1	[66]	SR0noZa	G1
[66]	SR0noZb	G2	[66]	SR1Z	G1
[66]	SR1noZ	G2			

Table A.5: Disjoint groups

13 TeV:

analysis	SR	group
[78]	SR0b3j	g1
[78]	SR0b5j	g1
[78]	SR1b	g2
[78]	SR3b	g2

analysis	SR	group
[79]	2jl	g1
[79]	2jm	g1
[79]	2jt	g1
[79]	4jt	g1
[79]	5j	g1
[79]	6jm	g1
[79]	6jt	g1

analysis	SR	group
[84]	SR-Gbb-A	-
[84]	SR-Gbb-B	-
[84]	SR-Gbb-C	-
[84]	SR-Gtt-0L-A	g1
[84]	SR-Gtt-0L-B	g1
[84]	SR-Gtt-0L-C	g1
[84]	SR-Gtt-1L-A	g2
[84]	SR-Gtt-1L-B	g2

Table A.6: Disjoint groups

analysis	SR	group
[81]	EM1	g1
[81]	EM2	g2
[81]	EM3	g3
[81]	EM4	g4
[81]	EM5	g5
[81]	EM6	g6
[81]	IM1	-
[81]	IM2	-
[81]	IM3	-
[81]	IM4	-
[81]	IM5	-
[81]	IM6	-
[81]	IM7	-

analysis	SR	group
[83]	0J_6j_2b	g1
[83]	0J_6j_3b	g1
[83]	0J_6j_m4b	g1
[83]	1J_6j_3b-HM	g2
[83]	1J_6j_3b_LM	g3
[83]	1J_6j_m4b-HM	g2
[83]	1J_6j_m4b_LM	g3
[83]	m2J_6j_2b	g4
[83]	m2J_6j_3b	g4
[83]	m2J_6j_m4b	g4

Table A.7: Disjoint groups

analysis	SR	group
[77]	4jHighX	g2
[77]	4jLowX	g2
[77]	5j	g2
[77]	5jSoftLep	g1
[77]	6j	g2

analysis	SR	group
[80]	SRZ	g1

Table A.8: Disjoint groups

analysis	SR	group
[82]	SRA_b0_b.mediumMET	g2
[82]	SRA_b0_c.highMET	g3
[82]	SRA_b0_d.vhighMET	g4
[82]	SRA_b1_a.lowMET	g5
[82]	SRA_b1_b.mediumMET	g6
[82]	SRA_b1_c.highMET	g7
[82]	SRA_b1_d.vhighMET	g8
[82]	SRB_b0_a.lowMET	g9
[82]	SRB_b0_b.mediumMET	g10
[82]	SRB_b0_c.highMET	g11
[82]	SRB_b0_d.vhighMET	g12
[82]	SRB_b1_a.lowMET	g13
[82]	SRB_b1_b.mediumMET	g14
[82]	SRB_b1_c.highMET	g15
[82]	SRB_b1_d.vhighMET	g16

[82]	SRC_ATLAS	-
[82]	SRD_cen_a.vlowMll_b0	g_1
[82]	SRD_cen_a.vlowMll_b1	g_1
[82]	SRD_cen_a.vlowMll_b_all	g_1
[82]	SRD_cen_b.lowMll_b0	g_2
[82]	SRD_cen_b.lowMll_b1	g_2
[82]	SRD_cen_b.lowMll_b_all	g_2
[82]	SRD_cen_c.mediumMll_b0	g_3
[82]	SRD_cen_c.mediumMll_b1	g_3
[82]	SRD_cen_c.mediumMll_b_all	g_3
[82]	SRD_cen_d.highMll_b0	g_4
[82]	SRD_cen_d.highMll_b1	g_4
[82]	SRD_cen_d.highMll_b_all	g_4
[82]	SRD_cen_e.vhighMll_b0	g_5
[82]	SRD_cen_e.vhighMll_b1	g_5
[82]	SRD_cen_e.vhighMll_b_all	g_5
[82]	SRE_for_a.vlowMll_b0	g_6
[82]	SRE_for_a.vlowMll_b1	g_6
[82]	SRE_for_a.vlowMll_b_all	g_6
[82]	SRE_for_b.lowMll_b0	g_7
[82]	SRE_for_b.lowMll_b1	g_7
[82]	SRE_for_b.lowMll_b_all	g_7
[82]	SRE_for_c.mediumMll_b0	g8
[82]	SRE_for_c.mediumMll_b1	g_8
[82]	SRE_for_c.mediumMll_b_all	g8
[82]	SRE_for_d.highMll_b0	g9
[82]	SRE_for_d.highMll_b1	g_9
[82]	SRE_for_d.highMll_b_all	g9
[82]	SRE_for_e.vhighMll_b0	g10
[82]	SRE_for_e.vhighMll_b1	g_10
[82]	SRE_for_e.vhighMll_b_all	g10

Table A.9: Disjoint groups for analysis [82]

A.4 Production of 8 TeV events (second calss) in detail

1. gluino pair production

```
Beams:idA = 2212 !first incoming beam is a 2212, i.e. a proton.  
Beams:idB = 2212 !second beam is also a proton.  
Beams:eCM = 8000. !the cm energy of collisions.  
  
#Allow this many times that pythia.next() returns false, i.e.  
# that an event is flawed, before aborting the run.  
Main:timesAllowErrors = 100000  
  
! Gluino Pair Production  
SUSY:gg2gluinogluino = on  
SUSY:qqbar2gluinogluino = on  
  
PartonLevel:FSR = on  
PartonLevel:ISR = on  
HadronLevel:Hadronize = on  
Tune:pp = 14  
  
SLHA:readFrom = 2 !read from SLHA file  
SLHA:useDecayTable = on !is default, but include it here anyway
```

2. gluino squark production/ gluino antisquark production

```
Beams:idA = 2212 !first incoming beam is a 2212, i.e. a proton.  
Beams:idB = 2212 !second beam is also a proton.  
Beams:eCM = 8000. !the cm energy of collisions.  
  
Main:timesAllowErrors = 100000  
  
! Associated Squark–Gluino Production  
SUSY:qg2squarkgluino = on  
  
PartonLevel:FSR = on  
PartonLevel:ISR = on  
HadronLevel:Hadronize = on  
Tune:pp = 14  
  
SLHA:readFrom = 2 !read from SLHA file  
SLHA:useDecayTable = on !is default, but include it here anyway
```

3. squark pair production / antisquark pair production

```
Beams:idA = 2212 !first incoming beam is a 2212, i.e. a proton.  
Beams:idB = 2212 !second beam is also a proton.  
Beams:eCM = 8000. !the cm energy of collisions.
```

```

Main:timesAllowErrors = 100000

! Squark Pair Production
SUSY:qq2squarksquark = on

PartonLevel:FSR = on
PartonLevel:ISR = on
HadronLevel:Hadronize = on
Tune:pp = 14

SLHA:readFrom = 2 !read from SLHA file
SLHA:useDecayTable = on !is default , but include it here anyway

```

4. squark-antisquark production

```

Beams:idA = 2212 !first incoming beam is a 2212, i.e. a proton.
Beams:idB = 2212 !second beam is also a proton.
Beams:eCM = 8000. !the cm energy of collisions.

Main:timesAllowErrors = 100000

! Squark Pair Production
SUSY:gg2squarkantisquark =on
SUSY:qbar2squarkantisquark = on

PartonLevel:FSR = on
PartonLevel:ISR = on
HadronLevel:Hadronize = on
Tune:pp = 14

SLHA:readFrom = 2 !read from SLHA file
SLHA:useDecayTable = on !is default , but include it here anyway

```

5. strong-electroweak mix production

```

Beams:idA = 2212 !first incoming beam is a 2212, i.e. a proton.
Beams:idB = 2212 !second beam is also a proton.
Beams:eCM = 8000. !the cm energy of collisions.

Main:timesAllowErrors = 100000

! Associated Neutralino/Chargino + Squark/Gluino Production
SUSY:qg2chi0squark =on
SUSY:qg2chi+—squark =on
SUSY:qbar2chi0gluino =on
SUSY:qbar2chi+—gluino =on

PartonLevel:FSR = on

```

```

PartonLevel:ISR = on
HadronLevel:Hadronize = on
Tune:pp = 14

SLHA:readFrom = 2 !read from SLHA file
SLHA:useDecayTable = on !is default , but include it here anyway

```

c++ program:

```

// Headers and Namespaces .
#include<iostream>
#include<string>
#include "Pythia8/Pythia.h" // Include Pythia headers .
#include "Pythia8Plugins/HepMC2.h"

using namespace Pythia8;

// Let Pythia8:: be implicit .
int main( int argc , char* argv [] ) {
    // Begin main program .
    // Interface for conversion from Pythia8::Event to HepMC event .
    HepMC::Pythia8ToHepMC ToHepMC;

    // Specify file where HepMC events will be stored .
    HepMC::IO_GenEvent ascii_io( argv [2] , std::ios ::out );

    // Set up generation .
    Pythia pythia ;
    // Declare Pythia object
    pythia.readFile( argv [1] );
    pythia.readString( std::string ("SLHA:file = ") + std::string ( argv [3] ) );
    int nEvent = atoi( argv [4] );
    std::cout << nEvent << std::endl ;
    pythia.init(); // Initialize ; incoming pp beams is default .
    // Generate event(s) .
    int iAbort=0;
    int nAbort=10000;
    for ( int iEvent = 0; iEvent < nEvent; ++iEvent ) {
        /*bool status =pythia.next ();
        if ( !status ){
            iEvent=iEvent -1;
            continue ;
        }*/
        if ( !pythia.next () ) {
            // First few failures write off as "acceptable" errors , then quit .
            if ( ++iAbort < nAbort ){
                iEvent = iEvent -1;

```

```

        continue;
    }
    cout << " Event generation aborted prematurely , owing to error!\\n";
    break;
}
// std :: cout<<"status "<<status<<std :: endl;
HepMC::GenEvent* hepmcevt = new HepMC::GenEvent();
ToHepMC. fill_next_event( pythia , hepmcevt );

// Write the HepMC event to file . Done with it .
ascii_io << hepmcevt;
delete hepmcevt;
}

pythia.stat();
std :: cout<<"+++xsection "<<pythia.info.sigmaGen()<<std :: endl;
std :: cout<<"+++xsection_err "<<pythia.info.sigmaErr()<<std :: endl;
return 0;
}

```

A.5 Production of 13 TeV events in detail

The syntax for process generation is:

1. gluino-gluino production

```

import model mssm
generate p p > go go

```

and with jet radiation:

```

import model mssm
define sq= go ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~
dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > go go @0
add process p p > go go j @1 $sq

```

The dollar sign forbids the particles after the dollar sign to go onshell.

2. squark-gluino/ antiquark-gluino production

```

import model mssm
define kn= ul ur dl dr cl cr sl sr b1 b2 t1 t2
define kna= ul~ ur~ dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > kn go
add process p p > kna go

```

and with jet radiation:

```

import model mssm
define sq= go ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~
dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
define knkna= ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~
dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > knkna go @0
add process p p > knkna go j @1 $sq

```

3. squark-squark / antisquark-antisquark production

```

import model mssm
define kn= ul ur dl dr cl cr sl sr b1 b2 t1 t2
define kna= ul~ ur~ dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > kn kn
add process p p > kna kna

```

and with jet radiation:

```

import model mssm
define sq= go ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~ dl~
dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
define kn= ul ur dl dr cl cr sl sr b1 b2 t1 t2
define kna= ul~ ur~ dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > kn kn @0
add process p p > kn kn j @1 $sq
add process p p > kna kna @2
add process p p > kna kna j @3 $sq

```

4. squark-antisquark production

```

import model mssm
define kn= ul ur dl dr cl cr sl sr
define kna= ul~ ur~ dl~ dr~ cl~ cr~ sl~ sr~
generate p p > kn kna

```

and with jet radiation:

```

import model mssm
define sq= go ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~
dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
define kn= ul ur dl dr cl cr sl sr
define kna= ul~ ur~ dl~ dr~ cl~ cr~ sl~ sr~
generate p p > kn kna @0
add process p p > kn kna j @1 $sq

```

5. stop1-antistop1 production

```

import model mssm
generate p p > t1 t1~

```

and with jet radiation:

```

import model mssm
define sq= go ul ur dl dr cl cr sl sr b1 b2 t1 t2 ul~ ur~
dl~ dr~ cl~ cr~ sl~ sr~ b1~ b2~ t1~ t2~
generate p p > t1 t1~ @0
add process p p > t1 t1~ j @1 $sq

```

6. stop2-antistop2 production analogous
7. sbottom1-antisbottom1 production analogous
8. sbottom2-antisbottom2 production analogous

The Madgraph run card setting which are changed from default are:

```

... = nevents
6500 = ebeam1
6500 = ebeam2
cteq6l1 = pdflabel
5 = asrwgtflavor
5 = maxjetflavor

#the following settings are only chaged if the mass gap is smaller
#than 300 GeV
1 = ickkw
False = auto_ptj_mjj
0.17 hard_scale = ptj
0.17 hard_scale = ptb
0.17 hard_scale = mmjj
0.17 hard_scale = xqcut
QCUT= 1.2 xqcut

```

where the hard scale is the average of the masses of all final state partons, like it is described on the madgraph homepage [97].

A.6 Implementation of 13 TeV analyses into Checkmate

The following two sections will show selected validation tables which compare the cutflow tables of the original analysis with the cutflow tables which have been obtained by using the analysis implemented into Checkmate.

The normalized events in the following tables have been calculated via the formula:

$$N_{Norm} = \frac{N\sigma_{process}\mathcal{L}_{int}}{N_{gen}}, \quad (\text{A.10})$$

where $\sigma_{process}$ pb is the NLO+NLL xsection for the corresponding process from NLL-Fast. The error on the cross section comes from the scale variation.

To get the error on N_{Norm} one does a simple error propagation:

$$\sigma_{N_{Norm}} = \sqrt{\left(\frac{\sigma_N \sigma_{process} \mathcal{L}_{int}}{N_{gen}}\right)^2 + \left(\frac{N \sigma_{process} \mathcal{L}_{int}}{N_{gen}}\right)^2}, \quad \sigma_N = \sqrt{N} \quad (\text{A.11})$$

The ratio and of the cutted events and the corresponding error intervals are calculated (like in the paper) from the normalized values.

The starting normalized events are sometimes not exactly equal due to slightly different cross sections.

A.6.1 Search for gluinos in events with an isolated lepton, jets and missing transverse momentum at 13 TeV

In this subsection we list the cutflow tables of analysis [77] and compare them to the cutflow tables which have been produced with Checkmate.

A.6.1.1 hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1105, 865, 625)\text{GeV}$ simplified model

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	509	496
$E_T^{miss} > 250$	56.2, 0.11	$51.1733 \pm 4.5694, 0.1032(0.0864, 0.1232)$
$N_{jets} \geq 5$	47.3, 0.84	$45.3578 \pm 4.059, 0.8864(0.7409, 1.0604)$
$p_T^{j_1} > 225 \text{ GeV}$	26.3, 0.556	$28.2204 \pm 2.5549, 0.6222(0.5194, 0.7452)$
$p_T^{j_5} > 50 \text{ GeV}$	17.4, 0.661	$19.9327 \pm 1.8272, 0.7063(0.5883, 0.8478)$
Jet Aplanarity > 0.04	10.4, 0.598	$12.263 \pm 1.1533, 0.6152(0.5106, 0.741)$
$m_T > 275 \text{ GeV}$	4.6, 0.44	$5.5345 \pm 0.56, 0.4513(0.3708, 0.5486)$
$\frac{E_T^{miss}}{m_{eff}} > 0.1$	4.6, 1	$5.5205 \pm 0.5587, 0.9975(0.8141, 1.2221)$
$m_{eff} > 1800 \text{ GeV}$	1.1, 0.239	$1.6435 \pm 0.2095, 0.2977(0.2359, 0.3735)$

Table A.10: Cutflow for hard 5j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	509	496.0
$E_T^{miss} > 250$	56.2, 0.1104	$51.1733 \pm 4.5694, 0.1032(0.0864, 0.1232)$
$N_{jets} \geq 6$	34, 0.60	$36.3537 \pm 3.2688, 0.7104(0.5935, 0.8502)$
$p_T^{j_1} > 125 \text{ GeV}$	33.4, 0.982	$35.2159 \pm 3.1689, 0.9687(0.8088, 1.0)$
$p_T^{j_6} > 30 \text{ GeV}$	33.4, 1	$30.8613 \pm 2.7867, 0.8763(0.7314, 1.0)$
Jet Aplanarity > 0.04	22.1, 0.661	$20.607 \pm 1.8864, 0.6677(0.5564, 0.8012)$
$m_T > 225 \text{ GeV}$	14.8, 0.67	$13.2042 \pm 1.236, 0.6408(0.5321, 0.7714)$
$\frac{E_T^{miss}}{m_{eff}} > 0.2$	13.7, 0.93	$11.5466 \pm 1.0902, 0.8745(0.7241, 1.0)$
$m_{eff} > 1000 \text{ GeV}$	12.9, 0.94	$11.0129 \pm 1.0433, 0.9538(0.7889, 1.153)$

Table A.11: Cutflow for hard 6j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	509	496
$E_T^{miss} > 200$	70, 0.137	$68.1702 \pm 6.0609, 0.1374(0.1151, 0.1641)$
$N_{jets} \geq 4$	66, 0.94	$66.2738 \pm 5.8945, 0.9722(0.8134, 1.162)$
$p_T^{j_1} > 325$ GeV	15.8, 0.23	$18.6966 \pm 1.7187, 0.2821(0.2353, 0.3381)$
$p_T^{j_4} > 30$ GeV	15.8, 1.0	$18.3314 \pm 1.6866, 0.9805(0.8153, 1.1791)$
$m_T > 425$ GeV	-	-
$\frac{E_T^{miss}}{m_{eff}} > 0.3$ GeV	1.2, 0.0759	$0.899 \pm 0.1373, 0.049(0.0381, 0.0623)$
$m_{eff} > 1800$ GeV	1.0, 0.83	$0.4495 \pm 0.0887, 0.5(0.3482, 0.7066)$

Table A.12: Cutflow for hard 4j high x. The last two cutflows do not agree perfectly, but this is due to the low number of events in the cutflows (30 events). The statistical error is highly underestimated and running the event generation again shows that too.

A.6.1.2 hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1385, 705, 25)$ GeV simplified model

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	88.7	86.4
$E_T^{miss} > 250$	17.0, 0.19	$17.2048 \pm 1.5422, 0.1991(0.1665, 0.2382)$
$N_{jets} \geq 5$	16.1, 0.95	$16.4361 \pm 1.4739, 0.9553(0.7981, 1.1435)$
$p_T^{j_1} > 225$ GeV	16.1, 1.0	$16.2607 \pm 1.4583, 0.9893(0.8265, 1.1843)$
$p_T^{j_5} > 50$ GeV	14.2, 0.88	$14.5712 \pm 1.3081, 0.8961(0.7485, 1.0728)$
Jet Aplanarity > 0.04	11.3, 0.80	$11.3583 \pm 1.0225, 0.7795(0.6509, 0.9335)$
$m_T > 275$ GeV	7.5, 0.66	$7.0652 \pm 0.6409, 0.622(0.5189, 0.7456)$
$\frac{E_T^{miss}}{m_{eff}} > 0.1$	7.4, 0.99	$6.9567 \pm 0.6312, 0.9846(0.8208, 1.1811)$
$m_{eff} > 1800$ GeV	6.7, 0.91	$6.2435 \pm 0.5678, 0.8975(0.748, 1.0768)$

Table A.13: Cutflow for hard 5j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	88.7	86.4
$E_T^{miss} > 250$	17.0, 0.19	$17.2048 \pm 1.5422, 0.1991(0.1665, 0.2382)$
$N_{jets} \geq 6$	13.4, 0.79	$14.3196 \pm 1.2858, 0.8323(0.6952, 0.9964)$
$p_T^{j_1} > 125$ GeV	13.4, 1.0	$14.3196 \pm 1.2858, 1.0(0.8352, 1.1973)$
$p_T^{j_6} > 30$ GeV	13.4, 1.0	$13.5533 \pm 1.2177, 0.9465(0.7905, 1.1333)$
Jet Aplanarity > 0.04	10.6, 0.80	$10.5666 \pm 0.9521, 0.7796(0.6509, 0.9338)$
$m_T > 225$ GeV	8.0, 0.75	$7.4068 \pm 0.6712, 0.701(0.5847, 0.8402)$
$\frac{E_T^{miss}}{m_{eff}} > 0.2$	4.7, 0.59	$4.3185 \pm 0.3966, 0.583(0.4855, 0.7)$
$m_{eff} > 1000$ GeV	4.7, 1.0	$4.3208 \pm 0.3968, 1.0005(0.8322, 1.2029)$

Table A.14: Cutflow for hard 6j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	88.7	86.4
$E_T^{miss} > 200$	19.2, 0.22	$19.0651 \pm 1.7076, 0.2207(0.1845, 0.2639)$
$N_{jets} >= 4$	19.0, 0.99	$18.9797 \pm 1.7, 0.9955(0.8318, 1.1914)$
$p_T^{j_1} > 325$ GeV	17.2, 0.91	$16.9624 \pm 1.5207, 0.8937(0.7467, 1.0696)$
$p_T^{j_4} > 150$ GeV	9.3, 0.54	$10.0149 \pm 0.9031, 0.5904(0.493, 0.707)$
Jet Aplanarity > 0.04	7.9, 0.85	$8.1408 \pm 0.7365, 0.8129(0.6782, 0.9743)$
$m_T > 125$ GeV	6.8, 0.86	$6.7697 \pm 0.6146, 0.8316(0.6934, 0.9973)$
$m_{eff} > 2000$ GeV	5.6, 0.82	$5.6872 \pm 0.5184, 0.8401(0.7, 1.0082)$

Table A.15: Cutflow for hard 4j low x

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	88.7	86.4
$E_T^{miss} > 200$	19.2, 0.22	$19.0651 \pm 1.7076, 0.2207(0.1845, 0.2639)$
$N_{jets} >= 4$	19.0, 0.99	$18.9797 \pm 1.7, 0.9955(0.8318, 1.1914)$
$p_T^{j_1} > 325$ GeV	17.2, 0.91	$16.9624 \pm 1.5207, 0.8937(0.7467, 1.0696)$
$p_T^{j_4} > 30$ GeV	17.2, 1.0	$16.9209 \pm 1.517, 0.9976(0.8334, 1.194)$
$m_T > 425$ GeV	-	-
$\frac{E_T^{miss}}{m_{eff}} > 0.3$ GeV	1.33, 0.077	$1.0687 \pm 0.1072, 0.0632(0.0521, 0.0763)$
$m_{eff} > 1800$ GeV	1.24, 0.93	$0.9371 \pm 0.0954, 0.8769(0.7158, 1.0739)$

Table A.16: Cutflow for hard 4j high x

A.6.1.3 hard SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1200, 160, 60) GeV$ simplified model

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	275	263.6
$E_T^{miss} > 250$	34.9, 0.13	$35.0323 \pm 3.272, 0.1329(0.1103, 0.1601)$
$N_{jets} >= 5$	33.2, 0.95	$33.9745 \pm 3.1744, 0.9698(0.8041, 1.1697)$
$p_T^{j_1} > 225$ GeV	33.1, 0.997	$33.8432 \pm 3.1623, 0.9961(0.8259, 1.2015)$
$p_T^{j_5} > 50$ GeV	30.5, 0.92	$30.8958 \pm 2.8903, 0.9129(0.7568, 1.1012)$
Jet Aplanarity > 0.04	23.2, 0.76	$23.491 \pm 2.2069, 0.7603(0.63, 0.9176)$
$m_T > 275$ GeV	2.1, 0.09	$2.3929 \pm 0.2573, 0.1019(0.0831, 0.1245)$
$\frac{E_T^{miss}}{m_{eff}} > 0.1$	2.07, 0.98	$2.3053 \pm 0.2491, 0.9634(0.7759, 1.1962)$
$m_{eff} > 1800$ GeV	1.89, 0.91	$2.1011 \pm 0.23, 0.9114(0.7324, 1.1337)$

Table A.17: Cutflow for hard 5j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	275	263.6
$E_T^{miss} > 250$	34.9, 0.13	$35.0323 \pm 3.272, 0.1329(0.1103, 0.1601)$
$N_{jets} \geq 6$	29.3, 0.84	$30.5821 \pm 2.8613, 0.873(0.7237, 1.053)$
$p_T^{j_1} > 125$ GeV	29.3, 1.0	$30.5821 \pm 2.8613, 1.0(0.8289, 1.2064)$
$p_T^{j_6} > 30$ GeV	29.3, 1.0	$29.0501 \pm 2.7199, 0.9499(0.7873, 1.1461)$
Jet Aplanarity > 0.04	22.2, 0.76	$22.28 \pm 2.0951, 0.767(0.6353, 0.9257)$
$m_T > 225$ GeV	2.5, 0.11	$3.582 \pm 0.368, 0.1608(0.1319, 0.1957)$
$\frac{E_T^{miss}}{m_{eff}} > 0.2$	0.72, 0.29	$0.9265 \pm 0.1186, 0.2587(0.2045, 0.3252)$
$m_{eff} > 1000$ GeV	0.72, 1.0	$0.9265 \pm 0.1186, 1.0(0.773, 1.2936)$

Table A.18: Cutflow for hard 6j SR

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	275	263.6
$E_T^{miss} > 200$	42.7, 0.16	$42.065 \pm 3.921, 0.1596(0.1325, 0.1923)$
$N_{jets} \geq 4$	42.4, 0.99	$41.9629 \pm 3.9116, 0.9976(0.8275, 1.2027)$
$p_T^{j_1} > 325$ GeV	40.8, 0.96	$39.8545 \pm 3.717, 0.9498(0.7877, 1.1451)$
$p_T^{j_4} > 150$ GeV	26.2, 0.64	$25.1398 \pm 2.359, 0.6308(0.5228, 0.7609)$
Jet Aplanarity > 0.04	21.6, 0.82	$20.5875 \pm 1.9389, 0.8189(0.6782, 0.9888)$
$m_T > 125$ GeV	8.3, 0.38	$8.5575 \pm 0.8282, 0.4157(0.3431, 0.5033)$
$m_{eff} > 2000$ GeV	6.8, 0.82	$7.0984 \pm 0.6934, 0.8295(0.6824, 1.0081)$

Table A.19: Cutflow for hard 4j low x

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	275	263.6
$E_T^{miss} > 200$	42.7, 0.16	$42.065 \pm 3.921, 0.1596(0.1325, 0.1923)$
$N_{jets} \geq 4$	42.4, 0.99	$41.9629 \pm 3.9116, 0.9976(0.8275, 1.2027)$
$p_T^{j_1} > 325$ GeV	40.8, 0.96	$39.8545 \pm 3.717, 0.9498(0.7877, 1.1451)$
$p_T^{j_4} > 30$ GeV	40.8, 1.0	$39.7962 \pm 3.7116, 0.9985(0.8282, 1.204)$
$m_T > 425$ GeV	-	-
$\frac{E_T^{miss}}{m_{eff}} > 0.3$ GeV	0.0, 0.0	$0.1094 \pm 0.03, 0.0027(0.0018, 0.0039)$
$m_{eff} > 1800$ GeV	0.0, -	$0.0948 \pm 0.0277, 0.8667(0.4814, 1.5431)$

Table A.20: Cutflow for hard 4j high x. In the last two signal regions we have 15 and 13 events. Therefore the given error which relies on the poisson distribution is highly underestimated.

A.6.1.4 soft SR, using $(m_{\tilde{g}}, m_{\tilde{\chi}_1^\pm}, m_{\tilde{\chi}_1^0}) = (1000, 110, 60) \text{GeV}$ simplified model

cut	$N_{weighted}$, fraction (from paper)	N_{Norm} , fraction (from CM analysis)
start	1059	1024
$E_T^{miss} > 375$	26.5, 0.025	$26.8044 \pm 2.4312, 0.0262(0.0219, 0.0312)$
$N_{jets} \geq 5$	24.7, 0.93	$25.0828 \pm 2.2855, 0.9358(0.7798, 1.0)$
$p_T^{j_1} > 200 \text{ GeV}$	24.4, 0.987	$25.0531 \pm 2.283, 0.9988(0.832, 1.0)$
$p_T^{j_2} > 200 \text{ GeV}$	22.1, 0.91	$22.9455 \pm 2.1046, 0.9159(0.7624, 1.0)$
$p_T^{j_3} > 200 \text{ GeV}$	14.6, 0.66	$15.4059 \pm 1.4653, 0.6714(0.5565, 0.8095)$
Jet Aplanarity > 0.02	13.4, 0.91	$13.6842 \pm 1.3188, 0.8882(0.7329, 1.0)$
$H_T > 1100 \text{ GeV}$	12.9, 0.96	$13.3577 \pm 1.291, 0.9761(0.8043, 1.0)$

Table A.21: Cutflow for soft 5j SR

A.6.2 Search for supersymmetry at 13 TeV in final states with jets and two same-sign leptons or three leptons

In this subsection we list the cutflow tables for analysis [78]. We compare the tables to the cutflow tables produced with Checkmate.

A.6.2.1 SR0b3j

cut	Events	Normalized events	ratio of cutted events (errors)
produced	20051	148	-
≥ 3 leptons ($p_T > 20, 20, 10 \text{ GeV}$)	2026	14.4 ± 0.4	0.097 (0.0946-0.1)
trigger	2024	14.3 ± 0.4	0.99 (0.94-1.0)
no bjet ($p_T > 20 \text{ GeV}$)	1665	11.66 ± 0.33	0.82 (0.77-0.86)
≥ 3 jets ($p_T > 50 \text{ GeV}$)	1556	10.91 ± 0.32	0.94 (0.88-0.99)
$E_T^{miss} > 200 \text{ GeV}$	1079	7.52 ± 0.27	0.69 (0.65-0.74)
$m_{eff} > 550 \text{ GeV}$	1079	7.52 ± 0.27	1.0 (0.93-1.0)

Table A.22: Cutflow table for SR SR0b3j (from paper) with simplified model from figure 1a in the paper and $m_{\tilde{g}} = 1.3 \text{ TeV}$ and $m_{\tilde{\chi}_1^0} = 0.5 \text{ TeV}$

cut	Events	Normalized events	ratio of cutted events (errors)
produced	44682	143.7	-
≥ 3 leptons ($p_T > 20, 20, 10 \text{ GeV}$)	4339	14.0 ± 1.3	0.0971(0.081, 0.116)
trigger	4202	13.5 ± 1.2	0.97(0.81, 1.0)
no bjet ($p_T > 20 \text{ GeV}$)	3565	11.46 ± 1.0	0.85(0.71, 1.0)
≥ 3 jets ($p_T > 50 \text{ GeV}$)	3343	10.7 ± 0.97	0.94(0.78, 1.0)
$E_T^{miss} > 200 \text{ GeV}$	2387	7.7 ± 0.70	0.71(0.60, 0.86)
$m_{eff} > 550 \text{ GeV}$	2387	7.7 ± 0.70	1.0(0.83, 1.0)

Table A.23: Cutflow table for SR SR0b3j (produced with Checkmate) with simplified model from figure 1a in the paper and $m_{\tilde{g}} = 1.3 \text{ TeV}$ and $m_{\tilde{\chi}_1^0} = 0.5 \text{ TeV}$

A.6.2.2 SR1b

cut	Events	Normalized events	ratio of cutted events (error intervall)
produced	94706	560	-
≥ 2 SS leptons ($p_T > 20$ GeV)	4115	25.5 ± 0.5	0.0455 (0.0446-0.0464)
trigger	3901	23.1 ± 0.4	0.91 (0.87-0.94)
≥ 1 bjet ($p_T > 20$ GeV)	3367	20.0 ± 0.4	0.87 (0.83-0.899)
≥ 4 jets ($p_T > 50$ GeV)	1881	11.23 ± 0.3	0.56 (0.54-0.59)
$E_T^{miss} > 150$ GeV	1148	7.08 ± 0.24	0.63 (0.59-0.67)
$m_{eff} > 550$ GeV	1148	7.08 ± 0.24	1.0 (0.93-1.0)

Table A.24: Cutflow table for SR SR0b3j (from paper) with simplified model from figure 1c in the paper and $m_{\tilde{b}_1} = 0.6$ TeV, $m_{\tilde{\chi}_l^\pm} = 150$ GeV and $m_{\tilde{\chi}_1^0} = 50$ GeV

cut	Events	Normalized events	ratio of cutted events (errors)
produced	26154	550	-
≥ 2 SS leptons ($p_T > 20$ GeV)	1143	$24. \pm 2.0$	0.044(0.037,0.052)
trigger	991	20.9 ± 1.8	0.87(0.73,1.0)
≥ 1 bjet ($p_T > 20$ GeV)	831	17.5 ± 1.5	0.84(0.70,1.0)
≥ 4 jets ($p_T > 50$ GeV)	418	8.8 ± 0.84	0.50(0.42,0.60)
$E_T^{miss} > 150$ GeV	246	5.2 ± 0.54	0.59(0.48,0.72)
$m_{eff} > 550$ GeV	246	5.2 ± 0.54	1.0(0.81,1.0)

Table A.25: Cutflow table for SR SR0b3j (from Checkmate) with simplified model from figure 1c in the paper and $m_{\tilde{b}_1} = 0.6$ TeV, $m_{\tilde{\chi}_l^\pm} = 150$ GeV and $m_{\tilde{\chi}_1^0} = 50$ GeV

A.6.2.3 SR3b with events from model in figure 1d

cut	Events	Normalized events	ratio of cutted events (errors)
produced	100000	275	-
≥ 2 SS leptons ($p_T > 20$ GeV)	3535	9.28 ± 0.18	0.0338 (0,0331-0,0344)
trigger	3386	8.53 ± 0.17	0.92 (0.88-0,956)
≥ 3 bjet ($p_T > 20$ GeV)	1704	4.26 ± 0.12	0.499 (0.48-0.524)
$E_T^{miss} > 125$ GeV	1320	3.31 ± 0.11	0.78 (0.73-0.826)
$m_{eff} > 650$ GeV	1280	3.20 ± 0.10	0.97 (0.91-1.0)

Table A.26: Cutflow table for SR SR0b3j (from paper) with simplified model from figure 1d in the paper and $m_{\tilde{g}} = 1.2$ TeV and $m_{\tilde{\chi}_1^0} = 700$ GeV

cut	Events	Normalized events	ratio of cutted events (errors)
produced	44348	267.5	-
≥ 2 SS leptons ($p_T > 20$ GeV)	1452	8.8 ± 0.80	$0.0327(0.0273,0.0391)$
trigger	1297	7.82 ± 0.72	$0.89(0.74,1.0)$
≥ 3 bjet ($p_T > 20$ GeV)	596	3.60 ± 0.35	$0.46(0.38,0.56)$
$E_T^{miss} > 125$ GeV	462	2.79 ± 0.28	$0.78(0.64,0.94)$
$m_{eff} > 650$ GeV	444	2.68 ± 0.27	$0.96(0.79,1.0)$

Table A.27: Cutflow table for SR SR0b3j (from Checkmate) with simplified model from figure 1d in the paper and $m_{\tilde{g}} = 1.2$ TeV and $m_{\tilde{\chi}_1^0} = 700$ GeV

Bibliography

- [1] F. Quevedo, S. Krippendorf, and O. Schlotterer, “Cambridge Lectures on Supersymmetry and Extra Dimensions,” [arXiv:1011.1491 \[hep-th\]](https://arxiv.org/abs/1011.1491).
- [2] I. J. R. Aitchison, “Supersymmetry and the MSSM: An Elementary introduction,” [arXiv:hep-ph/0505105 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0505105).
- [3] S. P. Martin, “A Supersymmetry primer,” [arXiv:hep-ph/9709356 \[hep-ph\]](https://arxiv.org/abs/hep-ph/9709356). [Adv. Ser. Direct. High Energy Phys.18,1(1998)].
- [4] M. Czakon, “Quantum field theory lecture notes.”.
- [5] T. Blum, A. Denig, I. Logashenko, E. de Rafael, B. Lee Roberts, T. Teubner, and G. Venanzoni, “The Muon (g-2) Theory Value: Present and Future,” [arXiv:1311.2198 \[hep-ph\]](https://arxiv.org/abs/1311.2198).
- [6] **Muon g-2** Collaboration, F. Gray, “Muon g-2 Experiment at Fermilab,” in Proceedings, 12th Conference on the Intersections of Particle and Nuclear Physics (CIPANP 2015): Vail, Co 2015. [arXiv:1510.00346 \[physics.ins-det\]](https://arxiv.org/abs/1510.00346).
<https://inspirehep.net/record/1395618/files/arXiv:1510.00346.pdf>.
- [7] S. P. Martin and J. D. Wells, “Muon anomalous magnetic dipole moment in supersymmetric theories,” [Phys. Rev. D64](https://doi.org/10.1103/PhysRevD.64.035003) (2001) 035003, [arXiv:hep-ph/0103067 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0103067).
- [8] Y. Uehara, “Right-handed neutrinos as superheavy dark matter,” [JHEP 12](https://doi.org/10.1088/1126-6708/12/03/034) (2001) 034, [arXiv:hep-ph/0107297 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0107297).
- [9] S. Weinberg, The Quantum Theory of Fields: Supersymmetry. Cambridge Press, 2000.
- [10] R. Haag, J. Lopusanski, and M. Sohnius [Nucl. Phys. B 88](https://doi.org/10.1016/0550-3213(75)90001-7) no. 257, (1975) .
- [11] R. N. Mohapatra, “Supersymmetry and R-parity: an Overview,” [Phys. Scripta 90](https://doi.org/10.1080/08333322.2015.1088004) (2015) 088004, [arXiv:1503.06478 \[hep-ph\]](https://arxiv.org/abs/1503.06478).
- [12] P. Z. Skands et al., “SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators,” [JHEP 07](https://doi.org/10.1088/1126-6708/07/03/036) (2004) 036, [arXiv:hep-ph/0311123 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0311123).
- [13] S. S. AbdusSalam et al., “Benchmark Models, Planes, Lines and Points for Future SUSY Searches at the LHC,” [Eur. Phys. J. C71](https://doi.org/10.1007/s00361-011-0385-9) (2011) 1835, [arXiv:1109.3859 \[hep-ph\]](https://arxiv.org/abs/1109.3859).
- [14] P. Bechtle et al., “Constrained Supersymmetry after two years of LHC data: a global view with Fittino,” [JHEP 06](https://doi.org/10.1007/JHEP06(2012)098) (2012) 098, [arXiv:1204.4199 \[hep-ph\]](https://arxiv.org/abs/1204.4199).

- [15] P. Bechtle, K. Desch, M. Uhlenbrock, and P. Wienemann, “Constraining SUSY models with Fittino using measurements before, with and beyond the LHC,” *Eur. Phys. J.* **C66** (2010) 215–259, [arXiv:0907.2589 \[hep-ph\]](#).
- [16] P. Bechtle et al., “Killing the cMSSM softly,” *Eur. Phys. J.* **C76** no. 2, (2016) 96, [arXiv:1508.05951 \[hep-ph\]](#).
- [17] **Muon g-2** Collaboration, G. W. Bennett et al., “Final Report of the Muon E821 Anomalous Magnetic Moment Measurement at BNL,” *Phys. Rev.* **D73** (2006) 072003, [arXiv:hep-ex/0602035 \[hep-ex\]](#).
- [18] M. Davier, A. Hoecker, B. Malaescu, and Z. Zhang, “Reevaluation of the Hadronic Contributions to the Muon $g-2$ and to $\alpha(MZ)$,” *Eur. Phys. J.* **C71** (2011) 1515, [arXiv:1010.4180 \[hep-ph\]](#). [Erratum: *Eur. Phys. J.* C72,1874(2012)].
- [19] S. Heinemeyer, W. Hollik, and G. Weiglein, “FeynHiggs: A Program for the calculation of the masses of the neutral CP even Higgs bosons in the MSSM,” *Comput. Phys. Commun.* **124** (2000) 76–89, [arXiv:hep-ph/9812320 \[hep-ph\]](#).
- [20] H. Bahl and W. Hollik, “Precise prediction for the light MSSM Higgs boson mass combining effective field theory and fixed-order calculations,” *Eur. Phys. J.* **C76** no. 9, (2016) 499, [arXiv:1608.01880 \[hep-ph\]](#).
- [21] T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, and G. Weiglein, “High-Precision Predictions for the Light CP -Even Higgs Boson Mass of the Minimal Supersymmetric Standard Model,” *Phys. Rev. Lett.* **112** no. 14, (2014) 141801, [arXiv:1312.4937 \[hep-ph\]](#).
- [22] M. Frank, T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, and G. Weiglein, “The Higgs Boson Masses and Mixings of the Complex MSSM in the Feynman-Diagrammatic Approach,” *JHEP* **02** (2007) 047, [arXiv:hep-ph/0611326 \[hep-ph\]](#).
- [23] G. Degrassi, S. Heinemeyer, W. Hollik, P. Slavich, and G. Weiglein, “Towards high precision predictions for the MSSM Higgs sector,” *Eur. Phys. J.* **C28** (2003) 133–143, [arXiv:hep-ph/0212020 \[hep-ph\]](#).
- [24] S. Heinemeyer, W. Hollik, and G. Weiglein, “The Masses of the neutral CP - even Higgs bosons in the MSSM: Accurate analysis at the two loop level,” *Eur. Phys. J.* **C9** (1999) 343–366, [arXiv:hep-ph/9812472 \[hep-ph\]](#).
- [25] **SLD Electroweak Group, DELPHI, ALEPH, SLD, SLD Heavy Flavour Group, OPAL, LEP Electroweak Working Group, L3** Collaboration, S. Schael et al., “Precision electroweak measurements on the Z resonance,” *Phys. Rept.* **427** (2006) 257–454, [arXiv:hep-ex/0509008 \[hep-ex\]](#).
- [26] **ATLAS, CDF, CMS, D0** Collaboration, “First combination of Tevatron and LHC measurements of the top-quark mass,” [arXiv:1403.4427 \[hep-ex\]](#).
- [27] W. Porod, “SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e+ e- colliders,” *Comput. Phys. Commun.* **153** (2003) 275–315, [arXiv:hep-ph/0301101 \[hep-ph\]](#).

- [28] W. Porod and F. Staub, “SPheno 3.1: Extensions including flavour, CP-phases and models beyond the MSSM,” *Comput. Phys. Commun.* **183** (2012) 2458–2469, [arXiv:1104.1573 \[hep-ph\]](#).
- [29] **CDF, D0** Collaboration, T. E. W. Group, “2012 Update of the Combination of CDF and D0 Results for the Mass of the W Boson,” [arXiv:1204.0042 \[hep-ex\]](#).
- [30] e. a. J. Beringer *Phys. Rev.* **D86** no. 010001, (2012) .
- [31] **LHCb, CMS** Collaboration, CMS and L. Collaborations, “Combination of results on the rare decays $B_{(s)}^0 \rightarrow \mu^+ \mu^-$ from the CMS and LHCb experiments,” .
- [32] F. Mahmoudi, “SuperIso v2.3: A Program for calculating flavor physics observables in Supersymmetry,” *Comput. Phys. Commun.* **180** (2009) 1579–1613, [arXiv:0808.3144 \[hep-ph\]](#).
- [33] **Heavy Flavor Averaging Group** Collaboration, Y. Amhis et al., “Averages of B-Hadron, C-Hadron, and tau-lepton properties as of early 2012,” [arXiv:1207.1158 \[hep-ex\]](#).
- [34] P. Bechtle, O. Brein, S. Heinemeyer, G. Weiglein, and K. E. Williams, “HiggsBounds: Confronting Arbitrary Higgs Sectors with Exclusion Bounds from LEP and the Tevatron,” *Comput. Phys. Commun.* **181** (2010) 138–167, [arXiv:0811.4169 \[hep-ph\]](#).
- [35] P. Bechtle, O. Brein, S. Heinemeyer, G. Weiglein, and K. E. Williams, “HiggsBounds 2.0.0: Confronting Neutral and Charged Higgs Sector Predictions with Exclusion Bounds from LEP and the Tevatron,” *Comput. Phys. Commun.* **182** (2011) 2605–2631, [arXiv:1102.1898 \[hep-ph\]](#).
- [36] P. Bechtle, O. Brein, S. Heinemeyer, O. Stal, T. Stefaniak, G. Weiglein, and K. Williams, “Recent Developments in HiggsBounds and a Preview of HiggsSignals,” *PoS CHARGED2012* (2012) 024, [arXiv:1301.2345 \[hep-ph\]](#).
- [37] P. Bechtle, O. Brein, S. Heinemeyer, O. Stål, T. Stefaniak, G. Weiglein, and K. E. Williams, “HiggsBounds – 4: Improved Tests of Extended Higgs Sectors against Exclusion Bounds from LEP, the Tevatron and the LHC,” *Eur. Phys. J.* **C74** no. 3, (2014) 2693, [arXiv:1311.0055 \[hep-ph\]](#).
- [38] P. Bechtle, S. Heinemeyer, O. Stal, T. Stefaniak, and G. Weiglein, “Applying Exclusion Likelihoods from LHC Searches to Extended Higgs Sectors,” *Eur. Phys. J.* **C75** no. 9, (2015) 421, [arXiv:1507.06706 \[hep-ph\]](#).
- [39] P. Bechtle, S. Heinemeyer, O. Stål, T. Stefaniak, and G. Weiglein, “*HiggsSignals*: Confronting arbitrary Higgs sectors with measurements at the Tevatron and the LHC,” *Eur. Phys. J.* **C74** no. 2, (2014) 2711, [arXiv:1305.1933 \[hep-ph\]](#).
- [40] PlanckCollaboration *Astron. Astrophys.* **571** no. A16, (2014) .
- [41] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “MicrOMEGAs: A Program for calculating the relic density in the MSSM,” *Comput. Phys. Commun.* **149** (2002) 103–120, [arXiv:hep-ph/0112278 \[hep-ph\]](#).

- [42] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “micrOMEGAs: Version 1.3,” *Comput. Phys. Commun.* **174** (2006) 577–604, [arXiv:hep-ph/0405253 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0405253).
- [43] **LUX Collaboration** Collaboration, D. S. Akerib, H. M. Araújo, X. Bai, A. J. Bailey, J. Balajthy, S. Bedikian, E. Bernard, A. Bernstein, A. Bolozdynya, A. Bradley, D. Byram, S. B. Cahn, M. C. Carmona-Benitez, C. Chan, J. J. Chapman, A. A. Chiller, C. Chiller, K. Clark, T. Coffey, A. Currie, A. Curioni, S. Dazeley, L. de Viveiros, A. Dobi, J. Dobson, E. M. Dragowsky, E. Druszkiewicz, B. Edwards, C. H. Faham, S. Fiorucci, C. Flores, R. J. Gaitskell, V. M. Gehman, C. Ghag, K. R. Gibson, M. G. D. Gilchriese, C. Hall, M. Hanhardt, S. A. Hertel, M. Horn, D. Q. Huang, M. Ihm, R. G. Jacobsen, L. Kastens, K. Kazkaz, R. Knoche, S. Kyre, R. Lander, N. A. Larsen, C. Lee, D. S. Leonard, K. T. Lesko, A. Lindote, M. I. Lopes, A. Lyashenko, D. C. Malling, R. Mannino, D. N. McKinsey, D.-M. Mei, J. Mock, M. Moongweluwan, J. Morad, M. Morii, A. S. J. Murphy, C. Nehrkorn, H. Nelson, F. Neves, J. A. Nikkel, R. A. Ott, M. Pangilinan, P. D. Parker, E. K. Pease, K. Pech, P. Phelps, L. Reichhart, T. Shutt, C. Silva, W. Skulski, C. J. Sofka, V. N. Solovov, P. Sorensen, T. Stiegler, K. O’Sullivan, T. J. Sumner, R. Svoboda, M. Sweany, M. Szydagis, D. Taylor, B. Tennyson, D. R. Tiedt, M. Tripathi, S. Uvarov, J. R. Verbus, N. Walsh, R. Webb, J. T. White, D. White, M. S. Witherell, M. Wlasenko, F. L. H. Wolfs, M. Woods, and C. Zhang, “First results from the lux dark matter experiment at the sanford underground research facility,” *Phys. Rev. Lett.* **112** (Mar, 2014) 091303. <http://link.aps.org/doi/10.1103/PhysRevLett.112.091303>.
- [44] P. Gondolo, J. Edsjo, P. Ullio, L. Bergstrom, M. Schelke, and E. A. Baltz, “DarkSUSY: Computing supersymmetric dark matter properties numerically,” *JCAP* **0407** (2004) 008, [arXiv:astro-ph/0406204 \[astro-ph\]](https://arxiv.org/abs/astro-ph/0406204).
- [45] N. Nguyen, D. Horns, and T. Bringmann, “AstroFit: An Interface Program for Exploring Complementarity in Dark Matter Research,” in Proceedings, 13th ICATPP Conference on Astroparticle, Particle, Space Physics and Detectors for Physics pp. 1025–1030. 2012. [arXiv:1202.1385 \[astro-ph.HE\]](https://arxiv.org/abs/1202.1385).
<https://inspirehep.net/record/1088245/files/arXiv:1202.1385.pdf>.
- [46] **Particle Data Group** Collaboration, K. A. Olive *et al.*, “Review of Particle Physics,” *Chin. Phys. C* **38** (2014) 090001.
- [47] “<http://home.cern/topics/large-hadron-collider>.”
- [48] “<https://home.cern/about/experiments>.”
- [49] K. J. de Vries *et al.*, “The pMSSM10 after LHC Run 1,” *Eur. Phys. J.* **C75** no. 9, (2015) 422, [arXiv:1504.03260 \[hep-ph\]](https://arxiv.org/abs/1504.03260).
- [50] M. Drees, H. Dreiner, D. Schmeier, J. Tattersall, and J. S. Kim, “CheckMATE: Confronting your Favourite New Physics Model with LHC Data,” *Comput. Phys. Commun.* **187** (2015) 227–265, [arXiv:1312.2591 \[hep-ph\]](https://arxiv.org/abs/1312.2591).
- [51] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer, “MadGraph 5 : Going Beyond,” *JHEP* **06** (2011) 128, [arXiv:1106.0522 \[hep-ph\]](https://arxiv.org/abs/1106.0522).
- [52] T. Sjostrand, S. Mrenna, and P. Z. Skands, “PYTHIA 6.4 Physics and Manual,” *JHEP* **05** (2006) 026, [arXiv:hep-ph/0603175 \[hep-ph\]](https://arxiv.org/abs/hep-ph/0603175).

- [53] **ATLAS** Collaboration, G. Aad *et al.*, “Search for the electroweak production of supersymmetric particles in $\sqrt{s}=8$ TeV pp collisions with the ATLAS detector,” *Phys. Rev.* **D93** no. 5, (2016) 052002, [arXiv:1509.07152 \[hep-ex\]](#).
- [54] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, “An Introduction to PYTHIA 8.2,” *Comput. Phys. Commun.* **191** (2015) 159–177, [arXiv:1410.3012 \[hep-ph\]](#).
- [55] W. Beenakker, R. Hopker, M. Spira, and P. M. Zerwas, “Squark and gluino production at hadron colliders,” *Nucl. Phys.* **B492** (1997) 51–103, [arXiv:hep-ph/9610490 \[hep-ph\]](#).
- [56] A. Kulesza and L. Motyka, “Threshold resummation for squark-antisquark and gluino-pair production at the LHC,” *Phys. Rev. Lett.* **102** (2009) 111802, [arXiv:0807.2405 \[hep-ph\]](#).
- [57] A. Kulesza and L. Motyka, “Soft gluon resummation for the production of gluino-gluino and squark-antisquark pairs at the LHC,” *Phys. Rev.* **D80** (2009) 095004, [arXiv:0905.4749 \[hep-ph\]](#).
- [58] W. Beenakker, S. Brening, M. Kramer, A. Kulesza, E. Laenen, and I. Niessen, “Soft-gluon resummation for squark and gluino hadroproduction,” *JHEP* **12** (2009) 041, [arXiv:0909.4418 \[hep-ph\]](#).
- [59] W. Beenakker, S. Brening, M. n. Kramer, A. Kulesza, E. Laenen, L. Motyka, and I. Niessen, “Squark and Gluino Hadroproduction,” *Int. J. Mod. Phys.* **A26** (2011) 2637–2664, [arXiv:1105.1110 \[hep-ph\]](#).
- [60] W. Beenakker, M. Kramer, T. Plehn, M. Spira, and P. M. Zerwas, “Stop production at hadron colliders,” *Nucl. Phys.* **B515** (1998) 3–14, [arXiv:hep-ph/9710451 \[hep-ph\]](#).
- [61] W. Beenakker, S. Brening, M. Kramer, A. Kulesza, E. Laenen, and I. Niessen, “Supersymmetric top and bottom squark production at hadron colliders,” *JHEP* **08** (2010) 098, [arXiv:1006.4771 \[hep-ph\]](#).
- [62] W. Beenakker, S. Brening, M. n. Kramer, A. Kulesza, E. Laenen, L. Motyka, and I. Niessen, “Squark and Gluino Hadroproduction,” *Int. J. Mod. Phys.* **A26** (2011) 2637–2664, [arXiv:1105.1110 \[hep-ph\]](#).
- [63] W. Beenakker, C. Borschensky, M. Krämer, A. Kulesza, E. Laenen, S. Marzani, and J. Rojo, “NLO+NLL squark and gluino production cross-sections with threshold-improved parton distributions,” *Eur. Phys. J.* **C76** no. 2, (2016) 53, [arXiv:1510.00375 \[hep-ph\]](#).
- [64] **ATLAS** Collaboration, G. Aad *et al.*, “Search for direct production of charginos and neutralinos in events with three leptons and missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector,” *JHEP* **04** (2014) 169, [arXiv:1402.7029 \[hep-ex\]](#).
- [65] **ATLAS** Collaboration, G. Aad *et al.*, “Search for direct production of charginos, neutralinos and sleptons in final states with two leptons and missing transverse momentum in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector,” *JHEP* **05** (2014) 071, [arXiv:1403.5294 \[hep-ex\]](#).

- [66] **ATLAS** Collaboration, G. Aad *et al.*, “Search for supersymmetry in events with four or more leptons in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector,” *Phys. Rev.* **D90** no. 5, (2014) 052001, [arXiv:1405.5086 \[hep-ex\]](#).
- [67] **ATLAS** Collaboration, G. Aad *et al.*, “Search for direct third-generation squark pair production in final states with missing transverse momentum and two b -jets in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector,” *JHEP* **10** (2013) 189, [arXiv:1308.2631 \[hep-ex\]](#).
- [68] **ATLAS** Collaboration, G. Aad *et al.*, “Search for direct top-squark pair production in final states with two leptons in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector,” *JHEP* **06** (2014) 124, [arXiv:1403.4853 \[hep-ex\]](#).
- [69] **ATLAS** Collaboration, G. Aad *et al.*, “Search for supersymmetry at $\sqrt{s}=8$ TeV in final states with jets and two same-sign leptons or three leptons with the ATLAS detector,” *JHEP* **06** (2014) 035, [arXiv:1404.2500 \[hep-ex\]](#).
- [70] **ATLAS** Collaboration, G. Aad *et al.*, “Search for squarks and gluinos with the ATLAS detector in final states with jets and missing transverse momentum using $\sqrt{s} = 8$ TeV proton–proton collision data,” *JHEP* **09** (2014) 176, [arXiv:1405.7875 \[hep-ex\]](#).
- [71] J. Cao, L. Shang, J. M. Yang, and Y. Zhang, “Explanation of the ATLAS Z-Peaked Excess in the NMSSM,” *JHEP* **06** (2015) 152, [arXiv:1504.07869 \[hep-ph\]](#).
- [72] **ATLAS** Collaboration, G. Aad *et al.*, “Search for top squark pair production in final states with one isolated lepton, jets, and missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector,” *JHEP* **11** (2014) 118, [arXiv:1407.0583 \[hep-ex\]](#).
- [73] **ATLAS** Collaboration, G. Aad *et al.*, “Search for pair-produced third-generation squarks decaying via charm quarks or in compressed supersymmetric scenarios in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector,” *Phys. Rev.* **D90** no. 5, (2014) 052008, [arXiv:1407.0608 \[hep-ex\]](#).
- [74] **ATLAS** Collaboration, G. Aad *et al.*, “Search for new phenomena in final states with an energetic jet and large missing transverse momentum in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector,” *Eur. Phys. J.* **C75** no. 7, (2015) 299, [arXiv:1502.01518 \[hep-ex\]](#). [Erratum: *Eur. Phys. J.* C75,no.9,408(2015)].
- [75] **ATLAS** Collaboration, G. Aad *et al.*, “Search for supersymmetry in events containing a same-flavour opposite-sign dilepton pair, jets, and large missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector,” *Eur. Phys. J.* **C75** no. 7, (2015) 318, [arXiv:1503.03290 \[hep-ex\]](#). [Erratum: *Eur. Phys. J.* C75,no.10,463(2015)].
- [76] J. Cao, L. Shang, J. M. Yang, and Y. Zhang, “Explanation of the ATLAS Z-Peaked Excess in the NMSSM,” *JHEP* **06** (2015) 152, [arXiv:1504.07869 \[hep-ph\]](#).
- [77] **ATLAS** Collaboration, G. Aad *et al.*, “Search for gluinos in events with an isolated lepton, jets and missing transverse momentum at $\sqrt{s} = 13$ TeV with the ATLAS detector,” [arXiv:1605.04285 \[hep-ex\]](#).
- [78] **ATLAS** Collaboration, G. Aad *et al.*, “Search for supersymmetry at $\sqrt{s} = 13$ TeV in final states with jets and two same-sign leptons or three leptons with the ATLAS detector,” *Eur. Phys. J.* **C76** no. 5, (2016) 259, [arXiv:1602.09058 \[hep-ex\]](#).

- [79] **ATLAS** Collaboration, M. Aaboud et al., “Search for squarks and gluinos in final states with jets and missing transverse momentum at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Eur. Phys. J.* **C76** no. 7, (2016) 392, [arXiv:1605.03814 \[hep-ex\]](https://arxiv.org/abs/1605.03814).
- [80] “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,” Tech. Rep. ATLAS-CONF-2015-082, CERN, Geneva, Dec, 2015.
<https://cds.cern.ch/record/2114854>.
- [81] **ATLAS** Collaboration, M. Aaboud et al., “Search for new phenomena in final states with an energetic jet and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector,” [arXiv:1604.07773 \[hep-ex\]](https://arxiv.org/abs/1604.07773).
- [82] **CMS** Collaboration, V. Khachatryan et al., “Search for new physics in final states with two opposite-sign, same-flavor leptons, jets, and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV,” Submitted to: *JHEP* (2016) , [arXiv:1607.00915 \[hep-ex\]](https://arxiv.org/abs/1607.00915).
- [83] “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,” Tech. Rep. ATLAS-CONF-2016-013, CERN, Geneva, Mar, 2016.
[http://cds.cern.ch/record/2140998](https://cds.cern.ch/record/2140998).
- [84] **ATLAS** Collaboration, G. Aad et al., “Search for pair production of gluinos decaying via stop and sbottom in events with b -jets and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Phys. Rev.* **D94** no. 3, (2016) 032003, [arXiv:1605.09318 \[hep-ex\]](https://arxiv.org/abs/1605.09318).
- [85] D. Kriesel, *Ein kleiner Überblick über Neuronale Netze*. 2007.
`erh\unhbox\vvoidb@x\bgroup\accent127a\penalty\OM\hskip\z@skip\egroup\ltlichauf`<http://www.dkriesel.com>.
- [86] R. Schmidt, F. Lang, and M. Heckmann, *Physiologie des Menschen: mit Pathophysiologie*. Heidelberg: Springer., 2011.
- [87] N. Menche, *Biologie Anatomie Physiologie*. München: Urban & Fischer/ Elsevier GmbH, 2012.
- [88] W. Pschyrembel, *Klinisches Wörterbuch*. Berlin: De Gruyter., 2014.
- [89] “Neurons.” http://readingroom.mindspec.org/?page_id=8852.
- [90] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.
- [91] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR abs/1412.6980* (2014) . <http://arxiv.org/abs/1412.6980>.
- [92] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on

heterogeneous systems,” 2015. <http://tensorflow.org/>. Software available from tensorflow.org.

- [93] http://sebastianraschka.com/Articles/2014_about_feature_scaling.html.
- [94] e. a. g. Aad, “Search for squarks and gluinos with the atlas detector in final states with jets and missing transverse momentum using 8 tev proton-proton collision data,” *JHEP JHEP*, **176** (2014) **2014** no. 9, (2014) 1–52.
[http://dx.doi.org/10.1007/JHEP09\(2014\)176](http://dx.doi.org/10.1007/JHEP09(2014)176).
- [95] J. Neyman and E. Pearson, “On the problem of the most efficient tests of statistical hypotheses,” *Philosophical Transactions of the Royal Society of London* **231** (1933) 289–337.
- [96] S. S. Wilks, “The large-sample distribution of the likelihood ratio for testing composite hypotheses,” *Ann. Math. Statist.* **9** no. 1, (03, 1938) 60–62.
<http://dx.doi.org/10.1214/aoms/1177732360>.
- [97] “<https://cp3.irmp.ucl.ac.be/projects/madgraph/wiki/matching>.”