Determination of Supersymmetric Parameters with Neural Networks at the Large Hadron Collider

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Nicki Bornhauser
aus
Bonn-Beuel

Bonn, 29.07.2013

Dieser Forschungsbericht wurde als Dissertation von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Bonn angenommen und ist auf dem Hochschulschriftenserver der ULB Bonn `http://hss.ulb.uni-bonn.de/diss_online` elektronisch publiziert.

1. Gutachter: Prof. Dr. Manuel Drees
2. Gutachter: Prof. Dr. Herbert Dreiner

Tag der Promotion: 31.10.2013
Erscheinungsjahr: 2013

Danksagung

Ich möchte mich bei meinen Eltern und meinen drei Brüdern sowie ihren Liebsten dafür bedanken, dass sie immer, wenn nötig, mit Rat und Tat zur Seite stehen und zur Freude in meinem Leben beitragen. Insbesondere möchte ich mich auch bei meiner Freundin Kayleigh bedanken, die ein wunderbarer Mensch ist und so viel Glück in mein Leben bringt.

Außerdem möchte ich mich bei meinem Betreuer Prof. Dr. Manuel Drees bedanken, durch dessen hervorragende Betreuung ich in den letzten Jahren viel dazu lernen konnte. Hierbei möchte ich auch Prof. Dr. Herbi Dreiner für die Übernahme der Zweitkorrektur meiner Arbeit danken. Weiterhin auch einen Dank an Prof. Dr. Klaus Desch sowie Priv.–Doz. Dr. Wolfgang Koch, die sich als fachnahes und fachfremdes Mitglied meiner Promotionskommission zur Verfügung gestellt haben.

Ich möchte mich bei meinen Kollegen für die ein oder andere interessante Diskussion sowie die schöne Arbeitsatmosphäre bedanken. Hier auch einen Dank an alle Mitglieder des Physikalischen Instituts, insbesondere an Dagmar, Petra, Patricia und Andreas, die mich bei dem aufgetretenden Drumherum der Arbeit unterstützt haben. Weiterhin möchte ich meinen Freunden danken, die mein Leben bereichern.

Zuletzt möchte mich noch bei der Bonn–Cologne Graduate School of Physics and Astronomy sowie der Universität Bonn für die finanzielle Unterstützung in den letzten Jahren bedanken. Und zuallerletzt natürlich auch einen Dank an all diejenigen, die ich hier zu nennen unglücklicherweise vergessen habe.

Abstract

The LHC is running and in the near future potentially some signs of new physics are measured. In this thesis it is assumed that the underlying theory of such a signal would be identified and that it is some kind of minimal supersymmetric extension of the Standard Model. Generally, the mapping from the measurable observables onto the parameter values of the supersymmetric theory is unknown. Instead, only the opposite direction is known, i.e. for fixed parameters the measurable observables can be computed with some uncertainties. In this thesis, the ability of artifical neural networks to determine this unknown function is demonstrated. At the end of a training process, the created networks are capable to calculate the parameter values with errors for an existing measurement. To do so, at first a set of mostly counting observables is introduced. In the following, the usefulness of these observables for the determination of supersymmetric parameters is checked. This is done by applying them on 283 pairs of parameter sets of a MSSM with 15 parameters. These pairs were found to be indistinguishable at the LHC by another study, even without the consideration of SM background. It can be shown that 260 of these pairs can be discriminated using the introduced observables. Without systematic errors even all pairs can be distinguished. Also with the consideration of SM background still most pairs can be disentangled (282 without and 237 with systematic errors). This result indicates the usefulness of the observables for the direct parameter determination. The performance of neural networks is investigated for four different parameter regions of the CMSSM. With the right set of observables, the neural network approach generally could also be used for any other (non–supersymmetric) theory. In each region, a reference point with around 1,000 events after cuts should be determined in the context of a LHC with a center of mass energy of 14 TeV and an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The parameters m_0 and $m_{1/2}$ can be determined relatively well down to errors of around 4.5 % and 1 %, respectively. Increasing the integrated luminosity to $500\,\mathrm{fb}^{-1}$ allows also a quite accurate determination of the other two continuous parameters $\tan\beta$ and A_0. The parameter $\tan\beta$ has relative errors as small as 4 %, and the estimated standard deviations for A_0 are roughly between 25 and 35 % of the true value of m_0.

The content of this thesis was mostly published in the following papers:

- N. Bornhauser, M. Drees, *Phys. Rev.* **D86** (2012) 015025 [arXiv:1205.6080 [hep-ph]]
- N. Bornhauser, M. Drees, arXiv:1307.3383 [hep-ph]

Contents

1. Introduction

This thesis describes the determination of supersymmetric parameters at the Large Hadron Collider (LHC) using artificial neural networks. The main content of this thesis is published in references [1] and [2].

In the beginning of 2010, the LHC started running and since then an enormous number of proton–proton–collisons were recorded by the corresponding experiments. Both multi–purpose detectors ATLAS and CMS saved each an integrated luminosity of around $5\,\mathrm{fb}^{-1}$ for a center of mass energy of 7 TeV and of around $23\,\mathrm{fb}^{-1}$ for a center of mass energy of 8 TeV. Recently, a longer shutdown of the LHC has started with the main goal of increasing the center of mass energy to (at least) 13 TeV close to the maximum design energy of 14 TeV. Presumably, the LHC will start running again at the end of 2014. The so far completed analyses of the collected data did not find any hints for the realization of supersymmetry within nature [3]. Potentially, this will change after the restart of the LHC with a higher center of mass energy, since the increase of energy expands the reachable supersymmetric parameter space.

In this thesis it is assumed that new physics is found at the LHC and in particular that the underlying theory can be identified as a minimal supersymmetric extension of the Standard Model (MSSM). The realization of the MSSM around the energy scale of the LHC is motivated by the possible cancellation of quadratic divergences in the radiative corrections of the Higgs boson mass [4]. The MSSM could also supply a dark matter candidate and it would predict the unification of gauge couplings at a high energy scale. The determination of the values of the supersymmetric parameters is a non–trivial task, even if the underlying theory is known. In general, once the values of the theory parameters are set, the corresponding measurement signature at the LHC can be simulated (with some uncertainty). But in contrast to that, the inverse function going from the measurable observables to the values of the parameters is in general unknown. Additionally, it even could be possible that the at the LHC available observables are not sufficient to discriminate between the measurement signatures produced by different choices of the supersymmetric parameters. This impossibility for the unique determination of the parameter values is called "LHC inverse problem".

The LHC inverse problem was studied in [5] for a MSSM with 15 parameters in the context of the LHC with a center of mass energy of 14 TeV and an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The authors identified 283 parameter set pairs which could not be distinguished with a 95 % confidence level within their comparison method. They used with 1,808 a very large number of mostly kinematical observables.

A successful determination of supersymmetric parameters relies on the right selection of observables. In the first part of this thesis, 84 observables are introduced which are in the following used for a new comparison of the 283 parameter set pairs from reference [5]. Contrary to [5], the 84 observables consist mostly of counting observables. It is shown that indeed a large fraction of the parameter set pairs can be distinguished using the described observables. This successful discrimination indicates that the observables are in particular also useful for the direct determination of supersymmetric parameters from a measurement (and not only useful for the discrimination of different supersymmetric signatures). Furthermore, in contrast to reference [5] also Standard Model background is included in the comparison of the parameter sets. This should verify the usefulness of the observables also in a more realistic situation.

In the second part of this thesis, the established observables are used for the determination of supersymmetric parameters from a measurement with the help of neural networks. This is not the first time that artificial neural networks are used within high energy physics. They were already used more than 25 years ago [6], although with quite a different set–up compared to the one presented in this thesis. In the beginning, they were deployed for tasks like the reconstruction of single tracks from hit patterns in wire chambers. In the meanwhile, neural networks are applied for a wide variety of purposes as e.g. in the optimization of experimental searches for superparticles [7] or also in the parametrization of parton distribution functions [8]. However, I am not aware of the previous implementation of neural networks for the parameter determination of a supersymmetric (or any other supposedly underlying physics) theory at the LHC.

The in this thesis considered theory is the constrained MSSM (CMSSM) with four continuous parameters and a sign choice. The low number of free parameters has the advantage that the creation of the neural networks can be done with a manageable amount of computations, respecting the available computing resources. On the other hand, the introduced method can also easily be used for supersymmetric theories with more parameters. In practice this means that a higher number of neural networks has to be created, since here for each theory parameter a single network is used. In particular, the introduced method can also be applied on any other (non–supersymmetric) theory, as long as the measured observables are adapted appropriately.

The introduced observables are used as input values for the neural networks and the output values are the parameters of the considered theory. The key of a successful determination of theory parameters using neural networks is the understanding of the statistical fluctuations of the observables, i.e. the knowledge of their variances and covariances. They can be used to improve the performance of the created neural networks. A priori, a neural network knows nothing about the errors of the input observables, while every measured observable obviously has a certain uncertainty. Therefore a significant performance improvement of a neural network is reached by "teaching" it the importance of the single input observables. Observables with generally bigger errors are less important than observables with smaller errors. This teaching can be done using the covariance matrix of the observables and is shown in this thesis. Furthermore, a priori a neural network gives an output value without a corresponding error, i.e. it is not known how well the value is determined. With the knowledge of the variances and covariances of the observables it is possible to determine the errors and correlations of the found parameters. Two independent methods leading to the statistically same results are described. The performance of a neural network is compared to a relatively simple χ^2–minimization, leading to the insight that the consideration of the statistical fluctuations is very important for a successful parameter determination.

This thesis is organized in detail in the following way. At first, Chapter 2 gives a basic understanding of the difficulty of supersymmetric parameter determination. It describes the general properties of supersymmetric events and summarizes the results of [5] concerning the LHC inverse problem. In Chapter 3, the considered MSSM with 15 parameters from reference [5] and the CMSSM are shortly described. Furthermore, the general implementation of the generation of supersymmetric events is explained. In the following Chapters 4 and 5, the used observables with their corresponding covariance matrix are introduced, their statistical behavior verified, and they are finally applied on the indistinguishable parameter set pairs from [5]. In the second part of this thesis, beginning with Chapter 6, the general construction of a neural network is explained and the adaption for the CMSSM parameter determination is described. Chapter 7 presents the results of the parameter determination using neural networks and compares the neural network performance to a relatively simple χ^2–minimization. Finally, Chapter 8 gives a summary about the results of this thesis and an outlook.

2. Difficulty of Parameter Determination

In this thesis constrained versions of the MSSM are considered. A general introduction to the MSSM can be found in [4]. The unconstrained MSSM has 105 free parameters next to the 19 Standard Model parameters. But such a high number of free parameters is for a study of the properties of supersymmetric events not feasible. Therefore, the number of free parameters can be reduced for the sake of practicabilty by making further sensible assumptions. Note that most of the 105 parameters arise from the soft supersymmetry breaking, which is not a priori specified and therefore written in a general way. An exact specification of the soft supersymmetry breaking mechanism allows for example a reduction of the number of parameters. The constrained versions considered in this thesis, which are shortly introduced in Sections 3.1 and 3.2, have only 15 and 4 (plus a discrete sign choice) free parameters, respectively.

Even for these reduced numbers of free parameters, the mapping between the measurable observables and the supersymmetric model parameters is in general unknown. For example, for the CMSSM this problem arises from the fact that model parameters are defined at a high energy scale (compared to the energy of the LHC collisions) and non–trivially connected to the superparticle spectrum and therefore to the measurable observables. On the other hand, setting the parameter values of a supersymmetric model allows the simulation of the experimental signature of the specific parameter choice. Therefore, the usual way of determining the parameters of an existing measurement is the comparison of this measurement to the simulations, resulting from different parameter choices of the supersymmetric model which is assumed to describe nature. The parameter values of the measurement can be identified (within certain uncertainties), if the experimental signature coincides with a particular simulation. This procedure assumes that different parameter choices lead to unique experimental signatures (see LHC inverse problem below). Note that if the superparticle spectrum belonging to the measurement can be identified from the measured (kinematical) observables, then it is not necessary to generate events with an event generator anymore. Instead, it is sufficient only to compute the superparticle spectrums for different parameter choices (and not additionally to generate events) and compare them with the measurement. This normally takes much less computation time than the generation of events.

In the following, at first the general properties of the supersymmetric events at the LHC are considered. Afterwards results of a study [5] concerning the LHC inverse problem are presented, whereas the LHC inverse problem describes the issue of a potentially impossible discrimination between experimental signatures from different model parameter choices.

2.1. Supersymmetric Events

In this section, general properties of the experimental signatures at the LHC from the considered supersymmetric models are discussed. A more detailed consideration of supersymmetric signals can be found in [4]. The considered supersymmetric models fulfill the so–called R–parity conservation, having the consequence that always an even number of supersymmetric particles (superparticles) participates in an interaction, since all Standard Model (SM) particles have positive R–parity and all superparticles negative R–parity. In particular this means, that only even numbers of superparticles, usually two, are produced in proton–proton–collisions at the LHC. Furthermore, the decay of such a produced superparticle will always lead to the lightest supersymmetric particle (LSP), which is then stable because of R–parity conservation. The following discussion concerns only the situation that the LSP is the lightest neutralino, i.e. an electrically neutral, colorless, only weakly interacting superparticle. This is the case for almost all parameter sets considered in this thesis.

The collision of two protons at the LHC then leads to the production of two superparticles, mostly gluinos and squarks, which then each decay in longer decay chains to the neutralino as being the LSP. Similar to neutrinos, the produced neutralinos would not be measured within the detector. Considering the high energy necessary for the production of the (in comparison to most SM particles) heavy superparticles, this would lead to a relatively high amount of energy leaving the detector unmeasured. Since not the whole protons, but instead single partons of the colliding protons are responsible for the production of the superparticles, the energy of the initial state is not known and therefore the whole missing energy can not be specified. But in contrast to the momentum of the partons longitudinal to the beam line, the transverse momentum of the interacting partons is known to vanish in the initial state. Therefore, it is possible to measure the missing *transverse* energy (momentum) of the event. The missing transverse energy of the event is then determined by the amount of transverse momentum, which is missing to set the summed transverse momenta of all measured final state particles to zero. Note that next to the neutralinos, of course, also SM neutrinos can contribute to this missing energy.

The missing transverse energy is the main divider between supersymmetric events and events resulting from the production of SM particles. Of course, also in SM events missing energy occurs from the detector escape of neutrinos (which are produced in the decay of W–, Z–bosons or of heavy quarks) or from mismeasurements, but in general the missing transverse energy resulting from the production of superparticles is much bigger. In this sense, the resulting supersymmetric final states can be broadly classified as having two or more jets and zero or more charged leptons (electrons, muons and the corresponding antiparticles) resulting from the long decay chains of the produced superparticles, and additionally having a large amount of missing tranverse energy [4]. The event cuts, which are applied on the considered supersymmetric events in this thesis, are listed in Appendix B. The application of such cuts should reduce the SM background, which is (because of the generally lower particle masses) much bigger than a supersymmetric signal, and therefore allow the investigation of the properties of the supersymmetric events. Depending on the values of the supersymmetric model parameters, which lead to different masses and branching ratios of the superparticles, the given cuts could be optimized to improve the signal–background–ratio. However, the chosen cuts, described in Appendix B, deliver an overall satisfying signal–background–ratio.

Note that next to the (hard) parton interaction producing a pair of superparticles also the other (spectator) partons of the two colliding protons interact (softly). Additionally, not only this "interesting" proton–proton–collision happens, but also multiple (normally uninteresting) other proton–proton–collisions happen at the same crossing of the two proton beam bunches. Therefore, the events at the LHC are kind of messy (compared to events at a lepton collider). This additionally supports the need of proper cuts to filter out the important event properties. Furthermore, a well adjusted trigger is needed to record the potentially interesting (supersymmetric) events out of the bulk of measured events.

Overall, generally it is not easy to determine the properties like masses and branching ratios of the superparticles from the measurement. On the one hand, supersymmetric events have to be filtered out of the enormous amount of SM background. On the other hand, the produced superparticles decay directly in potentially complicated decay chains and the only superparticle left at the end of such a decay chain, namely the neutralino being the LSP, is not directly measurable. Furthermore, a priori it is even not clear, if different parameter choices for a considered supersymmetric model actually lead to different at the LHC measurable experimental signatures. This potential problem is discussed in the following section.

2.2. LHC Inverse Problem

This section describes shortly the so–called LHC inverse problem and the results of Arkani–Hamed et al. [5] investigating this problem for a MSSM with 15 parameters in the context of a LHC with a center of mass energy of 14 TeV and an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The content of this section is published in [1].

The LHC inverse problem concerns the unique determination of parameters which belong to an existing measurement. It is assumed that the underlying physics theory is known. The choice of the values of the theory parameters determines the signature of the measurement. The inverse problem is to find the values of the parameters from the values of the measured observables. The LHC inverse problem refers to the situation that the values of the theory parameters cannot be determined uniquely from the at the LHC available observables, since at least two different parameter choices would lead to the (statistically) same values for all observables. In particular, this refers to parameter choices from quite different regions of the parameter space and not to choices with very small differences. Such close choices would naturally have indistinguishable measurement signatures within the experimental errors.

Arkani–Hamed et al. [5] simulated 43,026 parameter sets using a MSSM with 15 parameters. The model and the considered parameter space is described in Section 3.1. They compared the parameter sets to each other using 1,808 mostly kinematical observables, including less than ten percent counting observables. 283 of the compared pairs could not be discriminated with a 95 % confidence level within their comparison method. They called these indistinguishable parameter set pairs “degenerate pairs”.

In reference [5] a large number of observables was used. Furthermore, the simulated indistinguishable parameter sets have with on average around 25,000 events a relatively high number of events after cuts. Therefore, these 283 pairs, which cannot be uniquely connected to a specific choice of the parameters within their comparison method, could imply that the LHC inverse problem is actually real (under the assumption that nature follows a MSSM). Especially, meaning that this could also happen for other realizations of the MSSM. On the other hand, the analysis of Arkani–Hamed et al. has some weaknesses. At first, they do not consider initial state radiation as well as interactions between spectator partons (these are the partons which do not participate in the primary hard interaction) within their generated events. Both effects tend to make the measurement signatures of different parameter sets more similar, since they do not depend strongly on the final state of the event. This could actually mean that the LHC inverse problem is even more severe, because more parameter set pairs potentially could not be distinguished.

Another problem is that in [5] this high number of observables was considered, without taking the corresponding correlations into account. This is caused by the simple fact that the identification of the correlations of such a large number of observables, especially of kinematical observables, is rather difficult. As a consequence, the statistical behavior of the in reference [5] constructed quantity $(\Delta S_{AB})^2$, which should express the difference between two data sets, is a priori not clear. The degenerate pairs were identified with the requirement $(\Delta S_{AB})^2 < 0.285$. This requirement was deduced from the comparison of parameter sets with themselves, meaning that a specific parameter set was simulated with different statistics and the resulting data sets were compared to each other. In this self–comparison Arkani–Hamed et al. found, that 5 % of the compared data sets had values $(\Delta S_{AB})^2 > 0.285$. They conclude from that, that compared parameter sets with $(\Delta S_{AB})^2 > 0.285$ are distinguishable with a 95 % confidence level. Since the statistical properties of $(\Delta S_{AB})^2$ are questionable, the same applies to the identification of the degenerate pairs. More details on that can be found in [1].

In the following, the 283 degenerate pairs are reconsidered. Potentially, it is actually possible to discriminate between a large fraction of these pairs, using a different set of observables and in particular a statistically clean comparison method. Such a successful discrimination would indeed mitigate the

LHC inverse problem and suggest that with the right selection of possible LHC observables an unique determination of most supersymmetric theory parameters within certain errors would be manageable. Furthermore, the used observables would probably not only be useful for the discrimination of the degenerate pairs, but also be suitable for the direct determination of supersymmetric parameters from a LHC measurement.

As mentioned, it is necessary to understand the correlations between the considered observables to achieve a comparison method with a clear statistical behavior. Therefore, we consider mostly counting instead of kinematical observables, as it was done in [5]. It is much easier to identify the correlations between counting observables refering to the same events as compared to kinematical observables. The used observables and their variances and covariances are introduced in Chapter 4. In the following chapter, the general simulation of the parameter sets is described.

3. The Simulation

In this chapter, the two supersymmetric models which are considered in this thesis are shortly introduced and the general simulation of the supersymmetric events is explained. The parameter sets contained in the degenerate pairs from [5] were simulated with the first described model (MSSM with 15 parameters), whereas the second model (CMSSM) is used for the determination of parameters with neural networks. A general introduction into supersymmetry can be found in [4].

3.1. Parameter Sets from Degenerate Pairs

This section describes shortly the MSSM with 15 free parameters used in [5] and it is also contained in a similar form in [1]. The free parameters are the three gaugino masses M_1 (bino), M_2 (wino) and M_3 (gluino), the four independent slepton masses $m_{\tilde{e}_L} = m_{\tilde{\mu}_L}$, $m_{\tilde{e}_R} = m_{\tilde{\mu}_R}$, $m_{\tilde{\tau}_L}$ and $m_{\tilde{\tau}_R}$, the six independent squark masses $m_{\tilde{q}_{1L}} = m_{\tilde{q}_{2L}}$, $m_{\tilde{q}_{3L}}$, $m_{\tilde{u}_R} = m_{\tilde{c}_R}$, $m_{\tilde{t}_R}$, $m_{\tilde{d}_R} = m_{\tilde{s}_R}$ and $m_{\tilde{b}_R}$, the Higgs(ino) mass parameter μ, and finally the ratio $\tan\beta$ of the vacuum expectation values of the two Higgs doublets. The first and second generation sfermions masses are set equal. There are four additional parameters set to fixed values, namely the trilinear scalar couplings $A_t = A_b = A_\tau = 800\,\mathrm{GeV}$ and the pseudoscalar Higgs pole mass $m_A = 850\,\mathrm{GeV}$.

The parameter sets simulated in [5] are randomly chosen from certain parameter ranges, using flat probability distributions. The range for the parameters M_1, M_2 and μ and the four slepton masses goes from 100 GeV to 1 TeV. The gluino mass and the six squark masses are picked from the range between 600 GeV and 1 TeV. The last parameter $\tan\beta$ lies in the interval from 2 to 50. Furthermore, the following condition is required for the parameters:

$$m^{\max}_{\mathrm{slepton}} < m^{\max}_{\mathrm{ewino}} + 50\,\mathrm{GeV} < m^{\max}_{\mathrm{color}} + 100\,\mathrm{GeV} \tag{3.1}$$

The maximum slepton soft mass is denoted as $m^{\max}_{\mathrm{slepton}}$, the maximum of the three parameters M_1, M_2 and μ is called $m^{\max}_{\mathrm{ewino}}$, and $m^{\max}_{\mathrm{color}}$ is the maximum soft mass or mass parameter of any color–charged superparticle.

Note that probably most of the parameter sets simulated in [5] as described above are excluded by published analyses of the LHC data [3]. The main purpose of using the parameter sets from [5] is to reconsider the pairs of parameter sets, which Arkani–Hamed et al. could not distinguish within their comparison method using 1,808 observables. This allows a direct comparison of the results.

3.2. Constrained Minimal Supersymmetric Standard Model

In this section, the constrained minimal supersymmetric extension of the Standard Model (CMSSM, also often called mSUGRA) is shortly described. A more profound description can be found in [4]. The CMSSM has four continuous parameters, the common scalar mass m_0, the common gaugino mass $m_{1/2}$, the ratio of the Higgs vacuum expectation values $\tan\beta$, and the common trilinear coupling A_0, as well as one discrete parameter, the sign of the Higgsino mass parameter μ. Common refers here to the energy scale of a Grand Unified Theory (GUT) with $M_U \simeq 2 \cdot 10^{16}\,\mathrm{GeV}$, at which the $U(1)$, $SU(2)_L$ and $SU(3)_C$ gauge coupling strengths have the same value in the MSSM. The MSSM should stay valid up to this scale. In the CMSSM the scalar particles, i.e. the squarks, sleptons and Higgs bosons, have the common mass m_0 at the GUT–scale, the gaugino masses unify to $m_{1/2}$, and the trilinear couplings (sfermion-sfermion-Higgs) become A_0. In contrast, the parameter $\tan\beta$ is given at the electroweak scale.

The spontaneous breaking of supersymmetry is mediated (from a "hidden" sector to the observable sector, which includes the MSSM) by graviational strength interactions. Imposing the universal soft supersymmetry breaking parameters m_0, $m_{1/2}$, A_0 and $\tan\beta$ (equivalent for using B_0) and fixing the sign of μ then determines the whole superparticle mass spectrum. This connection between the soft supersymmetry breaking parameters at the GUT–scale and the MSSM parameters at the (lower) scale of the superparticle masses is thereby achieved with the renormalization group evolution equations.

3.3. Event Generation

This section describes the generation of the supersymmetric events and its content is included in references [1] and [2]. The events for the considered parameter sets from the previously described supersymmetric models are generated with multiple programs. At first, the superparticle and Higgs boson spectrum of a particular parameter set is calculated with the program SOFTSUSY [9]. In the following, the program SUSY–HIT [10] is used to compute the branching ratios of the kinematically allowed decays. Finally, events are generated with the program Herwig++ [11] in the context of the LHC with a center of mass energy of 14 TeV. CTEQ6.6 [12] is used for the parton distribution functions and the cross–sections are calculated in leading order in QCD.

The considered measurements for the parameter set discrimination in Section 5.2 and the determination of the CMSSM parameters via neural networks in Section 7.1 have each an integrated luminosity of $10\,\mathrm{fb}^{-1}$. Parameter sets are also simulated for higher luminosities up to $500\,\mathrm{fb}^{-1}$. For all these simulations, at first, 10,000 events for the specific parameter set are created in Herwig++ to determine the total cross–section of the supersymmetric events. In the following, this cross–section is used to generate the appropriate number of events for the chosen integrated luminosity.

Furthermore, interactions between spectator partons, i.e. the "underlying event", as well as initial state radiation are included using the default values of the corresponding parameters in Herwig++. There is no detector simulation realized. However, the imposed transverse momentum and pseudorapidity cuts on leptons and jets, as described in Appendix A, should be consistent with the main detector acceptances. Jets are determined using the program FastJet [13]. Hadronically decaying τ–leptons as well as b–jets, which fulfill the existing cuts, are only identified with a probability of 50 %. The b–jets which are not tagged are counted as normal jets. False positive tags are neither included for τ–leptons nor for b–jets. Overall, the absence of a detailed detector simulation should not distort the results for the discrimation of parameters sets as well as the direct parameter determination significantly, because as described in Chapter 4 mostly counting observables are employed. Counting observables are in contrast to kinematical observables relatively insensitive to detector resolution effects.

4. Event Characterization

In this chapter, the observables which are recorded to characterize important properties of the measured (simulated) events as well as the corresponding covariance matrix are introduced. The selected observables (and the covariance matrix) are published in [1]. In the introduction of [1], there is also an overview of the literature about supersymmetric parameter determination with different kinds of observables. Furthermore, a slightly modified selection of the observables is used in [2]. The first selection is used for the discrimination of the parameter set pairs described in Chapter 5, while the second choice is used for the parameter determination with neural networks as shown in Chapters 6 and 7. Both versions are presented and their differences are pointed out.

4.1. Observables

Assuming an underlying theory, the choice of LHC observables is determined by the purpose to find the values of the theory parameters belonging to a particular measurement. In this context, an unique parameter determination can only work if different parameter choices lead to different measurement signatures. Even if parameter changes cause changes in event affecting properties like masses, production cross–sections or decay branching ratios of particles, such changes do not necessarily show in measurable LHC observables.

Provided that different choices of parameters are recognizable in possible LHC observables, the particular choice of observables is not directly obvious. The number of observables has to be sufficient to read all information needed to determine the theory parameters. Therefore, observables have to be identified, which show the changes of certain parameters. But in general it is probably not clear, if a specific observable is sensitive to parameter changes over the whole parameter space. Potentially, it is only relevant for certain parameter regions. Therefore, the number of observables should be higher than the number of parameters which should be determined. On the other hand, an additional observable is not automatically useful for the parameter determination. First of all, it has to show a significant parameter dependence to be potentially helpful. If this is true, an additional observable cannot be considered independently, but the correlations with the other observables have to be taken into account.

The consequences of a non–consideration of correlations can be illustrated in an example with two observables, as can be seen in Figure 4.1. If both observables are positively correlated, meaning that if the first observable increases also the second one tends to increase and the other way around, two different parameter choices can look more similar as well as more different than they actually are. To explain that we look at the data sets produced by two different parameter choices. The second parameter set should have somewhat higher expectation values for the observables than the first parameter set. Both observables follow a two–dimensional correlated Gaussian distribution, each around the expectation values. If for the first data set the first observable has by chance a high value out of the Gaussian distribution, then also the second observable tends to take a high value out of its distribution (lower left red cross in Figure 4.1). By chance both observables from the second data set with somewhat higher expectation values could take lower values from their distribution (upper right red cross in Figure 4.1). Without the knowledge that the second observable tends to have closer values for both data sets because of the first observable having closer values, these data sets would

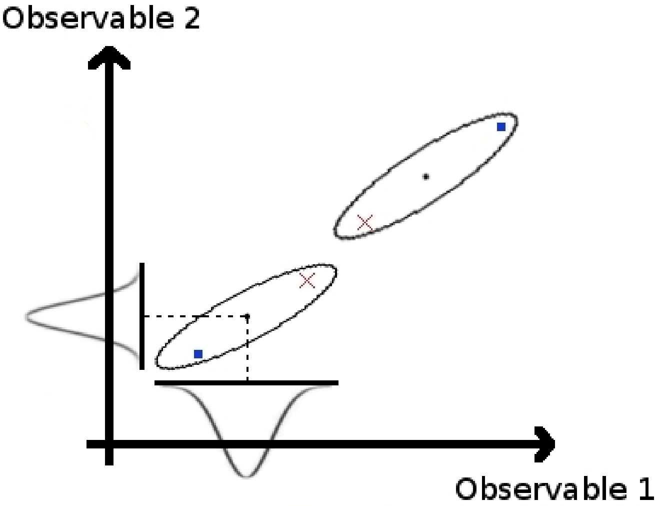

Figure 4.1.: Observables 1 and 2 are positively correlated, as can be seen in the tilting of the ellipses. The probability for the observables to take specific values follows a two–dimensional correlated Gaussian distribution around the expectation values (such a distribution is shown for parameter values in Figure 6.6). The ellipse marks value pairs with equal probability. The central dot within the lower left ellipse marks the expectation value of the observables for the first parameter set. The two lines next to the this ellipse show the possible observable value ranges (the one–dimensional Gaussian distributions without correlation are indicated). The red cross marks a high value pair for the observables. Without the present correlation, it would be less likely that both observables take such high values at the same time. If the observables for the second parameter set (ellipse on the upper right) take relatively low values (red cross) and the correlation is not taken into account, both data sets would look more similar than they actually are. For the blue squares both data sets would look more different than they actually are.

look more similar than they actually are, taking the correlation into account. In the same way, if the first data has lower and the second one has higher values for both observables (blue squares in Figure 4.1), then they would look more different. The same arguments can also be made for negatively correlated observables. The stronger the correlations are, the more is the statistical behavior distorted. As a result, a comparison of two different data sets can only be done in a statistical valid way, if the correlations between the observables are considered.

In the following, only observables with known correlations are used. To identify the correlations of multiple kinematical observables for the same events is challenging (since for example particle decays have to be considered, respecting energy and momentum conservation). In contrast, it is much easier to create statistically independent counting observables for the same events, because e.g. the number of jets has a priori nothing to do with the number of hadronically decaying τ–leptons. As a consequence, we define counting observables for distinct event classes and only one kinematical observable for each class. This permits the clear identification of the correlations.

In total 84 observables are considered. The first observable is the total number of events after cuts. The event cuts are described in detail in Appendix B. They depend on the number of charged leptons (meaning electrons, muons and tagged hadronically decaying taus) and should reduce Standard Model background. This is for example done by requiring a certain amount of missing transverse energy $\not{E}_T$, which in supersymmetric events originates from neutralinos (as being the lightest supersymmetric particle) escaping the detector. Only events which pass the selection cuts are considered in the following. For the parameter sets contained in the degenerate pairs, on average around 30 % of the generated events pass the defined cuts.

The events are split into twelve distinct classes which are categorized by the number, charge and flavor of the measured charged leptons. In this context only electrons and muons are considered, because τ–leptons decay within the detector and cannot be detected certainly. The electrons as well as muons have to fulfill cuts on the transverse momentum with $p_T > 10\,\text{GeV}$ and on the pseudorapidity with $|\eta| < 2.5$. Additionally they have to be isolated. The cuts are explained in more detail in Appendix A. Here is an overview of the twelf classes, which also appears in the same way in [1] and [2] ($l^\pm = e^\pm$ or $\mu^\pm$):

1. $0l$: Events with no charged leptons
2. $1l^-$: Events with exactly one charged lepton, with negative charge (in units of the proton charge)
3. $1l^+$: Events with exactly one charged lepton, with positive charge
4. $2l^-$: Events with exactly two charged leptons, with total charge -2
5. $2l^+$: Events with exactly two charged leptons, with total charge $+2$
6. $l_i^+ l_i^-$: Events with exactly two charged leptons, with opposite charge but the same flavor; i.e. e^-e^+ or $\mu^+\mu^-$
7. $l_i^+ l_{j;\ j\neq i}^-$: Events with exactly two charged leptons, with opposite charge and different flavor; i.e. $e^-\mu^+$ or $e^+\mu^-$
8. $l_i^- l_j^- l_j^+$: Events with exactly three charged leptons with total charge -1. There is an opposite–charged lepton pair with same flavor. For example $e^-\mu^-\mu^+$ or $e^-e^-e^+$
9. $l_i^+ l_j^+ l_j^-$: Events with exactly three charged leptons with total charge $+1$. There is an opposite–charged lepton pair with same flavor. For example $e^+\mu^-\mu^+$ or $e^+e^-e^+$
10. $l_i^- l_j^- l^\pm_{k;\ k\neq j,i \text{ for } +}$: Events with exactly three charged leptons with total negative charge, i.e. there are at least two negatively charged leptons. There is no opposite–charged lepton pair with same flavor. For example $e^-e^-\mu^+$ or $e^-e^-e^-$
11. $l_i^+ l_j^+ l^\pm_{k;\ k\neq j,i \text{ for } -}$: Events with exactly three charged leptons with total positive charge, i.e. there are at least two positively charged leptons. There is no opposite–charged lepton pair with same flavor. For example $e^+e^+\mu^-$ or $e^+e^+e^+$
12. $4l$: Events with four or more charged leptons

There is no difference made between electrons and muons. We defined these classes to capture the information of the events for the MSSM with 15 free parameters described in Section 3.1. In this model, the first and second generation sleptons are degenerate and therefore also lepton universality for the first and second generation is valid. As a consequence, no significant additional information is gained by separating between electrons and muons.

With two protons the LHC has an initial state with positive electric charge and as a consequence, in general not the same number of positively and negatively charged leptons should be produced. Therefore, the classes depend on the charge of the leptons. Furthermore, in classes with at least two leptons, a difference between opposite–charged lepton pairs with same and different flavor is made.

Pairs with same flavor can be created through the decay of a single neutralino $\tilde{\chi}_i^0 \to l^+ l^- \tilde{\chi}_j^0$ with $j < i$. On the other hand, all other pairs of charged leptons arise from the decay of two different (super)particles. Moreover, events with four or more leptons are probably relatively rare. Therefore these events are summarized within one class, because additional, mostly empty classes do not benefit (and as explained in Section 5.1 potentially even hurt) the event characterization.

For each of these twelve classes $c \in \{1, 2, \ldots, 12\}$ seven observables $O_{i,c},\ i \in \{1, 2, \ldots, 7\}$ are recorded. As mentioned before, the selection of observables differs for the discrimination of the degenerate pairs published in [1] and for the parameter determination with neural networks published in [2]. The letter "a" marks the observables which are used for the parameter set discrimination and the letter "b" marks the ones which are used for the neural networks. Observables 1, 5, 6 and 7 are used in both cases. The observables are listed in the following overview which is a combination of the lists included in [1] and [2]:

1. $O_{1,c} = n_c/N$: The number of events n_c contained in the given class c divided by the total number of events N, i.e. the fraction of all events contained in a given class

2a. $O_{2,c} = n_{c,\tau^-}/n_c$: The number of events in a given class c that contain at least one tagged hadronically decaying τ^- divided by the total number of events in this class

2b. $O_{2,c} = \langle \tau^- \rangle_c$: Average number of tagged hadronically decaying τ^- of all events within a given class c

3a. $O_{3,c} = n_{c,\tau^+}/n_c$: The number of events in a given class c that contain at least one tagged hadronically decaying τ^+ divided by the total number of events in this class

3b. $O_{3,c} = \langle \tau^+ \rangle_c$: Average number of tagged hadronically decaying τ^+ of all events within a given class c

4a. $O_{4,c} = n_{c,b}/n_c$: The number of events in a given class c that contain at least one tagged b–jet divided by the total number of events in this class

4b. $O_{4,c} = \langle b \rangle_c$: Average number of tagged b–jets of all events within a given class c

5. $O_{5,c} = \langle j \rangle_c$: Average number of non–$b$–jets of all events within a given class c

6. $O_{6,c} = \langle j^2 \rangle_c$: Average of the square of the number of non–b–jets* of all events within a given class c

7. $O_{7,c} = \langle H_T \rangle_c$: Average value of H_T of all events within a given class c, where H_T is the scalar sum of the transverse momenta of all hard objects, including the missing p_T

The observables 2 and 3 concern only τ–leptons which decay hadronically. Furthermore, they have to be isolated as well as satisfy $p_T > 20\,\text{GeV}$ and $|\eta| < 2.5$. Fulfilling these requirements, they are still only tagged with a 50 % probability in the simulations. The reconstruction of jets is done with the anti–k_T algorithm in FastJet [13]. The jets have to fulfill $p_T > 20\,\text{GeV}$ and $|\eta| < 4.8$. A jet which contains at least one (decay product of a) b–flavored hadron is identified as a b–jet, if a more stringent pseudorapidity cut with $|\eta| < 2.5$ is passed. In addition, similar to the τ–leptons, b–jets are only tagged with a 50 % probability. A b–jet which is not tagged is counted as a non–b–jet. Further details on the cuts are found in Appendix A.

*If event i in the given class contains $N_j^{(i)}$ non–b–jets, then $\langle j^2 \rangle_c = 1/n_c \sum_{i=1}^{n_c} (N_j^{(i)})^2$. The same footnote is included in [1] and [2].

In version “a” of the observables 2, 3 and 4 events are counted which contain at least one τ^-, τ^+ and b–jet, respectively. As a consequence, the observables have the same values, independently of how many τ–leptons or b–jets are included in the single events, as long as the same number of events each includes at least one of them. We used these observables for the parameter set discrimination, because we noticed that the number of tagged τ–leptons as well as b–jets was relatively low in the investigated degenerate pairs. Events containing more than one of those were very rare.[†] Therefore, the used observables gave adequate information about the events.

On the other hand, for the parameter determination different regions of the CMSSM parameter space are investigated. It is quite possible to have regions in which a higher number of τ–leptons or b–jets is produced within the events. Therefore, version “b” of the three observables records more information about the events by saving the average numbers of such tagged objects.

Supersymmetric events can contain higher numbers of jets. Therefore, with 5 and 6 there exist two observables to record information about the number distribution of non–b–jets. The last observable is the only kinematical information which is measured for the class events.

So far there are 85 observables. With the introduction of the covariance matrix in the following section it is shown that one observable can be dropped without losing any information about the events.

4.2. Covariance Matrix

The covariance matrix of the observables is needed to do a correct statistical comparison between different parameter sets by calculating a χ^2. The calculation of χ^2 is explained in Chapter 5. Using neural networks for the parameter determination, the knowledge of the covariance matrix is very useful to improve the performance of the network as well as necessary for the calculation of the errors of the determined CMSSM parameters.

There is only one kinematical observable in each of the event classes. Because of the mutual exclusiveness of the classes, there are no correlations between the single kinematical observables. Furthermore, the correlations between the counting observables can be clearly determined. In the following, all variances and non–vanishing covariances are presented.

The variance of the first observable, namely the number of events after cuts N, is

$$\sigma^2(N) = N. \tag{4.1}$$

The twelve event fractions n_c/N are not independent from each other, as they sum up to one because of the equation $\sum_{c=1}^{12} n_c = N$. As long as the total number of events after cuts is recorded, already eleven fractions provide the same information as twelve. Instead of using the event fractions and the total number of events, the twelve numbers of class events n_c could be saved. However, the advantage of using only one absolute number rather than twelve is the assignment of much lower systematical errors for fractions within the parameter set comparison, as described in Section 5.2. In the following, one non–vanishing event fraction can be dropped without loosing any information about the events.[‡] That means in total only 84 observables are left. Indeed, a consideration of all event fractions would lead to a covariance matrix with linear dependent rows and columns. Such a matrix would have a

[†] b–quarks will nearly always occur in quark–antiquark pairs, but the probability that both members of such pairs are not only taggable, but also tagged is rather low. Also, precisely because b–quarks almost always occur in pairs distinguishing between events with one or two b–tags adds additional information only if there is a significant number of events with more than one $b\bar{b}$ pair in the final state; such events are very rare in our scenarios. The same footnote is included in [1].

[‡] For the neural networks, the fraction of events without charged leptons, n_{0l}/N, is not included within the observables.

vanishing eigenvalue and could not be inverted. This is explicitly shown in Appendix D.1.

The covariance between two fractions out of classes c and c' is

$$\operatorname{cov}\left(\frac{n_c}{N}, \frac{n_{c'}}{N}\right) = \delta_{cc'} \frac{n_c}{N^2} - \frac{n_c\, n_{c'}}{N^3} \quad (c,\, c' \in \{1, 2, \ldots, 12\}) \,. \tag{4.2}$$

The covariance for same classes, i.e. $c = c'$, equals the variance for the particular event fraction n_c/N. In contrast, there is no correlation between the total number of events and the event fractions, i.e.

$$\operatorname{cov}\left(N, \frac{n_c}{N}\right) = 0 \quad (c \in \{1, 2, \ldots, 12\}) \,. \tag{4.3}$$

All other observables from different classes are not statistically correlated, meaning that a fluctuation of events in one class does not affect the events of other distinct classes. Furthermore, possible “physical” correlations between different classes, meaning that the change of an input parameter value would affect multiple observables from different classes, need not to be considered within the covariance matrix.

Depending on version “a” or “b”, the observables 2, 3 and 4 have different variances. For version “a”, in a particular class c the variances for the observables $O_{2,c} = n_{c,\tau^-}/n_c$, $O_{3,c} = n_{c,\tau^+}/n_c$ and $O_{4,c} = n_{c,b}/n_c$ are

$$\sigma^2(O_{i,c}) = O_{i,c} \cdot (1 - O_{i,c})/n_c \quad (i \in \{2, 3, 4\}) \,. \tag{4.4}$$

In contrast, the variances for version “b” with the observables $O_{i,c} = \langle o_i \rangle_c$ with $o_2 = \tau^-$, $o_3 = \tau^+$ and $o_4 = b$ have the form

$$\sigma^2(O_{i,c}) = \frac{1}{n_c - 1} \cdot (\langle o_i^2 \rangle_c - \langle o_i \rangle_c^2) \quad (i \in \{2, 3, 4\}) \,. \tag{4.5}$$

In both cases there is obviously no correlation between the number of b–jets and τ–leptons. On the other hand, the number of τ^- and τ^+ within the class events would have some correlation if their is a sizable number of events containing at least one $\tau^-\tau^+$ pair. In the following, this correlation is ignored because of two reasons. First, in most cases a τ–lepton is produced with the corresponding τ–(anti)neutrino rather than in a $\tau^-\tau^+$ pair. Second, in the cases a $\tau^-\tau^+$ pair is produced, both τ–leptons have to decay hadronically, pass the existing cuts and especially be tagged with each a 50 % probability to be counted in both observables. If these requirements can be summarized in a probability p_τ to measure a τ, the variances of the τ–observables would each be proportional to p_τ, while the covariance between both would scale like p_τ^2. Since p_τ is relatively small, the resulting covariance should be negligible. This assumption is explicitly confirmed for test statistics in Section 5.1.

The variances of the last three observables are calculated in the same way as equation (4.5). With $O_{i,c} = \langle o_i \rangle_c$ and $o_5 = j$, $o_6 = j^2$, $o_7 = H_T$ the variances are

$$\sigma^2(O_{i,c}) = \frac{1}{n_c - 1} \cdot (\langle o_i^2 \rangle_c - \langle o_i \rangle_c^2) \quad (i \in \{5, 6, 7\}) \,. \tag{4.6}$$

Lastly, the average number of jets and the average squared number of jets are correlated within a given class. The corresponding covariance is

$$\operatorname{cov}(\langle j \rangle_c, \langle j^2 \rangle_c) = \frac{1}{n_c - 1} \cdot (\langle j^3 \rangle_c - \langle j \rangle_c \langle j^2 \rangle_c) \tag{4.7}$$

with $\langle j^3 \rangle_c$ being the average of the number of jets to the power of three. All other entries of the covariance matrix are zero.

5. Usefulness of the Observables

The usefulness of the observables described in Chapter 4 is checked in this chapter. For this purpose, they are used for the discrimination of the 283 degenerate pairs described in [5]. Arkani–Hamed et al. could not distinguish these parameter set pairs using their comparison method with 1,808 mostly kinematical observables. Consequently, if a large fraction of these pairs can be discriminated using the described observables, this would indicate their usefulness for the determination of supersymmetric parameters. The content of this chapter is published in [1].

To compare two data sets A and B, a χ^2 is defined in the following way:

$$\chi^2_{AB} = \sum_{m,n} (O^A_m - O^B_m) V^{-1}_{mn} (O^A_n - O^B_n) \tag{5.1}$$

The double sum runs over all up to 84 relevant observables with O^A_m being the m–th observable of data set A and O^B_m being the m–th one of data set B, respectively. Which of the observables are relevant is described in Section 5.1. V^{-1} is the inverse of the covariance matrix V with entries

$$V_{mn} = \mathrm{cov}(O^A_m, O^A_n) + \mathrm{cov}(O^B_m, O^B_n)\,. \tag{5.2}$$

The covariances of each parameter set are calculated as described in Section 4.2, whereas the diagonal entries of V contain the added variances. Equation (5.2) then corresponds to adding the errors of A and B in quadrature.

To quantify how similar two different data sets are the p–value is more praticable than χ^2_{AB}. For a proper χ^2–distribution the probability p of finding a χ^2 bigger than χ^2_{AB} is

$$p = \int\limits_{\chi^2_{AB}}^{\infty} f(z, n_d) dz \tag{5.3}$$

with $f(z, n_d)$ being the χ^2 probability density function with

$$f(z, n_d) = \frac{z^{(n_d - 2)/2}\, \mathrm{e}^{-z/2}}{2^{n_d/2}\, \Gamma(n_d/2)} \tag{5.4}$$

and n_d being the number of degrees of freedom. The number of degrees of freedom equals the number of observables included in the sum in equation (5.1). $\Gamma(n_d/2)$ is the gamma function with the argument $n_d/2$. Equations (5.3) and (5.4) are only correct if χ^2_{AB} was constructed from n_d Gaussian random variables. In the following section, the calculation of χ^2_{AB} with the given observables and especially the form of the covariance matrix given in Section 4.2 should be checked. This is done with some test statistics. After that, the observables are used on the degenerate pairs to confirm their usefulness for supersymmetric parameter determination.

5.1. Test Statistics

In this section it is checked, if the χ^2_{AB} from equation (5.1) using the given observables and covariance matrix from Chapter 4 follows a proper χ^2–distribution. For this purpose, parameter sets are simulated twice for different statistics and then compared to themselves. In practice this means that a particular parameter set is simulated for a specific seed in Herwig++ in the context of the LHC with a center of mass energy of 14 TeV and an integrated luminosity of $10\,\mathrm{fb}^{-1}$. Now the same parameter set is simulated a second time with a different seed in Herwig++. Both generated data sets are then compared with equation (5.1) and the p–value is calculated with equations (5.3) and (5.4). If such a comparison is done for multiple parameter sets, the resulting p–value distribution should be flat, if χ^2_{AB} follows a proper χ^2–distribution and especially the covariance matrix is correct.

As a test sample 3305 parameter sets from reference [5] are used. As mentioned before, their degenerate pairs fulfill $(\Delta S_{AB})^2 < 0.285$ with $(\Delta S_{AB})^2$ being their variable to quantify the difference between two data sets A and B. Parameter set pairs with higher or equal values of $(\Delta S_{AB})^2$ are denoted distinguishable, whereas their distinguishability grows with increasing $(\Delta S_{AB})^2$. The 3305 parameter sets are contained in the 4654 pairs being in the range $0.285 \leq (\Delta S_{AB})^2 < 0.44$.* So by the criteria of [5], these pairs are still relatively hard to discriminate. The upper bound of the range was determined by the wish of having a sufficient number of parameter sets for the p–distribution of the self–comparison, while at the same time respecting the limited, existing computing resources.

In the following, all 3305 parameter sets are each simulated twice with different statistics. The parameter sets are then compared using either only the total number of events or one of the seven other observables described in Section 4.1 (version "a" of the observables is used). As mentioned before, the calculation of the p–values computed from the χ^2_{AB}–values can only be correct if the used observables follow Gaussian statistics. Looking at the total number of events N, this is satisfied for all considered parameter sets, because for each set a suitable number of events pass the required cuts described in Appendix B.

On the contrary, the other observables only follow a Gaussian distribution if they are each based on enough class events. This can be clarified with the following example. As mentioned before, there are in general not many events with τ–leptons included in the parameter sets from [5]. As an example it is assumed that on average only one in hundred class events contains at least one tagged, hadronically decaying τ–lepton. In this case, it would not make much sense to compare the τ–observables of two data sets for classes with only ten events, because in such classes normally no or at most one τ–lepton is included. As a consequence, such a τ–observable would not follow Gaussian statistics. In particular, this can be seen for the p–value which results from the comparison of the τ–observable of such a single class for both data sets A and B. The p–value would mostly take one of two discrete values. In this example, the fraction of τ^-–events is picked as the τ–observable. The calculation of χ^2_{AB} with equation (5.1) for this comparison of the fraction of τ^-–events for a single class c from two different

*The given $(\Delta S_{AB})^2$ range actually contains 4658 pairs and 3307 parameter sets, but with two sets problems occurred simulating them with SUSY–HIT followed by Herwig++. The same footnote is included in [1].

data sets A and B, generated from the same parameter set, would look like:

$$\begin{aligned}
\chi^2_{AB} &= \left(\frac{n^A_{c,\tau^-}}{n_{c,A}} - \frac{n^B_{c,\tau^-}}{n_{c,B}}\right)^2 \cdot \left[\frac{n^A_{c,\tau^-}}{n^2_{c,A}} \cdot \left(1 - \frac{n^A_{c,\tau^-}}{n_{c,A}}\right) + \frac{n^B_{c,\tau^-}}{n^2_{c,B}} \cdot \left(1 - \frac{n^B_{c,\tau^-}}{n_{c,B}}\right)\right]^{-1} \\
&\approx \frac{(n^A_{c,\tau^-} - n^B_{c,\tau^-})^2}{n^2_{c,A}} \cdot \frac{n^2_{c,A}}{n^A_{c,\tau^-} + n^B_{c,\tau^-}} \\
&= \frac{(n^A_{c,\tau^-} - n^B_{c,\tau^-})^2}{n^A_{c,\tau^-} + n^B_{c,\tau^-}} \qquad (5.5)
\end{aligned}$$

The variances of the fraction of τ^-–events used in the first line are from equation (4.4). In the second line, the approximations $n_{c,A} \approx n_{c,B}$ and n^A_{c,τ^-}, $n^B_{c,\tau^-} \ll n_{c,A}$ are used, which follow from the fact that the same parameter set is simulated for both data sets and the number of τ^-–events is assumed to be very low. If there are no τ^-–events in class c, i.e. $n^A_{c,\tau^-} = n^B_{c,\tau^-} = 0$, then the numerator as well as the denominator would become zero in equation (5.5). In particular, the added variances within the square brackets in the first line would be zero (leading to the zero in the denominator) and a meaningful comparison of the observables would not be possible. This actually happens for a few cases as can be seen later. If either $n^A_{c,\tau^-} = 0$ and $n^B_{c,\tau^-} = 1$ or vice versa, it leads to $\chi^2_{AB} \approx 1$. Furthermore, if $n^A_{c,\tau^-} = 1$ and $n^B_{c,\tau^-} = 1$ the resulting value would be $\chi^2_{AB} \simeq 0$.

The p–value distribution for such a comparison would then not be flat, if the number of class events n_c is small and therefore the number of τ^-–events in the class, n_{c,τ^-}, is in most cases either zero or one. Using equations (5.3) and (5.4) with $n_d = 1$ (i.e. one class is compared), there would be a peak at $p \simeq 1$ resulting from $\chi^2_{AB} \simeq 0$ and another one at $p \approx 0.32$ resulting from $\chi^2_{AB} \approx 1$. This example shows that it makes sense to require a minimal number of class events for each observable to insure its Gaussian distribution and with that the correctness of the p–distribution for the parameter set comparison. Only observables which fulfill the imposed minimal number for their classes are in the following considered in the calculation of χ^2_{AB} with equation (5.1). Note that this is the reason why it is not sensible to split up the events into too many classes, as mentioned earlier. For the same number of events each class would on average contain less events and therefore the corresponding observables would also be based on less events. If the numbers of class events are too low, the observables would not follow (approximate) Gaussian statistics and could not be used in a meaningful calculation of the p–value.

Table 5.1.: The minimal numbers $n_{i,\min}$ are listed for our seven kinds of observables. For the fraction of all events that belong to a given class, n_c/N, at least one of the compared data sets A and B has to fulfill this condition. For all other observables both data sets have to contain at least $n_{i,\min}$ events in a given class c for $O_{i,c}$ to be included in the calculation of χ^2_{AB}. Table taken from [1].

Observable	$n_{i,\min}$
n_c/N	10
$n_{c,\tau^-}/n_c$	500
$n_{c,\tau^+}/n_c$	500
$n_{c,b}/n_c$	50
$\langle j\rangle_c$ and $\langle j^2\rangle_c$	50
$\langle H_T\rangle_c$	10

Table 5.1 lists the minimal number of class events which are required for a given observable $O_{i,c}$. They depend only on the kind of observable $i \in \{1, \ldots, 7\}$ and not on the class c. For the comparison of the event fractions $O_{1,c} = n_c/N$ for a particular class c, at least $n_{1,\text{min}} = 10$ events are required for one of the two data sets. In contrast, for all other observables always the classes from both data sets have to contain at least the minimal number $n_{i,\text{min}}$ of events. The difference is made, because two different parameter sets can be distinguished well, if one of them has a high number of events in a particular class while the other one has a low number. Especially the Gaussian statistics should be acceptable in this case, because the error for both data sets calculated with equation (4.2) would be dominated by the data set with the higher number of class events. On the other hand, the error of all other observables should only be (approximately) Gaussian if the classes of both data sets include a sufficient number of events.

All minimal numbers are determined by comparing the 3305 parameter sets to themselves. This is done by only looking at one particular observable at a time and calculating the 3305 p–values for this observable using equations (5.1), (5.3) and (5.4). In each self–comparison up to twelve classes are compared. The number of compared classes is obviously influenced by the choice of the required minimal number of class events. The minimal numbers are then chosen in such a way that the resulting p–distribution is approximately flat. As written in Table 5.1, the minimal numbers for the observables $O_{2,c} = n_{c,\tau^-}/n_c$ and $O_{3,c} = n_{c,\tau^+}/n_c$ are with $n_{2,\text{min}} = n_{3,\text{min}} = 500$ relatively high. This follows as expected from the low number of τ–leptons in the considered parameter sets. Furthermore, for the considered parameter sets most events are in the three classes with at most one lepton. Hence, the observables from these classes tend to affect the calculation of χ^2_{AB} the most.

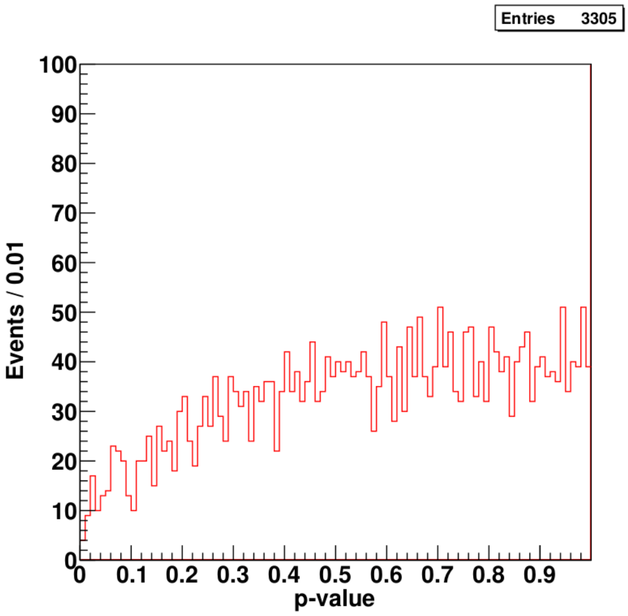

Figure 5.1.: The p–value distribution of the self–comparison for the sample comprising 3305 parameter sets. Only the total number of events after cuts, N, is included in the calculation of χ^2_{AB} and p. Figure taken from [1].

In the following, the single p–value distributions for the self–comparison of the 3305 parameter sets are presented. Figure 5.1 shows the p–value distribution from only comparing the number of total events after cuts N. Obviously, the distribution is not flat and instead shifted to higher values of p. This means that the compared parameter sets look more similar than they actually are. The origin for this bias is the way how the numbers of events after cuts for both data sets, N_A and N_B, are created. As mentioned earlier, at first 10,000 events for a parameter set are simulated to determine the cross–section. This cross–section is now used for both data sets A and B to simulate each an integrated luminosity of $10\,\mathrm{fb}^{-1}$. Before the application of the event cuts both data sets then have obviously the same number of events $N_A = N_B$, which would lead to $p = 1$ for all self–comparisons if the events before cuts would be compared. Since only a fraction of the events pass the cuts, the distribution actually becomes relatively flat for $p \geq 0.4$. The bias could be removed, if instead of simulating the same number of events for both data sets, the simulated number of events is each picked from a Poisson distribution. The mean value of this distribution would be the number of events for $10\,\mathrm{fb}^{-1}$ integrated luminosity calculated from the 10,000 event cross–section. As seen in Figure 5.1, the bias is only affecting very low values in the p–distribution. Therefore we neglect it and simulate the fixed number of events as described. For the consideration of the degenerate pairs this approach is conservative, because the p–value tends to be bigger. Furthermore, for the determination of the CMSSM parameters with neural networks this bias does not appear at all, because there every parameter set is simulated with a different seed in Herwig++.

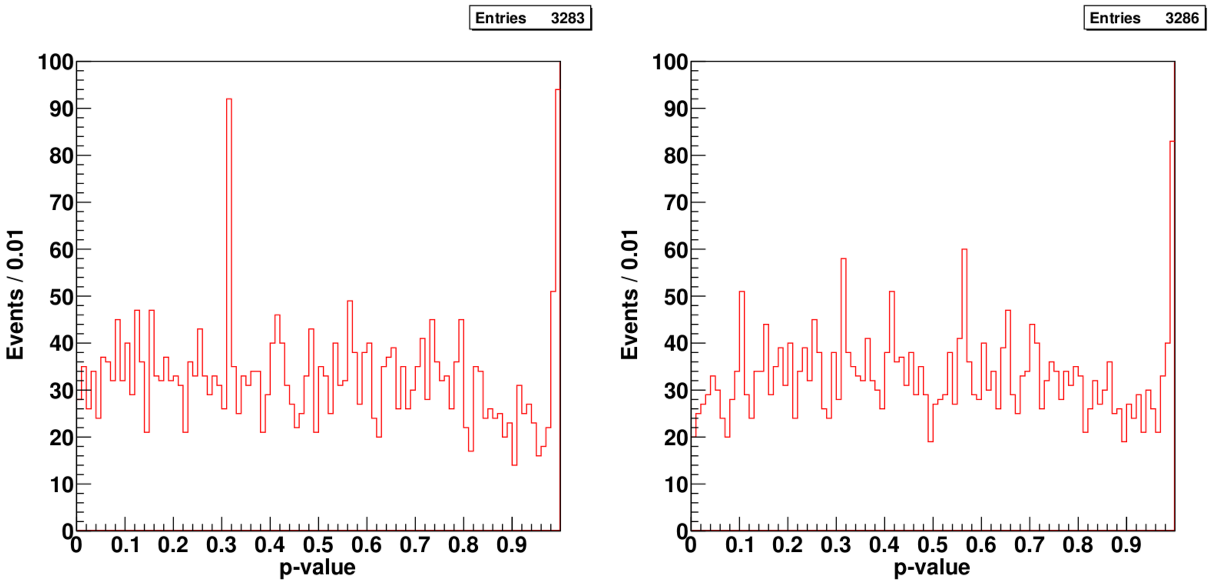

Figure 5.2.: The p–value distribution of the self–comparison for the sample containing 3305 parameter sets. Only the fraction of events in a given class containing an identified τ^- (left) or τ^+ (right) is included in the calculation of p. On average 2.8 classes satisfy the requirement $n_c \geq 500$ (see Table 5.1) and are thus included in the definition of χ^2_{AB}. The number of entries are slightly smaller than 3305, since a pair of data sets for a given set of input parameters is excluded if all variances (of the considered observables) in the compared classes vanish for both data sets or if no class fulfills the $n_c \geq 500$ requirement. Figure taken from [1].

Figure 5.2 shows the p–value distributions for the fraction of events containing a tagged τ^- and τ^+, respectively. The distributions are overall flat except for having peaks at certain positions. As mentioned earlier, the peak at $p \approx 0.32$ arises from the comparison of only one class containing one τ in the first and no τ in the second data set. Similarly, the peak at $p \simeq 1$ originates from the comparison of data sets which have in all classes the same number of tagged τ–leptons. The peaks are more pronounced for the τ^-–event fractions than for the τ^+–event fractions. As said, the overall cause for the peaks is the low number of τ–events. This is more acute for the negatively charged τ–leptons, since the LHC has a positive electrical charge in the initial state and the minimal number of required class events $n_{2,\mathrm{min}} = n_{3,\mathrm{min}} = 500$ are the same for τ^- and τ^+. Furthermore, both distributions in Figure 5.2 have each less than 3305 entries. The reason for that is that pairs which do not have any tagged hadronically decaying τ–leptons in their data sets are not compared, because the numerator as well as the denominator in equation (5.1) vanishs. Additionally, there are also some data set pairs which do not include any common class with $n_c \geq 500$ events.

Although the distributions in Figure 5.2 are not perfectly flat, they are acceptable. The peak at $p \simeq 1$ leads again to a by trend more conservative discrimination of the degenerate pairs. On the other hand, the peak at $p \approx 0.32$ probably undervalues the true p–value for the situation that one data set contains one tagged τ–lepton while the other one does not. This is explained in Appendix D.2. Overall, these peaks do not affect the calculation of χ^2_{AB} including all observables by too much, as can be seen later. The reason for that is that they only occur for parameter sets with very low numbers of tagged τ–leptons and in such a case the contribution of the τ–observables in equation (5.1) is not too high.

Figure 5.3 shows the p–value distributions for the other five observables doing the self–comparison with the minimal numbers listed in Table 5.1. The average number of jets and the average squared number of jets are combined in one distribution to also verify their correlation. As seen, all distributions are overall flat, which confirms the corresponding variances and covariances in Section 4.2.

Finally, Figure 5.4 shows the p–distribution for the self–comparison of the 3305 parameter sets using all observables. On average 40 out of 84 observables are compared for each data set pair. The shown distribution is overall flat. This confirms the choice for the minimal numbers listed in Table 5.1 and in particular the correctness of the calculation of χ^2_{AB} and p with equations (5.1), (5.3) and (5.4) as well as of the covariance matrix described in Section 4.2. Furthermore, as expected, the deviations seen in the single p–distributions of the total number of events after cuts N as well as of both τ–observables do not significantly distort the common p–value. In the following, the p–value can be used to discriminate the degenerate pairs in a statistical valid manner.

5.2. Discrimination of the Degenerate Pairs

In the previous section, the statistical correctness of the used observables and covariance matrix in the context of the calculation of χ^2_{AB} and p was demonstrated. In this section, the usefulness of the chosen observables for the determination of supersymmetric parameters should be shown. To do so, the observables are used to try to distinguish between the 283 degenerate pairs described in [5]. As mentioned before, Arkani–Hamed et al. failed within their comparison method to discriminate between the 10 fb^{-1} measurement signatures of the parameter sets composing these pairs, in the context of the LHC with a center of mass energy of 14 TeV. If a significant number of these pairs could be disentangled with the use of the observables described in Section 4.1, this would indicate their usefulness for the determination of supersymmetric parameters.

The 283 degenerate pairs contain 384 different parameter sets.[†] Doing the comparison between two parameter sets A and B in a realistic case, one of them would be a measurement and the other one a simulated prediction. The difference is that a prediction can be simulated with an in principle

[†] In almost all 384 parameter sets, which compose the degenerate pairs from [5], the lightest neutralino is the LSP. However, there are four sets with a different LSP, namely the lighter $\tilde{\tau}$–eigenstate. Such a parameter set can easy be discriminated from a parameter set which has the neutralino as LSP, when a search for stable, charged particles is done. Note that the degenerate pairs including one of these four parameter sets can be distinguished in the following, using only the described comparison method with the 84 introduced observables. There is no degenerate pair including two of these sets.

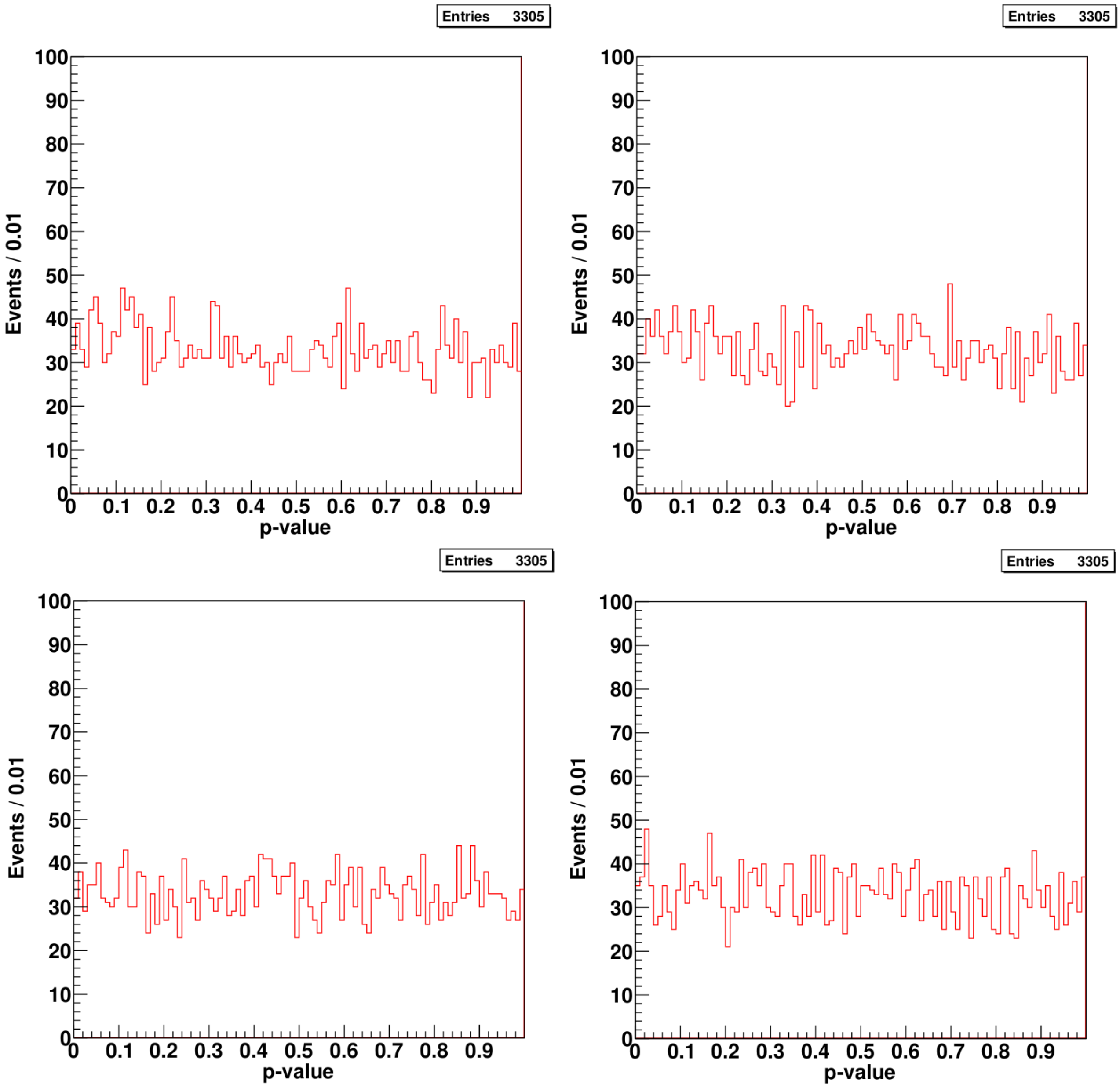

Figure 5.3.: The p–value distribution of the self–comparison for the sample with 3305 parameter sets. Either only the average class fractions n_c/N (top left), the average number of class events containing at least one identified b–jet (top right), the average number of jets $\langle j \rangle_c$ and average squared number of jets $\langle j^2 \rangle_c$ (bottom left), or the kinematical observable $\langle H_T \rangle_c$ (bottom right) is included in the calculation of χ^2_{AB} and p. The required minimal numbers are listed in Table 5.1. On average 8.1 classes have at least ten events for one of both data sets, 7.8 classes include at least ten events for both data sets and 5.9 classes have at least 50 events for both data sets.

unlimited precision by generating a high enough number of events. To that end the prediction would give the true expectation values of the 84 observables for the chosen parameter set. On the other hand, the statistical uncertainty of the measurement is determined by the measured integrated luminosity, in

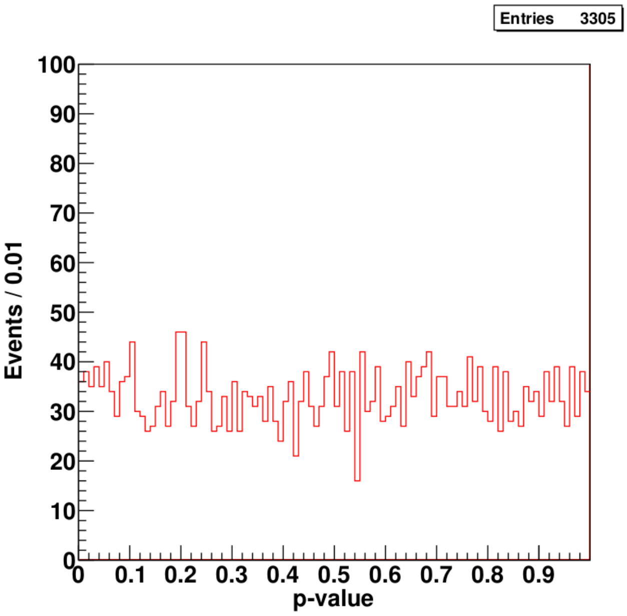

Figure 5.4.: The p–value distribution of the self–comparison for the sample containing 3305 parameter sets, including all relevant observables. On average around 40 observables are compared for a pair of data sets. Figure taken from [1].

our case $10\,\text{fb}^{-1}$ of data. In the following, data sets of a measurement and a prediction are compared, which is in contrast to [5], where data sets of two measurements were compared. Respecting the available computing resources, it is not possible to simulate the predictions with a completely vanishing uncertainty. Instead, the predictions are based on an integrated luminosity of $100\,\text{fb}^{-1}$, which makes their uncertainties nearly (also not completely) negligible.

In the comparison of the degenerate pairs, parameter sets A and B are emancipated, i.e. it is not defined which parameter set belongs to the measurement and which one belongs to the prediction. In principle a specific choice should not substantially influence the result of the comparison of the parameter sets. The reason for that is that a specific choice would only make a difference, if for the same integrated luminosity the statistical uncertainties for the compared observables of both data sets differ significantly. But in such a case also the values of the observables should normally differ significantly, which then should make sure that they actually can be distinguished independent of the specific choice of measurement and prediction. However, to have a clear rule for the comparison of the parameter sets the covariance matrix from equation (5.2) is symmetrized under the exchange of A and B in the following way:

$$\begin{aligned} V_{mn} &= \frac{\operatorname{cov}(O^A_{m,10}, O^A_{n,10}) + \operatorname{cov}(O^B_{m,10}, O^B_{n,10})}{2} + \frac{\operatorname{cov}(O^A_{m,100}, O^A_{n,100}) + \operatorname{cov}(O^B_{m,100}, O^B_{n,100})}{2} \\ &+ \delta_{mn} \left(k^{(\text{syst})}_m \frac{O^A_{m,10} + O^B_{m,10}}{2} \right)^2 \end{aligned} \tag{5.6}$$

This is the average of the covariance matrices which are formed by comparing a measurement A to a prediction B and the other way around. The subscript “10” stands for a $10\,\text{fb}^{-1}$ measurement and the “100” for a $100\,\text{fb}^{-1}$ prediction, respectively. This means that the first term expresses the statistical errors for the measurements and the second term the in our case not completely vanishing statistical errors for the predictions. Here has to be noted that the total number of events after cuts as an absolute value is treated differently than the other observables. The predicted expectation values of the other (relative) observables do not depend on the integrated luminosity. In contrast, the prediction of the total number of events as well as its corresponding variance has to be divided by ten to allow a comparison to the measurement. Furthermore, compared to equation (5.2) there are additional systematic errors located at the diagonal entries of the covariance matrix formed with equation (5.6).

Systematic errors are treated in the same way as in [5] to insure comparability of the results as much as possible. The factor $k_m^{(\text{syst})}$ fixes the systematic error of the m–th observable to be a certain fraction of the measured observable value. For the total number of events after cuts N as being an absolute number, the factor is picked in [5] to be $k_N^{(\text{syst})} = 0.15$. This systematic uncertainty of 15 % includes the theoretical error on the calculation of the cross–section as well as the experimental error on the measurement of the integrated luminosity. The systematic errors of all other observables as being fractions are chosen to be much smaller. Reference [5] fixes the errors of those to be each 1 %, having a factor $k_m^{(\text{syst})} = 0.01$.

In the following, the χ^2_{AB} of the degenerate pairs are calculated using equations (5.1) and (5.6). The relevant observables for the double sum are determined by requiring the corresponding minimal numbers of class events in Table 5.1, whereas the total number of events after cuts is always compared. After that, the belonging p–value is calculated with equations (5.3) and (5.4).

On average around 32 out of 84 observables are compared for the degenerate pairs. This number is lower than for the self–comparison of the 3305 parameter sets in Section 5.1, which was around 40. Comparing data sets generated from the same parameter set should make it more likely that for a specific observable both classes fulfill the required minimal numbers in Table 5.1 at the same time. On the other hand, for data sets simulated from different parameter sets it is not naturally that this happens at the same time. Therefore, on average less observables are included in the calculation of χ^2_{AB} for different parameter sets. Furthermore, on average 25,000 events pass the selection cuts described in Appendix B. As before for the self–comparison, most of these events include no or one lepton (electron, muon and corresponding antiparticles) and are therefore mostly located in the three corresponding lepton classes.

In the following, the discrimination of the degenerate pairs is considered for the four different combinations of the (non–)inclusion of systematic errors and Standard Model background. At the end, the usefulness of the observables is discussed.

5.2.1. Discrimination without Standard Model Background

In this subsection no Standard Model background is considered for the comparison of the degenerate pairs. Without the consideration of systematic errors, i.e. all $k_m^{(\text{syst})} = 0$ in the covariance matrix from equation (5.6), all degenerate pairs can be distinguished with a 95 % confidence interval, meaning all determined p–values satisfy $p < 0.05$. The mean value for all pairs is $\bar{p} = 6.8 \cdot 10^{-5}$. Note that also in [5] the distinction of parameter sets was defined with an in their method estimated 95 % confidence interval.

In the following, systematic errors are included as described in the beginning of this section. This is the same combination (systematic errors but no background) as the one used in [5]. The comparison leads to 23 indistinguishable pairs with p–values higher than 0.05. On the other hand, still most of the pairs, meaining 260, can be discriminated using the described comparison method. Especially, these 260 pairs could not be distinguished in reference [5]. The resulting mean value is $\bar{p} = 0.038$.

To better understand the p–distribution, we consider the quantity $(\Delta P_{AB})^2$ from [5] which is defined as

$$(\Delta P_{AB})^2 = \frac{1}{n_{para}} \sum_{i=1}^{n_{para}} \left(\frac{P_i^A - P_i^B}{\bar{P}_i^{AB}} \right)^2 . \tag{5.7}$$

P_i^A and P_i^B are the i–th parameter of the parameter sets A and B, respectively. The number of compared parameters is labeled n_{para} and $\bar{P}_i^{AB} = (P_i^A + P_i^B)/2$ gives the average value of the i–th parameter for both parameter sets. $(\Delta P_{AB})^2$ can take values in the range [0, 4], because all 15 parameters in the considered supersymmetric model have positive values. For small values, the quantity gives approximately the quadratic relative difference between the parameters of A and B, meaning that $(\Delta P_{AB})^2 = 0.01$ would correspond to a difference of around 10 % between the parameters of both sets.

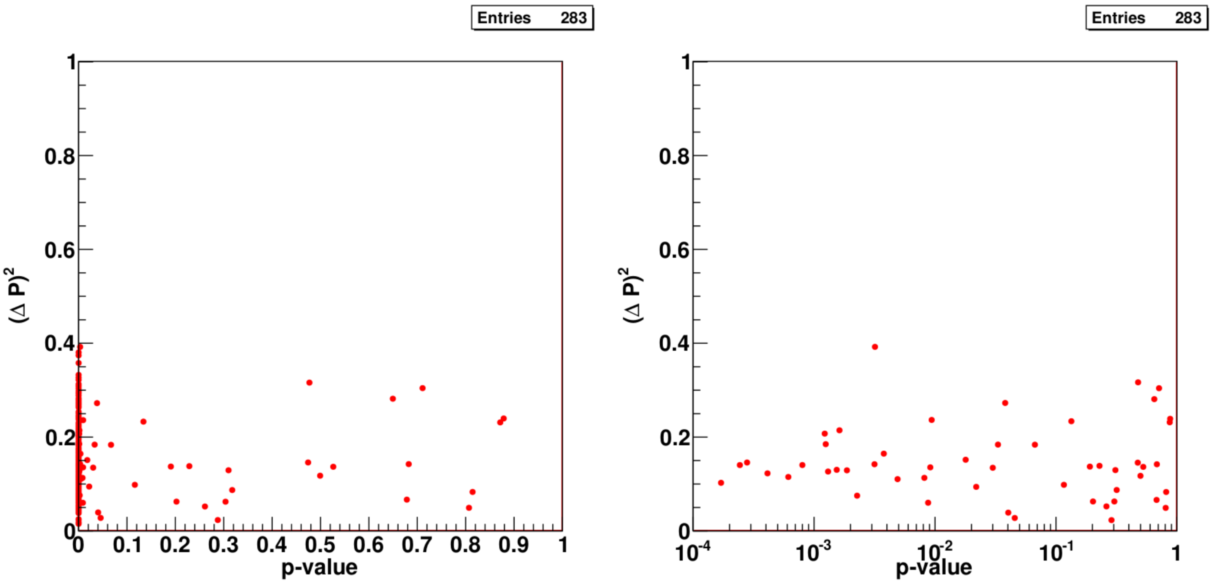

Figure 5.5.: The total parameter difference $(\Delta P_{AB})^2$ defined in equation (5.7) including all 15 parameters versus the p–value for the 283 pairs deemed indistinguishable in reference [5], including systematic errors. The p–value is shown on a linear (left) and logarithmic scale (right); in the latter case, pairs with with $p < 10^{-4}$ are not shown. Figure taken from [1].

Figure 5.5 shows the p–value distribution with their corresponding $(\Delta P_{AB})^2$–values, comparing all 15 parameters of the sets A and B. The smaller the value of $(\Delta P_{AB})^2$ is, the more similar are both parameter sets. Therefore one could expect that in general small values of $(\Delta P_{AB})^2$ go along with big values of p and the other way around. Looking at Figure 5.5, this expectation is not fulfilled. It rather seems that there is no clear correlation between the values of $(\Delta P_{AB})^2$ and p. In particular this means that there are parameter sets with only small differences, but which still produce quite different measurement signatures. For example, this could be the case if a small change of a certain mass would open or close an important decay channel, which in the following would strongly change the produced measurement signature. In contrast, there are pairs with relatively big parameter differences, but which still cannot be distinguished. In Figure 5.5 all parameters of both sets are compared. However, not all parameters are equally responsible for the differences in $(\Delta P_{AB})^2$. Certain parameters like the gluino mass or squark masses can be distinguished better between two parameter sets than other parameters like, for example, $\tan\beta$ which can be discriminated much worse. More details on that can be found in [1].

5.2.2. Discrimination with Standard Model Background

In this subsection Standard Model background is added to each generated data set from the 384 parameter sets making up the 283 degenerate pairs. The considered backgrounds are productions of Z+jets, W+jets and $t\bar{t}$ as well as single top production. The first three backgrounds are simulated only using Herwig++, while for the last one the programs MadGraph [14] as well as Herwig++ are used. For the backgrounds Z + jets and W + jets only leptonic decays of W and Z, respectively, are simulated, because only those have a chance to pass cuts on the missing transverse momentum described in Appendix B. Furthermore, W and Z are required each to have a minimal transverse momentum of 100 GeV at the parton level. Because almost all events with a lower transverse momentum for the boson should also not pass the required missing transverse momentum cut. Practically, it changes the background after cuts only negligible, but reduces the number of events necessary to be generated significantly. Note that all background (as well as signal) events are only considered to leading order in perturbation theory.

For an integrated luminosity of 10 fb^{-1}, in total 29,052 background events pass the selection cuts which is of similar size as the average number of signal events with around 25,000. Less than 0.2 % of the simulated background events pass the imposed selection cuts. Note that the total cross–section for the considered backgrounds were already reduced by the 100 GeV requirement on the transverse momentum of the W and Z as well as the consideration of only leptonic decays. Since practically all of these non–considered events should not pass the cuts, the total cut efficiency is even much lower. The comparison between the 30 % supersymmetric and the less than 0.2 % SM events, which pass the implied cuts, shows the satisfying quality of the cuts. Likewise, 293,875 background events remain after cuts for an integrated luminosity of 100 fb^{-1}. As mentioned in the beginning, the backgrounds are each added to the measurements and predictions of the parameter sets, i.e. they are added to the observable values and also considered in the covariance matrix. Most of the background events which pass the event cuts result from top pair and single top production followed by W + jets and Z + jets.

Furthermore, with the inclusion of the background events it is not feasible anymore to use the minimal numbers in Table 5.1 as a requirement for the comparison of the observables. Instead, in the following at least a "3σ–signal" is required for an observable to be compared, meaning that the number of supersymmetric class events should be at least three times greater than the square–root of the number of background events in the corresponding class. As before, for the fraction of class events only one data set has to fulfill this requirement and for all other observables the classes of both data sets. Similarly, the total number of supersymmetric events after cuts has to be at least three times greater than the square–root of the total number of background events after cuts for one of the compared data sets to include this observable in the comparison. This last requirement is fulfilled for all degenerate pairs.

For the consideration of SM background without systematic errors only one degenerate pair has a p–value greater than 0.05 and therefore cannot be distinguished. The resulting mean value is $\bar{p} = 0.0030$. The inclusion of systematic errors increases the number of indistinguishable pairs to 46 for a 95 % confidence interval with an average p–value of 0.079. That means, even with the consideration of SM background and systematic errors, still most of the 283 degenerate pairs can actually be discriminated.

5.2.3. Conclusion for the Observables

Table 5.2 gives an overview about the results of the comparison of the 283 degeneate pairs from [5] for the different combinations of systematic errors and Standard Model background in the context of a LHC with a center of mass energy of 14 TeV and an integrated luminosity of 10 fb^{-1} for the measurement. With the same assumptions as reference [5], i.e. with systematic errors but no SM background, 260 of the 283 as indistinguishable marked pairs can actually be discrimated within the

Table 5.2.: Number of pairs of parameter sets with $p > 0.05$, out of the 283 pairs deemed indistinguishable in [5], for different levels of sophistication of our analysis (with or without systematic errors and Standard Model backgrounds). The mean and median values of p for all 283 pairs are also given. Table taken from [1].

Syst. Errors	Backgrounds	no. of pairs with $p > 0.05$	$\bar{p}$	median p
No	No	0	$6.8 \cdot 10^{-5}$	$3.6 \cdot 10^{-146}$
Yes	No	23	0.038	$1.1 \cdot 10^{-36}$
No	Yes	1	0.0030	$2.6 \cdot 10^{-79}$
Yes	Yes	46	0.079	$1.4 \cdot 10^{-13}$

described comparison method. Especially, this means that the 84 observables introduced in Chapter 4 seem not only to be useful for the discrimation of two different supersymmetric parameter sets but also be suitable for the direct determination of supersymmetric parameters from a LHC measurement.

Looking at the results in Table 5.2, the effect of including systematic errors seems to be more severe than the one of including SM background. In particular, this can be seen at the changes of the median value of p. The median p–value including none of both rises after the inclusion of just systematic errors by a factor of 10^{110}, compared to 10^{67} if just SM background is considered. If both are included, it increases by a factor of 10^{133}, but even then it is still relatively small with $1.4 \cdot 10^{-13}$. This shows in the fact that still most degenerate pairs can be distinguished.

It has to be noted that the considered supersymmetric parameter sets have relatively big cross–sections and as a consequence, the backgrounds are less troubling for the data set discrimination. Furthermore, systematic errors have to be understood well on the experimental as well as on the theoretical side to allow a successful parameter determination.

In the following parameter determination with neural networks neither systematic errors nor Standard Model backgrounds are considered. However, the same selection cuts, which are described in detail in Appendix B, are applied to reduce the SM background to a manageable level. The general goal is to examine the performance of neural networks for parameter determination. As seen, the described observables are also useful after the inclusion of backgrounds and systematic errors. Therefore if the neural network method shows in general a good performance without them, then it should also still perform well with them, although then giving bigger errors for the determined parameters. Especially, experimental systematic errors can only be determined in a proper way by the considered experiment. Furthermore, probably also the used selection cuts can still be optimized for the respective considered supersymmetric parameter region.

6. Creation of the Neural Networks

In the first part of this chapter the general construction of a neural network for the CMSSM parameter determination is described. The second part of this chapter introduces the four investigated regions of the CMSSM parameter space. Furthermore, measures to improve the performance of a neural network are described, in particular to handle the statistical fluctuations of the measured observables. Finally, two different methods for the determination of the CMSSM parameter errors including the correlations are explained. The content of this chapter is published in [2].

6.1. General Neural Network Construction

In this section, the general construction of a neural network is described. General information about neural networks can be found in [15]. The presented neural network is designed for 84 input values corresponding to the 84 measured observables and four output values corresponding to the four CMSSM parameters.* The neural network should imitate the function going from the measured observables to the searched associated CMSSM parameters. In the beginning, the neural network is a function template with a specific number of free parameters. These parameters are set during a learning procedure to imitate the searched function between the input and output values. In this procedure, the neural network is trained with sets of known input and output values. The training sets consist each of the 84 observables (input), measured for the simulation of certain values of the four CMSSM parameters (output). The resulting neural network should be able to compute the unknown CMSSM parameters belonging to a specific measurement with certain errors, if the training sets are a proper representation of the investigated CMSSM parameter space.

6.1.1. Layout

Figure 6.1 shows the layout of a neural network with two weight layers in between three neuron layers. A neuron is an entity which processes an input value to an output value, as is illustrated on the right side of Figure 6.1. The shown network has 84 input values, one for each measured observable (version "b" of the observables described in Section 4.1), and four output values, one for each CMSSM parameter (the sign of μ is set positive). The first neuron layer, i.e. the input layer, contains 85 neurons, one for each input value x_i with $i = 1, \dots, 84$ and an additional one having a fixed output value $x_0 = 1$. The output values of the input neurons equal their input values, i.e. the neuron processing function is just the identity function. The input values x_i with $i = 1, \dots, 84$ are each normalized values of the measured observables, as described later in Subsection 6.1.2.

The intermediate neuron layer includes a so far not determined number $v + 1$ of "hidden" neurons. There are v hidden neurons with input values $z_1, \dots, z_v$ and as before, an additional one with a fixed output value equal to one. The number of hidden neurons determines the number of free parameters of the neural network. The network can represent a more complex function with an increasing number of hidden neurons.

*At the end instead of using one neural network for all four CMSSM parameters actually four neural networks each for one CMSSM parameter are used, as described later in Subsection 6.2.2.

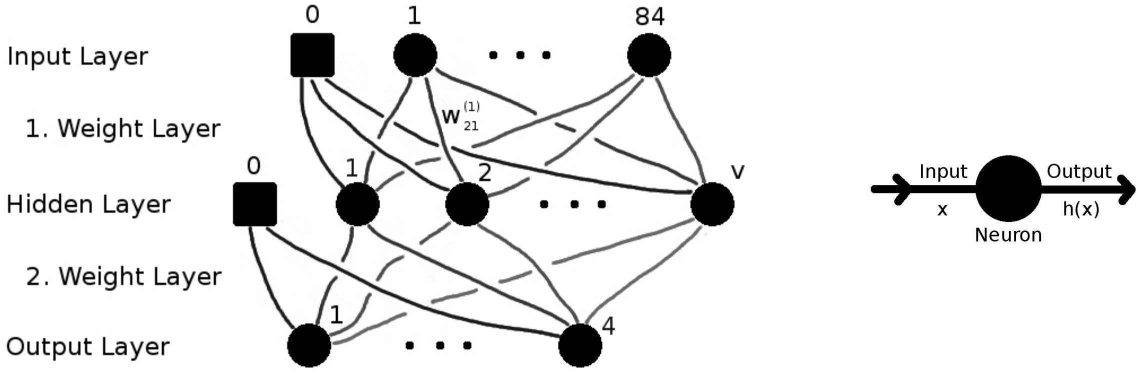

Figure 6.1.: Neural network with two weight layers (left). The squares represent the neurons with constant output and the circles the neurons with varying output depending on the network input. The circles of one layer are connected with all neurons of the previous layer. Each connection is weighted differently, e.g. the connection between the first input neuron and the second hidden neuron has the weight $w^{(1)}_{21}$. A single neuron (right) processes an input value x to an output value $h(x)$. Here, the processing function h is the identity $h(x) = x$ for the input and output neurons and the hyperbolic tangent $h(x) = \tanh(x)$ for the hidden neurons. Left figure taken from [2].

The last neuron layer, i.e. the output layer, includes four output neurons with values $y_1, \ldots, y_4$. Similar to the first layer, the input and output values for the output neurons are the same, meaning that the identity is the processing function. At the end, a normalization is applied on the output values yielding to the CMSSM parameters, as explained in Subsection 6.1.2. Note that in the following, input neurons are labeled with i, j, hidden neurons with a, b and output neurons with r, s.

There are two weight layers located between the three neuron layers. The weight layers consist of the connections between the single neurons. In the first weight layer, which is located between the input and hidden neuron layer, each input neuron is connected with all hidden neurons expect for the zeroth hidden neuron which has a fixed output value. The connection between the a–th hidden neuron and the i–th input neuron is weighted with the weight $w^{(1)}_{ai}$ with $a = 1, \ldots, v$ and $i = 0, \ldots, 84$. Similarly, in the second weight layer all hidden neurons are connected to all output neurons. The connections are weighted with the weights $w^{(2)}_{ra}$ with $r = 1, \ldots, 4$ and $a = 0, \ldots, v$. The weights in the first and second weight layer are the free parameters of the neural network function. As mentioned, their number is obviously fixed by the number of hidden neurons. Their values are determined with the learning procedure described in Subsection 6.1.3.

The neural network processes the 84 input values by computing the hidden neuron values which are then used to calculate the values of the four output neurons. The input value z_a of the a–th hidden neuron with $a = 1, \ldots, v$ is calculated with the weighted sum over all connected input neurons with

$$z_a = \sum_{i=1}^{84} w^{(1)}_{ai} x_i + w^{(1)}_{a0} = \sum_{i=0}^{84} w^{(1)}_{ai} x_i \,. \tag{6.1}$$

The output value of the a–th hidden neuron is then $\tanh(z_a)$. The hyperbolic tangens function is used as processing function for the hidden neurons to enable the neural network to imitate an arbitrary function (using a high enough number of hidden neurons). This should follow from the fact that the hyperbolic tangens function has a linear behavior for small $|z_a|$ and a binary behavior for big $|z_a|$, as can be seen in Figure 6.2. Note that in general such an imitation works only for the considered parameter region. Leaving this region, it is obviously not guaranteed that the network function still gives correct outputs. In other words, with a neural network a searched function can be interpolated but not necessarily extrapolated.

In the same way the input value y_r of the r–th output neuron is calculated with the weighted sum over all connected hidden neurons with

$$y_r = \sum_{a=1}^{v} w_{ra}^{(2)} \tanh(z_a) + w_{r0}^{(2)} = \sum_{a=0}^{v} w_{ra}^{(2)} \tanh(z_a)\,. \tag{6.2}$$

The zeroth hidden neuron has the fixed output value $\tanh(z_0) = 1$. Note that the fixed output of this neuron allows the adding of a constant value, $w_{r0}^{(2)}$, in the searched function between the input and output values. As said before, the output values of the output neurons equal their input values.

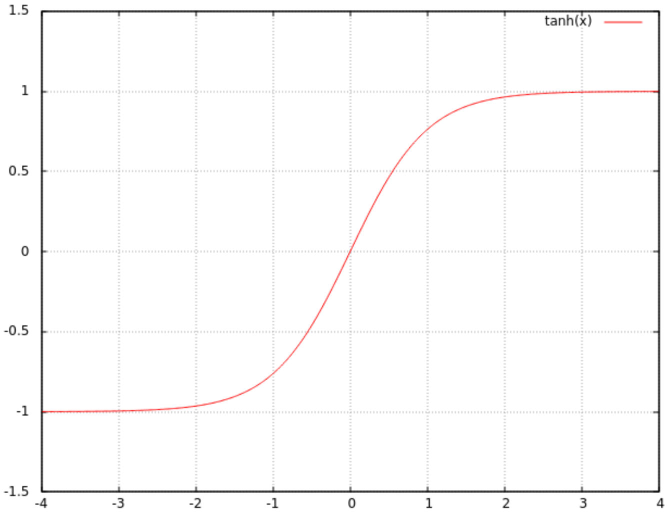

Figure 6.2.: The shape of the tangens hyperbolic function.

6.1.2. Normalization and First Weights

The 84 measured observables are not directly used as input values for the neural network, but instead at first normalized. This normalization permits an adequate first choice of the neural network weights, which in general should lead to a shorter learning process, as explained in the following. The first values of the weights should be unbiased to prevent the favorization of any learnable function. Such a neutrality is only possible if the order of magnitude of the input values is known. This order of magnitude is determined by the behavior of the tangens hyperbolic processing function which is used within the hidden neurons. As can be seen in Figure 6.2, the tangens hyperbolic function has a linear behavior for small input values. For larger values it has a binary behavior. In the beginning, neither the linear nor the binary range should be preferred. Therefore, the input values of the hidden neurons should be on the order $O(1)$. As a consequence, the observables are each normalized to input values on the order $O(1)$. This is done by determining the mean value $\bar{O}_i^{tg}$ and the variance $(\sigma_i^{tg})^2$ over all N training sets for each observable O_i with $i = 1, \ldots, 84$. If O_i^n is the observable value for the n–th training set with $n = 1, \ldots, N$, the expressions for $\bar{O}_i^{tg}$ and $(\sigma_i^{tg})^2$ are:

$$\bar{O}_i^{tg} = \frac{1}{N} \sum_{n=1}^{N} O_i^n \tag{6.3}$$

$$(\sigma_i^{tg})^2 = \frac{1}{N-1} \sum_{n=1}^{N} (O_i^n - \bar{O}_i^{tg})^2 \tag{6.4}$$

The normalized input value x_i^n corresponding to the observable value O_i^n is then calculated with the formula

$$x_i^n = \frac{O_i^n - \bar{O}_i^{tg}}{\sigma_i^{tg}} \,. \tag{6.5}$$

The mean value of the x_i^n over all training sets equals zero and the corresponding variance equals one, i.e. the input values are on the order $O(1)$ as desired. The described normalization can also be understood as using equation (6.5) as a processing function in the i–th input neuron (instead of the identity function) and feeding the neuron with the normal observable value.

Since the tangens hyperbolic functions in the hidden neurons naturally have output values on the order $O(1)$, also the input values of the output neurons should be on the order $O(1)$. This is done in the same way as for the input neurons by determining the mean value $\bar{r}^{tg}$ and the variance $(\sigma_r^{tg})^2$ over all training sets for each CMSSM parameter r with $r \in \{m_0,\, m_{1/2},\, \tan\beta,\, A_0\}$ (corresponding to the previously used output neuron labels $r = 1, \ldots, 4$). With r^n being the CMSSM parameter value of the n–th training set it follows:

$$\bar{r}^{tg} = \frac{1}{N} \sum_{n=1}^{N} r^n \tag{6.6}$$

$$(\sigma_r^{tg})^2 = \frac{1}{N-1} \sum_{n=1}^{N} (r^n - \bar{r}^{tg})^2 \tag{6.7}$$

The (inversely) normalized output value y_r^n is then

$$y_r^n = \frac{r^n - \bar{r}^{tg}}{\sigma_r^{tg}} \,. \tag{6.8}$$

Again this can be understood as using the inverse of equation (6.8) as processing functions for each output neuron instead of the identity function. Note that the normalizations are determined by only considering the training sets but that they are applied on every network input and output. This requires that the training sets are a good representation of the considered CMSSM parameter region. Furthermore, the normalization for each measured observable is done independently, i.e. the correlations between the observables, which are described in Section 4.2, are not considered. Instead, the correlations are taken into account in the preparation of the training sets, as described later in Subsection 6.2.2.

With the normalization of the input and output values of the neural network the first values of the weights can be chosen appropriately. The weights in both weight layers are each picked from Gaussian distributions around zero, because neither positive nor negative values are preferred in the beginning. As required before, the values on the input as well as output side should be for both weight layers on the order $O(1)$. Since each hidden neuron (except for the zeroth one) is connected to all 85 input neurons, the Gaussian distribution for the first layer has a variance equal 1/85. Similarly, the variance of the Gaussian distribution in the second weight layer is chosen to be $1/(v + 1)$. These choices are verified in Appendix D.3.

6.1.3. Learning Procedure

The goal of the learning prodecure is that the neural network determines the unknown CMSSM parameter values of a measurement. For this purpose, the neural network has to learn the mapping between the measured observables and the corresponding CMSSM parameters in the investigated parameter space. The network is trained with sets of known input (measured observables) and output values (CMSSM parameters). In a learning step the network is fed with the input values of the training sets to calculate the corresponding network output values, using the present weight values. The calculated output values are compared with the known, correct output values of the training sets in the following. Since the output values of the training sets should be reproduced, the weights are changed appropriately to reduce the difference between the calculated and the correct output. As a consequence, the neural network can reproduce the training sets better and better with an increasing number of learning steps. In particular if the network has enough free parameters, the training sets can be memorized and reproduced exactly. But such a memorization should be connected with a loss of generality, meaning that sets which are not included in the training sets would in general be determined worse. Therefore it is necessary to have some control sets which are not included in the training of the neural network to ensure that the parameters of a previous unseen measurement would be determined "optimally".

Just like the training sets, control sets are sets of known input and output values. In contrast to training sets, they are not used for the improvement of the weights in the learning steps. After each learning step, it is examined how well the control sets are reproduced by the current neural network. The learning procedure is stopped when the control sets are reproduced the best. As long as the control sets are like the training sets a proper representation of the investigated CMSSM parameter region, this ensures an optimal generality of the neural network function.

The quality of the reproduction of the control sets can be quantified with the normalized control error

$$\tilde{F} = \sqrt{\frac{\sum\limits_{n=1}^{M}\left(\vec{y}^{n} - \vec{k}^{n}\right)^{2}}{\sum\limits_{n=1}^{M}\left(\bar{\vec{k}} - \vec{k}^{n}\right)^{2}}}\,. \tag{6.9}$$

Here, M is the number of control sets, the vector $\vec{y}^{n}$ includes the four calculated network output values for the input values of the n–th control set, the vector $\vec{k}^{n}$ contains the correct output values, and the vector $\bar{\vec{k}}$ is the mean value calculated over all correct control output values with

$$\bar{\vec{k}} = \frac{1}{M}\sum_{n=1}^{M}\vec{k}^{n}\,. \tag{6.10}$$

The control error is calculated after each learning step and the optimal neural network function is achieved at the (global) minimum of the control error. Further learning steps would lead to an increasing control error, since the better reproduction of the training sets should corrupt the general parameter determination and therefore worsen the control set reproductions. A training error calculated with equation (6.9) for the training sets would in contrast monotonously decrease with an increasing learning step number. Note that also before the global minimum is reached the control error is not a monotonously decreasing function, because a better reproduction of the training sets does not necessarily come along with a better reproduction of the control sets in every learning step.

The described behavior of the control and training errors for a neural network is shown in Figure 6.3.

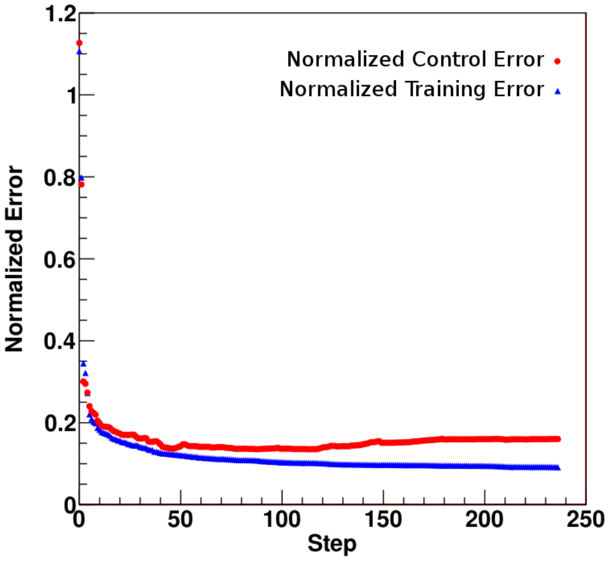

Figure 6.3.: Error evolution of a neural network with 20 hidden neurons for the parameter $m_{1/2}$ of reference point 3, which is later introduced in Subsection 6.2.1. The blue triangles show the normalized training error and the red dots the corresponding normalized control error. The minimum of the control error is reached with 0.136 at learning step 107. Figure taken from [2].

The improvement of the weights in each learning step is done with the "conjugate gradient" algorithm. The procedure is described in detail in Appendix C.

6.2. Neural Networks for Parameter Determination

In this section, the four different CMSSM parameter regions which are investigated are described. For each region a reference point is introduced whose parameters should each be determined using neural networks. The results are described in Section 7.1. Furthermore, the used training and control sets are listed in the following. Afterwards it is explained, how the performance of the neural networks can be improved by understanding the statistical fluctuations of the measured observables. To do so, the covariance matrix of the observables, which is introduced in Section 4.2, is used. At the end, the variances and covariances of the observables are also used to compute the errors including the correlations of the determined CMSSM parameters. Two different methods for the error determination are explained.

6.2.1. Reference Points, Training and Control Sets

The determination of the CMSSM parameters is examined for four different reference regions of the CMSSM parameter space to obtain a comprehensive overview about the performance of the neural networks for different measurement signatures. The in Chapter 4 described observables are used as input. Each reference measurement should have around 1,000 events after cuts for an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The reference points are:

Table 6.1.: CMSSM input parameters and selected superparticle and Higgs boson masses for the four reference points. All mass parameters are in GeV. Note that first and second generation sfermions with the same gauge quantum numbers have identical masses. Moreover, $m_{\tilde{d}_L} \simeq m_{\tilde{u}_L}$ in these scenarios, while $m_{\tilde{d}_R} \simeq m_{\tilde{u}_R}$, and $m_{\tilde{\nu}} \simeq m_{\tilde{e}_L}$. Similarly, $m_{\tilde{\chi}_1^\pm} \simeq m_{\tilde{\chi}_2^0}$ in all cases, while $m_{\tilde{\chi}_4^0} \simeq m_{\tilde{\chi}_2^\pm}$ are about 10 to 30 GeV above $m_{\tilde{\chi}_3^0}$. Finally, in all cases $m_H \simeq m_A \simeq m_{H^\pm}$. Table taken from [2].

	Point 1	Point 2	Point 3	Point 4
m_0	150	2000	1000	400
$m_{1/2}$	700	450	600	700
$\tan\beta$	10	10	10	30
A_0	0	0	1500	0
$m_{\tilde{g}}$	1570	1133	1407	1578
$m_{\tilde{u}_L}$	1437	2171	1575	1483
$m_{\tilde{u}_R}$	1382	2161	1541	1430
$m_{\tilde{b}_1}$	1322	1816	1402	1323
$m_{\tilde{b}_2}$	1371	2145	1529	1377
$m_{\tilde{t}_1}$	1118	1384	1168	1145
$m_{\tilde{t}_2}$	1357	1824	1414	1366
$m_{\tilde{e}_R}$	306	2005	1024	480
$m_{\tilde{e}_L}$	494	2015	1074	616
$m_{\tilde{\tau}_1}$	298	1988	1010	414
$m_{\tilde{\tau}_2}$	494	2007	1068	606
$m_{\tilde{\chi}_1^0}$	291	187	249	293
$m_{\tilde{\chi}_2^0}$	551	344	468	555
$m_{\tilde{\chi}_3^0}$	848	463	668	830
m_h	116	116	114	117
m_A	969	2040	1244	889

1) $m_0 = 150\,\text{GeV}, \quad m_{1/2} = 700\,\text{GeV}, \quad \tan\beta = 10 \quad \text{and} \quad A_0 = 0\,\text{GeV}$

2) $m_0 = 2000\,\text{GeV}, \quad m_{1/2} = 450\,\text{GeV}, \quad \tan\beta = 10 \quad \text{and} \quad A_0 = 0\,\text{GeV}$

3) $m_0 = 1000\,\text{GeV}, \quad m_{1/2} = 600\,\text{GeV}, \quad \tan\beta = 10 \quad \text{and} \quad A_0 = 1500\,\text{GeV}$

4) $m_0 = 400\,\text{GeV}, \quad m_{1/2} = 700\,\text{GeV}, \quad \tan\beta = 30 \quad \text{and} \quad A_0 = 0\,\text{GeV}$

The Higgsino parameter μ is positive for all four points. These points are chosen to be just beyond the current exclusion limits for the CMSSM [3]. Table 6.1 shows the four mass spectra of the superparticles and Higgs bosons. Note that a Higgs mass close to 125 GeV is not required, although a (more and more) Higgs–like boson with this mass was recently discovered at the LHC [16]. Requiring this mass would obviously constrain the allowed parameter space and therefore in general support the determination of the CMSSM parameters. But in this thesis, it should be shown that the parameter determination using neural networks with the described observables alone leads to compelling results for qualitevely different regions of the parameter space, without taking any other constraints into account. Therefore it is also not important whether the considered points are already excluded. Note that for all four reference points the lightest neutralino $\tilde{\chi}_1^0$ is the LSP.

Reference point 1 has a low value for the scalar mass parameter m_0, which leads to small slepton masses, as can be seen in Table 6.1. Because of these small masses, $\tilde{\chi}_2^0$ decays with a branching ratio of around 46 % directly into a charged slepton and lepton (and with around 50 % in a sneutrino and neutrino). Furthermore, $\tilde{\chi}_1^{\pm}$ decays with around 97 % into either a charged lepton and a sneutrino or a charged slepton and a neutrino. A produced charged slepton then decays to its corresponding lepton (and normally a $\tilde{\chi}_1^0$). The decays of left–handed squarks frequently lead to the production of $\tilde{\chi}_2^0$ and $\tilde{\chi}_1^{\pm}$. Furthermore, squarks are produced in most gluino decays. Since in most initial states gluinos, left–handed squarks, $\tilde{\chi}_2^0$, $\tilde{\chi}_1^{\pm}$ or charged sleptons are produced, the events of point 1 should include a relatively high number of charged leptons. This can also be seen in Table 6.2, which lists the event distribution within the twelve distinct lepton classes introduced in Section 4.1.

Table 6.2.: Distribution of class events in percent into the twelve mutually exclusive lepton classes with their statistical errors for an integrated luminosity of 10 fb^{-1}, for the four reference points. The numbers were determined from 500 fb^{-1} of simulated data. Table taken from [2].

Class	Point 1	Point 2	Point 3	Point 4
1. $0l$	46.6 ± 2.1	47.1 ± 2.1	54.3 ± 2.2	61.2 ± 2.5
2. $1l^-$	12.2 ± 1.1	16.2 ± 1.2	15.0 ± 1.1	12.4 ± 1.1
3. $1l^+$	18.0 ± 1.3	16.9 ± 1.3	16.8 ± 1.2	15.5 ± 1.3
4. $2l^-$	1.6 ± 0.4	2.8 ± 0.5	1.7 ± 0.4	1.3 ± 0.4
5. $2l^+$	3.9 ± 0.6	3.3 ± 0.6	2.2 ± 0.4	1.9 ± 0.4
6. $l_i^+ l_i^-$	7.7 ± 0.8	4.3 ± 0.6	3.2 ± 0.5	2.8 ± 0.5
7. $l_i^+ l_{j;\ j \neq i}^-$	3.5 ± 0.6	5.0 ± 0.7	3.8 ± 0.6	2.9 ± 0.5
8. $l_i^- l_j^- l_j^+$	1.9 ± 0.4	1.4 ± 0.4	0.9 ± 0.28	0.6 ± 0.25
9. $l_i^+ l_j^+ l_j^-$	3.4 ± 0.6	1.4 ± 0.4	1.0 ± 0.3	0.7 ± 0.27
10. $l_i^- l_j^- l_{k;\ k \neq j,i \text{ for } +}^{\pm}$	0.1 ± 0.10	0.5 ± 0.22	0.3 ± 0.16	0.2 ± 0.14
11. $l_i^+ l_j^+ l_{k;\ k \neq j,i \text{ for } -}^{\pm}$	0.2 ± 0.14	0.5 ± 0.22	0.3 ± 0.16	0.3 ± 0.17
12. $4l$	0.8 ± 0.27	0.6 ± 0.24	0.5 ± 0.21	0.2 ± 0.14

Unlike the first point, reference point 2 has a high value of m_0 being much bigger than the gaugino mass parameter $m_{1/2}$. As can be seen in Table 6.1, the masses of squarks and sleptons are similar for this point with $\tilde{t}_1$ being by far the lightest sfermion. Here, the gluino (being lighter than the sfermions) decays mostly in three particles including a quark, an antiquark and either a neutralino or a chargino, i.e. gluino decays lead quite often to the production of $\tilde{\chi}_2^0$ and $\tilde{\chi}_1^{\pm}$ (The allowed gluino two–body decays into a neutralino and gluon amount only to 2 %.). In contrast to reference point 1, the decays of $\tilde{\chi}_2^0$ do not frequently lead to charged sleptons and therefore to charged leptons. Instead, $\tilde{\chi}_2^0$ decays in a two–body decay to the lightest neutralino $\tilde{\chi}_1^0$ (which is here the LSP) and either the light CP–even scalar Higgs–boson h (91 % branching ratio) or the Z–boson (9 %). Furthermore, $\tilde{\chi}_1^{\pm}$ decays always in $\tilde{\chi}_1^0$ and $W^{\pm}$. Of course, the decays of h, Z and $W^{\pm}$ should also produce some charged leptons. Since the decay of a gluino leads here on average to 0.99 top–(anti)quarks, charged leptons should also be produced from the semileptonic decays of those. But compared to reference point 1, there should be less events in higher lepton classes and in particular less events in classes with an opposite–charged lepton pair with the same flavor (classes 6, 8 and 9), because decays like $\tilde{\chi}_i^0 \to l^+ l^- \tilde{\chi}_j^0$ with $j < i$ are much less frequent. This smaller fraction of class events including an opposite–charged same flavor lepton pair shows also in Table 6.2.

Similar to point 2, reference point 3 has a relatively high m_0, which is bigger than $m_{1/2}$. But the difference is not as big as in point 2. The decay branching ratios of $\tilde{\chi}_2^0$ and $\tilde{\chi}_1^{\pm}$ are almost the same for both points. In contrast to point 2, reference point 3 has a high absolute value $|A_0|$ of the common trilinear coupling. The lightest squark is here again $\tilde{t}_1$, but this time it is lighter than the gluino (and in particular the only squark fulfilling that except for $\tilde{b}_1$, which has a very similar mass to the gluino). Therefore, the gluino decays either into a light stop and antitop or into the corresponding antiparticles. Furthermore, the light stop decays with a probability of around 60 % into a neutralino and a top–quark. That means, the decay of a gluino should lead on average to 1.6 top–(anti)quarks. Therefore, the events of point 3 may include multiple b–jets, non–b–jets as well as hard leptons, resulting from the decays of the top–(anti)quarks. However, in general there are less leptons than in reference points 1 and 2, as can be seen in Table 6.2. In particular there is almost no direct $\tilde{t}_1\tilde{t}_1^*$ pair production (less than 1 %), since the light stop is still relatively heavy.

The last reference point 4 has similar to point 1 a m_0 smaller than $m_{1/2}$, leading again to slepton masses smaller than squark masses. However, $\tan\beta$ is here relatively large with the consequence that $\tilde{\tau}_1$ has a considerable smaller mass than the other leptons, as can be seen in Table 6.1. In contrast to point 1, for the slepton masses only $m_{\tilde{\tau}_1}$ and $m_{\tilde{e}_R}$ are smaller than $m_{\tilde{\chi}_1^\pm} \simeq m_{\tilde{\chi}_2^0}$. But $\tilde{\chi}_2^0$ has only a decay branching ratio of around 0.2 % into first and second generation sleptons. On the other hand, around 77 % of the $\tilde{\chi}_2^0$ decay into either a light stau and antitau or the corresponding antiparticles. The same is true for the decay of $\tilde{\chi}_1^\pm$, which has a branching ratio of around 77 % to a light (anti)stau and (anti)tau–neutrino. Furthermore, a $\tilde{\tau}_1$ always decays into the LSP $\tilde{\chi}_1^0$ and a τ–lepton. The other possible, relevant decay for $\tilde{\chi}_2^0$ is the decay into $\tilde{\chi}_1^0$ and h with a branching ratio of around 21 %, and $\tilde{\chi}_1^\pm$ only decays otherwise in $\tilde{\chi}_1^0$ and $W^\pm$. Overall, this should lead to a considerable number of events with one or more τ–lepton(s), since the decays of strongly interacting superparticles frequently lead to the production of $\tilde{\chi}_2^0$ and $\tilde{\chi}_1^\pm$. The gluino decays with a probability of around 72 % into a third generation squark, which decays quite frequently into a top–(anti)quark. On average 0.99 top–(anti)quarks are produced in the decay of a gluino. Overall, the events may include τ– and b–jets as well as hard leptons from semileptonic decays of top–(anti)quarks and soft leptons from leptonic τ–decays.

Overall, Table 6.2 also shows that for all four reference points more leptons with positive than negative charge are produced. As mentioned before, this follows from the fact that the LHC is a proton–proton collider and the proton includes more up– than down–quarks. This leads to the production of more up– than down–squarks. The higher the squark production and the more squark decays lead to charged leptons, the higher is the number of positively charged leptons compared to negatively charged ones (under the assumption that these events pass the required cuts). Therefore, reference point 4 and in particular point 1 have a relative high positive to negative lepton ratio. On the other hand, point 2 has a smaller squark production (because of relatively high squark masses) and the decay of $\tilde{\chi}_1^\pm$ leads less frequently to a charged lepton, as described above. As a consequence the number of produced positive and negative charged leptons is almost equal for this point.

In the following, the number of events, which pass the required cuts, are listed for the four reference points. The events cuts are the same as those used for the discrimination of the degenerate pairs in the first part of this thesis. The cuts generally require a certain amount of missing transverse energy (resulting from the escaping LSP $\tilde{\chi}_1^0$) and are listed in detail in Appendix B. For an integrated luminosity of 10 fb^{-1} and specific seeds in Herwig++, the number of events before and after cuts are:

1) 1940 events before and 1080 after cuts

2) 4080 events before and 1047 after cuts

3) 1970 events before and 1135 after cuts

4) 1618 events before and 991 after cuts

Note that these numbers were determined with a simulated integrated luminosity of $500\,\mathrm{fb}^{-1}$ and scaled down to $10\,\mathrm{fb}^{-1}$ (The distributions of these events into the lepton classes were shown in Table 6.2). As desired, for all four reference points around 1,000 events pass the selection cuts. Compared to the other points, reference point 2 has much more events before cuts. The reason for that are the smaller neutralino and chargino masses, as can be seen in Table 6.1. The production cross–section is therefore dominated by chargino and neutralino production. As discussed above, the decays of $\tilde{\chi}_2^0$ and $\tilde{\chi}_1^\pm$ only rarely lead to charged leptons. The selection cuts for a low number of leptons listed in Appendix B are generally more optimized for gluino and squark production. As a consequence, most of the events do not pass the event cuts.

In the following, the training and control sets for the four reference regions are presented. The quality of them obviously determines the performance of the neural networks, i.e. in particular, as mentioned earlier, they need to be proper representations of the investigated parameter regions. The training and control sets for the four regions are therefore each picked from different parameter ranges.[†] The CMSSM parameters of a particular training or control set are then (with a flat distribution) randomly chosen from the defined parameter ranges. Furthermore, each simulation in Herwig++ is done with a different seed, which again is randomly chosen with a flat distribuion. This should prevent any possible correlations between the different training and control sets.

It is assumed that the CMSSM parameters m_0 and $m_{1/2}$ generally can be determined easier than $\tan\beta$ and A_0, since they influence the mass spectrum of the superparticles stronger. For example, the value pair (m_0, $m_{1/2}$) can already be roughly localized by only considering the total number of events after cuts. Therefore, the considered parameter ranges of these first two parameters are chosen to be smaller. This should save some computing time, since parameter sets which in general clearly can be separated from a measurement are neither simulated nor considered in the neural network learning procedure. For a fixed number of training sets, a smaller parameter range is scanned with a higher granularity than a bigger parameter range. To a certain degree, a higher granularity should lead to a smaller control error for the trained neural network, i.e. in particular a smaller error for the determined parameter. More training sets in the same parameter range increase the available information for the neural network. However, with a certain number of sets enough information is available to determine the searched function as exact as possible. For example, if the searched function is a straight line, two points would be enough for the function determination (neglecting statistical errors) and more points would not improve the result. Note that the following chosen numbers of training and control sets may not be high enough to provide the neural network learning procedure with the necessary amount of information to determine the CMSSM parameters as good as it would be possible for the considered integrated luminosity. Instead, the size of the numbers are mostly determined by the existing computing resources. In particular, this thesis should only demonstrate the general power of the neural network approach for the parameter determination and not deliver a perfect solution. Furthermore, note that it is obviously assumed that the searched CMSSM parameter values of the reference points lie within the considered parameter ranges of the corresponding neural networks. For the four reference regions the training and control sets are chosen from the following parameter ranges:

1) $m_0 : 100 - 350\,\mathrm{GeV}, \quad m_{1/2} : 660 - 740\,\mathrm{GeV}, \quad \tan\beta : 5 - 45 \quad \text{and} \quad |A_0| \leq 2 \cdot m_0$

2) $m_0 : 1850 - 2200\,\mathrm{GeV}, \quad m_{1/2} : 410 - 490\,\mathrm{GeV}, \quad \tan\beta : 5 - 45 \quad \text{and} \quad |A_0| \leq 2 \cdot m_0$

3) $m_0 : 850 - 1200\,\mathrm{GeV}, \quad m_{1/2} : 560 - 640\,\mathrm{GeV}, \quad \tan\beta : 5 - 45 \quad \text{and} \quad |A_0| \leq 2 \cdot m_0$

[†] Note that instead of creating neural networks for four different parameter regions presumably it should be possible to consider the whole CMSSM parameter space at the same time. However, this should come along with the need of simulating much more training and control sets and the creation of a neural network with many more hidden neurons. Taking into account the limited, existing computing resources, this option is not feasible.

4) $m_0 : 300 - 550\,\mathrm{GeV}, \quad m_{1/2} : 660 - 740\,\mathrm{GeV}, \quad \tan\beta : 10 - 50 \quad \text{and} \quad |A_0| \leq 2 \cdot m_0$

The discrete sign of the Higgsino parameter μ is here always positive. The common trilinear coupling A_0 is chosen from a range which depends on the picked value of m_0. The constraint $|A_0| \leq 2 \cdot m_0$ should prevent problems in the generation of the superparticle spectrum with SOFTSUSY, since the spontaneous electroweak symmetry breaking could fail. For each reference point around 1,000 training sets and 300 control sets are generated. A higher number of control sets should generally not change the final weights of the neural network significantly, because the control sets only influence the determination of the weights by setting the point of time when the network learning procedure is stopped. As long as the learning process is stopped at (nearly) the same step, also (nearly) the same weights are found. It is only important that the control sets are a good representation of the investigated parameter region. The stop of the learning procedure at roughly the same learning step for a higher number of control sets was concretely checked for an example.

On the other hand, as mentioned above, the number of training sets can influence the resulting performance of the neural network. However, as can be seen later in Section 7.1, the with the networks estimated CMSSM parameter errors are much bigger than the average distance between the CMSSM parameter values in the simulated training sets, at least for the $10\,\mathrm{fb}^{-1}$ measurements. Furthermore, at the end of Section 7.1 it is shown that the use of around one and a half times more training sets for reference point 4 does not noticeably improve the parameter determination. Therefore, the number of simulated training sets here should overall be sufficient.

6.2.2. Performance Improvement

The input values of the neural network are measured observables and therefore have statistical uncertainties. These uncertainties are not considered a priori within the neural network. However, the consideration of these uncertainties should improve the performance of the neural network significantly, as can be motivated with the following example of a network with just two input values and one output value. The example is illustrated in Figure 6.4. The first input value should have generally a relatively small uncertainty and the second one a relatively big uncertainty. In contrast, the output value should not have an error within the training sets (as it is the case here for the CMSSM parameter sets). This means that for a specific output value the first input could take values from a much smaller range than the second input. Therefore, for the determination of the output value, the exact value of the first input would be more meaningful than for the second output. A neural network which knows that should perform better than a network which does not.

In the following, three different measures are done to take the uncertainties of the measured observables into account. First, the parameter sets within the training and control sets are simulated with a higher luminosity, i.e. $500\,\mathrm{fb}^{-1}$ instead of $10\,\mathrm{fb}^{-1}$ for the considered measurements. The higher amount of data reduces the (relative) uncertainties of the measured observables (proportional to one over the square–root of the number of simulated events). Since the training and control sets are used to teach the neural network the searched function between the observables and the CMSSM parameters, this reduces the uncertainty of this function. In particular it is desirable that the neural network function uncertainty is (much) smaller than the uncertainty of the considered measurement. Since the measured total number of events after cuts obviously increases with higher luminosity, this has to be taken into account by adapting the input normalization. If a $10\,\mathrm{fb}^{-1}$ measurement is considered, the mean value of the total number of events after cuts over all training sets within the input normalization has to be divided by 50 and the corresponding variance by 2,500. Note that a measurement with an arbitrary integrated luminosity lower than $500\,\mathrm{fb}^{-1}$ can be considered by changing the input normalization appropriately. The consideration of a measurement with higher luminosity is unreasonable, since the errors of the training sets would be higher than of the measurement and distort the errors of the

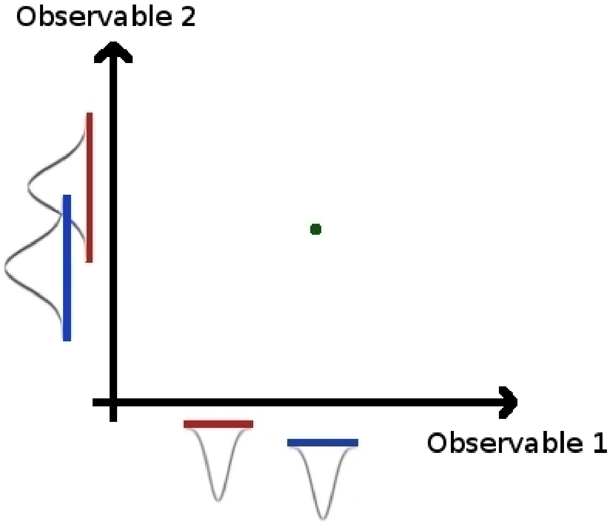

Figure 6.4.: Observable 1 (input 1) has a generally smaller uncertainty than observable 2 (input 2). The red inner lines show the possible ranges for both observables for a specific value of a single considered CMSSM parameter (output). Note that the distribution of the observable values within the shown ranges would be Gaussian as indicated. For another value of the parameter, the observables take values from the blue outer lines. The green dot in the middle represents an existing measurement. Because of the relatively big uncertainty of observable 2, the measured value could not be uniquely assigned to the first or second value of the CMSSM parameter. On the other hand, the small uncertainty of observable 1 permits an assignment to the second (blue) value of the parameter.

found parameters. Already considering a measurement with a similar integrated luminosity than the training and control sets affects the determined parameter errors, as can be seen later in Section 7.1. For such a measurement it would be desirable to use training and control sets with a (much) higher integrated luminosity than $500\,\text{fb}^{-1}$, but this is not feasible for the existing computing resources. The determination of the network output value errors is described in the following Subsection 6.2.3.

Second, similar to the χ^2–calculation done for the comparison of two data sets in Chapter 5, minimal numbers of (class) events are required for the consideration of a specific input observable. The observables described in Chapter 4 are each based on a certain number of events. The lower the number of these events is, the higher is the (relative) uncertainty of the observable. A high enough number of events ensures approximate Gaussian statistics for the observable. In contrast to the χ^2–calculation, the number of input neurons for the neural network is fixed and therefore values for all observables have to be considered to calculate the corresponding output values. A solution is to set the normalized input value of a specific observable, which does not fulfill the required number of events, to zero. Such a zero would not contribute to the weighted sums calculated within the neural network, see equations (6.1) and (6.2). The required minimal number of (class) events is set to 50 for the total number of events after cuts and to 500 for all other observables, considering the training and control sets with an integrated luminosity of $500\,\text{fb}^{-1}$. For a $10\,\text{fb}^{-1}$ measurement the minimal numbers would be one and ten, respectively. The dependence of the minimal numbers on the integrated luminosity should ensure that for a specific parameter set the same normalized observables are set to zero, independent of the amount of data. Of course, because of statistical fluctuations of the (class) events, this is not always fulfilled.

Third, for each of the approximately 1,000 training sets 100 Gaussian distributed copies are generated. This leads to approximalely 100,000 training sets. For each training set the covariance matrix, see Section 4.2, is known. This covariance matrix can be used to create a 84–dimensional (correlated) Gaussian distribution with the measured observables as mean values. The set–up of the distribution is shown in Appendix D.4. In the following, 100 sets are chosen with this distribution. All these training sets have the same CMSSM output values, but different Gaussian distributed input values.[‡] As described in the example above, there are input values with smaller and bigger value ranges. Since all input values lead to the same output values, the input values with smaller variances are more important for the parameter determination. In contrast, during the learning process the network would be confronted with highly varying input values (for observables with higher variances) and recognize their irrelevance (less importance) for the determination of the output values. As a consequence, the values of the weights for the connections with the corresponding input neurons would be reduced. On the other hand, the weights for input values with smaller uncertainties would be (in comparison) higher, making these input values more important for the parameter determination. Note that the creation of Gaussian distributed copies confronts the neural network with the existing correlations of the measured observables. Overall, within the learning process this measure should make the network aware of the observable uncertainties relative to each other. In particular the relative observable uncertainties should not change for different integrated luminosities. This ensures that the trained network can sensibly be used for a measurement with a smaller luminosity than the training and control sets.

Furthermore, another measure to improve the performance of the parameter determination, next to the consideration of statistical uncertainties, is the creation of four independent neural networks with each one CMSSM parameter output instead of the creation of one network with four outputs. Note that all four networks still each have $84 + 1$ input neurons. The four networks each should be smaller, i.e. include less hidden neurons, than one combined network and their learning procedure should therefore each consume less time. Furthermore, each network can be (easier) specialized for the particular CMSSM parameter, which could allow a better parameter determination. In particular the minimum of the total normalized control error in equation (6.9) for a network with four output values does not necessarily coincide with the minimum of the control error for each single output parameter.

6.2.3. Parameter Errors

The trained neural network determines the CMSSM parameter value which belongs to a particular measurement. At this point of time the error of the determined parameter is not known. In the following, two different ways are described how to determine the parameter error and also the correlations between the four found CMSSM parameters. In both cases the covariance matrix of the measurement, as introduced in Section 4.2, is used.

The first possibility is to generate Gaussian distributed measurement copies similar to the training set copies described in Subsection 6.2.2. Each of the generated copies is used as an input for the considered neural network. The computed network outputs should form a Gaussian distribution. The mean value of this distribution is the found parameter value and the square–root of the variance of the distribution gives the corresponding parameter error. Figure 6.5 shows the $m_{1/2}$–distribution for reference point 3, which is created by feeding the appropriate neural network with 100,000 Gaussian distributed copies of the $10\,\mathrm{fb}^{-1}$ measurement. The found mean value with standard deviation is $m_{1/2} = (607.4 \pm 9.3)\,\mathrm{GeV}$ and is consistent with the searched value $m_{1/2} = 600\,\mathrm{GeV}$. Such an error

[‡] Note that the use of Gaussian distributed training set copies generally prevents that the neural network memorizes all training sets (i.e. the control error equals zero), since 100 different sets of input values each need to lead to the same output values. However, in practice this is irrelevant because the learning procedure should anyway be stopped before such a memorization happens.

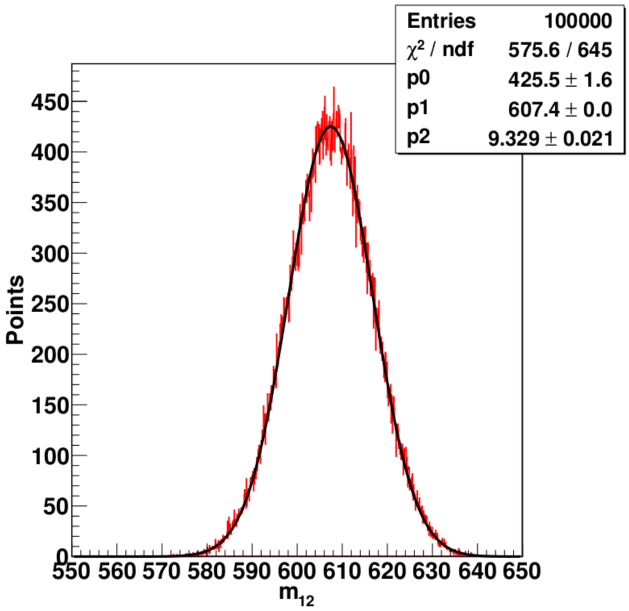

Figure 6.5.: One–dimensional $m_{1/2}$–distribution in GeV formed by feeding a neural network with 100,000 Gaussian distributed 10 fb^{-1} measurement copies of reference point 3. The fitted Gaussian distribution has the form $g(m_{1/2}) = p_0 \cdot \exp(-1/2 \cdot [(m_{1/2} - p_1)/p_2]^2)$ with p_1 being the mean value and p_2 being the standard deviation. The neural network consists of 20 hidden neurons and underwent 107 learning steps with a final control error of 0.136. The searched value is $m_{1/2} = 600$ GeV. Figure taken from [2].

determination can be done for all four neural networks for the investigated CMSSM parameter region to obtain the parameters with errors for a specific measurement. However, the four single determinations do not give the covariances between the found CMSSM parameters.

The covariances between two different CMSSM parameters are determined by forming a two–dimensional Gaussian distribution. As before, Gaussian measurement copies are generated. Each copy is put into the two neural networks which belong to the considered CMSSM parameters. The two output values are saved as a pair for each measurement copy. The computed output pairs should form a two–dimensional (correlated) Gaussian distribution. From this distribution the variances and the covariance of both CMSSM parameters can be determined. The resulting variances (up to rounding errors because of different binning) should be the same as the ones determined from the single distributions described above. Figure 6.6 shows the two–dimensional distribution for the parameters m_0 and $m_{1/2}$ for reference point 3, which is generated by feeding the appropriate neural networks each with 1,000,000 Gaussian distributed 10 fb^{-1} measurement copies. The mean values with corresponding standard deviations for the two–dimensional distribution are $m_0 = (1034 \pm 42.3)$ GeV and $m_{1/2} = (607.5 \pm 9.3)$ GeV and are in agreement with the searched values $m_0 = 1000$ GeV and $m_{1/2} = 600$ GeV. As mentioned, the values for $m_{1/2}$ are the same as the ones determined with the one–dimensional distribution from before.

The covariance $\mathrm{cov}(m_0, m_{1/2})$ between m_0 and $m_{1/2}$ can be expressed with the correlation coefficient $\rho_{m_0\, m_{1/2}}$ with

$$\rho_{m_0\, m_{1/2}} = \frac{\mathrm{cov}(m_0,\, m_{1/2})}{\sigma_{m_0}\, \sigma_{m_{1/2}}} \,. \tag{6.11}$$

Here, σ_{m_0} and $\sigma_{m_{1/2}}$ are the standard deviations of m_0 and $m_{1/2}$, respectively. The parameters are

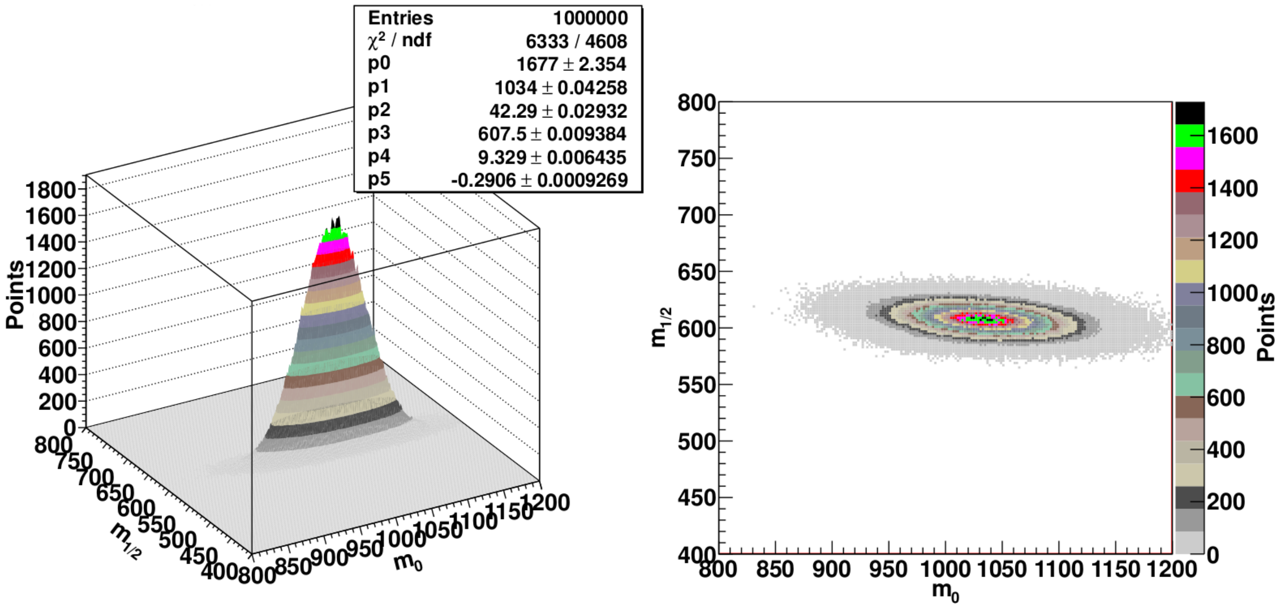

Figure 6.6.: Two–dimensional distribution of the parameters m_0 and $m_{1/2}$ in GeV formed by feeding both neural networks with 1,000,000 Gaussian distributed $10\,\mathrm{fb}^{-1}$ measurement copies of reference point 3. The fitted Gaussian distribution has the form $g(m_0,\, m_{1/2}) = p_0 \cdot \exp[-0.5/(1-p_5^2)\cdot([(m_0-p_1)/p_2]^2+[(m_{1/2}-p_3)/p_4]^2-2\cdot p_5/(p_2\,p_4)\cdot(m_0-p_1)\cdot(m_{1/2}-p_3))]$ with p_1 and p_2 being the mean value and standard deviation of m_0, and p_3 and p_4 the mean value and standard deviation of $m_{1/2}$. The correlation factor p_5 is given by equation (6.11). The neural network for m_0 consists of 15 hidden neurons and underwent 244 learning steps with a final control error of 0.106, and the one for $m_{1/2}$ consists of 20 hidden neurons and underwent 107 learning steps with a final control error of 0.136. The searched values are $m_0 = 1000\,\mathrm{GeV}$ and $m_{1/2} = 600\,\mathrm{GeV}$. Figure taken from [2].

negatively correlated with $\rho_{m_0\,m_{1/2}} = -0.291$, as can be seen on the left side of Figure 6.6. This correlation is pictured in the tilting of the ellipse formed at the bird's eye view of the two–dimensional Gaussian distribution, as seen on the right side of Figure 6.6. The major ellipse axis angle ϕ (which is zero for a horizontal ellipse with no correlation) is related to the correlation coefficient with

$$\tan(2\phi) = \frac{2\,\rho_{m_0\,m_{1/2}}\,\sigma_{m_0}\,\sigma_{m_{1/2}}}{|\sigma^2_{m_0} - \sigma^2_{m_{1/2}}|}\,. \tag{6.12}$$

The negative correlation shows in the tilting of the ellipse with $\phi = -3.84\,^\circ$ to the right bottom.

The second way for the determination of the variances and covariances of the CMSSM parameters is the propagation of uncertainty method (Gaussian error propagation). With the neural network functions and the covariance matrix of the measured observables, the variances and covariances can directly be calculated. For the CMSSM parameter r with $r \in \{m_0,\, m_{1/2},\, \tan\beta,\, A_0\}$ the variance is calculated with

$$\sigma_r^2 = \sum_{i\in E}\left(\frac{\partial f_r^*}{\partial O_i}\,\sigma(O_i)\right)^2 + 2\cdot\sum_{i\in E\setminus e}\,\sum_{j>i,\in E}\frac{\partial f_r^*}{\partial O_i}\,\frac{\partial f_r^*}{\partial O_j}\,\mathrm{cov}(O_i, O_j)\,. \tag{6.13}$$

In this formula, $\sigma(O_i)$ stands for the standard deviation of the input observable O_i and $\mathrm{cov}(O_i, O_j)$

for the covariance between the observables O_i and O_j. The sums run over all observables which fulfill the minimal number of events mentioned in Subsection 6.2.2. The coefficients of these observables are summarized in the set E with $e \leq 84$ being the highest coefficient number. The neural network function f_r^* for the CMSSM parameter r has the measured observables as input values and the CMSSM parameter as output value. The derivate with respect to a specific observable O_i is

$$\begin{aligned}\frac{\partial f_r^*}{\partial O_i} &= \frac{\partial x_i}{\partial O_i}\,\frac{\partial f_r^*}{\partial f_r}\,\frac{\partial f_r}{\partial x_i} = \frac{\sigma_r^{tg}}{\sigma_i^{tg}} \cdot \frac{\partial f_r}{\partial x_i} \\ &= \frac{\sigma_r^{tg}}{\sigma_i^{tg}} \cdot \left[\sum_{a=1}^{v} w_{ra}^{(2)}\, w_{ai}^{(1)} \left(1 - \tanh^2(\sum_{j=1}^{84} w_{aj}^{(1)}\, x_j + w_{a0}^{(1)})\right)\right] . \end{aligned} \tag{6.14}$$

The function $f_r^*(O_1, \ldots, O_{84})$ with $f_r^* = r$ is actually not the trained neural network, because the neural network is created for normalized input and inversely normalized output values. The corresponding real network function is labeled $f_r(x_1, \ldots, x_{84})$ and written down in equation (6.2) in the sense $f_r = y_r$ for a network with one inversely normalized CMSSM parameter output. Switching between both network functions comes along with the standard deviations over the training sets for the input values, σ_i^{tg}, and the output value, σ_r^{tg}, as defined in equations (6.4) and (6.7). In the second line of equation (6.14), the derivative of f_r with respect to the normalized input value x_i is calculated using equations (6.1) and (6.2). Note that the normalized input values x_i, which do not fulfill the required minimal event numbers, are set to zero and do not add in the sum within the tangens hyperbolic function.

Furthermore, the covariance cov(r, s) between two CMSSM parameters r and s is calculated using both neural network functions with

$$\operatorname{cov}(r,\, s) = \sum_{i \in E} \frac{\partial f_r^*}{\partial O_i}\,\frac{\partial f_s^*}{\partial O_i}\,\sigma^2(O_i) + \sum_{i \in E \setminus e}\ \sum_{j > i, \in E} \operatorname{cov}(O_i,\, O_j) \cdot \left(\frac{\partial f_r^*}{\partial O_i}\frac{\partial f_s^*}{\partial O_j} + \frac{\partial f_r^*}{\partial O_j}\frac{\partial f_s^*}{\partial O_i}\right) . \tag{6.15}$$

Both described ways for the determination of the variances and covariances of the CMSSM parameters should lead to the same results within statistical uncertainties.

6.2.4. Number of Hidden Neurons

As mentioned in Subsection 6.1.1, the number of hidden neurons determines the number of free parameters of the neural network function. A higher number of hidden neurons enables the imitation of a more complex function, but generally comes along with a longer learning procedure. This follows from the fact that the learning procedure includes the repeated calculation of a matrix with dimension $W = 86{\cdot}v{+}1$ for a neural network with 84 input values and one output value, as described in Appendix C. As a consequence, for a fixed number of learning steps the training time scales with the squared number of hidden neurons v^2. Therefore, the number of hidden neurons should not be increased without an expectable benefit.

An appropriate number of hidden neurons for a considered neural network is estimated by starting with a low number of hidden neurons (like ten) and training the network for some number of steps. Every other step when a local minimum is reached, the performance of the network is examined by looking at the form of the output distribution which is created by feeding the network with the Gaussian distributed input values (of the measurement). This output distribution should follow a Gaussian distribution as well. If the form is (roughly) Gaussian the training is continued. Otherwise, the number of hidden neurons is increased by like five and the new network is trained. Each additional neuron leads to an extra number of $85 + 1 = 86$ free parameters for a neural network with 84 input neurons (plus the zeroth input neuron) and one output neuron. The number of hidden neurons is increased until a satisfying result is reached.

Note that this method most likely does not lead to the optimal number of neurons which should be as low as possible, whereas the network has a minimal final control error and a Gaussian formed output. For example, the training could be stopped too early before a Gaussian output distribution is reached for a certain number of hidden neurons. Since the number of hidden neurons is normally increased by a few at a time, the optimal network could also be missed. Furthermore, a network with a few more hidden neurons than the one found to have a relatively satisfying Gaussian distribution may lead to a little bit nicer Gaussian output and somewhat smaller control error. However, overall this approach should lead to a neural network which is not too far off the optimal solution for the given network layout.

7. Results of the CMSSM Parameter Determination

In this chapter, the results for the CMSSM parameter determination at the LHC using neural networks are presented. This includes the results for the four different reference points, which are introduced in Subsection 6.2.1, for different integrated luminosities. It is shown that both methods mentioned in Subsection 6.2.3 for the calculation of the variances and covariances of the parameters lead to consistent results. For reference point 4, neural networks with one and a half times more training sets are created to compare their performances to the usual networks. It is also checked, how much the creation of Gaussian distributed training set copies improves the performances of the neural networks. Furthermore, the neural networks created for reference region 4 additionally are fed with a different measurement than the one for reference point 4 to verify that the neural networks work generally for arbitrary measurements within an investigated region. Afterwards, another method for the determination of the CMSSM parameters of a measurement is introduced. This method is based on a χ^2–minimization. The results for reference point 4 for this method are compared to the corresponding neural network results. The main content of this chapter is published in [2].

Table 7.1.: Overview about the used neural networks with number of hidden neurons, number of learning steps and resulting control errors. The control error is defined in equation (6.9). Table taken from [2].

		Hidden Neurons	Learning Steps	Control Error
Reference Point 1	m_0	20	1009	0.101
	$m_{1/2}$	20	407	0.081
	$\tan\beta$	20	653	0.135
	A_0	20	561	0.532
Reference Point 2	m_0	20	929	0.123
	$m_{1/2}$	15	452	0.116
	$\tan\beta$	20	499	0.387
	A_0	25	155	0.711
Reference Point 3	m_0	15	244	0.106
	$m_{1/2}$	20	107	0.136
	$\tan\beta$	20	240	0.681
	A_0	15	763	0.405
Reference Point 4	m_0	22	488	0.156
	$m_{1/2}$	22	637	0.067
	$\tan\beta$	25	511	0.178
	A_0	25	391	0.310

7.1. Neural Network Results

In this section, the results of the parameter determination at the LHC for the four CMSSM reference regions using neural networks are described. Table 7.1 shows the number of hidden neurons used for the 16 networks and the control errors resulting at the end of the learning processes after the given numbers of learning steps. The size of the control error estimates the performance of the neural network in the investigated parameter region, meaning that a network with a smaller control error tends to determine the appropriate parameter with a smaller uncertainty.[*]

The created networks contain between 15 and 25 hidden neurons. The networks for m_0 and $m_{1/2}$ tend to need slightly less hidden neurons than the networks for $\tan\beta$ and A_0. The networks undergo between 100 and 1,000 learning steps before the global minimum of the control error is reached. To verify the position of the global minimum, the error evolution is usually checked for around 100 steps after the minimum. The training of the neural networks often took several months on one processor. This long computation time arises from the repeated calculation of the Hessian matrix with dimension $W = 86 \cdot v + 1$ for the approximately 100,000 training sets in each learning step, as described in detail in Appendix C. However, in particular for the neural networks with a high number of learning steps the control error usually decreases very slightly in the last hundreds of steps. Therefore, for these networks almost as good results for the determination of the CMSSM parameters can be reached with much less computational effort.

For all reference regions the CMSSM parameters m_0 and $m_{1/2}$ are generally determined (much) better, i.e. have smaller control errors, than the other two continuous parameters $\tan\beta$ and A_0. As mentioned earlier, this result could be expected from the much stronger influence of these first two parameters on the superparticle spectrum. Except for reference point 3, the control errors for $m_{1/2}$ are smaller than the ones for m_0. As described in Subsection 6.2.1, reference point 3 has a relatively high m_0, which is bigger than $m_{1/2}$, leading to squark masses which are not much bigger than the slepton masses (in contrast to points 1 and 4). Note that the difference of m_0 for points 1 and 4 is 250 GeV with the same value for $m_{1/2}$, but the squark masses are very similar, as can be seen in Table 6.1. For point 3, a change of m_0 has a much bigger influence on the squark masses and therefore can be easier noticed in a change of the observables. However, m_0 is not as big as in point 2, where the squark masses are much bigger than the gluino mass and as a consequence the production of one or two squarks is much less frequent than gluino production. The combination of a relatively high squark production with a relatively strong influence of m_0 on the squark masses allows therefore for reference point 3 a better determination of this CMSSM parameter. Altogether, the smallest control error is found for the $m_{1/2}$ neural network of point 4 with 0.067.

The control errors for $\tan\beta$ are smaller than the ones for A_0 for all reference points, expect for point 3 again. Compared to the other points, reference point 3 has a relatively large A_0. As described in Subsection 6.2.1, the light stop $\tilde{t}_1$ is relatively light and lighter than the gluino. In particular all gluinos decay into $\tilde{t}_1$. Furthermore, the mass and branching ratios of $\tilde{t}_1$ depend only weakly on the value of $\tan\beta$ for the considered values, as can e.g. be seen for $m_{\tilde{t}_1}$ in Table 6.1, comparing points 1

[*] The control error for a particular neural network in Table 7.1 is determined by starting with specific values for the first weights. Note that different first weight values could lead to a somewhat different final value of the control error. The different final weights would then obviously also lead to a somewhat different error of the determined CMSSM parameter. Furthermore, the final weights follow a similar Gaussian distribution than the first weights. As mentioned in Subsection 6.1.2, in the beginning, the first layer weights follow a Gaussian distribution around zero with variance 1/85. Except for a few outliers, also the final neural network weights for the first layer follow such a distribution with a very similar (or almost equal) variance. Because of the small statistics this can not be checked for the second weight layer, although the value ranges agree. At the end of the learning procedure, the distribution of the weight values obviously does not significantly change. However, it is crucial how the weights are distributed on the single neuron connections.

and 4. The weak $\tan\beta$ dependence of $\tilde{t}_1$ and the full decay of gluinos into $\tilde{t}_1$ leads to the high control error for $\tan\beta$ for reference region 3, the highest of all four regions. The better determination of A_0 compared to $\tan\beta$ for point 3 follows therefore in particular from the especially bad determination of $\tan\beta$. Overall, the biggest control error is given with 0.711 for A_0 of point 2. This is more than ten times bigger than the smallest control error for $m_{1/2}$ of point 4.

In comparison, reference points 1 and 4 show in total the best results for all four CMSSM parameters. As described in Subsection 6.2.1, both points have relatively light sleptons. For point 1, the decays of gluinos and squarks lead very frequently to the production of charged sleptons. Since the sleptons decay into their corresponding charged leptons, there are relatively many events with multiple leptons, i.e. there should be more observables (also in higher lepton classes) being sensitive on CMSSM parameter changes (which are then responsible for mass and branching ratio changes). For point 4, most $\tilde{\chi}_1^{\pm}$ and $\tilde{\chi}_2^0$ decay into $\tilde{\tau}_1$, which leads to the production of a relatively high number of τ–leptons. This leaves distinctive information in the τ–observables (in addition to the information in the other observables), which should give extra sensitivity to CMSSM parameter changes.

Table 7.2.: Found CMSSM parameters, when the appropriate neural networks are fed with the 10 fb^{-1} or 500 fb^{-1} measurements of the reference points. The standard deviations and the correlation coefficients are calculated with the propagation of uncertainty method, see equations (6.13), (6.14) and (6.15). The unit for m_0, $m_{1/2}$ and A_0 is GeV. Table taken from [2].

	Point 1	Point 2	Point 3	Point 4
	10/fb			
$\overline{m}_0 \pm \sigma_{m_0}$	167.25 ± 33.78	1998.93 ± 92.20	1055.97 ± 47.26	482.94 ± 61.23
$\overline{m}_{1/2} \pm \sigma_{m_{1/2}}$	697.51 ± 7.36	446.55 ± 11.30	607.47 ± 11.53	695.48 ± 7.87
$\overline{\tan\beta} \pm \sigma_{\tan\beta}$	21.35 ± 5.96	9.67 ± 18.81	23.41 ± 37.42	23.37 ± 11.08
$\overline{A}_0 \pm \sigma_{A_0}$	463.43 ± 326.00	1406.37 ± 2898.67	1453.49 ± 1891.58	-73.52 ± 628.16
$\rho_{m_0\,m_{1/2}}$	-0.037	-0.051	-0.309	-0.407
$\rho_{m_0\,\tan\beta}$	0.131	-0.068	0.044	0.057
$\rho_{m_0\,A_0}$	0.021	-0.251	0.200	-0.671
$\rho_{m_{1/2}\,\tan\beta}$	0.201	0.174	-0.084	-0.350
$\rho_{m_{1/2}\,A_0}$	0.121	-0.119	0.065	0.139
$\rho_{\tan\beta\,A_0}$	-0.035	0.249	-0.218	0.238
	500/fb			
$\overline{m}_0 \pm \sigma_{m_0}$	156.23 ± 4.86	2004.05 ± 10.61	1015.66 ± 4.49	391.86 ± 7.70
$\overline{m}_{1/2} \pm \sigma_{m_{1/2}}$	701.05 ± 0.87	451.15 ± 1.00	598.75 ± 1.21	700.71 ± 0.88
$\overline{\tan\beta} \pm \sigma_{\tan\beta}$	9.39 ± 0.70	14.66 ± 2.17	20.79 ± 5.20	30.50 ± 1.24
$\overline{A}_0 \pm \sigma_{A_0}$	-43.61 ± 55.37	261.47 ± 474.68	774.39 ± 295.63	183.05 ± 101.48
$\rho_{m_0\,m_{1/2}}$	-0.125	-0.519	-0.550	-0.695
$\rho_{m_0\,\tan\beta}$	-0.233	-0.131	0.294	0.441
$\rho_{m_0\,A_0}$	0.038	-0.068	-0.557	-0.232
$\rho_{m_{1/2}\,\tan\beta}$	0.259	0.076	-0.110	-0.416
$\rho_{m_{1/2}\,A_0}$	0.033	0.165	0.509	0.250
$\rho_{\tan\beta\,A_0}$	-0.211	-0.158	-0.055	0.464

The values of the normalized control errors in Table 7.1 cannot directly be compared for different CMSSM parameters and reference points without taking into account the sizes of the considered parameter ranges within the control sets. As defined in equation (6.9), the control error scales inversely with the size of the investigated parameter region. The in Subsection 6.2.1 introduced training and control regions cover always 80 GeV in $m_{1/2}$ and between 250 and 350 GeV in m_0. This means that the same control error in $m_{1/2}$ and m_0 expresses an accordingly smaller relative error of the network output for $m_{1/2}$. A good part of the theoretically allowed parameter ranges of $\tan\beta$ and A_0 are covered, whereas the size of the considered A_0 region depends on the parameter value of m_0. As a consequence, the relative errors of the determined parameters $m_{1/2}$ and m_0 should be, compared to $\tan\beta$ and A_0, much smaller than the control errors in Table 7.1 would suggest without taking the parameter ranges into account. Further details on the performance differences between the created neural networks can be found in [2].

Finally, Table 7.2 contains the determined CMSSM parameters for the four reference point measurements, using the appropriate neural networks. Measurements with an integrated luminosity of 10 fb^{-1} and 500 fb^{-1} are considered. The standard deviations and correlations of the CMSSM parameters are calculated by using the propagation of uncertainty method described at the end of Subsection 6.2.3. The determined standard deviations and the normalized control errors from Table 7.1 can be related to each other using the sizes of the investigated parameter ranges. The ratio of the standard deviation divided by the product of the control error and the parameter range size is for all simulated 10 fb^{-1} (500 fb^{-1}) measurements distributed around 1.2 ± 0.4 (0.15 ± 0.04). This means that the standard deviation of a found CMSSM parameter can be approximately estimated by multiplying the control error and the investigated parameter range size (for A_0 the average size is used) with this factor. The biggest deviations from this relation for 10 fb^{-1} integrated luminosity are m_0 and A_0 of reference point 2, where the factors are 2.1 and 0.5. However, the exact same factor for all found parameter values is not expected, since the control error expresses only the average performance over all approximately 300 considered control sets. Therefore, the performance can obviously differ in different regions of the parameter space.

For the 10 fb^{-1} measurements, 12 out of 16 found parameter values are within one estimated standard deviation from the searched value, as can be checked in Table 7.2. The other four values are between one and two standard deviations away from the true value. Within statistics this coincides well with the expectation of $0.6827 \cdot 16 \approx 10.92$ for the 1σ–interval and ≈ 4.35 for the 1 to 2σ–interval. On the other hand, for the 500 fb^{-1} measurements ten found parameters differ more than one standard deviation from the true value, four of these values more than two and even one (m_0 of point 3) more than three standard deviations. This follows most likely from the fact that the training and control sets also “only” have an integrated luminosity of 500 fb^{-1}, i.e. the uncertainty of the neural network function should not be negligible for a measurement with a similar (here the same) luminosity. An inclusion of the network uncertainty then would lead to an increase of the determined parameter errors. With these corrected errors, the deviations between searched and true parameter values should very probably follow the expectation.

Furthermore, the 500 fb^{-1} measurement for reference point 3 seems to be especially far off the expectation, since all found parameter values are more than one estimated standard deviation away from the true values, three of those more than two and one more than three. However, the results for two other generations of the same measurement with different seeds in Herwig++ mitigated this extreme deviation. For both additional versions, the found value for m_0 is less than one estimated standard deviation away from the searched value, in contrast to the more than three standard deviations of the original version in Table 7.2. Therefore, this seems to be a rather extreme statistical fluctuation. Only the estimated value of A_0 is for all three versions always more than two (but still less than three) standard deviations away from the true value. Furthermore, the found value is always smaller than the searched value 1500 GeV. This slight tendency to smaller values may be caused by the circumstance that the searched value lies quite close to the upper end of the considered parameter range of A_0 for reference point 3.

The underestimation of the parameter errors for an integrated luminosity of $500\,\mathrm{fb}^{-1}$ can also be seen in the ratio of the errors for different luminosities. Including only statistical uncertainties, the parameter error for a $500\,\mathrm{fb}^{-1}$ measurement should be by around a factor of $\sqrt{50} \approx 7$ smaller than the one for an integrated luminosity of $10\,\mathrm{fb}^{-1}$. Note that there is also some statistical uncertainty on the standard deviation which obviously also should decrease with increasing luminosity. However, for all four reference points the standard deviations for $m_{1/2}$, m_0 and $\tan\beta$ decrease on average by a factor of 9.6, 8.5 and 8.3 which is systematically higher than the factor 7. This agrees with the already seen underestimation of the $500\,\mathrm{fb}^{-1}$ error. Note that the factor increases with a better relative determination of the parameters. Since the parameter $m_{1/2}$ can be determined especially well, the non–consideration of the neural network function error for the $500\,\mathrm{fb}^{-1}$ measurements is for this parameter more noticable.

On the other hand, for A_0 the error decreases on average only by a factor of 6.2 going from the $10\,\mathrm{fb}^{-1}$ to the $500\,\mathrm{fb}^{-1}$ measurements. This behavior may be caused by the fact that for three of the four $10\,\mathrm{fb}^{-1}$ measurements the estimated 1σ–intervals lie already partly outside of the considered parameter ranges with $|A_0| \leq 2m_0$. As mentioned earlier in Subsection 6.1.1, the neural network function is only reliable within the investigated parameter space. Therefore, the $10\,\mathrm{fb}^{-1}$ errors for A_0 are probably not fully reliable. Furthermore, because of the above mentioned reasons, the $500\,\mathrm{fb}^{-1}$ errors might still be underestimated.

However, overall the neural networks deliver quite satisfactory results. For an integrated luminosity of $10\,\mathrm{fb}^{-1}$, the common gaugino mass $m_{1/2}$ can already be determined with a relative accuracy between 1 and 2.5 %. As the normalized control errors in Table 7.1 already suggested, points 1 and 4 have the smallest relative errors for $m_{1/2}$ (Recall that all four regions have the same $m_{1/2}$ parameter range size). The parameter m_0 has for reference points 2 and 3 accuracies of around 4.5 %. As mentioned earlier, for these points m_0 is high enough to significantly affect the squark masses. For smaller values of m_0 the relative uncertainty increases, up to approximately 20 % for point 1. However, point 1 has the smallest absolute error for m_0 of all four measurements.

As can be seen in Table 7.2, $10\,\mathrm{fb}^{-1}$ of data with around 1,000 events after cuts is generally not enough to allow a meaningful determination of $\tan\beta$ and especially of A_0 (taking here in particular the considered parameter region with $|A_0| \leq 2m_0$ into account). The smallest relative error for $\tan\beta$ with respect to the found value is with 28 % for point 1 still quite high. However, increasing the integrated luminosity to $500\,\mathrm{fb}^{-1}$ leads to estimated relative accuracies as small as 4 % for point 4. For reference point 3, $\tan\beta$ is determined the worst with a relative error of 25 %, which was already seen from the control error in Table 7.1. The sizes of the estimated standard deviations for the common trilinear coupling parameter A_0 for $500\,\mathrm{fb}^{-1}$ of data are roughly 25 % to 35 % compared to the true value of m_0.

Table 7.2 also includes the correlation coefficients between the CMSSM parameters, as defined in equation (6.11). The true values for the correlation coefficients should be independent of the integrated luminosity. However, the estimated values obviously vary because of statistical fluctuations, as can be seen in Table 7.2. Note that some variation can also occur, if not the same lepton classes fulfill the required minimal event numbers for the different integrated luminosities and therefore not the same observables are considered in the calculation of the variances and covariances. This is explained in more detail below. Overall, most correlations are quite weak. E.g. for all four reference points, there is a negative correlation between m_0 and $m_{1/2}$. The increase of either m_0 or $m_{1/2}$ should lead to higher masses for the strongly interacting superparticles. This should come along with a smaller total number of produced events and a higher average value of H_T within the events. Therefore, the increase of m_0 should to some extent be compensated by the decrease of $m_{1/2}$, and vice versa. However, for points with m_0 much smaller than $m_{1/2}$, the gluino and squark masses are quite independent from the value of m_0. As mentioned earlier, this can be seen in Table 6.1 for reference points 1 and 4, whose values of

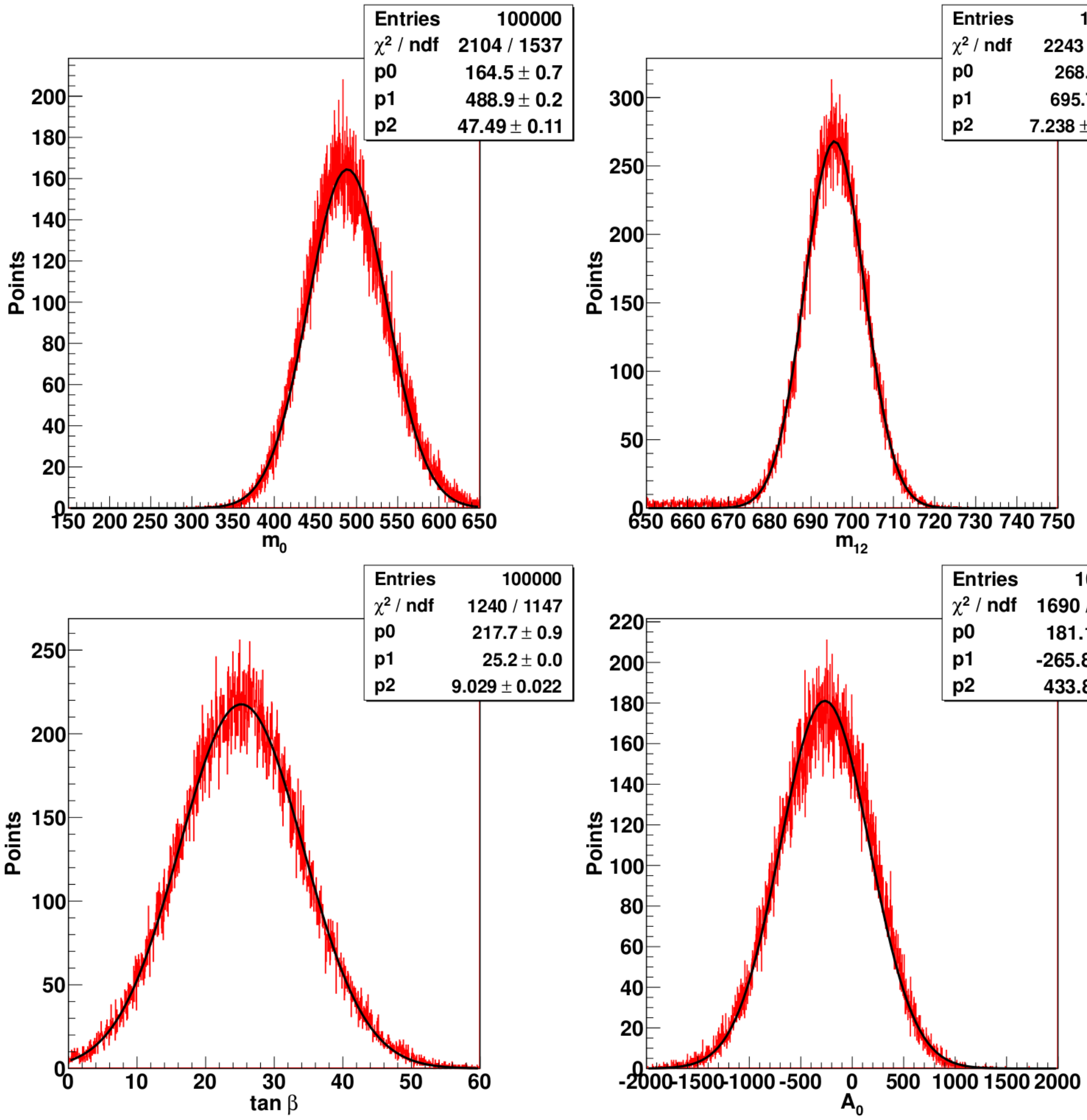

Figure 7.1.: One–dimensional Gaussian distributions for reference point 4, achieved by feeding the neural networks each with 100,000 Gaussian distributed $10\,\text{fb}^{-1}$ measurement copies. The unit for m_0, $m_{1/2}$ and A_0 is GeV. The fitted Gaussian distributions have each the form $g(x) = p_0 \cdot \exp(-1/2 \cdot [(x - p_1)/p_2]^2)$ with x being the appropriate CMSSM parameter, p_1 the mean value and p_2 the standard deviation. The searched values are located in the middle of the lower axes, i.e. $m_0 = 400\,\text{GeV}$, $m_{1/2} = 700\,\text{GeV}$, $\tan\beta = 30$, and $A_0 = 0\,\text{GeV}$. The neural network settings are given in Table 7.1. Figure taken from [2].

m_0 differ by 250 GeV. Therefore, the negative correlation of m_0 and $m_{1/2}$ for point 1 is much weaker compared to the other points. Further details on the correlations can be found in reference [2].

As claimed in Subsection 6.2.3, both methods for the determination of the CMSSM parameter errors should lead to the same results within statistical uncertainties. So far, only the results for the

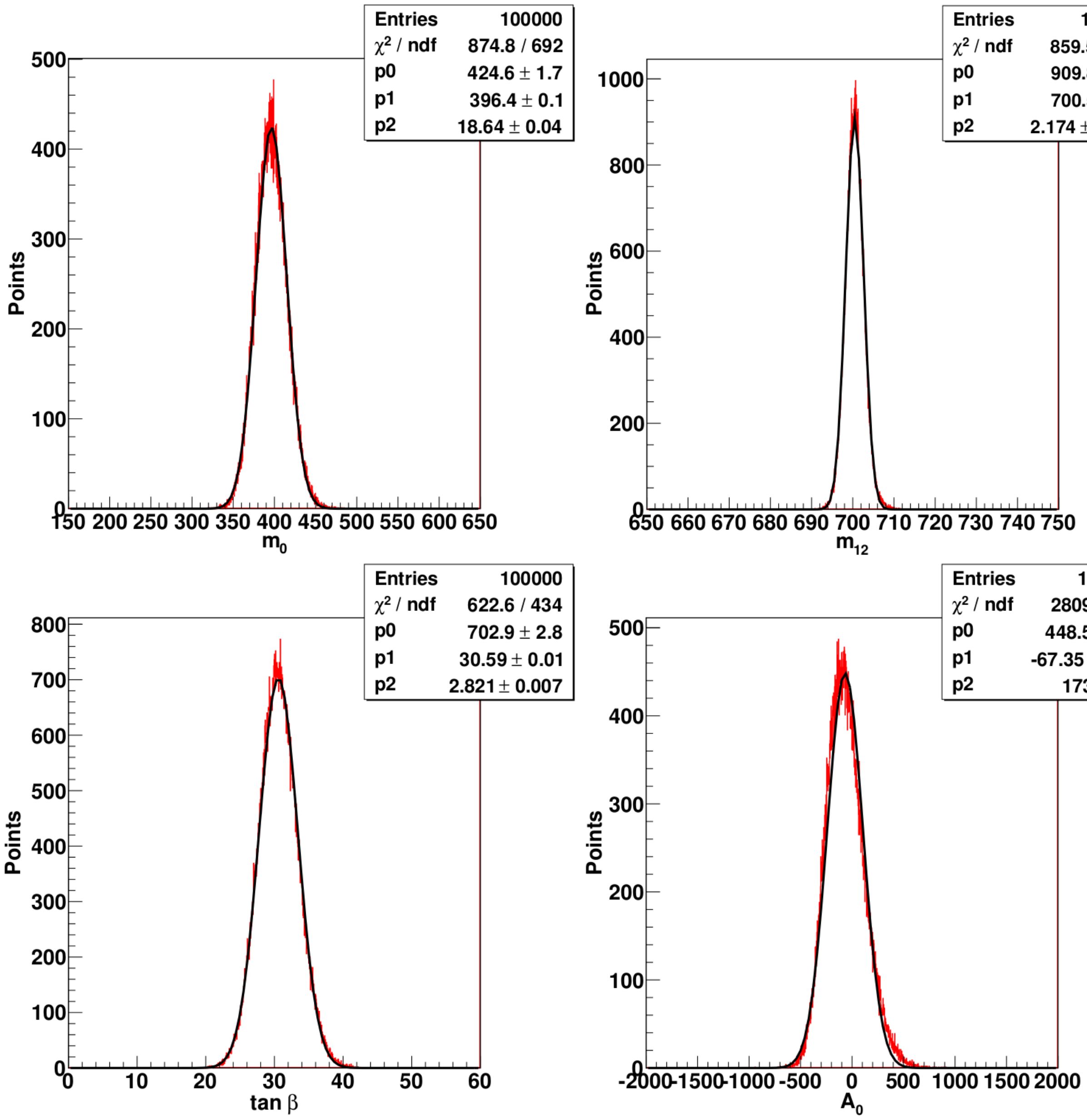

Figure 7.2.: The same distributions like the ones in Fig. 7.1, but originating from a 100 fb^{-1} instead of a 10 fb^{-1} measurement.

propagation of uncertainty method are shown in Table 7.2 for all reference points. Additionally, two figures for the method using Gaussian measurement copies for point 3 are included in Subsection 6.2.3. The values from the left side of Figure 6.6 are with $\overline{m}_0 \pm \sigma_{m_0} = (1034 \pm 42.3)\,\text{GeV}$, $\overline{m}_{1/2} \pm \sigma_{m_{1/2}} = (607.5 \pm 9.3)\,\text{GeV}$ and $\rho_{m_0\, m_{1/2}} = -0.291$ already consistent with the values calculated with the error propagation from Table 7.2 for 10 fb^{-1} of data with $\overline{m}_0 \pm \sigma_{m_0} = (1056 \pm 47.3)\,\text{GeV}$, $\overline{m}_{1/2} \pm \sigma_{m_{1/2}} = (607.5 \pm 11.5)\,\text{GeV}$ and $\rho_{m_0\, m_{1/2}} = -0.309$. In the following part, the one– and two–dimensional Gaussian distributions for all parameters of reference point 4 are presented. Exemplary, this should show the agreement of both methods.

In Figure 7.1, the one–dimensional distributions are shown which are created by feeding the ap-

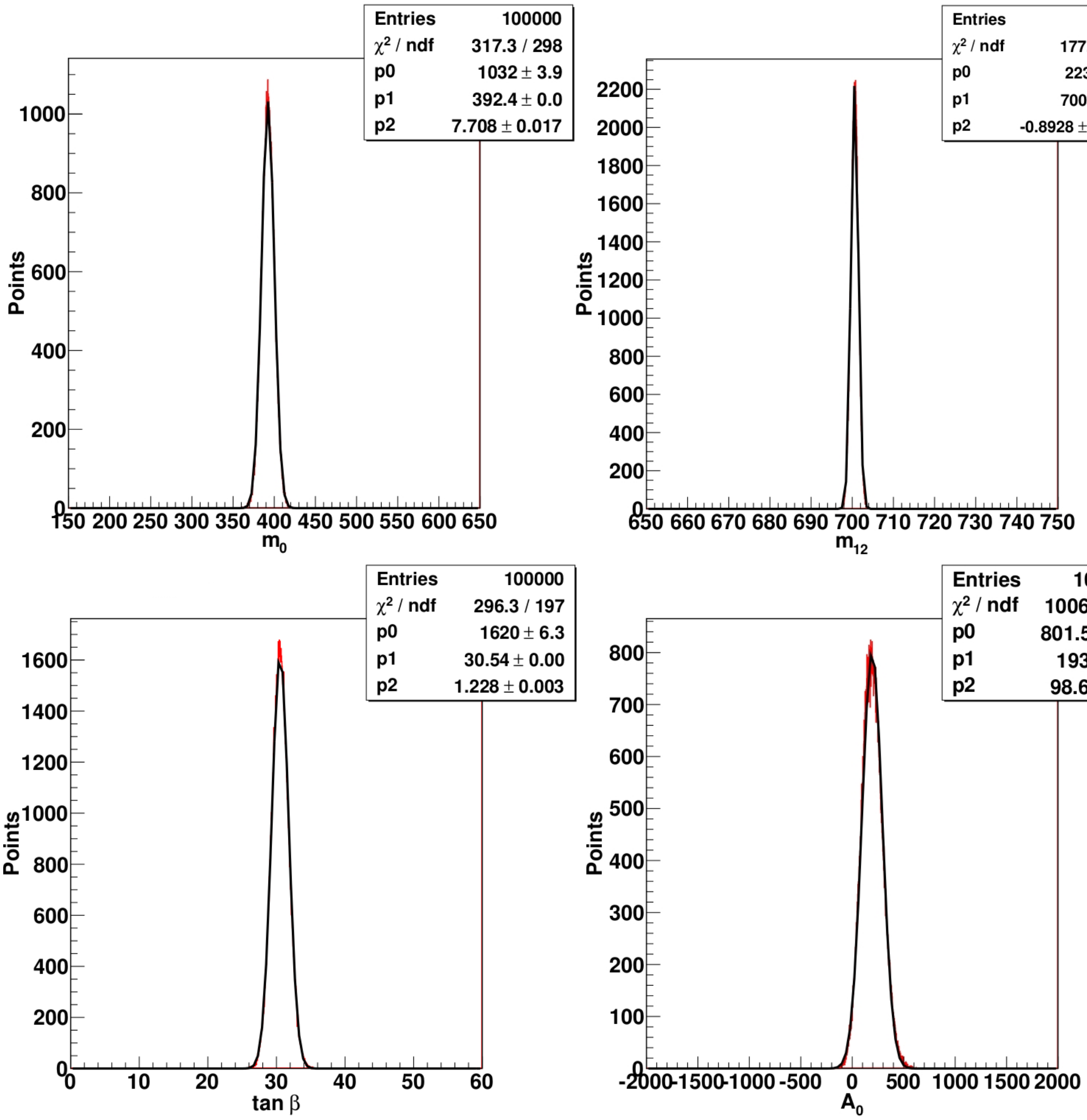

Figure 7.3.: The same distributions like the ones in Fig. 7.1, but originating from a 500 fb^{-1} instead of a 10 fb^{-1} measurement. Figure taken from [2].

propriate neural networks each with 100,000 Gaussian distributed copies of the 10 fb^{-1} measurement for point 4. Figures 7.2 and 7.3 show the same distributions as Figure 7.1, but for an integrated luminosity of 100 fb^{-1} and 500 fb^{-1}, respectively. First of all, all shown distributions look quite Gaussian. Furthermore, the three figures show nicely how the standard deviations (parameters p_2 of the Gaussian fits) become smaller with an increasing amount of data. Note that also here the standard deviations for an integrated luminosity of 500 fb^{-1} by trend should be somewhat underestimated, since the network uncertainty is not included in the Gaussian distributions. Comparing the errors with the ones computed in Table 7.2 shows a very good agreement (up to 3 % for A_0) for 500 fb^{-1}, but some differences for 10 fb^{-1} of data. This coincides with the expectation that the error on the uncertainty increases for a lower amount of data. Furthermore, additional differences can occur because of the fixed minimal numbers of events which are required for the consideration of an observable. All neural network input values for observables, which do not fulfill the required minimal numbers, are set to zero, as described in Subsection 6.2.2.

As an example, a particular lepton class within a measurement should contain 7 events for an integrated luminosity of $10\,\mathrm{fb}^{-1}$, as this is for example the case for class 9 of point 4 in Table 6.2 (using 991 events after cuts). The observables of this class would not be considered in the propagation of uncertainty method, since the class contains less than ten events. In contrast, the creation of Gaussian distributed measurement copies would surely lead to copies, which have ten or more events in this particular class (since the standard deviation is $\sqrt{7} \approx 2.65$). For such copies, the observables would be considered within the calculation of the neural network output and therefore in particular contribute to the determined standard deviation of the CMSSM parameter. As a consequence, there would be some deviation between both methods. Note that also a non–consideration of a class for a Gaussian measurement copy may happen if this class contains some more than ten events in the original measurement (e.g. class 4 of point 4 has 13 events), while the observables of this class would be considered in the error propagation. Such a deviation between the methods should decrease with increasing luminosity, since the (relative) errors on the number of class events become smaller. E.g. class 9 of point 4 has around 350 events after cuts for an integrated luminosity of $500\,\mathrm{fb}^{-1}$. This is more than eight standard deviations away from the required minimal number of 500 class events. Therefore, none of the up to 1,000,000 generated Gaussian copies should normally include equal or more than 500 events in this class.

Figure 7.4 shows the two–dimensional Gaussian distributions for point 4 for an integrated luminosity of $500\,\mathrm{fb}^{-1}$. The distributions for all six combinations of the four CMSSM parameters are created by feeding both appropriate neural networks with each 1,000,000 Gaussian distributed measurement copies. The determined correlation coefficients agree well with the ones listed in Table 7.2. The agreement of both methods within statistical uncertainties confirms the reliability of the calculated standard deviations and correlations of the determined parameters, however, without the consideration of the network uncertainty.

As noted earlier, around 1,000 training sets are each used for the learning procedure of the neural networks. A priori, it was not sure, if the number is sufficient to allow an optimal determination of the CMSSM parameters. However, the sizes of the determined standard deviations in Table 7.2 are especially for the measurements with an integrated luminosity of $10\,\mathrm{fb}^{-1}$ much bigger than the average distances between the values of the CMSSM parameters in the training sets. This indicates that the number of used training sets is indeed sufficient. Furthermore, for reference region 4, neural networks were created using around one and a half more training sets (1620 instead of 1053, i.e. with 567 extra ones). Although the control errors are around 4 to 8 % smaller, the estimated standard deviations for $500\,\mathrm{fb}^{-1}$ of data (which have a smaller statistical uncertainty than the $10\,\mathrm{fb}^{-1}$ ones) are almost identical and sometimes even slightly bigger. The difference for errors of m_0, $m_{1/2}$ and $\tan\beta$ are less than 0.3 %. Only σ_{A_0} is with 96.3 GeV for the higher number of training sets slightly smaller than 98.6 GeV from Figure 7.3, using Gaussian distributed measurement copies. However, the supposed improvement of less than 3 % should be consistent with statistical uncertainties, since already both error determination methods led to a difference of around 3 % for σ_{A_0} of reference point 4, see Table 7.2. Therefore, overall, the number of used training sets seems to be sufficient to allow an optimal determination of the CMSSM parameters for the considered measurements.

As explained in Subsection 6.2.2, from each original training set 100 Gaussian distributed copies (for the input values) were created to improve the performance of the created neural networks. Note that in contrast to the simulation of more training sets, which includes the computationally costly generation of events in Herwig++ for an integrated luminosity of $500\,\mathrm{fb}^{-1}$, the computational effort

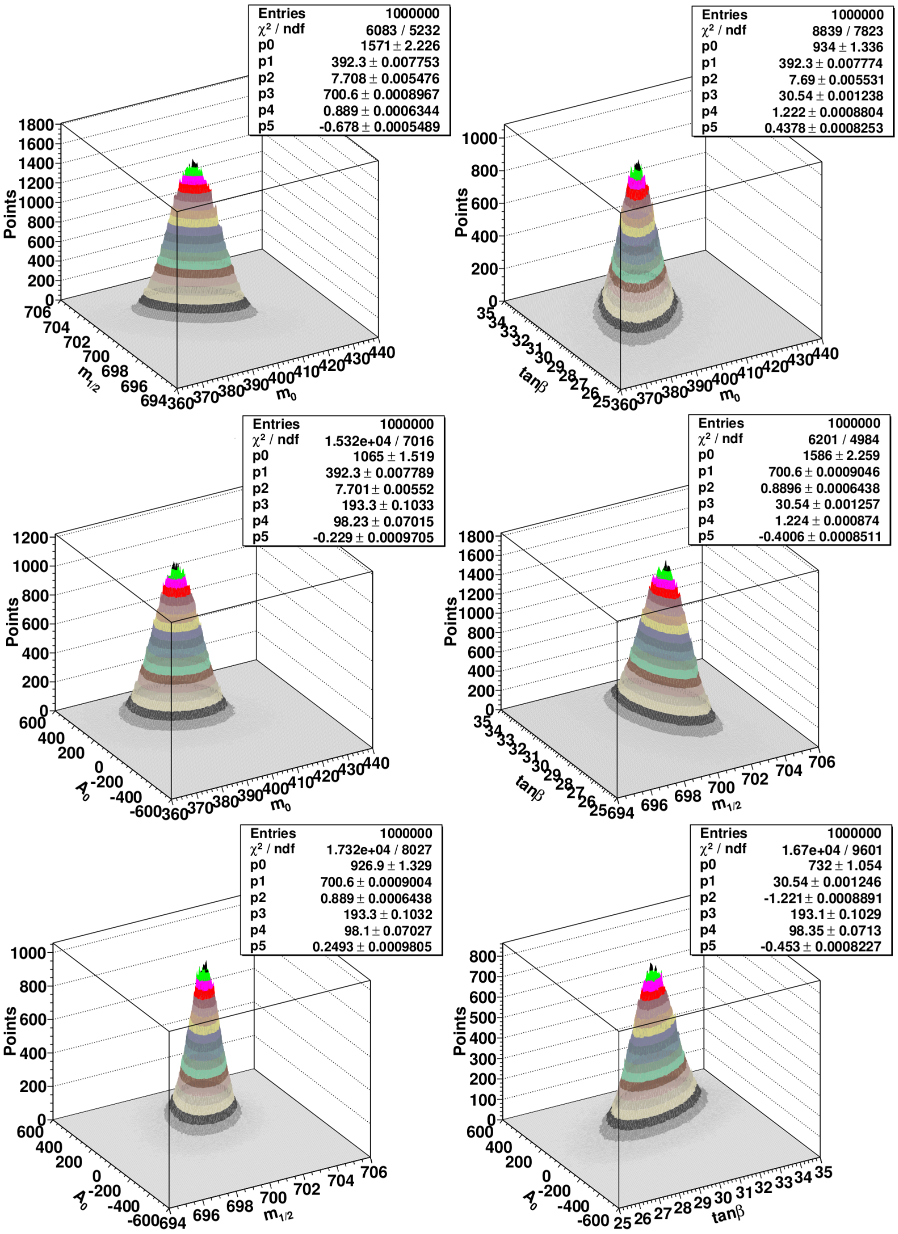

Figure 7.4.: Two–dimensional Gaussian distributions of all CMSSM parameter combinations for reference point 4. The unit for m_0, $m_{1/2}$ and A_0 is GeV. Both considered neural networks are each fed with 1,000,000 Gaussian distributed $500\,\mathrm{fb}^{-1}$ measurement copies. The fitted Gaussian distribution has each the form $g(x,\, y) = p_0 \cdot \exp[-0.5/(1-p_5^2)\cdot([(x-p_1)/p_2]^2 + [(y-p_3)/p_4]^2 - 2\cdot p_5/(p_2\, p_4)\cdot(x-p_1)\cdot(y-p_3))]$ with x and y being the appropriate CMSSM parameters, p_1 and p_2 the mean value and standard deviation of x, and p_3 and p_4 the mean value and standard deviation of y. The correlation factor p_5 is given by equation (6.11). The searched values are located in the middle of the axes, i.e. $m_0 = 400\,\mathrm{GeV}$, $m_{1/2} = 700\,\mathrm{GeV}$, $\tan\beta = 30$, and $A_0 = 0\,\mathrm{GeV}$. The neural network settings are listed in Table 7.1. The ellipse angles ϕ are distorted because of different axis lengths. Figure taken from [2].

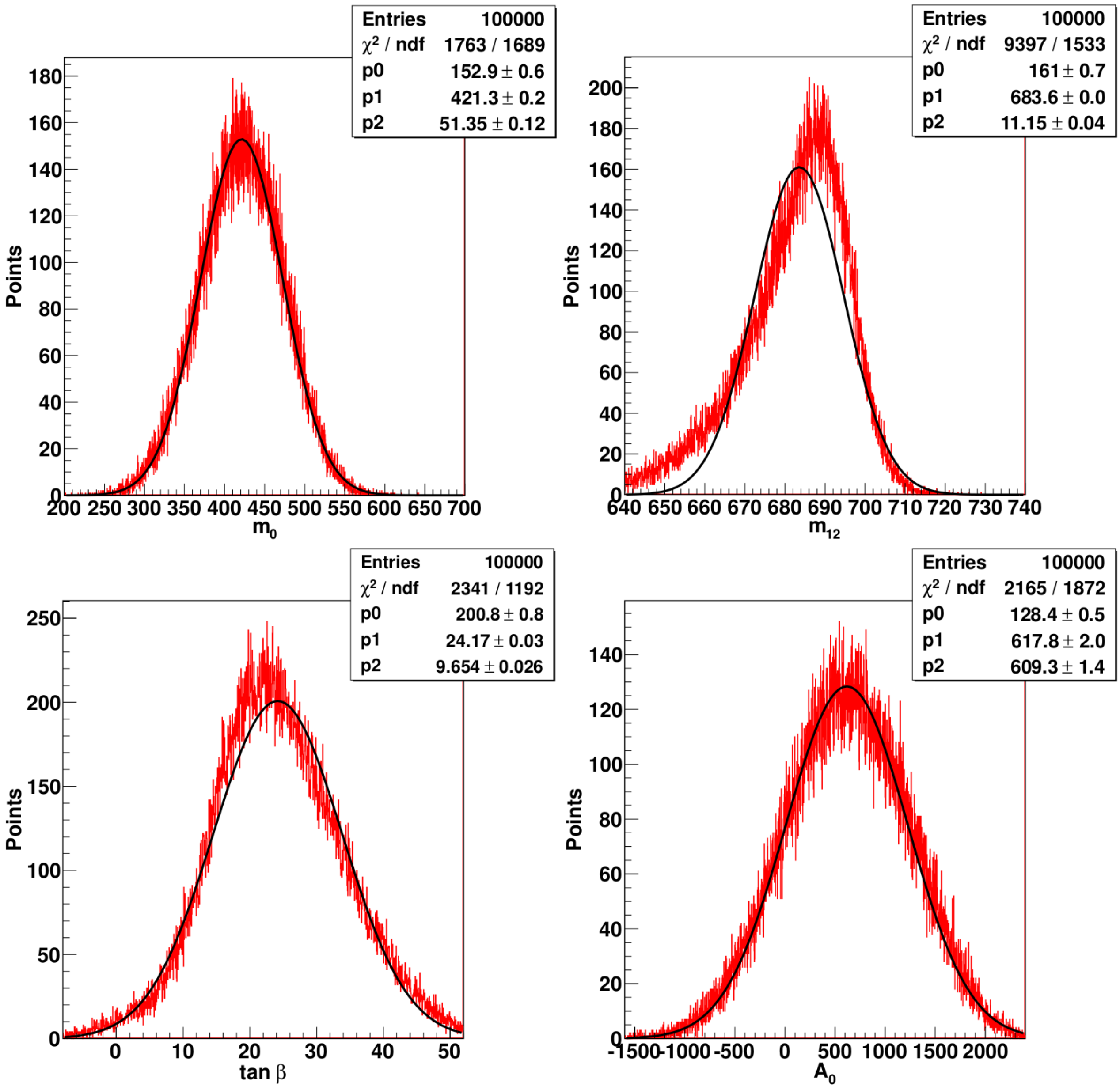

Figure 7.5.: One–dimensional Gaussian distributions for a different point in reference region 4, achieved by feeding the neural networks each with 100,000 Gaussian distributed $10\,\text{fb}^{-1}$ measurement copies. The unit for m_0, $m_{1/2}$ and A_0 is GeV. The fitted Gaussian distributions have each the form $g(x) = p_0 \cdot \exp(-1/2 \cdot [(x - p_1)/p_2]^2)$ with x being the appropriate CMSSM parameter, p_1 the mean value and p_2 the standard deviation. The searched values are located in the middle of the lower axes, i.e. $m_0 = 450\,\text{GeV}$, $m_{1/2} = 690\,\text{GeV}$, $\tan\beta = 22$, and $A_0 = 400\,\text{GeV}$. The neural network settings are given in Table 7.1.

for the generation of the training set copies is negligible. Of course, for the same number of learning steps, the learning time increases linearly with the number of used training sets. Comparing the results of neural networks for reference point 4, which are trained with only the 1,000 original training sets, to the presented results in Figures 7.1, 7.3 and Table 7.2 shows a significant improvement by using

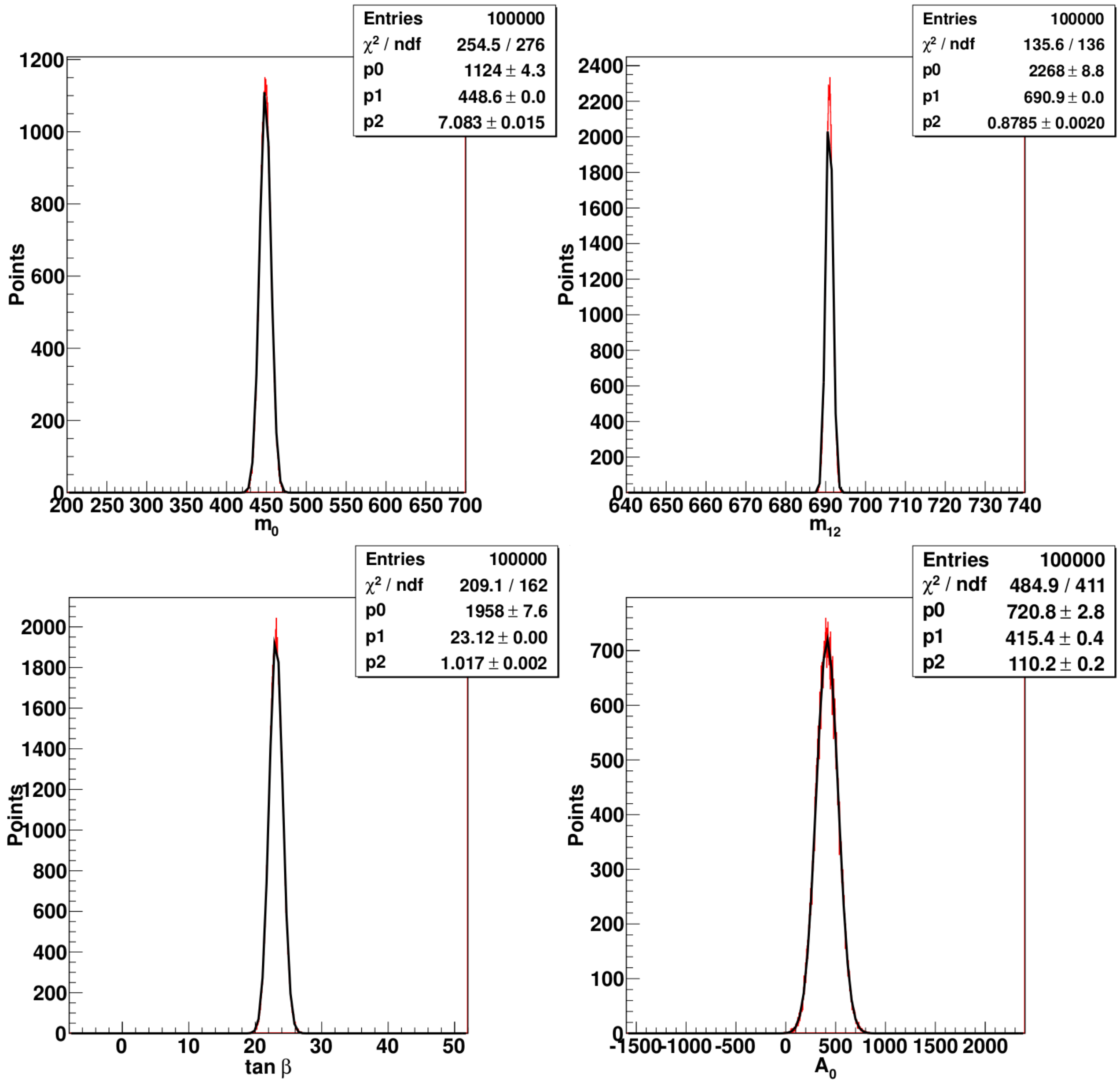

Figure 7.6.: The same distributions like the ones in Fig. 7.5, but originating from a 500 fb^{-1} instead of a 10 fb^{-1} measurement.

the 100 Gaussian copies for each training set. Note that the networks consist of the same number of hidden neurons. The estimated errors are for the networks without training set copies for all four CMSSM parameters higher (for 500 fb^{-1} of data 22 % for m_0, 35 % for $m_{1/2}$, 28 % for $\tan\beta$ and 12 % for A_0). Furthermore, the outputs for $\tan\beta$ and A_0, in particular for an integrated luminosity of 10 fb^{-1}, do not have such a nice Gaussian form. Therefore, the inclusion of the uncertainties for the input observables by creating Gaussian distributed training set copies leads to neurals networks with a significantly better performance, for the same amount of simulated data.

Once the neural networks are created for the investigated part of the parameter space, it should be possible to use them for the parameter determination of an arbitrary measurement originating from this part of the parameter space. In the following, this feature is demonstrated by determining the parameters of a different measurement within reference region 4. This measurement has the searched CMSSM parameters $m_0 = 450\,\text{GeV}$, $m_{1/2} = 690\,\text{GeV}$, $\tan\beta = 22$, and $A_0 = 400\,\text{GeV}$ with $1,057$ events after cuts for an integrated luminosity of $10\,\text{fb}^{-1}$ and a fixed seed in Herwig++. Figures 7.5 and 7.6 show the Gaussian distributions for an integrated luminosity of $10\,\text{fb}^{-1}$ and $500\,\text{fb}^{-1}$, respectively. They are created using the same neural networks like for reference point 4. The $m_{1/2}$–distribution in Figure 7.5 does not have a very nice Gaussian form. However, note that the right side of the distribution looks quite good. Recall that the values of $m_{1/2}$ for the training and control sets in reference region 4 are chosen from the range between 660 and 740 GeV. The left side of the shown distribution obviously leaves this range. As mentioned earlier, the neural network function should only be reliable within the investigated parameter space. The extrapolation to values outside the considered range generally should not give reliable results, as can be seen here. The $m_{1/2}$–distribution for $500\,\text{fb}^{-1}$ of data actually looks quite Gaussian, since all values of the distribution are here clearly within the considered parameter range. Otherwise, all determined parameter values of this point are within one or two estimated standard deviations from the true values. The size of the relative errors are comparable to the ones estimated for reference point 4.

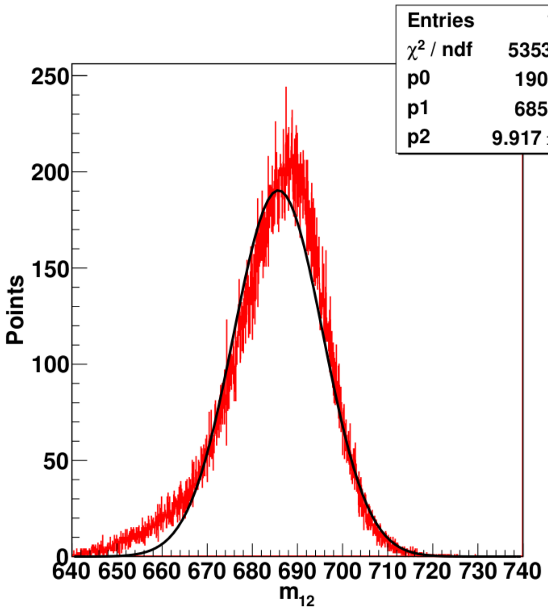

Figure 7.7.: One–dimensional distribution of $m_{1/2}$ for a different point in reference region 4 for an integrated luminosity of $10\,\text{fb}^{-1}$. The used neural network was trained with one and a half times more training sets than the one used in Figure 7.5. Everything else is the same as in Figure 7.5.

Furthermore, Figure 7.7 shows the $m_{1/2}$–distribution for the considered different point in region 4 for an integrated luminosity of $10\,\text{fb}^{-1}$, using the previous described neural network which was trained with one and a half times more training sets ($1,620$ instead of $1,053$). Compared to the $m_{1/2}$–distribution in Figure 7.5, the distribution has a nicer Gaussian form for values of $m_{1/2}$ around the lower bound of the investigated parameter range. This performance improvement for a higher number of used training sets is in contrast to the previously considered reference point 4, where the performance did not significantly change. However, the searched $m_{1/2}$–value for point 4 is located in the center of the investigated parameter range. Taking into account that the neural network function can only be reliably interpolated and not extrapolated, it is not surprising that parameter values in the center can already be determined optimally for a lower number of used training sets, compared to values closer to the edge of the parameter range. Therefore, increasing the number of training sets here still leads to a performance improvement, as could be seen in Figure 7.7 and was already indicated by a 6 % lower control error for the $m_{1/2}$ neural network with more training sets. Note that also the found parameters for m_0, $\tan\beta$ and A_0 have smaller estimated uncertainties by around 3 to 5 % for the networks with more training sets, comparing the results for 500 fb^{-1} of data for this particular point. Overall, Figure 7.5 shows the consequence of considering only a relatively small range for the gaugino mass parameter $m_{1/2}$, which was done because of the limited, existing computing resources. However, for a high enough number of training sets, it should be possible to reliably determine all parameter values within the investigated parameter range.

In the following section, the results for the determination of the CMSSM parameters for reference point 4 using neural networks are compared with the results which are achieved by using a χ^2–minimization.

7.2. Comparison with a χ^2–Minimization

In this section, the results of a parameter determination using a relatively simple χ^2–minimization are presented. As described before, the creation of four neural networks for a considered parameter region is a considerable amount of effort. Of course, once the networks are created arbitrary measurements can be directly assigned to their corresponding CMSSM parameters within the considered parameter region. In the following, the parameter determination is done using a χ^2–minimization instead of neural networks. Both methods are based on the same observables. A priori, a χ^2–minimization seems to be less work than the creation of four neural networks. If both methods would lead to comparable results, the method connected to less effort would obviously be preferable. But at least in the case of the CMSSM parameter determination considered in this thesis, it turns out that the present statistical fluctuations of the observables are a heavy burden for the described χ^2–minimization.

In the following, the difference between the measurement M and a prediction P is calculated with

$$\chi^2_{MP} = \sum_{m,n} (O^M_m - O^P_m) V^{-1}_{mn} (O^M_n - O^P_n) \tag{7.1}$$

which is the same formula as equation (5.1) used for the discrimination of two data sets. The m–th observable for the measurement is labeled O^M_m and for the prediction O^P_m. The inverse covariance matrix V^{-1} is determined with the covariance matrix entries

$$V_{mn} = \mathrm{cov}(O^M_m, O^M_n) + \mathrm{cov}(O^P_m, O^P_n) \tag{7.2}$$

which is the same equation as formula (5.2). The observables and the corresponding covariance matrix (version “b”) are listed in Chapter 4. The double sum in equation (7.1) runs over all observables which fulfill the following required (class) event numbers. For a measurement (and a prediction) based on an integrated luminosity of 10 fb^{-1} the total number of events after cuts is considered, if either the measurement or the prediction has at least one event after cuts. The event fraction n_c/N for a particular class c is taken into account, if the class c contains at least ten events either for the measurement or the prediction. All other observables have to be based on at least ten class events, both for the measurement and for the prediction. These are the same minimal numbers as the ones used for the neural networks. The minimal numbers are partly different to the ones listed in Table 5.1, which were used for the discrimination of the degenerate pairs in the first part of this thesis.

For the same integrated luminosity of $10\,\mathrm{fb}^{-1}$, the average number of events after cuts is with around 25,000 for the parameter sets from the degenerate pairs much higher than for the investigated CMSSM reference points with around 1,000. Against this background, the chosen minimal numbers showed better results than the previous higher ones with 50 for the observables $n_{c,b}/n_c$ and $\langle j \rangle_c$ and 500 for $n_{c,\tau^+}/n_c$ and $n_{c,\tau^-}/n_c$.

The χ^2–minimization is done in three steps. In the first two steps the measurement is compared to different predictions, and the corresponding χ^2_{MP} are calculated with equation (7.1). The measurement as well as all simulated predictions are based on an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The more similar the measurement and a prediction are, the smaller is the value of χ^2_{MP}. The searched CMSSM parameters are found by minimizing χ^2_{MP}. In the third and last step the errors of the found CMSSM parameters are estimated.

The first step of the χ^2–minimization should give a rough estimation of the CMSSM parameters belonging to the investigated measurement. This is done by running a simple simulated annealing algorithm. In each step of the algorithm a prediction is simulated for certain CMSSM parameters (picked randomly from stepwise fixed ranges) and compared to the measurement. The resulting value of χ^2_{MP} is then compared with the so far minimal value from the previous steps. If the new value for χ^2_{MP} is smaller than or equal the existing minimum, it is defined as the new minimum and the CMSSM parameters from the current prediction are saved as the best ones reproducing the measurement. On the other hand, if the new value for χ^2_{MP} is greater than the existing minimum, there is still a certain probability that the worse value is defined as the new minimum. This probability decreases exponentially with the difference of the new and old value of χ^2_{MP}. This should prevent the algorithm from getting stuck in a local minimum. The value of a particular CMSSM parameter is changed as long as a new minimum is found or a maximum number of tries is reached. During this changes all other parameters stay constant. In the following, the next CMSSM parameter is changed and this procedure is repeated as long as a maximum number of steps (here 250) is reached. In total, every CMSSM parameter is scanned multiple times to improve its determination.

There was not much effort put into choosing the values of the available parameters in the simulated annealing algorithm. The goal of the first step is just getting a rough estimation of the location of the searched CMSSM parameters. This location is used as a starting point for the second step of the χ^2–minimization. In this step, χ^2_{MP} is minimized using the algorithms "Simplex" and "Migrad" of TMinuit [17] in the data analysis framework ROOT [18]. The allowed ranges of the CMSSM parameters in the algorithms have to be specified to prevent the simulation of problematic parameter sets. In particular if $|A_0|$ is much bigger than the chosen m_0, this can lead to problems in the spectrum generation. Therefore, the main reason for doing a rough estimation of the CMSSM parameters in the first step is to be able to make a sensible choice for the allowed parameter ranges. Additionally, the algorithm has a better performance if the starting point is near the minimum and the distance to the minimum can very roughly be estimated (using the p–value of the starting point).

The CMSSM parameters found at the end of the second step have almost vanishing estimated errors, especially too small to be correct. This wrong error estimation should be caused by the statistical uncertainty of the simulated predictions, being present for an integrated luminosity of $10\,\mathrm{fb}^{-1}$. The minimum of χ^2_{MP} found at the end of the second step should be located at an extreme statistical fluctuation too lower values of χ^2_{MP} in a region of low values for χ^2_{MP}. The variation of the CMSSM parameters becomes very small at the end of the algorithm. From the location of an extremely negative fluctuation, a small parameter variation would then lead to a steeply increasing χ^2_{MP}. As a consequence, the algorithm would conclude a very narrow quadratic form around the minimum of χ^2_{MP}. Since the broadness of the curve corresponds to the errors of the found CMSSM parameters, this translates into very small errors.

One way of fixing this problem should be the reduction of the statistical uncertainties of the prediction by increasing the luminosity. The increase of the luminosity comes along with the increase of the simulation time, since the algorithm and in particular the generation of events with Herwig++ runs here only on one computer processor and the predictions are simulated one after the other. In particular, the statistical uncertainty decreases only inversely to the square–root of the number of generated events, while the simulation time increases linearly. Therefore, the simulation of hundreds of predictions with a high (enough) integrated luminosity leads to an impractical simulation time. Although the integrated luminosity was increased to $500\,\mathrm{fb}^{-1}$ for a test case, this did not turn out to be enough to solve the problem of the extreme statistical fluctuations.

In the third step of the described χ^2–minimization the errors of the CMSSM parameters associated with the considered measurement should be determined. To do so, 500 predictions randomly distributed around the second step minimum are simulated (on multiple processors) each for an integrated luminosity of $500\,\mathrm{fb}^{-1}$. These predictions are compared with the measurement and the corresponding values of χ^2_{MP} are calculated. Note that for the comparison of a $10\,\mathrm{fb}^{-1}$ measurement and a $500\,\mathrm{fb}^{-1}$ prediction, the total number of events after cuts and the corresponding variance for the prediction is divided by 50. The distribution of 500 values of χ^2_{MP} is in the following fitted to the four–dimensional quadratic function $\chi^2(m_0,\, m_{1/2},\, \tan\beta,\, A_0)$ with

$$\begin{aligned}\chi^2 &= \chi^2_{min} + \Delta\chi^2 \\ &= \chi^2_{min} + \begin{pmatrix} m_0 - m_0^{min} \\ m_{1/2} - m_{1/2}^{min} \\ \tan\beta - \tan\beta^{min} \\ A_0 - A_0^{min} \end{pmatrix}^T \cdot V^{-1} \cdot \begin{pmatrix} m_0 - m_0^{min} \\ m_{1/2} - m_{1/2}^{min} \\ \tan\beta - \tan\beta^{min} \\ A_0 - A_0^{min} \end{pmatrix} . \end{aligned} \tag{7.3}$$

In this function, $\chi^2_{min} = \chi^2(m_0^{min},\, m_{1/2}^{min},\, \tan\beta^{min},\, A_0^{min})$ is the minimal value of χ^2, and m_0^{min}, $m_{1/2}^{min}$, $\tan\beta^{min}$ and A_0^{min} are then the values of the CMSSM parameters, which would be associated to the searched values of the measurement. These determined values for the measurement are in general different to the values which are found at the end of the second step of the described χ^2–minimization. The matrix V^{-1} is the inverse of the symmetric covariance matrix which includes the searched four variances and six covariances of the CMSSM measurement parameters. So in total, 15 free parameters are included in equation (7.3). The advantage of using so many predictions at the same time for the determination of the errors is that ideally the statistical fluctuations cancel out.

The fit of $\chi^2(m_0,\, m_{1/2},\, \tan\beta,\, A_0)$ is done using the algorithms “Simplex” and “Migrad” in TMinuit again. In practice, the summed quadratic differences

$$\sum_{i=1}^{500} \frac{(\chi^2_{MP,\,i} - \chi^2(m_0^i,\, m_{1/2}^i,\, \tan\beta^i,\, A_0^i))^2}{(\chi^2_{MP,\,i})^d} \tag{7.4}$$

are minimized. The i–th of the 500 predictions has the parameter values m_0^i, $m_{1/2}^i$, $\tan\beta^i$ and A_0^i, the real value $\chi^2_{MP,\,i}$ from the comparison with the measurement and this real value is compared to the with current fit parameters computed value $\chi^2(m_0^i,\, m_{1/2}^i,\, \tan\beta^i,\, A_0^i)$. The fit parameters which should be determined are χ^2_{min}, m_0^{min}, $m_{1/2}^{min}$, $\tan\beta^{min}$, A_0^{min} and the entries of V^{-1}. The value of the parameter d determines the influence of the lower and the higher values of the $\chi^2_{MP,\,i}$ on the fit. A high positive value of d reduces the influence of higher $\chi^2_{MP,\,i}$ on the fit, i.e. points which are in general farther away from the minimum are suppressed. Such a reduction could make sense if the χ^2–distribution does not follow a quadratic form far away from the mininum. This could be caused by a dramatic change of the supersymmetric spectrum for CMSSM parameters far away from the minimum (because e.g. decay branches open or close).

Doing the fit, it turns out that the choice of d in equation (7.4) has a relatively big influence on the results. With the present information it is impossible to decide which fit is the best one. A solution for this dilemma is the consideration of (correlated) Gaussian distributed measurement copies similar to the error determination for the neural network described in Subsection 6.2.3. In the following, 1,000 Gaussian distributed copies of the $10\,\mathrm{fb}^{-1}$ measurement are created. These 1,000 copies are each compared with 500 simulated predictions. The resulting 1,000 χ^2–distributions with each 500 points are fitted as described above with equations (7.3) and (7.4).[†] Each of these fits gives values for the CMSSM measurement parameters m_0^{min}, $m_{1/2}^{min}$, $\tan\beta^{min}$ and A_0^{min}. Since the measurements are Gaussian distributed, also the fitted CMSSM measurement parameters should be Gaussian distributed. As a consequence, the forms of the distributions for the four CMSSM measurement parameters prove the quality of the fits and in particular the choice of the parameter d in equation (7.4). The square–roots of the variances of the Gaussian distributions give the searched CMSSM measurement parameter errors.

Finally, the results for the χ^2–minimization are presented, exemplary for reference point 4. Figure 7.8 shows the fitted Gaussian distributions of the four CMSSM measurement parameters originating from 1,000 Gaussian distributed measurement copies. The fits done to determine the CMSSM measurement parameters used $d = 3$ in equation (7.4). The shown distributions do not clearly follow Gaussian distributions. In particular the tails decline too slowly. This indicates that the parameter errors determined with these distributions are not completely reliable. The reliability of the results, i.e. the form of the Gaussian distributions, can probably be improved by using more than 500 χ^2_{MP}–points for each measurement fit. Figure 7.9 shows for example the distribution for the parameter m_0 using only 300 χ^2_{MP}–points for each fit. The results in Figures 7.8 and 7.9 are somewhat different with $m_0 = (404.5 \pm 134.5)\,\mathrm{GeV}$ and $m_0 = (398.5 \pm 152.8)\,\mathrm{GeV}$. Compared to the fit with 500 χ^2_{MP}–points, the one using 300 χ^2_{MP}–points has a somewhat worse behavior at the tails. However, the difference between using 300 and 500 χ^2_{MP}–points is not very big. This indicates that a significant improvement probably can only be expected using many more than 500 χ^2_{MP}–points for each CMSSM parameter fit. This obviously would require a much higher computational effort which is not feasible with the existing computing resources in a reasonable amount of time.

The same procedure described above can be done for a measurement with higher luminosity. The resulting distributions of the CMSSM measurement parameters should lead to appropriately smaller errors for the parameters. Increasing the integrated luminosity to $500\,\mathrm{fb}^{-1}$ compared to $10\,\mathrm{fb}^{-1}$ should lead to Gaussian errors which are reduced by around a factor of $\sqrt{50} \approx 7$.[‡] Figure 7.10 shows the m_0–distribution created with Gaussian distributed measurement copies with an integrated luminosity of $500\,\mathrm{fb}^{-1}$ and using 500 χ^2_{MP}–points as well as $d = 3$ in equation (7.4) as before. The resulting standard deviation can be compared to the one for a $10\,\mathrm{fb}^{-1}$ measurement in Figure 7.8. The reduction from $\sigma^{10}_{m_0} = 134.5$ to $\sigma^{500}_{m_0} = 83.5$ is with a factor of approximately 1.6 much smaller than the assumed factor of around 7. Additionally to the previously seen non–perfect Gaussian form of the distribution, this questions the reliability of the found parameter errors.

Furthermore, Figure 7.11 shows the m_0–distribution for a $500\,\mathrm{fb}^{-1}$ measurement, setting $d = 2$ or $d = 4$ in equation (7.4) of the fitting procedure instead of $d = 3$ as done before in Figure 7.10. As mentioned before, the fits obviously depend relatively strongly on the choice of d, as can be seen in the shown distributions. In particular, the estimated errors differ by a factor of two four

[†] Note that “only” 1,000 measurement copies (instead of 100,000 copies like for the neural network output distributions) are used, since each resulting 500 point χ^2–distribution has to be fitted using “Simplex” and “Migrad”. Such a fit takes a non–negligible amount of time.

[‡] The minimal (class) event numbers required for the comparison of the measurement with a prediction, both for an integrated luminosity of $500\,\mathrm{fb}^{-1}$, are increased by a factor of 50 (compared to $10\,\mathrm{fb}^{-1}$) to 50 for the total number of events after cuts and 500 for all other observables.

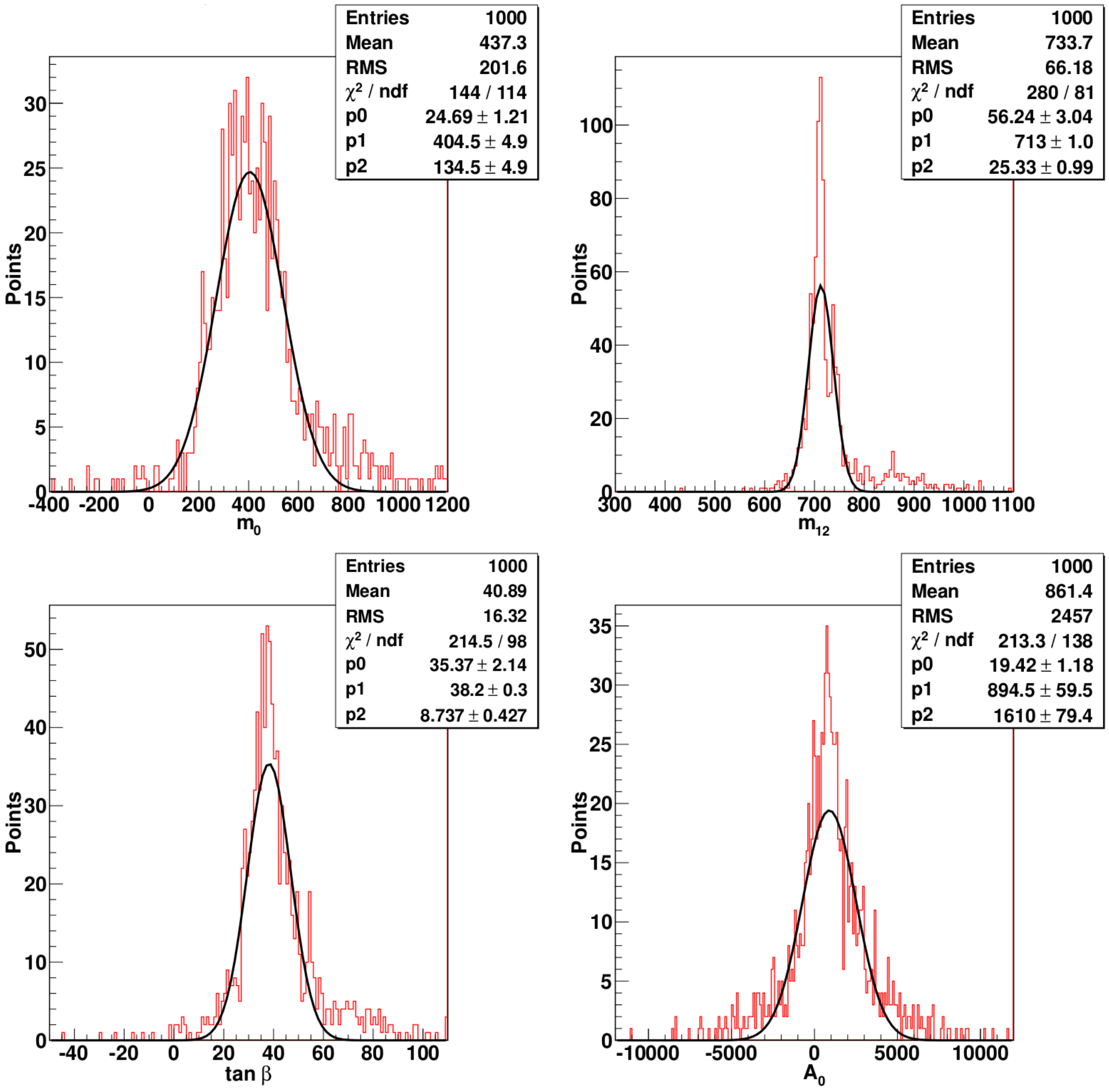

Figure 7.8.: Fitted Gaussian distributions for reference point 4 with an integrated luminosity of 10 fb^{-1} determined by the χ^2–minimization. The unit for m_0, $m_{1/2}$ and A_0 is GeV. 1,000 Gaussian distributed measurement copies are each compared with 500 predictions with an integrated luminosity of 500 fb^{-1}. The 500 values of χ^2_{MP} are each fitted to determine the CMSSM measurement parameters, with $d = 3$ in equation (7.4). The determined CMSSM measurement parameters are fitted in the four histograms. "Mean" and "RMS" are the usual mean value and standard deviation of the entries in the histograms. The other parameters are included in the Gaussian fit $g(x) = p_0 \cdot \exp(-1/2 \cdot [(x - p_1)/p_2]^2)$ with x being the appropriate CMSSM parameter. The searched values are located in the middle of the lower axes, i.e. $m_0 = 400$ GeV, $m_{1/2} = 700$ GeV, $\tan\beta = 30$, and $A_0 = 0$ GeV. Figure taken from [2].

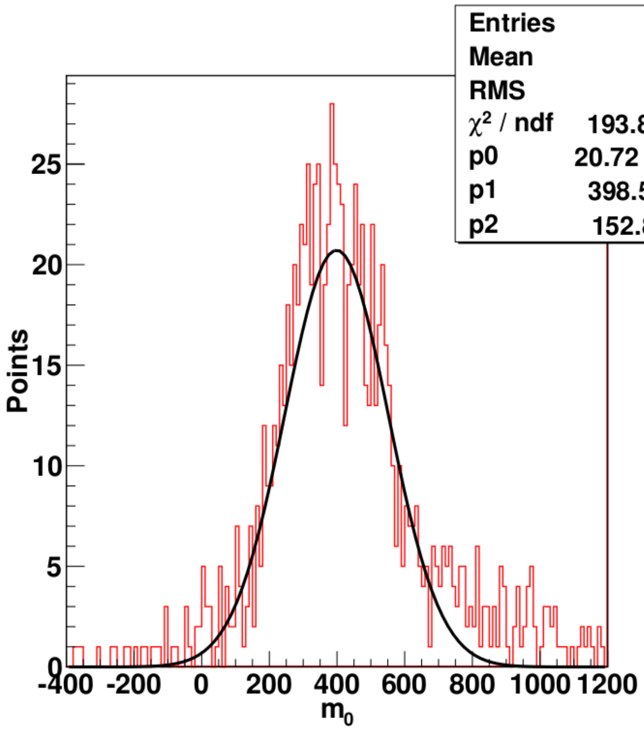

Figure 7.9.: The same m_0–distribution as the one in Figure 7.8, but using only 300 instead of 500 χ^2–points for each CMSSM measurement parameter fit.

both distributions. However, note that the determined values of the CMSSM parameters are for all shown distributions in Figures 7.8 to 7.11 less than one estimated standard deviation away from the true values. Furthermore, each of the 1,000 measurement copy fits gives next to the central values m_0^{min}, $m_{1/2}^{min}$, $\tan\beta^{min}$ and A_0^{min} also values for the corresponding standard deviations in equation (7.3). These values for the standard deviation have around the same order of magnitude as the errors estimated with the Gaussian fits in Figures 7.10 and 7.11. However, the values for the 1,000 fits fluctuate quite a lot, and in particular much more than the central values.

Overall, the current results for the χ^2–minimization are not satisfying. Although the determination of the searched CMSSM parameters works in principle, the computed errors seem to be rather a rough estimation for the true errors. This probably can be improved by further increasing the integrated luminosity of the simulated predictions and especially simulating a higher number of them for the third step. But in contrast to the first expectation, this would further increase the amount of effort connected to the χ^2–minimization. Comparing the results and the spent effort to the neural network method, at this moment it seems to be more reasonable to put no further work into the described χ^2–minimization. The parameter determination using neural networks seems to give better (for 10 fb^{-1} of data for m_0 and $m_{1/2}$ around two to three times smaller errors, for 500 fb^{-1} an order of magnitude smaller) and in particular more reliable results for a comparable amount of effort. In particular, it seems that also putting much more computational effort into the χ^2–minimization would still lead to results which are worse than the results for the neural networks.

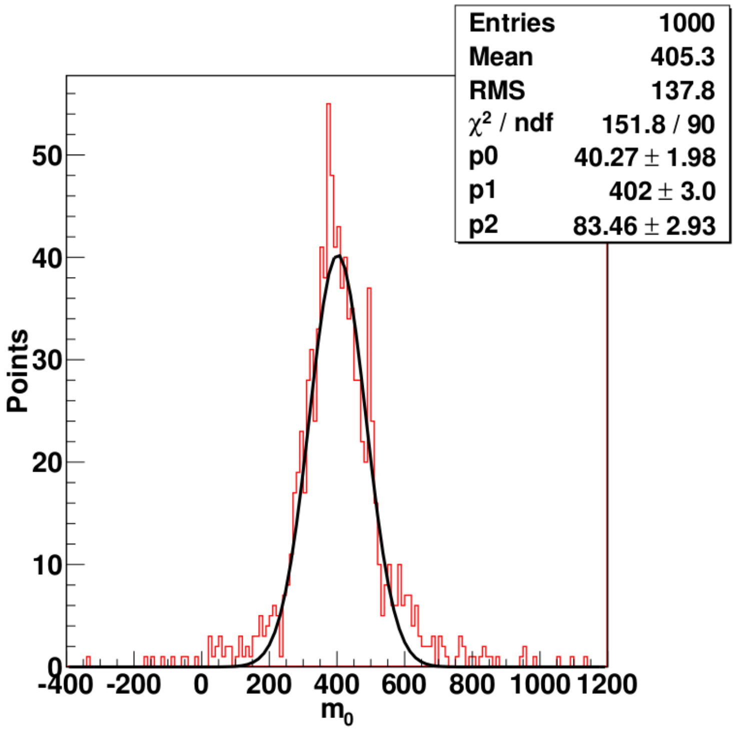

Figure 7.10.: The same m_0–distribution as the one in Figure 7.8, but using $500\,\text{fb}^{-1}$ instead of $10\,\text{fb}^{-1}$ measurement copies. Figure taken from [2].

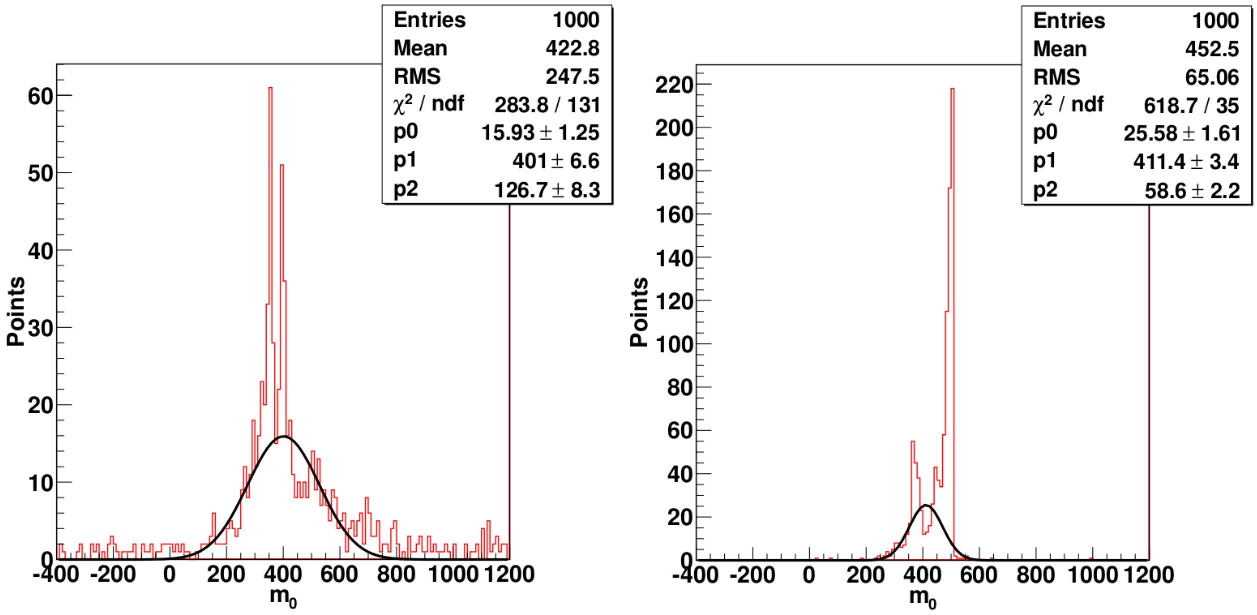

Figure 7.11.: The same distribution as in Figure 7.10, but using the factor $d = 2$ (left) and $d = 4$ (right) instead of $d = 3$ in equation (7.4) of the fitting procedure. Figure taken from [2].

8. Summary and Outlook

In this thesis the supersymmetric parameter determination at the LHC with the help of neural networks was studied. For this purpose, 84 mostly counting observables were defined in Chapter 4 to record the information about measured events at the LHC. For the remaining part of this study, it was essential that the variances and covariances of the observables are known. In the first part of this thesis, the usefulness of the observables for supersymmetric parameter determination was checked. To do so, 283 pairs of parameter sets for a MSSM with 15 parameters were investigated. In reference [5], these pairs were claimed to be indistinguishable for an experiment at the LHC with a center of mass energy of 14 TeV and $10\,\mathrm{fb}^{-1}$ of accumulated data. The data sets of the supposedly non–discriminable pairs have on average around 25,000 events after cuts, i.e. a relatively large number. Therefore, such an indistinguishability could generally doubt the capability for the determination of many supersymmetric parameters at the LHC.

In contrast to [5], the in Chapter 5 described comparison of two different data sets by calculating a single χ^2–variable takes the correlations of the observables into account. Because of the enormous number of 1,808 mostly kinematical observables used in their data set comparison, this was not done in [5]. The consideration of correlations ensures the statistical validity of the used comparison method, as was explicitly checked in Section 5.1. Furthermore, initial state radiation as well as the underlying event were included in the simulation of the events.

It turned out that 260 of the 283 pairs could be discriminated with 95 % confidence level using the 84 observables within the described comparison method, assuming the same systematic errors as reference [5]. Without systematic errors even all pairs could be distinguished. Furthermore, the influence of Standard Model background was explored. The consideration of SM backgrounds and systematic errors led to 46 pairs which could not be distinguished with 95 % probability, whereas without systematic uncertainties only one pair could not be discriminated. Systematic errors therefore seem to have a stronger influence on the parameter discrimination than SM background. Note here that the supersymmetric cross–sections of the parameter sets within the 283 pairs are relatively high. A proper handling of systematic errors is therefore necessary, whereas for the experimental uncertainties this only can be done accurately by the corresponding experimental groups. To reach the relatively small systematic errors, which were used, the inclusion of quantum corrections for all relevant cross–sections as well as branching ratios is needed.

Overall, this result mitigates the LHC inverse problem, meaning that a measurement could not be uniquely assigned to certain parameter values of a supersymmetric theory, even if a relatively large number of events is measured. Using the right observables, generally for most supersymmetric parameters it seems to be possible to discriminate data sets resulting from different parameter choices. On the other hand, this result indicates the usefulness of the used observables for the direct determination of the supersymmetric parameter values from a measurement. Therefore, in the second part of this thesis the observables were used as input values for neural networks.

Neural networks are trained to find the mapping between the measured observables and the parameter values of the considered supersymmetric theory. In this thesis the considered supersymmetric theory is the CMSSM, which only has 4 continuous parameters and a sign choice. Note that the sign of μ was fixed to positive values. The low number of free parameters reduces the amount of necessary computations and therefore respects the limited, available computing resources. However, the described neural network method can just as well and easily be used for a supersymmetric theory with more parameters, only with the consequence that the computational effort is higher. In particular, the method can also be used for any other non–supersymmetric theory as long as appropriate observables are used.

The goal of the neural network learning process is the ability of the network to determine the unknown parameter values and their corresponding errors for an existing measurement. The key for a successful learning process is the consideration of the statistical uncertainties of the observables. First of all, statistical uncertainties were reduced by simulating with $500\,\mathrm{fb}^{-1}$ a much higher integrated luminosity for the training and control sets than for the considered $10\,\mathrm{fb}^{-1}$ measurements. Additionally, only observables were considered, which were based on a minimal number of events. For each training set, (correlated) Gaussian distributed copies were created with the consequence that the neural network learning process was confronted with the relative observable uncertainties and could reduce the weights for less exact observables. This measure led to an improvement of around 24 % for the considered estimated standard deviations. Furthermore, for each CMSSM parameter a single network was created, which should allow an easier specialization on the parameter, meaning that the determination is improved. The creation of a single network also needs less computation time than the creation of a bigger network for all four parameters.

In contrast, the in Section 7.2 processed χ^2–minimization showed the challenges arising from the statistical fluctuations. At least with the used χ^2–minimization method, it would take large computational efforts to get reliable results, since many parameter points with a high integrated luminosity have to be generated. Additionally, the estimated standard deviations for m_0 and $m_{1/2}$ were two to three times bigger than the ones determined with the neural networks for $10\,\mathrm{fb}^{-1}$ of data. From this perspective the amount of effort put into the creation of neural networks seems to be worth it. In particular, the uncertainties of the generated training and control sets for the neural networks should be (much) smaller than the error of the measurement to achieve reliable results. On the other hand, for any kind of χ^2–minimization, the uncertainty of the calculated χ^2_{MP} between the measurement and a prediction needs to be (much) smaller than the change of χ^2_{MP}, which is induced by a change of the CMSSM parameters of the prediction. Since the changes of the CMSSM are at the end of a χ^2–minimization usually quite small, also the changes of χ^2_{MP} would be very small. Therefore, the integrated luminosity for the predictions has to be very high. The neural network approach should generally give better results for a similar (or smaller) computational effort.

Furthermore, an outstanding property of the neural network is the fact that it can be used for the whole considered parameter region to determine unknown parameters from a measurement, and not only for one specific measurement. If only one measurement is available, this is of course no advantage compared to something like a χ^2–minimization, although on the other hand, a neural network can be created even before a measurement is available. Additionally, the adaption of the input normalization and the required minimal numbers permits the determination of parameters for an arbitrary integrated luminosity (much) smaller than the one of the training and control sets.

For all four CMSSM reference regions, i.e. for qualitively different scenarios, the common scalar mass m_0 as well as the common gaugino mass $m_{1/2}$ can already be determined relatively well for an integrated luminosity of $10\,\mathrm{fb}^{-1}$ with around 1,000 events after cuts. The relative errors of m_0 go down to 4.5 % and for $m_{1/2}$ even to 1 %. On the other hand, the other two continuous CMSSM parameters $\tan\beta$ and A_0 could at best be determined very roughly. Increasing the integrated luminosity to $500\,\mathrm{fb}^{-1}$ improves the determination of m_0 for the four reference points to around 0.5 to 3 %. The parameter $m_{1/2}$ is always determined with a relative error well below 1 %. Also the parameters $\tan\beta$ and A_0 can be determined quite accurately. The absolute errors for $\tan\beta$ are for three points smaller or equal around 2. The errors for A_0 are around 25 to 35 % compared to the input value of m_0. Furthermore, most correlations between the CMSSM parameters are quite small.

The errors of the parameters could be determined with two different methods based on the knowledge of the variances and covariances of the measured observables, namely with the creation of Gaussian distributed measurement copies as well as with the propagation of uncertainties. The results of both methods are consistent. This reassures the reliability of the given results. Furthermore, the form of the output distribution resulting from the Gaussian distributed input distribution gives, next to the normalized control error, a handle for the performance of the neural network imitating the searched function. Note that the performance of the created neural networks can potentially and probably somewhat be improved by either changing the number of hidden neurons or increasing the number of training sets. However, at least for reference point 4 there was no significant change using one and a half times more training sets for the creation of the neural networks. Therefore, for the considered reference points the number of used training sets overall seems to be sufficient. For measurements which lie closer to the edges of the investigated parameter space, the use of more training sets may improve the parameter determination, as indicated by the considered different point in region 4. If for the creation of neural networks for a bigger part of the parameter space or a different physics model than the CMSSM more training sets or hidden neurons are needed, it could be sensible to make the learning procedure faster, i.e. reducing the amount of necessary computations. This could for example be done by approximating the Hessian matrix instead of explicitly calculating it, as it is done here and described in Appendix C. Furthermore, another set–up for the neural network, e.g. by introducing one (or more) additional layer(s) of hidden neurons, could allow the finding of the searched function between observables and parameters with less network weights, which then should lead to a faster learning process.

The results of the neural networks may be improved by using additional observables. E.g. so far only one kinematical observable per class is used. Additional information like kinks or edges of kinematical distributions may be included and deliver additional useful information for the determination of the searched parameters. However, the correlations of any new observable to the existing observables needs to be known. On the other hand, some of the used observables may not be needed since they do not deliver significant information for the determination of the CMSSM parameters. The connections from such observables should have low weights within the created neural networks, because their unimportance for the parameter determination should be recognized in the described learning procedure. However, the non–consideration of these observables should reduce the computational effort for creating the networks.

Furthermore, the knowledge of the mass and couplings of one of the CP–even neutral Higgs bosons should highly reduce the available parameter space. As mentioned earlier, the considered reference points do not respect the mass of the almost certainly found 125 GeV Higgs boson [16]. Considered supersymmetric scenarios should obviously agree with the experimental information. However, the purpose of this work was only to examine the performance of neural networks for parameter determination for different kinds of supersymmetric scenarios. The consideration of less constrained supersymmetric models than the CMSSM should also still allow similar scenarios in agreement with the experimental Higgs data.

The used selection cuts were not optimized for the different parameter regions, but instead should just give a generally satisfying reduction of the Standard Model background, as shown in Subsection 5.2.2. An optimization could probably further improve the results. However, taking the fact that the LHC is running and therefore potentially in the upcoming years some kind of supersymmetric signal is discovered, if at all, such an optimization seems to be more appropriate at that point of time. Furthermore, systematic errors as well as Standard Model backgrounds were not considered within the neural network parameter determination. As was seen in Subsection 5.2.2, the used observables were still useful for the discrimination of different parameter sets, even when SM background and systematic errors were taken into account, which in particular means that the observables could still show significant differences for different parameter choices. Depending on the size and the properties of the background events, i.e. how strongly the supersymmetric signal is “covered”, the estimated CMSSM parameter errors should increase appropriately after the inclusion of SM backgrounds and systematic errors. However, the basic determination of the parameters using neural networks should still work.

Overall, the described neural network approach delivers quite compelling and in particular statistically quite reliable results for the determination of supersymmetric parameters at the LHC. Even though it was only applied on a supersymmetric theory with four continuous parameters, it can easily be expanded to multiple parameters by just creating additional networks for each additional parameter. The method seems to be generally very useful for the determination of the parameters of an arbitrary (also non–supersymmetric) theory.

A. Definition of the Observables

This appendix is published in the same form in [1]. In this Appendix we describe in detail how our observables are defined. In particular, charged leptons and jets have to fulfill certain acceptance cuts to be counted. The τ–jets which arise from hadronically decaying τ–leptons are in general just called taus here. The determination of b– and non–b–jets is also explained in the following. The final sections deal with the calculation of the missing transverse momentum $\not{p}_T$ and the observable H_T.

Note that we count visible particles as measurable, only if they have pseudorapidity $|\eta| < 5$; neutrinos and the lightest neutralino are not considered visible, since they (usually) do not interact in the detector.

A.1. Electrons and Muons

We only consider *isolated* electrons and muons, i.e. a charged lepton l has to fulfill the following criteria:

- $|\eta^l| < 2.5$
- $p_T^l > 10\,\mathrm{GeV}$
- For all measurable particles with $\Delta R = \sqrt{(\eta^l - \eta)^2 + (\phi^l - \phi)^2} \leq 0.2$ the transverse energy E_T is summed. This sum has to be $\sum E_T < 5\,\mathrm{GeV}$

A.2. Taus

Only hadronically decaying taus, i.e. τ–jets, are considered here; the electrons and muons produced in leptonic τ–decays are counted as all other charged leptons, if they satisfy the criteria listed in the previous section. We use generator information to check whether a given τ–lepton indeed decays hadronically. The four–momentum p^τ of the τ–jet is then defined as the difference between the four–momentum of the parent τ–lepton and the four–momentum of the corresponding ν_τ. We impose the following acceptance cuts on the τ–jets:

- $|\eta^\tau| < 2.5$
- $p_T^\tau > 20\,\mathrm{GeV}$

Furthermore, the τ–jet should be isolated, i.e. there must not be any charged hadrons with $p_T > 1\,\mathrm{GeV}$ and no photons with $p_T > 1.5\,\mathrm{GeV}$ within a cone $\Delta R = \sqrt{(\eta^\tau - \eta)^2 + (\phi^\tau - \phi)^2} < 0.5$ around the tau; note that this condition is usually more stringent than that used for electrons and muons. Moreover, a τ–jet satisfying all conditions is only tagged with a $50\,\%$ probability. Recall also that some $35\,\%$ of all τ–leptons decay leptonically. Altogether we thus see that a sample of events containing equal numbers of electrons, muons and τ–leptons with identical kinematical distributions would indeed yield equal numbers of observed electrons and muons in our simulation, but far fewer identified τ–leptons. Note finally, that identified τ–jets are not included in the identification of other jets.

A.3. Jets

Jets are determined with the program FastJet [13] using the anti$-k_t$ algorithm. All measurable particles are taken into account, except for isolated electrons and muons that satisfy the criteria of Section A.1 as well as isolated τ–jets defined in Section A.2. We use the following parameter choices for the jet finding algorithm:

- double dou_R = 0.5
- fastjet::RecombinationScheme RecSch_scheme = fastjet::E_scheme
- fastjet::Strategy Stra_strategy = fastjet::Best

→ fastjet::JetDefinition JetDef_def(JetAlgo_algo, dou_R, RecSch_scheme, Stra_strategy)

Reconstructed jets are only counted if they satisfy:

- $p_T^j > 20\,\mathrm{GeV}$
- $|\eta^j| < 4.8$

In order to determine whether a given jet originates from a b–quark, we first check the progenitors of all particles in the jet (except for the photons). All particles that originate from the decay of one of the b–hadrons B^0, $\bar{B}^0$, B^+, B^-, B_s^0, $\bar{B}_s^0$, Λ_b^0 or $\Lambda_{\bar{b}}^0$ are marked. If a jet contains at least one such particle and the pseudorapidity fulfills $|\eta^{b-jet}| < 2.5$, then the jet is identified as a b–jet at the generator level. Note that the number of b–jets could differ from the number of decaying b–hadrons already at this stage. In particular, the products of multiple b–hadrons could end up in one jet, which is not unlikely if the two corresponding b–quark momenta are nearly parallel to each other. In principle the decay products of one b–hadron could end up in multiple jets, but this happens only rarely. Finally, a b–jet is only tagged with a 50 % probability. All jets which are not tagged as b–jets are counted as non–b–jets. Note that we ignore the possibility that jets which do not originate from a b–quark are tagged as b–jets (false positive tags).

A.4. Missing Transverse Momentum

In the simulation the transverse momentum vectors of all measurable particles are added. The missing transverse momentum $\vec{\not{p}}_T$ is the negative of this sum, $\vec{\not{p}}_T = -\sum_i \vec{\not{p}}_{i,T}$. Our event selection includes a cut on the absolute value of this quantity, $\not{p}_T = |\vec{\not{p}}_T|$.

A.5. H_T

H_T is defined as the sum of the transverse momenta of all hard objects and the absolute value of the missing p_T. Hard objects are all isolated leptons and jets that satisfy the criteria outlined above.

B. Selection Cuts

This appendix is published in the same form in [1]. As well known, the production of heavy superparticles can be detected at the LHC on top of SM backgrounds only after cuts have been applied to reduce these backgrounds. We apply three different sets of selection cuts, depending on the number and charge of identified leptons. Note that we include identified τ–jets as leptons for the purpose of defining these cuts. In the following, we outline cuts for events with at most one lepton, events with exactly two opposite–charged leptons, and events containing at least two leptons of the same charge (and possibly some additional lepton(s) of arbitrary charge). All leptons and jets have to satisfy the criteria described in Appendix A.

B.1. Events with at Most One Lepton

We impose the following cuts:

- $\not{p}_T > 200\,\mathrm{GeV}$
- $H_T > 1000\,\mathrm{GeV}$

Additionally there are either exactly two or at least four jets, on which we apply the following cuts:

B.1.1. Two Jets

This event sample will get contributions from squark pair production if at least one squark decays directly to the lightest neutralino (possibly plus very soft particles).

- $E_T^j > 300,\ 150\,\mathrm{GeV}$ for the hardest and second–hardest jet
- There are no additional jets with transverse energy $E_T^j > 30\,\mathrm{GeV}$ and no b–jets; this is intended to reduce backgrounds from top production
- $m^{jj} > 200\,\mathrm{GeV}$
- If there is exactly one lepton, then its transverse mass with the missing transverse momentum in the event should fulfill $m_T(\vec{p}^{\,l}, \vec{\not{p}}_T) > 80\,\mathrm{GeV}$; this suppresses backgrounds where the missing p_T comes from $W^\pm \to l^\pm \nu_l$ decays. This event sample will get contributions from squark pair production where exactly one squark decays into a chargino which in turn decays leptonically.

B.1.2. Four or More Jets

This event sample will get contributions from gluino pair production, and from squark production in the presence of long decay chains and/or additional QCD radiation.

- $E_T^j > 100,\ 50,\ 50,\ 50\,\mathrm{GeV}$ for the four hardest jets

- Find the smallest invariant mass of two of the four hardest jets, $m^{jj}_{\min}$ (there are six combinations). Find the smallest invariant mass of three of the four jets, $m^{3j}_{\min}$ (there are four combinations). At least one of the following three requirements has to be fulfilled, in order to suppress $t\bar{t}$ backgrounds:

  Either: $m^{jj}_{\min} > 100\,\mathrm{GeV}$

  Or: $m^{3j}_{\min} > 200\,\mathrm{GeV}$

  Or: None of the four jets is a b–jet

- If there is exactly one lepton then its transverse mass with the missing transverse momentum in the event should fulfill $m_T(\vec{p}^{\,l}, \vec{\not{p}}_T) > 80\,\mathrm{GeV}$; this again suppresses backgrounds where the missing p_T originates from a W–decay

B.2. Events with Two Opposite–Charged Leptons

We impose the following cuts:

- $\not{p}_T > 100\,\mathrm{GeV}$
- If the two leptons are a e^+e^- or a $\mu^+\mu^-$ pair their invariant mass should satisfy one of the following requirements, which will remove events where the leptons come from a $Z \to l^+l^-$ decay:

  Either: $m^{ll} \leq 75\,\mathrm{GeV}$

  Or: $m^{ll} \geq 105\,\mathrm{GeV}$

In addition, we demand that the event contains either no, at least three or at least four jets; in the latter two cases we apply additional cuts in order to suppress the $t\bar{t}$ background:

B.2.1. No Jets

- There are no jets with $E^j_T > 30\,\mathrm{GeV}$

B.2.2. Three or More Jets

- There are at least three jets with $E^j_T > 100,\ 100,\ 50\,\mathrm{GeV}$
- None of the three highest E^j_T jets has been tagged as a b–jet

B.2.3. Four or More Jets

- There are at least four jets, with $E^j_T > 100,\ 50,\ 50,\ 50\,\mathrm{GeV}$

B.3. Events with at Least Two Leptons of the Same Charge

We impose the following cuts:

- $\not{p}_T > 50\,\mathrm{GeV}$
- $\not{p}_T > 3 \cdot \sqrt{H_T}$ (in GeV). This is intended to suppress events where the missing p_T is due to mismeasurements; for example it is not unlikely to find 1 TeV measured energy accompanied by more than 50 GeV missing transverse momentum

- Now consider all lepton pairs with opposite charge but same flavor, i.e. e^+e^-, $\mu^+\mu^-$ or $\tau^+\tau^-$. If there is *no* such lepton pair, the event is accepted. Otherwise, we impose the following two additional cuts:

  If the event contains exactly three charged leptons, define the third lepton l_3 as the one *not* counted in the opposite–charged same–flavored lepton pair. The event then has to satisfy $m_T(\vec{p}_{l_3}, \not{\vec{p}}_T) > 80\,\text{GeV}$. If all three leptons are of the same flavor, there are two possible l^+l^- pairs, and hence two possible choices for l_3; in this case this last cut has to be satisfied for both choices.

  In addition, each l^+l^- pair must be an e^+e^- or $\mu^+\mu^-$ pair with invariant mass below 75 GeV or above 105 GeV (to suppress $Z \to l^+l^-$ backgrounds), or the event must contain at least two jets with $E_T^j > 100,\ 100\,\text{GeV}$

C. Calculation of Neural Network Weights

This appendix is published in a similar form in [2]. In this appendix we look at the calculation of the weights of a neural network with $84 + 1$ input, $v + 1$ hidden and 4 output neurons. We combine all weights in one vector $\vec{w}$, which first $v \cdot 85$ entries are from the first weight layer and the following $4 \cdot (v + 1)$ entries are from the second weight layer:

$$\begin{aligned}
\vec{w} &= (w_1,\, \ldots,\, w_W)^T \\
&= (w_1,\, \ldots,\, w_{v \cdot 85},\, w_{v \cdot 85+1},\, \ldots,\, w_W)^T \\
&= \left(w_{10}^{(1)},\, w_{11}^{(1)},\, \ldots,\, w_{1\,84}^{(1)},\, w_{20}^{(1)},\, \ldots,\, w_{v\,84}^{(1)},\, w_{10}^{(2)},\, \ldots,\, w_{4v}^{(2)}\right)^T
\end{aligned}$$

In total the weight vector has $W = v \cdot 85 + 4 \cdot (v + 1)$ entries. The change from one to two indices is done like $w_k = w_{ai}^{(1)}$ with $k = 1, \ldots, v \cdot 85$, $a = \lceil k/85 \rceil$ (The number is rounded up, i.e. e.g. $\lceil 1/85 \rceil = 1$ or $\lceil 1 \rceil = 1$) and $i = [(k - 1) \mod 85]$. The same is done for the second layer weights $w_l = w_{rb}^{(2)}$ with $l = v \cdot 85 + 1, \ldots, W$, $r = \lceil (l - v \cdot 85)/(v + 1) \rceil$ and $b = [(l - 1 - v \cdot 85) \mod (v + 1)]$.

C.1. First Step

In the beginning of learning step t the weight vector is labeled $\vec{w}_t$. As mentioned in Subsection 6.1.2, the first weight vector $\vec{w}_1$ is chosen randomly with Gaussian distributions. To calculate the new weights in the first step, we set the first search direction (for better weights) to $\vec{s}_1 = -\vec{g}_1$ equal to the negative gradient. The first step gradient vector $\vec{g}_1 = \vec{g}(\vec{w}_1) = \vec{\nabla} F(\vec{w}_1)$ is calculated using the error function

$$F = \sum_{n=1}^{N} F^n = \sum_{n=1}^{N} \frac{1}{2} \left(\vec{y}^n(\vec{x}^n) - \vec{k}^n\right)^2 . \tag{C.1}$$

As before, N is the number of training sets and $\vec{y}^n(\vec{x}^n)$ the network output for the normalized input $\vec{x}^n$ from training set n. The correct inversely normalized CMSSM output of the training set is labeled $\vec{k}^n$. The normalizations are done as written in equations (6.5) and (6.8). Using equations (C.1), (6.1) and (6.2) the first $v \cdot 85$ entries in the gradient vector are

$$\vec{\nabla}_k F(\vec{w}) = \frac{\partial F}{\partial w_k} = \frac{\partial F}{\partial w_{ai}^{(1)}} = \sum_{n=1}^{N} \delta_a^{(1)n}\, x_i^n \tag{C.2}$$

and the following $4 \cdot (v + 1)$ entries are

$$\vec{\nabla}_l F(\vec{w}) = \frac{\partial F}{\partial w_l} = \frac{\partial F}{\partial w_{rb}^{(2)}} = \sum_{n=1}^{N} \delta_r^{(2)n}\, h(z_b^n)\,. \tag{C.3}$$

The function h stands for the hidden neuron processing function with $h(z_b^n) = \tanh(z_b^n)$ and $\delta_a^{(1)n}$ with $a = 1, \cdots, v$ and $\delta_r^{(2)n}$ with $r = 1, \cdots, 4$ are abbreviations for:

$$\delta_a^{(1)n} = \sum_{r=1}^{4} \frac{\partial F^n}{\partial y_r^n} \frac{\partial y_r^n}{\partial z_a^n} = \left(\sum_{r=1}^{4} \delta_r^{(2)n} \, w_{ra}^{(2)} \right) \cdot h'(z_a^n) \tag{C.4}$$

$$\delta_r^{(2)n} = \frac{\partial F^n}{\partial y_r^n} = y_r^n - k_r^n \tag{C.5}$$

The first derivative of the hidden neuron function is $h'(z_b^n) = (1 - \tanh^2(z_b^n))$.

C.2. Repeated Steps

The following calculation steps are repeated as long as the desired "best" weights are found. The calculation of the new weight vector is done in step t with the formula

$$\vec{w}_{t+1} = \vec{w}_t + \alpha_t \, \vec{s}_t \tag{C.6}$$

with the search direction $\vec{s}_t$ and the coefficient α_t, which can be calculated using the Hessian matrix H_t in step t like

$$\alpha_t = -\frac{\vec{s}_t^T \, \vec{g}_t}{\vec{s}_t^T \, H_t \, \vec{s}_t} \, . \tag{C.7}$$

The explicit calculation of the Hessian matrix is shown at the end of this appendix. With the new weights the normalized control error from equation (6.9) can be calculated and depending on the stopping criterium the learning process would be finished. We take the smallest control error as stopping criterium and find it by looking at the error evolution of the following learning steps. Note that the global minimum of the control error could be missed, if there is a relatively deep and broad local minimum at an earlier learning step. The error would increase behind such a minimum for a relatively high number of learning steps and the learning procedure could be stopped before the error (unexpectedly) decreases again. This situation is tried to be avoided by looking at a quite high number of steps (usually around 100) after the found minimum.*

If the stopping criterium is not fulfilled, the new gradient vector $\vec{g}_{t+1}$ would be calculated followed by the calculation of the new search direction with

$$\vec{s}_{t+1} = -\vec{g}_{t+1} + \beta_t \, \vec{s}_t \, . \tag{C.8}$$

The coefficient β_t is calculated using the Hessian matrix like

$$\beta_t = \frac{\vec{g}_{t+1}^T \, H_t \, \vec{s}_t}{\vec{s}_t^T \, H_t \, \vec{s}_t} \, . \tag{C.9}$$

*In practice a minimum is determined by comparing the current value of the normalized control error to the value three steps earlier. If the earlier value was smaller, this earlier value would be denoted as a "minimum". In the following, the learning procedure is restarted from this minimum and continued for at least a certain number of steps (e.g. 20). Note that the control error evolution with such a restarting looks somewhat different compared to an error evolution without such breaks, because the calculation of the search direction differs between the first and all other steps. However, the final minimum should not be very different and in particular not more different than a minimum which would be found using different randomly chosen original first weights. Furthermore, the requirement of continuing the learning procedure for a certain number of steps could in principle lead to the missing of a very close better minimum. However, especially around the global minimum the control error (and weights) usually change only very little, i.e. the improvement of such a missed "better" minimum normally should be negligible.

Now the next step $t \to t+1$ starts by calculating the new weights with equation (C.6) again.

C.3. Hessian Matrix

The symmetric Hessian matrix with dimension W differs from step to step. In a particular step the entries are

$$H_{kl} = \frac{\partial^2 F}{\partial w_k \partial w_l}, \tag{C.10}$$

with w_k being the k–th entry of the weight vector $\vec{w}$. We can distinguish three different cases, first, both weights are from the first layer, second, both weights are from the second layer and third, one weight is from the first the other one from the second weight layer:

1) Both weights are from the first weight layer, i.e. $k,\, l = 1,\, \ldots,\, v \cdot 85$:

$$\begin{aligned} H_{kl} &= \frac{\partial^2 F}{\partial w_k \partial w_l} = \sum_{n=1}^{N} \frac{\partial^2 F^n}{\partial w_{ai}^{(1)} \partial w_{bj}^{(1)}} = \sum_{n=1}^{N} \frac{\partial}{\partial w_{ai}^{(1)}} (\delta_b^{(1)n}\, x_j^n) \\ &= \sum_{n=1}^{N} \frac{\partial}{\partial w_{ai}^{(1)}} \left[\left(\sum_{s=1}^{4} \delta_s^{(2)n}\, w_{sb}^{(2)} \right) \cdot h'(z_b^n) \right] x_j^n \\ &= \sum_{n=1}^{N} \left[\left(\sum_{s=1}^{4} w_{sa}^{(2)}\, w_{sb}^{(2)} \right) h'(z_a^n)\, h'(z_b^n) + \delta_{ba} \left(\sum_{s=1}^{4} \delta_s^{(2)n}\, w_{sb}^{(2)} \right) h''(z_b^n) \right] x_i^n\, x_j^n \end{aligned}$$

This can be checked using the formulas (C.1), (C.4), (C.5), (6.1) and (6.2). The second derivative of the hidden neuron processing function is

$$h''(z) = 2\, \tanh(z)\, (\tanh^2(z) - 1) = 2\, h(z)\, (h^2(z) - 1)\,. \tag{C.11}$$

2) Both weights are from the second weight layer, i.e. $k,\, l = v \cdot 85 + 1,\, \ldots,\, W$:

$$\begin{aligned} H_{kl} &= \frac{\partial^2 F}{\partial w_k \partial w_l} = \sum_{n=1}^{N} \frac{\partial^2 F^n}{\partial w_{ra}^{(2)} \partial w_{sb}^{(2)}} = \sum_{n=1}^{N} \frac{\partial}{\partial w_{ra}^{(2)}} (\delta_s^{(2)n}\, h(z_b^n)) \\ &= \sum_{n=1}^{N} \delta_{rs}\, h(z_a^n)\, h(z_b^n) \end{aligned}$$

3) One weight is from the first and the other one from the second weight layer, i.e. $k = v \cdot 85 + 1,\, \ldots,\, W$ and $l = 1,\, \ldots,\, v \cdot 85$ or vice versa:

$$\begin{aligned} H_{kl} &= \frac{\partial^2 F}{\partial w_k \partial w_l} = \sum_{n=1}^{N} \frac{\partial^2 F^n}{\partial w_{ra}^{(2)} \partial w_{bi}^{(1)}} \\ &= \sum_{n=1}^{N} \frac{\partial}{\partial w_{ra}^{(2)}} \left[\left(\sum_{s=1}^{4} \delta_s^{(2)n}\, w_{sb}^{(2)} \right) \cdot h'(z_b^n) \right] x_i^n \\ &= \sum_{n=1}^{N} \left(h(z_a^n)\, w_{rb}^{(2)} + \delta_{ab}\, \delta_r^{(2)n} \right) h'(z_b^n)\, x_i^n \end{aligned}$$

D. Calculations

In this appendix some smaller calculations are presented which are used in certain parts of this thesis.

D.1. Covariance Matrix for Event Fractions

This appendix refers to the statement in Section 4.2 that a covariance matrix including all twelve event fractions n_c/N with $c = 1, \ldots, 12$ would not be invertible, since it would be linear dependent. For example, looking at the first event fraction column, all twelf entries would add up to zero with the use of equation (4.2):

$$\begin{aligned}
\sum_{c=1}^{12} \operatorname{cov}\left(\frac{n_c}{N}, \frac{n_1}{N}\right) &= \sigma^2(\frac{n_1}{N}) + \sum_{c=2}^{12} \operatorname{cov}\left(\frac{n_c}{N}, \frac{n_1}{N}\right) \\
&= \frac{n_1}{N^2} - \frac{n_1^2}{N^3} - \sum_{c=2}^{12} \frac{n_c\, n_1}{N^3} \\
&= \frac{1}{N^3} \cdot (n_1\, N - n_1^2 - (N - n_1)\, n_1) \\
&= 0 \qquad \text{(D.1)}
\end{aligned}$$

In the third line $\sum_{c=2}^{12} n_c = N - n_1$ is used. In the same way, also all other eleven columns would each add up to zero. As a consequence, the twelf event fraction lines are linear dependent as claimed. The invertibility of the covariance matrix can be made possible by dropping one non–vanishing event fraction $n_c/N \neq 0$. This reduces the number of observables from 85 to 84, but does not come along with any information loss about the events.

D.2. Peak at $p \approx 0.32$ for Poisson Statistics

In Section 5.1 it is stated that the peak at $p \approx 0.32$ in Figure 5.2 probably undervalues the true p–value for cases where one data set has one τ–event while the other one has none. Figure 5.2 shows the p–value distribution of the self–comparison which only uses the fractions of events with tagged τ–leptons. If the data set has either one or zero τ–events in one specific class, the best estimate of the true expectation value of the number of τ–events in this class would be 1/2, whereas the difference of the number of class events between both data sets should be neglected. With Poisson statistics the probability of having t events with tagged τ–leptons would then be

$$P(t) = \frac{1}{2^t\, t!} \cdot \exp(-1/2)\,. \qquad \text{(D.2)}$$

With equation (5.5) in Section 5.1 it is shown that the same number of τ–events in both data sets leads to $\chi^2_{AB} \simeq 0$ and the peak at $p \simeq 1$ for the comparison of one class. The peak at $p \approx 0.32$ results from $\chi^2_{AB} \approx 1$. With Poisson statistics the situation of one τ–event in one data set and zero τ–events in the other data set has the probability

$$P(\chi^2_{AB} \approx 1) = 2 \cdot P(0) \cdot P(1) \approx 0.37\,. \tag{D.3}$$

This is higher than the p–value. For $\chi^2_{AB} \geq 1$, equation (5.5) leads to the requirement $(n^A_{c,\tau^-} - n^B_{c,\tau^-})^2 \geq (n^A_{c,\tau^-} + n^B_{c,\tau^-})$. This requirement would for example be fulfilled for $n^A_{c,\tau^-} = 0$ and $n^B_{c,\tau^-} \geq 1$ or vice versa. The probability for $\chi^2_{AB} \geq 1$ calculated with equation (D.2) would be:

$$\begin{aligned}
P(\chi^2_{AB} \geq 1) &= 2 \cdot P(0) \cdot P(t \geq 1) + \; 2 \cdot P(1) \cdot P(t \geq 3) + \; 2 \cdot P(2) \cdot P(t \geq 5) + \; \ldots \\
&= 2 \cdot P(0) \cdot (1 - P(0)) \\
&\quad + 2 \cdot P(1) \cdot (1 - P(0) - P(1) - P(2)) \\
&\quad + 2 \cdot P(2) \cdot (1 - P(0) - P(1) - P(2) - P(3) - P(4)) \\
&\quad + \ldots \\
&= 0.4773 + 0.0087 + 2.6 \cdot 10^{-5} + \cdots \\
&\approx 0.49
\end{aligned} \tag{D.4}$$

The contribution from the higher terms "…" can be neglected.

D.3. Gaussian Distribution for First Weights

In this appendix the Gaussian distributions for the first weights are considered. In Subsection 6.1.2 it is stated, that a Gaussian distribution with mean value 0 and variance 1/85 for the first weight layer and $1/(v+1)$ for the second weight layer, respectively, leads to values on the order $O(1)$ for the corresponding output side.

As written in equation (6.1), the input value z_a of a hidden neuron is

$$z_a = \sum_{i=1}^{84} w^{(1)}_{ai} x_i + w^{(1)}_{a0} = \sum_{i=0}^{84} w^{(1)}_{ai} x_i \tag{D.5}$$

with x_i being the value of the i–th input neuron and $w^{(1)}_{ai}$ being the weight of the connection between the i–th input neuron and the a–th hidden neuron. The expectation value $\overline{z_a}$ then would be

$$\overline{z_a} = \sum_{i=0}^{84} \overline{w^{(1)}_{ai}\, x_i} = \sum_{i=0}^{84} \overline{w^{(1)}_{ai}}\; \overline{x_i} = \overline{w^{(1)}_{a0}} = 0\,. \tag{D.6}$$

The normalized input values x_i and the weights $w^{(1)}_{ai}$ are not correlated, i.e. their mean values are independent from each other. As described in Subsection 6.1.2, the mean values (over the training sets) of the normalized input values x_i with $i = 1, \ldots, 84$ are each equal zero. The zeroth input neuron has the constant value $x_0 = 1$. If the weights from the first weight layer follow a Gaussian distribution with the mean value equal zero, this is in particular valid for $\overline{w^{(1)}_{a0}} = 0$.

The variance $\sigma^2_{z_a}$ for the hidden neuron input value is the difference between the expectation value of the squared value $\overline{z_a^2}$ and the squared expectation value $\overline{z_a}^2$:

$$\begin{aligned} \sigma^2_{z_a} &= \overline{z_a^2} - \overline{z_a}^2 = \overline{\left(\sum_{i=0}^{84} w^{(1)}_{ai}\, x_i\right) \left(\sum_{j=0}^{84} w^{(1)}_{aj}\, x_j\right)} - 0 \\ &= \sum_{i,\,j=0}^{84} \overline{w^{(1)}_{ai}\, w^{(1)}_{aj}}\;\overline{x_i\, x_j} = \sum_{i=0}^{84} \sigma^2_{w^{(1)}} \overline{x_i^2} = \sum_{i=0}^{84} \sigma^2_{w^{(1)}} \\ &= \sigma^2_{w^{(1)}} \cdot 85 \end{aligned} \tag{D.7}$$

In the second line the non–correlation of the normalized input values and the weights is used again. Furthermore, weights from different connections are also not correlated as they are chosen independently from each other, i.e. for $i \neq j$ the expectation value for both weights can be considered separately with

$$\overline{w^{(1)}_{ai}\, w^{(1)}_{aj}} = \overline{w^{(1)}_{ai}}\;\overline{w^{(1)}_{aj}} = 0\,. \tag{D.8}$$

For the same weights, i.e. $i = j$, the term can be written as

$$\overline{w^{(1)\,2}_{ai}} = \overline{w^{(1)\,2}_{ai}} - 0 = \overline{w^{(1)\,2}_{ai}} - \overline{w^{(1)}_{ai}}^2 = \sigma^2_{w^{(1)}}\,. \tag{D.9}$$

Then overall the term is $\overline{w^{(1)}_{ai}\, w^{(1)}_{aj}} = \delta_{ij}\sigma^2_{w^{(1)}}$. In the next step $\overline{x_i^2} = \overline{x_i^2} - \overline{x_i}^2 = \sigma^2_{x_i}$ is used, whereas the mean value of the normalized input values is zero and the corresponding variance equals one as chosen in Subsection 6.1.2. With the choice $\sigma^2_{w^{(1)}} = 1/85$, the variance for the hidden neuron input values would be $\sigma^2_{z_a} = 1$, i.e. in particular the input values would be on the order $O(1)$ as stated.

In the same way, the variance for the first choice of the second layer weights can be determined. Using equation (6.2) the expectation value of the r–th output neuron would be

$$\overline{y_r} = \sum_{a=0}^{v} \overline{w^{(2)}_{ra}\, \tanh(z_a)} = \sum_{a=0}^{v} \overline{w^{(2)}_{ra}}\;\overline{\tanh(z_a)} = 0\,. \tag{D.10}$$

As before, the weights and the output values of the hidden neurons are not correlated. The weights of the second weight layer should also be chosen from a Gaussian distribution with a mean value equal zero, i.e. the weight expectation value $\overline{w^{(2)}_{ra}} = 0$. The variance of the output value would then be:

$$\begin{aligned} \sigma^2_{y_r} &= \overline{y_r^2} - \overline{y_r}^2 = \overline{\left(\sum_{a=0}^{v} w^{(2)}_{ra}\, \tanh(z_a)\right) \left(\sum_{b=0}^{v} w^{(2)}_{rb}\, \tanh(z_b)\right)} - 0 \\ &= \sum_{a=0}^{v} \sigma^2_{w^{(2)}}\, \overline{\tanh^2(z_a)} \lesssim \sigma^2_{w^{(2)}} \cdot (v+1) \end{aligned} \tag{D.11}$$

As before, $\overline{w^{(2)}_{ra}\, w^{(2)}_{rb}} = \delta_{ab}\sigma^2_{w^{(2)}}$ and $|\tanh(z)| \leq 1$ for $z \in \mathbb{R}$. Therefore $\sigma^2_{w^{(2)}} = 1/(v+1)$ should lead to output values on the order $O(1)$.

D.4. Set–up of a Correlated Gaussian Distribution

In this appendix, the set–up of a correlated Gaussian distribution is explained as it is for example used for the creation of the training set copies in Subsection 6.2.2. The vector $\vec{x}$ should include the

Gaussian distributed variables, which are here the in Chapter 4 described 84 measured observables. For each training set copy, the observables should follow the correlated Gaussian distribution

$$g(\vec{x},\, \bar{\vec{x}},\, V) = \frac{1}{k} \cdot \exp[-\frac{1}{2}\,(\vec{x} - \bar{\vec{x}})^T\, V^{-1}\, (\vec{x} - \bar{\vec{x}})]\,. \tag{D.12}$$

The vector $\bar{\vec{x}}$ includes the mean values of the observables, which are the measured observables of the to be copied training set, and V is the covariance matrix including the variances and covariances of the measured observables. The factor k is just for scaling. In the following it is assumed that such a multidimensional correlated Gaussian distribution is not available, but instead only one–dimensional Gaussian distributions. The multidimensional distribution can be expressed with multiple one–dimensional distributions. To do so, 84 variables y_i with $i = 1, \ldots, 84$ are independently chosen from an one–dimensional Gaussian distribution with the mean value equal zero and the variance equal one in the form

$$g(y_i,\, 0,\, 1) = \frac{1}{k'} \cdot \exp[-\frac{1}{2}\, y_i^2]\,. \tag{D.13}$$

The 84 y_i summarized in the vector $\vec{y}$ can be expressed as a multidimensional uncorrelated Gaussian distribution like

$$g(\vec{y},\, \vec{0},\, \mathbb{1}) = \frac{1}{k''} \cdot \exp[-\frac{1}{2}\, \vec{y}^T\, \mathbb{1}\, \vec{y}]\,. \tag{D.14}$$

The symmetric positive definite covariance matrix V in equation (D.12) can be written as $V = LL^T$, doing a Cholesky decomposition. The correlated Gaussian distributed vector $\vec{x}$ can then be expressed with the uncorrelated Gaussian distributed vector $\vec{y}$ with

$$\vec{x} = L\vec{y} + \bar{\vec{x}}\,. \tag{D.15}$$

In the following it is shown that $\vec{x}$ from equation (D.15) inserted into the correlated distribution in equation (D.12) can be expressed as $\vec{y}$ following an uncorrelated distribution as in formula (D.14):

$$\begin{aligned} g(\vec{x},\, \bar{\vec{x}},\, V) &= \frac{1}{k} \cdot \exp[-\frac{1}{2}\,(\vec{x} - \bar{\vec{x}})^T\, V^{-1}\, (\vec{x} - \bar{\vec{x}})] \\ &= \frac{1}{k} \cdot \exp[-\frac{1}{2}\,(L\vec{y})^T\, (LL^T)^{-1}\, (L\vec{y})] \\ &= \frac{1}{k} \cdot \exp[-\frac{1}{2}\,\vec{y}^T L^T\, (L^T)^{-1} L^{-1}\, L\vec{y}] \\ &= \frac{1}{k} \cdot \exp[-\frac{1}{2}\,\vec{y}^T\, \mathbb{1}\, \vec{y}] \end{aligned} \tag{D.16}$$

The scaling factors can be picked appropriately with $k = k''$. With the creation of 84 independently Gaussian distributed values $\vec{y}$ with equation (D.14) and the use of equation (D.15), 84 correlated Gaussian distributed values $\vec{x}$ can be generated which follow equation (D.12).

The described creation of the correlated Gaussian distribution can for example be done using the GNU Scientific Library [19].

Bibliography

[1] N. Bornhauser, M. Drees, *Phys. Rev.* **D86** (2012) 015025 [arXiv:1205.6080 [hep-ph]].

[2] N. Bornhauser, M. Drees, arXiv:1307.3383 [hep-ph].

[3] ATLAS collab., G. Aad *et al.*, *Phys. Rev. Lett.* **106** (2011) 131802 [arXiv:1102.2357 [hep-ex]]; *Phys. Lett.* **B701** (2011) 186 [arXiv:1102.5290 [hep-ex]]; *Phys. Lett.* **B701** (2011) 398 [arXiv:1103.4344 [hep-ex]]; *Eur. Phys. J.* **C71** (2011) 1682 [arXiv:1103.6214 [hep-ex]]; *Phys. Lett.* **B710** (2012) 67 [arXiv:1109.6572 [hep-ex]]; *Phys. Rev.* **D85** (2012) 012006 [arXiv:1109.6606 [hep-ex]]; *Phys. Lett.* **B709** (2012) 137 [arXiv:1110.6189 [hep-ex]]; *Eur. Phys. J.* **C72** (2012) 1993 [arXiv:1202.4847 [hep-ex]]; *Phys. Rev. Lett.* **108** (2012) 241802 [arXiv:1203.5763 [hep-ex]]; *Phys. Rev.* **D85** (2012) 112006 [arXiv:1203.6193 [hep-ex]]; *Phys. Lett.* **B714** (2012) 197 [arXiv:1204.3852 [hep-ex]]; *Phys. Rev. Lett.* **108** (2012) 261804 [arXiv:1204.5638 [hep-ex]]; *Phys. Lett.* **B715** (2012) 44 [arXiv:1204.6736 [hep-ex]]; *Eur. Phys. J.* **C72** (2012) 2174 [arXiv:1207.4686 [hep-ex]]; *Phys. Rev.* **D87** (2013) 012008 [arXiv:1208.0949 [hep-ex]]; *Phys. Rev. Lett.* **109** (2012) 211802 [arXiv:1208.1447 [hep-ex]]; *Phys. Rev. Lett.* **109** (2012) 211803 [arXiv:1208.2590 [hep-ex]]; *Phys. Lett.* **B718** (2013) 879 [arXiv:1208.2884 [hep-ex]]; *Phys. Lett.* **B718** (2013) 841 [arXiv:1208.3144 [hep-ex]]; *Eur. Phys. J.* **C72** (2012) 2237 [arXiv:1208.4305 [hep-ex]]; *Phys. Rev.* **D86** (2012) 092002 [arXiv:1208.4688 [hep-ex]]; *Phys. Lett.* **B720** (2013) 13 [arXiv:1209.2102 [hep-ex]]; *Eur. Phys. J.* **C72** (2012) 2215 [arXiv:1210.1314 [hep-ex]]; *JHEP* **1301** (2013) 131 [arXiv:1210.2852 [hep-ex]]; *JHEP* **1212** (2012) 124 [arXiv:1210.4457 [hep-ex]]; *Phys. Lett.* **B719** (2013) 261 [arXiv:1211.1167 [hep-ex]]; *Phys. Lett.* **B720** (2013) 277 [arXiv:1211.1597 [hep-ex]]; *JHEP* **02** (2013) 095 [arXiv:1211.6956 [hep-ex]]; and *Eur. Phys. J.* **C73** (2013) 2362 [arXiv:1212.6149 [hep-ex]]; CMS collab., S. Chatrchyan *et al.*, *Phys. Lett.* **B698** (2011) 196 [arXiv:1101.1628 [hep-ex]]; *JHEP* **07** (2011) 113 [arXiv:1106.3272 [hep-ex]]; *JHEP* **08** (2011) 156 [arXiv:1107.1870 [hep-ex]]; *Phys. Rev. Lett.* **107** (2011) 221804 [arXiv:1109.2352 [hep-ex]]; *Phys. Lett.* **B716** (2012) 260 [arXiv:1204.3774 [hep-ex]]; *JHEP* **08** (2012) 110 [arXiv:1205.3933 [hep-ex]]; *Phys. Rev. Lett.* **109** (2012) 071803 [arXiv:1205.6615 [hep-ex]]; *Phys. Lett.* **B718** (2013) 815 [arXiv:1206.3949 [hep-ex]]; *JHEP* **10** (2012) 018 [arXiv:1207.1798 [hep-ex]]; *Phys. Rev. Lett.* **109** (2012) 171803 [arXiv:1207.1898 [hep-ex]]; *Phys. Rev.* **D86** (2012) 072010 [arXiv:1208.4859 [hep-ex]]; *JHEP* **11** (2012) 147 [arXiv:1209.6620 [hep-ex]]; *Phys. Lett.* **B719** (2013) 42 [arXiv:1210.2052 [hep-ex]]; *JHEP* **01** (2013) 077 [arXiv:1210.8115 [hep-ex]]; *Phys. Rev.* **D87** (2013) 052006 [arXiv:1211.3143 [hep-ex]]; *JHEP* **03** (2013) 111 [arXiv:1211.4784 [hep-ex]]; *JHEP* **03** (2013) 037 [arXiv:1212.6194 [hep-ex]]; *Eur. Phys. J.* **C73** (2013) 2404 [arXiv:1212.6428 [hep-ex]]; arXiv:1212.6961 [hep-ex]; *Phys. Rev.* **D87** (2013) 072001 [arXiv:1301.0916 [hep-ex]]; arXiv:1301.2175 [hep-ex]; arXiv:1301.3792 [hep-ex]; arXiv:1303.2985 [hep-ex]; arXiv:1305.2390 [hep-ex]; and arXiv:1306.6643 hep-ex].

[4] M. Drees, R. M. Godbole and P. Roy, "Theory and Phenomenology of Sparticles", World Scientific, Singapore (2004); For a "shorter" introduction into supersymmetry see for example: J. Beringer *et al.* (Particle Data Group), *Phys. Rev.* **D86** (2012) 010001 [`http://pdg.lbl.gov/2012/reviews/contents_sports.html`].

[5] N. Arkani–Hamed, G. L. Kane, J. Thaler and L.-T. Wang, *JHEP* **0608** (2006) 070 [hep-ph/0512190].

[6] B. Denby, *Comput. Phys. Commun.* **49** (1988) 429.

[7] CMS Collab., S. Chatrchyan et al., *Phys. Rev.* **D87** (2013) 072001 [arXiv:1301.0916 [hep-ex]].

[8] C. S. Deans, arXiv:1304.2781 [hep-ph], and references therein.

[9] B.C. Allanach, *Comput. Phys. Commun.* **143** (2002) 305 [hep-ph/0104145].

[10] A. Djouadi, M. Mühlleitner and M. Spira, *Acta Phys. Polon.* **B38** (2007) 635 [hep-ph/0609292].

[11] M. Bähr *et al.*, *Eur. Phys. J.* **C58** (2008) 639 [arXiv:0803.0883 [hep-ph]].

[12] P. M. Nadolsky *et al.*, *Phys. Rev.* **D78** (2008) 013004 [arXiv:0802.0007 [hep-ph]].

[13] M. Cacciari, G. P. Salam and G. Soyez, *Eur. Phys. J.* **C72** (2012) 1896 [arXiv:1111.6097 [hep-ph]].

[14] J. Alwall *et al.*, *JHEP* **1106** (2011) 128 [arXiv:1106.0522 [hep-ph]].

[15] C. M. Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, Great Britain (2008).

[16] ATLAS Collab., G. Aad et al., *Phys. Lett.* **B716** (2012) 1 [arXiv:1207.7214 [hep-ex]]; arXiv:1307.1427 [hep-ex]; and arXiv:1307.1432 [hep-ex]; CMS Collab., S. Chatrchyan et al., *Phys. Lett.* **B716** (2012) 30 [arXiv:1207.7235 [hep-ex]]; *Phys. Rev. Lett.* **110** (2013) 081803 [arXiv:1212.6639 [hep-ex]]; and *JHEP* **06** (2013) 081 [arXiv:1303.4571 [hep-ex]].

[17] `seal.cern.ch/documents/minuit/mnusersguide.pdf`.

[18] R. Brun and F. Rademakers, *Nucl. Inst. and Meth.* **A389** (1997) 81 [`http://root.cern.ch/`].

[19] M. Galassi *et al.*, GNU Scientific Library Reference Manual (3rd Ed.), ISBN 0954612078 [`http://www.gnu.org/software/gsl/`].

List of Figures

List of Tables