$B^0 \to K^0_S \pi^0$ and Direct CP Violation at Belle

Anton Hawthorne-Gonzalvez

orcid.org/0000-0002-7387-2120

Submitted in total fulfilment of the requirements of the degree of Master of Philosophy

August 2017

School of Physics
University of Melbourne

Abstract

Rare B-meson decays such as the $B^0 \to K_s\pi^0$ which proceed without a charm quark provide a probe for physics beyond the standard model. This decay proceeds mainly via the $b \to s$ penguin transition, with the $b \to u$ transition being colour suppressed, allowing CP-violating effects to be observable.

The asymmetric e^+e^- KEKB collider and the Belle detector provide the large luminosity and data collection required to observe these rare B decays.

Methods to reduce the large $q\bar{q}$ backgrounds are investigated. The use of optimised neural networks using TensorFlow shows a significant improvement compared to the commonly used NeuroBayes software. Techniques for reducing correlations between variables introduced by TensorFlow are also investigated, proving that the use of adversarial neural networks can provide an improved background suppression as compared to NeuroBayes, whilst minimising correlations introduced by the neural network.

An improved method of measuring the direct CP violation is introduced. Using Monte Carlo data with sample sizes corresponding to the full Belle datatset of $(771.581 \pm 10.566) \times 10^6 B\bar{B}$ events, the statistical uncertainty in $\mathcal{A}_{CP}$ using this method is reduced from the latest Belle result of 0.13 to 0.1035 ± 0.0032. This method would also provide an up to date measurement on $\mathcal{B}(B^0 \to K^0\pi^0)$.

Declaration

This is to certify that:

1. The work in this thesis is my original work towards the qualification of Master of Philosophy except where indicated in the preface.

2. All other materials and sources used have been duly acknowledged in the text.

3. This thesis is fewer than 40,000 words in length, exclusive of tables, bibliographies and appendices.

Anton Hawthorne-Gonzalvez

Preface

All work is entirely my own (unless stated otherwise), other than in the following chapters, where the work is described in [1]:

- **Chapter 4**: All work, time, thought and effort towards the event reconstruction and particle selection as laid out in Chapter 4 was produced by James Kahn.
- **Chapter 6**: Some of the work towards setting up NeuroBayes was also undertaken by James Kahn.

Acknowledgements

Above all else I would like to extend my deep and sincere gratitude to Martin Sevior who took me on as a student. He has always been extremely understanding and patient. Without his guidance and support this work would not have been possible.

I must also extend my gratitude to James Kahn, who laid the foundations on which this work was built, and provided me with a head start in this analysis.

Experimental particle physics is a huge collaborative effort, so to each and every member of Belle and KEKB who have made it all possible, thank you.

I would like to thank Phillip Urquijo and Michele Trenti who were always happy to help and offer wise advice.

David Dossett and Tristan Bloomfield would always be on hand and happy to help. Chia-Ling Hsu deserves a special mention for her super-human patience, without which I would have had a *much* harder time.

Wessam Bader, for everything, thank you.

Finally, a big thank you to everyone who has made my experience of Melbourne such a good one, and to my loved ones in Melbourne and around the world.

Contents

List of Figures

List of Tables

1|Introduction

The standard model of particle physics is the most complete theory we have, accurately describing the constituents of matter and (apart from gravity) all the known forces of nature. The fundamental particles are the three generations of quarks and leptons (along with their anti-particles) which make the atoms and molecules, the force mediator bosons, and the famous Higgs boson, see Figure 1.1.

Although the standard model has proven excellent at producing testable results to a high accuracy and predicting the existence of later discovered particles (the latest of which being the discovery of the long anticipated Higgs boson in 2012[2]), it is not a complete theory of everything. Aside from not including gravity (and large scale phenomena such as dark matter and dark energy) at all, there are measurable failures at the high energy scale. One example being the phenomena of neutrino oscillation, requiring that the neutrinos have (small but non-zero) mass[3]. There is also a large matter-antimatter asymmetry in the universe. A priori we would expect equal abundances of matter and antimatter. We know that there is an asymmetry as there are no regions of space where anti-matter dominates (we would observe interactions at the boundary), and there are no known processes by which pockets of matter and anti-matter could be separated at the scale of the observable universe[4]. In addition to this, studies of the cosmic microwave background show that the ratio of the number baryonic particles(inclusive of baryons and anti-baryons) to photons in the universe is in the range 5.8×10^{-10} to 6.6×10^{-10}[5]. As matter and anti-matter annihilate, this number shows an overabundance of matter to anti-matter of around 1 in 10^9. Electroweak baryogenesis in the standard model on the other hand puts an upper limit on the ratio of baryons to photons of 10^{-26}[6]. Clearly the baryon asymmetry must come from physics beyond the standard model.

In 1967 Andrei Sakharov proposed the following three conditions necessary for baryogenesis(the process creating the baryon number asymmetry) to occur[7]:

1. Baryon number violation. Processes in which $\Delta N_B \neq 0$ are obviously needed in order to produce the baryon number asymmetry.

2. Violation of C-parity (an operation exchanging a particle for its anti-particle and vice versa) and CP (applying C along with flipping the spatial coordinates).

3. A period in which the universe is out of thermal equilibrium. Necessary as any baryon number violating process would just as likely occur in reverse.

Although CP violation does not occur in electromagnetic processes, it does occur

$$\begin{array}{ccc} \text{Quarks} & \text{Leptons} & \text{Bosons} \\ \begin{pmatrix} u & c & t \\ d & s & b \end{pmatrix} & \begin{pmatrix} e & \mu & \tau \\ \nu_e & \nu_\mu & \nu_\tau \end{pmatrix} & \begin{pmatrix} H & \gamma & Z & W^\pm & g \end{pmatrix} \end{array}$$

Figure 1.1: Showing the particles of the standard model. The three generations of quarks (top and bottom rows having charges $+\frac{2}{3}$ and $-\frac{1}{3}$ respectively). The three generations of lepton - the (massive) top row (charge -1) and their corresponding chargeless neutrinos (massless in the standard model). The bosons, comprised of the spin-0 Higgs boson and the spin-1 force mediators; the photon (massless, chargeless) - electromagnetic force, Z (massive, chargeless) and $W^\pm$(massive, charge of ± 1) - weak force, g (massless, chargeless) - strong force.

in weak processes. Quantum chromodynamics allows CP violation, although it has never been observed in strong interactions.

Decays of b quarks provide a rich avenue for investigation into CP and flavour physics. B factories such as Belle and BaBar produce B mesons at extremely high luminosities and allow us to provide tight constrains on CP violating processes and probe for physics beyond the standard model.

1.1 *CP* violation

Applying the parity operator P flips the spacial coordinates, thus changing a particle from a right-handed(spin aligned with momentum) particle into a left-handed(spin anti-aligned to momentum) particle and vice versa (and similarly for antiparticles). So in the case of a neutrino:

$$\begin{aligned} P\,|\nu_r\rangle &= |\nu_l\rangle \\ P\,|\bar{\nu}_l\rangle &= |\bar{\nu}_r\rangle \end{aligned} \tag{1.1}$$

P-violation was first observed in 1957 in the famous Wu experiment in which beta decays (a weak process) of cobalt-60 were found to have a directional preference which maximally violated P-symmetry[8].

The application of CP takes a left handed particle to a right handed antiparticle and vice versa:

$$\begin{aligned} CP\,|\nu_l\rangle &= |\bar{\nu}_r\rangle \\ CP\,|\bar{\nu}_r\rangle &= |\nu_l\rangle \end{aligned} \tag{1.2}$$

Violation of CP-symmetry occurs via three processes, CP violation through decay, CP-violation through mixing, and interference between the two.

CP violation can be explored in the case of neutral meson mixing in which a neutral meson oscillates between its CP eigenstates. This was first observed in in kaon sector by Cronin and Fitch in 1964[9]. The combination of CP with time reversal (CPT) is thought to be an unbroken symmetry of nature.

CP violation theory for neutral mesons will now be introduced, following the full analysis in [10]. The superposition of states of a neutral meson P^0 can be written

as:

$$|\psi(t)\rangle = \alpha(t)\left|P^0\right\rangle + \beta(t)\left|\bar{P}^0\right\rangle \tag{1.3}$$

Where $|P^0\rangle$ and $\left|\bar{P}^0\right\rangle$ are flavour - strong and electromagnetic - eigenstates, and $\alpha(t)$ and $\beta(t)$ are their complex time dependent coefficients. This wave-function is governed by the Schrödinger equation:

$$H\,|\psi(t)\rangle = i\frac{\partial\,|\psi(t)\rangle}{\partial t} \tag{1.4}$$

Where H can be decomposed into:

$$H = M - \frac{i}{2}\Gamma \tag{1.5}$$

Where M gives a mass term, and Γ describes the exponential decay, both are Hermitian (2×2) matrices. Expressing the two mass-eigenstates P_1 and P_2 as superpositions of the flavour eigenstates gives:

$$\begin{aligned} |P_1\rangle &= p\left|P^0\right\rangle - q\left|\bar{P}^0\right\rangle \\ |P_2\rangle &= p\left|P^0\right\rangle + q\left|\bar{P}^0\right\rangle \end{aligned} \tag{1.6}$$

The complex coefficients p and q are retrieved by solving for the eiganvalues of 1.5 and 1.6 giving:

$$\frac{p}{q} = \sqrt{\frac{M_{12}^* - \frac{i}{2}\Gamma_{12}^*}{M_{12} - \frac{i}{2}\Gamma_{12}}} \tag{1.7}$$

Where the subscript 12 corresponds to the second element in the first row for matrices M and Γ. Now looking at the decay of the flavour eigenstates $|P^0\rangle$ and $\left|\bar{P}^0\right\rangle$, to final states $|f\rangle$ and $\left|\bar{f}\right\rangle$, we have the following decay amplitudes:

$$\begin{aligned} A_f &= \langle f|\,O\left|P^0\right\rangle \\ \bar{A}_f &= \langle f|\,O\left|\bar{P}^0\right\rangle \\ A_{\bar{f}} &= \left\langle \bar{f}\right|O\left|P^0\right\rangle \\ \bar{A}_{\bar{f}} &= \left\langle \bar{f}\right|O\left|\bar{P}^0\right\rangle \end{aligned} \tag{1.8}$$

By defining:

$$\lambda_f = \frac{q}{p}\frac{\bar{A}_f}{A_f} \tag{1.9}$$

and:

$$\begin{aligned} D_f &= \frac{2\,\mathrm{Re}(\lambda_f)}{1+|\lambda_f|^2} \\ C_f &= \frac{1-|\lambda_f|^2}{1+|\lambda_f|^2} \\ S_f &= \frac{2\,\mathrm{Im}(\lambda_f)}{1+|\lambda_f|^2} \end{aligned} \tag{1.10}$$

we can express the decay rates of P^0 and $\bar{P}^0$ to f at time t as:

$$\Gamma_{P^0\to f}(t) = |A_f|^2(1+|\lambda_f|^2)\frac{e^{-t\Gamma_{CP}}}{2}\Big[\cosh\left(\frac{\Delta\Gamma t}{2}\right) + D_f\sinh\left(\frac{\Delta\Gamma t}{2}\right) + C_f\cos\left(\Delta m t\right) - S_f\sin\left(\Delta m t\right)\Big] \tag{1.11}$$

$$\Gamma_{\bar{P}^0\to f}(t) = \left|\frac{p}{q}\right|^2|A_f|^2(1+|\lambda_f|^2)\frac{e^{-t\Gamma_{CP}}}{2}\Big[\cosh\left(\frac{\Delta\Gamma t}{2}\right) + D_f\sinh\left(\frac{\Delta\Gamma t}{2}\right) - C_f\cos\left(\Delta m t\right) + S_f\sin\left(\Delta m t\right)\Big] \tag{1.12}$$

Where $\Gamma_{CP} = \frac{1}{2}(\Gamma_1+\Gamma_2)$, $\Delta\Gamma = \Gamma_2 - \Gamma_1$, and $\Delta m = m_2 - m_1$ given the decay rates $(\Gamma_{1/2})$and masses $(m_{1/2})$ of P_1 and P_2 respectively.

1.1.1 Direct *CP* Violation

CP violation through decay, or direct *CP* violation, occurs when the decay rate of a particle P^0 to a final state f is not equal to the decay rate of the *CP* conjugate of this process ($\bar{P}^0$ to $\bar{f}$):

$$\Gamma(P^0\to f) \neq \Gamma(\bar{P}^0\to\bar{f}) \tag{1.13}$$

i.e. when:

$$\left|\frac{\bar{A}_{\bar{f}}}{A_f}\right| \neq 1 \tag{1.14}$$

We define the *CP* asymmetry as:

$$\mathcal{A}_{CP} = \frac{\Gamma(P^0\to f) - \Gamma(\bar{P}^0\to\bar{f})}{\Gamma(P^0\to f) + \Gamma(\bar{P}^0\to\bar{f})} \tag{1.15}$$

Which in the case of $|f\rangle$ being a *CP* eigenstate (i.e. $CP|f\rangle = \pm|f\rangle = |\bar{f}\rangle$) becomes:

$$\mathcal{A}_{CP} = \frac{\Gamma(P^0\to f) - \Gamma(\bar{P}^0\to f)}{\Gamma(P^0\to f) + \Gamma(\bar{P}^0\to f)} \tag{1.16}$$

And occurs when:

$$\left|\frac{\bar{A}_f}{A_f}\right| \neq 1 \tag{1.17}$$

This is possible when there are two leading channels proceeding via the weak and strong forces. Direct *CP* violation cannot proceed through a *CP* violating phase in the weak sector alone, which can be shown as follows. Assuming the process $P^0\to f$ proceeds via one channel, we have *CP* conjugate decay amplitudes given by:

$$\begin{aligned} A_f &= |a|e^{i\delta}e^{i\phi} \\ \bar{A}_{\bar{f}} &= |a|e^{-i\delta}e^{i\phi} \end{aligned} \tag{1.18}$$

Where δ is the weak phase and ϕ is the CP invariant strong phase and a is the decay amplitude. Then there is no CP violation as:

$$\left|\frac{\bar{A}_{\bar{f}}}{A_f}\right|^2 = \left|\frac{e^{i\delta}e^{i\phi}e^{-i\delta}e^{i\phi}}{e^{-i\delta}e^{i\phi}e^{i\delta}e^{i\phi}}\right| = 1 \tag{1.19}$$

Instead, assume we have the decay proceeding via two Feynman diagrams but with only a weak phase:

$$\begin{aligned} A_f &= |a|e^{i\delta_1} + |b|e^{i\delta_2} \\ \bar{A}_{\bar{f}} &= |a|e^{-i\delta_1} + |b|e^{-i\delta_2} \end{aligned} \tag{1.20}$$

Where a and b are the decay amplitudes corresponding to each decay channel. Then there is no CP violation as:

$$\left|\frac{\bar{A}_{\bar{f}}}{A_f}\right|^2 = \left|\frac{|a|^2e^{i\delta_1}e^{-i\delta_1} + |b|^2e^{i\delta_2}e^{-i\delta_2} + |a||b|e^{i\delta_1}e^{-i\delta_2} + |a||b|e^{i\delta_2}e^{-i\delta_1}}{|a|^2e^{-i\delta_1}e^{i\delta_1} + |b|^2e^{-i\delta_2}e^{i\delta_2} + |a||b|e^{-i\delta_1}e^{i\delta_2} + |a||b|e^{-i\delta_2}e^{i\delta_1}}\right| = 1 \tag{1.21}$$

Finally, assuming two Feynman diagrams contributing to the decay, and both having strong and weak phases:

$$\begin{aligned} A_f &= |a|e^{i\delta_1}e^{i\phi_1} + |b|e^{i\delta_2}e^{i\phi_2} \\ \bar{A}_{\bar{f}} &= |a|e^{-i\delta_1}e^{i\phi_1} + |b|e^{-i\delta_2}e^{i\phi_2} \end{aligned} \tag{1.22}$$

We get :

$$\begin{aligned} |A_f|^2 &= |a|^2 + |b|^2 + 2|a||b|\cos(\Delta\phi + \Delta\delta) \\ \left|\bar{A}_f\right|^2 &= |a|^2 + |b|^2 + 2|a||b|\cos(\Delta\phi - \Delta\delta) \end{aligned} \tag{1.23}$$

where $\Delta\delta$ and $\Delta\phi$ are the differences between δ_1 and δ_2, and between ϕ_1 and ϕ_2 respectively. As can be seen, we can only get CP violation through decays if there are both the strong and weak phases from multiple Feynman diagrams.

1.1.2 *CP* Violation Through Mixing

In the neutral meson case, CP violation can proceed via particle oscillations when:

$$\text{Rate}(P^0 \to \bar{P}^0) \neq \text{Rate}(\bar{P}^0 \to P^0) \tag{1.24}$$

The CP asymmetry in mixing is given by:

$$\mathcal{A}_{CP} = \frac{|p/q|^2 - |q/p|^2}{|p/q|^2 + |q/p|^2} \tag{1.25}$$

and is of course non-zero if:

$$\left|\frac{p}{q}\right| \neq 1 \tag{1.26}$$

as the oscillation probabilities will not be equal (where p/q is given by 1.7). These decays are measured from semi-leptonic decays as the flavour of the neutral meson at decay time is known, so any oscillations can be counted.

1.1.3 *CP* Violation Through Interference

The third type of *CP* violation can proceed when there is interference between *CP* violation in decays and *CP* violation through mixing, even if there are no *CP* asymmetries present in them individually. This occurs when the final state is a *CP* eigenstate (both P^0 and $\bar{P}^0$ can decay to f). There is *CP* violation if the following holds:

$$\Gamma(P^0 \to f)(t) \neq \Gamma(\bar{P}^0 \to f)(t) \tag{1.27}$$

when both processes $P^0 \to \bar{P}^0 \to f$ and $P^0 \to f$ are viable (and similarly when swapping $\bar{P}^0$ and P^0).

Assuming that $|q/p| = 1$ and $|\bar{A}_f/A_f| = 1$ (i.e. that there are no *CP* asymmetries from mixing or directly through decays) we get:

$$\mathcal{A}_{CP}(t) = \frac{-\operatorname{Im}(\lambda_f)\sin(\Delta m t)}{\cosh(\frac{1}{2}\Delta\Gamma t) + \operatorname{Re}(\lambda_f)\sinh(\frac{1}{2}\Delta\Gamma t)} \tag{1.28}$$

Therefore there will still be *CP* violation as long as $\operatorname{Im}(\lambda_f) \neq 0$.

1.2 The CKM Matrix and Flavour Physics

Quark flavour - although unchanged in QCD and QED processes - is not a conserved quantity in weak interactions. This is the only known source of *CP* violation in the standard model that has been verified (*CP* violation *is* permitted in QCD but has never been observed).

The *CP* violating interactions are mediated by the $W^{\pm}$ bosons, allowing flavour changes when converting up-type to down-type quarks and vice versa. The coupling strengths between them are laid out in the CKM (Cabibbo–Kobayashi–Maskawa) matrix:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \tag{1.29}$$

Strictly this gives the quark flavour eigenstates as combinations of the mass eigenstates. So mass eigenstate u couples to flavour eigenstate d', and similarly c to s' and t to b'. The flavour eigenstates are then given by:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V_{CKM} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{1.30}$$

where the couplings V_{ij} are to be measured experimentally.

The CKM matrix - the couplings - can be expressed in terms of rotation angles relating the 1st and 2nd, 1st and 3rd, and 2nd and 3rd generations (θ_{12}, θ_{13} and θ_{23}

respectively), along with the *CP* violating phase (δ)[11]:

$$V_{CKM} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix} \tag{1.31}$$

Where $c_{ij} = \cos(\theta_{ij})$ and $s_{ij} = \sin(\theta_{ij})$. The coupling is very strongly weighted to the same generation, with the diagonal elements being close to one. For example V_{ub} is much smaller than V_{cb}, which is in turn much smaller than V_{tb}.

A common representation of V_{CKM} is the Wolfenstein parameterisation, given by its Taylor expansion in $\lambda(= |V_{us}|)$, which up to third order is given by[12]:

$$V_{CKM} \approx \begin{pmatrix} 1 - \frac{\lambda^2}{2} & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} \tag{1.32}$$

Where A, ρ, and η are of order one. It can be clearly seen from the powers of λ that $V_{cb} >> V_{ub}$.

The CKM matrix must of course be unitary ($V_{CKM}V_{CKM}^{\dagger} = I$) to be physical, as it must conserve probability. This leads to nine equations, three unitarity relations (for each of the diagonal elements in I) and six orthogonality relations (for the off-diagonals). These orthogonality relations give six independent relations that can be represented as triangles in the complex plane. The triangle most commonly chosen is given by:

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0 \tag{1.33}$$

This is known as the unitarity triangle, shown in Figure 1.2. Each side is divided by $V_{cd}V_{cb}^*$ to give a base length of one. The area of the triangle corresponds to the amount of *CP*-violation arising in the weak sector. Firm experimental measurements are required to verify that the angles sum to 180°.

The angles are given by:

$$\phi_1 = arg\left[\frac{-V_{cd}V_{cb}^*}{V_{td}V_{tb}^*}\right] \tag{1.34}$$

$$\phi_2 = arg\left[\frac{-V_{td}V_{tb}^*}{V_{ud}V_{ub}^*}\right] \tag{1.35}$$

$$\phi_3 = arg\left[\frac{-V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right] \tag{1.36}$$

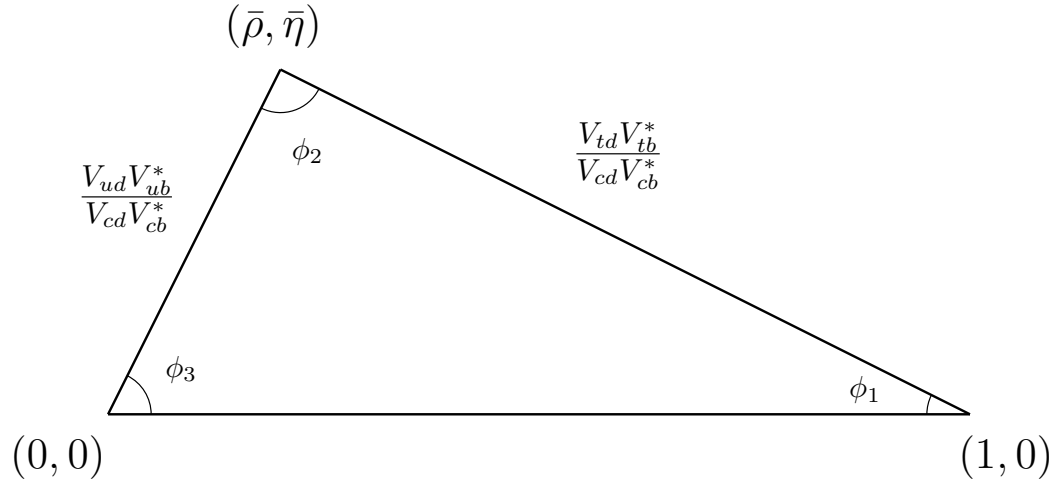

Figure 1.2: Showing the unitarity triangle in the complex plane, where $\bar{\rho} = \rho(1 - \lambda^2/2)$ and $\bar{\eta} = \eta(1 - \lambda^2/2)$.

1.3 B Physics

The decays of b quarks provides a rich area of study in the search for physics beyond the standard model, the measuring of the angles of the unitarity triangle, and investigating *CP* violation. The B meson (composed of a $\bar{b}$ and either a d, u, s or c) in particular exhibits the behaviour laid out above; mixing and *CP* violating decays.

Previous experiments have been used to investigate B physics but the number of $B\bar{B}$ events was small. The CESR collider produced around thirty $B\bar{B}$ events per day. To make new discoveries in this sector, much greater luminosities were needed. The B factories Belle and BaBar were built in the 1990s for this purpose, producing more than one-million $B\bar{B}$ events daily (at the end of their runs)[13].

The production of $B^0\bar{B}^0$ pairs allows us to measure a range of properties of B decays using the information from both B-mesons. The B of interest will be referred to as B_{CP}, as it will show the *CP* violating effects that we measure, and the 'other' B meson as B_{tag}, as we use it to deduce the flavour - 'flavour tagging'. The flavour of B_{tag} at creation tells us the flavour of of B_{CP} at decay time (taking account of mixing effects) and can thus be used for time-independent (i.e. direct) *CP* studies. Time dependent investigations also allow for investigations into *CP* violation from mixing and interference.

This wouldn't be possible without the large number of coherent $B\bar{B}$ pairs produced at the B-factories. They are made by producing Υ mesons at the $4s$ resonance. $\Upsilon(4s)$ (or $\Upsilon(10580)$) is composed of $b\bar{b}$ at a resonance with a mass of (10.5794 ± 0.0012) GeVc^{-2}[5], just above double the mass of a B meson at a mass of (5279.63 ± 0.15) MeVc^{-2}. They decay primarily (more than 96% of the time) to $B\bar{B}$ (48.6% to $B^0\bar{B}^0$)[5].

As shown above, the strongest coupling of the b (to lighter quarks) is V_{cb}, so the vast majority of B decays occur via $b \to c$ transitions. Charmless-rare B decays (named as such due to their very small branching fractions of order 10^{-5}) proceed

mainly via a Cabbibo suppressed tree diagram and a penguin diagram, the amplitudes of which are not of greatly different order. This allows the associated weak and strong phases to produce *CP* violation.

1.4 $B^0 - \bar{B}^0$ Oscillations

Knowledge of the flavour of the B^0_{CP} is obtained from the flavour of B^0_{tag}, given that they are produced in a coherent state. Neutral B meson flavour eigenstates are superpositions of the mass eigenstates. At a time t they are given by[14]:

$$\begin{aligned} \left|B^0(t)\right\rangle &= \frac{|B_L(t)\rangle + |B_H(t)\rangle}{2p} \\ \left|\bar{B}^0(t)\right\rangle &= \frac{|B_L(t)\rangle - |B_H(t)\rangle}{2q} \end{aligned} \tag{1.37}$$

Where B_L and B_H correspond to P_2 and P_1 in Equation 1.6, where B_L and B_H are the mass eigenstates with lower and higher masses respectively. p and q are the complex coefficients from Equation 1.7. As B_H and B_L propagate at different rates, a state $|B^0(t)\rangle$ at time $t = 0$ can then be measured as a $\bar{B}^0$ at time $t = t'$ and vice versa. The dominant Feynman box-diagrams governing this process are given in Figure 1.3.

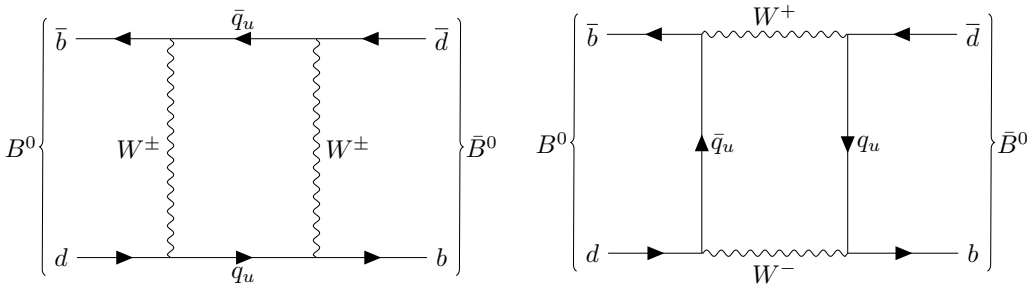

Figure 1.3: Showing the dominant Feynman diagrams responsible for $B^0 - \bar{B}^0$ mixing. q_u corresponds to either a u, c or t quark.

The $B^0\bar{B}^0$ pairs are produced in a coherent state given by[13]:

$$\left|B^0, \bar{B}^0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|B^0\right\rangle\left|\bar{B}^0\right\rangle - \left|\bar{B}^0\right\rangle\left|B^0\right\rangle\right) \tag{1.38}$$

There is always exactly one B^0 and one $\bar{B}^0$ until one of them decays. At this point the other B^0 is free to oscillate. Figure 1.4 shows the an example of an event where a B^0 oscillates before decay.

This process is obviously time-dependent. The time-integrated probability(χ_d) that a B^0 decays as a $\bar{B}^0$ (and vice versa) is 0.186 ± 0.004[5]. This is not a negligible effect.

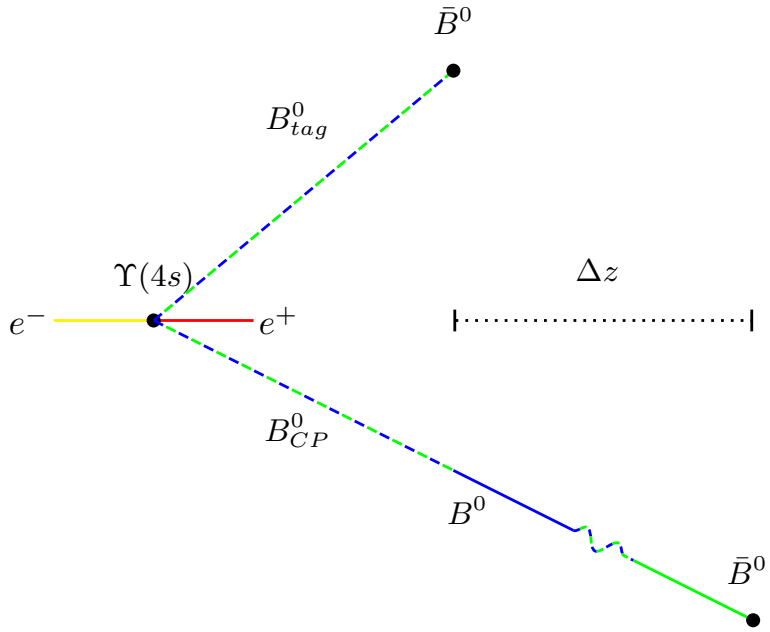

Figure 1.4: Showing the production of a coherent $B^0\bar{B}^0$ pair in laboratory frame. The B^0_{tag} decays as a $\bar{B}^0$, so at this point, B^0_{CP} is a B^0. B^0_{CP} then oscillates, decaying as a $\bar{B}^0$. Δz is the distance between decays along the beamline axis, in laboratory frame.

1.5 $B^0 \to K_S\pi^0$ Decays

The $B^0 \to K^0\pi^0$ decays proceed primarily through the tree and penguin processes shown in Figure 1.5. The tree process is largely suppressed by the small V_{ub} due to the $b \to u$ transition. The penguin process is therefore the dominant mode in this decay (which proceeds via $\bar{u}$, $\bar{c}$ or $\bar{t}$ quark). The amplitudes being not of vastly different order - with the relative weak and strong phases between the two - allow the CP violating processes to occur.

The amplitude of the decay is therefore the sum of the tree component:

$$V^*_{ub}V_{us}A_t \tag{1.39}$$

and penguin component:

$$V^*_{ub}V_{us}A^u_p + V^*_{cb}V_{cs}A^c_p + V^*_{tb}V_{ts}A^t_p \tag{1.40}$$

where A_t is the tree amplitude and A^q_p are the amplitudes for the three penguins(for each of the up type quarks in the penguin loop). And since the orthogonality relations of the unitarity triangle give:

$$V^*_{ub}V_{us} + V^*_{cb}V_{cs} + V^*_{tb}V_{ts} = 0 \tag{1.41}$$

the full amplitude can be written as:

$$A\left(B^0 \to K^0\pi^0\right) = V^*_{ub}V_{us}(A_t + A^c_p - A^t_p) + V^*_{cb}V_{cs}(A^c_p - A^t_p) \tag{1.42}$$

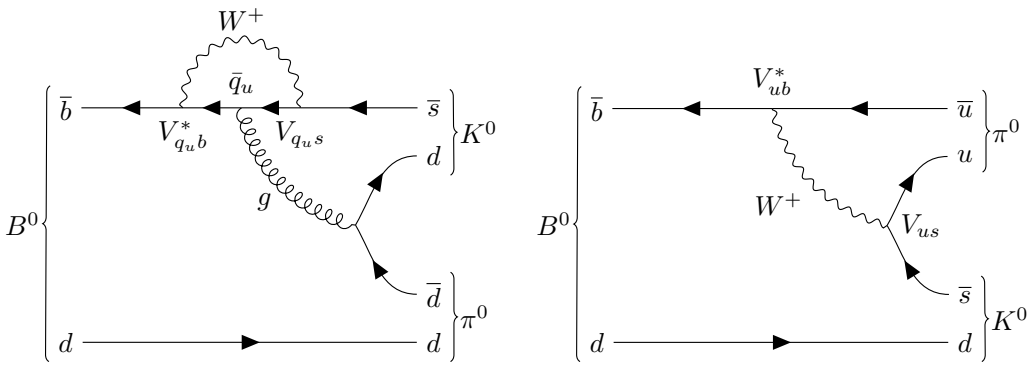

Figure 1.5: The $B^0 \to K^0\pi^0$ Feynman diagrams. The dominant penguin process on the left (where $\bar{q}_u$ is either a $\bar{u}$, $\bar{c}$ or $\bar{t}$). The colour suppressed tree diagram on the right.

As the $B^0 \to K^0\pi^0$ decays will lead to $K_S\pi^0$ half of the time, and K_S are a easier to reconstruct, studies of $B^0 \to K_S\pi^0$ will be investigated. Since B^0 and its CP conjugate $\bar{B}^0$ both decay to CP eigenstate $K_S\pi^0$, this channel sees both direct and mixing-induced CP violation - we will study direct CP violation.

The direct CP asymmetry in this channel is defined as:

$$\mathcal{A}_{CP}(K_S\pi^0) = \frac{\Gamma(\bar{B}^0 \to K_S\pi^0) - \Gamma(B^0 \to K_S\pi^0)}{\Gamma(\bar{B}^0 \to K_S\pi^0) + \Gamma(B^0 \to K_S\pi^0)} \tag{1.43}$$

Or equivalently:

$$\mathcal{A}_{CP}(K_S\pi^0) = \frac{N(\bar{B}^0 \to K_S\pi^0) - N(B^0 \to K_S\pi^0)}{N(\bar{B}^0 \to K_S\pi^0) + N(B^0 \to K_S\pi^0)} \tag{1.44}$$

Where N is the measured number of events of a given decay.

1.5.1 Motivations

A constraint on the CP asymmetries and branching ratios ($\mathcal{B}$) of $B \to K\pi$ decays has been placed by the isospin sum rule. Assuming flavour SU(3) and isospin symmetries, we get the relation[15]:

$$\mathcal{A}_{CP}(K^+\pi^-) + \mathcal{A}_{CP}(K^0\pi^+)\frac{\mathcal{B}(K^0\pi^+)}{\mathcal{B}(K^+\pi^-)}\frac{\tau_0}{\tau_+} = \mathcal{A}_{CP}(K^+\pi^0)\frac{2\mathcal{B}(K^+\pi^0)}{\mathcal{B}(K^+\pi^-)}\frac{\tau_0}{\tau_+} + \mathcal{A}_{CP}(K^0\pi^0)\frac{2\mathcal{B}(K^0\pi^0)}{\mathcal{B}(K^+\pi^-)} \tag{1.45}$$

where τ_0 and τ_+ are the lifetimes of the B^0 and B^+ respectively. A violation of this relation would point to new physics. With the latest experimental data this relation predicts $\mathcal{A}_{CP}(K^0\pi^0) = -0.15 \pm 0.036$, so a precise measurement of $\mathcal{A}_{CP}(K^0\pi^0)$ could point to physics beyond the standard model.

Additionally, there has been a pronounced ($\sim 5\sigma$) difference between the measured values of $\mathcal{A}_{CP}(K^+\pi^0)$ and $\mathcal{A}_{CP}(K^+\pi^-)$ - known as the $K\pi$ puzzle. This can be explained with a modified electroweak penguin amplitude[16]. Improved $\mathcal{A}_{CP}(K^0\pi^0)$ measurements will help us to probe for new physics in this channel.

The latest $\mathcal{A}_{CP}$ measurement from Belle was a time-dependent study measuring direct and mixing induced *CP*-violation terms, using both $K_S\pi^0$ and $K_L\pi^0$ decays and using an incomplete Belle dataset of $656 \times 10^6 B\bar{B}$ events[17]. The results from Belle and BaBar are laid out in Table 1.1. As can be seen the $\mathcal{A}_{CP}$ measurements have opposite signs, and large statistical uncertainties ($\sim 100\%$) so clearly the uncertainty on this measurement needs to be reduced to find any physics beyond the standard model in this channel.

	$\mathcal{A}_{CP}(K^0\pi^0)$	$\mathcal{B}(B^0 \to K^0\pi^0) \times 10^{-6}$
Belle	$0.14 \pm 0.13(stat) \pm 0.06(sys)$[17]	$9.68 \pm 0.46(stat) \pm 0.5$[18]
BaBar	$-0.13 \pm 0.13(stat) \pm 0.03(sys)$[19]	$10.1 \pm 0.6(stat) \pm 0.4(sys)$[20]

Table 1.1: The experimental results from Belle and BaBar.

2|Belle and KEKB

The Belle experiment was first proposed in the early 1990s and began taking data in 1999. The end of Belle's run came in 2010, after having produced $(771.581 \pm 10.566) \times 10^6 B\bar{B}$ pairs, corresponding to an integrated luminosity of $711 fb^{-1}$ at the $\Upsilon(4s)$ resonance[13]. The Belle integrated luminosity per year is shown in Figure 2.1

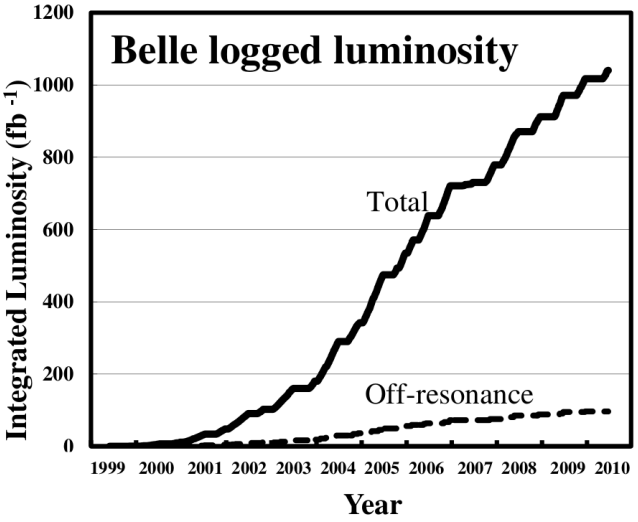

Figure 2.1: Showing the integrated total and off-resonance luminosities over Belle's run.

The electron-positron collisions for Belle are provided by the KEKB collider at the High Energy Accelerator Research Organisation(KEK) in Tsukuba, Japan. KEKB consists of two concentric rings of e^+ and e^- of asymmetric energy, producing boosted $\Upsilon(4s)$ in large numbers. As well as runs at the $\Upsilon(4s)$ resonance, there were shorter runs at the $1s$,$2s$,$3s$ and $5s$ resonances, as well as off-resonance energy runs (to investigate backgrounds).

We will now consider runs at the $\Upsilon(4s)$ resonance. The electrons and positrons are injected from a linac into the ($\sim$ 3km) rings; the High Energy Ring (HER) is of electrons at 8 GeV and the Low Energy Ring (LER) is of positrons at 3.5 GeV. The beams meet at the Interaction Point(IP) with a small crossing angle of 22 mrad giving a centre of mass energy of 10.58 GeV. The KEK accelerator is shown in Figure 2.2. This asymmetry is needed because the $\Upsilon(4s)$ must be boosted to increase

the lifetime of the $B\bar{B}$ in lab frame (to increase the distance between B decay points), and to ensure that the time between B decays can be resolved (vital for CP measurements). In COM frame the $B\bar{B}$ are produced almost at rest (and have a lifetime of order 10^{-12} s), so the boost of $\beta\gamma = 0.425$ means the decay vertices have an average separation of 200 µm, which is resolvable[21].

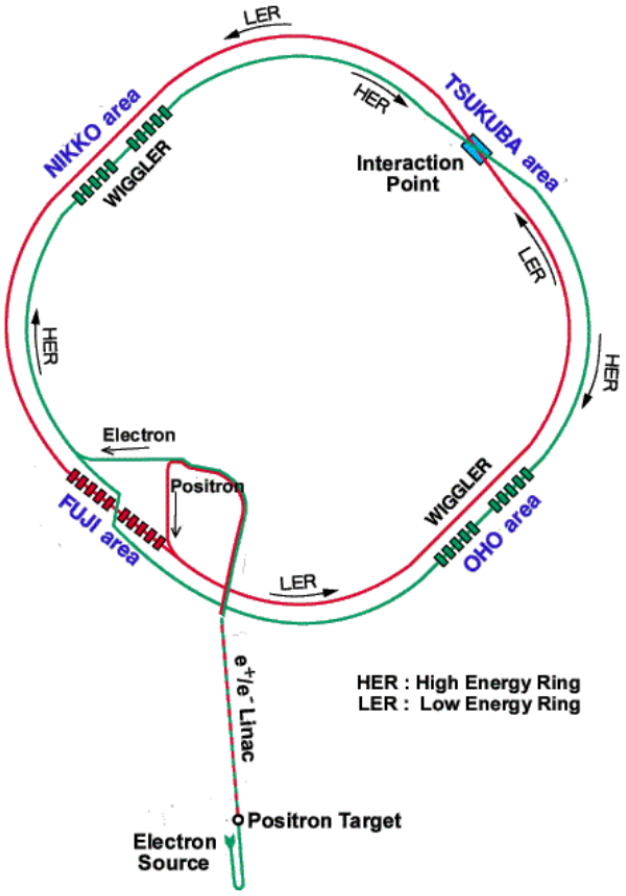

Figure 2.2: The KEKB accelerator showing the high energy electron beam (green), and the low energy positron beam (red) which meet at the IP point - Belle in the Tsukuba area.

The Belle detector consists of various layered detectors for vertexing and particle identification(PID) of the different decay products, shown in Figure 2.3. It covers the range $17° < \theta < 150°$ where θ is the angle from the HER beam axis. The components of the Belle detector are introduced below, see [21], [13], and [22] for more details, images sourced from these unless otherwise stated. Throughout the run, multiple data collection runs (called 'experiments', in the range 7-65 for $\Upsilon(4s)$) were conducted, corresponding to the changing conditions of the beam and detector.

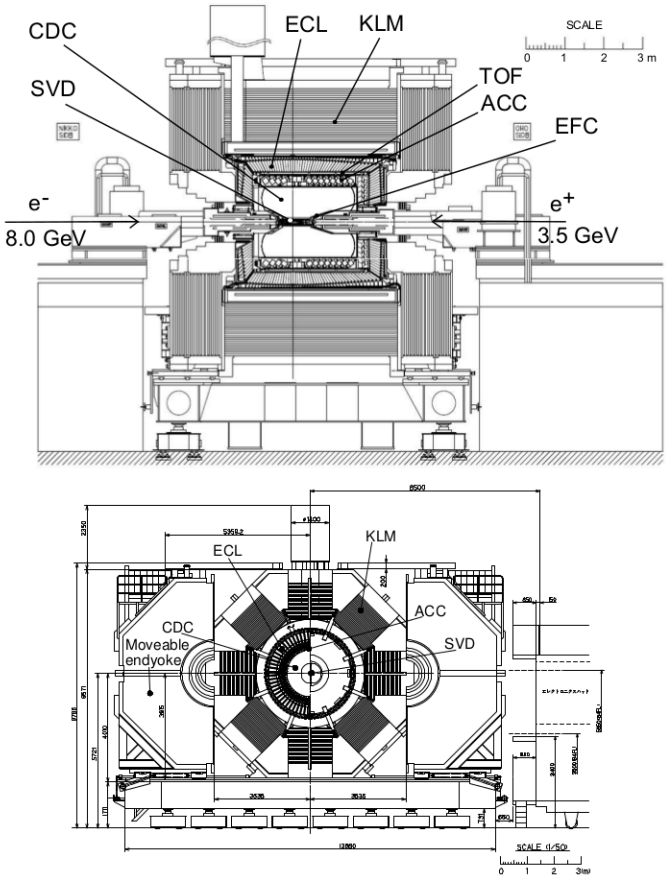

Figure 2.3: Showing the Belle detector and components with the longitudinal (top) and transverse (bottom) cross-sections.

2.1 Beam Pipe

The beam pipe consists of two beryllium cylinders of 0.5 mm thickness, see Figure 2.4. The radii were of 20 mm and 23mm for the inner and outer cylinder respectively. Helium is circulated in the 2.5 mm gap to provide cooling to the beam pipe. The outer layer is coated with a 20 μm gold layer to reduce the low energy X-ray background. After the upgrade to SVD2 (see below) the inner radius was changed to 15 mm.

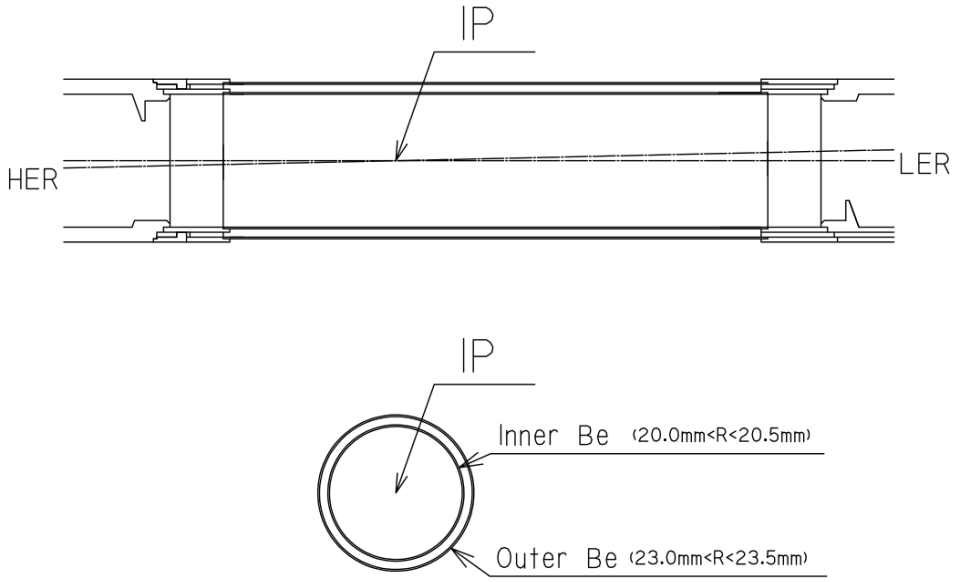

Figure 2.4: Showing the longitudinal (top) and transverse (bottom) cross-sections of the original beam pipe before the SVD2 upgrade.

2.2 Silicon Vertex Detector

The Silicon Vertex Detector (SVD) is the innermost detector located just outside of the beam pipe. It is used to detect the trajectory of charged decay products for precise vertexing of the decay positions along the beam axis. It was updated from the old version (SVD1) after 4 years (collecting 15% of total Belle data) to the new vertex detector (SVD2). See Figure 2.5.

SVD1 consists of three double-sided silicon-strip detectors(DSSD) of radii 30 mm, 40.5 mm and 60.5 mm. SVD1 covered $23° < \theta < 139°$ providing 83% solid angle coverage. The upgraded SVD2 has four DSSD layers (at radii 20 mm, 43.5 mm, 70 mm and 88 mm) covering the full range of angular acceptance provided by the rest of the detector.

The readout chip was also upgraded from the VA1 to the VA1TA providing improved temporal resolution and radiation hardness.

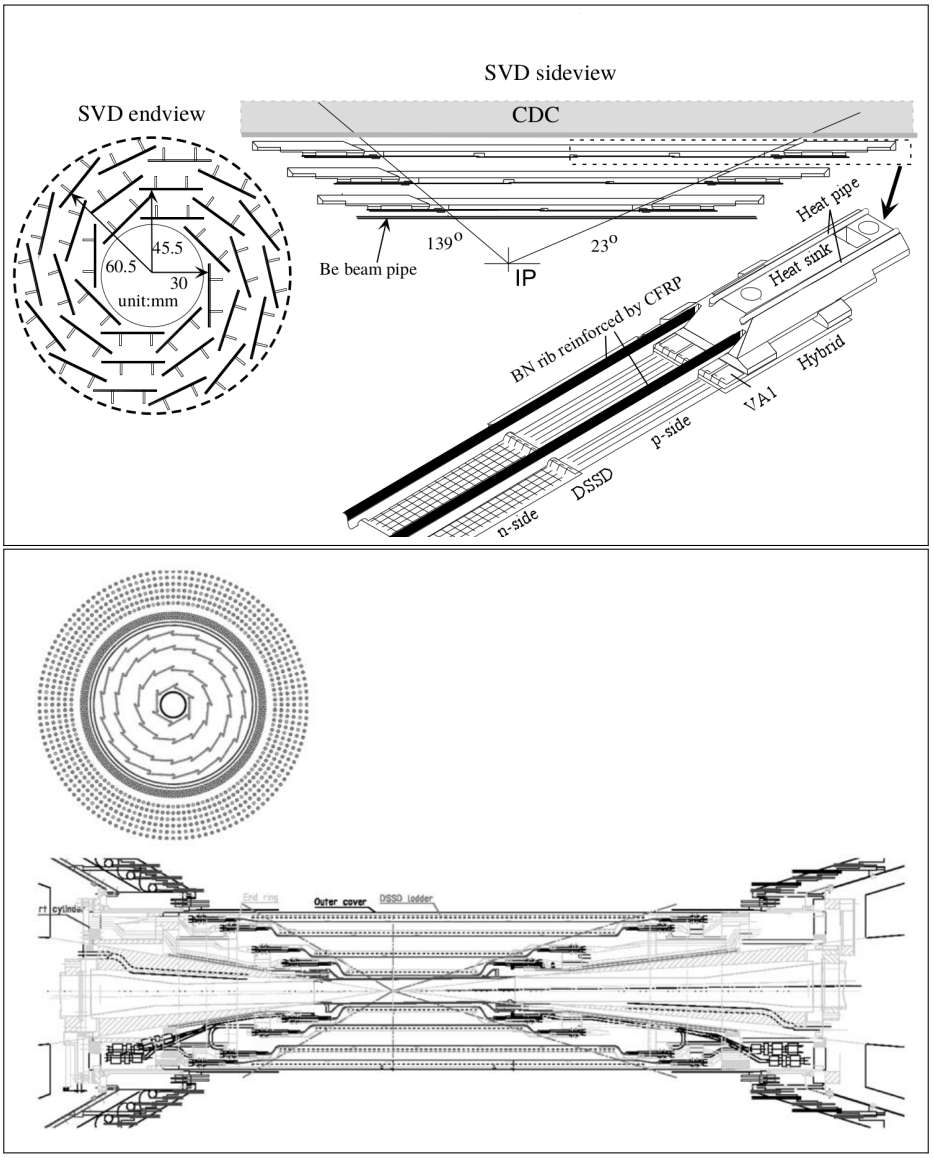

Figure 2.5: Showing SVD1 (top) and SVD2 (bottom) transverse and longitudinal cross-sections.

2.3 Extreme Forward Calorimeter

The Extreme Forward Calorimeter (EFC) provides extended angular coverage that the ECL doesn't cover, needed for specific B decays, and to absorb radiation to protect the CDC. It is also used for beam and luminosity monitoring. It covers the angular ranges $6.4° < \theta < 11.5°$ and $163.3° < \theta < 171.2°$ in the forward and negative directions (relative to the electron beam) respectively. It consists of eighty bismuth germanate crystals (chosen for radiation hardness, energy resolution, and cost), in both the forward and backwards directions (at 600 mm and 435 mm in the forward and background directions respectively) surrounding the beam pipe (with an inner radius of 65 mm). See Figure 2.6.

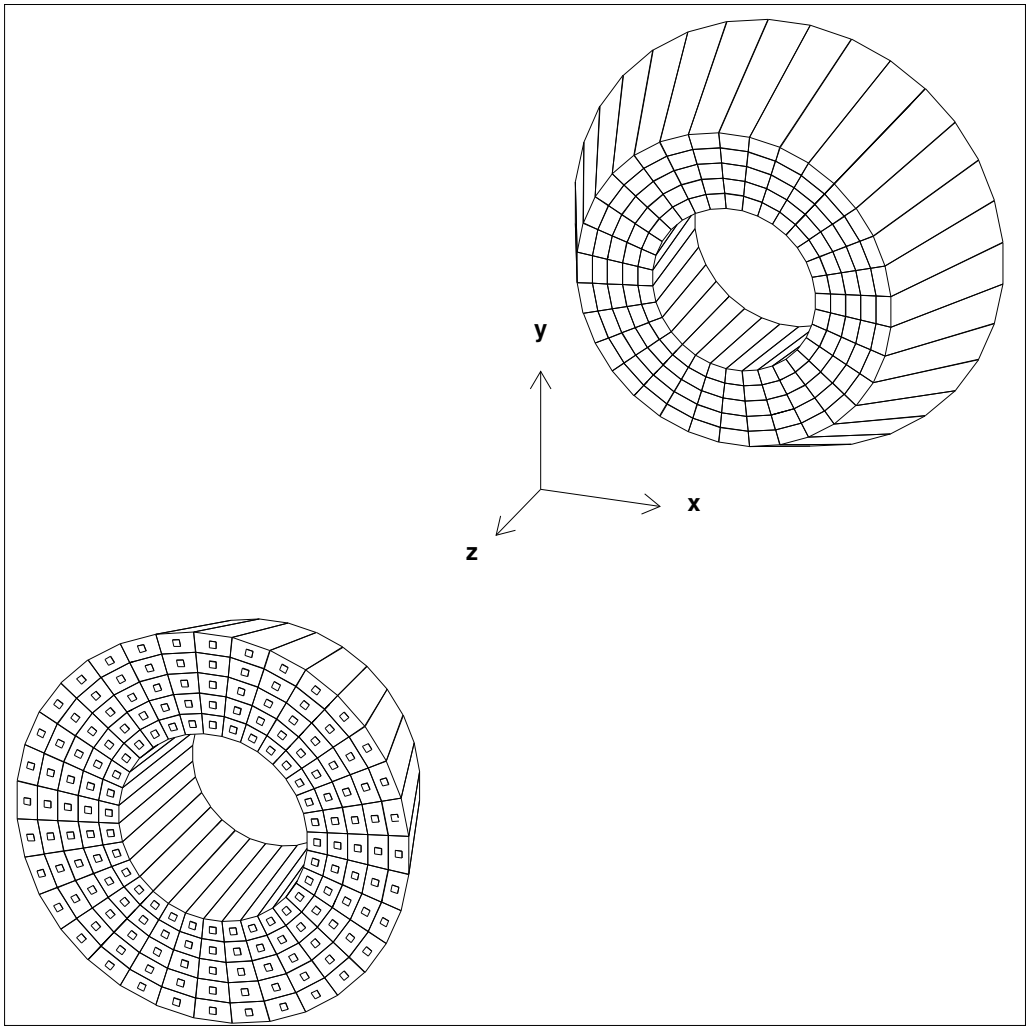

Figure 2.6: Showing the EFC components - forward (bottom-left) and backwards (top-right).

2.4 Central Drift Chamber

The Central Drift Chamber (CDC) is used for the tracking of charged particles and for measuring their momenta, as well as providing vital information for triggering. It allows the measurement of energy loss over the distance travelled (dE/dx), providing PID information, it is particularly effective for low momentum particles. The plot of dE/dx against momentum, shown in Figure 2.7, shows the different distributions providing distinguishing information between charged pions, kaons, protons and electrons.

To reduce the backgrounds from synchrotron radiation and to reduce multiple Coulomb scattering, the gas nuclei must have low proton number. To meet this condition, an equal mixture of ethane(to keep a good resolution on dE/dx) and helium is used. The chamber is composed of 50 cylindrical layers, and a total of 8400 nearly-square drift cells. Readout from each layer is from between three and six stereo strip layers, and three cathode strip layers. It is asymmetric in the direction

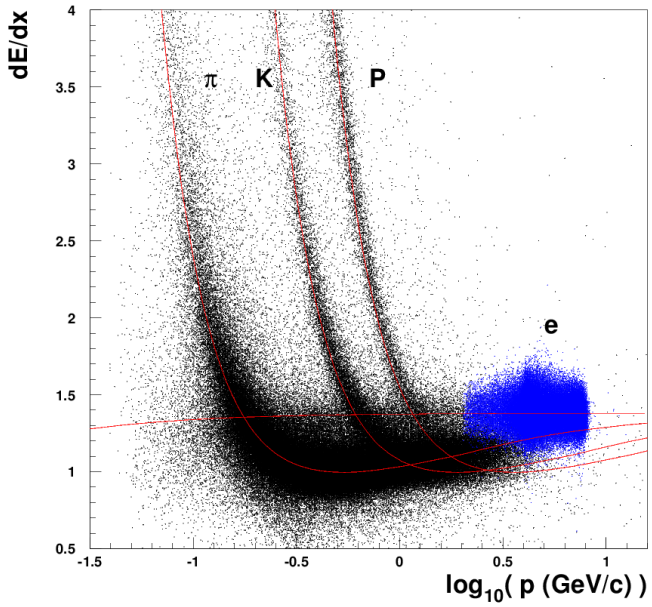

Figure 2.7: Showing how energy loss and momentum in the CDC can be used to distinguish charged particles.

of the electron beam to provide full angular coverage. See Figure 2.8.

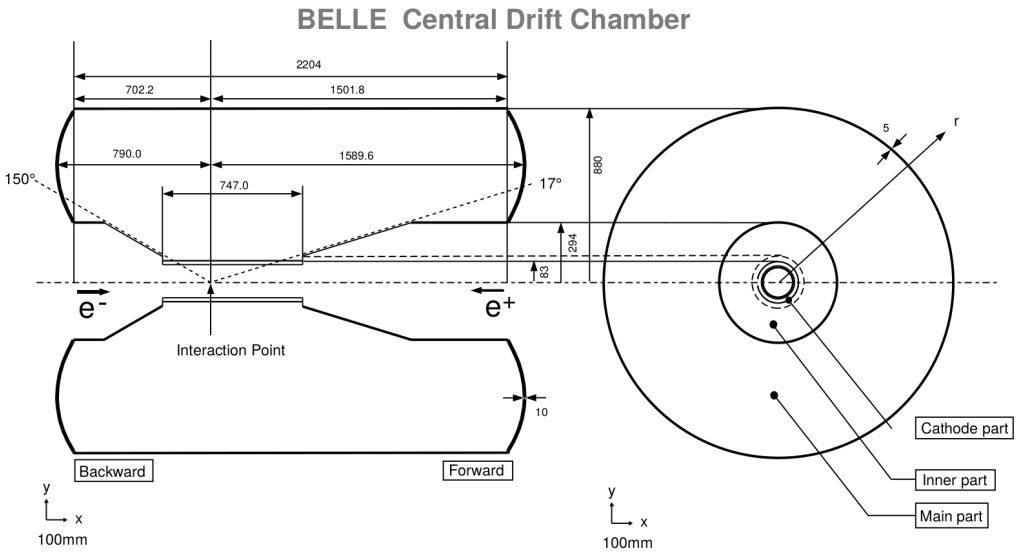

Figure 2.8: Showing the CDC longitudinal (left) and transverse (right) cross-sections.

2.5 Aerogel Cerenkov Counter

The Aerogel Cerenkov Counter (ACC), shown in Figure 2.9, is used for PID (in particular to distinguish charged pions from kaons at high momentum). It consists of the end-cap covering $13.6° < \theta < 33.4°$ and the barrel covering $33.3° < \theta < 127.9°$. The Cerenkov radiation, produced by high velocity charged particles, is detected by photo-multiplier tubes (PMTs). There are 1188 aerogel modules of varying refractive index (in the range of 1.01 to 1.03), varying PMT size, and either one or two PMTs. The set up is used to distinguish $\pi^{\pm}$ and $K^{\pm}$ with high momenta, in the range 1.2 GeVc^{-1} to 4 GeVc^{-1}. Cerenkov radiation is produced when the particle velocity is greater than the phase velocity of light (in the aerogel), so in order for a particle to cause Cerenkov radiation, the following condition must be met:

$$v_p > \frac{c}{n} \tag{2.1}$$

Where v_p is the particle velocity, c is the speed of light (in a vacuum), and n is the refractive index of the aerogel. It acts as a threshold Cerenkov radiation detector as for a given momentum (given the right choice of n), the lighter pions will have a higher velocity than the kaons, and so the pions will cause radiation whereas the kaons will not.

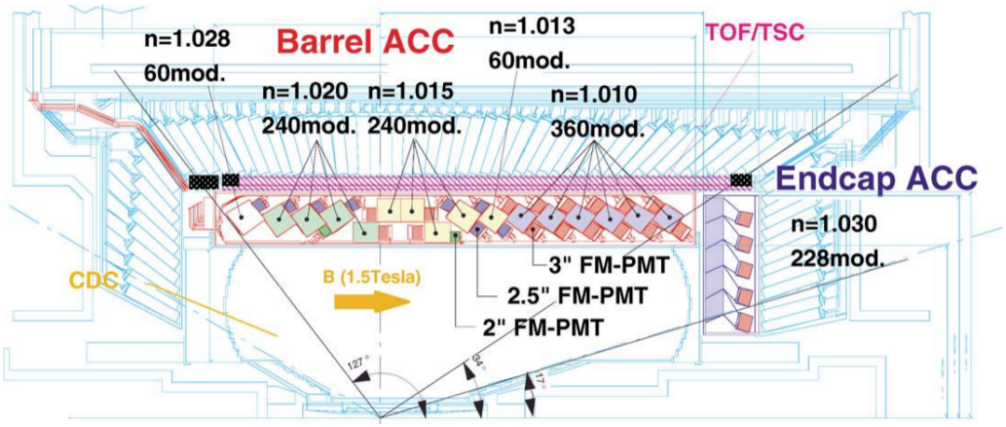

Figure 2.9: Showing a schematic of the ACC.

2.6 Time of Flight Counters

The Time of Flight (TOF) system is used for PID which can be used for efficient flavour tagging, as well as providing information for the trigger. It efficiently distinguishes $\pi^{\pm}$ and $K^{\pm}$ of momenta below 1.2 GeVc^{-1} by measuring energy deposition and detection times across the detector. The TOF consists of 64 modules, shown in Figure 2.10, each with two plastic scintillator counters and one trigger scintillation counter (TSC). It covers the regions $34° < \theta < 120°$ and has a minimum radius of

120 cm. It has a time resolution of 10^{-10} s which allows for the measuring of the charged particle velocity and so is used to distinguish between protons, kaons an pions at lower momenta (less than 1.2 GeVc^{-1}).

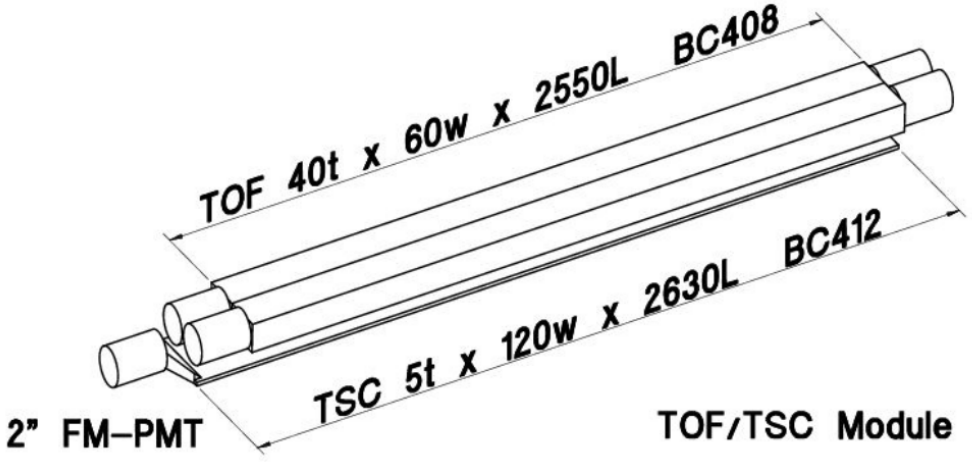

Figure 2.10: Showing a schematic of a Time of Flight module.

2.7 Electromagnetic Calorimeter

The Electromagnetic Calorimeter (ECL) is primarily used to efficiently detect photons produced as part of the B decay chain with a high resolution in position and energy. The photons produce cascading electromagnetic showers within the ECL, and the energy deposited is used to measure the photon energy. Electron identification also relies on the energy deposited in the ECL. It must meet the requirements that it performs well at lower energies of less than $\sim$ 100 MeV for photons at the end of long decay chains. A good resolution at high energies and a fine grained segmentation are required for decays where the photons are produced at high energy and with a small angle between them, for example $\pi^0 \to \gamma\gamma$ decays.

The ECL consists of three sections; the 3 m long barrel (with a radius of 1.25 m), and the forward and backward end-caps (+2 m and -1 m along the electron beamline respectively). The 8736 crystals point nearly to the IP point, angled slightly to prevent gaps that would allow photons through, see Figure 2.11. The material chosen is CsI(Tl) to meet these requirements.

Although performing very well, the ECL suffers from shower leakage from high energy photons. Each crystal is 30 cm in length, corresponding to 16.2 ($e^{\pm}$) interaction lengths (0.8 K_L interaction lengths), which is normally enough for the electromagnetic showers to end within the crystal. Despite this, in the case of high energy photons, the electromagnetic shower reaches the end of the crystal and not all of the energy is deposited in the ECL. In the cases of shower leakage the photon energy is wrongly measured as lower than it is.

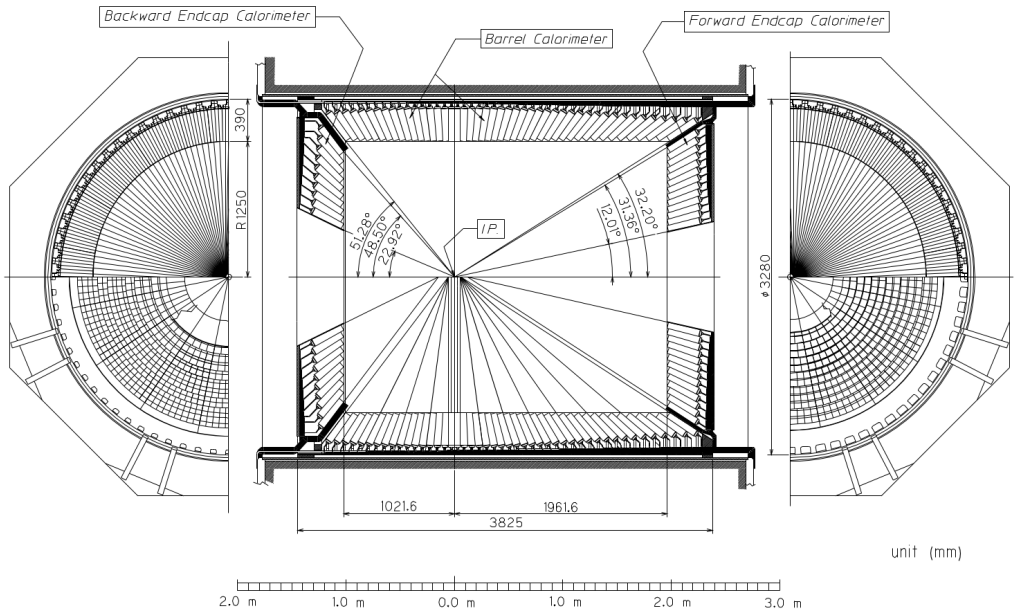

Figure 2.11: Showing a schematic of the ECL (centre) and transverse cross-section at each end (left, right).

2.8 K_L Muon Detection System

The K_L Muon Detection System (KLM) is used to detect the longer lived particles K_L and μ, hence the name. It detects those with momenta above 600 MeVc^{-1} very efficiently. It consists of the barrel section and end caps, covering the range $20° < \theta < 155°$. The KLM has fourteen iron plates inter-spaced between fifteen(fourteen for the end caps) glass-electrode-resistive plate counters(RPC) for detection. The iron adds 3.9 interaction lengths to the 0.8 interactions lengths from the ECL. Each RPC has a 2 mm gap between two electrodes that is filled with a gas that can be ionised by muons or K_L decay products (due to interaction with the iron plates), producing a discharge at the electrodes. The ionisation from the muons or K_L decay clusters can then produce a position and time readout. K_L identification uses this information along with information from the CDC. Because muons are not involved in strong interactions, they are not absorbed in the ECL or KLM (unlike $\pi^{\pm}$ and $K^{\pm}$) and only lose energy via ionization, and so can be distinguished from charged pions and kaons.

3|Data Simulation

Physics analyses at Belle do not initially have access to real data, as optimising an analysis to real data would allow the model to be adjusted to get the desired result, introducing a bias. To account for this we must perform a 'blind analysis', where the analysis is performed on Monte Carlo(MC) simulated data. Models are then validated on a 'control sample' (a different decay channel that produces physically similar results to the decay channel being investigated) to validate that any differences between MC and real data do not invalidate the model. All data used in this thesis are Monte Carlo (produced at the KEK computing cluster using the Belle Analysis Software Framework - BASF[23]) unless otherwise stated.

3.1 $B^0 \to K_S\pi^0$ Data

Signal data is generated in two steps; first generate the decay, and then simulate the propagation and interactions of these decay products in the detector. The output of this is then saved in mini Data Summary Tapes(mDST) of same form as real data (BASF populates decay tables with particle candidates and detector information).

The first step is achieved using the EvtGen package[24]. The expected yields (event numbers) are scaled by the integrated luminosity of each Belle experiment. The decay channels, branching fractions and physical models are specified. For $B^0 \to K_S\pi^0$ MC generation, alias $B^0/\bar{B}^0$ mesons are defined that decay exclusively to $K_S\pi^0$. The decay chain of the $\Upsilon(4s)$ is then defined as $\Upsilon(4s) \to (B^0_{CP}/\bar{B}^0_{CP})(B^0_{tag}/\bar{B}^0_{tag})$, in any of the four combinations at different branching fractions (depending on the specified $\mathcal{A}_{CP}$), where the $B^0_{CP}/\bar{B}^0_{CP}$ are the alias mesons and $B^0_{tag}/\bar{B}^0_{tag}$ decay generically. The EvtGen decay models used are VSS_MIX for the vector-$\Upsilon(4s)$ to the two scalar-B mesons - including the effects of $B^0 - \bar{B}^0$ mixing, and SSS_CP for the B decay to scalar-K_S and scalar-π^0. These models include CP violating effects.

The output of EvtGen is then passed to GSIM(Geant SIMulation), a package built on GEANT3[25], to simulate the propagation through the Belle detector. For this reason MC data generated this way is referred to as GSIM data. The EvtGen output is divided by Belle experiment so (aside from correct yield fractions) GSIM can take account of the effects of the changing Belle detector.

Using this procedure three $\mathcal{A}_{CP} = 0$ datasets of one-million events each were generated, each having $\Upsilon(4s) \to B^0_{CP}\bar{B}^0_{tag}$ and $\Upsilon(4s) \to \bar{B}^0_{CP}B^0_{tag}$ in equal quantities. Two of these datasets are used for training and optimising the neural networks, with the third being used for the physics analysis.

The $\mathcal{A}_{CP}$ is generated at the analysis stage from an $\mathcal{A}_{CP} = 0$ dataset. To validate

the $\mathcal{A}_{CP}$ generation method, a fourth signal dataset (also one-million events) for $\mathcal{A}_{CP} = +1$ is generated. To generate the $\mathcal{A}_{CP} = +1$ in EvtGen, the $\Upsilon(4s)$ decay would be entirely to $\bar{B}^0_{CP}B^0_{tag}$ (if there were no neutral B-meson oscillations), but to simulate the effects of $B^0 - \bar{B}^0$ mixing, the defined branching fraction is reduced to 0.814 for $\Upsilon(4s) \to \bar{B}^0_{CP}B^0_{tag}$, and the branching fraction of 0.186 is defined for $\Upsilon(4s) \to \bar{B}^0_{CP}\bar{B}^0_{tag}$ (when B^0_{tag} has oscillated to $\bar{B}^0_{tag}$).

3.2 Background Data

To properly take account of the backgrounds at the analysis stage, each background type must be treated individually. The main backgrounds in this study are backgrounds where the $\Upsilon(4s)$ isn't produced ($e^+e^- \to q\bar{q}$) and backgrounds from other B decay modes. Background events from from generic B decays ($b \to c$ transitions) are expected to be of order 1 so are not included in this study.

3.2.1 Rare Backgrounds

Rare backgrounds are non-negligible in this study. They are B decays that don't include the $b \to c$ transition, therefore have much smaller branching fractions. Rare-charmless decays have branching fractions of the order 10^{-5}. These $b \to u, d, s$ decays are split into two categories; charged and mixed, corresponding to decays from B^+/B^- and $B^0/\bar{B}^0$ respectively, and will be referred to as charged rare and mixed rare backgrounds from here on.

The pre-existing mixed and charged rare MC datasets are available to the Belle collaboration - 50 streams of each(where one stream has the expected number of events over the entire Belle run). $B^0 \to K_S\pi^0$ is a rare decay so these must be removed from the mixed-rare sample.

3.2.2 Continuum

Continuum is by far the biggest background in this study. It is produced at the IP point in electron-positron collisions where instead of creating an $\Upsilon(4s)$, a quark pair($u\bar{u}$, $d\bar{d}$, $c\bar{c}$ or $s\bar{s}$) is produced, which proceeds to hadronise instantly producing a jet-like decay. As this process doesn't need to be at the $\Upsilon(4s)$ energy, off-resonance and energy scans can be used to investigate continuum.

Real off-resonance (at a COM energy of 10.52 GeV) data is available at 10.35% of the expected continuum yield at the $\Upsilon(4s)$ resonance. Six streams of continuum data are available; three are used for training and optimising the neural networks, and three are used for physics analysis.

4|Event Reconstruction

The signal($B^0 \to K_S\pi^0$) and background data are reconstructed from the GSIM mDST output files, containing the necessary data, which includes the particle candidates, detector data, and in the case of MC data, *gen_hepevt* (a table of the known simulated decay chain). *gen_hepevt* provides vital information for checking that the particles have been correctly reconstructed, and is used to remove $B^0 \to K_S\pi^0$ from the mixed rare dataset.

4.1 B_{CP} Reconstruction and Selection

4.1.1 π^0 Selection

The neutral pion decays around 98.8% of the time to a pair of photons. Pre-reconstructed pion candidates have the condition that the recorded energy deposited in the ECL by the daughter photons is greater than 50 MeV each. Reconstructed pions ($m_\pi = 134.98$ MeVc^{-2}) with a mass outside of the range 100 MeVc^{-2} to 162 GeVc^{-2} are then discarded. The quality of the pion reconstruction must be good so only those with a χ^2 less than 50 were selected. To increase the signal yield, π^0 were reconstructed from a combination of photons that decayed early to e^+e^- (that were detected in the SVD) - with others - and with pre-detected photons. These pions also have the condition that their mass is in the range 100 MeVc^{-2} to 162 MeVc^{-2}.

4.1.2 K_S Selection

K_S decays 69.2% of the time to $\pi^-\pi^+$, and since vertexing information is important, only these decays are used (the only other non-negligible decay mode is to $\pi^0\pi^0$, which decaying to $\gamma\gamma$, can't be used for vertexing). The cuts applied to the reconstructed K_S are shown in table 4.1, see [26] for more details.

4.1.3 ΔE, M_{bc}, and B Selection

The process of reconstructing B candidates from (possibly wrongly) reconstructed K_S and π^0 can cause B-mesons in real B events to be reconstructed from particles that aren't in their decay chain, meaning that their reconstructed momentum and energy will be incorrect. Additionally, pure background events (where there was no $B^0 \to K_S\pi^0$ decay) will produce wrongly reconstructed B candidates with a much

$p/\mathrm{MeV}c^{-1}$	dr/cm	$d\phi/rad$	z_{dist}/cm	fl/cm
<0.5	>0.05	<0.3	<0.8	N/A
0.5-1.5	>0.03	<0.1	<1.8	>0.08
>1.5	>0.02	<0.03	<2.4	>0.22

Table 4.1: Showing the K_S cuts where: p is the K_S momentum. dr is the closest K_S distance to the IP point (in the plane perpendicular to the beamline). $d\phi$ is the azimuthal angle between p and the K_S decay vertex. z_{dist} is the separation(in the beamline axis) between the $\pi^+\pi^-$ at the K_S decay vertex. fl is the K_S flight length in the plane perpendicular to the beam axis.

wider range of momenta and energies (in COM frame) than real B candidates. To address this, two uncorrelated parameters are used; ΔE and M_{bc}.

ΔE is the difference between the reconstructed B energy and half the $\Upsilon(4s)$ energy:

$$\Delta E = E_B - E_{beam} \tag{4.1}$$

Where E_B is the reconstructed B meson energy (in COM frame) and E_{beam} is the beam energy (half the COM energy in the e^+e^- collision). A perfectly reconstructed B would have the same energy as E_{beam}, so signal candidates have a ΔE distribution peaked at zero. The ΔE value is dependent on correct particle identification as E_B depends on the mass of the daughter particles. In the case of $B^0 \to K_S\pi^0$ the distribution is asymmetric due to incorrect E_B(from incorrect π^0 reconstruction due to ECL energy leakage, see 4.1.4). The signal and continuum ΔE distributions are shown in Figure 4.1.

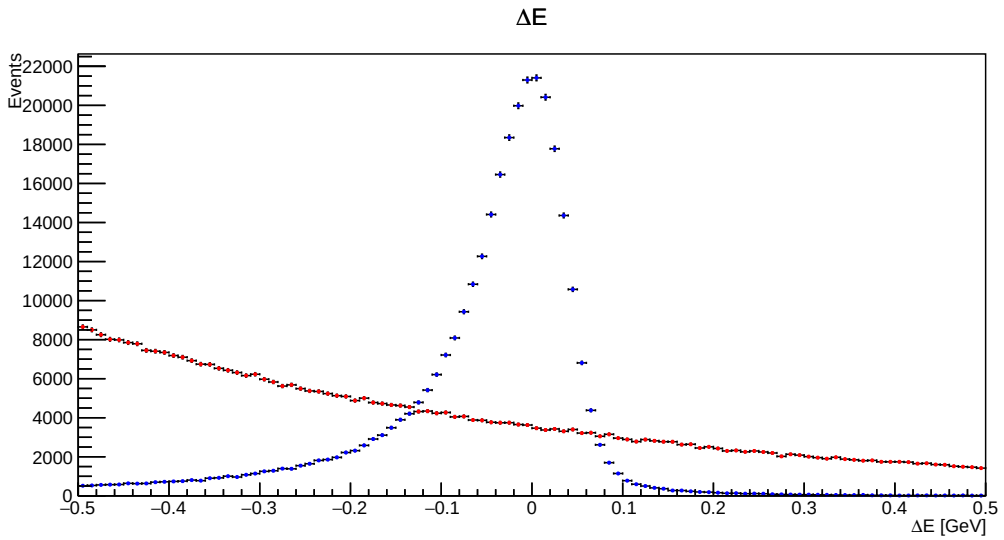

Figure 4.1: Showing ΔE MC distributions for scaled signal(blue) and continuum(red).

M_{bc} is the signal beam constrained mass, the B mass, given $\vec{p}_{B^0_{recon}}$ (its reconstructed momentum in COM frame) and its energy, where the energy is the *true* energy(E_{beam}) instead of its reconstructed energy:

$$M_{bc} = \sqrt{E^2_{beam} - |\vec{p}_{B^0_{recon}}|^2} \tag{4.2}$$

It is complementary to ΔE as is doesn't depend on the daughter particles being correctly identified(just correct momentum reconstruction). The signal-M_{bc} distribution peaks around the B^0 mass, with a small low mass tail(from underestimating the π^0 momentum, thus over estimating the B momentum in the COM frame). The continuum M_{bc} distribution is generally a smoothly falling distribution up to E_{beam}. The M_{bc} distributions for signal and continuum are shown in Figure 4.2.

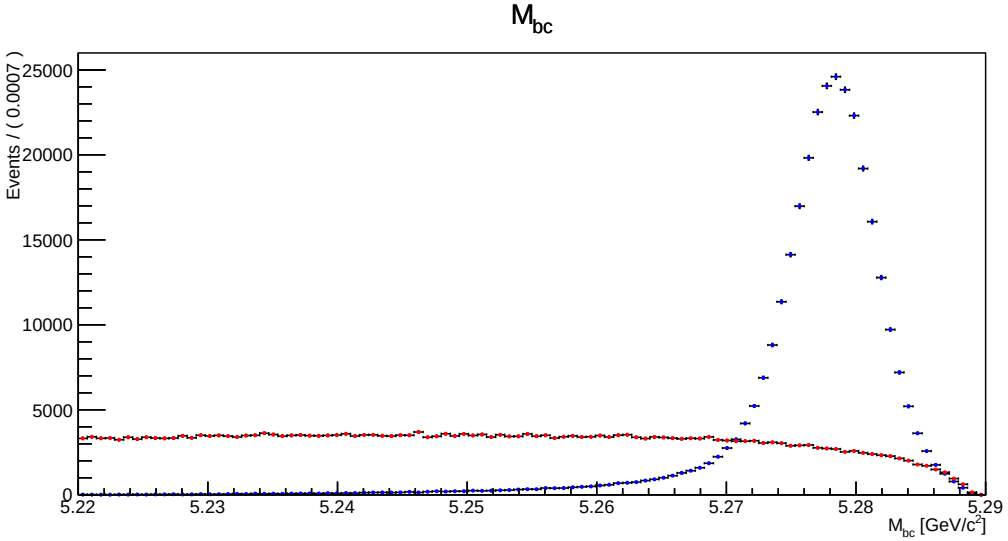

Figure 4.2: Showing M_{bc} MC distributions for scaled signal(blue) and continuum(red).

Using these two parameters, we require that reconstructed B candidates have the condition that ΔE is in the range ± 0.5 GeV and M_{bc} is between 5.2 GeVc^{-2} and 5.29 GeVc^{-2}. We only expect one signal decay per event. If more than one B candidate is present, the one with the lowest χ^2_{sum} is chosen, where χ^2_{sum} is the sum of χ^2 from the K_S and π^0 reconstructions.

This reconstruction and selection procedure leaves us with $401,686 \pm 634$ continuum events in one stream (this number is the number of events left in the sixth continuum stream, its uncertainty is its square-root as we assume a Poisson distribution), 932 ± 4 charged-rare events, and 383 ± 3 mixed-rare events(the number of rare events is one-fiftieth of the MC data for each stream, the uncertainty is one-fiftieth of the square root of the full dataset size). Of one-million signal MC events, 315,434 events remain, so the efficiency of reconstruction (ϵ_{recon}) is $(31.54 \pm 0.06)\%$. From this we get the expected number of signal events:

$$N_{signal} = N_{B\bar{B}} \times R_{B^0\bar{B}^0} \times \mathcal{B}\left(B^0 \to K^0\pi^0\right) \times \epsilon_{recon} \tag{4.3}$$

Where $N_{B\bar{B}}$ (the total number of $B\bar{B}$ events) is $(771.581 \pm 10.566) \times 10^6$, $R_{B^0\bar{B}^0}$ (the fraction of $B\bar{B}$ that are $B^0\bar{B}^0$) is 0.486±0.006, $\mathcal{B}(B^0 \to K^0\pi^0)$ is $(9.9\pm0.5)\times10^{-6}$[5]. Note there is a factor of two (as there are two B mesons that could decay in the signal channel) and a factor of 0.5 (the fraction of K^0 that go to K_S) that are not shown. This gives an expected signal yield of 1171 ± 63.

We will have true signal events which are incorrectly reconstructed, for example combining the decay products of B^0_{CP} and B^0_{tag}. In this case some physical parameters may be different. We can retrieve the number of incorrectly reconstructed B^0_{CP} in the signal dataset using the information from the *gen_hepevt* table. These Self Cross Feed (SCF) events comprise $(11.50 \pm 0.06)\%$ of the total signal dataset.

As can be seen there is far more continuum than signal (around 343 times as much) and it is vital to reduce this background before any physics analysis can proceed.

4.1.4 π^0 Momentum Correction

The daughter photons from the high momentum π^0 create high energy showers inside the ECL crystals (from which their energy is measured). The electromagnetic showers can lose energy out of the sides of the crystal, and the back of the crystal in the case of high energy photons where the electromagnetic showers can reach the end of the crystal. This means that it is common for the energy deposited in the ECL to be lower than the true photon energy. This leads to the reconstructed pion momentum being underestimated. As a consequence the reconstructed B momentum will be overestimated. Instead of the reconstructed B momentum being the sum of K_S and π^0 reconstructed COM momenta ($\vec{p}_{K_S}$ and $\vec{p}_{\pi^0}$ respectively), the momentum is corrected to:

$$\vec{p}_{B^0_{corrected}} = \vec{p}_{K_S} + \frac{\vec{p}_{\pi^0}}{|\vec{p}_{\pi^0|}} \sqrt{(E_{beam} - E_{K_S})^2 - m^2_{\pi^0}} \tag{4.4}$$

Where E_{K_S} is the reconstructed K_S energy(COM frame) and m_{π^0} is the actual pion mass. This keeps the direction of $\vec{p}_{\pi^0}$ unchanged but sets it to the *true* pion momentum(as its true energy is the energy difference between the beam - the true B^0 energy - and K_S energies).

Having corrected the B momentum, we can define the corrected M_{bc} as:

$$M^{corr}_{bc} = \sqrt{E^2_{beam} - |\vec{p}_{B^0_{corrected}}|^2} \tag{4.5}$$

This has the effect of removing the low M_{bc} tail and sharpening the peak, see Figure 4.3. It does not impact ΔE as neither the pion nor B-meson energies are adjusted. Correcting M_{bc} has the effect of decorrelating it from ΔE as the wrongly reconstructed pions giving the low energy ΔE tail are the same pions giving the low mass M_{bc} tail. Since the M_{bc} is corrected, these two variables lose their correlation, see Figure 4.4

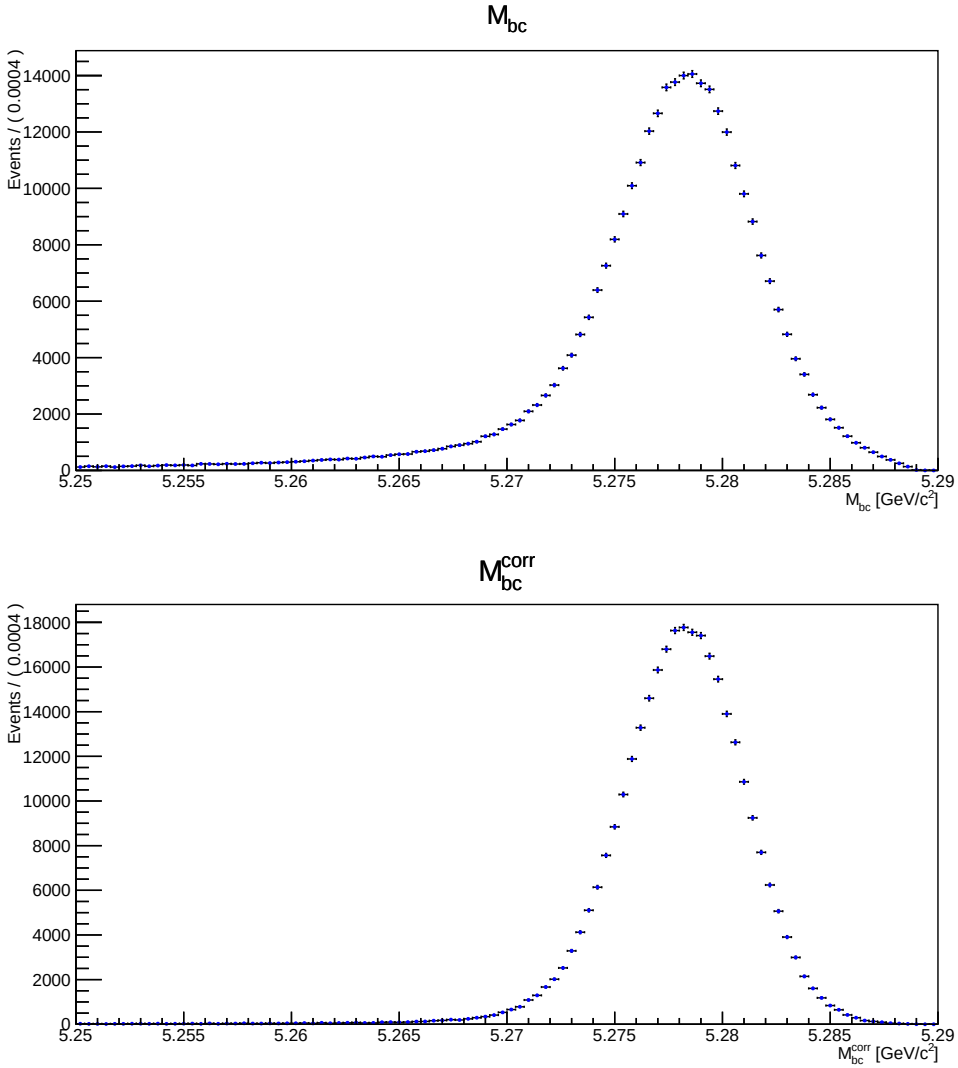

Figure 4.3: Showing the signal M_{bc}(top) and M_{BC}^{corr}(bottom) distributions, notice the removal of the low mass tail and the sharpening of the peak.

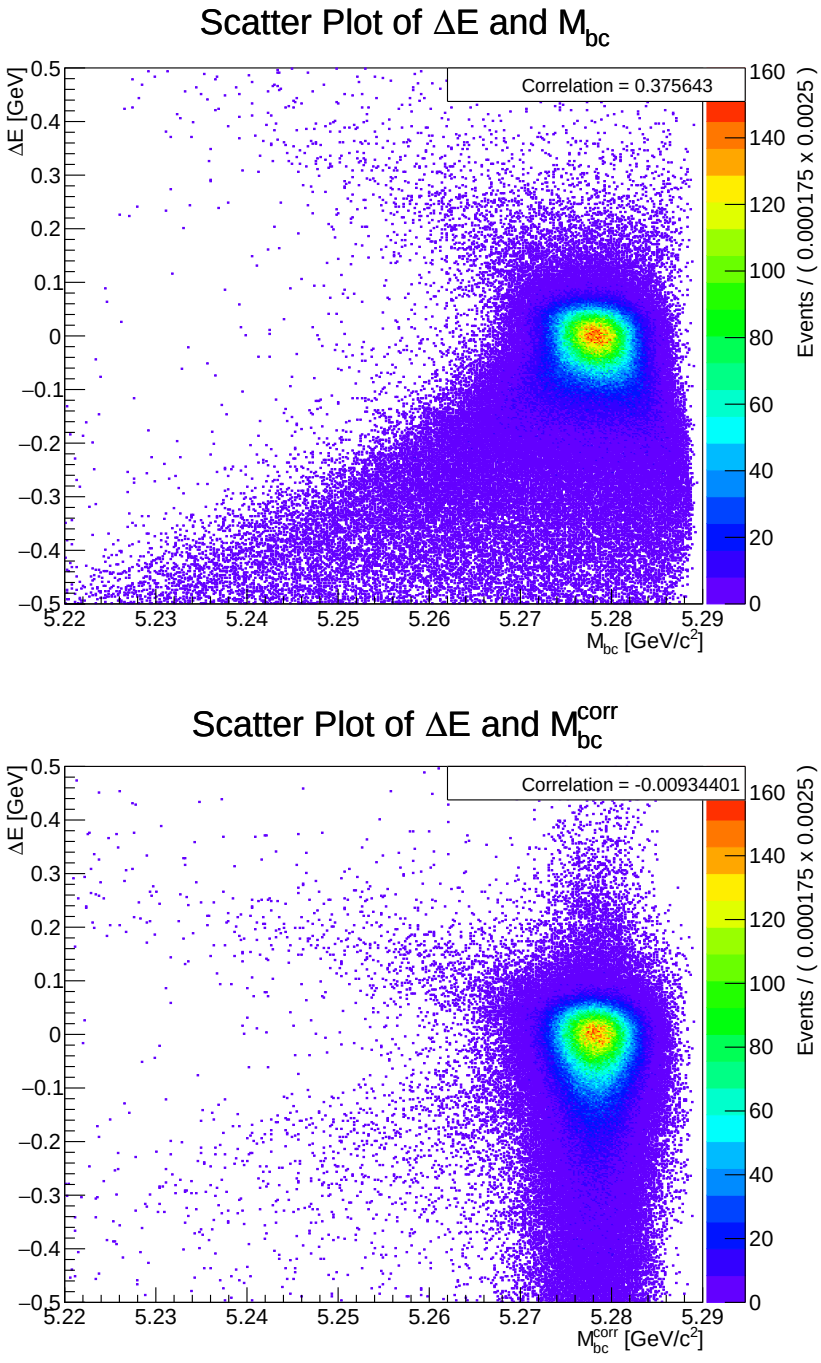

Figure 4.4: Showing the signal ΔE vs M_{bc}(top) and ΔE vs M_{BC}^{corr}(bottom). Notice the decorrelation effect of the π^0 momentum correction on M_{bc}.

4.2 Flavour Tagging

Studying direct *CP* violation requires that we know the flavour of the B^0 to a good level of certainty. We cannot discern the flavour of B_{CP} directly, but we can measure the flavour of B_{tag}, and taking account of $B^0 - \bar{B}^0$ mixing, get the flavour information of the signal B-meson.

The HAMLET[27] tagging package used in this analysis gives the flavour of B_{tag} using a categorical algorithm. This works by looking for common decays where the flavour information can be deduced from the decay products, for example the charge of the lepton in semi-leptonic decays or of the kaon in B^0 to charged kaon decays. The parameter returned is $q.r$. The value of q is the flavour of B_{tag}, $+1$ for B^0 and -1 for a $\bar{B}^0$. The certainty is given by r which ranges from no discernible flavour discrimination (zero) to a certain flavour tag (one). Thus the $q.r$ distribution has the range ± 1, with values at $q.r = +1(-1)$ definitely being a $B^0(\bar{B}^0)$, and events at $q.r = 0$ having a 50% of being either B^0 or $\bar{B}^0$. The $q.r$ distributions for continuum and $\mathcal{A}_{CP} = 0$ signal are shown in Figure 4.5.

We can therefore use $q.r$ to discern the $\mathcal{A}_{CP}$ as any preference of B^0 or $\bar{B}^0$ decays to $K_S\pi^0$ can be observed. As can be seen in Figure 4.6 the $\mathcal{A}_{CP}$ is clear from the $q.r$ distribution. An increased amount of B_{tag} at $q.r > 0$ means an increased amount of B^0_{CP} being $\bar{B}^0$, and as per the definition at 1.44 is a positive $\mathcal{A}_{CP}$. Note that even for the $\mathcal{A}_{CP} = +1$ sample, the number of entries at $q.r = -1$ is not zero due to $B^0 - \bar{B}^0$ mixing (see 1.4), where the B^0_{tag} and B^0_{CP} have the same flavour due to one of them oscillating before decay(the dilution from the mixing factor $\chi_d = 0.186 \pm 0.004$ means the number of entries at $q.r = -1$ is 18.6% of the sum of the entries at $q.r = \pm 1$).

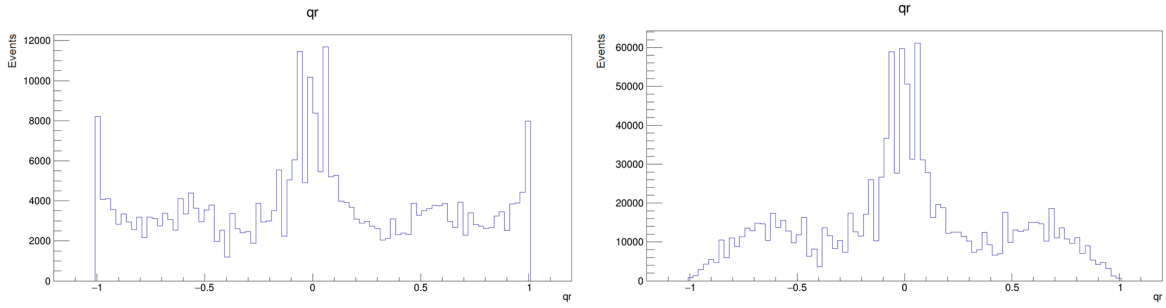

Figure 4.5: Showing the $q.r$ distributions for $\mathcal{A}_{CP} = 0$ signal (left), and continuum (right).

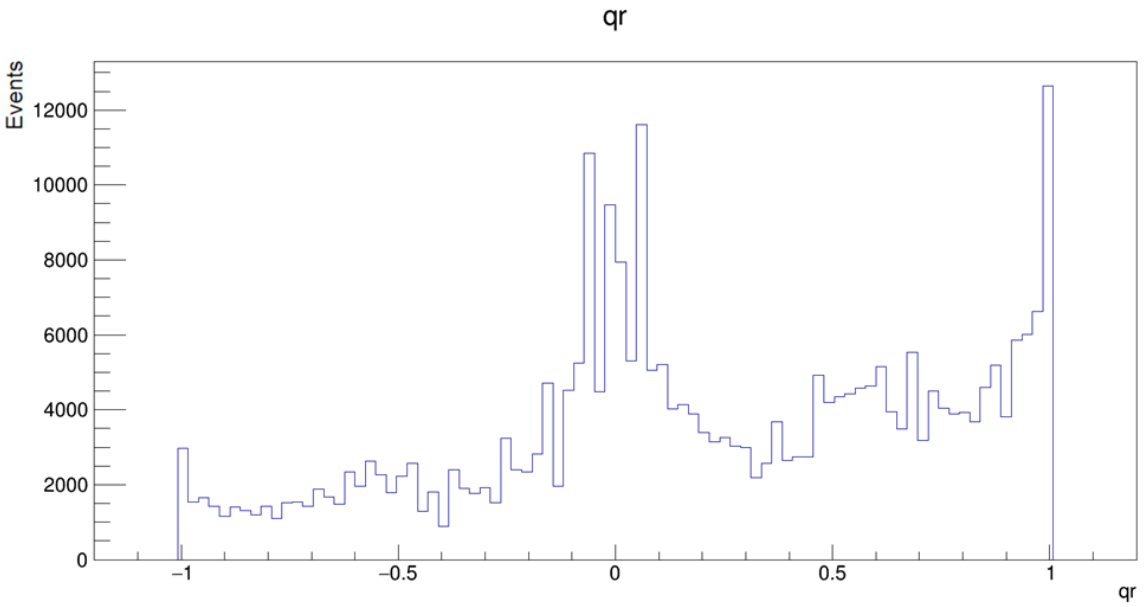

Figure 4.6: Showing the signal $q.r$ distribution for $\mathcal{A}_{CP} = +1$.

4.3 Kinematic Variables

Kinematic variables from the decay also have discriminating information for use in determining whether an event is signal or continuum. Figure 4.7 shows the differing event topologies for signal and background.

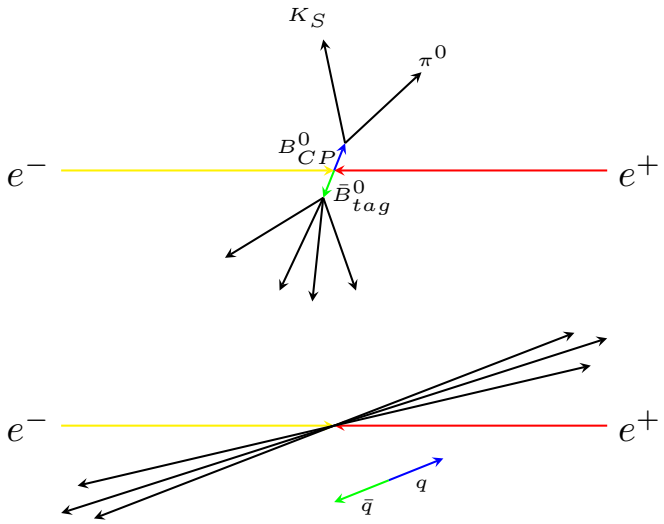

Figure 4.7: Showing more spherical signal(top) and jet-like continuum(bottom) decay topologies in COM frame.

Using the ExKFitter package, the tracks of the $\pi^{\pm}$ from the K_S are used to constrain the vertex of the signal B^0_{CP} decay. The B^0_{tag} vertex is calculated with the TagV package(which uses kfitter) using the charged particles not used in B^0_{CP}

vertexing. The momentum vectors, energies, decay vertices and charges of the B mesons and their decay products can then be used to calculate a range of useful kinematic variables.

4.3.1 ΔZ

The distance along the beamline axis between the decay vertices of B^0_{CP} and B^0_{tag} in COM frame is ΔZ. The reconstructed decay vertices of true signal events will have some separation due to the B-meson lifetime, the B^0 will travel some distance before decaying. Continuum events on the other hand have a much smaller ΔZ as the $q\bar{q}$ pair hadronise instantly, meaning both reconstructed B^0 decay vertices will be close to the IP point. See Figure 4.8 showing the larger spread in ΔZ for signal events compared to continuum.

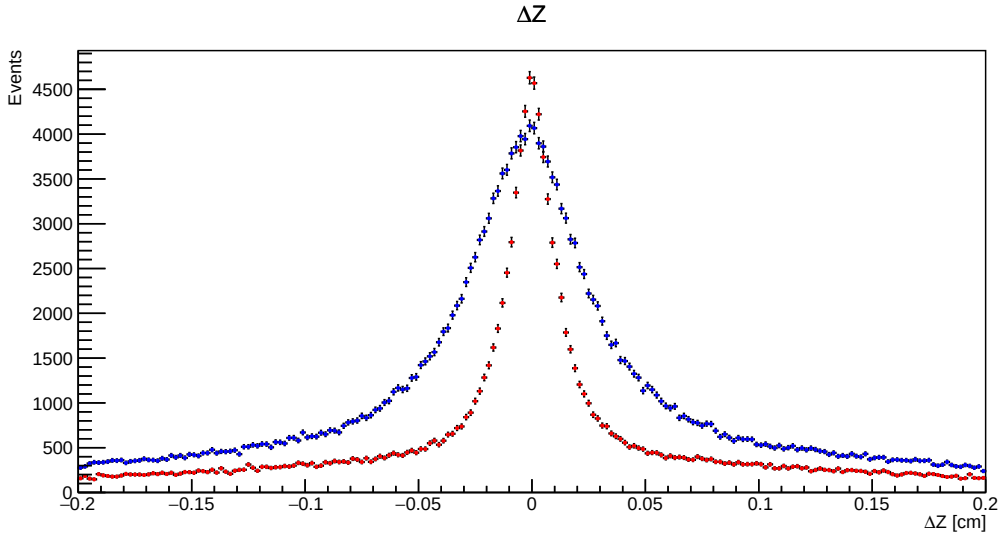

Figure 4.8: Showing scaled signal MC (blue) and continuum MC (red) ΔZ distributions.

4.3.2 $\cos(\theta_B)$

The angle between the reconstructed B^0_{CP} momentum and the beamline in COM frame (θ_B) can also be used to distinguish signal from continuum. The $|\cos(\theta_B)|$ distributions for signal and continuum are shown in Figure 4.9. The $\Upsilon(4s)$ is a vector meson (spin 1) and so the final state with two scalar (spin 0) B-mesons must have an orbital angular momentum of 1. So the angular distribution for signal follows a $1 - \cos^2(\theta)$ distribution. This results in the $\cos(\theta_B)$ distribution seen in Figure 4.9. On the other hand, the continuum $\cos(\theta_B)$ distribution is roughly uniform due to being composed of randomly selected tracks and acceptance effects, see [13] for more details.

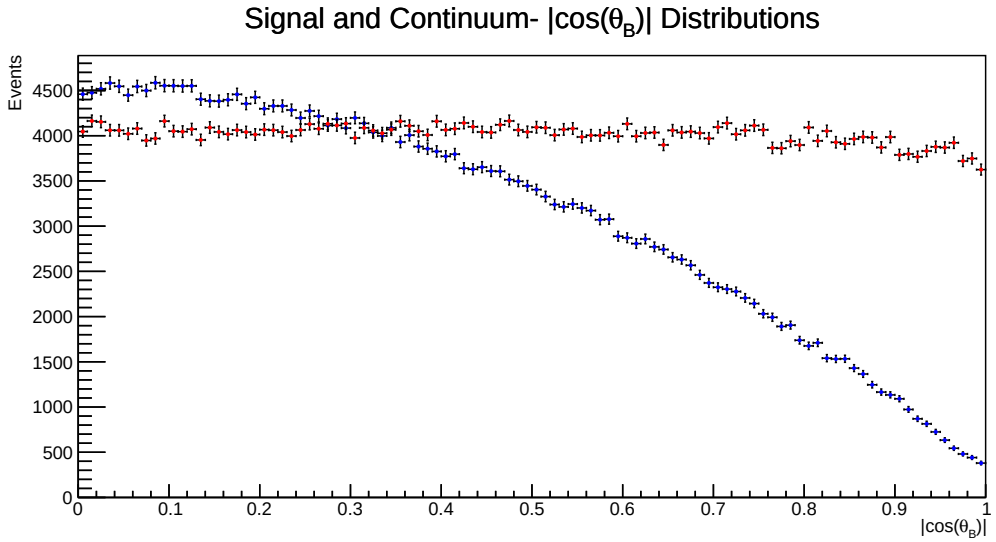

Figure 4.9: Showing scaled signal MC (blue) and continuum MC (red) $|\cos(\theta_B)|$ distributions.

4.3.3 $\cos(\theta_{thrust})$

The scalar thrust T given a set of N particles is given by choosing the unit vector $\hat{n}$ (the thrust vector) that maximises:

$$T = \frac{\sum_{i=1}^{N} |\hat{n} \cdot \vec{p}_i|}{\sum_{i=1}^{N} |\vec{p}_i|} \tag{4.6}$$

Where $\vec{p}_i$ is the momentum of particle i. The signal thrust is calculated where the sum is over the B^0_{CP} daughters, and the rest-of-event thrust is calculated where the sum is over all remaining particles. The angle between the signal thrust vector and the rest-of-event thrust vector is θ_{thrust} (in COM frame). In COM frame the B-mesons are produced nearly at rest, so the thrust axis are randomly distributed, and so $\cos(\theta_{thrust})$ follows a roughly uniform distribution. Continuum events, with the back to back jet like decay topology, will see the thrust axes strongly collimated, and so the $\cos(\theta_{thrust})$ distribution is sharply peaked at ± 1. See Figure 4.10.

4.3.4 The KSFW-moments

The shape of the event can be further described by the Fox-Wolfram moments[28]. By constructing a set of moments from the B-decay daughters a set of mostly uncorrelated variables can be constructed. In Belle the improved Kakuno-Super-Fox-Wolfram(KSFW) moments were originally developed for $B \to \pi\pi$ decays and are an improvement over the Fox-Wolfram moments in the case of two or three body charmless decays[29]. They are constructed using the momenta, charges and Legendre polynomials using the angles between the flight directions of the daughter particles (all in COM frame). They are divided into two categories; '*so*' and '*oo*' where '*s*' means that daughter particles from the B^0_{CP} are used and '*o*' corresponds

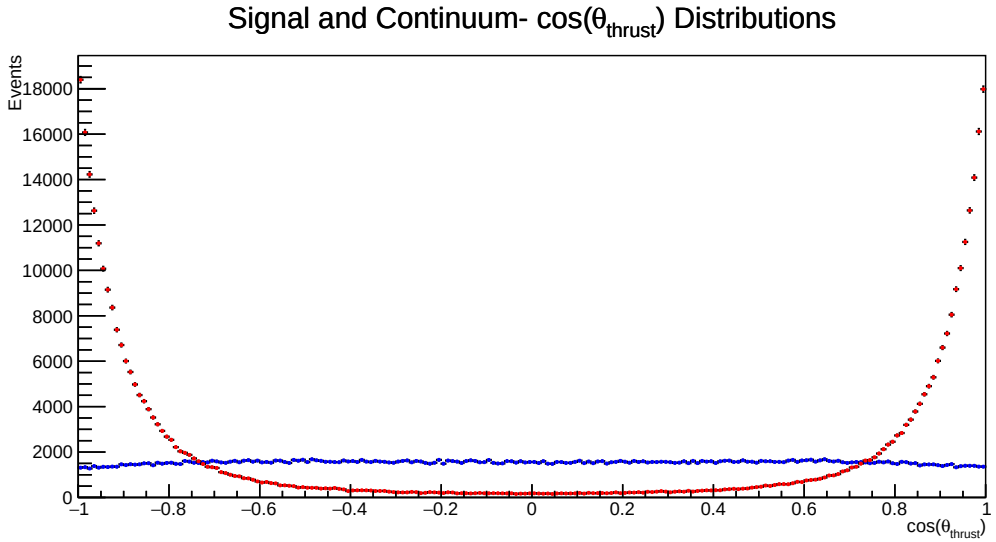

Figure 4.10: Showing scaled signal MC (blue) and continuum MC (red) $\cos(\theta_{thrust})$ distributions.

to the particles from the rest-of-event ('o' being decay products from the *other* B - the B^0_{tag}). 'so' then means that the calculation is performed over each signal decay product and each of the rest-of-event particles, 'oo' is just over rest-of-event particles. 'ss' is not used as summing over just the relations between signal decays will introduce high correlations between the variables, and with ΔE. The 'so' KSFW-moments are further divided into three categories indexed by x, where x is the 'type' of the rest-of-event; either charged ($x = 0$), neutral ($x = 1$) or missing ($x = 2$) i.e. summing over rest-of-event charged particles, neutral particles, and missing particles (reconstructed momentum that doesn't correspond to measured particles) respectively.

The 'so' KSFW-moments for Legendre polynomials of order(l) 1 and 3 are given by:

$$R^{so}_{xl} = \frac{\sum_a \sum_b q_a q_b |\vec{p}_b| P_l\left(\cos\left(\theta_{ab}\right)\right)}{E_{beam} - \Delta E} \tag{4.7}$$

Where a runs over the signal B daughter particles, and b runs over the rest-of-event particles in each x category. Here q_a and q_b are the charges of particles a and b respectively. P_l is the Legendre polynomial of order l, and θ_{ab} is the angle between particles a and b. As can be seen, q_a is always zero as the B^0_{CP} decay products are neutral, and therefore R^{so}_{xl} are zero when $l = 1, 3$.

For $l = 0, 2, 4$ on the other hand, the so KSFW-moments are given by:

$$R^{so}_{xl} = \frac{\sum_a \sum_b |\vec{p}_b| P_l\left(\cos\left(\theta_{ab}\right)\right)}{E_{beam} - \Delta E} \tag{4.8}$$

Giving us three 'so' KSFW-moments ($x = 0, 1, 2$) for each of the even order Legendre polynomials ($l = 0, 2, 4$).

The '*oo*' KSFW-moments for $l = 1, 3$ for are given by:

$$R_l^{oo} = \frac{\sum_a \sum_b q_a q_b |\vec{p}_a||\vec{p}_b| P_l\left(\cos\left(\theta_{ab}\right)\right)}{(E_{beam} - \Delta E)^2} \tag{4.9}$$

Where both a and b run over the rest-of-event particles. For $l = 0, 2, 4$ they are given by:

$$R_l^{oo} = \frac{\sum_a \sum_b |\vec{p}_a||\vec{p}_b| P_l\left(\cos\left(\theta_{ab}\right)\right)}{(E_{beam} - \Delta E)^2} \tag{4.10}$$

Giving 5 '*oo*' KSFW-moments, amounting to 14 KSFW-moments in total. The factors of $(E_{beam} - \Delta E)$ are included to normalise the KSFW-moments in such a way as to remove the ΔE correlations. The KSFW distributions are shown in Figures 4.11 - 4.24.

In addition to these there are two extra parameters calculated in this process. The sum of the transverse momenta (from the beamline) over all particles:

$$p_t^{sum} = \sum_{n=1}^{N} |\vec{p}_{t,n}| \tag{4.11}$$

Where $\vec{p}_{t,n}$ is the transverse component of the momentum of particle n, and N is the total number of particles. The p_t^{sum} distributions for signal and continuum are shown in Figure 4.25. As the vector of the total missing momentum is equal to $\sum_{n=1}^{N} -\vec{p}_n$, where $\vec{p}_n$ is the momentum of particle n, the squared-missing-mass is defined as :

$$M_{miss}^2 = \left(2E_{beam} - \sum_{n=1}^{N} E_n\right)^2 - \left|\sum_{n=1}^{N} \vec{p}_n\right|^2 \tag{4.12}$$

Where E_n is the energy of particle n. The distributions are shown in Figure 4.26

Each of the KSFW-moments (and the other kinematic variables) have some discriminating information from the event, but not enough individually to confirm that an event is signal or background. These moments are traditionally combined into a Fisher discriminant on which a selection can be placed. A common method is to combine them using multivariate classifiers such as boosted decision trees or neural networks.

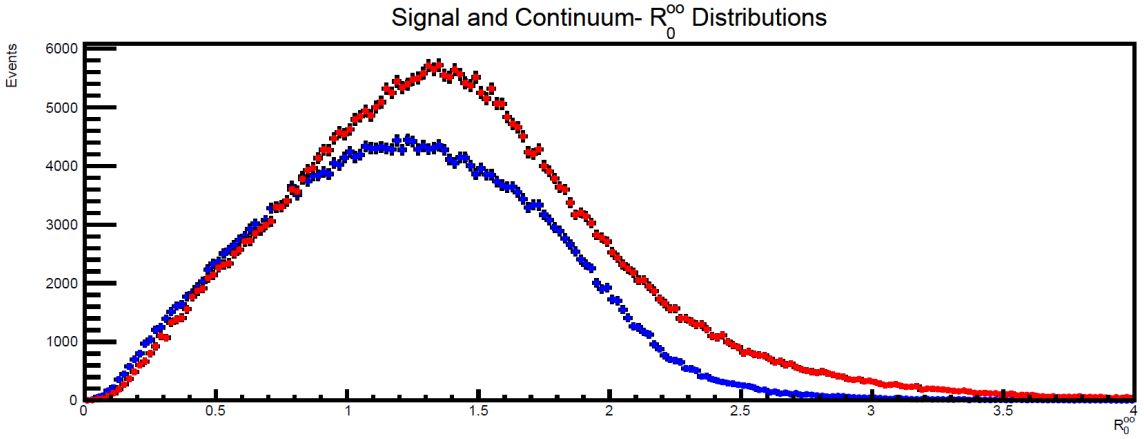

Figure 4.11: Showing the scaled signal MC (blue) and continuum MC (red) R_0^{oo} distributions.

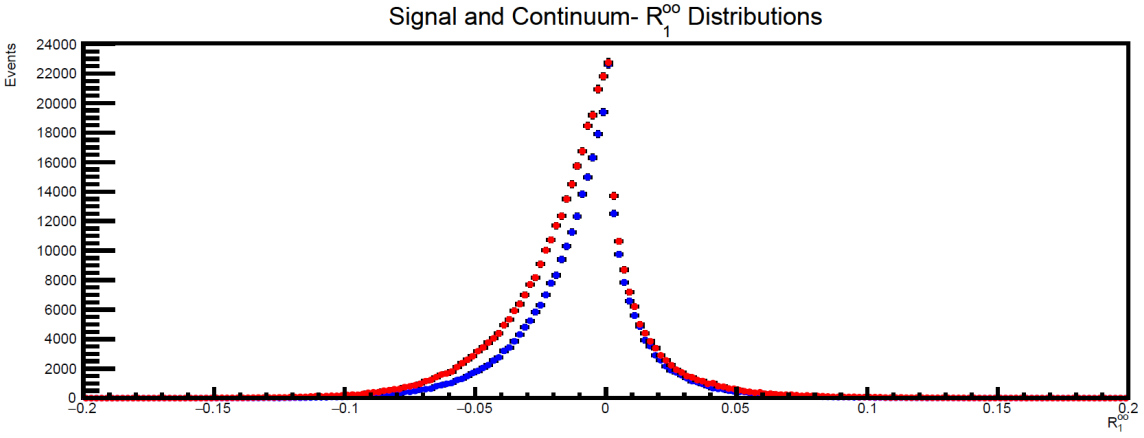

Figure 4.12: Showing the scaled signal MC (blue) and continuum MC (red) R_1^{oo} distributions.

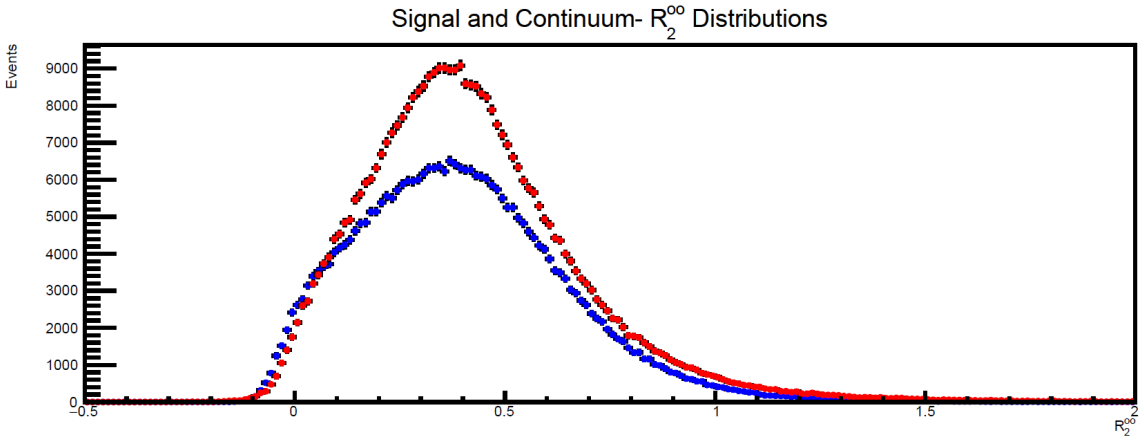

Figure 4.13: Showing the scaled signal MC (blue) and continuum MC (red) R_2^{oo} distributions.

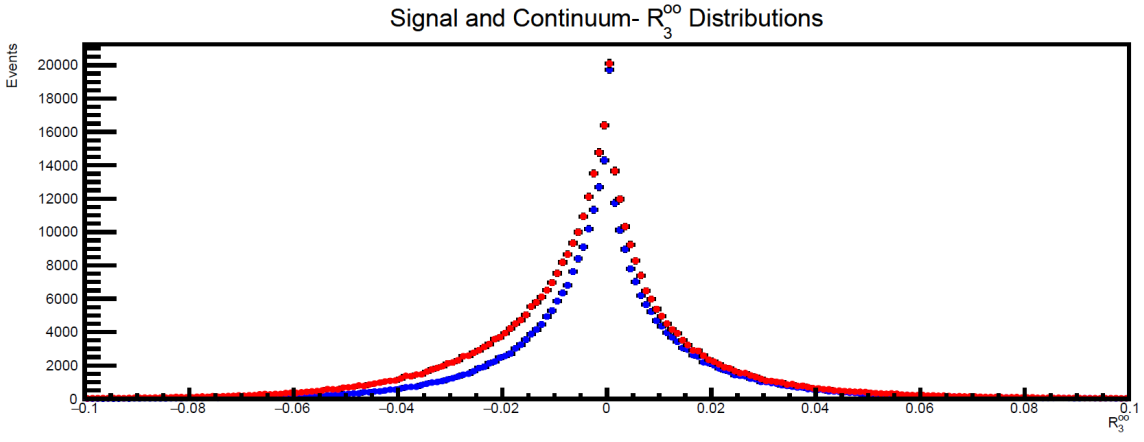

Figure 4.14: Showing the scaled signal MC (blue) and continuum MC (red) R_3^{oo} distributions.

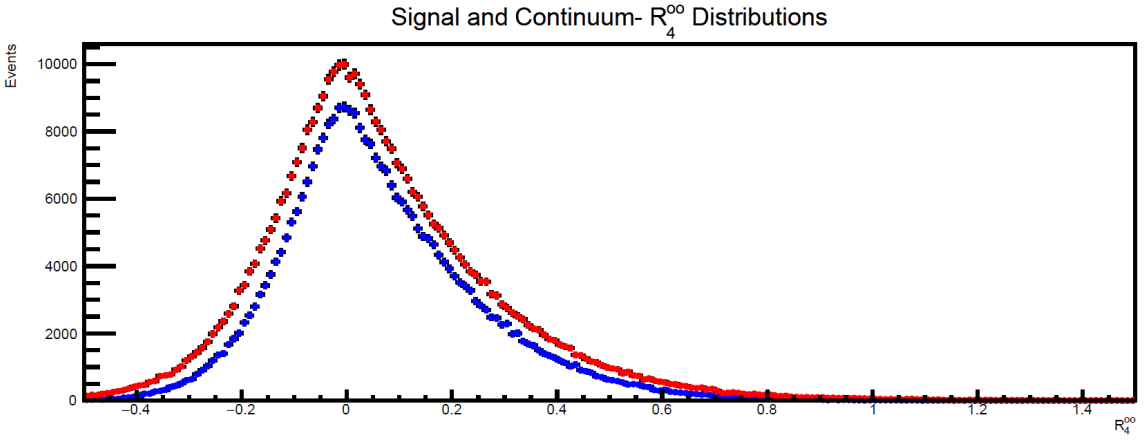

Figure 4.15: Showing the scaled signal MC (blue) and continuum MC (red) R_4^{oo} distributions.

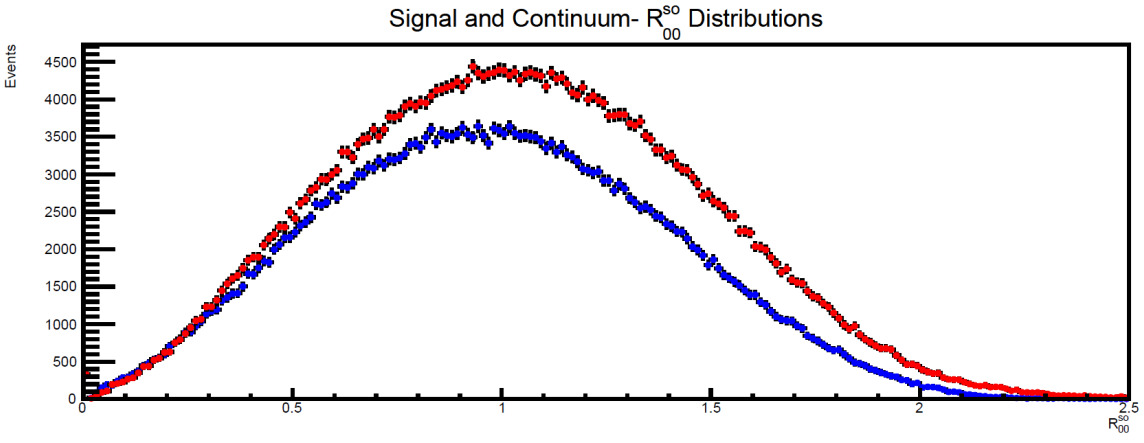

Figure 4.16: Showing the scaled signal MC (blue) and continuum MC (red) R_{00}^{so} distributions.

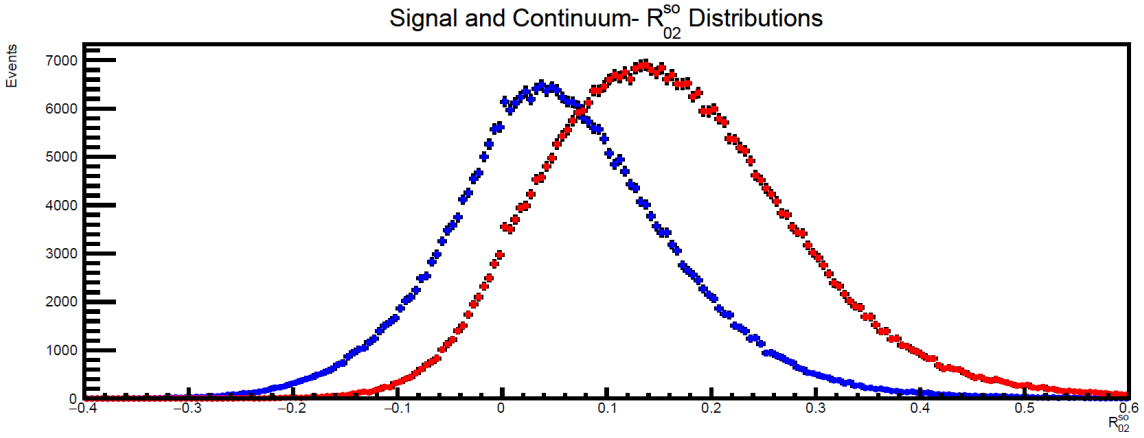

Figure 4.17: Showing the scaled signal MC (blue) and continuum MC (red) R_{02}^{so} distributions.

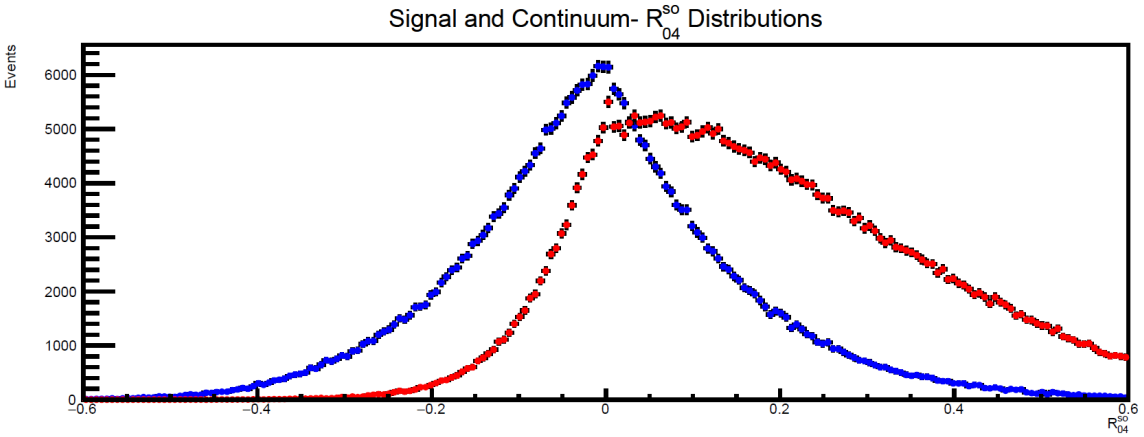

Figure 4.18: Showing the scaled signal MC (blue) and continuum MC (red) R_{04}^{so} distributions.

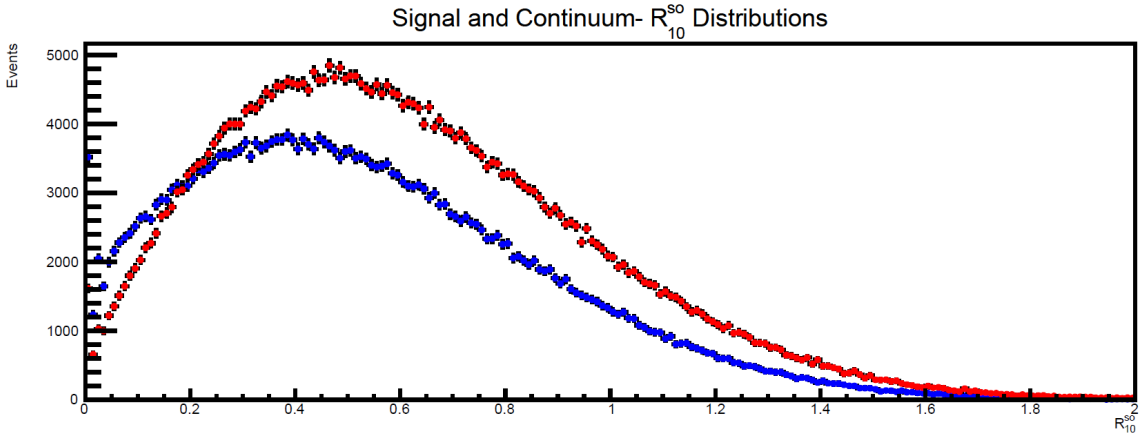

Figure 4.19: Showing the scaled signal MC (blue) and continuum MC (red) R_{10}^{so} distributions.

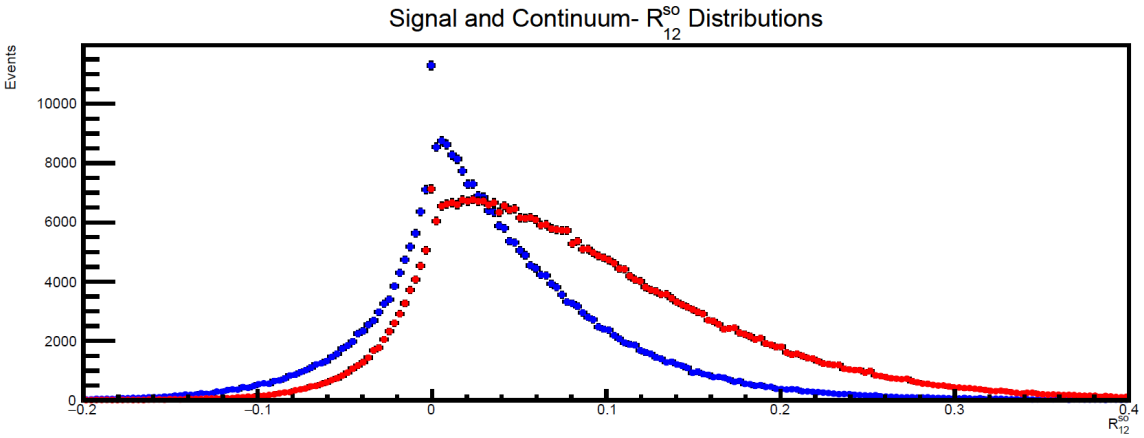

Figure 4.20: Showing the scaled signal MC (blue) and continuum MC (red) R_{12}^{so} distributions.

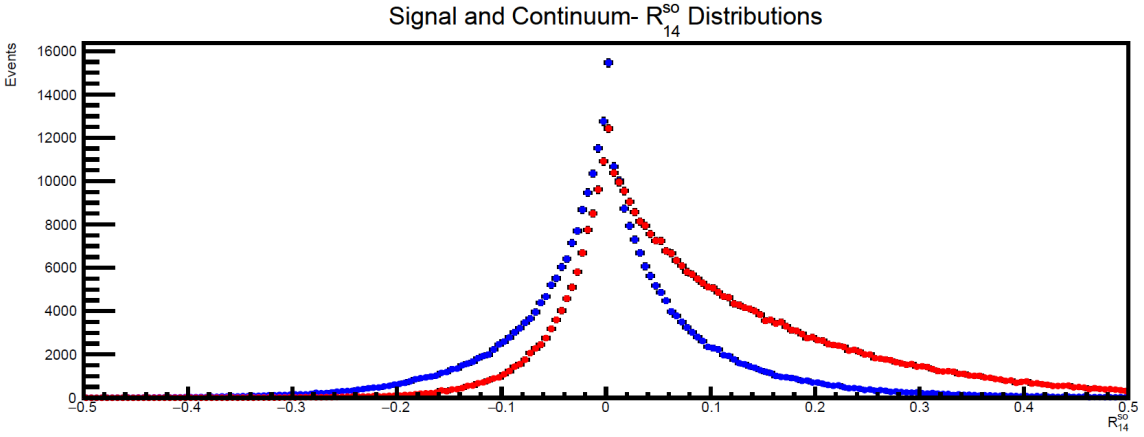

Figure 4.21: Showing the scaled signal MC (blue) and continuum MC (red) R_{14}^{so} distributions.

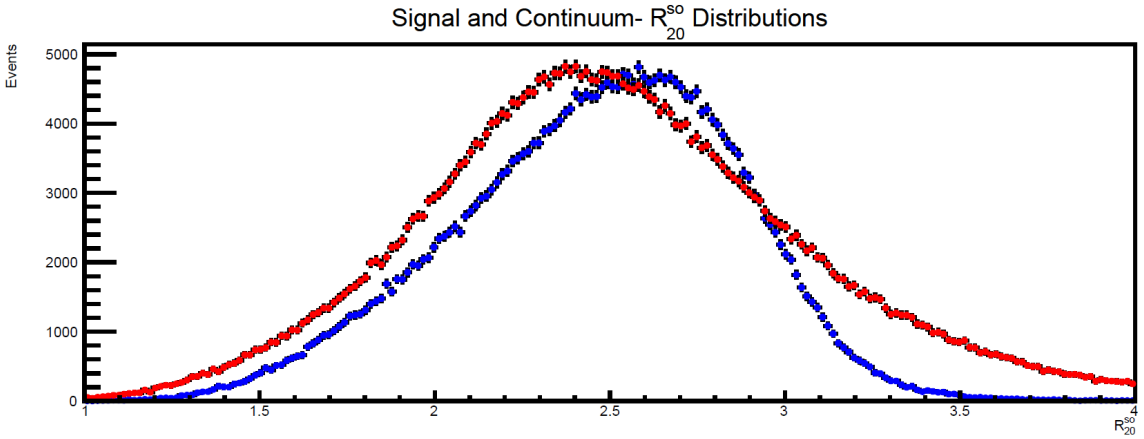

Figure 4.22: Showing the scaled signal MC (blue) and continuum MC (red) R_{20}^{so} distributions.

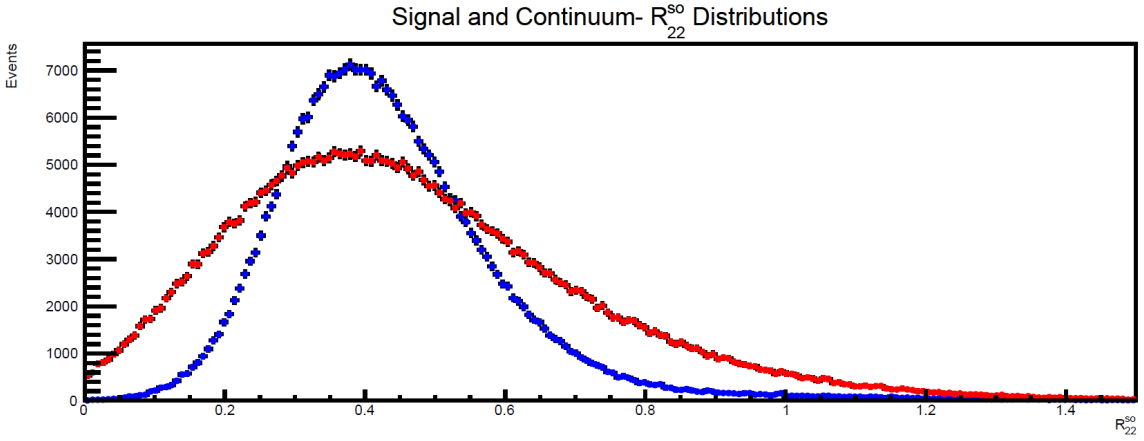

Figure 4.23: Showing the scaled signal MC (blue) and continuum MC (red) R_{22}^{so} distributions.

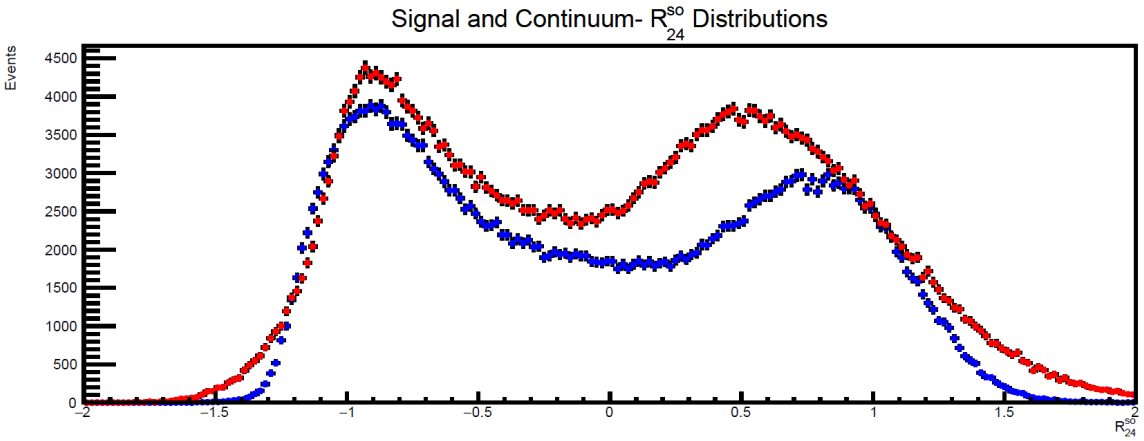

Figure 4.24: Showing the scaled signal MC (blue) and continuum MC (red) R_{24}^{so} distributions.

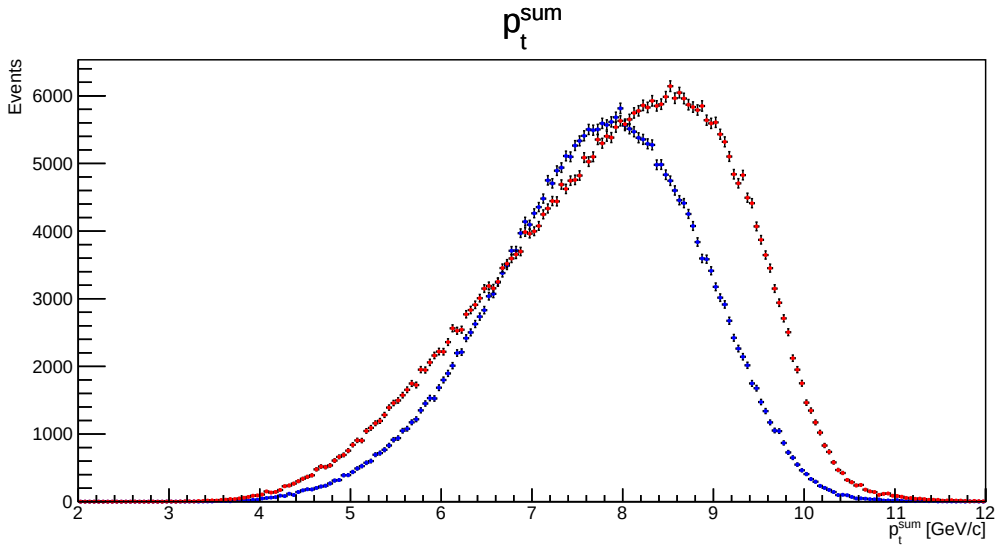

Figure 4.25: Showing the scaled signal MC (blue) and continuum MC (red) p_t^{sum} distributions.

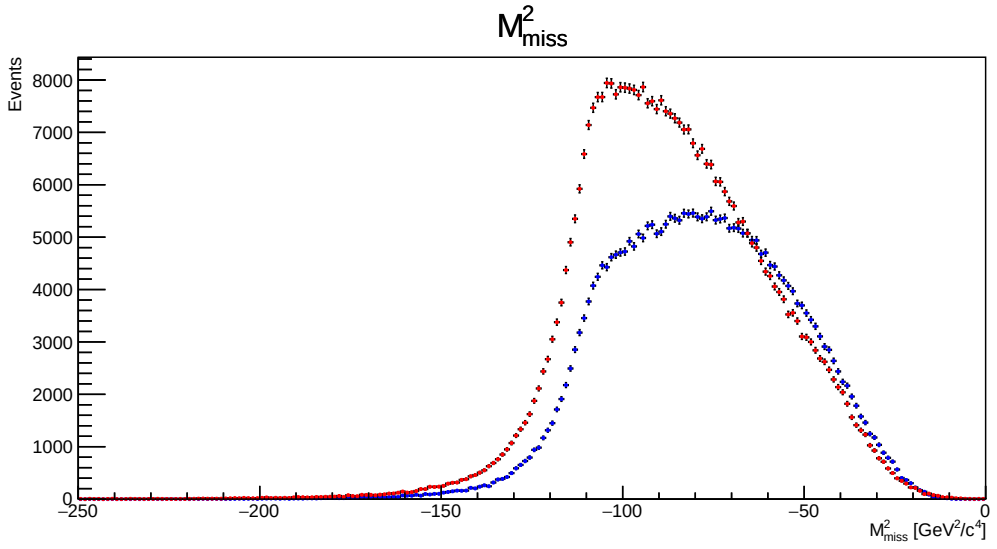

Figure 4.26: Showing the scaled signal MC (blue) and continuum MC (red) M^2_{miss} distributions.

5|Neural Networks for Continuum Suppression

Continuum suppression is achieved using neural networks to combine the kinematic information (see 4.3) into a single variable, the classifier output. This output can then be used to distinguish signal and continuum, by enforcing a selection, or a more complex analysis.

A basic neural network is essentially a complex matrix function taking a vector of input values and returning a vector of outputs. In the case of continuum suppression, we want a scalar output corresponding to the likelihood that the input vector for an event corresponds to a signal or continuum.

A feed-forward neural network is described by an architecture of connected nodes. Each node takes a weighted sum of the inputs and passes it through a non linear activation function (necessary as the best classification function is unlikely to be a linear combination of the input data).

A hidden layer is a set of nodes where each node takes a weighted sum of the outputs of the nodes in the previous layer (or the input data in the case of the first hidden layer), where multiple layers and a large number of nodes allow very complex relations between the input variables to be learned. Figure 5.1 shows and example feed-forward neural network with three hidden layers and a single output.

The output of a node j in layer k is given by:

$$x_j^k = f_j^k \left(w_{bj}^k b^k + \sum_i w_{ij}^{k-1} x_i^{k-1} \right) \tag{5.1}$$

Where $f_j^k(x)$ is the activation function of node j in layer k, traditionally the sigmoid function - $f(x) = (1+e^{-x})^{-1}$ - has been used. The node b^k represents the bias node for layer k, which is set to 1, w_{bj}^k is the weighting the bias has for node j in layer k. This bias is needed as it allows the input to the activation function to be shifted, resulting in different possible output values for a given weighted sum of inputs. The b^k term is redundant but is useful for visualising the model.

Once a particular network architecture has been chosen, the performance of the network depends on the values of the weights. The weights must be initialised to reasonable initial values (shouldn't have $|w_{ij}^k|$ much greater than 1). The weights are adjusted using a training dataset, where each event is known to be either signal or continuum. The weights are trained to target values ($\hat{y}$), for example rewarding network outputs of 1 for signal and 0 for continuum (penalising the reverse). This

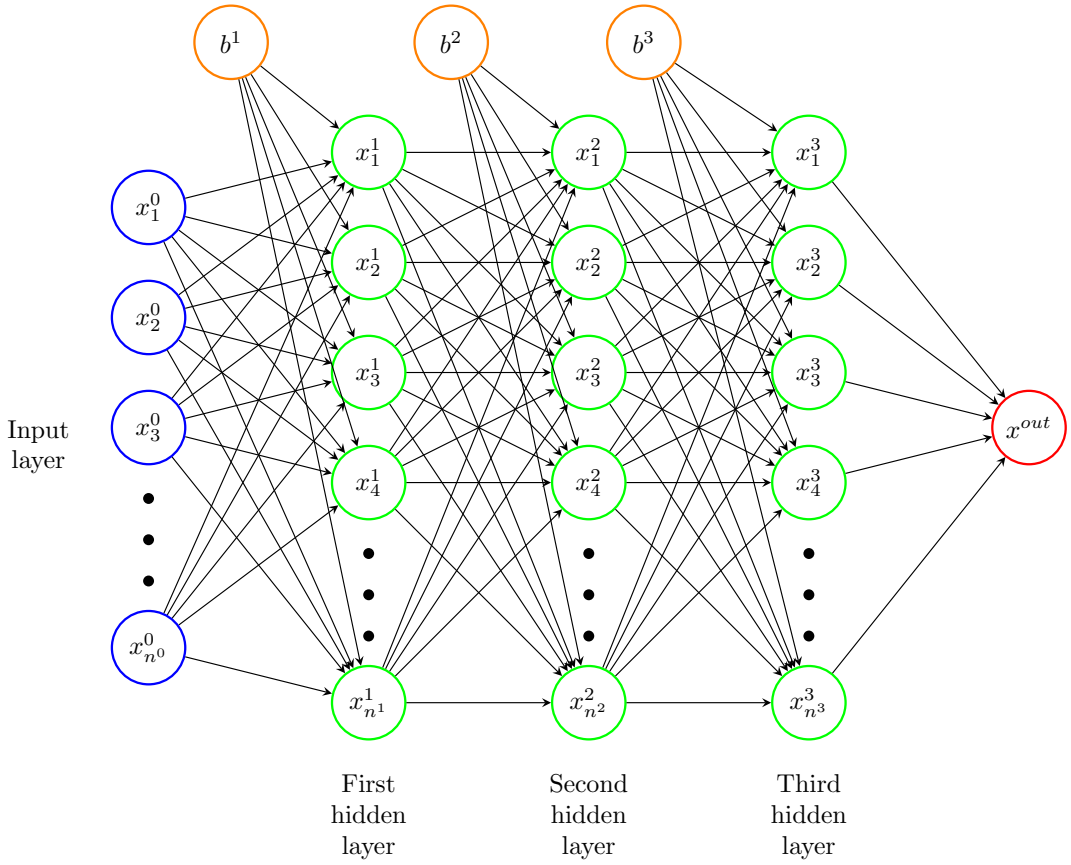

Figure 5.1: Showing an example feed-forward neural network with three hidden layers, n^k nodes in layer k and one output node, x^{out}. x_i^k corresponds to the ith node in layer k, x_i^0 correspond the the input parameters, and b^k are the biases added to layers k. The arrows correspond to the weights w_{ij}^k connecting the ith node in layer k to the jth node in the subsequent layer.

process is achieved by using back-propagation to minimise a cost (or loss) function (a function of the network output and the desired network output).

There are many options of loss function depending on the specifics of the situation. A good choice of loss function in the case of a binary classifier is the cross entropy loss, given by:

$$L\left(\vec{x}^0, \hat{y}\right) = -\hat{y} \cdot log\left(x^{out}\left(\vec{x}^0\right)\right) - (1-\hat{y}) \cdot log\left(1 - x^{out}\left(\vec{x}^0\right)\right) \tag{5.2}$$

Where x^{out} is the neural network work output as a function of the vector of inputs $\vec{x}^0$, and $\hat{y}$ is the target value, one for signal MC events and zero for continuum MC events. Thus the loss is lower when signal events produce a network output closer to one, and continuum events produce an output closer to zero.

The weights are adjusted based on the gradients of the loss function with respect to the weights. Back-propagation calculates the partial derivatives of L for each weight, and then the weights are individually shifted in the direction of the (negative) gradient. In this way, over many training steps over many events, the weights settle into the values that best minimise the overall loss. Of course the dimensionality is

large and there are many minima, so training large neural networks is not a trivial procedure.

Due to training depending on the gradients of the loss function with respect to each of the weights, the form of the activation has a large impact on training. The gradient of the sigmoid function at $|x| >> 1$ (where x is the input to the activation function) approaches zero. Therefore the training will be extremely slow if the inputs are far away from zero (the same is true for tanh), this is the vanishing gradient problem. Additionally the sigmoid function has the drawback that it is always positive which has been shown to be sub-optimal for training [30]. The activation functions are shown in Figure 5.2. An improved activation function which has been shown to be an improvement is the exponential linear unit (elu) [31]:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ e^x - 1, & \text{otherwise} \end{cases} \tag{5.3}$$

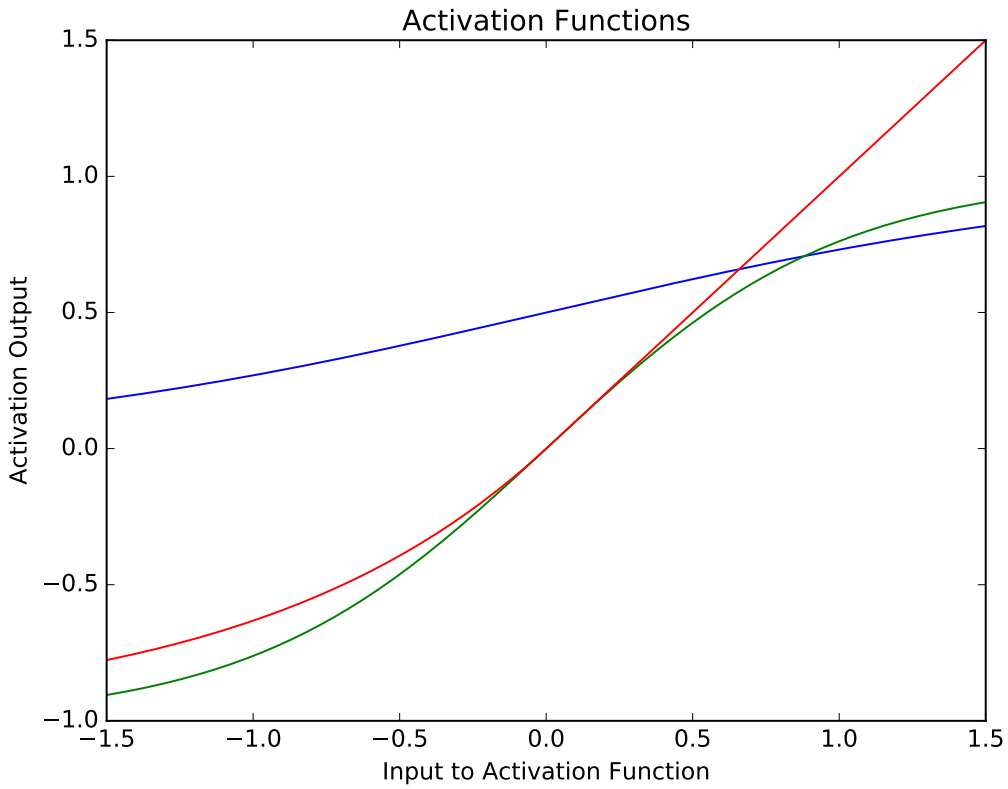

Figure 5.2: Showing the activation outputs for the sigmoid (blue), tanh (green) and elu (red) activation functions for inputs in the range ±1.5.

As the ability to train the network depends on the input values to the activation functions, the input data should be pre-processed in order to be within the range ±1. This can be achieved by scaling and shifting, or by more complex transformations, but is almost always vital.

There are a range of options (hyper-parameters) relating to the training of the neural network that must be chosen and tuned, and are heavily dependent on the input data, the neural network architecture, and each other. Some are laid out below, with the descriptions of the impact that they have is assuming that the other hyper-parameters are held constant.

- The learning rate defines the size of the steps taken when adjusting the weights. Too large a learning rate will result in the weight values oscillating around the minima, resulting in a sub-optimal network. Too small a learning rate will often lead to the weights settling to a false minima (in addition to training being slow).
- The number of epochs is the number of times that the entire dataset is trained over. Too small an epoch number and the network will not have had enough training time, and not learned all of the information obtainable from the training dataset. Too large an epoch number, in addition to taking longer than needed, often leads to over-training to the training dataset.
- The batch size defines the number of events that the network trains on in one step. Often an individual event will have particular quirks that are not necessarily representative of the data at large, leading to the weights being shifted in the wrong directions, and a rough, noisy path being taken to the minima. This is alleviated by training over a batch of events. Too small a batch size and the benefits of batching aren't seen. Too large a batch size and the subtler details in the training set can be lost, and the total number of training steps will be too small.

A larger batch size needs a smaller learning rate to keep the learning speed the same (as training to multiple events at once, a batch size of 10 would need a learning rate one-tenth the size of the learning rate needed for a batch size of one). A larger batch size results in fewer training steps per epoch, so more epochs are needed. As can be seen, the interplay between these three hyper-parameters must be taken into account. There can be many variable hyper-parameters, depending on the neural network and algorithms employed.

In addition to the training dataset, the quality of the network is tested against a separate testing datastet to verify its performance on new data. Regularisation refers to a range of procedures by which the classifying ability of the neural network is purposefully stunted when it is performing well to the training dataset but not the testing dataset. Often a neural network will be fine tuned to the training data (over-trained) and its actual performance is not optimal, regularisation is therefore vital. There are many ways to do this, most simple being early stopping; when the loss on the testing dataset no longer decreases, further training would only improve the performance on the training dataset.

Optimising the neural network to the testing dataset could also lead to a configuration that is optimal specifically with regards to that dataset. This is unavoidable but the true performance, and later physics analysis, is performed on a third (the validation) dataset which is not at all involved in the training and hyper-parameter selection procedure.

6|NeuroBayes and Analysis of $B^0 \to K^0_S\pi^0$ Events

The NeuroBayes software is introduced and the setup for continuum suppression is described. The method for measuring $\mathcal{A}_{CP}$ is then laid out in detail, and the results of this analysis using the NeuroBayes-processed datasets is covered. Note that unlike in previous chapters, when referring to the number of events and selection efficiencies, unless otherwise stated, the selection criteria 5.265 GeV$c^{-2} < M_{bc}^{corr} <$ 5.3 GeVc^{-2} and -0.4 GeV $< \Delta E < 0.3$ GeV is assumed (see 6.2).

6.1 NeuroBayes

Widely used at Belle for continuum suppression is the NeuroBayes neural network package [32]. A proprietary software, it uses a neural network and Bayes' theorem (hence the name) along with employing input data pre-processing and regularisation of the output. The internal architecture consists of one hidden layer where the number of nodes are specified by the user.

This neural network was trained with all nineteen kinematic variables (see 4.3) where the NeuroBayes pre-processing first transformed each of them to a Gaussian distribution. The number of nodes in the hidden layer was 21. The training batch-size was 100, and it was trained over 150 epochs using the Broyden–Fletcher–Goldfarb–Shanno algorithm (see [33] for more information). Regularisation is employed using the 'Bayesian regularisation procedure' (see [34] for details on the NeuroBayes algorithm). During training NeuroBayes employs pruning; removal of the least important weights to prevent overtraining. The loss function used is the cross-entropy (see equation 5.2).

In this study NeuroBayes was trained on 125,000 signal events (from the signal training dataset) and 125,000 continuum events (from the continuum training dataset; the first two continuum streams). All 250,000 training events were in the signal peaking ranges of 5.26 GeV$c^{-2} < M_{bc}^{corr} <$ 5.285 GeVc^{-2} and -0.4 GeV $< \Delta E < 0.2$ GeV. Additionally, only signal events where the B^0-mesons were correctly reconstructed were used in training.

The set up was not tweaked to improve performance, therefore this network was not applied to the testing datasets, and instead applied directly to the signal validation dataset and the continuum validation dataset (the final three continuum streams) which were then used in the following physics analysis.

6.1.1 Analysis of the Neural Network Performance

The validation datasets are processed by the trained neural network, using information from the kinematic variables (which individually don't provide much discriminating information) with a neural network can provide a clear separation between signal and continuum events, see Figure 6.1. Selections can the be placed on this variable (referred to as *NN* from here on) to remove most continuum events.

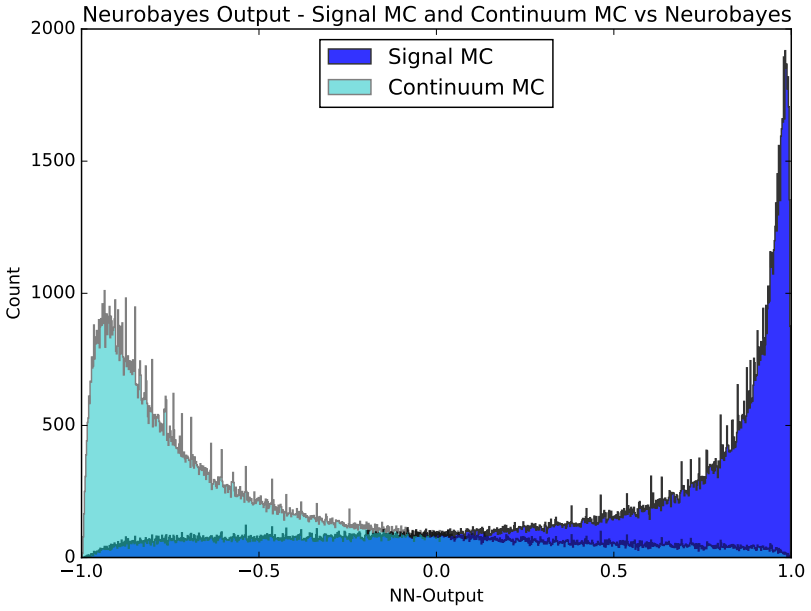

Figure 6.1: The NeuroBayes neural network output, *NN* for the continuum and signal validation datasets. The distributions shown have the same number of signal and continuum events and are not representative of the expected number of events.

Note that the signal validation dataset contains the events where the B^0-mesons were wrongly reconstructed, this could in principle further reduce the performance on the validation dataset compared to the training and testing datasets. These wrongly reconstructed events comprise 11.2% of the total signal validation dataset. Figure 6.2 shows *NN* for both properly and wrongly reconstructed B^0-mesons. The distribution for the wrongly reconstructed events is only slightly worse, and since it only forms a small part of the signal validation dataset, they are analysed together from here on.

As there is far more continuum than signal with 61385 ± 143 expected events compared to 1139 ± 61 expected events respectively, a tight selection must be placed. Figure 6.3 shows *NN* with the expected number of events; signal is clearly dwarfed by continuum. A common method of choosing a value of *NN* on which to place the selection criteria is the Figure of Merit (FOM), at a given *NN* value (NN_{cut}) it is defined as:

$$FOM\,(NN_{cut}) = \frac{N_{sig}}{\sqrt{N_{sig} + N_{cont}}} \tag{6.1}$$

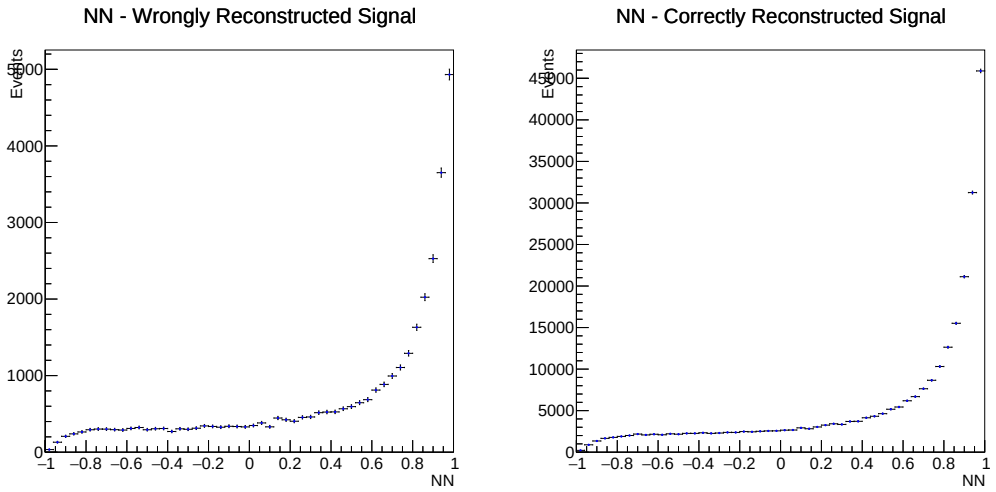

Figure 6.2: Showing the signal NN distributions for correctly (right) and incorrectly (left) reconstructed B^0 mesons.

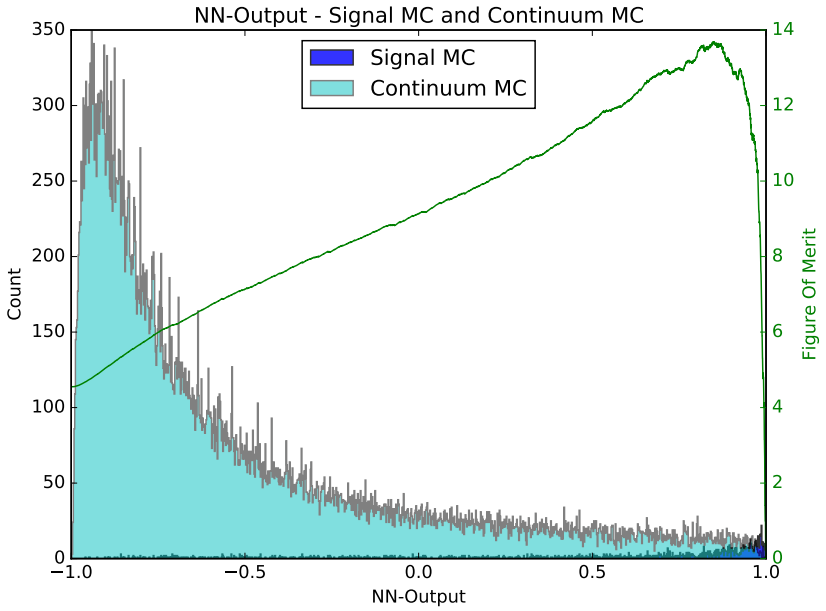

Figure 6.3: Showing the NN distributions for signal and continuum with the expected number of events. The figure of merit (green) is also plotted for possible NN_{cut} over the entire NN range.

Where N_{sig} and N_{cont} are the number of signal and continuum events with $NN > NN_{cut}$. The best FOM (in the whole NN range) can therefore be used as a measure of the neural network's performance.

The ability of a classifier to distinguish between two classes can be measured using the ROC (Receiver Operating Characteristic) curve. This is the plot of true-positive rate (the ratio of N_{sig} with $NN > NN_{cut}$ to the total number of signal

events) against the false-positive rate (the ratio of N_{cont} with $NN > NN_{cut}$ to the total number of continuum events) for all NN_{cut} over the whole range of NN. The ROC curve for signal MC and continuum MC is shown in Figure 6.4. A classifier that performs no better than random chance would have a ROC 'curve' of $y = x$. The measure of the performance can also be summarised with the AUC (Area Under the Curve), an AUC of 1 being a perfect classifier, and an AUC of 0.5 being no better than random.

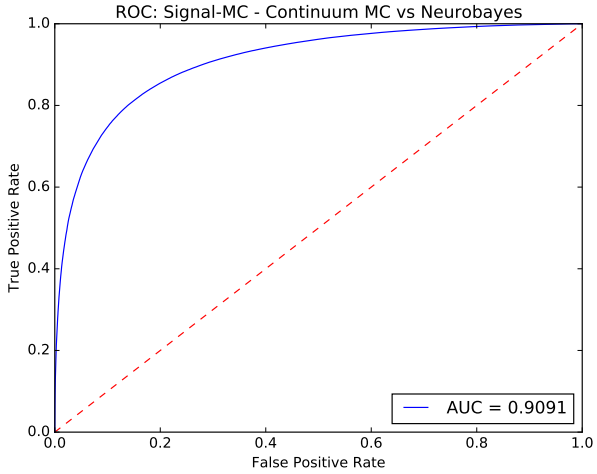

Figure 6.4: The ROC curve for signal and continuum MC, with an AUC of 0.909.

This neural network has an AUC of 0.909 (when using signal and continuum MC). The best FOM is 13.2 ± 0.4, where this value is obtained by finding the best FOM over 25 random signal and continuum MC samples (with the expected number of events), and its uncertainty is the standard deviation of these measurements. Selecting NN_{cut} to leave 13.00% of continuum leaves 79.01% of signal remaining. Similarly choosing a NN_{cut} to keep 70.20% of signal leaves 7.65% of continuum.

As this network was trained entirely with MC data, the performance of the network is further validated with (real) off-resonance data. The signal and off-resonance NN distributions are shown in figure 6.5. The ROC curve for signal with off-resonance is shown in figure 6.6. The AUC is 0.891, which is not significantly worse than for signal MC with continuum MC. Similarly as above; a value of NN_{cut} leaving 13.00% of continuum leaves 74.38% signal, and a value leaving 70.20% signal leaves 10.08%. As predicted by the AUC values, these percentages show that although the performance of the neural network is not significantly different for off-resonance data than continuum MC, it is clearly worse.

To maximise the signal yield, NN_{cut} is chosen to be -0.52 as preliminary investigations showed that the statistical uncertainty on $\mathcal{A}_{CP}$ would be lower with a lower NN_{cut} value than that given by the best FOM. This selection leaves 92.3% of signal, and 34.0% of continuum. In addition this leaves 89.5% and 88.7% of charged rare and mixed rare backgrounds respectively, NN for the rare backgrounds are shown in figure 6.7. This gives expected yields of :

- Signal : 1052 ± 57,

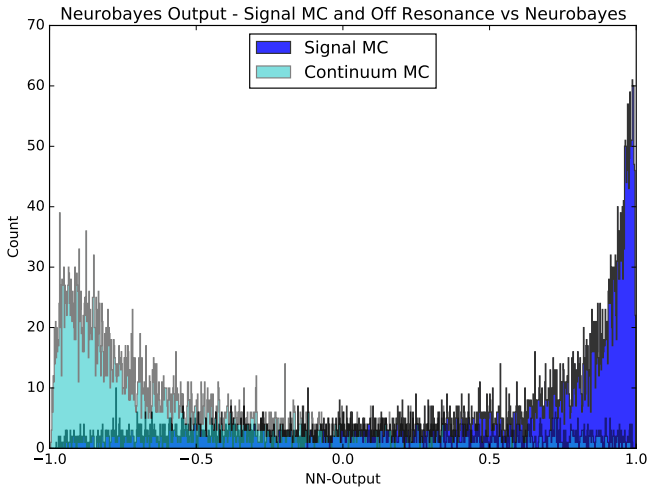

Figure 6.5: Showing the *NN* distributions for signal and off-resonance. Note that the noisiness of the distributions is due to the off-resonance data sample size being roughly one-thirtieth of the continuum MC validation dataset. A subsample of the signal validation dataset is chosen in order to have the same number of signal and off-resonance events.

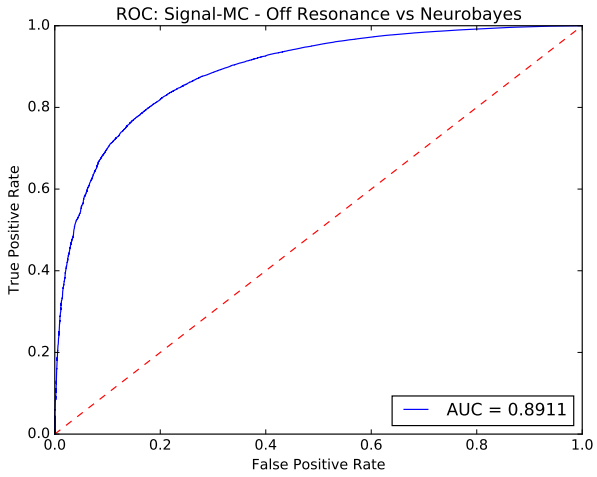

Figure 6.6: The ROC curve for signal and off-resonance, with an AUC of 0.891.

- Continuum : 20898 ± 49,
- Charged Rare: 331 ± 3,
- Mixed Rare: 126 ± 2,

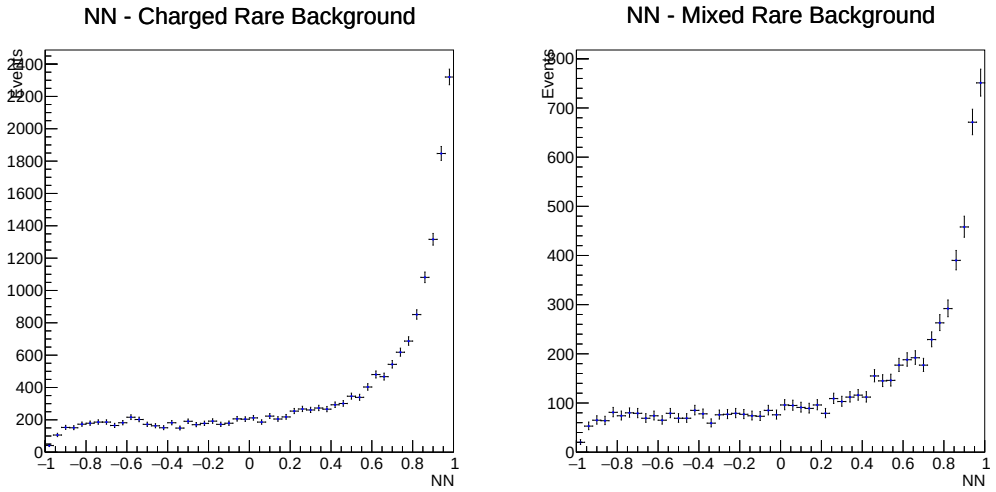

Figure 6.7: Showing the *NN* distributions for the charged (left) and mixed (right) rare backgrounds.

6.2 Analysis of $B^0 \to K^0_S\pi^0$ Events

Note that all likelihood fitting and distribution modelling is performed using the ROOT[35] statistics and data-analysis package (ver 6.04), and its extension RooFit[36] unless otherwise stated.

The signal yield (and therefore the branching ratio) and the $\mathcal{A}_{CP}$ are measured by fitting probability distribution functions (PDFs) to histograms in multiple dimensions. By fitting signal MC along with each of the background MC datasets, the expected PDF forms are found for each channel and fitting-variable (where 'channel' refers to either signal, continuum, charged rare or mixed rare datasets). Then by fitting these to samples with the expected number of events (and in principle real data, not done in this analysis) physical measurements can be extracted.

The signal yield is the area under the signal PDF (in one dimension). The $\mathcal{A}_{CP}$ is measured from the $q.r$ distribution. This is achieved first by fitting a kernel density estimation function to the signal $q.r$ distribution (with $\mathcal{A}_{CP} = 0$), see Figure 6.8. A kernel density estimation function combines a Gaussian for each data point to model the rough distribution, and is implemented using a RooKeysPdf (a RooFit Class) with mirroring at both edges and a smoothing factor (ρ, larger values corresponding to a smoother distribution) of 0.75. Then by taking the product of this with a 1st order polynomial with a fixed y-intercept but free gradient, and fitting this to the actual $q.r$ distribution, the $\mathcal{A}_{CP}$ can be extracted from the gradient. The signal $q.r$ PDF is therefore given by:

$$f^{q.r}_{signal}(q.r) = f^{q.r|\mathcal{A}_{CP}=0}_{signal}(q.r) \cdot (1 + \mathcal{A}_{CP} \cdot q.r \cdot (1 - 2\chi_d)) \tag{6.2}$$

Where $f^{q.r|\mathcal{A}_{CP}=0}_{signal}(q.r)$ is the signal kernel density estimation PDF for $\mathcal{A}_{CP} = 0$. $(1 - 2\chi_d)$ is the factor taking account of mixing (see 1.4). Figure 6.9 shows the $f^{q.r}_{signal}(q.r)$ fit to a $q.r$ signal $\mathcal{A}_{CP} = +1$ distribution. The $\mathcal{A}_{CP}$ is generated from

the signal $\mathcal{A}_{CP} = 0$ validation dataset by shifting the $q.r$ bin values accordingly (taking account of mixing).

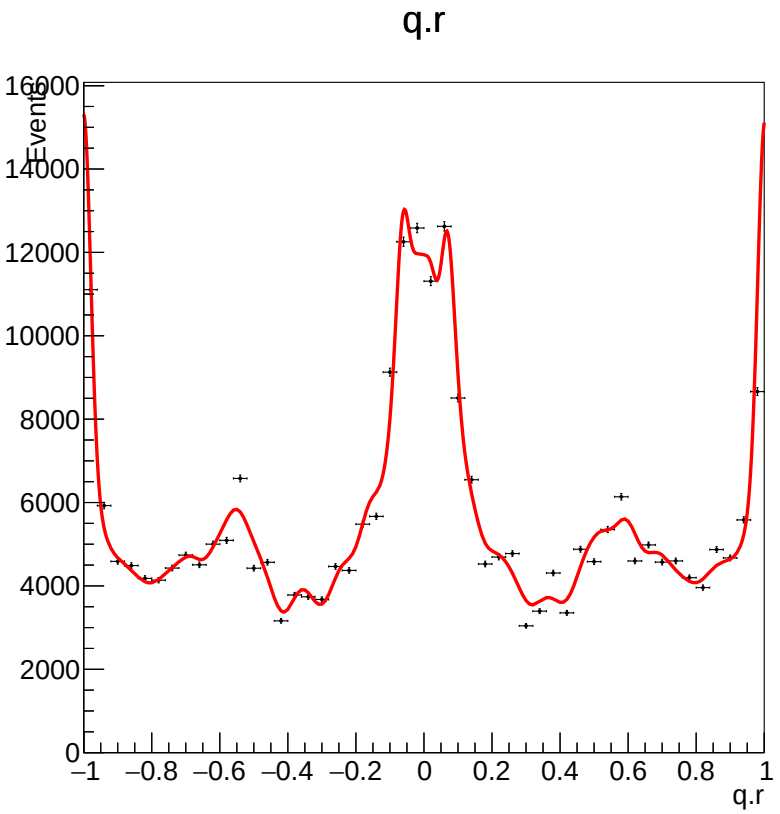

Figure 6.8: Showing the signal $q.r$ kernel density estimation PDF fit to the $\mathcal{A}_{CP} = 0$ signal validation dataset.

Due to the large and various backgrounds, a one dimensional PDF fit to $q.r$ would not have enough distinguishing information and the statistical uncertainty on the $\mathcal{A}_{CP}$ measurement would be huge. To account for this, the yield information for each channel is kept under control by fitting a four dimensional fit (for signal and separately for each of the backgrounds) to $q.r$, ΔE, M_{bc}^{corr} and NN^{trans}, where NN^{trans} is a transform on the neural network output, NN, given by:

$$NN^{trans} = log\left(\frac{NN - NN_{cut}}{NN_{max} - NN}\right) \tag{6.3}$$

Where $NN_{cut} = -0.52$ and $NN_{max} = 0.999591$ (the largest output given by the network). This transforms the NN distributions into Gaussian like distributions that are easier to model with analytic PDFs. This is important as even after the selection is placed on NN, there is still distinguishing information (between signal and the backgrounds) in the remaining NN region.

As the total number of events is unknown, the joint likelihood function used is an un-binned extended maximum likelihood function. The likelihood function in

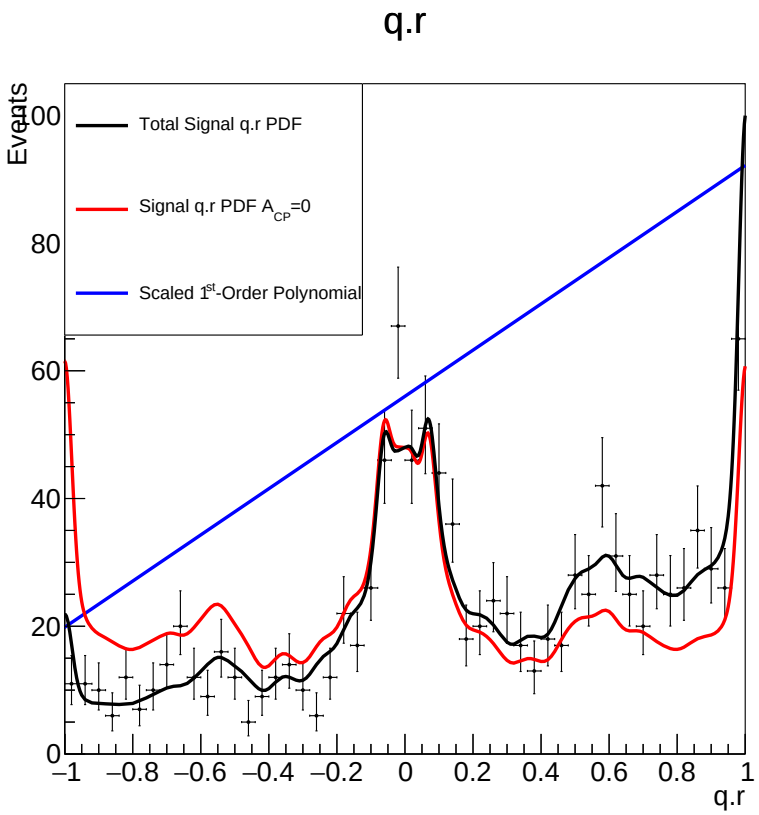

Figure 6.9: Showing the signal $q.r$ PDF fit to a $\mathcal{A}_{CP} = +1$ sample. The PDF (black) is the product of the kernel density estimation function (red) shown in Figure 6.8, and the 1st order polynomial (blue). Note that the polynomial is scaled in order to be visible, it has a y-intercept of one, ensuring that it only provides a skew to the distribution.

one dimension (for one channel, i.e. just signal) is given by [13]:

$$\mathcal{L} = \left(\prod_j^{N_{obs}} f(x_j) \right) \cdot \frac{e^{-N_{fit}}}{N_{obs}!} N_{fit}^{N_{obs}} \tag{6.4}$$

Where $f(x)$ is the one-dimensional probability distribution function in the variable x, and x_j is the x value of the jth event. N_{obs} is the observed number of events and N_{fit} is the number of events expected by the fitter.

We are dealing with multiple channels (where the number of events in each channel expected by the fitter is given by N_i where i =signal, continuum, charged rare, mixed rare) so $N_{fit} = \sum_i N_i$. As this is a four-dimensional fit, the likelihood function is then given by:

$$\mathcal{L} = \frac{e^{-\sum_i N_i}}{N_{obs}!} \prod_j^{N_{obs}} \left(\sum_i N_i \cdot f_i^{4d} \left([\Delta E]_j, [M_{bc}^{corr}]_j, [NN^{trans}]_j, [q.r]_j \right) \right) \tag{6.5}$$

Where $[k]_j$ is the k value for event j, (where k is ΔE, M_{bc}^{corr}, NN^{trans} or $q.r$), and

the four-dimensional PDF for channel i is given by:

$$f_i^{4d}\left(\Delta E, M_{bc}^{corr}, NN^{trans}, q.r\right) = \prod_k f_i^k\left(k\right) \tag{6.6}$$

Where f_i^k is the one-dimensional PDF in dimension k, for channel i. The 16 f_i^k PDFs are obtained individually, and then fixed, so the only free parameters when maximising $\mathcal{L}$ are N_i for signal and continuum (as $N_{charged-rare}$ and $N_{mixed-rare}$ are held constant, see 6.2.3), and $\mathcal{A}_{CP}$ (contained in $f^{q.r}_{signal}(q.r)$). In this way, the signal yield (i.e. the branching ratio) and $\mathcal{A}_{CP}$ can be measured.

The fitting regions (for the individual 1-dimensional PDFs and the 4-dimensional fit) are:

- -0.4 GeV $< \Delta E < 0.3$ GeV
- 5.265 GeV$c^{-2} < M_{bc}^{corr} <$ 5.3 GeVc^{-2}
- $-10.0 < NN^{trans} < 10.0$
- $-1.0 < q.r < 1.0$

6.2.1 Signal

As the wrongly reconstructed B^0-mesons only comprise 11.2% of the total signal dataset, and the distributions in each of the four dimensions are very similar(see Figure 6.10), these events are not treated separately.

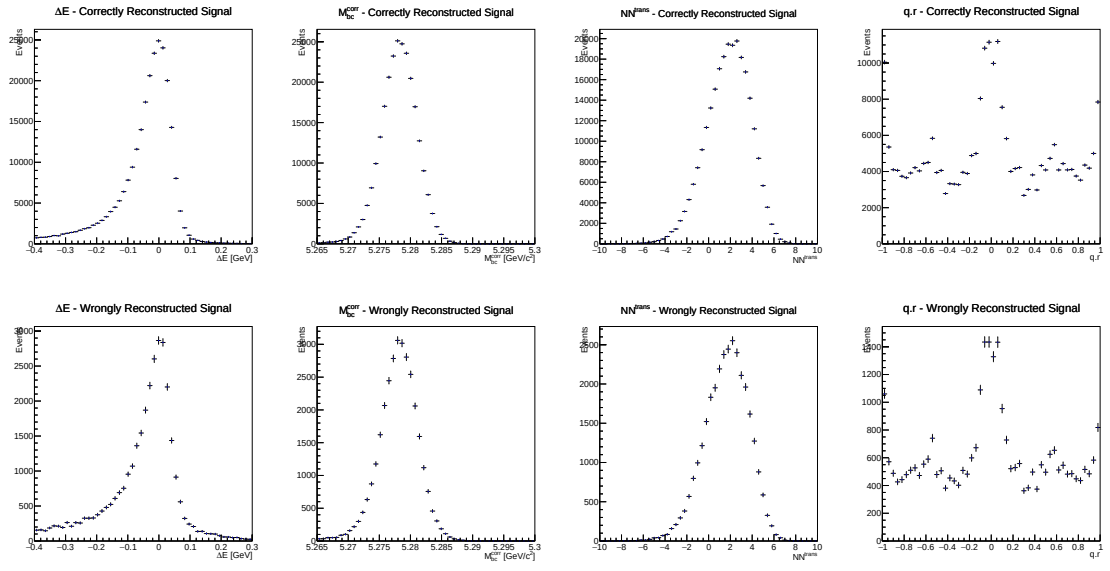

Figure 6.10: Showing the signal data distributions for (left to right) ΔE, M_{bc}^{corr}, NN^{trans} and $q.r$, for the correctly (top row) and incorrectly (bottom row) reconstructed B^0-mesons.

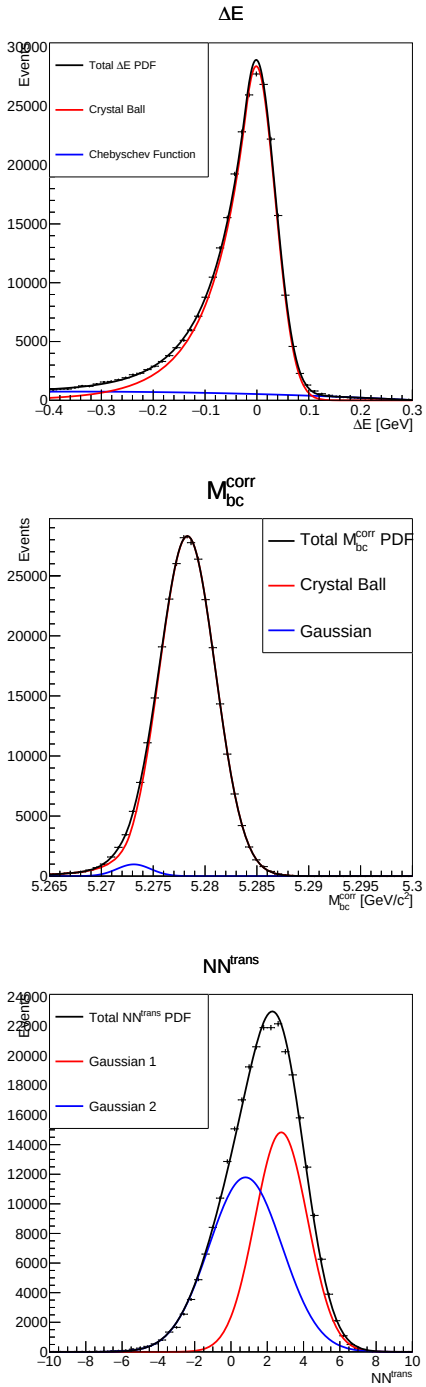

Figure 6.11: Showing the signal one-dimensional PDFs for ΔE (top), M_{bc}^{corr} (middle) and NN^{trans} (bottom), along with the component functions.

ΔE is fitted with a combination of a Crystal Ball function and a second order Chebyschev polynomial (basis set of the first kind). The Crystal Ball function is

defined by:

$$\mathrm{CB}(x;\alpha,n,\mu,\sigma) = \begin{cases} e^{-(x-\mu)^2/2\sigma^2}, & \text{if } \frac{x-\mu}{\sigma} < -\alpha \\ A(B - \frac{x-\mu}{\sigma})^{-n}, & \text{otherwise} \end{cases} \tag{6.7}$$

Where $A = (n/|\alpha|)^n)e^{-|\alpha|^2/2}$ and $B = (n/|\alpha|) - |\alpha|$. A function in the Chebyschev polynomial basis up to order N is defined as:

$$\mathrm{Cheby}^N(x;a_0,\ldots,a_N) = \sum_{n=0}^{N} a_n T_n(x) \tag{6.8}$$

$T_n(x)$ is the nth Chebyschev polynomial of the first kind. The M_{bc}^{corr} distribution is fit with a combination of a Crystal Ball and a Gaussian. The NN^{trans} distribution is fit with two Gaussians. The one dimensional signal PDFs for ΔE, M_{bc}^{corr} and NN^{trans} are shown in Figure 6.11. The $q.r$ signal distribution is introduced above, see Figure 6.8 and Equation 6.2.

The scatter plots in each pair of dimensions are shown in Appendix B.1 with their correlations shown in Table B.1. This is with the full processed signal validation dataset within the fitting regions. The small correlations suggest that fitting each dimension individually is justified. Note the largest correlation is $NN^{trans} - \Delta E$ of 3.8%.

6.2.2 Continuum

The one-dimensional continuum PDFs are shown in Figure 6.12. Continuum ΔE is fit with a Chebyschev function up to 3rd order. The M_{bc}^{corr} distribution for continuum is fit with an Argus function, defined as:

$$\mathrm{Argus}(x;m_0,c,p) = x \cdot \left(1 - \left(\frac{x}{m_0}\right)^2\right)^p \cdot e^{c\left(1-\left(\frac{x}{m_0}\right)^2\right)} \tag{6.9}$$

The NN^{trans} distribution for continuum is fit with two Gaussians. The continuum $q.r$ distribution is fit with a kernel density estimation function (RooKeysPdf) without edge mirroring and a smoothing of $\rho = 2$.

The scatter plots for every pair of dimensions for continuum are shown in Appendix B.1, with their correlations shown in Table B.2. As with signal, the correlations are small, the largest being $NN^{trans} - \Delta E$ at 4.4%

6.2.3 Rare Backgrounds

The ΔE distribution for the charged rare background is fit with a kernel density estimation PDF (due to difficulty finding a good analytic fit) with a smoothing of $\rho = 2$ and mirroring on the left-side only. The mixed rare ΔE distribution is fit with a combination of two Gaussians. The M_{bc}^{corr} distributions for both charged and mixed rare background are fit with Argus functions. Both charged and mixed rare NN^{trans} distributions are fit with a pair of Gaussians. Finally the $q.r$ distributions for both charged and mixed rare-backgrounds are fit with kernel density estimation

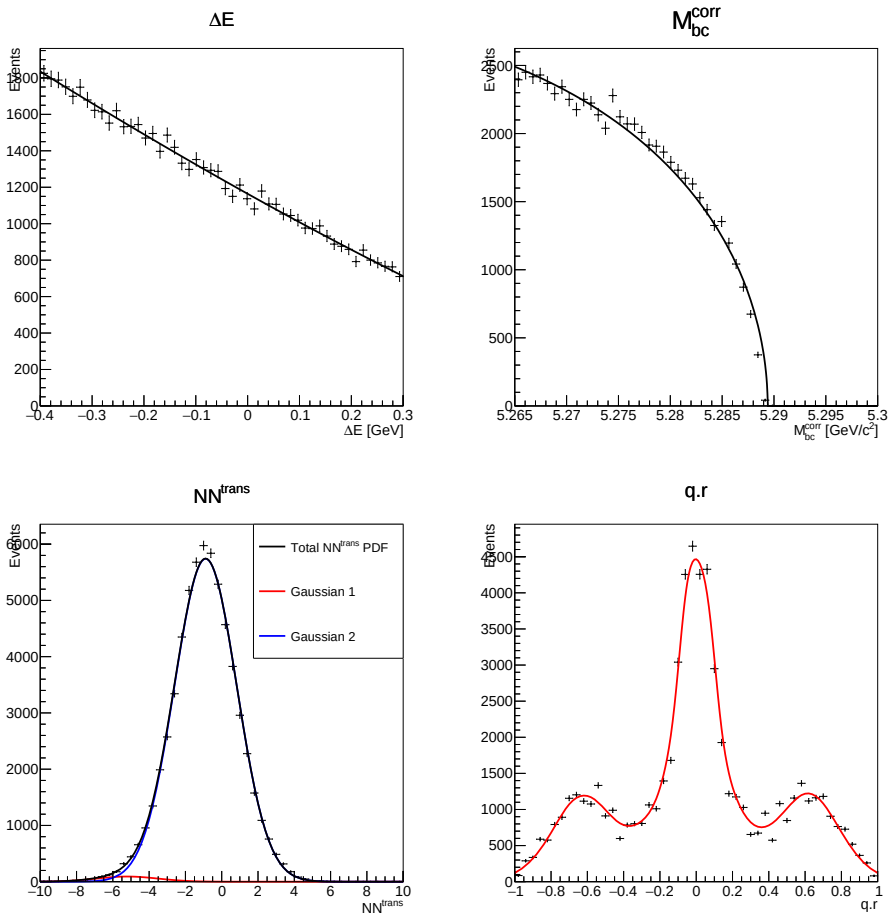

Figure 6.12: Showing the continuum one-dimensional PDFs for ΔE (top left), M_{bc}^{corr} (top right) and NN^{trans} (bottom left) and $q.r$ (bottom right).

functions with mirroring at both edges, and smoothing factors of $\rho = 1$. The one-dimensional PDFs for charged and mixed rare backgrounds are shown in Figures 6.13 and 6.14 respectively.

The scatter plots for every pair of dimensions, and their correlations (for both charged and mixed rare backgrounds) are shown in Appendix B.1.

The signal PDFs (namely in NN^{trans} and $q.r$) are more similar to the rare background PDFs than to the continuum PDFs. Because of this, and the relatively small expected number of rare-background events, the yields for charged and mixed rare background are fixed in the four-dimensional fit. This reduces the statistical uncertainty in $\mathcal{A}_{CP}$ and signal yield measurement, at the cost of introducing systematic uncertainty to the signal yield measurement.

The most common decay modes present in the rare datasets are found (from the Monte-Carlo information), and the expected event numbers (and uncertainties) for these decay modes are calculated (using the branching ratio and uncertainties from [5]). These will be referred to as the 'known rare-backgrounds', see Appendix A for information on the individual decay modes present. The uncertainties on the rest of the rare-backgrounds ('unknown rare backgrounds') is then assumed to be ±40%.

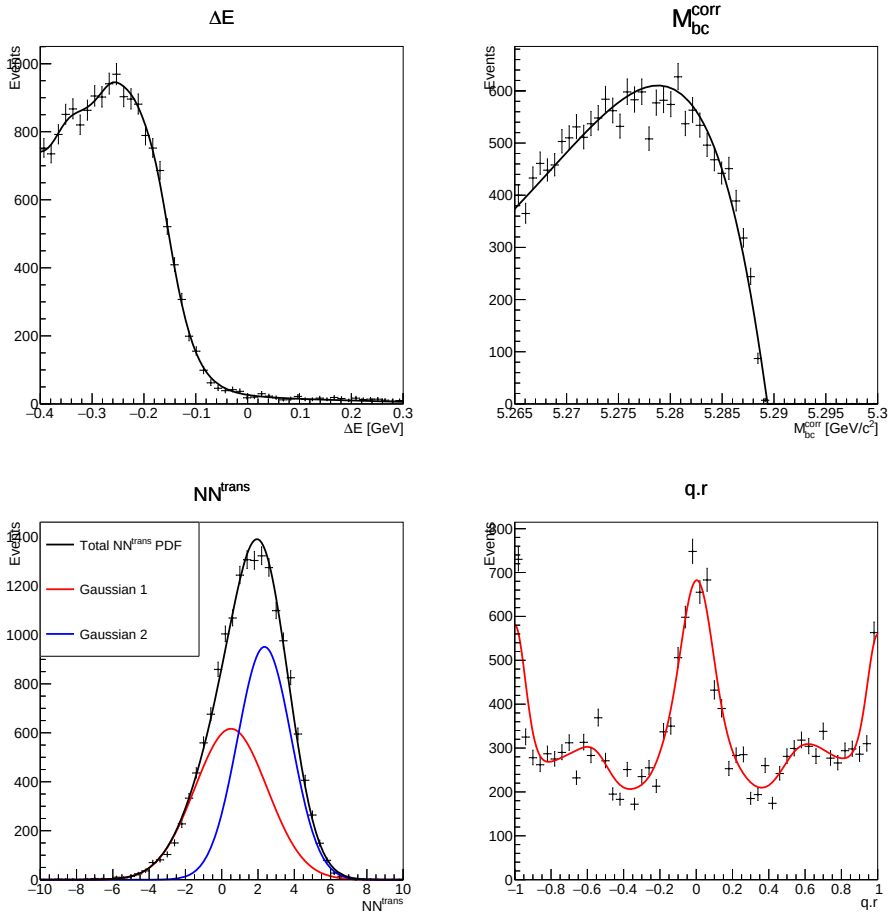

Figure 6.13: Showing the charged rare one-dimensional PDFs for ΔE (top left), M_{bc}^{corr} (top right) and NN^{trans} (bottom left) and $q.r$ (bottom right).

Given a total charged rare event number of 331 ± 3, the known and unknown expected event numbers are:

- Known : 228 ± 46
- Unknown : 103 ± 41

And similarly for mixed rare, with 126 ± 2 events expected, we have:

- Known : 71 ± 21
- Unknown : 55 ± 21

6.2.4 The 4-Dimensional Fit Results

Having combined all of the one-dimensional PDFs, the measurement results are tested tested on toy MC data. This is when we create many datasets comparable to the distributions expected with real data, perform fits to the samples, and measure the fitter performance over the collection of samples. Note that unless otherwise

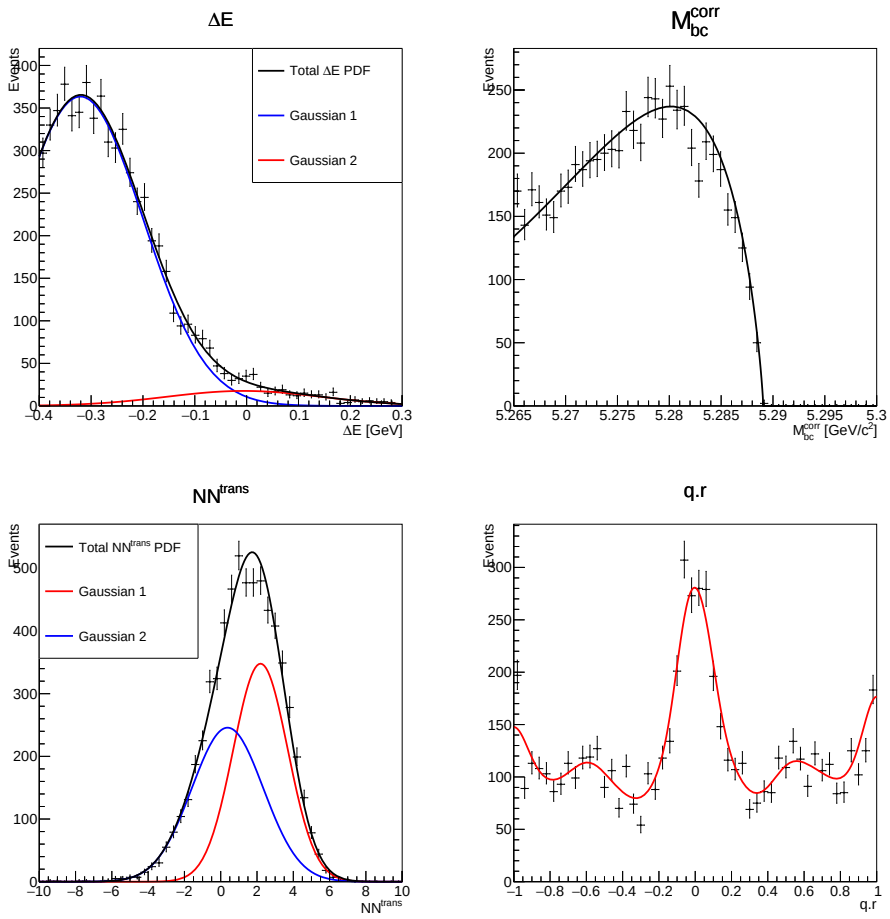

Figure 6.14: Showing the mixed rare one-dimensional PDFs for ΔE (top left), M_{bc}^{corr} (top right) and NN^{trans} (bottom left) and $q.r$ (bottom right).

stated, the datasets mentioned are the *processed* datasets, after the selection on *NN*. Given the large signal validation dataset size compared to the expected number of events (the expected number of events being approximately 0.37% of the full signal validation dataset), the events are randomly sampled from the dataset. Similarly for the rare-backgrounds (each with an expected number of events 2.00% the size of the full rare datasets). This is not the case for continuum as the expected number of continuum events is a third of the continuum dataset. For an individual sample this is acceptable, but when performing tests on the fitter, requiring multiple samples to be taken, each continuum sample will not be statistically independent. Because of this, the continuum events are generated to the four-dimensional continuum PDF.

The number of signal and continuum events (and the $\mathcal{A}_{CP}$) are free parameters in the fit, whereas the number of charged and mixed rare events in the fit are fixed to be 331 and 126 respectively. The sample sizes for each channel are randomly selected from a Poisson distribution centred around the expected number of events in that channel.

The results of the four-dimensional fit on a single MC sample with $\mathcal{A}_{CP} = 0$ are shown in Figure 6.15. Figure 6.16 shows the projection plots for the same fit,

these are the PDFs and data in each dimension, where selections have been placed in the other dimensions in order to reduce the backgrounds visible in the plots. The projection selections are only placed on M_{bc}^{corr} and NN^{trans} (due to difficulty in implementing projections when selecting ranges on kernel density estimation PDFs). The projection selection ranges are:

- ΔE:
  - $5.273\ \text{GeV}c^{-2} < M_{bc}^{corr} < 5.283\ \text{GeV}c^{-2}$
  - $0.0 < NN^{trans} < 8.0$
- M_{bc}^{corr}:
  - $2.0 < NN^{trans} < 8.0$
- NN^{trans}:
  - $5.278\ \text{GeV}c^{-2} < M_{bc}^{corr} < 5.279\ \text{GeV}c^{-2}$
- $q.r$:
  - $5.273\ \text{GeV}c^{-2} < M_{bc}^{corr} < 5.283\ \text{GeV}c^{-2}$
  - $0.0 < NN^{trans} < 6.0$

Additionally, the four-dimensional fit and projections for data samples with $\mathcal{A}_{CP} = +1$ and $\mathcal{A}_{CP} = -1$ are shown in Figure 6.17.

The impact that fixing the rare-background yields has on the signal yield measurement is investigated. The Poisson mean for the known charged rare event number (that will be sampled from the dataset) is raised to the upper end of its uncertainty, and the fit performed 500 times on samples with this raised charged rare event number. This is repeated at the low end of the known charged rare event number, and the systematic uncertainty being half of the difference between the measured signal yield means. This process is repeated for the unknown charged rare event numbers, and for the known and unknown mixed rare event numbers. It is found that the process of fixing the rare background event number in the fitter introduces a systematic uncertainty in the measured signal yield of only ± 10.0 events. See Appendix A for more details.

Signal Yield Measurement

The fit is run on one-thousand data samples with a signal event number selected from a Poisson distribution with a mean equalling the expected signal event number; 1052. The mean statistical uncertainty (in the measured signal yield) returned by the fitter is 47.35 ± 0.03 (with the standard deviation in in this measurement at 0.80 ± 0.02), see Figure 6.18. Figure 6.19 shows the measured signal event number over the thousand fits. The mean of the measured number of signal events is 1058.2 ± 1.5, with a standard deviation of 46.7 ± 1.0.

As can be seen from the signal yield measurements, the fitter signal yield uncertainty is accurate, being within the error range of the standard deviation of the measured signal yield results. The mean of the signal yield measurements of 1058.2 ± 1.5

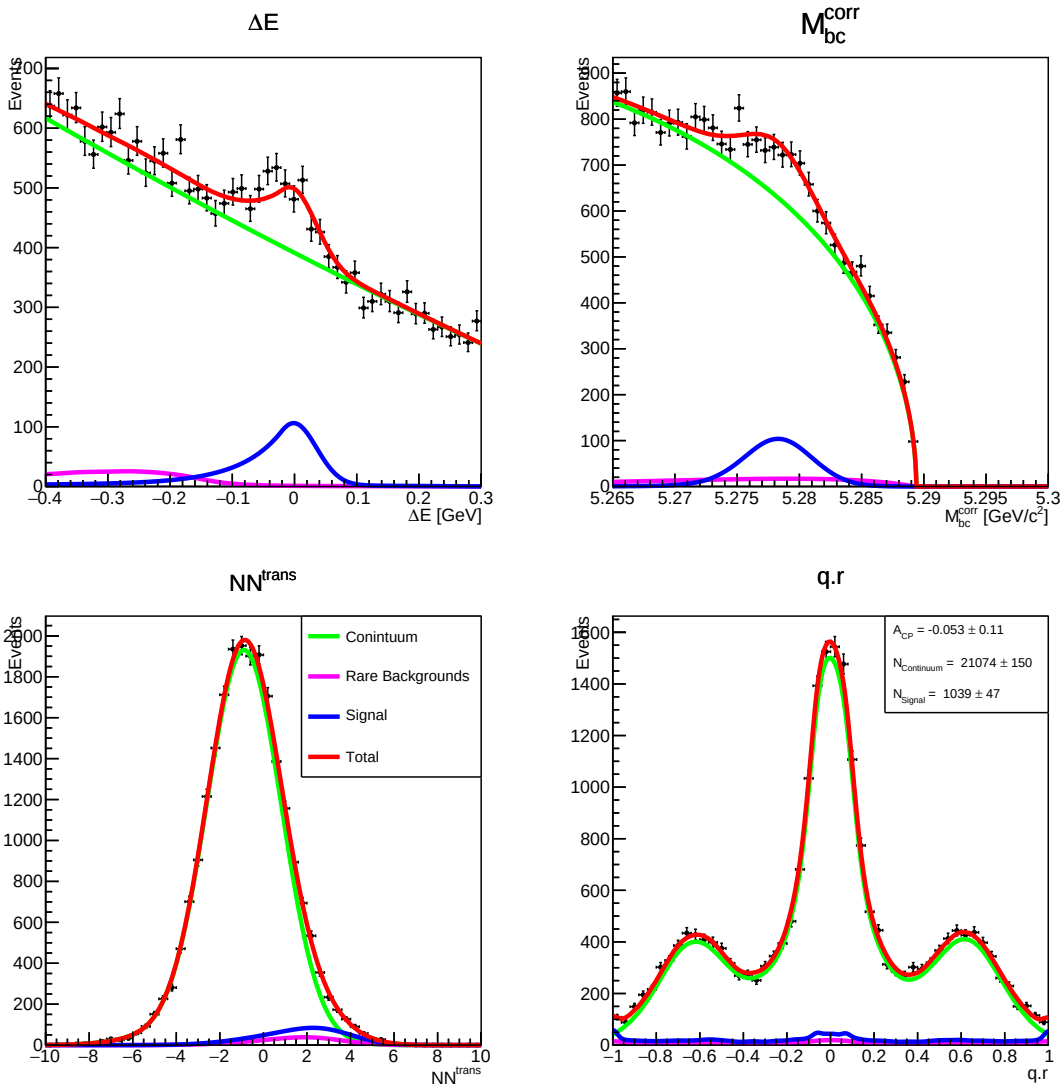

Figure 6.15: Showing a 4-D fit to a sample with $\mathcal{A}_{CP} = 0$.

is overestimating the signal yield (with Poisson mean of 1052). To study the quality of the model, the 'pull' value for each of the thousand fits is calculated. The signal yield pull is defined as:

$$\text{Pull}_{N_{signal}} = \frac{N_{signal} - N^{exp}_{signal}}{\epsilon_{N_{signal}}} \tag{6.10}$$

Where N_{signal} is the measured signal yield in that fit. N^{exp}_{signal} is the expected signal yield, the mean of the Poisson distribution from which the input signal event numbers are selected. $\epsilon_{N_{signal}}$ is the statistical uncertainty in the signal yield measurement (returned by the fitter) for that fit. Then by finding the mean and standard-deviation of the distribution of one-thousand pulls, the quality of the model can be verified. Clearly a perfect model would see a mean-pull of zero (with the mean of the N_{signal} measurements being equal to the mean of the input signal event numbers),

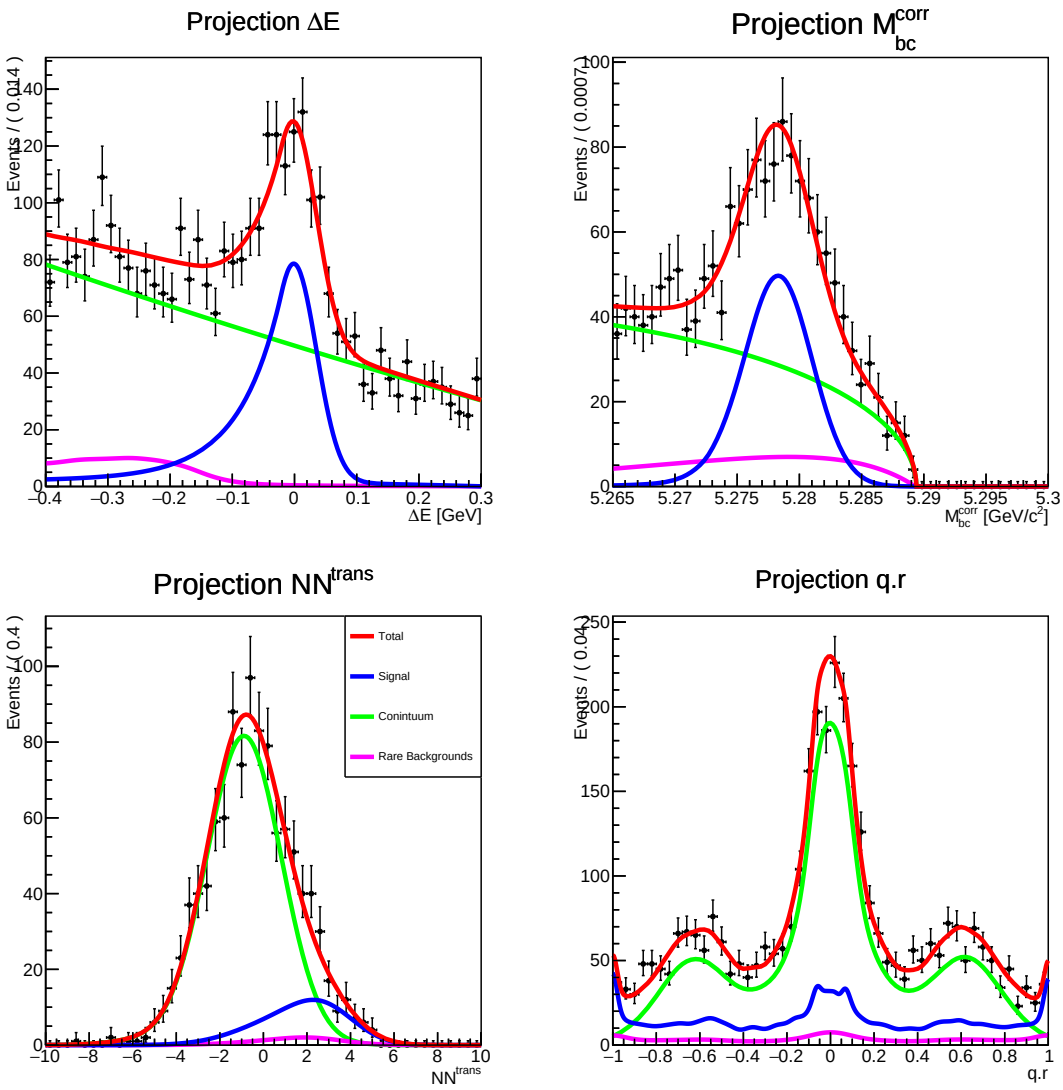

Figure 6.16: Showing the projection plots corresponding to the fit in 6.15.

and a pull standard-deviation of one (meaning that the fitter error is equal to the standard deviation of N_{signal}). The pull distribution for the signal yield measurement is shown in Figure 6.20. The mean of the pulls is 0.12 ± 0.03, being small, clearly the fit is not bad, but the positive value (and zero not being within the error range) shows that the model consistently slightly overestimates the signal yield. The standard-deviation of the pull distribution is 0.98 ± 0.02, showing that the mean of the signal yield uncertainties is slightly larger than the standard-deviations of this measurement, but 1.0 being within the error range, the fitter quoted uncertainty is accurate.

A final test of how well the model performs when measuring the signal yields is to measure how its performance varies for different input signal yields. The input signal event numbers are varied (the mean of the Poisson distribution from which the input signal event numbers are sampled) from 75% (789 events) of the expected

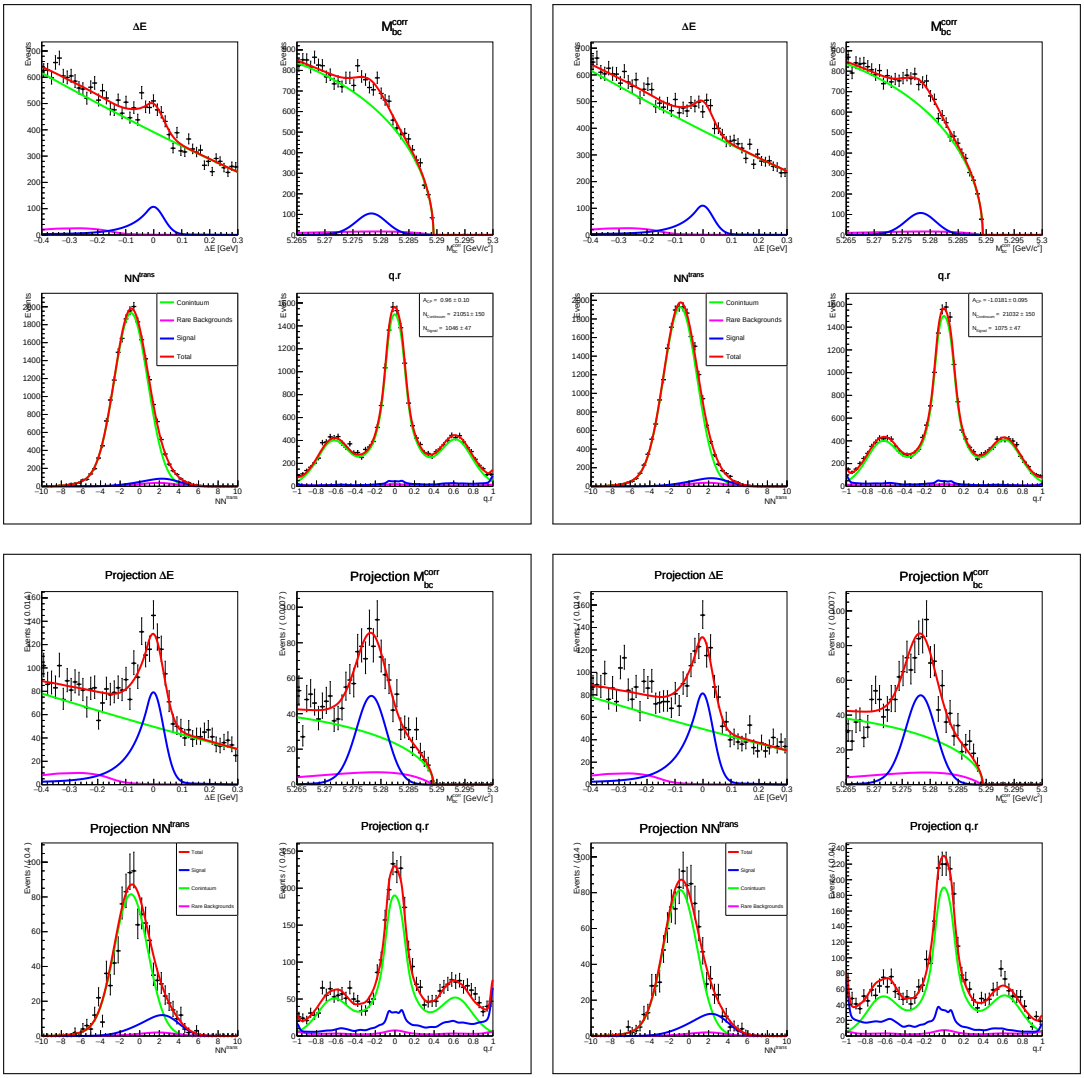

Figure 6.17: Showing example 4-D fits (top row) and projection plots (bottom row) to data samples with $\mathcal{A}_{CP} = +1$ (left column) and $\mathcal{A}_{CP} = -1$ (right column).

signal yield, to 125% (1315 events) in steps off 5%. Five-hundred samples are taken for each Poisson mean, and the how the measured values vary with input value is investigated. A perfect model would have a y-intercept of zero and a gradient of one. The gradient is 0.999 ± 0.005, within the error range of one, clearly the model scales equally with the change in input, accurately measuring a change in signal yield. The y-intercept is 8.0 ± 4.8, and given that the gradient is (very close to being) one, shows that the model has a consistent bias. This consistent bias in the signal yield measurement can also be seen in Figure 6.21, where the 1st order polynomial fit to the data is (almost) parallel to, and consistently above $y = x$.

Given the good model results when measuring signal yield, the slight bias can be corrected by simply subtracting 6.2 (the difference between the mean of the measured signal yields and the expected number of signal events) from the measured signal yield.

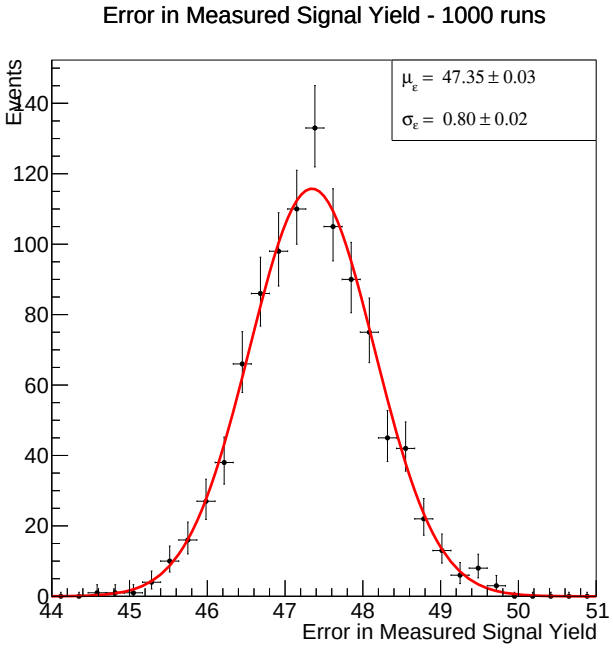

Figure 6.18: Showing the statistical uncertainty in the measured signal yield distribution over one-thousand fits.

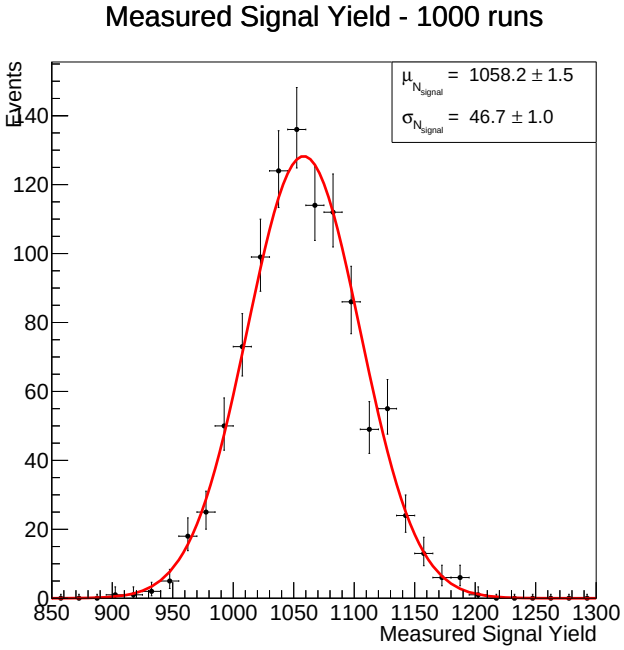

Figure 6.19: Showing the measured signal yield distribution over one-thousand fits.

$\mathcal{A}_{CP}$ Measurement

Similarly as is done for the signal yield measurements, the quality of the model when measuring the $\mathcal{A}_{CP}$ is investigated. The fit is run one-thousand times on samples

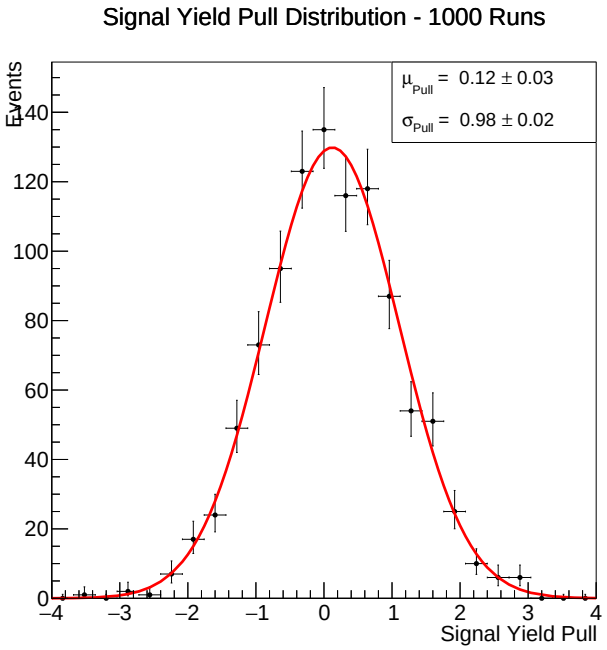

Figure 6.20: Showing the pull distribution in signal yield measurement, over one-thousand runs.

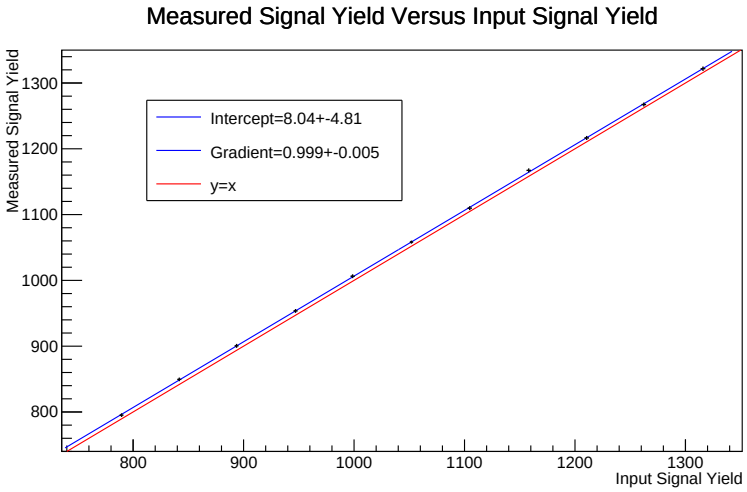

Figure 6.21: Showing the means of the measured signal yields plotted against the corresponding means of the input signal yields.

with $\mathcal{A}_{CP} = 0$. The mean error on $\mathcal{A}_{CP}$ given by the fitter (over a thousand runs) is 0.1112 ± 0.0001 (and the standard-deviation on this error is 0.00354 ± 0.00008), see Figure 6.22. The mean of the measured $\mathcal{A}_{CP}$ values is 0.011 ± 0.003 and the standard-deviation in the $\mathcal{A}_{CP}$ measurements is 0.109 ± 0.002, see Figure 6.23. The mean of the $\mathcal{A}_{CP}$ results is slightly above zero but not significantly. Additionally, the fitter quoted uncertainty is slightly overestimating the uncertainty.

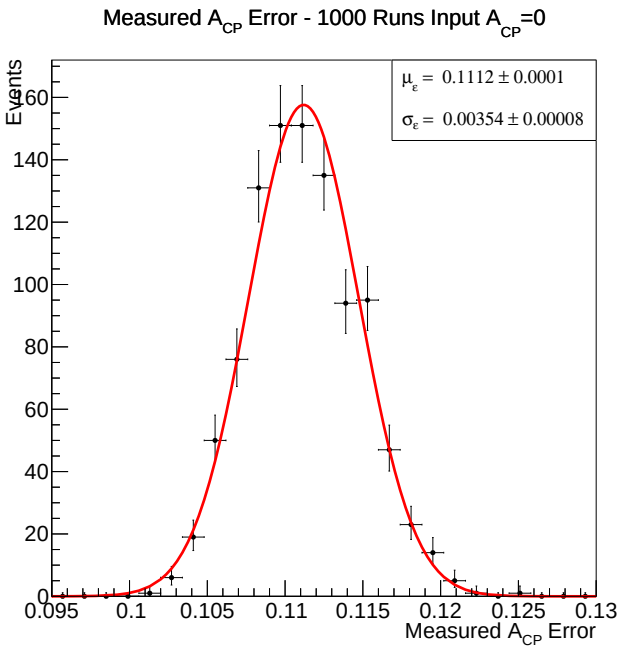

Figure 6.22: Showing the error in the measured $\mathcal{A}_{CP}$ over one-thousand runs.

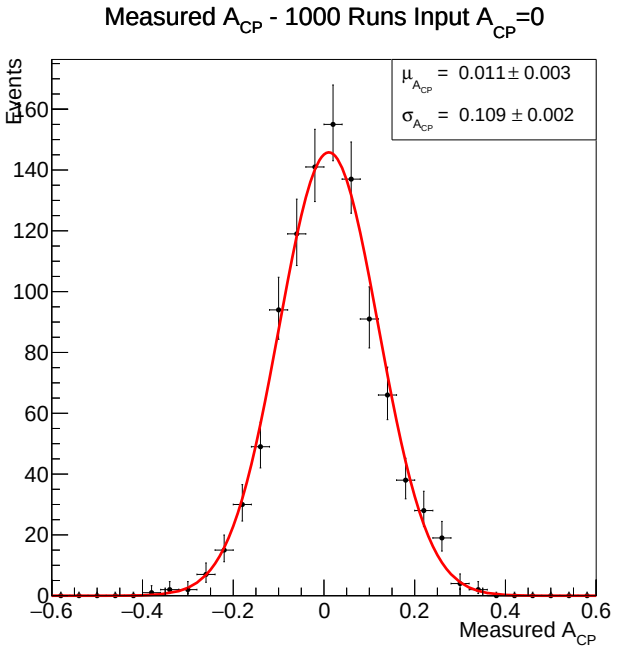

Figure 6.23: Showing the measured $\mathcal{A}_{CP}$ over one-thousand runs.

To verify the quality of the model when measuring $\mathcal{A}_{CP}$, the pull for each of the thousand fits is calculated. The pull for the $\mathcal{A}_{CP}$ measurement of a single fit is

defined as:

$$\text{Pull}_{\mathcal{A}_{CP}} = \frac{\mathcal{A}_{CP} - \mathcal{A}^{exp}_{CP}}{\epsilon_{\mathcal{A}_{CP}}} \tag{6.11}$$

Where $\mathcal{A}_{CP}$ is the measured $\mathcal{A}_{CP}$ for that fit, $\mathcal{A}^{exp}_{CP}$ is the expected (i.e. the input) $\mathcal{A}_{CP}$, in this case zero. $\epsilon_{\mathcal{A}_{CP}}$ is the fitter error on the measured $\mathcal{A}_{CP}$ for that run. The pull distribution for the $\mathcal{A}_{CP}$ measurements is shown in Figure 6.24. The pull distribution has a mean of 0.10±0.03 and a standard deviation of 0.98±0.02. Clearly, although the measured value is still good, the $\mathcal{A}_{CP}$ is being slightly overestimated.

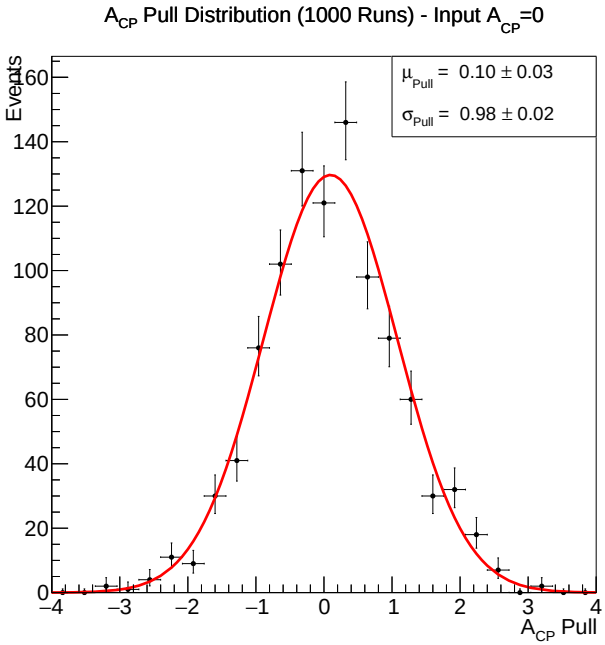

Figure 6.24: Showing the distribution in the pulls for $\mathcal{A}_{CP}$ over one-thousand runs.

Finally the model performance for a range of input $\mathcal{A}_{CP}$ values is investigated. The fit is run over 500 samples with signal $\mathcal{A}_{CP} = \pm 1$, and at $\mathcal{A}_{CP}$ from -0.5 to 0.5 in steps of 0.1. The plot of measured (mean over 500 runs) $\mathcal{A}_{CP}$ against input $\mathcal{A}_{CP}$ is shown in Figure 6.25. The y-intercept of 0.0083 ± 0.0013 shows that the $\mathcal{A}_{CP}$ is being overestimated. Additionally the gradient of 0.982 ± 0.003 (being 6 times the error away from one) is also clearly away from $y = x$. Although there are clear issues with the model when measuring the $\mathcal{A}_{CP}$, the discrepancies are small, and the model could be manually adjusted (a displacement and scaling of the gradient of the first-order polynomial in the signal $q.r$ PDF).

The latest belle measurement for the $\mathcal{A}_{CP}$ had a statistical uncertainty of ±0.13, so statistical uncertainty from this model of 0.111 ± 0.004 is a definite improvement. Even so, the ratio of signal to continuum is a limiting factor, and given the latest state of the art software packages for machine learning, there is room for further reduction in the continuum background and this statistical uncertainty could be further reduced.

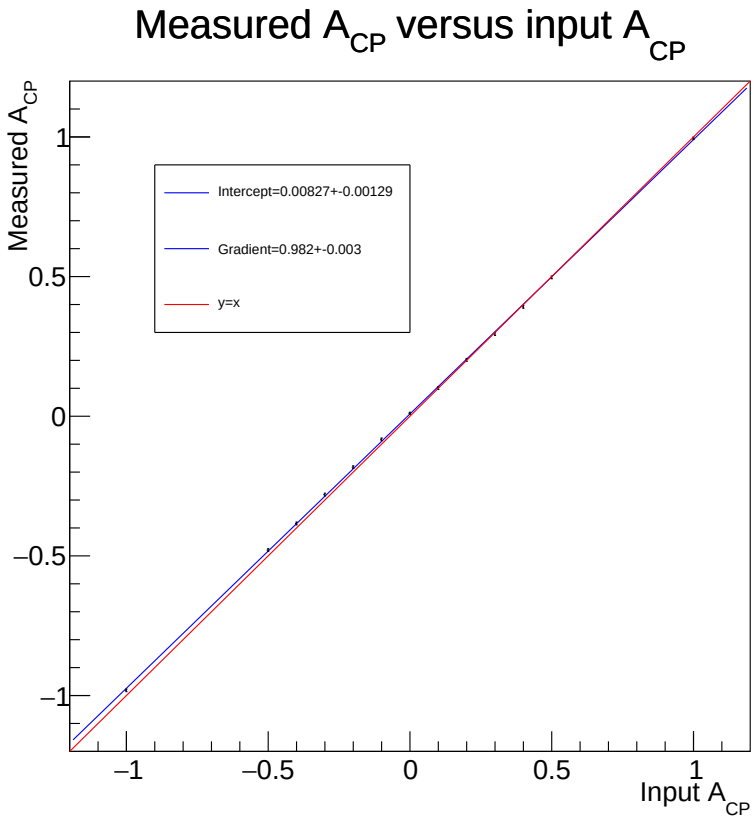

Figure 6.25: Showing the mean of the measured $\mathcal{A}_{CP}$ values against the $\mathcal{A}_{CP}$ of the data samples.

7|Continuum Suppression With TensorFlow

In order to reduce the statistical uncertainty on $\mathcal{A}_{CP}$, we investigate whether we can improve continuum suppression. There are a handful of state of the art software packages for machine learning and neural networks including Torch[37] and Theano[38]. In this study, the use of TensorFlow[39] was investigated.

The procedure follows closely that in 6.1, variable names and abbreviations will not be redefined. The same selection criteria of $5.265\ \mathrm{GeV}c^{-2} < M_{bc}^{corr} < 5.3\ \mathrm{GeV}c^{-2}$ and $-0.4\ \mathrm{GeV} < \Delta E < 0.3\ \mathrm{GeV}$ is assumed unless otherwise stated.

7.1 TensorFlow

TensorFlow is an open source software package for machine learning maintained by Google, with the interface implemented in Python and optionally running on GPUs (which when dealing with large matrix operations can see an immense speed improvement compared to running on CPUs).

It allows the entire architecture of the neural network to be built from the ground up, with various algorithms vital to training neural networks either pre-implemented, or implemented by the user.

Before building and training the network, all of the input data must be pre-processed. All nineteen training variables (see 4.3) are transformed to be within the range ± 1 by implementing equal frequency binning. This is done by first finding the bin edges in the un-transformed variable such that each bin (of variable width) has the same number of entries. This is done with the combined signal and continuum training datasets. The transformed variable between -1 and 1 has equal bin widths, and a one-to-one correspondence to each bin in the un-transformed variable is implemented such that all events in the first un-transformed bin are placed in the first bin in the transformed variable, and similarly for all bins. In this way, the transformed variable is uniform (when looking at the combined signal and continuum training sets). Signal and continuum individually have different distributions, and all information and correlations are preserved. This is done for each of the nineteen kinematic variables. Figure 7.1 shows the equal frequency binning procedure as applied to a mock variable where signal and continuum are each a Gaussian, with different means and standard-deviations. The bin edges in the un-transformed variables calculated from the training datasets are then used when processing all remaining data. As when training the NeuroBayes neural network, 125,000 signal

and 125,000 continuum events are used in the training (and in this case calculating the bin-edges for pre-processing). Five-hundred bins are used when transforming the kinematic variables (this process is only performed with two-hundred bins in 7.1).

The events are shuffled between every training epoch to ensure that no two batches are the same, and that each event appears in random order. This is done firstly so that different combinations of events make up the batches, preventing particular event combinations impacting the training (so that repeating the training process won't see very different results). Secondly this changes the order in which events are trained over, as this could bias the training.

The architecture of the network is highly variable. The number of hidden layers, nodes per layer and activation functions are all adjustable, as well as options relating to the training and optimisation of each network such as batch size, learning rate, number of epochs, loss function and choice of training algorithm. All of these options (and many more) will be referred to as the hyper-parameters of the neural network. The trainable-variables are the parameters that are adjusted in the training process, namely the weights between the nodes. The weights in a given layer are initialised randomly to a uniform distribution in the range:

$$\pm\frac{\sqrt{6}}{\sqrt{n_{sum}}} \tag{7.1}$$

Where n_{sum} is the sum of the number of nodes in both layers in which the weights connect. See [40] for more details.

The loss function used for training is the cross-entropy (see 5.2). The loss function of the output node is set as tanh, and then the output is transformed to ensure that NN is between 0 and 1.

The neural network was trained using the Adam algorithm[41], shown to perform better (faster and settles to a better minimum) than other gradient descent algorithms. It uses a 'momentum' to reduce the rate at which the search oscillates in trainable-variable space. Often, the learning rate is reduced manually (i.e. based on the current epoch number) so that as the algorithm converges towards the loss-function minimum, it takes smaller steps. In the Adam algorithm, the 'momentum' is reduced (according to speed of learning) effectively reducing the learning rate in an optimised manner, optimising the convergence.

More complex options to aid in training and to prevent overtraining were also implemented. These include using batch-normalisation (transforming the data at each node according the mean and variance in the batch), L2-normalisation (adding a term proportional to the square of the weights to the loss function to favour larger weights only if they bring a clear improvement to the classifying ability of the network) and dropout (randomly dropping nodes - setting the weights to zero - during training to force the network to learn as many relationships as possible).

As training can be slow, one of the biggest hurdles in training a good neural network is the choice of hyper-parameters. Even with a fast training neural network, selecting the best configuration from the huge hyper-parameter space is a time consuming activity. Grid-searching (selecting multiple options for each hyper-parameter, and training the network on every possible configuration) wastes a lot of

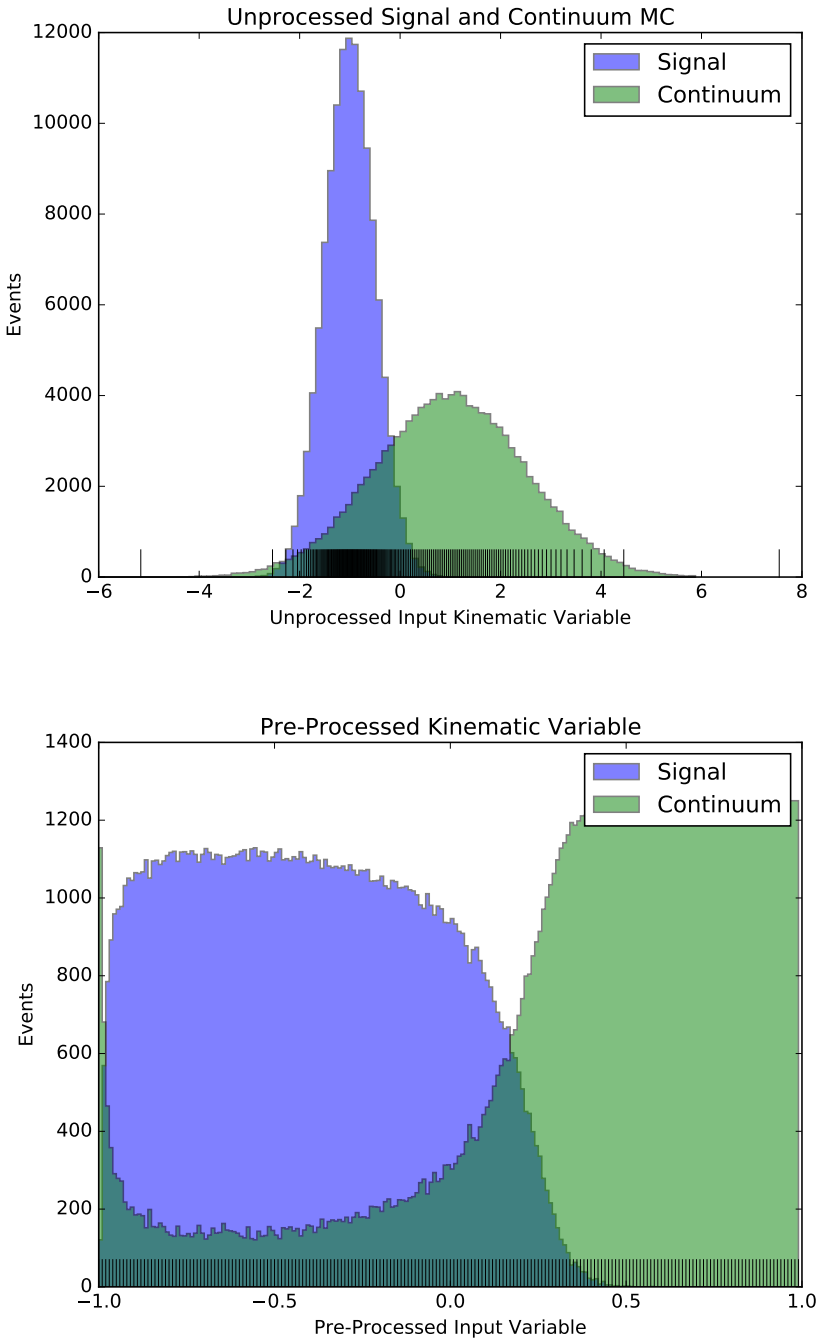

Figure 7.1: Showing the equal frequency binning applied to a mock variable where signal and continuum both have Gaussian distributions (top). The vertical black lines correspond the bin edges such that the sum of the signal and continuum event numbers are equal for every bin. The bins in the transformed distribution (bottom) again have the same number of entries per bin, but are also transformed to have the same bin-width.

training time on 'bad' configurations. More advanced algorithms based on Bayesian optimisation are very popular.

In this study the Hyperband algorithm[42] is used, shown to converge to a good hyper-parameter set up much faster than other optimisation algorithms. Hyperband first assumes that given a set of neural networks (with different hyper-parameter configurations), the networks that perform better after a few training epochs are more likely to perform better when fully trained. This is obviously not always the case (slower converging networks often settle closer to the loss-function minimum), but running all options for the full number of epochs is computationally expensive. Taking account of this, a large sample of neural networks are trained for a small number of epochs, and a small sample for a large number of epochs. For example if the maximum epoch number is 200, then 200 different hyper-parameter configurations are randomly selected and the neural networks trained for one-epoch each, one configuration is trained for the full number of epochs, and a range in between with a decreasing number of neural network configurations as the epoch number increases. The best performing networks are selected and the process begins again, with the minimum number of training epochs increased. This process is repeated (selecting the best configurations and increasing the minimum number of training epochs) until a handful of neural networks are trained for the full number of epochs. Hyperband cannot optimise the learning rate (it will have a preference for the sub-optimal larger learning rates), so a learning rate of 10^{-4} is chosen (known to have the potential for good convergence from previous investigations). Ideally Hyperband would be run for a range of different learning rates. This process can reduce the time taken to find the best configuration from weeks to days.

Once every five epochs (or every epoch when total number of epochs is less than 10) the performance of the neural network is evaluated. This is done by averaging the losses of the signal and continuum testing datasets. Figure 7.2 shows the loss value for training and testing datasets against the number of epochs. The performance of the network is not validated on the training set as Hyperband would select a configuration that would lead to massive over-training. Additionally, to increase the speed of the hyper-parameter optimisation, the best test-loss (during the training of a given network) is saved, and training stopped if the test-loss does not decrease over 50 epochs. This best test-loss is used to define the performance of the network, even if the test-loss diverged after more training. Because this would still lead to a configuration that would show preference to the testing dataset compared to what would actually be expected (due to the large datasets this shouldn't have much impact), the validation datasets are needed.

7.1.1 Analysis of the Neural Network Performance

Once the best configuration is found, the neural network is trained with this configuration, only saving the trainable-variables at the epoch giving the best test-loss (in effect implemented early stopping, and not needing to consider future over-training, and allowing for a large maximum number of epochs without worrying about the impact on the network). Finally the network performance is tested on the validation datasets.

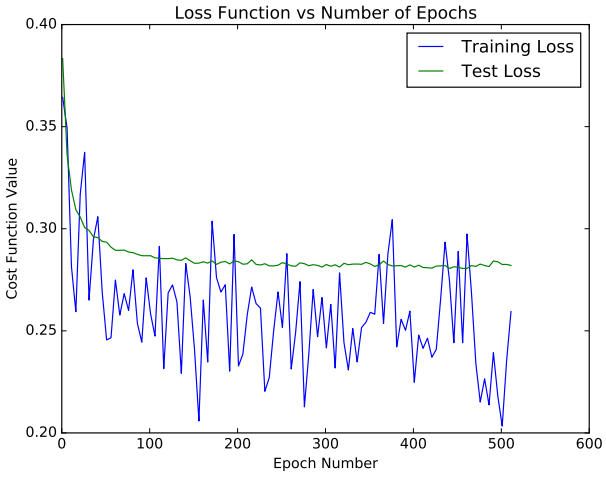

Figure 7.2: Showing how the loss on the training (blue) and testing (green) varies as the training proceeds. Note that the test loss is a lot less noisy than the training loss as the entire testing dataset was used when calculating the test loss. It can be seen that the average training loss is significantly lower than the average test loss.

The hyper-parameter configuration for the best performing network is as follows:

- A maximum number of epochs of 600.
- 50 events per batch.
- A Learning rate of 0.0001.
- Six hidden layers.
- 47 nodes per hidden layer.
- Exponential linear unit activation function.
- No batch normalisation.
- A dropout chance of 0.007 and only applying to every hidden layer (in effect not applying dropout).
- No L2 regularisation.

This trained neural network is then applied to the data. Note that *NN* will be used to refer to the output of *this* neural network, unless otherwise stated. The neural network outputs for signal and continuum are shown in Figure 7.3. This clearly shows an improved classifying ability over the NeuroBayes neural network (see Figure 6.1).

As in Chapter 6, the signal validation dataset contains both the correctly and incorrectly ($\sim 11.2\%$ of the signal validation dataset) reconstructed B^0-mesons, for which the *NN* distributions are plotted in Figure 7.4. As can be seen, that while

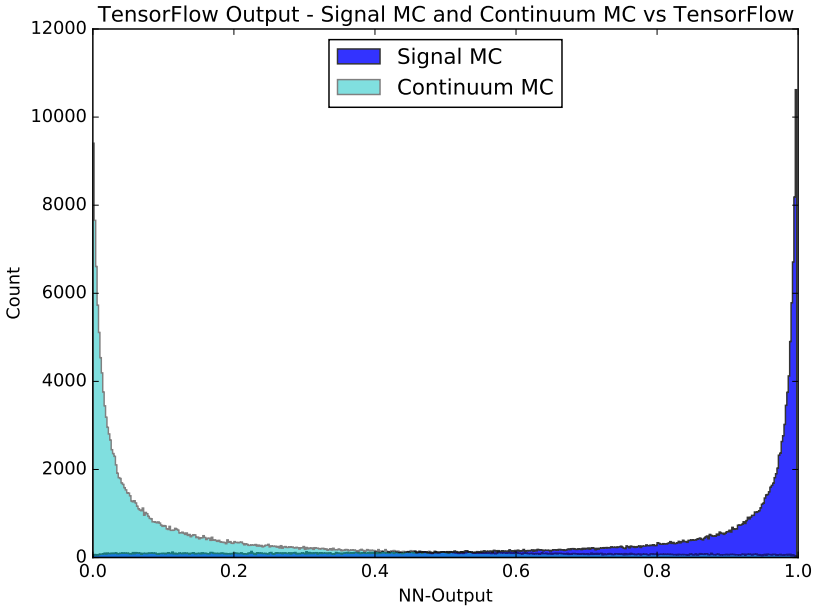

Figure 7.3: Showing the signal and continuum *NN* distributions (equal numbers) for the trained TensorFlow network.

good, classifying ability is slightly worse for the events with wrongly reconstructed B^0-mesons (to be expected as the training was performed on only the events with correctly reconstructed B^0-mesons). As the performance of the neural network isn't greatly different in both cases, and that the events with wrongly reconstructed B^0-mesons are a small proportion of the signal validation dataset, they are not analysed separately from here on.

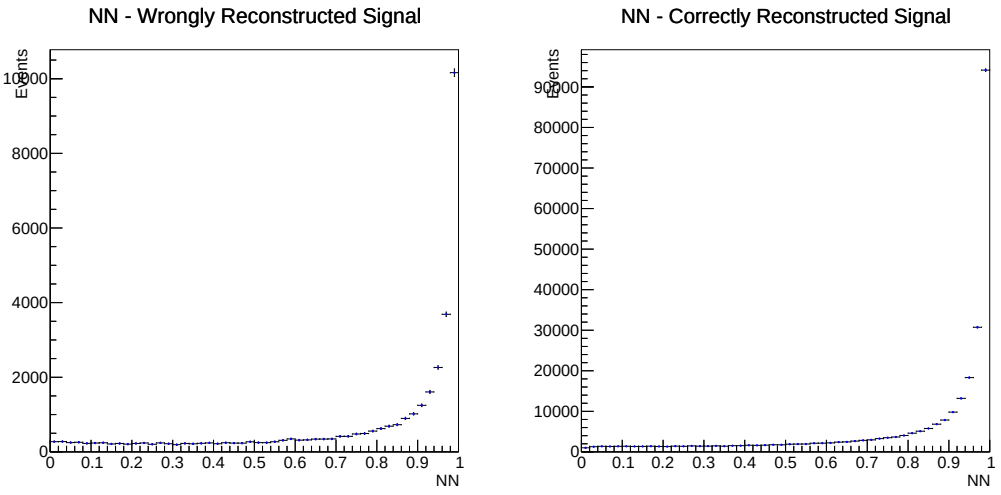

Figure 7.4: Showing the signal *NN* distributions for the incorrectly (left) and correctly (right) reconstructed B^0-mesons.

The signal and continuum MC *NN* distributions are plotted with the expected number of events, along with the FOM distribution, see Figure 7.5. The best FOM is found to be 17.3 ± 0.4, which shows a clear improvement over NeuroBayes, with a best FOM of 13.2 ± 0.4.

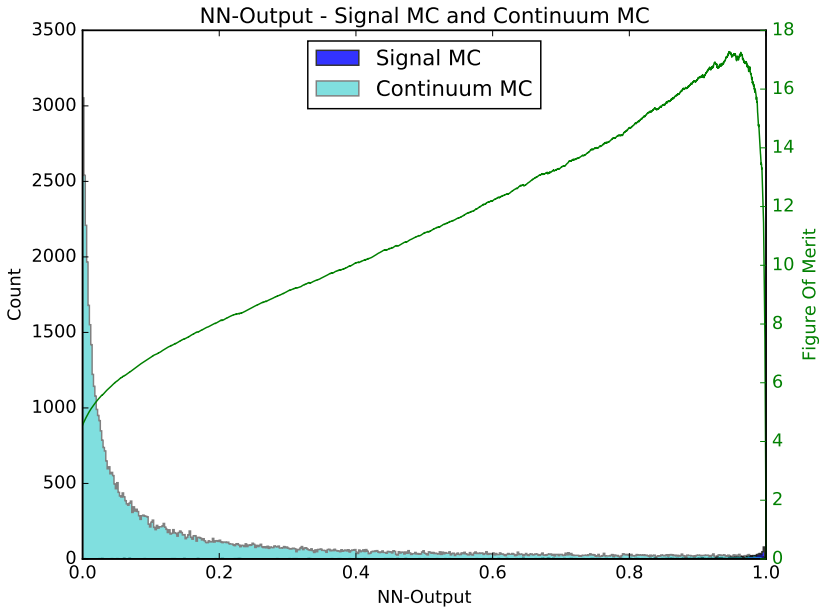

Figure 7.5: Showing the signal and continuum *NN* distributions with the expected numbers of events, and the FOM (green).

The ROC curve (for signal and continuum MC)is shown in Figure 7.6, giving an AUC of 0.947, confirming that this neural network has a much better classifying performance than the NeuroBayes neural network (with an AUC of 0.909).

With this neural network, a NN_{cut} value chosen to keep 13.00% of continuum, leaves 88.11% of signal (contrast with 79.01% from the NeuroBayes neural network). A value of NN_{cut} chosen to keep 70.20% of signal, leaves 3.35% of continuum remaining (contrast with 7.65% from the NeuroBayes neural network). These results show that (depending on the NN_{cut} choices), the continuum background (for a given signal efficiency) could be halved by using an optimised neural network in TensorFlow as compared to an un-optimised NeuroBayes neural network.

As this neural network is trained entirely on Monte Carlo data, the real off-resonance data was processed by this network to validate that it performs similarly to the continuum MC. The signal and off-resonance *NN* distributions (with an equal number of events) is shown in Figure 7.7. A NN_{cut} value chosen to keep 13.00% of off-resonance leaves 85.85% of signal, this performance is slightly worse than with the continuum MC data but still good, and much better than the NeuroBayes result (74.38% signal when keeping 13.00% of off-resonance data). Selecting the NN_{cut} that keeps 70.20% of signal leaves 4.55% of off-resonance remaining, again showing a worse result than with continuum MC, but less than half than is given by NeuroBayes (10.08%).

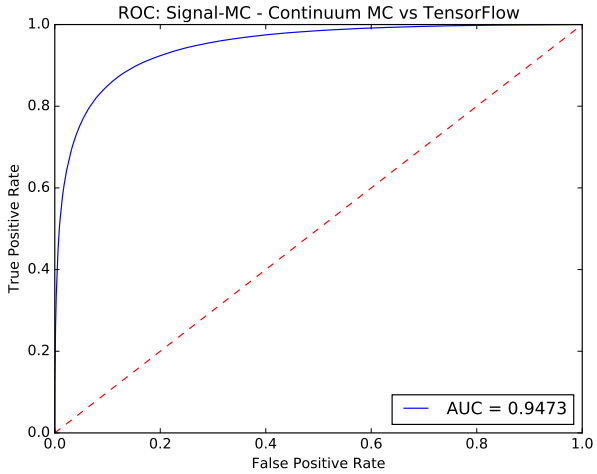

Figure 7.6: Showing the ROC curve of signal and continuum MC, with an AUC of 0.947.

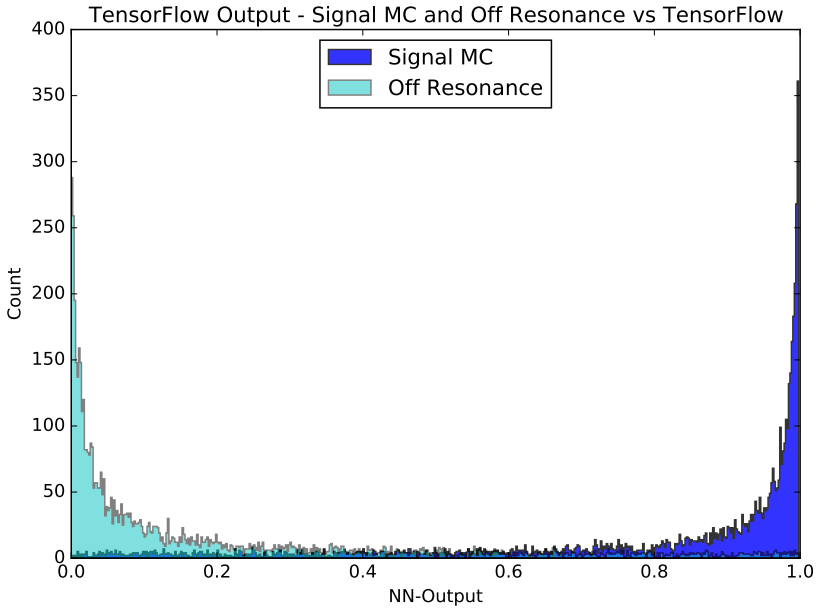

Figure 7.7: Showing the *NN* distributions for signal and off-resonance data, in equal numbers. Note that the distributions are noisy due to the much smaller sample size.

The ROC for signal MC and off-resonance is shown in Figure 7.8. The AUC is 0.938, again showing a clear improvement over the NeuroBayes neural network AUC (of signal and off-resonance data) of 0.891

The *NN* distributions for the charged and mixed rare backgrounds are shown in Figure 7.9. The distributions do not just peak at 1(as in the NeuroBayes case, see Figure 6.7), but also at 0. This means that for any choice of NN_{cut}, the number of rare background events that make it through all of the selections should be lower in

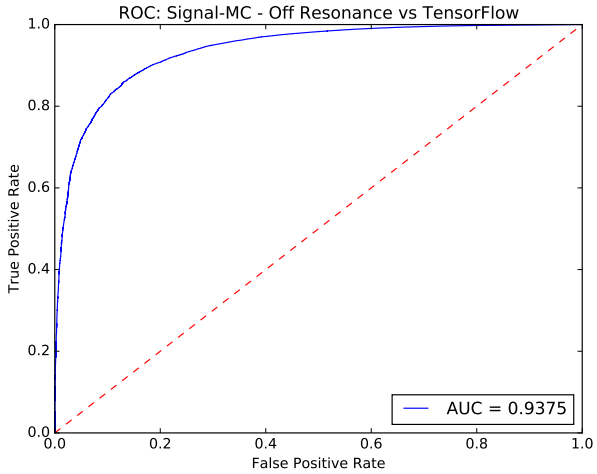

Figure 7.8: Showing the ROC curve for signal and off-resonance data, giving an AUC of 0.938.

the TensorFlow case (the differences were not analysed in detail, but are clear from the *NN* distributions).

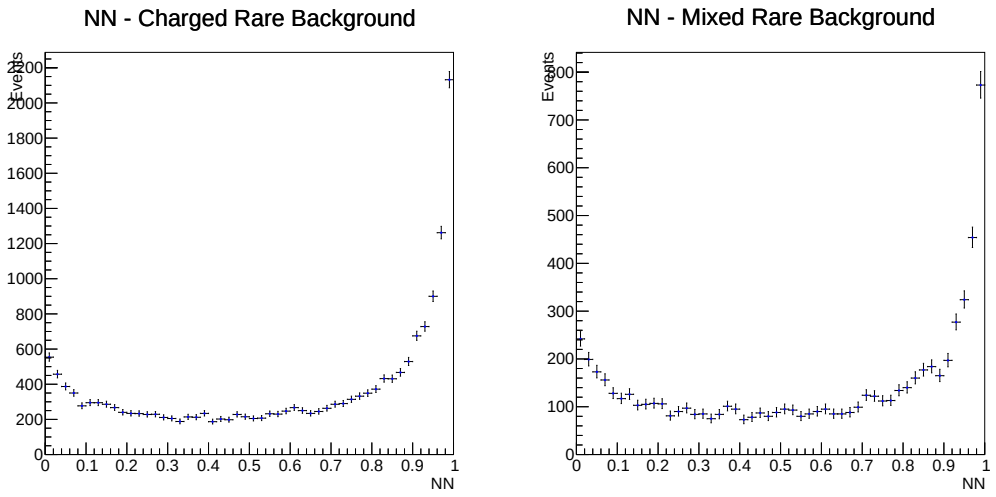

Figure 7.9: Showing the *NN* distributions for the charged (left) and mixed (right) rare backgrounds.

This investigation into improved neural networks has shown to give a *very* good improvement in the ability to separate signal and continuum. Sadly, it comes with a cost.

$\Delta E - NN$ Correlations

Investigation into the ΔE distribution at different NN_{cut} values shows that there is a correlation between the two. When looking the continuum ΔE distributions

for different *NN* slices, the distribution is sculpted to be more signal-like as *NN* increases. On the flip side, for a low *NN* the distribution shows the reverse, a trough where signal peaks. Figure 7.10 shows the continuum ΔE distributions at different *NN* ranges. The off-resonance data also shows the same effect. Similarly for the signal distribution (see Figure 7.11), where the ΔE distribution becomes less signal-like as *NN* decreases.

Note that this correlation is minimal in the neural network output for NeuroBayes. Figure 7.12 shows the ΔE distributions for different NeuroBayes neural network output slices, for signal and continuum respectively.

The TensorFlow neural network is learning that there is a relation between the ΔE value, and whether an event is signal or continuum. This can only be the case if there is some correlation between ΔE and the kinematic variables on which the neural network is trained. This was most likely not picked up in the NeuroBayes neural network due to its regularisation and node pruning algorithms.

An investigation into the correlations between ΔE and the input kinematic variables indeed finds some large correlations. The scatter plots of ΔE with the kinematic variables, and their correlations can be seen in Appendix C. The kinematic variables with the largest correlations were found to be, in decreasing order: R_{20}^{so}, R_{0}^{oo}, R_{2}^{oo} and R_{22}^{so}, with signal MC(continuum MC) correlations of 29.1%(43.0%), 18.2%(27.1%), 12.6%(19.8%), and 13.4%(17.0%) respectively, see Table 7.1.

	Signal ΔE	Continuum ΔE
$\cos(\theta_B)$	4.8%	2.4%
$\cos(\theta_{thrust})$	0.0%	0.0%
ΔZ	0.9%	0.1%
P_t^{sum}	6.8%	3.5%
M_{miss}^2	7.8%	4.6%
R_0^{oo}	18.2%	27.1%
R_1^{oo}	2.6%	3.4%
R_2^{oo}	12.6%	19.8%
R_3^{oo}	0.0%	0.2%
R_4^{oo}	2.1%	2.5%
R_{00}^{so}	9.3%	10.7%
R_{02}^{so}	2.2%	5.1%
R_{04}^{so}	0.4%	3.5%
R_{10}^{so}	4.8%	6.3%
R_{12}^{so}	1.2%	1.8%
R_{14}^{so}	1.3%	0.7%
R_{20}^{so}	29.1%	43.0%
R_{22}^{so}	13.4%	17.0%
R_{24}^{so}	0.0%	0.6%

Table 7.1: The (absolute) correlation percentages between all of the neural network inputs and ΔE for signal and continuum. Calculated for events in the range -0.4 GeV $< \Delta E < 0.2$ GeV (the ΔE selection placed on the training datasets.)

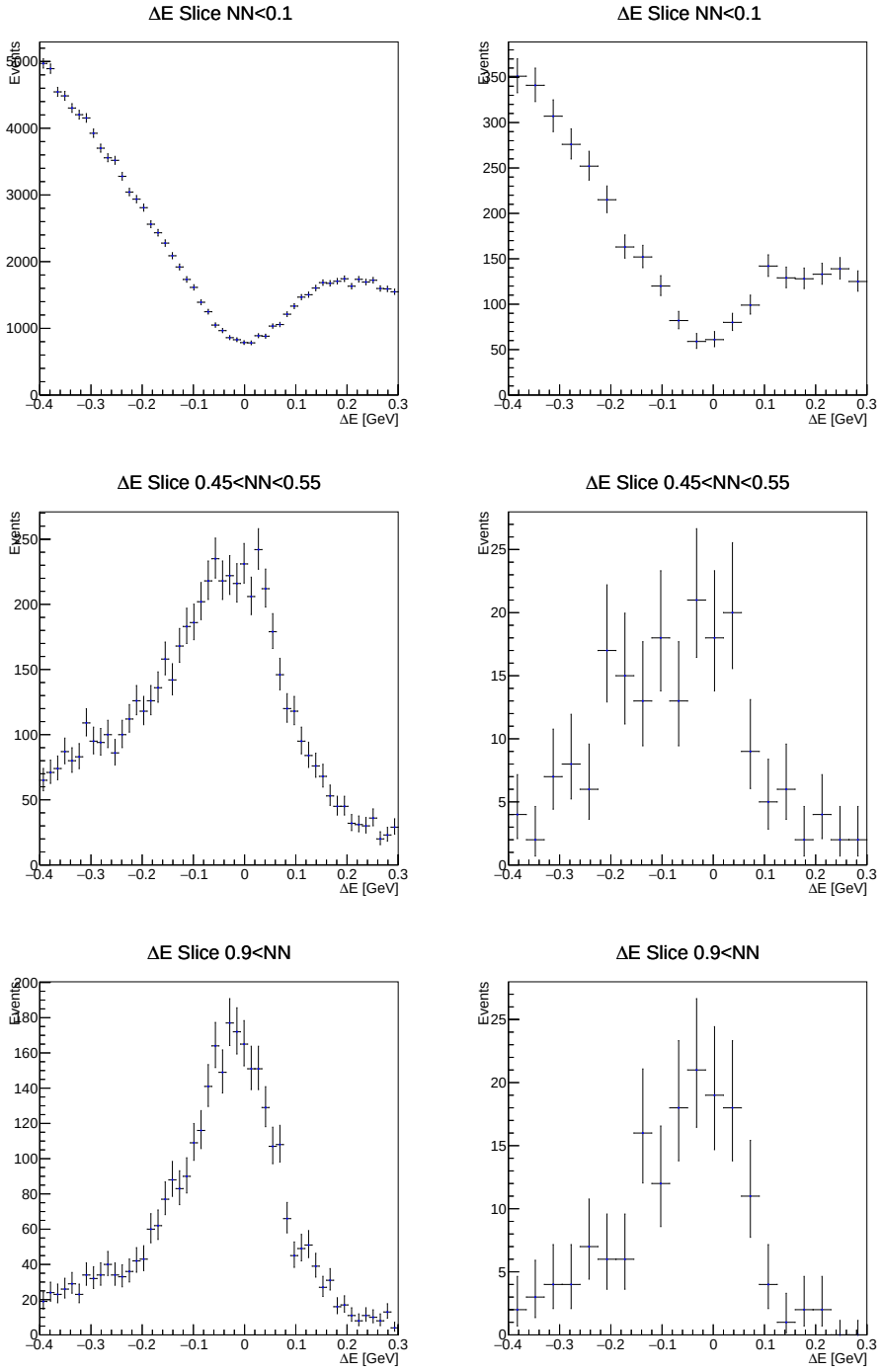

Figure 7.10: Showing the ΔE distributions at different NN slices. The effect is seen in both continuum MC (left column) and off-resonance (right column).

The kinematic variables with the highest correlation to ΔE were removed from the training of the neural network and the performance of the neural networks, along with the $\Delta E - NN$ correlation were investigated. This involved training four neural networks, removing the highest, two-highest, three-highest and four-highest

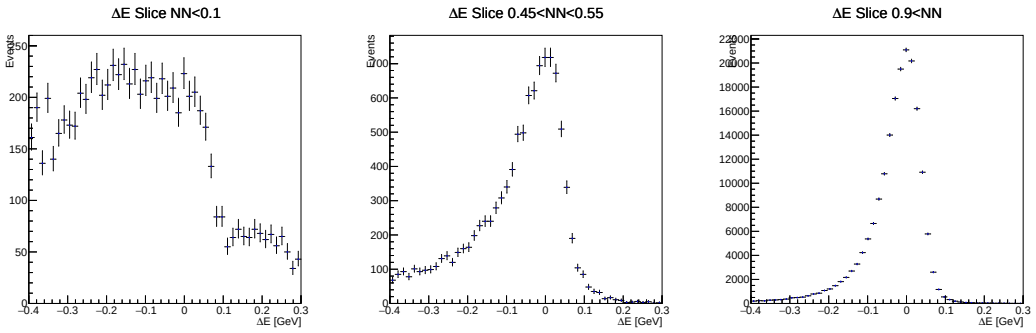

Figure 7.11: Showing the signal ΔE distributions for different NN slices.

correlated (to ΔE) variables.

The neural network configuration was different to above, as this was the best configuration found at the time of testing. The best performing neural network configuration used above was found at a later date. The neural network hyper-parameters were as follows:

- A maximum number of epochs of 600.
- 500 events per batch.
- A Learning rate of 0.0001.
- Ten hidden layers.
- 49 nodes per hidden layer.
- Exponential linear unit activation function.
- Using batch normalisation.
- A dropout chance of 0.004 and only applying to every hidden layer (in effect not applying dropout).

The performance of the neural networks, and the correlations between ΔE and the neural network outputs are summarised in Table 7.2. Successively removing the highest correlated variables does indeed reduce the $\Delta E - NN$ correlations in signal and continuum, at the cost of classifying power. Note that the correlations are calculated for $NN > 0.2$ ($NN > -0.6$ for the NeuroBayes neural network) as the smallest NN_{cut} is around this value. The neural network with 4 variables removed still shows a clear improvement in FOM and AUC compared to the NeuroBayes network, with only a slight increase in $\Delta E - NN$ correlation. This improvement however is not substantial, and the analysis is performed on the TensorFlow neural network with all kinematic variables, in the hope that the correlation introduced is outweighed by the greatly improved classifying ability.

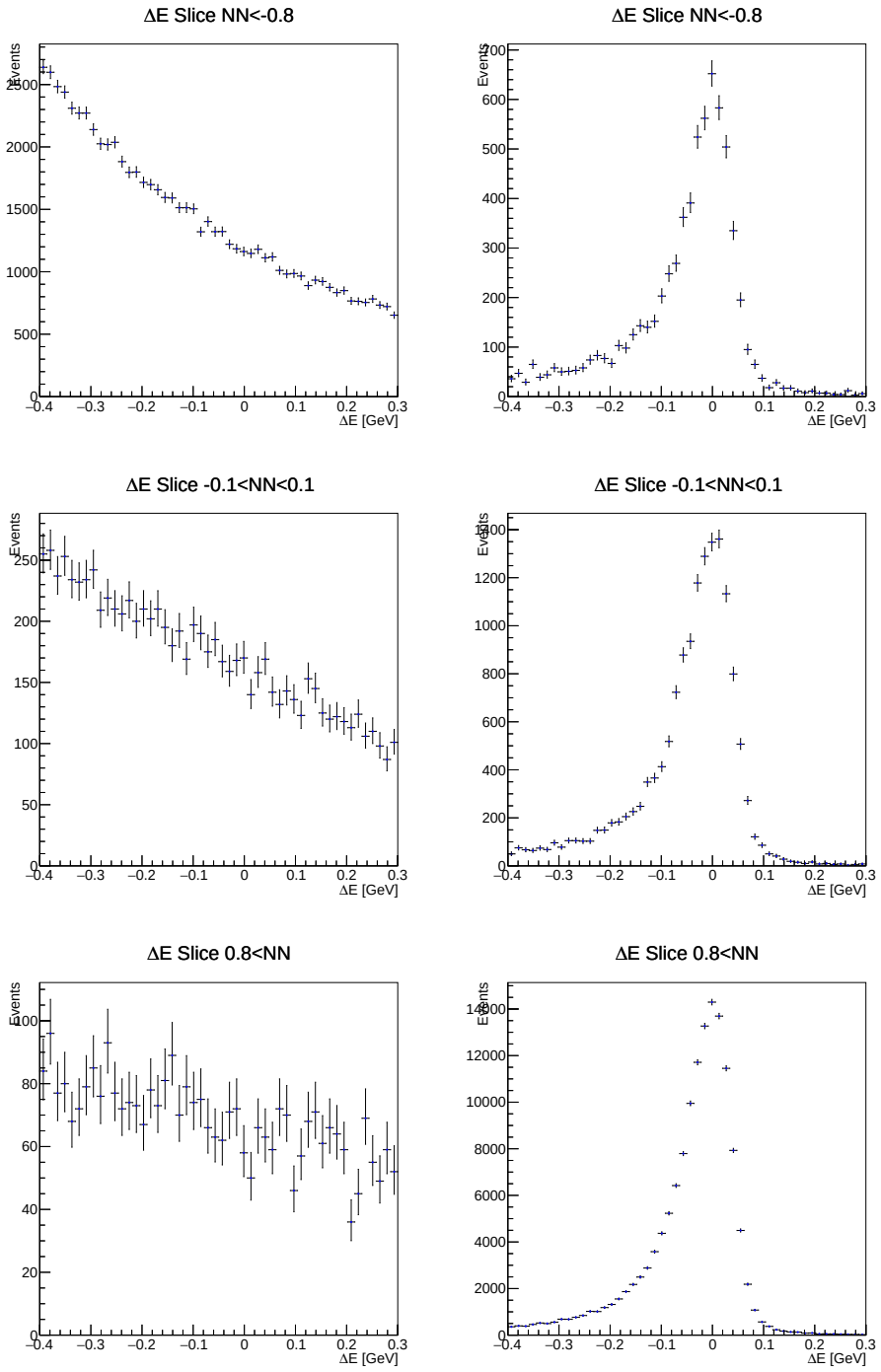

Figure 7.12: Showing the continuum (left column) and signal (right column) ΔE distributions at different slices of NN from the NeuroBayes neural network. Note that as NN is in the range ± 1 for the NeuroBayes output, the NN slices are adjusted accordingly.

	Best FOM	AUC	Correlation-Sig	Correlation-Cont
All (TF)	17.3 ± 0.4	0.947	17.9%	8.0%
1 Removed	16.3 ± 0.4	0.938	11.5%	7.3%
2 Removed	15.0 ± 0.4	0.928	7.9%	8.3%
3 Removed	14.6 ± 0.5	0.923	4.9%	5.6%
4 Removed	14.2 ± 0.3	0.918	4.3%	6.2%
All (NB)	13.2 ± 0.4	0.909	3.0%	4.7%

Table 7.2: The best FOMs, AUCs, signal and continuum correlations between ΔE and *NN*. 'All(NB)' refers to the network from Chapter 6, and 'All(TF)' refers to the network outlined above in 7.1.1 (with all kinematic variables).

7.2 Analysis of $B^0 \to K^0_S\pi^0$ Events

Although the $\Delta E - NN$ correlation in the neural network trained with all 19 kinematic variables is not ideal, the separation between signal and continuum is large, so there should be some reduction in the statistical uncertainty of $\mathcal{A}_{CP}$. The analysis follows the procedure laid out in 6.2.

A NN_{cut} value of 0.2971 is chosen to give the same number of signal events as in 6.2. This again leaves 92.3% of signal and now reduces the remaining continuum to 19.9%. The neural network transform will also be performed (NN_{max} is 0.999978). This NN_{cut} value gives the following expected event numbers:

- Signal : 1052 ± 57,
- Continuum : 12219 ± 28,
- Charged Rare: 280 ± 2,
- Mixed Rare: 104 ± 1,

The fitting regions are again:

- -0.4 GeV $< \Delta E < 0.3$ GeV
- 5.265 GeV$c^{-2} < M_{bc}^{corr} < 5.3$ GeVc^{-2}
- $-10.0 < NN^{trans} < 10.0$
- $-1.0 < q.r < 1.0$

7.2.1 Signal

The signal distributions for the correctly and incorrectly reconstructed B^0-mesons are shown in Figure 7.13. Their distributions are not treated separately.

The signal ΔE distribution is again given by a combination of a Crystal Ball function and a Chebyschev function of second order. The M_{bc}^{corr} distribution for signal is given by a Crystal Ball function and a Gaussian. Signal NN^{trans} is again a pair of Gaussians, given by a pair of Gaussians. And again $f_{signal}^{q.r|\mathcal{A}_{CP}=0}(q.r)$ is given by a kernel density estimation function, with mirroring at both edges, and a smoothing factor of $\rho = 0.75$. The one-dimensional signal PDFs are shown in Figure 7.14.

The scatter plots in every pair of dimensions are shown in Appendix B.2. As before, all correlations are relatively small (see Table B.3), but as expected, the $\Delta E - NN^{trans}$ correlation is now massive, at 17%.

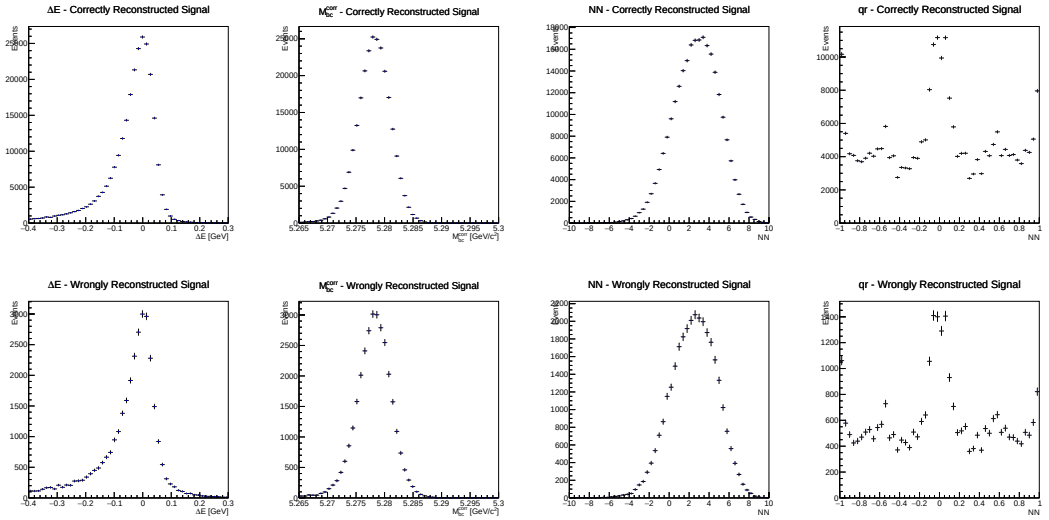

Figure 7.13: Showing the signal fitting variable distributions for the correctly (top row) and incorrectly (bottom row) reconstructed B^0-mesons. From left to right; ΔE, M_{bc}^{corr}, NN^{trans}, $q.r$.

7.2.2 Continuum

The ΔE distribution for continuum is no longer a simple Chebyschev function, it also contains a Gaussian due to the signal-like sculpting. The distribution is now given by a combination of a third-order Chebyschev function and a Gaussian. Continuum M_{bc}^{corr} is again modelled by an Argus distribution. The NN^{trans} distribution for continuum is given by a pair of Gaussians. Finally, $f_{continuum}^{q.r}(q.r)$ is given by a kernel density estimation function, with no mirroring and a smoothing of $\rho = 2$. The one-dimensional PDFs for continuum are shown in Figure 7.15.

The correlations between the dimensions for continuum are shown in Table B.4, alond with the scatter plots in Appendix B.2. Again the largest correlation, at 7%, is between ΔE and NN^{trans}

7.2.3 Rare Backgrounds

The ΔE distribution, $f_{rare-charged}^{\Delta E}(\Delta E)$ is modelled by a kernel density estimation function with a smoothing of $\rho = 2$ and mirroring on the left edge. $f_{rare-mixed}^{\Delta E}(\Delta E)$ is modelled with two Gaussians. The rare-background M_{bc}^{corr} distributions are modelled with Argus functions. The NN^{trans} distributions for both rare backgrounds are given by single Gaussians. And finally, the $q.r$ distributions are both modelled by kernel density estimation functions, with a smoothing of $\rho = 1$, and mirroring at both edges. The one-dimensional PDFs for the charged and mixed rare backgrounds are shown in Figures 7.16 and 7.17 respectively.

The correlations and scatter plots between each parameter (for both charged and mixed rare backgrounds) are shown in Appendix B.2. The largest correlations for charged and mixed backgrounds (as to be expected) are between ΔE and NN^{trans}

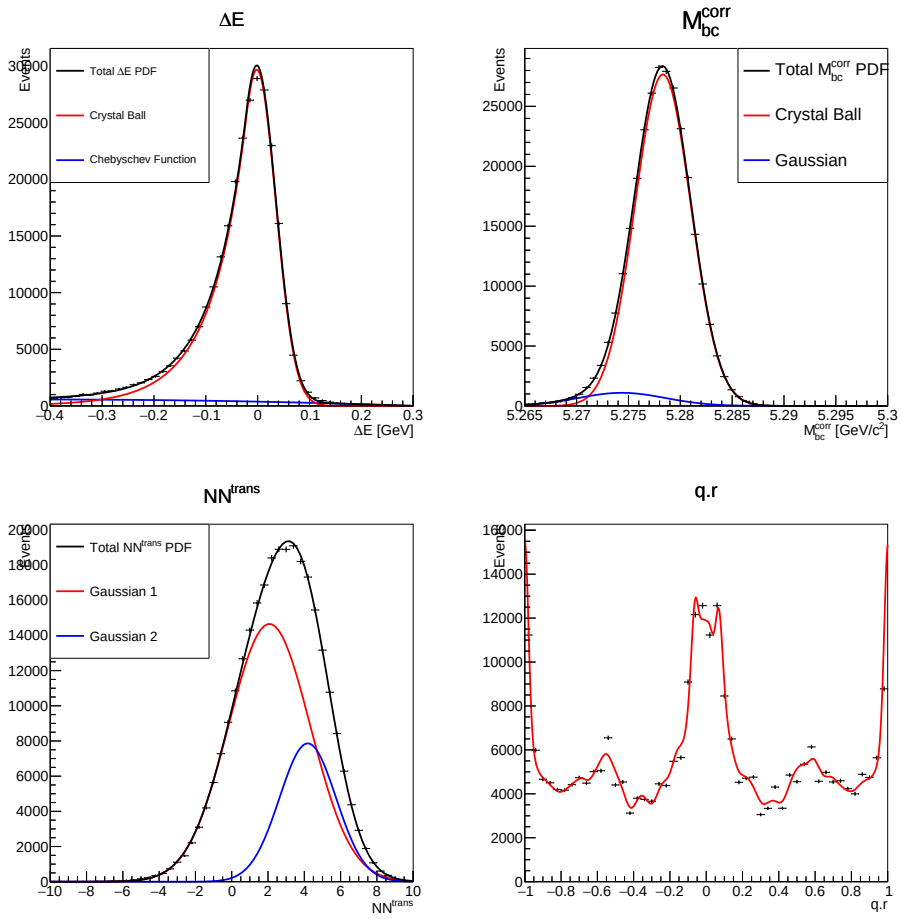

Figure 7.14: Showing the one-dimensional signal PDFs.

with correlations of 15% and 18% respectively.

As before, the expected yields for the most common rare decays are investigated (from the MC data), along with their uncertainties (using the branching ratio and uncertainties from [5]). See Appendix A for more details. This gives, for charged rare backgrounds:

- Known : 193 ± 39
- Unknown : 87 ± 35

And similarly for mixed-rare:

- Known : 59 ± 17
- Unknown : 46 ± 18

7.2.4 The 4-Dimensional Fit Results

The results of individual 4-dimensional fits for data samples with $\mathcal{A}_{CP}$ of 0 and 1 are shown in Figure 7.18. The statistical uncertainty introduced by fixing the rare background event numbers in the fitter is ± 9 signal events.

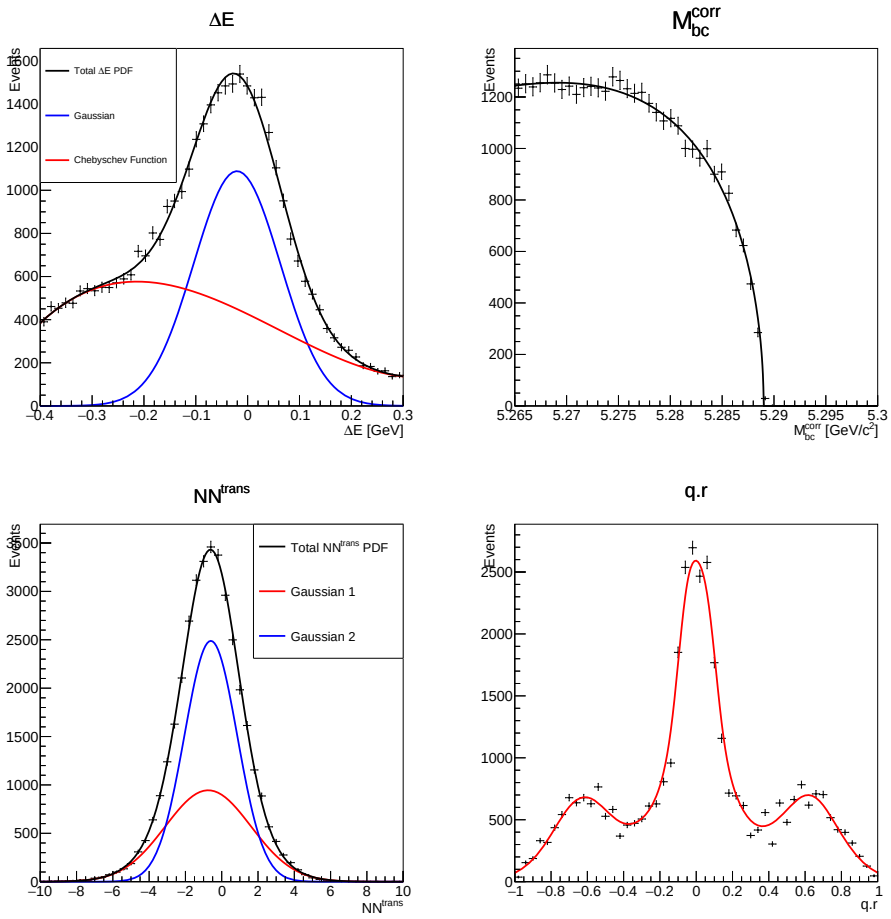

Figure 7.15: Showing the one-dimensional continuum PDFs.

Signal Yield Measurement

The fit is run with one-thousand data samples (with the signal sample size taken from a Poisson distribution of mean 1052) with $\mathcal{A}_{CP} = 0$, following the method explained in 6.2.4. The distribution of the measured signal event number has a mean of 1070.9 ± 1.4 and a standard-deviation of 44.5 ± 1.0. The distribution of statistical uncertainties returned by the fitter is 45.56 ± 0.02 and the standard deviation of this distribution is 0.75 ± 0.02. They are shown in Figure 7.19

Clearly the fitter is overestimating the signal event number, and slightly overestimating its uncertainty. This can be seen in the pull of the measured signal event number, (the distribution of which is shown in Figure 7.20). With a mean of 0.41 ± 0.03 and a standard-deviation of 0.97 ± 0.02, this distribution clearly shows that the fitter overestimates the signal event number.

Running the 500 fits at a range of input signal event numbers gives a clearer picture on how the fitter behaves. The input signal event numbers (the mean of the Poisson distributions from which the number of events to sample is selected) are varied from 0.75 to 1.25 times the expected signal event number, in steps of 0.05 times the expected event number. The measured results against the input are

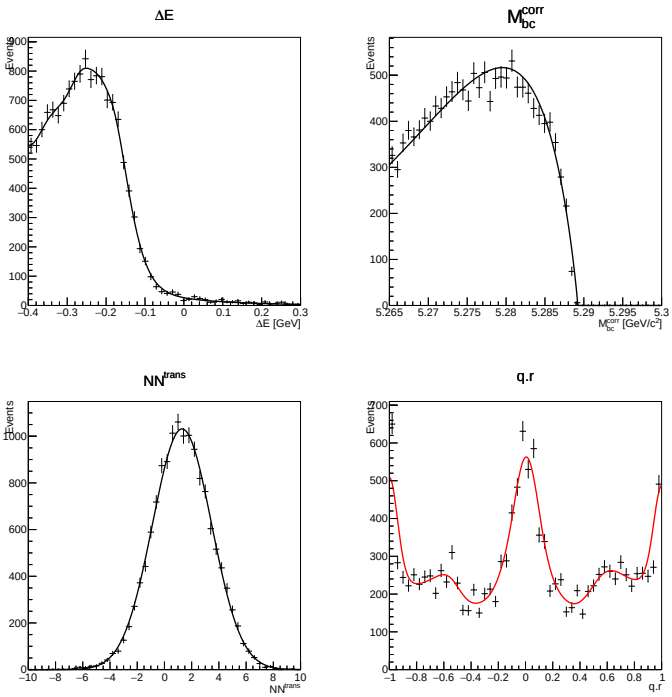

Figure 7.16: Showing the one-dimensional charged rare background PDFs.

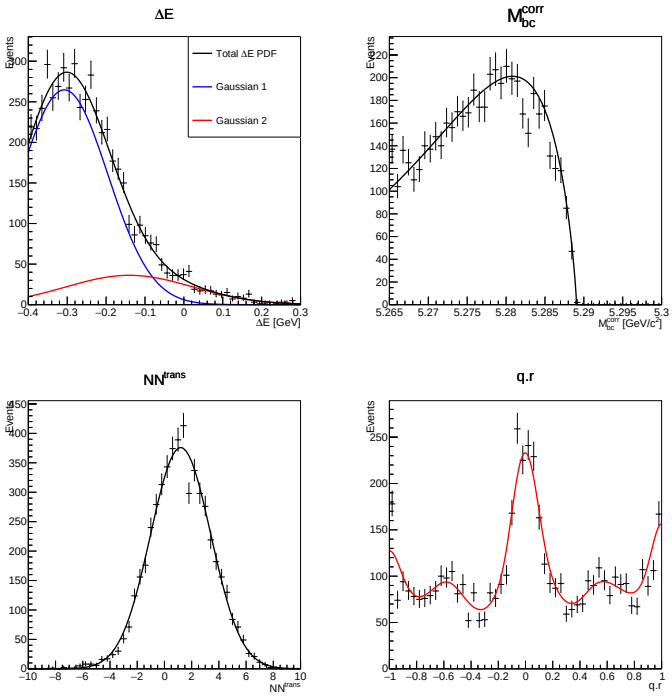

Figure 7.17: Showing the one-dimensional mixed rare background PDFs.

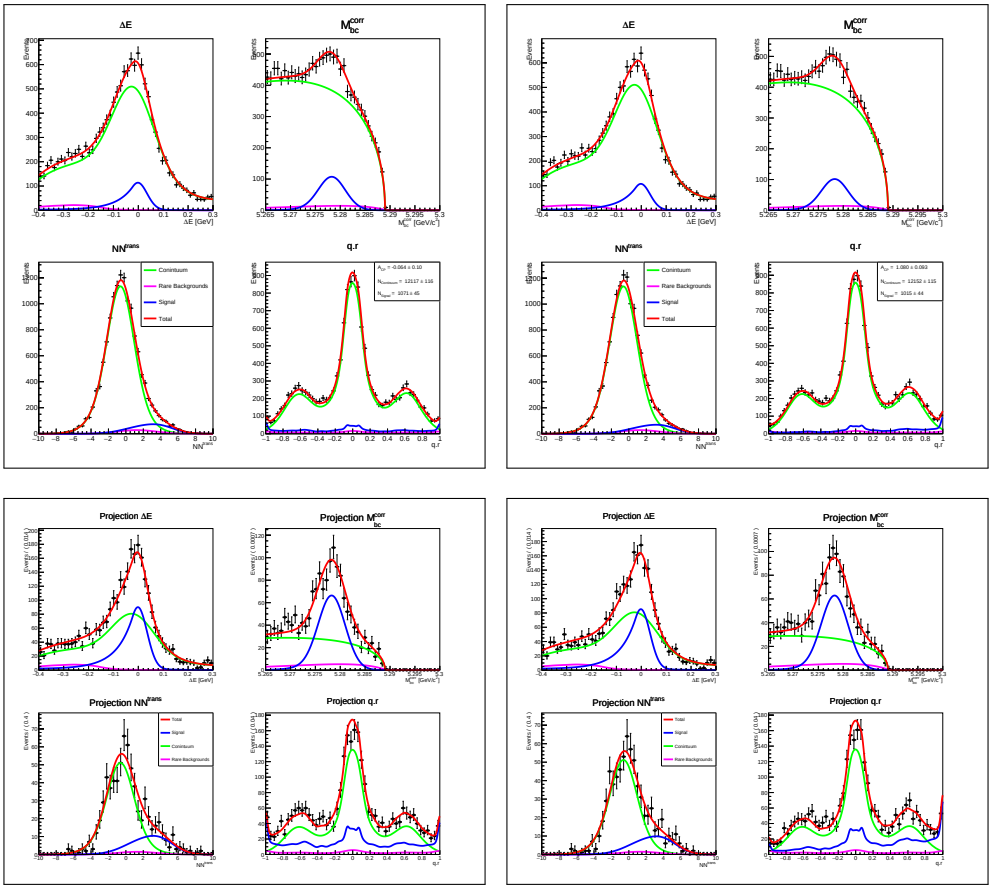

Figure 7.18: Showing the 4-dimensional fits (top row) and projections plots (bottom row) to data samples with $\mathcal{A}_{CP} = 0$ (left column) and $\mathcal{A}_{CP} = 1$ (right column).

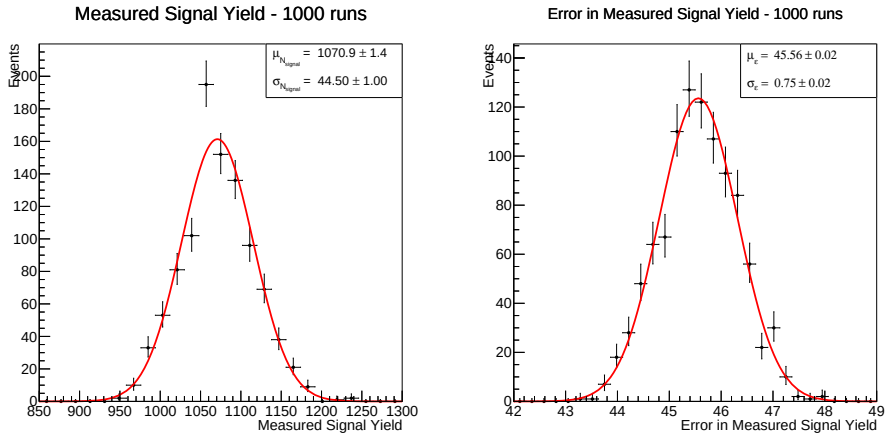

Figure 7.19: Showing the distribution of measured signal yields (left) and the errors in the signal yields (right) over a thousand runs.

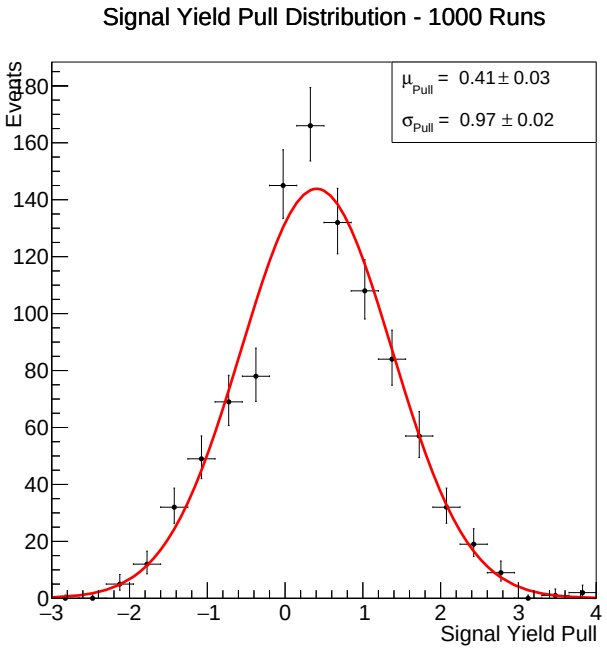

Figure 7.20: Showing the pull in signal yield over one-thousand runs.

shown in Figure 7.21. The 1st order polynomial fit to this data has a gradient of 0.990 ± 0.004 (which although not within the error range of one, very close), and a y-intercept of 30.1 ± 4.6. Clearly the fitter consistently overestimates the number of signal events, adding a correction of -18.9 events to the fitter could fix this.

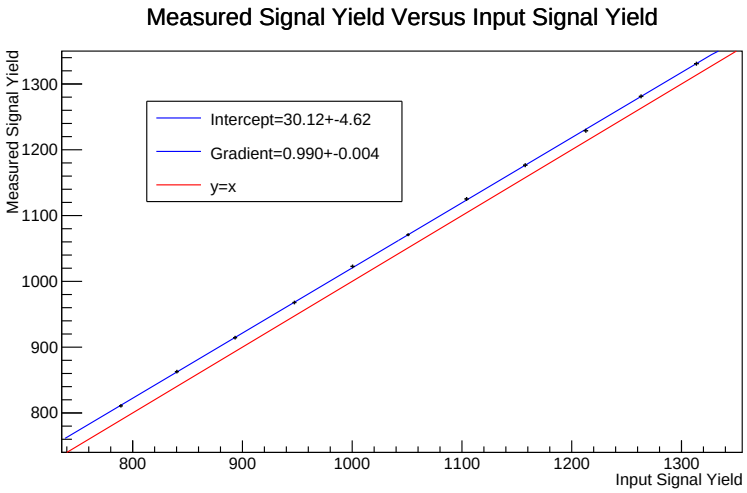

Figure 7.21: Showing the mean of the measured signal yields against the mean of the signal data sample sizes..

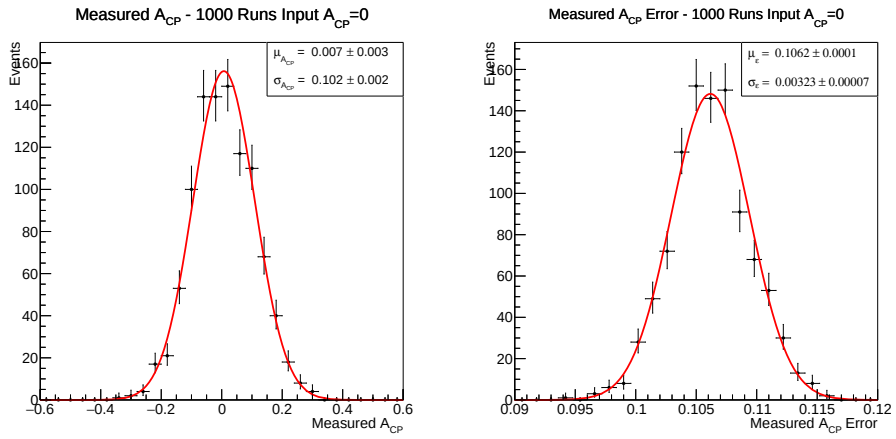

Figure 7.22: Showing the measured $\mathcal{A}_{CP}$ (left) and error in measured $\mathcal{A}_{CP}$ (right) over one-thousand runs.

$\mathcal{A}_{CP}$ Measurement

The measured $\mathcal{A}_{CP}$ distribution (over one-thousand samples with input $\mathcal{A}_{CP} = 0$) is shown in Figure 7.22. The mean of this distribution is 0.007 ± 0.003. The standard-deviation on these measurements is 0.102 ± 0.002 - a significant improvement over the results from the NeuroBayes neural network.

The fitter statistical uncertainty distribution is also shown in Figure 7.22. The mean is 0.1062 ± 0.0001 and its standard-deviation is 0.00323 ± 0.00007. The quoted statistical uncertainty is worse than the actual standard deviation in the measurements. Both results still show an improvement over the previous results.

The pull distribution, shown in Figure 7.23, has a mean of 0.07 ± 0.03 and a standard-deviation of 0.97 ± 0.02.

Finally the fit is performed 500 times for data samples with $\mathcal{A}_{CP}$ values of ± 1 and from -0.5 to 0.5 in steps of 0.5. The measured $\mathcal{A}_{CP}$ versus the input data $\mathcal{A}_{CP}$ is shown in Figure 7.24. The y-intercept of 0.0103 ± 0.0012 and gradient of 0.974 ± 0.003 show that the model would be a better fit if a small shift and scaling were applied to the $\mathcal{A}_{CP}$ in the first-order polynomial in signal $q.r$.

This result already shows a clear improvement over the original model in Chapter 6, and the previous Belle result. Sadly the ΔE correlation with the neural network output seems to be preventing the statistical uncertainty in $\mathcal{A}_{CP}$ being reduced significantly further. Finding a way to keep the good performance of the neural network whilst not introducing the $\Delta E - NN$ correlations could show promising results.

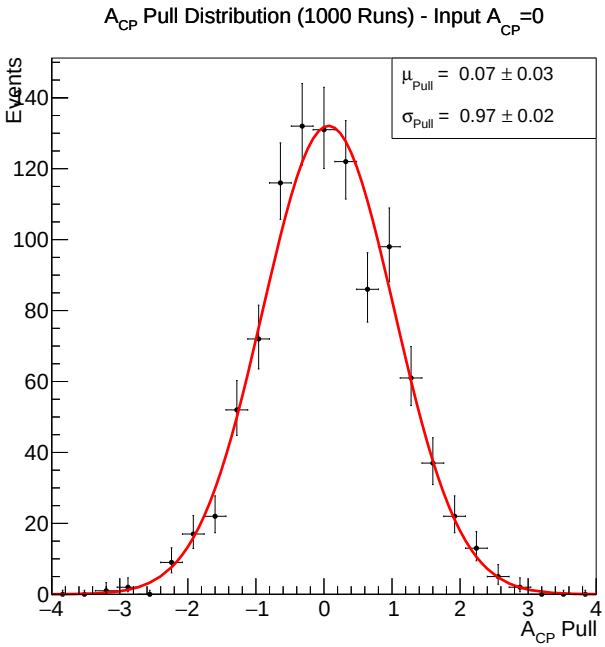

Figure 7.23: Showing the $\mathcal{A}_{CP}$ pull distribution over a thousand runs.

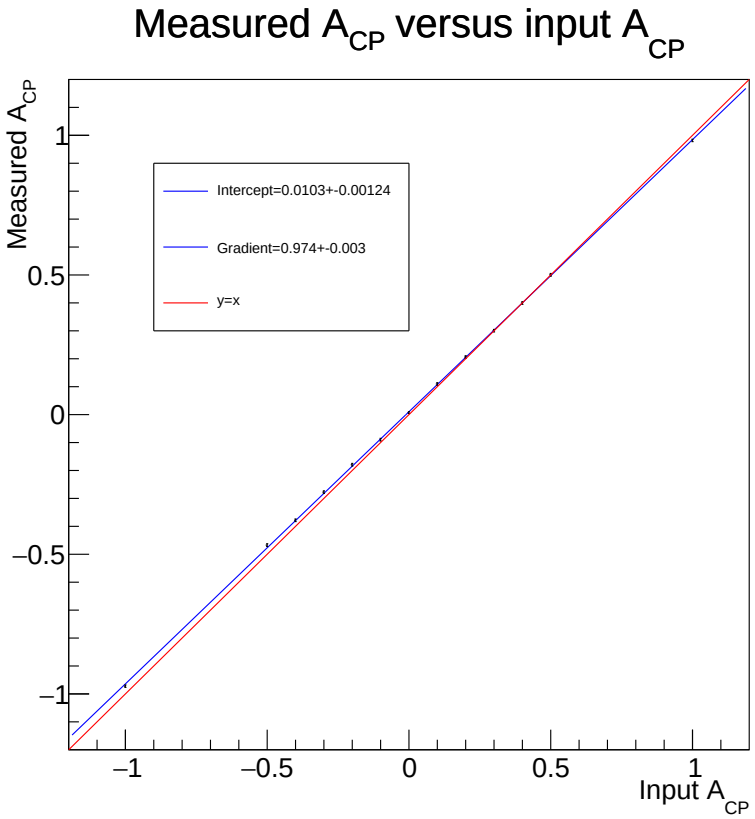

Figure 7.24: Showing the mean measured $\mathcal{A}_{CP}$ against data sample $\mathcal{A}_{CP}$.

8|Adversarial Neural Networks

A possible way to further reduce the statistical uncertainty on the $\mathcal{A}_{CP}$ measurement is to use the gains from the improved continuum suppression (see Chapter 7) but reduce the correlation between ΔE and the neural network output. Removing the input kinematic variables that are correlated with ΔE achieves this, but at too great a cost to the continuum suppression. A way forward is to use all of the kinematic variables and the TensorFlow neural network, but to train it in a way that correlations between NN and ΔE are penalised.

8.1 Adversarial Neural Network

Adversarial neural networks were first developed in order to generate images from a trained image-recognition convolutional neural network, and used to further train the convolutional neural network in order to perform better in classification tasks [43]. The idea of an adversarial neural network can be used to reduce the correlations between the output of a neural network (referred to as the classifying neural network) and other parameters associated with the event. This method is used to reduce the correlation between NN and ΔE, although in principle this could also be used for any one of, or multiple parameters that have correlations with NN. The method laid out here closely follows that in [44].

The adversarial network tries to model the ΔE distribution by taking NN as input, and trying to predict the value of ΔE that a given event will have. The network used in this study has one input (NN), two hidden layers with 20 nodes each, and 15 outputs. This models ΔE with five Guassians (indexed by i), with 3 outputs for each, corresponding to the means ($\mu_i(NN)$), widths ($\sigma_i(NN)$), and fractional weighting of that Gaussian ($f_i(NN)$) - the fractions are not normalised so they are first passed through a softmax function (giving $f_i'(NN)$, scaled to sum to one). The adversary loss function (for a single event) is given by:

$$\mathcal{L}_{adv}\left(NN, \Delta E\right) = -log\left(\sum_{i=1}^{5} \frac{f_i'\left(NN\right)}{\sqrt{2\pi\sigma_i^2\left(NN\right)}} \exp\left\{\frac{-\left(\mu_i\left(NN\right) - \Delta E\right)^2}{2\sigma_i^2\left(NN\right)}\right\}\right) \tag{8.1}$$

Training the adversarial neural network to minimise this loss function will allow it to predict the ΔE distribution from the input NN distribution if the two parameters are correlated and the ΔE distribution can be roughly modelled with five Gaussians.

We want to penalise the classifying neural network if NN is correlated to ΔE, so the classifier is further trained to minimise:

$$L_{tot} = L_{class} - \lambda_{adv} L_{adv} \tag{8.2}$$

Where L_{class} is our loss function for the classifier network (the cross entropy, see 5.2), and λ_{adv} is a constant chosen to specify how much to penalise $\Delta E - NN$ correlations. Training to this new loss function has the desired impact of reducing the correlations at the cost of the continuum suppression. A λ_{adv} of zero would result in a trained classifier maximally separating continuum and signal, whereas a larger λ_{adv} results in a drastically reduced $NN - \Delta E$ correlation but with significantly worse classifying power. The configuration of the networks is shown in Figure 8.1.

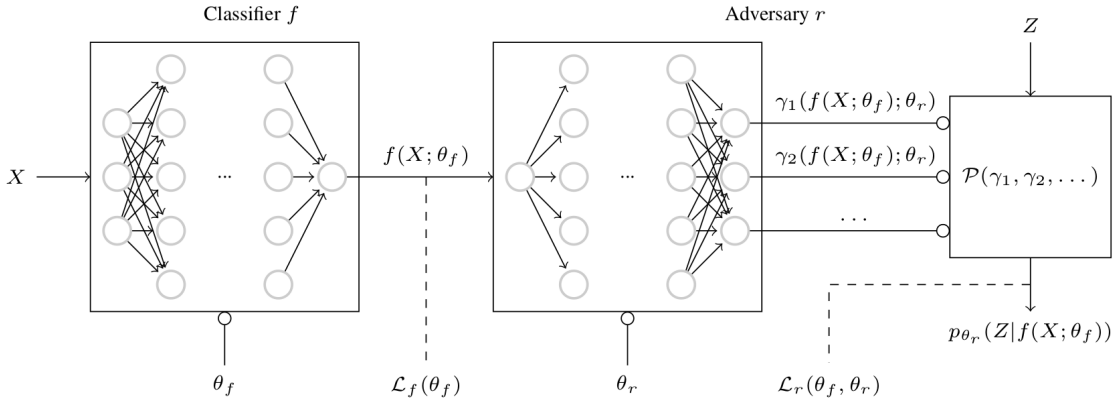

Figure 8.1: Showing the configuration of the classifying and adversarial neural networks. θ_f and θ_r are the trainable-weights in the classifier and adversarial network respectively. X is the vector of input kinematic variables. $f(X;\theta_f)$ is NN. Z is ΔE. γ_{1-15} are the Gaussian means, standard-deviations and fractions, and $\mathcal{P}$ is the function that combines these (with ΔE) into the likelihood function p_{θ_r}. $\mathcal{L}_f(\theta_f)$ and $\mathcal{L}_r(\theta_f,\theta_r)$ are L_{class} and L_{adv} respectively. Image from [44].

There are now the additional hyper-parameters associated with the architecture of the adversary network. These include the number of outputs (related to the number of Gaussians with which to model the ΔE distribution), the number of hidden layers and nodes per hidden layer. The additional hyper-parameters associated with training the adversary network are its batch-size, training steps and learning rate.

The classifier neural network has the same architecture, and is initialised to the optimal weight values from the previous training in 7.1.1. For every classifier training step, the adversary network is first trained for 100 steps using the Adam optimiser with a batch size of 125 and a learning rate of 0.01. The learning rate for training the classifier is reduced to 10^{-6} and the training is run for 4 epochs.

The main hyper-parameters for the classifier are:

- A maximum number of epochs of 4.
- 50 events per batch.

- A Learning rate of 10^{-6}.
- Six hidden layers.
- 47 nodes per hidden layer.
- Exponential linear unit activation function.
- No batch normalisation.
- A dropout chance of 0.007 and only applying to every hidden layer (in effect not applying dropout).
- No L2 regularisation.

And the hyper-parameters associated with the adversarial neural network are:

- 100 training steps.
- 125 events per batch.
- A Learning rate of 0.01.
- Two hidden layers.
- 20 nodes per hidden layer.
- Exponential linear unit activation function for the nodes in the hidden layer.
- 15 output nodes (three output nodes corresponding to each Gaussian):
  - 5 output nodes corresponding to μ_i - no activation function (identity operator).
  - 5 output nodes corresponding to un-normalised fractions f_i - no activation function (identity operator).
  - 5 output nodes corresponding to σ_i, where the 'activation' is the exponential function, to ensure that the widths of the Gaussians are positive.
- No batch normalisation, dropout or L2 regularisation.

The method is then as follows:

1. Train the neural network to optimally separate signal and continuum as in Chapter 7. Save the weights.
2. Create the adversary neural network, and the classifying (the original) neural network with the same architecture, and initialise the weights to that of the saved best model.
3. For every (20,000 steps as there are four epochs and a batch size of 50) classifier training step and a given choice of λ_{adv}:
   (a) Train the adversary neural network for the given number of adversary training steps (100 steps), where for each step:
      i. For every event in the batch (where the number of events in the batch is the adversary batch size, 125 events), get the *NN* output from the classifier.

ii. Using NN and ΔE get the adversarial loss given by 8.1.

iii. Train the adversarial neural network given the adversarial loss, adversarial learning rate (0.01) and gradient-decent algorithm of choice (Adam optimiser).

(b) Train the classifier neural network as normal for one training step, with the difference that the loss function is now given by 8.2 and has a dependence on ΔE, as well as NN and $\hat{y}$ (one or zero depending on if an event is signal or continuum).

4. Save the weights of the classifying neural network and use this updated neural network for further analysis.

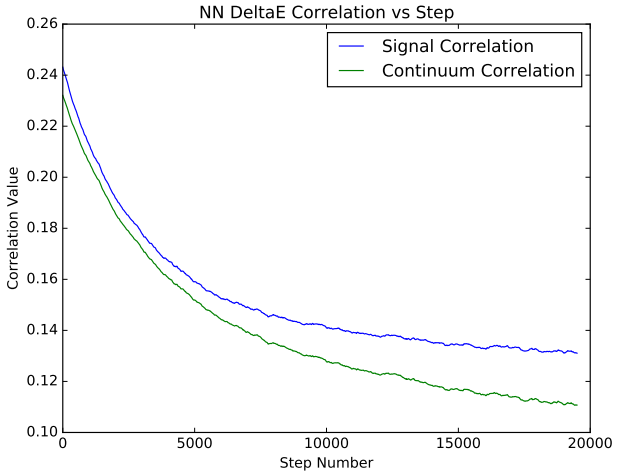

Figure 8.2: Showing the signal (blue) and continuum (red) testing dataset correlations between ΔE and NN as the training proceeds. This corresponds to 4 epochs, of 5000 classifier-training steps each, where the adversarial network is trained for 125 steps per classifier training step. Note that these correlations are in the testing datasets, and calculated over the entire range $0 < NN < 1$

The investigation into the varying λ_{adv} is performed. The procedure was repeated for a range of λ_{adv}; 0.25, 0.5, 0.75, 1.0 and 1.5. As the training proceeds, the $\Delta E - NN$ correlations steadily decrease to a minimum. For larger λ_{adv}, the correlations quickly decrease to zero, before bouncing and increasing. This behaviour can be explained by the competing neural networks, for smaller λ_{adv} they settle to coupled situation where training in one network is counteracted by the other. The larger λ_{adv} see the adversarial network dominate quickly before the classifier has had enough training steps to counteract it, this should be investigated further. The $\Delta E - NN$ correlations against the number of training steps for $\lambda_{adv} = 0.5$ is shown in Figure 8.2. The continuum MC (validation dataset) ΔE distributions (at different NN slices) for each λ_{adv} are shown in Figure 8.3.

The details of how the neural networks perform and the $\Delta E - NN$ correlations on the signal and continuum testing datasets are summarised in Table 8.1. Note that as

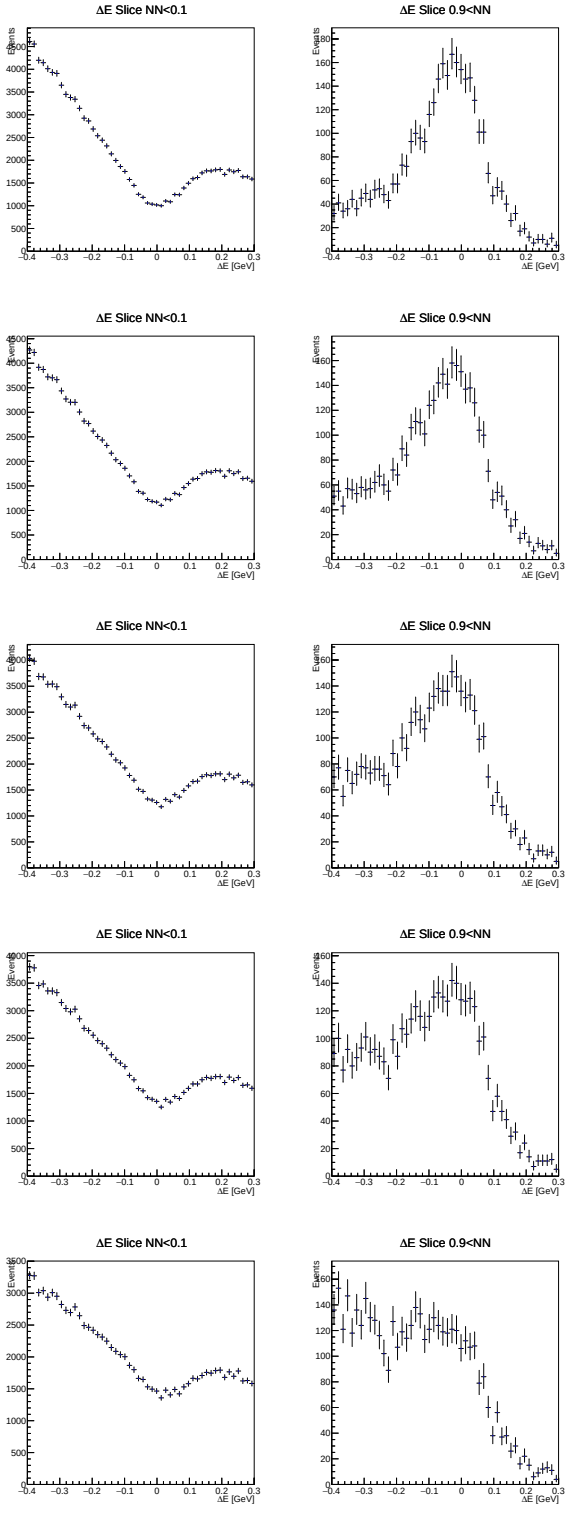

Figure 8.3: Showing the continuum ΔE slices at $NN < 0.1$ (left column) and $NN > 0.9$ (right column). Rows one to five correspond to λ_{adv} values of 0.25, 0.5, 0.75, 1.0 and 1.5 respectively.

λ_{adv}	Best FOM	AUC	Correlation-Signal	Correlation-Continuum
N/A (TF)	17.3 ± 0.4	0.947	17.9%	8.0%
0.25	17.3 ± 0.5	0.947	13.9%	7.8%
0.50	17.2 ± 0.5	0.945	10.8%	6.8%
0.75	16.9 ± 0.5	0.942	8.3%	4.8%
1.00	16.8 ± 0.4	0.939	6.4%	3.0%
1.50	15.7 ± 0.5	0.931	1.8%	1.0%
N/A (NB)	13.2 ± 0.4	0.909	3.0%	4.7%

Table 8.1: The best FOMs, AUCs, signal and continuum correlations between ΔE and NN. 'N/A (NB)' refers the the network from Chapter 6, and 'N/A (TF)' refers to the best TensorFlow network outlined in 7.1.1.

before, the correlations are calculated for $NN > 0.2$ ($NN > -0.6$ for the NeuroBayes neural network) as the smallest NN_{cut} is around this value. The results for the network with $\lambda_{adv} = 1.5$ show both a better classifying power than NeuroBayes *and* smaller correlations - the NeuroBayes network is therefore worse even when considering the worrying $\Delta E - NN$ correlations. As can be seen from the results of the network with $\lambda_{adv} = 0.5$, at a very small price paid (almost negligible) with respect to classifying ability, the $\Delta E - NN$ correlations can be significantly reduced. In order to minimise the $\Delta E - NN$ correlation whilst maintaining a large classifying ability, the neural network trained with $\lambda_{adv} = 0.5$ is investigated further.

8.1.1 Analysis of the Neural Network Performance

The neural network trained with $\lambda_{adv} = 0.5$ is applied to the validation datasets. The signal and continuum MC NN distributions are shown in Figure 8.4. A value of NN_{cut} chosen to keep 13.0% of continuum MC leaves 87.51% of signal remaining, and a choice of NN_{cut} to keep 70.2% of signal leaves 3.49% of continuum remaining.

The NN distributions with the expected numbers of signal and continuum MC, and the FOM distribution with NN_{cut} are shown in Figure 8.5. The best FOM is 17.2 ± 0.5.

The ROC curve for signal and continuum MC is shown in Figure 8.6, and gives an AUC of 0.945.

The signal and off-resonance NN distributions are shown in Figure 8.7. A NN_{cut} for 13.0% of off-resonance gives 84.57% of signal, and a NN_{cut} for 70.2% of signal leaves 5.13% of off-resonance remaining.

The ΔE at different NN slices for signal, continuum MC, and off-resonance are shown in Figure 8.8. The off-resonance distribution forms match that of continuum MC. Both signal and continuum MC show less sculpting than with the non-adversarial TensorFlow neural network in 7.1.1 (see Figures 7.10 and 7.11).

The ROC curve for signal and off-resonance gives an AUC of 0.934, see Figure 8.9.

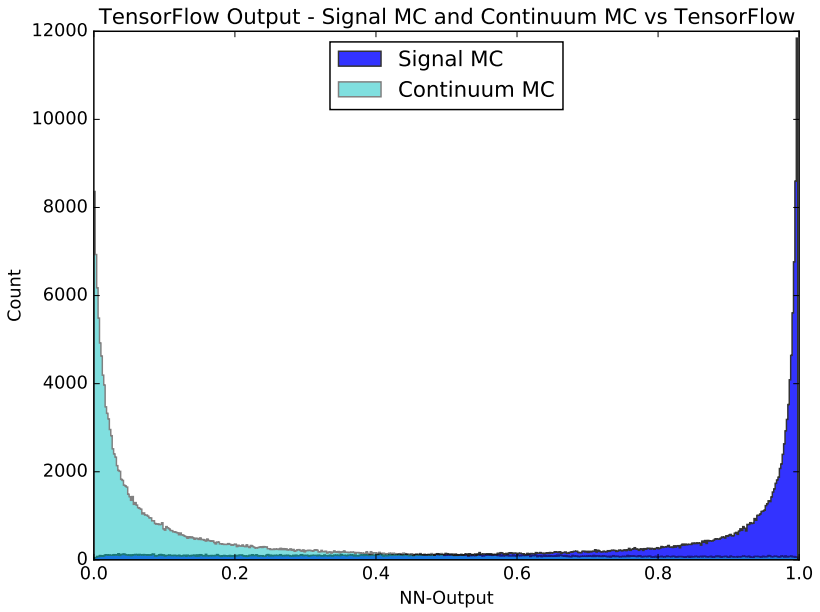

Figure 8.4: Showing the *NN* distributions for signal and continuum MC in equal numbers.

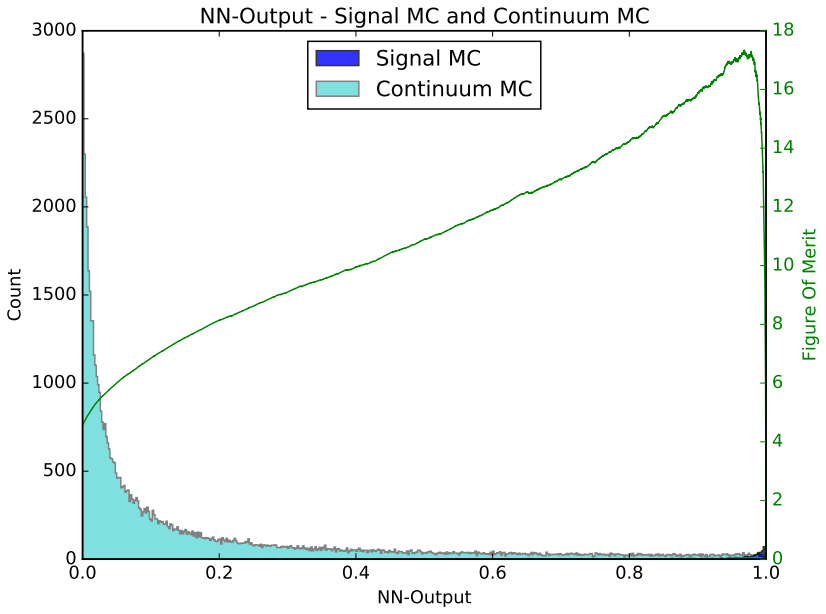

Figure 8.5: Showing the *NN* distributions in their expected numbers. The FOM distribution (green) is also plotted.

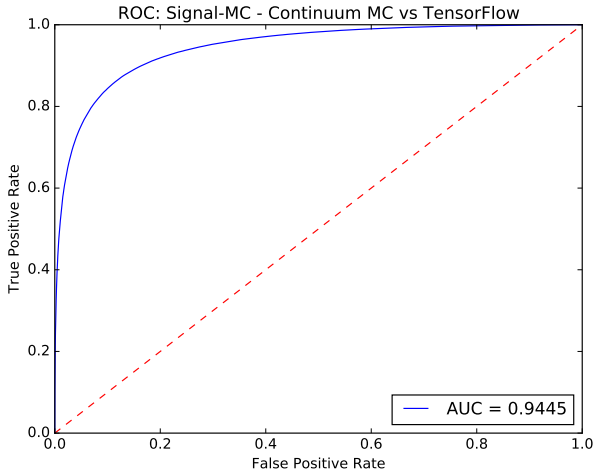

Figure 8.6: Showing the ROC curve of signal and continuum MC, giving an AUC of 0.945.

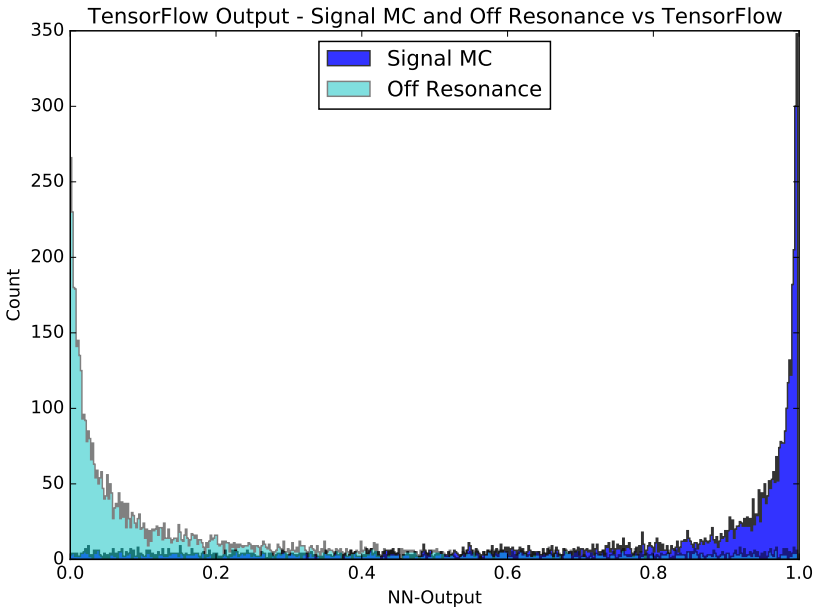

Figure 8.7: Showing the *NN* distributions for signal and off-resonance in equal numbers, where the noisiness is due to the smaller sample size.

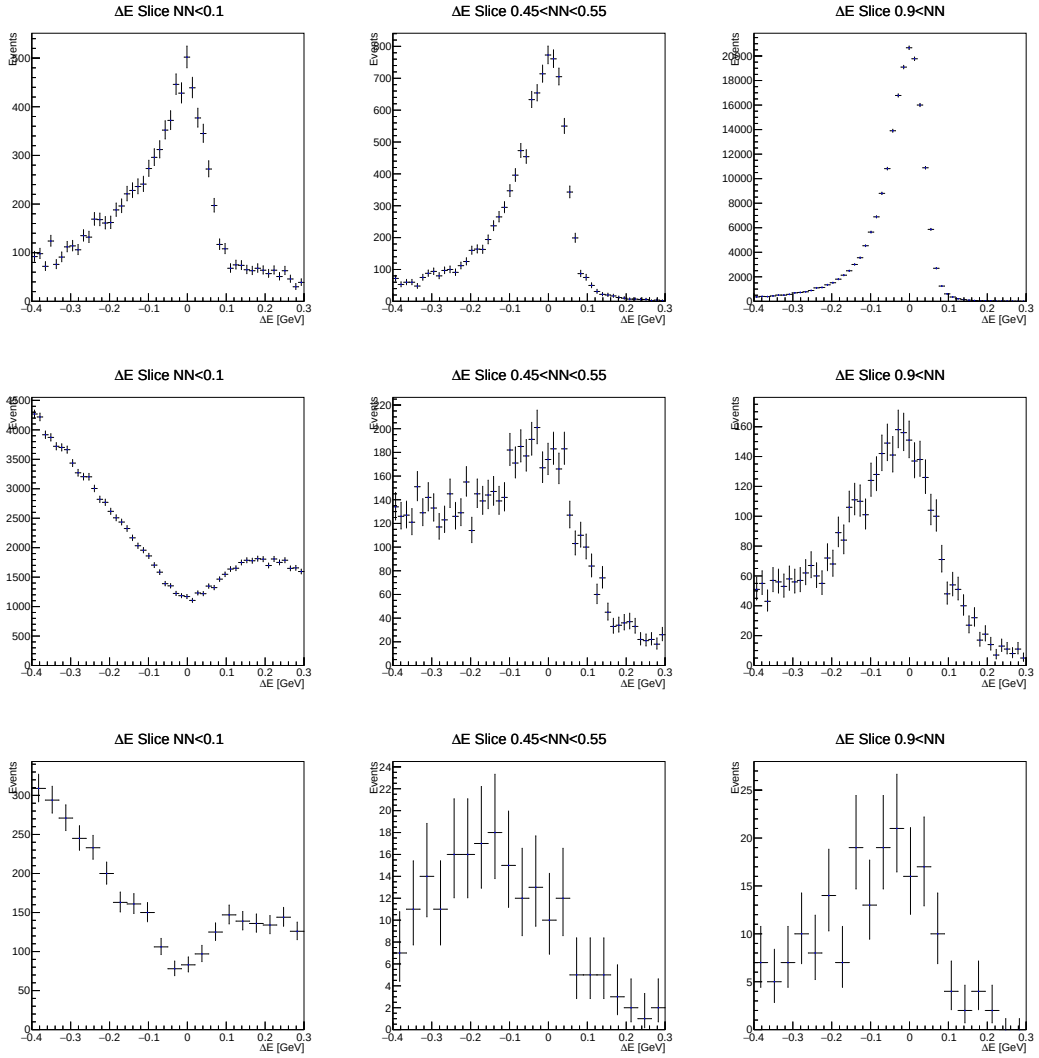

Figure 8.8: Showing the signal (top row), continuum (middle row) and off-resonance (bottom row) ΔE distributions for $NN < 0.1$ (left column), $0.45 < NN < 0.55$ (middle column) and $0.9 < NN$ (right column).

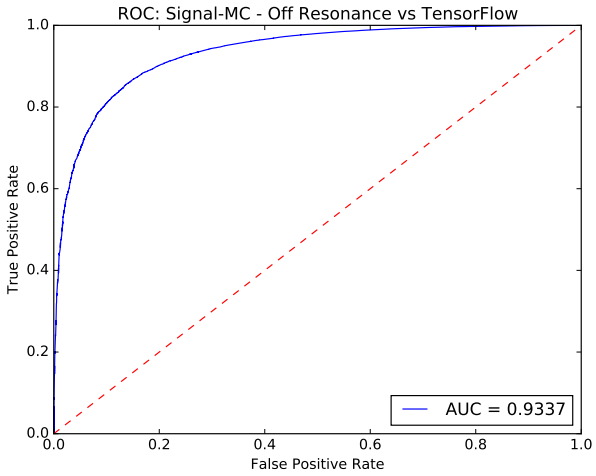

Figure 8.9: Showing the ROC curve for signal and off-resonance data, giving an AUC of 0.934.

8.2 Analysis of $B^0 \to K_S\pi^0$ Events

The analysis is now performed on the signal and continuum validation datasets, along with the rare background datasets, processed by this retrained neural network (with $\lambda_{adv} = 0.5$). The fitting procedure follows the method laid out in 6 in order to compare the performance of the four-dimensional fit for the adversarialy trained and non-adversarialy trained neural networks. The fitting regions are again:

- -0.4 GeV $< \Delta E < 0.3$ GeV
- 5.265 GeV$c^{-2} < M_{bc}^{corr} < 5.3$ GeVc^{-2}
- $-10.0 < NN^{trans} < 10.0$
- $-1.0 < q.r < 1.0$

The NN_{cut} value of 0.2796 (note that NN_{max} is 0.999995) is chosen (keeping 92.3% and 21.0% of signal and continuum respectively) to give the expected event numbers of:

- Signal : 1052 ± 57,
- Continuum : 12904 ± 30,
- Charged Rare: 308 ± 2,
- Mixed Rare: 115 ± 1,

8.2.1 Signal

The signal ΔE distribution is modelled with a Crystal Ball function and a second order Chebyschev function. The M_{bc}^{corr} distribution is modelled with a Gaussian and a Crystal Ball function. The signal NN^{trans} distribution is modelled with two Gaussians. The signal $q.r$ distribution is again modelled with a kernel density estimation function with a smoothing factor of $\rho = 0.75$ and mirroring at both edges. The one-dimensional signal PDFs are shown in Figure 8.10.

The scatter plot of every pair of dimensions is shown in Appendix B.3. The largest correlation is again $\Delta E - NN^{trans}$ at a, smaller than in 7, but still large, 12%. See Table B.5.

8.2.2 Continuum

The continuum ΔE distribution is modelled with (as in 7) a Chebyschev function and a Gaussian. The continuum M_{bc}^{corr} distribution is fit with an Argus function. The NN^{trans} distribution for continuum is modelled with a pair of Gaussians. The $q.r$ distribution for continuum is modelled with a kernel density estimation function with a smoothing of $\rho = 2$ and no mirroring. The continuum one-dimensional PDFs are shown in Figure 8.11.

The scatter plots between the fitting dimensions for continuum are shown in Appendix B.3, with their correlations in Table B.6. The correlation between ΔE and NN^{trans} at 5.9%

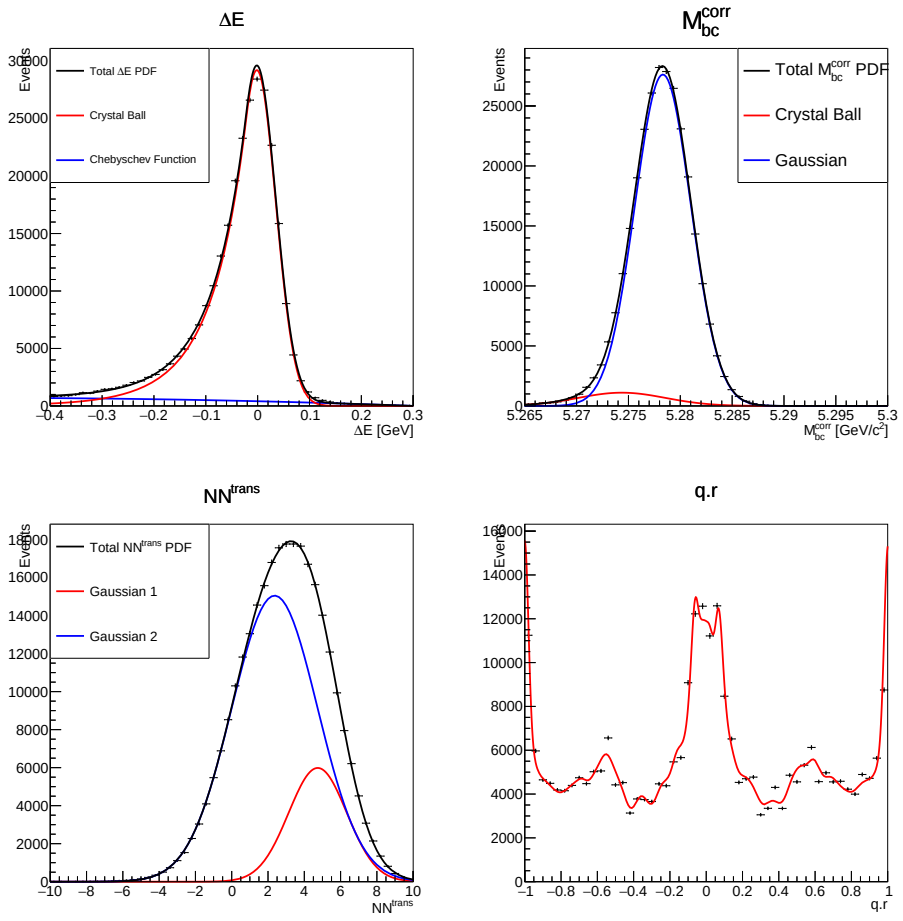

Figure 8.10: Showing the one-dimensional signal PDFs.

8.2.3 Rare Backgrounds

The ΔE distribution for the charged-rare background is a kernel density estimation function with mirroring at the left edge and a ρ of 2. The mixed-rare background ΔE distribution is modelled by a pair of Gaussians. The M_{bc}^{corr} distributions for the rare backgrounds are fit with Argus functions. The NN^{trans} distributions for both charged and mixed rare backgrounds are modelled with single Gaussians. The $q.r$ distributions for both charged and mixed rare-backgrounds are given by kernel density estimation functions with mirroring at both edges and a smoothing of $\rho = 1$. The one-dimensional PDFs for the charged and mixed rare backgrounds are shown in Figures 8.12 and 8.13 respectively.

The correlations between each of the dimensions for both charged and mixed rare backgrounds are shown in Appendix B.3. The correlations between ΔE and NN^{trans} for charged and mixed rare backgrounds being 9% and 11% respectively.

Investigating the expected number of events from the most common rare decays gives for charged rare backgrounds:

- Known : 212 ± 43

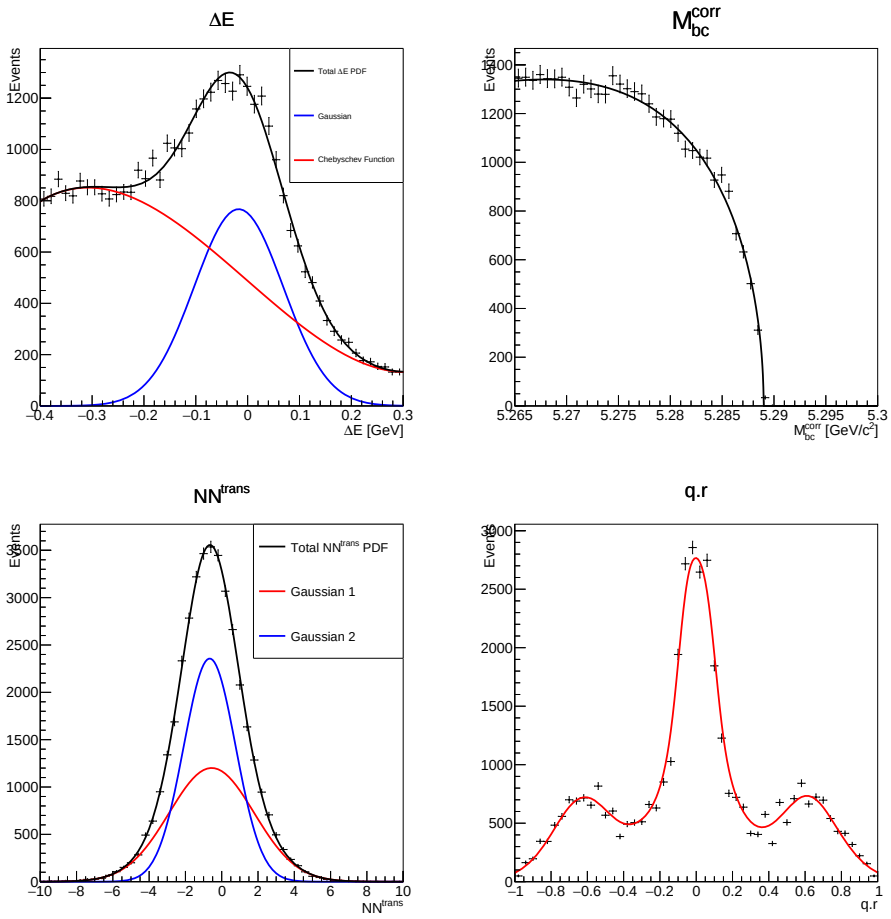

Figure 8.11: Showing the one-dimensional continuum PDFs.

- Unknown : 96 ± 38

And for mixed-rare backgrounds:

- Known : 65 ± 19
- Unknown : 51 ± 20

8.2.4 The 4-Dimensional Fit Results

The four-dimensional fit results for data samples with $\mathcal{A}_{CP}$ of 0 and 1 and the corresponding projection plots are shown in Figure 8.14. The projection selection ranges are the same as in Chapter 6.

The systematic uncertainty on the signal event number introduced by fixing the charged and mixed rare event numbers in the fitter is now ± 9 events.

The fitter tests are now run as in Chapter 6 and Chapter 7.

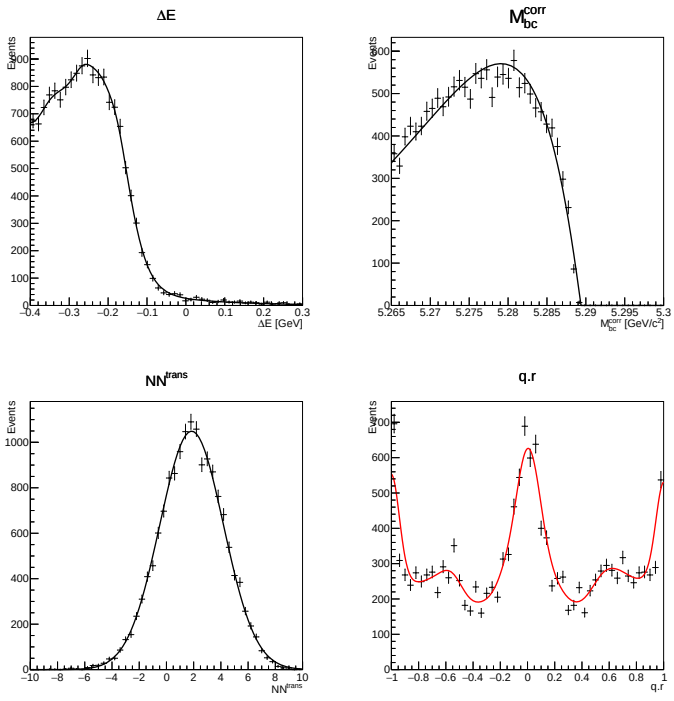

Figure 8.12: Showing the one-dimensional charged rare background PDFs.

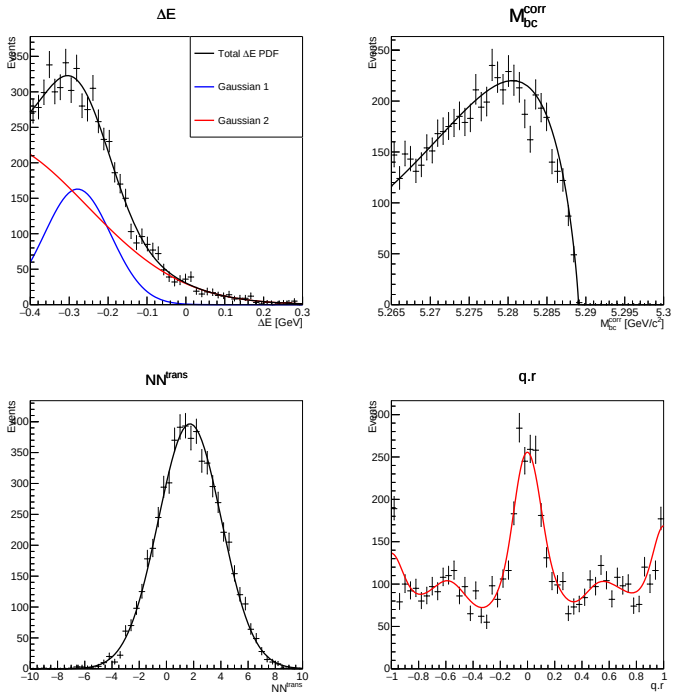

Figure 8.13: Showing the one-dimensional mixed rare background PDFs.

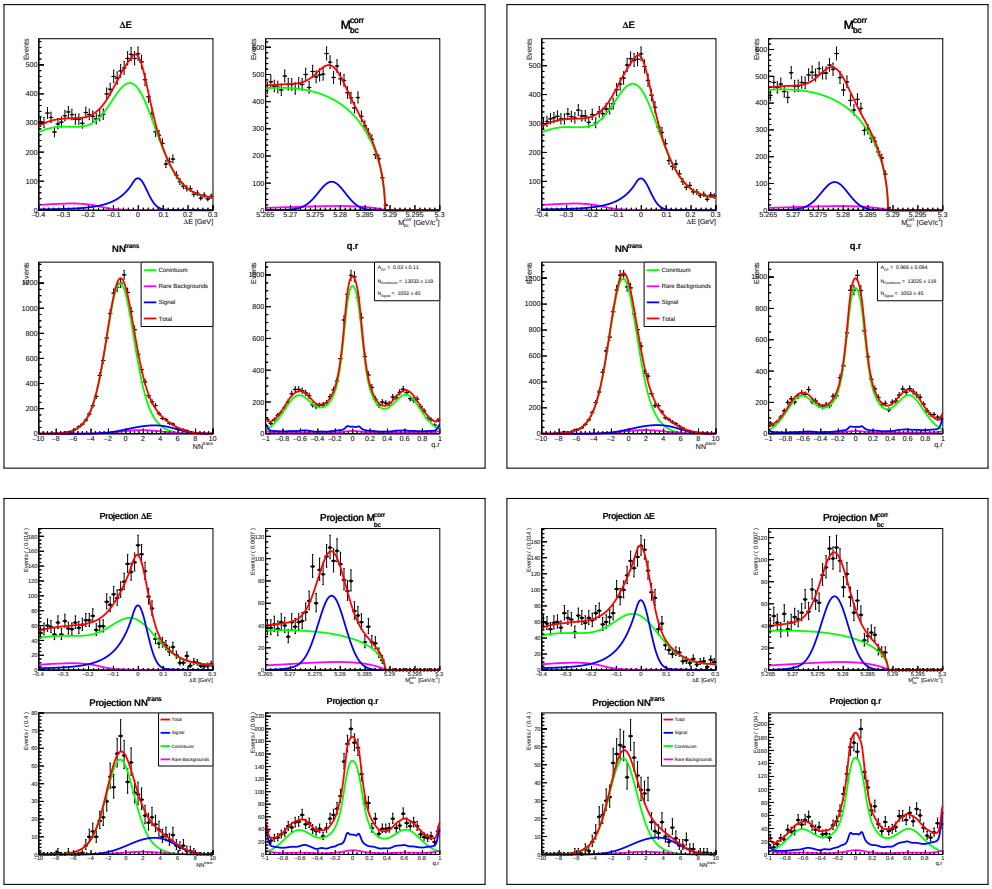

Figure 8.14: Showing the 4-dimensional fits (top row) and projections plots (bottom row) to data samples with $\mathcal{A}_{CP} = 0$ (left column) and $\mathcal{A}_{CP} = 1$ (right column).

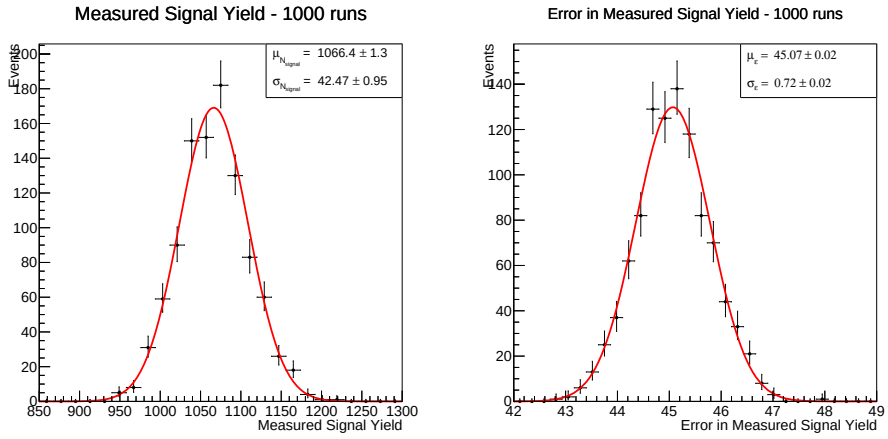

Figure 8.15: Showing the distributions in measured (left) and error in measured (right) signal yields, over one-thousand runs.

Signal Yield Measurement

The number of signal events measured over thousand runs has a mean of 1066.4±1.3 and a standard-deviation of 42.5±1.0. The distribution of the statistical uncertainty on signal yield returned by the fitter has a mean of 45.07 ± 0.02 and a standard-deviation of 0.72 ± 0.02. Both distributions are shown in Figure 8.15.

The pull distribution for the signal event number is shown in Figure 8.16. It has a mean of 0.31 ± 0.03 and a standard-deviation of 0.94 ± 0.02.

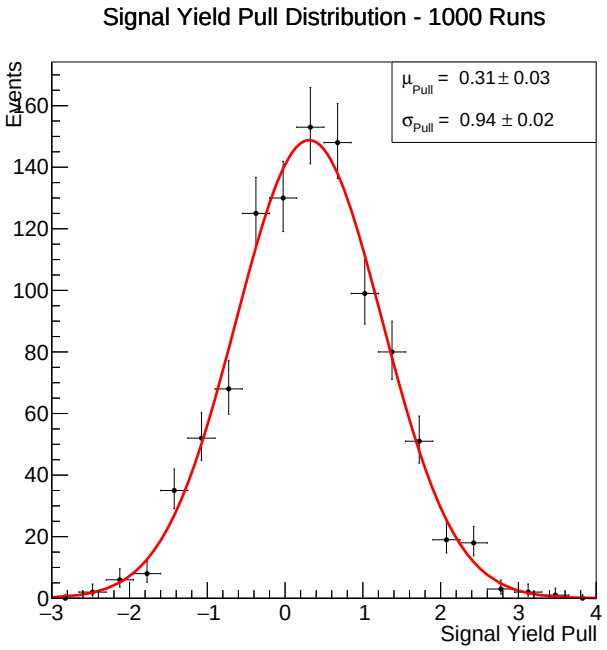

Figure 8.16: Showing the distribution in signal pulls over a thousand runs.

The measured signal event number versus the input signal data sample size is shown in Figure 8.17. It has a gradient of 0.989 ± 0.004 and a y-intercept of 24.7 ± 4.5. As can be seen from the pull and linearity test, the fitter is again consistently overestimating the signal event number. Shifting the fit result by -14.4 signal events would correct for this.

$\mathcal{A}_{CP}$ Measurement

To verify that the method of generating $\mathcal{A}_{CP}$ from the $\mathcal{A}_{CP} = 0$ signal validation dataset is valid, a signal dataset was generated (in EvtGen, see Chapter 3) with $\mathcal{A}_{CP} = +1$. For both methods of generating $\mathcal{A}_{CP}$, the distributions of the measured $\mathcal{A}_{CP}$ and the uncertainty on the measured $\mathcal{A}_{CP}$ are shown in Figure 8.18. These results show that there should be no issue in generating the $\mathcal{A}_{CP}$ from the $\mathcal{A}_{CP} = 0$ signal validation dataset.

The measured $\mathcal{A}_{CP}$ distribution over a thousand data samples has a mean of 0.006 ± 0.003 and a standard deviation of 0.108 ± 0.002. The distribution of the statistical uncertainties on the $\mathcal{A}_{CP}$ has a mean of 0.1058 ± 0.0001 and a standard-deviation of 0.0031 ± 0.0001. The measured $\mathcal{A}_{CP}$ is slightly overestimated and the

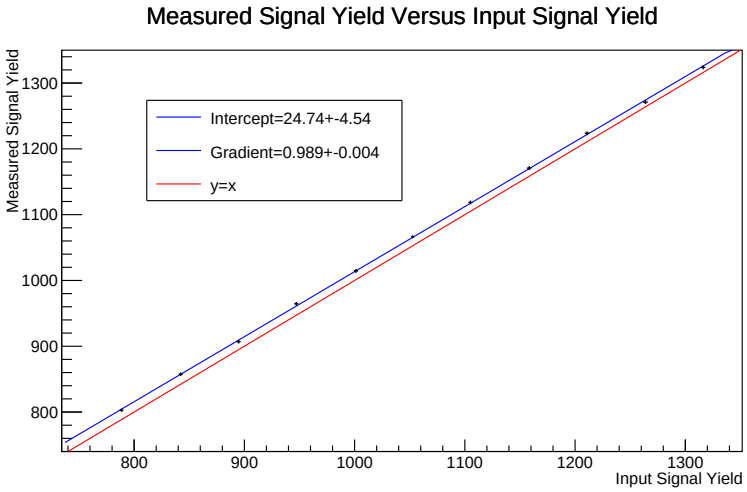

Figure 8.17: Showing the mean of the measured signal yields plotted against the mean of the input signal yields.

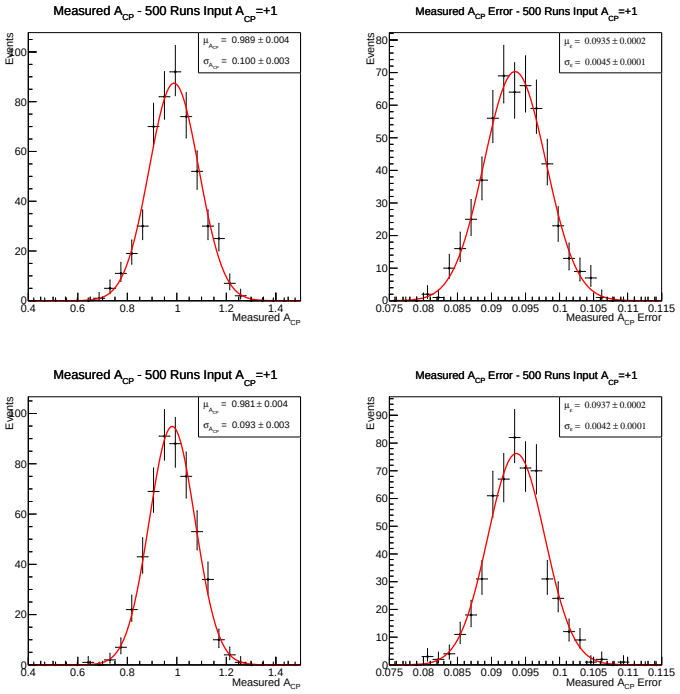

Figure 8.18: Showing the distribution in measured $\mathcal{A}_{CP}$ (left column) and the error in measured $\mathcal{A}_{CP}$ (right column) for five-hundred data samples with $\mathcal{A}_{CP} = +1$, where the $\mathcal{A}_{CP}$ is generated either in Evtgen (top row) or from the $\mathcal{A}_{CP} = 0$ dataset (bottom row).

statistical uncertainty is slightly underestimated. Both the quoted fitter uncertainty and the standard-deviation in the measured $\mathcal{A}_{CP}$ values see an improvement on the results in Chapter 6, but its is not clear that the use of the adversarial neural

network has seen an improvement on this front. The distributions of measured $\mathcal{A}_{CP}$ and error in measured $\mathcal{A}_{CP}$ are shown in Figure 8.19.

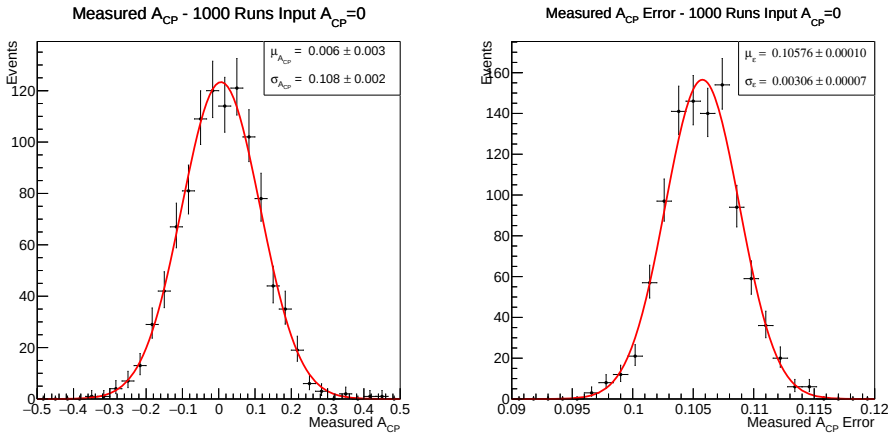

Figure 8.19: Showing the distribution in measured (left) $\mathcal{A}_{CP}$ and the error (right) distribution over a thousand runs. The input data is of $\mathcal{A}_{CP} = 0$.

The pull distribution on the measured $\mathcal{A}_{CP}$ is shown in Figure 8.20. The pull distribution has a mean of 0.06 ± 0.03 and a standard-deviation of 1.02 ± 0.02. The standard deviation shows that the statistical uncertainty is being accurately quoted by the fitter. The mean on the other hand is slightly high.

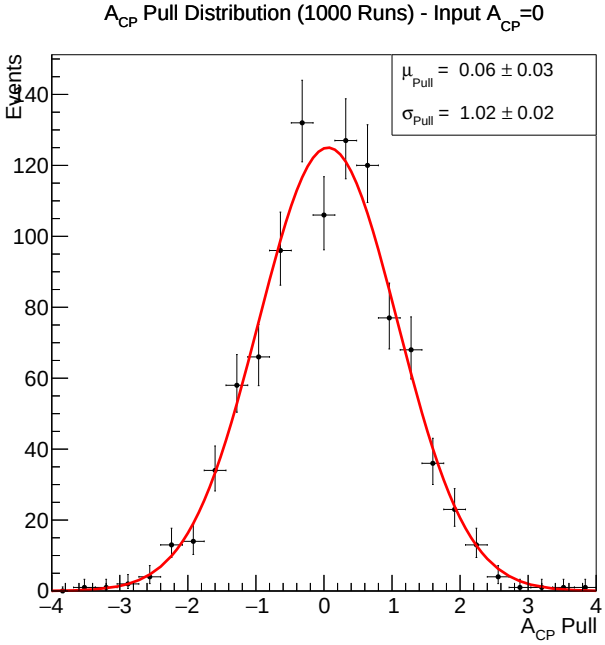

Figure 8.20: Showing the distribution in $\mathcal{A}_{CP}$ pull over a thousand runs.

The linearity test, the mean of the measured $\mathcal{A}_{CP}$ versus the $\mathcal{A}_{CP}$ of the data

sample is shown in Figure 8.21. It has a y-intercept of 0.007 ± 0.001 and a gradient of 0.975 ± 0.002.

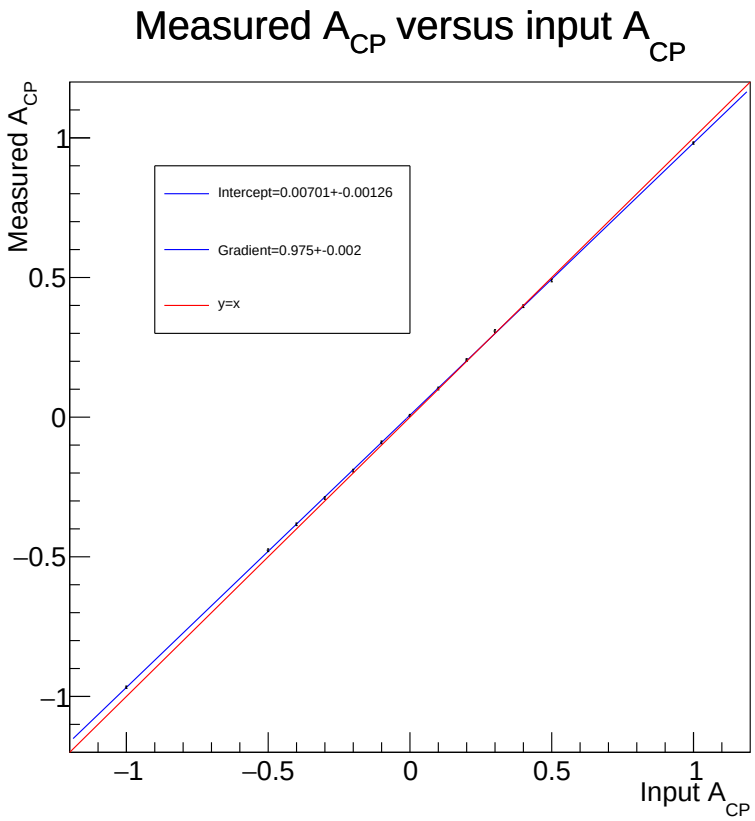

Figure 8.21: Showing the mean of the measured $\mathcal{A}_{CP}$ measurements against $\mathcal{A}_{CP}$ of the data samples..

Overall this model shows a good fit to the data, a good statistical uncertainty on the $\mathcal{A}_{CP}$; 0.106 ± 0.003 as compared to 0.111 ± 0.003 (from the NeuroBayes neural network analysis in Chapter 6), where the uncertainties are the standard-deviations on the statistical uncertainties.

Sadly no clear improvement over the result from the original, non-adversarialy trained neural network (0.106 ± 0.003) in Chapter 7 is seen, and is not statistically significant. A greater improvement was expected as the number of continuum events in the signal region in ΔE (the range ± 0.1) decreased from 6106 ± 15 to 5256 ± 12 (while signal remained relatively constant, going from 846 ± 46 to 834 ± 45 events). We investigate if this is due to peculiarities in the fitter and the large remaining $\Delta E - NN$ correlations.

8.2.5 Generating Correlated Data and the 4-Dimensional Fit

The data samples for the four-dimensional fitter were selected from the Monte Carlo datasets for signal and the rare-backgrounds, and so have the $\Delta E - NN^{trans}$ correla-

tion present. As continuum was generated to the one-dimensional PDF distributions, the correlations expected with real data would not be present. A complete study of the model would need to take into account these correlations in continuum. Investigation into fitting ΔE and NN^{trans} with a two-dimensional PDF for each channel, and using these in the fitter show a worse model, possibly due to low statistics in the PDF tail regions. The four-dimensional PDF (as a product of one-dimensional PDFs) is kept, investigating the biases and systematic uncertainties introduced by the correlations in the data.

The continuum two-dimensional $\Delta E - NN^{trans}$ distribution is modelled with a two-dimensional kernel density estimation function (RooNDKeysPdf), with mirroring at all edges, a smoothing of $\rho = 2$, and using adaptive kernels (the width of each kernel varying with event density). Figure 8.22 shows this continuum PDF, $f_{continuum}^{\Delta E, NN^{trans}}\left(\Delta E, NN^{trans}\right)$.

Now the four-dimensional fit (the products of the same one-dimensional PDFs above) tests are repeated, using the exact same procedure, but now with the continuum data sample generated to, but not fit with:

$$f_{continuum}^{4d}\left(\Delta E, M_{bc}^{corr}, NN^{trans}, q.r\right) = f_{continuum}^{\Delta E, NN^{trans}}\left(\Delta E, NN^{trans}\right) \cdot f_{continuum}^{M_{bc}^{corr}}\left(M_{bc}^{corr}\right) \cdot f_{continuum}^{q.r}\left(q.r\right) \quad (8.3)$$

The systematic uncertainty in the signal yield from fixing the rare backgrounds is now ± 10 events.

Signal Yield Measurement

The signal yield measurement and error distributions are shown in Figure 8.23. The mean of the signal yield measurements is 1132.2 ± 1.5 and the standard-deviation is 47.4 ± 1.1. Clearly the signal yield now has a much bigger bias. The mean of the signal yield errors is 46.35 ± 0.02 and the standard-deviation is 0.78 ± 0.02. As the pull distribution shows (see Figure 8.24)The uncertainty is accurately predicted, with a pull standard-deviation of 1.00 ± 0.02 but as expected, the pull mean of 1.72 ± 0.03 confirms that including the $\Delta E - NN^{trans}$ correlations in all of the data (as real data would have) biases the 4-dimensional fitter.

The linearity check, shown in Figure 8.25, with a gradient of 0.996 ± 0.005 (within the error range of one) and a y-intercept of 82.1 ± 4.9 shows that the fit very consistently overestimates the measured signal yield.

$\mathcal{A}_{CP}$ Measurement

The mean of the measured $\mathcal{A}_{CP}$ values is 0.014 ± 0.003, and the standard-deviation is 0.102 ± 0.002. The mean value sees an increase compared to the fits with continuum data generated without correlations, but it is not large. The mean of the statistical uncertainties on the $\mathcal{A}_{CP}$ measurements is 0.1035 ± 0.0001 and its standard deviation is 0.000315 ± 0.00007. This is *smaller* than the statistical uncertainty on the $\mathcal{A}_{CP}$ from the fits with continuum generated without correlations. This could just be a statistical fluctuation as the values of 0.1035 ± 0.0032 and 0.1058 ± 0.0031 easily

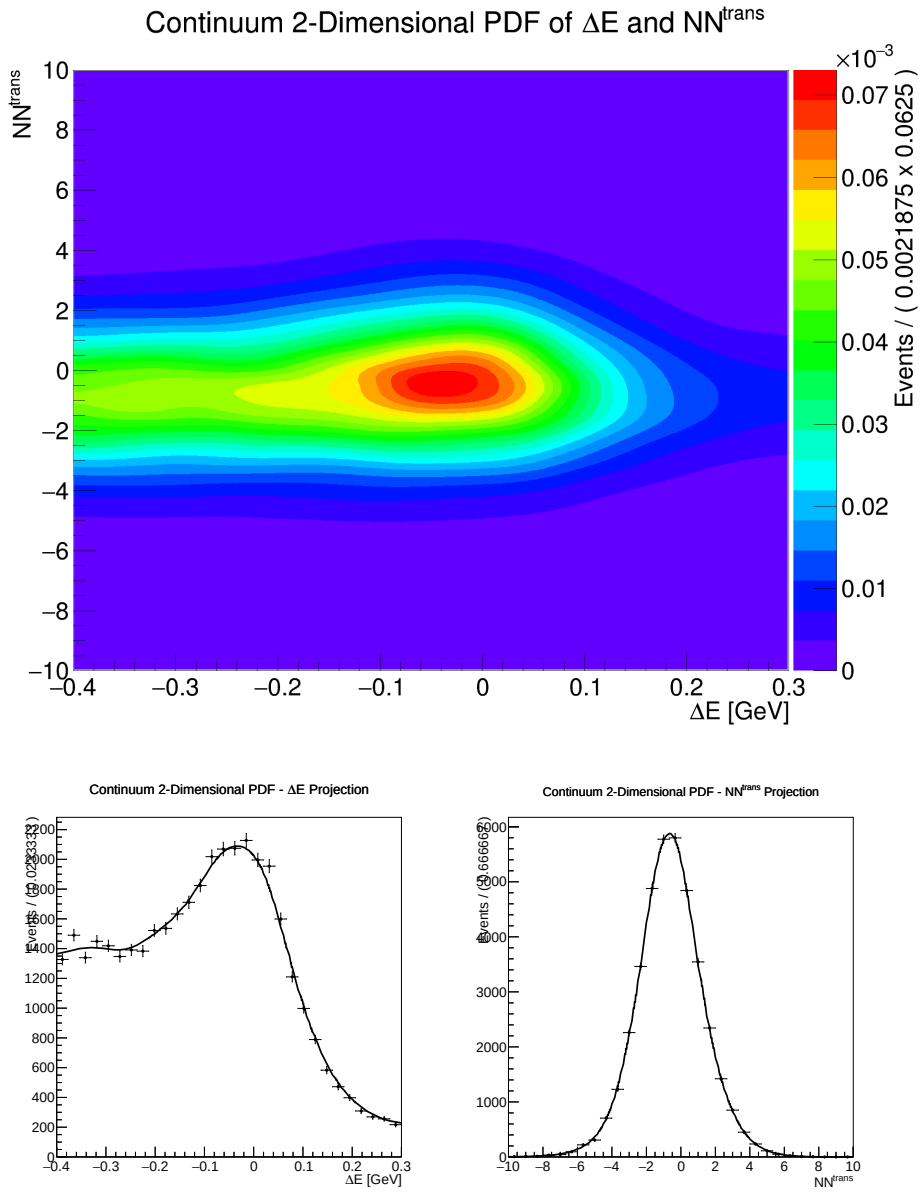

Figure 8.22: Showing the two-dimensional continuum PDF of ΔE and NN^{trans} (top), and its projections (along with the continuum MC data) in ΔE (bottom left) and NN^{trans} (bottom right).

overlap within one-standard deviation. The distribution in the measured $\mathcal{A}_{CP}$ and the statistical uncertainty in the measured $\mathcal{A}_{CP}$ are shown in Figure 8.26.

The distribution of $\mathcal{A}_{CP}$ pulls is shown in Figure 8.27, with a mean of 0.14 ± 0.03 and a standard-deviation of 0.99 ± 0.02. Clearly a worse fit than for fits with continuum generated without correlations, but still a good model overall.

Finally the mean of the measured $\mathcal{A}_{CP}$ values is plotted against the data sample $\mathcal{A}_{CP}$ values in Figure 8.28. With a gradient of 0.939 ± 0.002 and a y-intercept of 0.012 ± 0.001, there is a clear bias as to be expected, but not large.

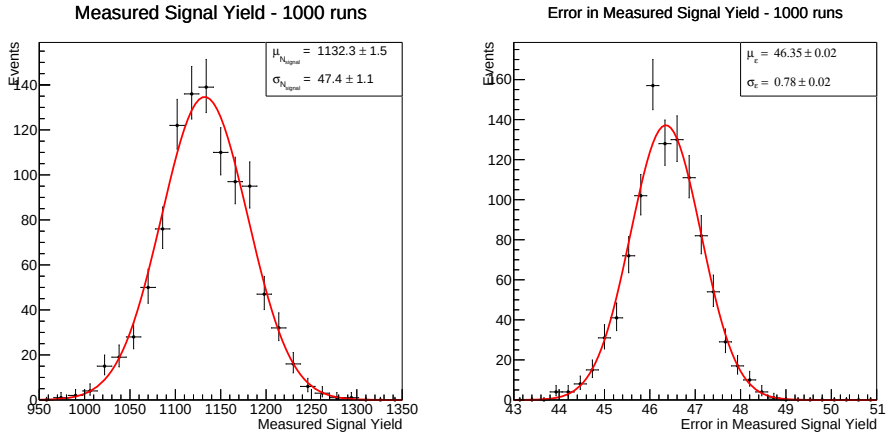

Figure 8.23: Showing the distributions in measured (left) and error in measured (right) signal yields, over one-thousand runs.

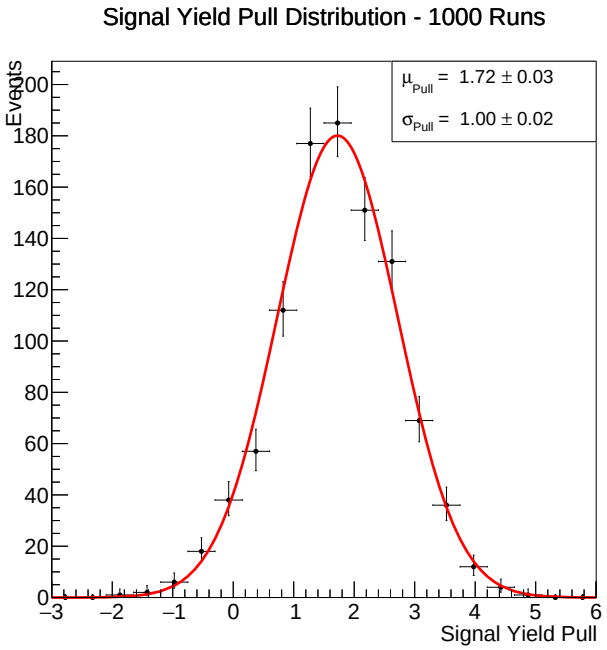

Figure 8.24: Showing the distribution in signal pulls over a thousand runs.

Biases in the 4-Dimensional Fit

To investigate the systematic uncertainty in the the $\mathcal{A}_{CP}$ measurement, the mean of the measured values for the fits with the data generated with the continuum correlations is subtracted from the mean of the measured values for the fits generated without the continuum correlations. This is repeated for each input $\mathcal{A}_{CP}$. In other words, the $\mathcal{A}_{CP}$ for the data in Figure 8.28 is subtracted from the data in Figure 8.21. The result is shown in Figure 8.29. If we assume the systematic error to be

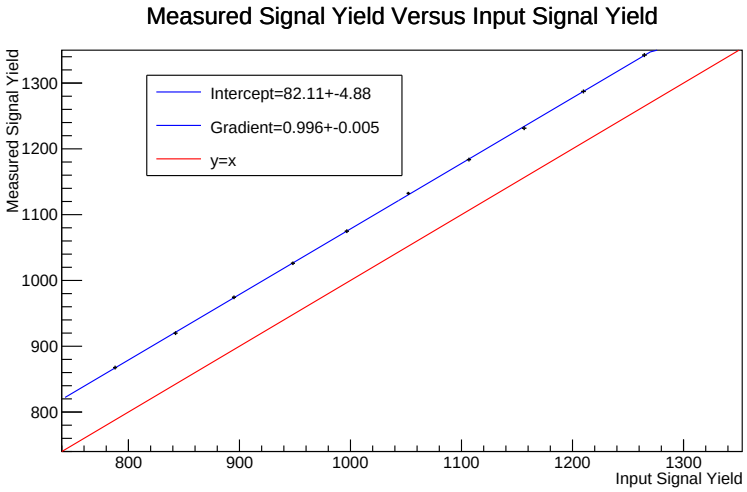

Figure 8.25: Showing the mean of the measured signal yields plotted against the mean of the input signal yields.

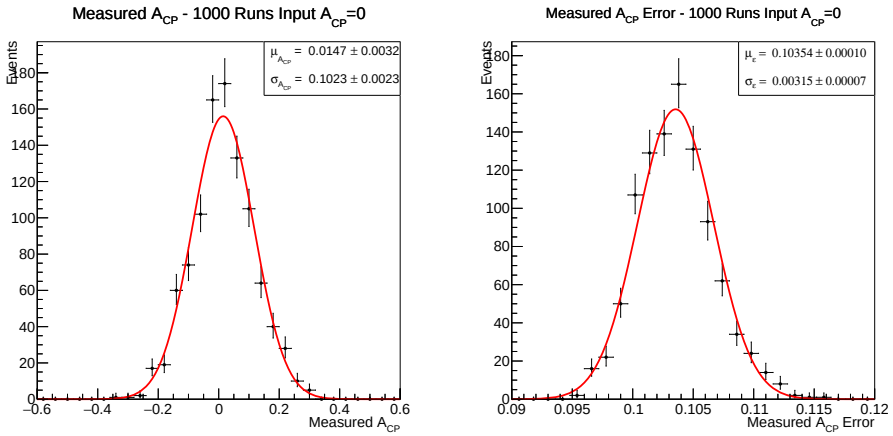

Figure 8.26: Showing the distribution in measured (left) $\mathcal{A}_{CP}$ and the error (right) distribution over a thousand runs. The input data is of $\mathcal{A}_{CP} = 0$.

half of the correction due to the bias, we get:

$$\epsilon^{syst}_{\mathcal{A}_{CP}} = \frac{0.0353 \cdot \mathcal{A}_{CP} - 0.00451}{2} \tag{8.4}$$

Assuming that the fitter will be shifted and skewed by the average of the linearity tests gradients and y-intercepts. So if the $\mathcal{A}_{CP}$ value is measured at (the latest Belle result) the value of 0.14, the systematic uncertainty in $\mathcal{A}_{CP}$ would be ± 0.0004 (alternatively, taking the average of the latest Belle and BaBar values gives a systematic uncertainty of ± 0.002).

Similarly, repeating the same process for the measured signal yields gives (see

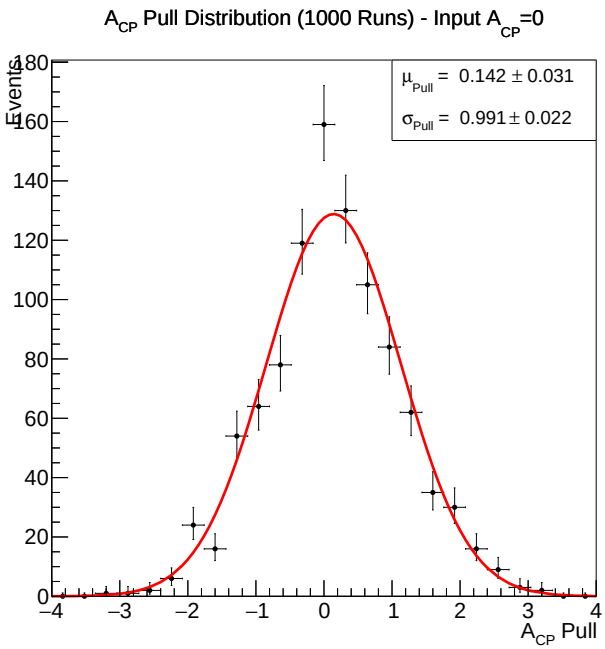

Figure 8.27: Showing the distribution in $\mathcal{A}_{CP}$ pull over a thousand runs.

Figure 8.30):

$$\epsilon^{syst}_{N_{signal}} = \frac{0.009 \cdot N_{sig} + 55.72}{2} \tag{8.5}$$

So assuming 1052 signal events, this would give a systematic uncertainty of ±32 events.

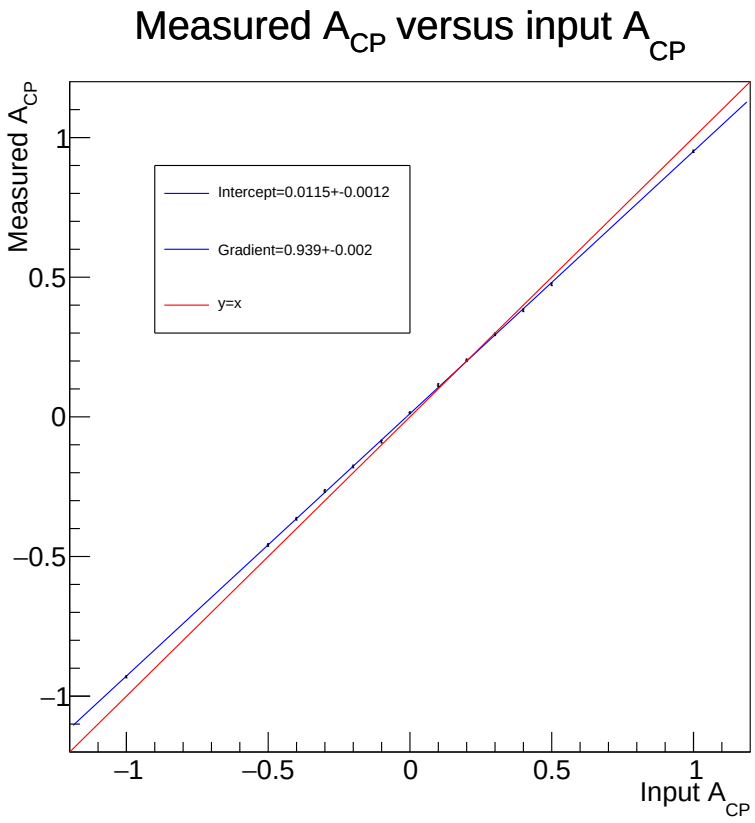

Figure 8.28: Showing the mean of the measured $\mathcal{A}_{CP}$ measurements against $\mathcal{A}_{CP}$ of the data samples..

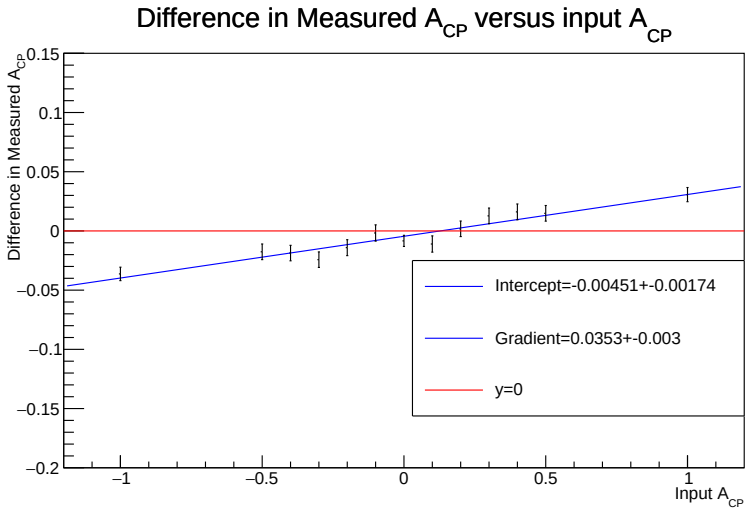

Figure 8.29: Showing the difference between the means of the measured $\mathcal{A}_{CP}$ values for both methods of generating continuum data, plotted against the $\mathcal{A}_{CP}$ of the data.

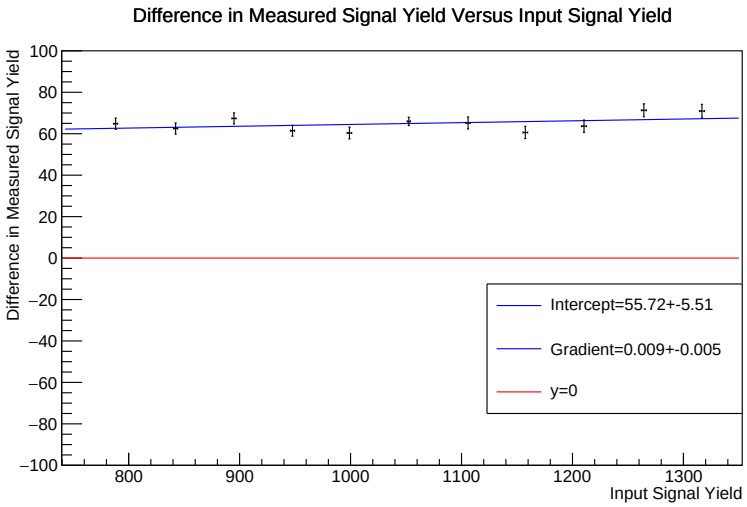

Figure 8.30: Showing the difference between the means of the measured signal yield values for both methods of generating continuum data, plotted against the mean of the signal sample size.

9|Conclusion

A new time-independent method of measuring the direct $\mathcal{A}_{CP}$ in $B^0 \to K_S\pi^0$ decays whilst also providing a measurement on the branching ratio is presented. This is applied to Monte Carlo data and sees an improved result over the latest Belle measurement. Due to the correlations between ΔE and NN^{trans} that would be expected in real data, the results of the four-dimensional fit to the data samples generated with these correlations in continuum are used. The latest Belle measurement of $0.14 \pm 0.13(stat)$ was a time-dependent study and included information from the measurement of $\mathcal{A}_{CP}$ in $B^0 \to K_L\pi^0$. The Monte Carlo studies give a statistical uncertainty in the measured $\mathcal{A}_{CP}$ of 0.1035 ± 0.0032, which as compared to 0.13 shows a significant improvement.

Given a measured number of $K_S\pi^0$ events (N_{signal}), the branching ratio of $B^0 \to K^0\pi^0$ is given by:

$$\mathcal{B}\left(B^0 \to K^0\pi^0\right) = \frac{N_{signal}}{N_{B\bar{B}} \times R_{B^0\bar{B}^0} \times \epsilon} \tag{9.1}$$

Where $N_{B\bar{B}} = (771.581 \pm 10.566) \times 10^6$ is the total number of $B\bar{B}$ events produced at Belle, $R_{B^0\bar{B}^0} = 0.486 \pm 0.006$ being the fraction of $B\bar{B}$ that are $B^0\bar{B}^0$, and ϵ is the final signal efficiency (28.33% of signal events remaining in the fitting region after all other selections have been placed). Given the statistical uncertainty in the measured signal yield of 46.4 ± 0.8 events, the statistical uncertainty in the branching ratio is then $(4.4 \pm 0.1) \times 10^{-7}$. Compared to the statistical uncertainty in the latest Belle branching ratio measurement of 4.6×10^{-7} there is a slight improvement, although clearly the power of this method is in the $\mathcal{A}_{CP}$ measurement. A full analysis of the systematic uncertainties would be needed.

The benefits of using TensorFlow to build an optimised neural network is clear, with the best figure of merit (of separation between signal MC and continuum MC) being 17.3 ± 0.4 compared to that of the NeuroBayes network of 13.2 ± 0.4.

Both networks perform slightly worse with off-resonance data but the difference in performance is small and present in both. The TensorFlow AUC for signal with continuum MC and with off-resonance being 0.947 and 0.938 respectively. Similarly for NeuroBayes the AUC from signal with continuum MC is 0.909 and for signal with off-resonance is 0.891.

This increased classifying power between signal and continuum comes with the unforeseen consequence of introducing correlations between the neural network output and ΔE. This limits the reduction in the statistical uncertainty in the $\mathcal{A}_{CP}$ measurement. The statistical uncertainty in the measurement (from data generated without the $\Delta E - NN$ correlations for continuum) is reduced from 0.111 ± 0.004 to

0.106 ± 0.003 by using the optimised TensorFlow neural network.

The use of adversarial neural networks to reduce the $\Delta E - NN$ correlations was investigated and it was found that the correlations could be reduced at a small cost to the classification power. The correlations could be reduced to be smaller than that of NeuroBayes whilst keeping significantly greater separation between signal and continuum.

The four-dimensional fit was then run on data from the updated neural network, which while reduced, still had large $\Delta E - NN$ correlations (in order to maintain optimal continuum suppression). The statistical uncertainty did not decrease further. This could be due to the conservative choice in correlation reduction. More work should be done to optimise the weighting that is given to the adversarial neural network in training. The benefits from using the data processed by the TensorFlow neural networks are significant, these should be translated to the four-dimensional fit. Further investigations into fitting ΔE and NN^{trans} with the two-dimensional pdfs in all channels should be further investigated. Even so, the $(20.4 \pm 2.5)\%$ reduction in statistical uncertainty from the latest Belle result is significant.

The next step in the analysis is to validate the procedure on a control mode. Testing the differences between real data and signal Monte Carlo is vital as (other than the validation with off-resonance data) all training and testing has been performed using Monte Carlo data, and the potential difference between MC and real data should be investigated. Performing the measurement on real $B^0 \to K_S\pi^0$ data is of course the main aim of this study.

Possible next steps to improve on the uncertainty in $\mathcal{A}_{CP}$ would be to further the work towards best utilising the adversarial neural network. Also, the input parameters into the classifying neural networks could be further investigated. There is now freedom to add more variables into the neural network that provide distinguishing information between signal and continuum that may otherwise be ignored due to their correlations with ΔE, M_{bc}^{corr}, and especially $q.r$.

As in the latest Belle result, incorporating $K_L\pi^0$ would further constrain the $\mathcal{A}_{CP}$ measurement so is an obvious extension to this study.

The increased luminosity from the upgraded SuperKEKB, planned to have forty times the peak luminosity of KEK, and the improved Belle II detector will provide ample statistics, allowing for a much improved branching ratio and $\mathcal{A}_{CP}$ measurement.

A|Rare Decay Modes

The most common rare decay modes in the charged (A.1) and mixed (A.2) rare datasets after reconstruction and B-meson candidate selection are investigated. The observed number of events refers to the number of events of a given decay mode before any further selections are placed on ΔE or M_{bc}^{corr}, or any neural network processing. Assuming that the percentage of events which pass the neural network selection is the same for all of the decay modes (not necessarily true but a good approximation), the branching ratios (along with the total number of $B\bar{B}$ events) and, observed number of events and neural network selection efficiency give ϵ, the fraction of events of a given decay mode that pass all selections. This is done in order to propagate the uncertainties for these 'known' decay modes.

The remaining events comprise the number of events from 'unknown' decay modes, where the uncertainty is assumed to be 40%. Table A.3 shows the total, 'known' and 'unknown' expected event numbers in the charged and mixed streams, for the three datasets (from the different neural networks) used in fitting.

Decay Mode	Branching Ratio	Observed Events	$\epsilon(\%)$	Expected Number of Events		
				NB	TF(NoAdv)	TF(Adv)
$\rho(770)^-[\pi^-\pi^0]K_S[\pi^-\pi^+]$	$(2.77 \pm 0.52) \times 10^{-6}$	287.1	13.8 ± 2.6	102 ± 19	86 ± 16	95 ± 18
$K^{*+}[K_S[\pi^-\pi^+]\pi^+]\pi^0$	$(1.42 \pm 0.33) \times 10^{-6}$	224.8	21.2 ± 4.9	80 ± 19	68 ± 16	74 ± 17
$K_S[\pi^-\pi^+]\pi^-\pi^0$	$(1.14 \pm 1.14) \times 10^{-5}$	102.2	1.2 ± 1.2	36^{+37}_{-36}	31^{+32}_{-31}	34^{+35}_{-34}
$K^{*-}[K_S[\pi^-\pi^+]\pi^-]\gamma$	$(7.28 \pm 0.31) \times 10^{-6}$	26.8	0.5 ± 0.1	9.5 ± 0.4	8.0 ± 0.4	8.9 ± 0.4

Table A.1: Showing the most common charged rare decays. The square brackets are the further decays showing the full decay chain, to which the branching ratio corresponds. The observed events and efficiencies ϵ are calculated without the final fitting variable and neural network selections. NB refers to the the data for the fit processed by NeuroBayes (Chapter 6), TF(NoAdv) refers to that of the TensorFlow neurarl network (Chapter 7) and TF(Adv) to the TensorFlow network retrained with the adversary (Chapter 8).

Decay Mode	Branching Ratio	Observed Events	$\epsilon(\%)$	Expected Number of Events		
				NB	TF(NoAdv)	TF(Adv)
$f_0(980)[\pi^0\pi^0]K_S[\pi^-\pi^+]$	$(1.21 \pm 0.16) \times 10^{-6}$	52.2	5.7 ± 0.8	17 ± 3	14 ± 3	16 ± 3
$f_2(1270)[\pi^0\pi^0]K_S[\pi^+\pi^-]$	$(2.63 \pm 1.22) \times 10^{-7}$	44.8	22.7 ± 10.5	15 ± 10	12 ± 8	13 ± 9
$K^*(1680)^0[K_S[\pi^-\pi^+]\pi^0]\pi^0$	$(1.67 \pm 1.67) \times 10^{-7}$	33.6	26.8 ± 26.8	11^{+16}_{-11}	9^{+13}_{-9}	10^{+14}_{-10}
$K_2^*(1430)^0[K_S[\pi^-\pi^+]\pi^0]\pi^0$	$(8.03 \pm 8.03) \times 10^{-8}$	18.1	30.1 ± 0.3	6^{+8}_{-6}	5^{+7}_{-5}	5^{+8}_{-5}
$K_S[\pi^0\pi^0]K_S[\pi^-\pi^+]$	$(1.29 \pm 0.17) \times 10^{-7}$	16.8	17.4 ± 2.3	6 ± 1	5 ± 1	6 ± 1
$K^*(892)^0[K_S[\pi^-\pi^+]\pi^0]\pi^0$	$(3.80 \pm 0.69) \times 10^{-7}$	15.5	5.4 ± 1.0	5 ± 1	4 ± 1	5 ± 1
$K^*(892)^0[K_S[\pi^-\pi^+]\pi^0]\gamma$	$(4.98 \pm 0.17) \times 10^{-6}$	12.9	0.34 ± 0.05	4.3 ± 0.6	3.5 ± 0.5	3.9 ± 0.5
$f_0(1710)[\pi^0\pi^0]K_S[\pi^-\pi^+]$	$(9.13 \pm 1.86) \times 10^{-8}$	11.8	17.2 ± 3.5	4 ± 1	3 ± 1	4 ± 1
$K^*(892)^-[K_S[\pi^-\pi^+]\pi^-]\rho(770)^+[\pi^+\pi^0]$	$(2.37 \pm 0.60) \times 10^{-6}$	9.4	0.53 ± 0.17	3 ± 1	3 ± 1	3 ± 1

Table A.2: Showing the most common mixed rare decays. The square brackets are the further decays showing the full decay chain, to which the branching ratio corresponds. The observed events and efficiencies ϵ are calculated without the final fitting variable and neural network selections. NB refers to the the data for the fit processed by NeuroBayes (Chapter 6), TF(NoAdv) refers to that of the TensorFlow neurarl network (Chapter 7) and TF(Adv) to the TensorFlow network retrained with the adversary (Chapter 8).

		Total	Known	Unkown
NB	Charged	331 ± 3	228 ± 46	103 ± 41
	Mixed	126 ± 2	71 ± 21	55 ± 21
TF(NoAdv)	Charged	280 ± 2	193 ± 39	87 ± 35
	Mixed	104 ± 1	59 ± 17	46 ± 18
TF(Adv)	Charged	308 ± 2	212 ± 43	96 ± 38
	Mixed	115 ± 1	65 ± 19	51 ± 20

Table A.3: Showing the breakdown of expected rare event numbers into 'known' and 'unknown' decays for charged and mixed rare backgrounds. NB refers to the the data for the fit processed by NeuroBayes (Chapter 6), TF(NoAdv) refers to that of the TensorFlow neural network (Chapter 7) and TF(Adv) to the TensorFlow network retrained with the adversary (Chapter 8).

		Known		Unkown		Signal
		High	Low	High	Low	Systematic
NB	Charged	1066	1052	1064	1052	±10
	Mixed	1062	1059	1064	1052	
TF(NoAdv)	Charged	1076	1066	1075	1062	±9
	Mixed	1072	1070	1074	1068	
TF(Adv)(4dGen)	Charged	1073	1061	1069	1059	±9
	Mixed	1070	1060	1066	1062	
TF(Adv)(3dGen)	Charged	1136	1123	1138	1126	±10
	Mixed	1131	1131	1134	1126	

Table A.4: Showing the systematic uncertainties in measured signal yield introduced by fixing the rare backgrounds in the 4-dimensional fitter. NB, TF(NoAdv) and TF(Adv) refer to the 4-D fits to data processed by NeuroBayes, TensorFlow with no adversary and TensorFlow with adversary respectively. 4dGen and 3dGen refer to the 4-D fits to the data generated without and with continuum $\Delta E - NN^{trans}$ respectively. The values are the mean of the measured signal yield measurements over 500 runs when the known or unknown components of the charged or mixed rare backgrounds are run at their uncertainty limits. The 'Signal Systematic' is the systematic uncertainty calculated by combining the mean measured signal yield differences (divided by two) between each high and low measurement, propagated accordingly.

B|Scatter Plots of the Fitting Variables

B.1 Data Processed by NeuroBayes

M_{bc}^{corr}	NN^{trans}	$q.r$	
2.5%	3.8%	0.2%	ΔE
	0.0%	0.2%	M_{bc}^{corr}
		0.0%	NN^{trans}

Table B.1: Showing the (absolute) correlations between the four fitting variables, for the signal data processed by NeuroBayes.

M_{bc}^{corr}	NN^{trans}	$q.r$	
1.2%	4.4%	0.3%	ΔE
	1.6%	0.0%	M_{bc}^{corr}
		0.0%	NN^{trans}

Table B.2: Showing the (absolute) correlations between the four fitting variables, for the continuum data processed by NeuroBayes.

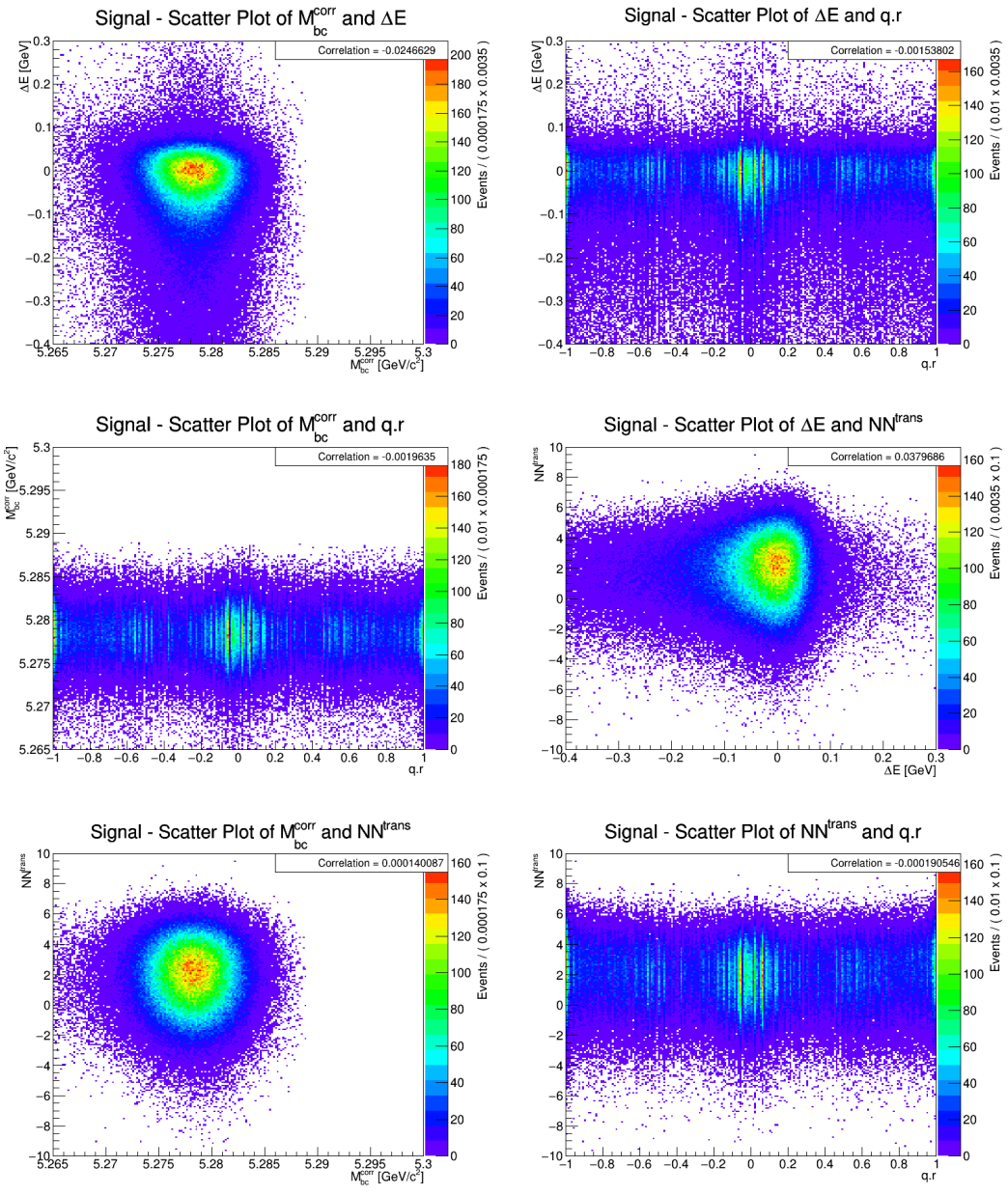

Figure B.1: The signal scatter plots in every pair of fitting dimensions.

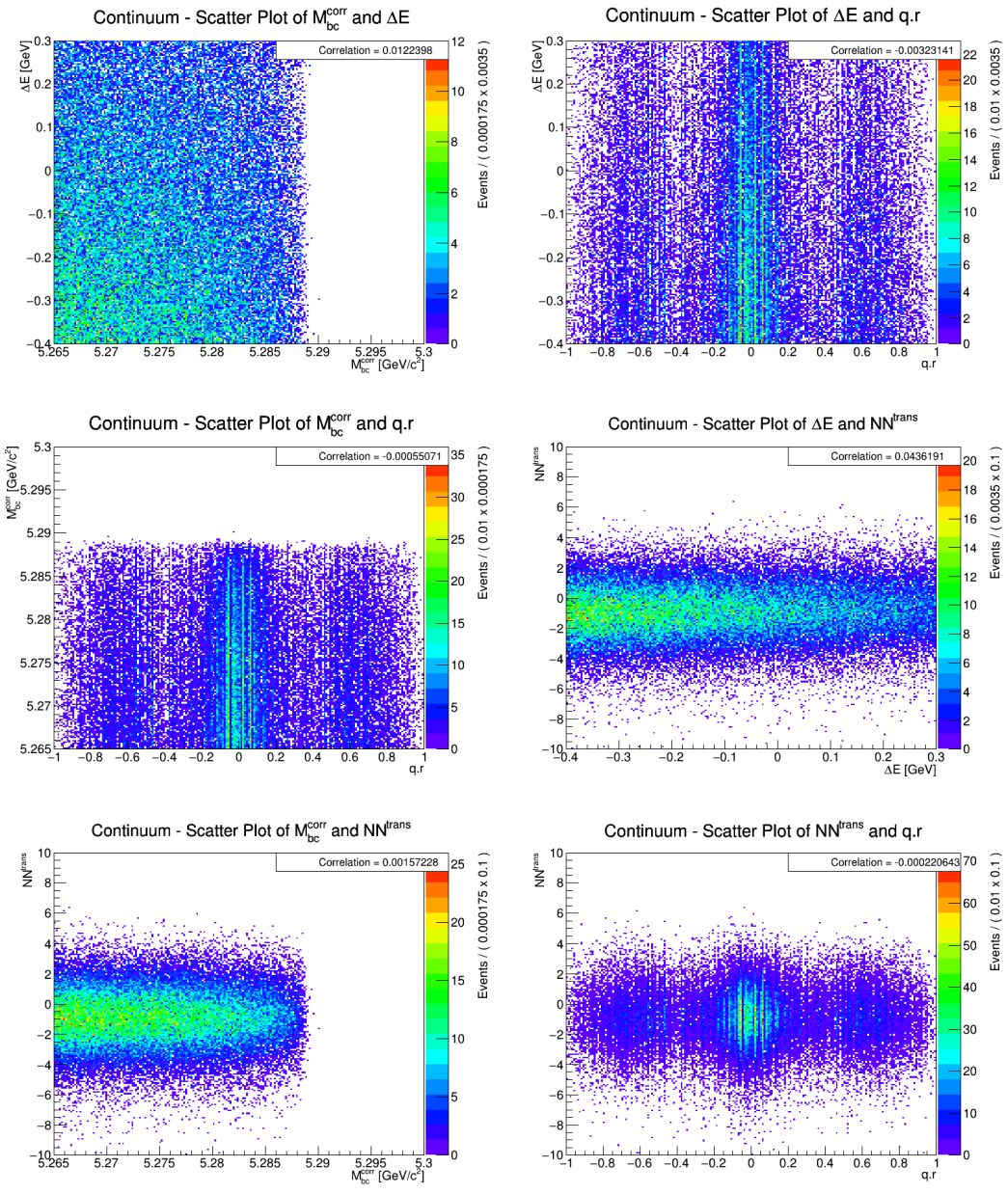

Figure B.2: The continuum scatter plots in every pair of fitting dimensions.

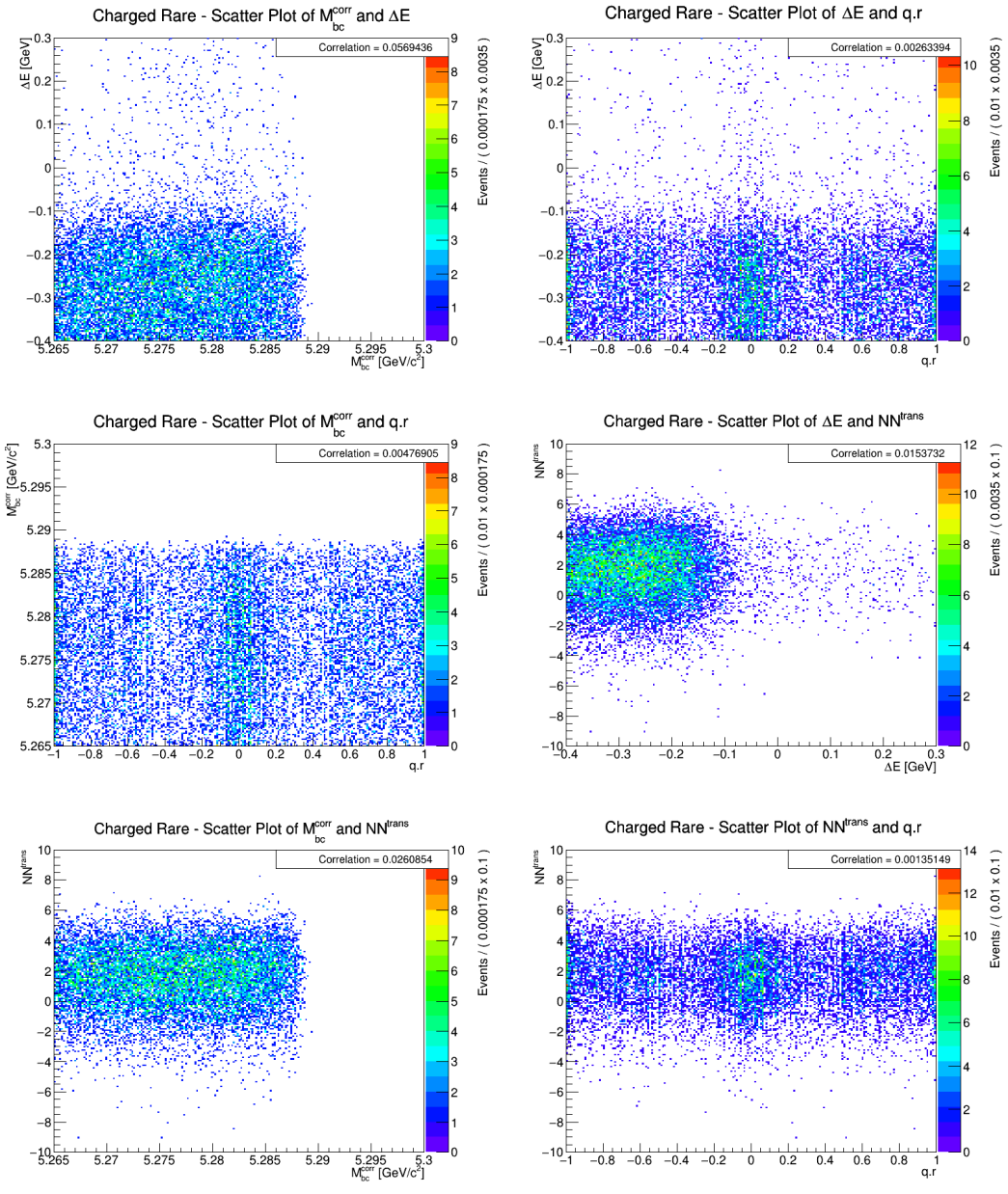

Figure B.3: The charged rare scatter plots in every pair of fitting dimensions.

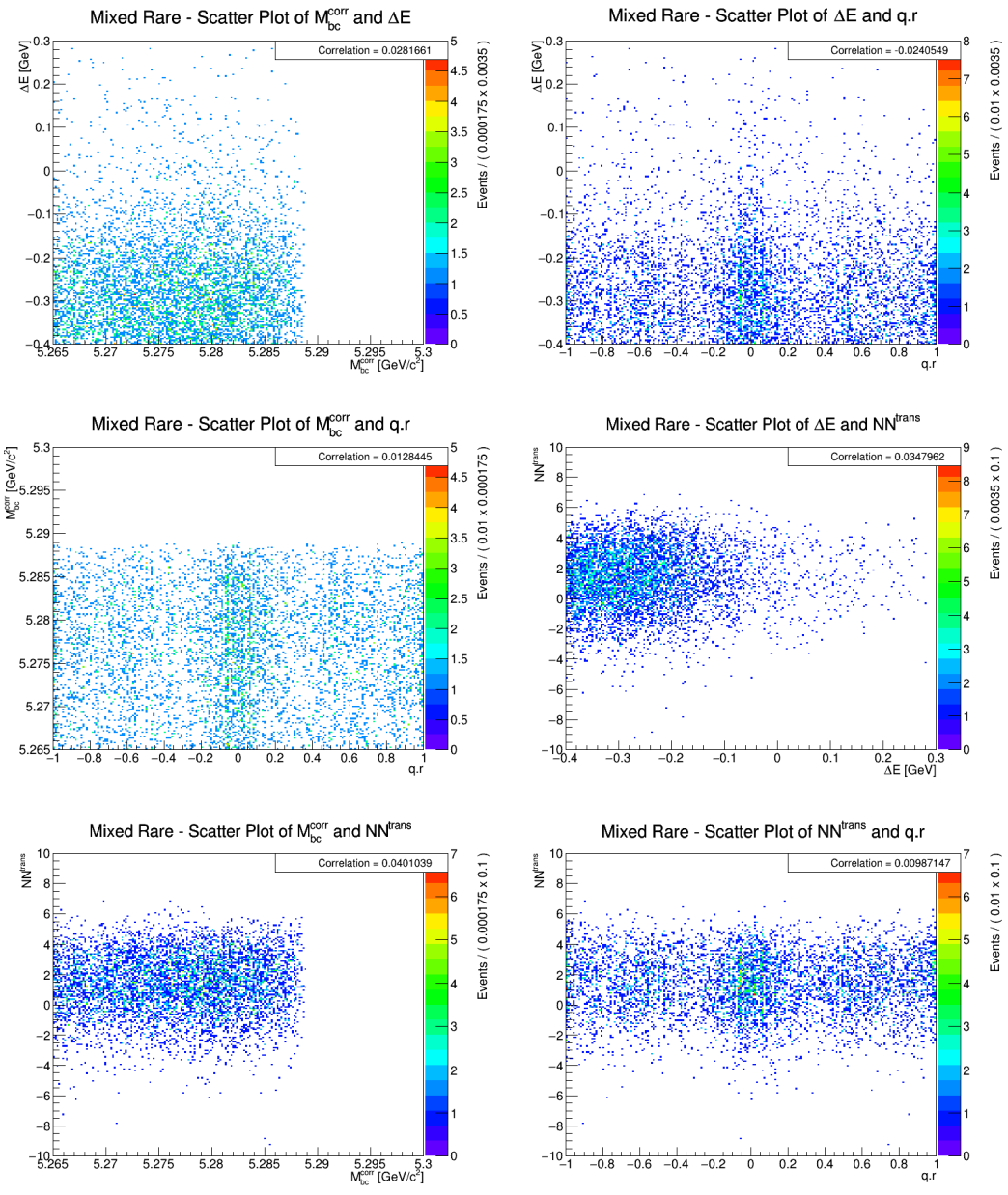

Figure B.4: The mixed rare scatter plots in every pair of fitting dimensions.

B.2 Data Processed by the TensorFlow Neural Network

M_{bc}^{corr}	NN^{trans}	$q.r$	
2.8%	17.3%	0.3%	ΔE
	2.3%	0.2%	M_{bc}^{corr}
		0.1%	NN^{trans}

Table B.3: Showing the (absolute) correlations between the four fitting variables, for the signal data processed by the TensorFlow neural-network (no adversary).

M_{bc}^{corr}	NN^{trans}	$q.r$	
0.4%	7.0%	0.0%	ΔE
	0.5%	0.6%	M_{bc}^{corr}
		0.2%	NN^{trans}

Table B.4: Showing the (absolute) correlations between the four fitting variables, for the continuum data processed by the TensorFlow neural-network (no adversary).

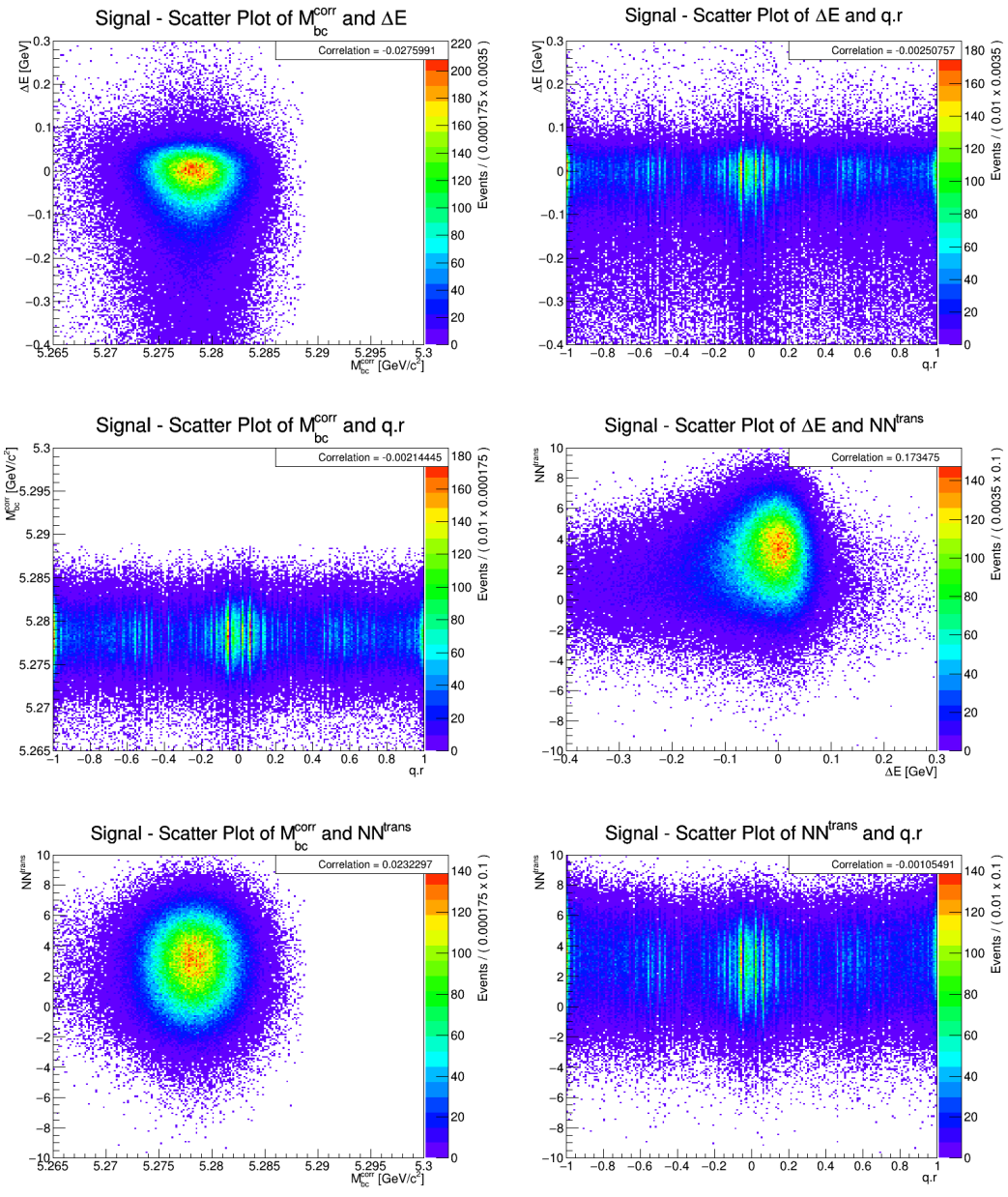

Figure B.5: The signal scatter plots in every pair of fitting dimensions.

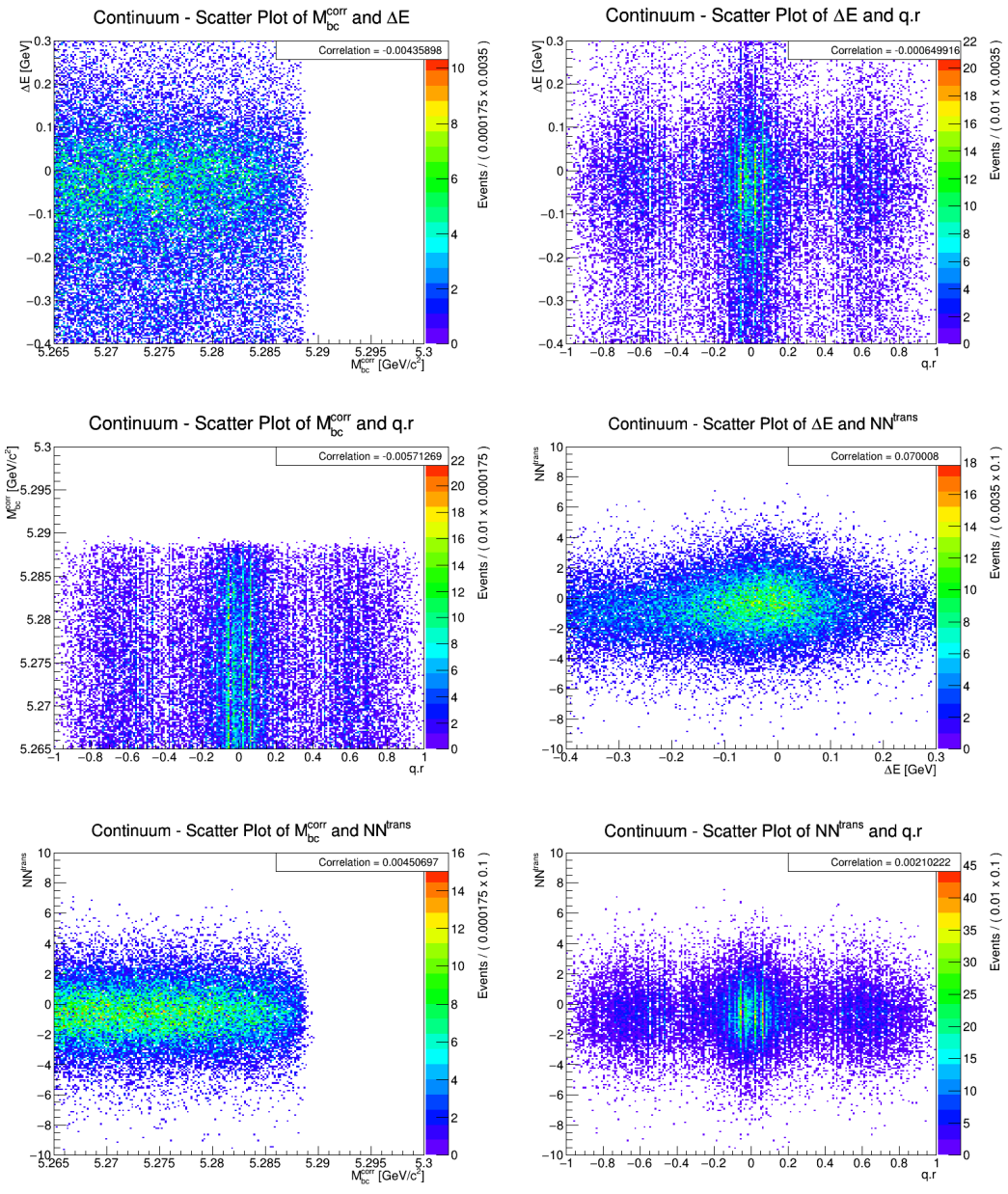

Figure B.6: The continuum scatter plots in every pair of fitting dimensions.

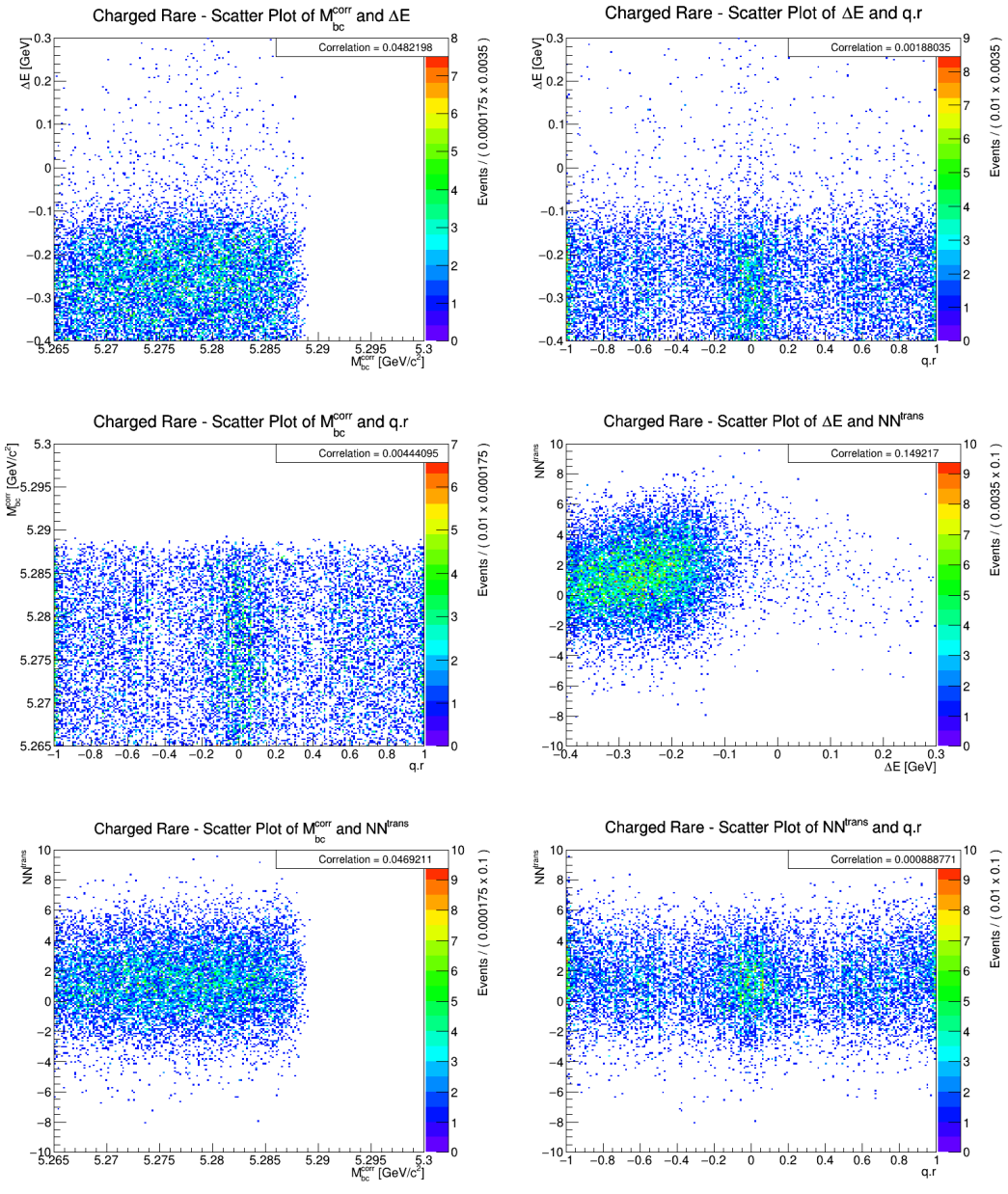

Figure B.7: The charged rare scatter plots in every pair of fitting dimensions.

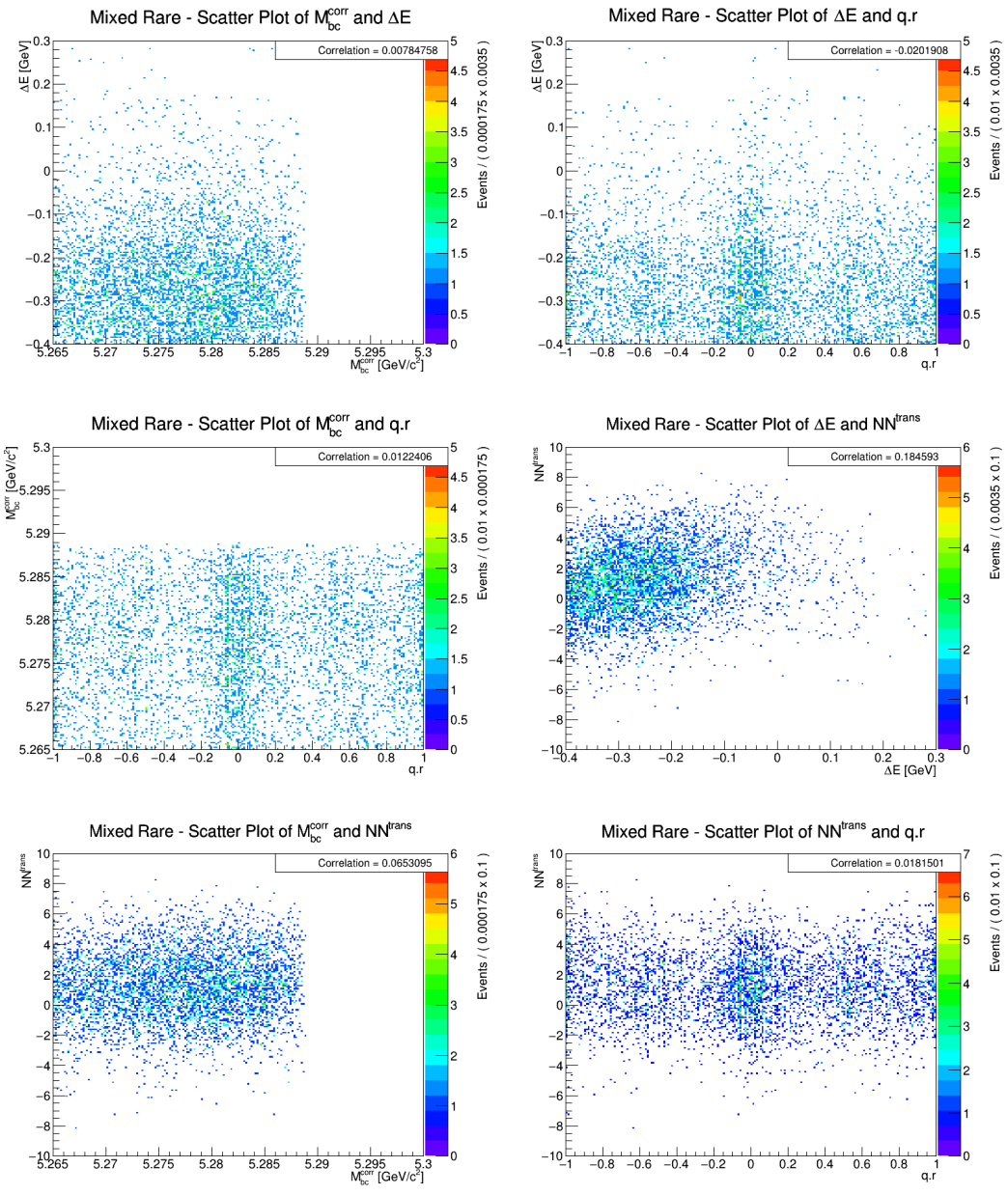

Figure B.8: The mixed rare scatter plots in every pair of fitting dimensions.

B.3 Data Processed by the TensorFlow Neural Network With the Adversarial Neural Network

M_{bc}^{corr}	NN^{trans}	$q.r$	
2.5%	11.8%	0.2%	ΔE
	1.8%	0.2%	M_{bc}^{corr}
		0.0%	NN^{trans}

Table B.5: Showing the (absolute) correlations between the four fitting variables, for the signal data processed by the TensorFlow neural-network trained with the adversary.

M_{bc}^{corr}	NN^{trans}	$q.r$	
0.4%	5.9%	0.0%	ΔE
	1.1%	0.4%	M_{bc}^{corr}
		0.8%	NN^{trans}

Table B.6: Showing the (absolute) correlations between the four fitting variables, for the continuum data processed by the TensorFlow neural-network trained with the adversary.

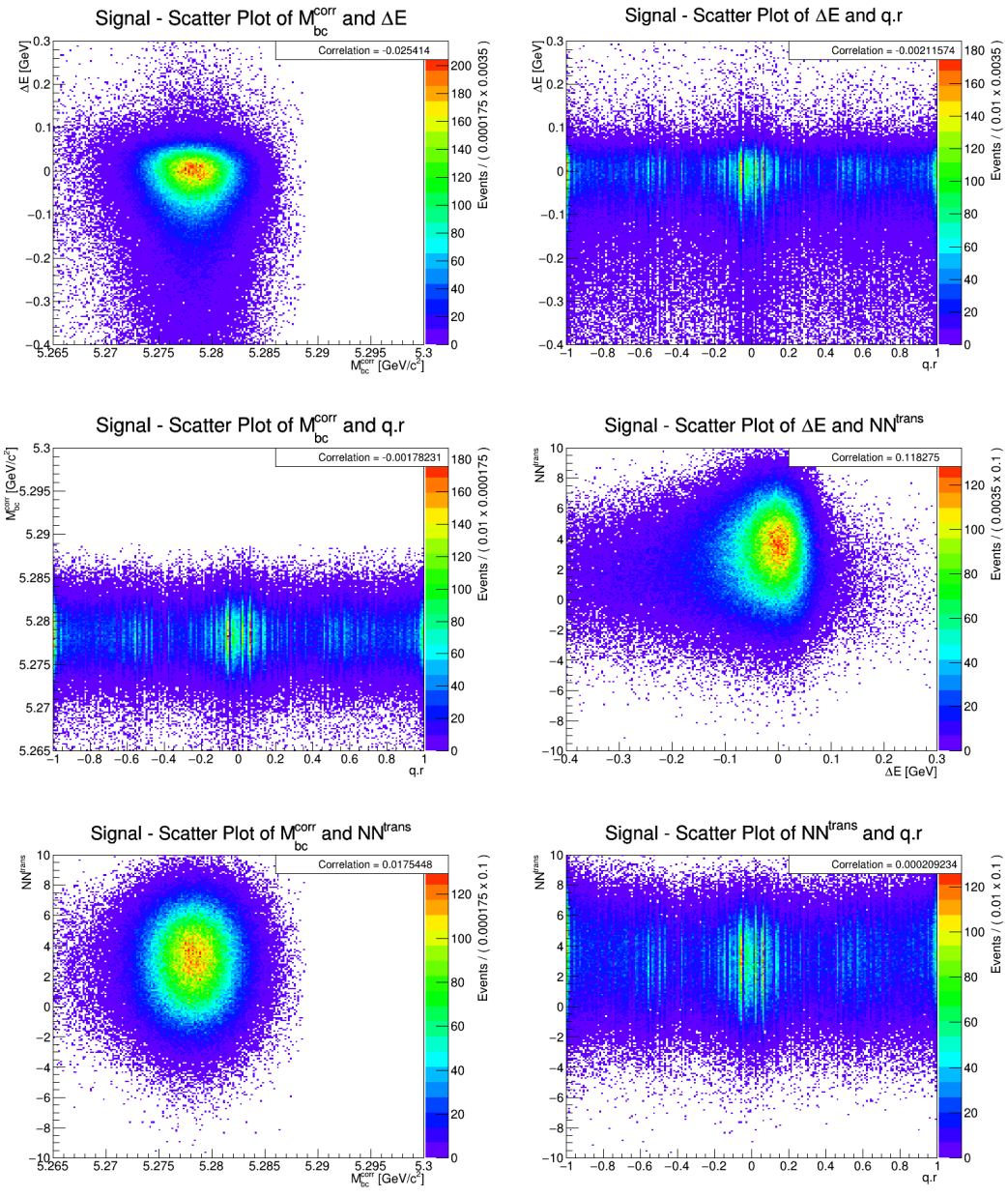

Figure B.9: The signal scatter plots in every pair of fitting dimensions.

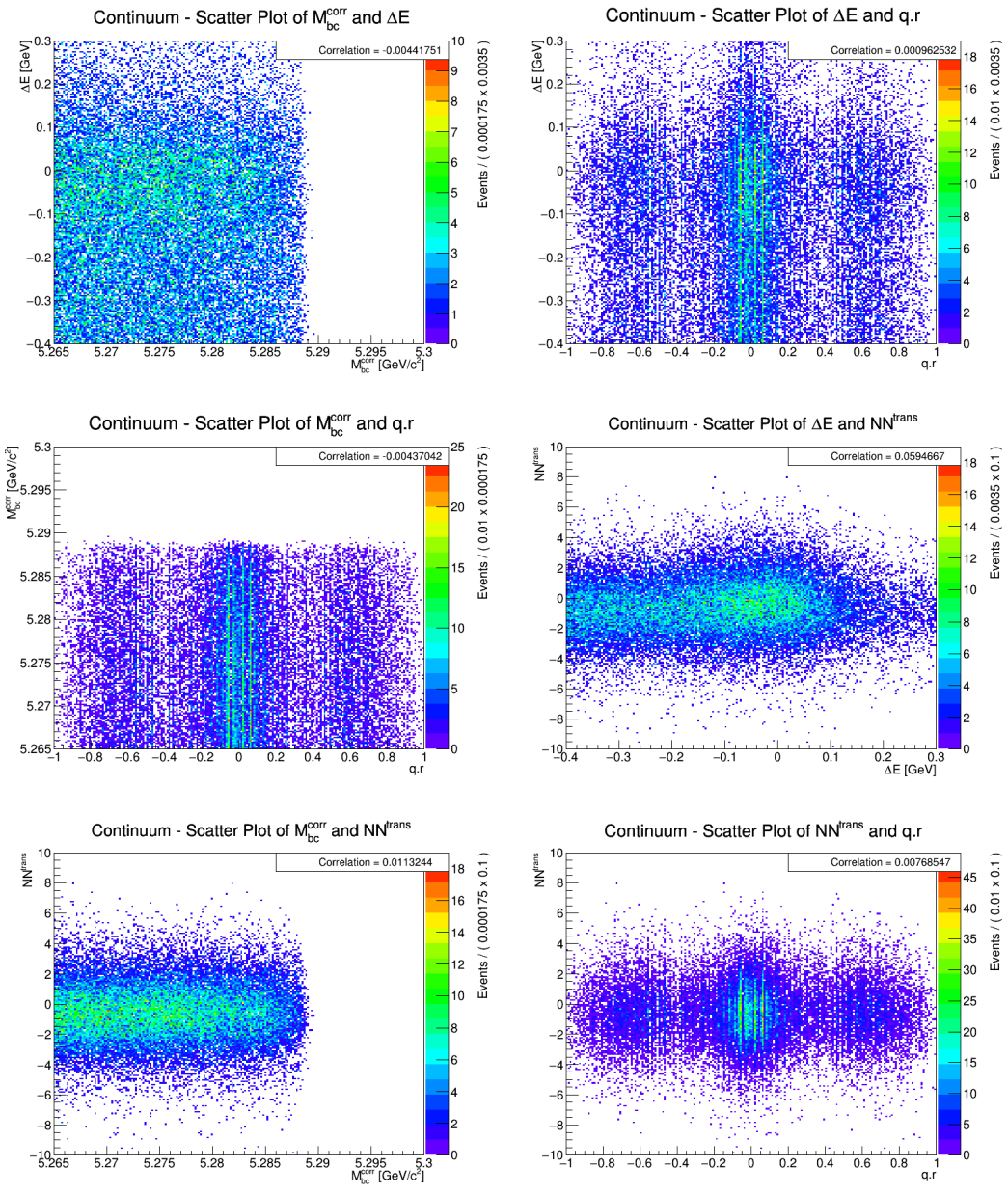

Figure B.10: The continuum scatter plots in every pair of fitting dimensions.

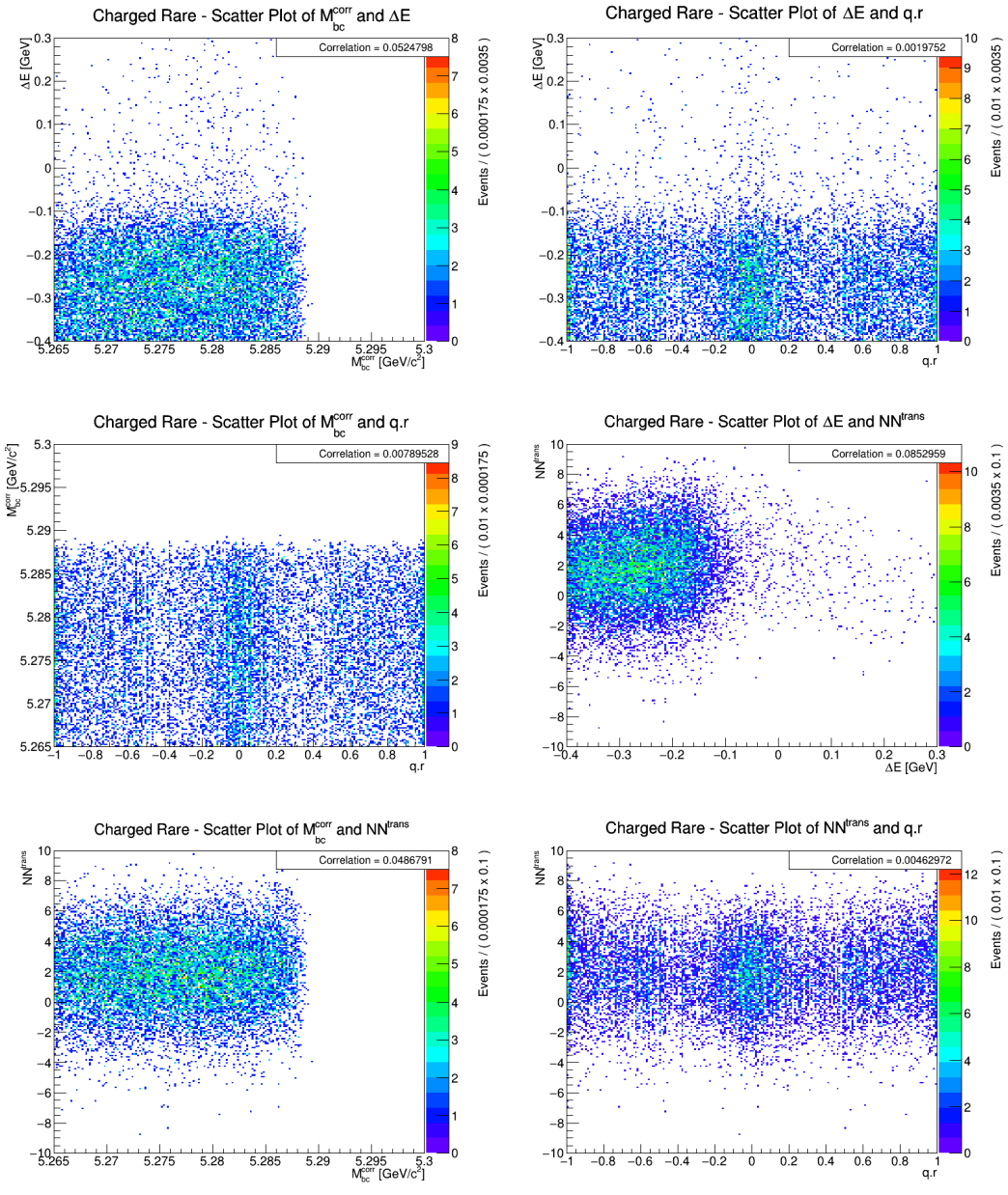

Figure B.11: The charged rare scatter plots in every pair of fitting dimensions.

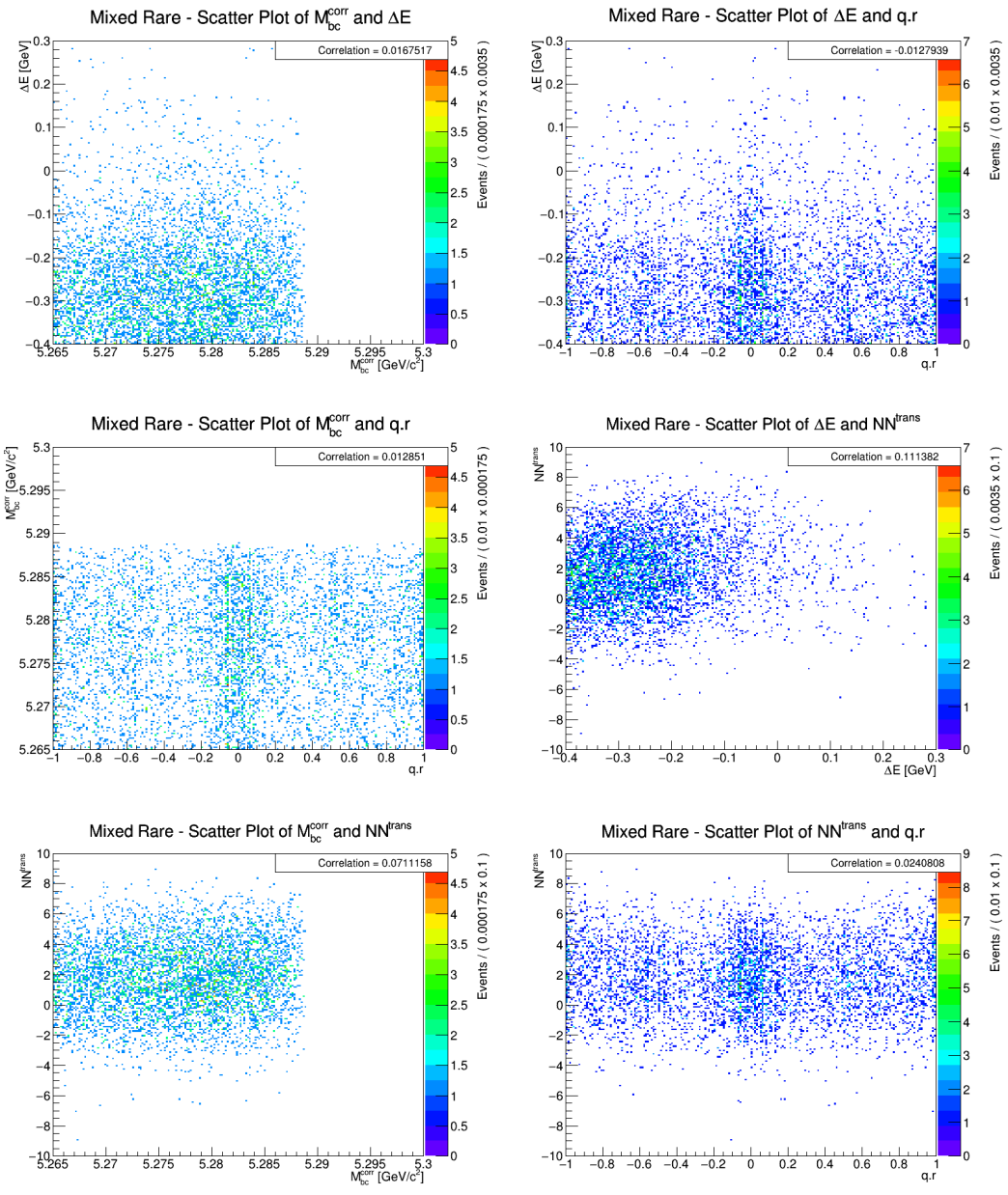

Figure B.12: The mixed rare scatter plots in every pair of fitting dimensions.

C|Scatter Plots and Correlations Between ΔE and the Kinematic Variables

Scatter plots of the ΔE with the kinematic variables used in training the neural networks. The ΔE range is set to be $-0.4 < \Delta E < 0.2$, the ΔE selection placed on all signal and continuum training dataset events before training the neural networks.

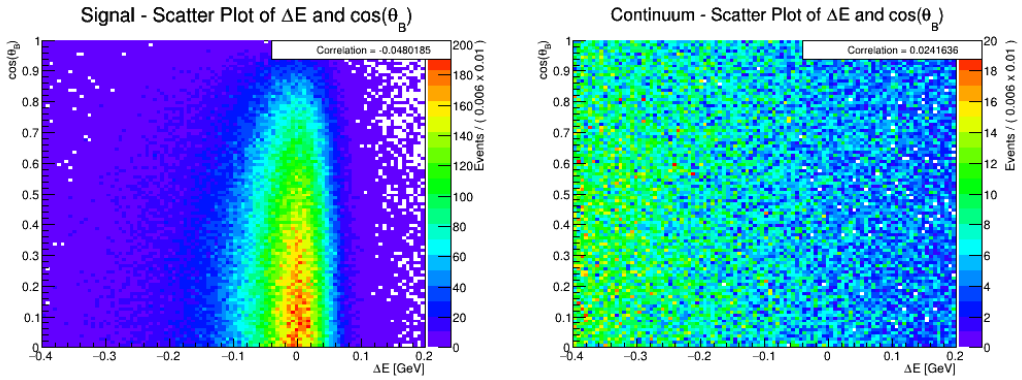

Figure C.1: Showing the signal (left) and continuum (right) scatter plots of ΔE and $cos(\theta_B)$

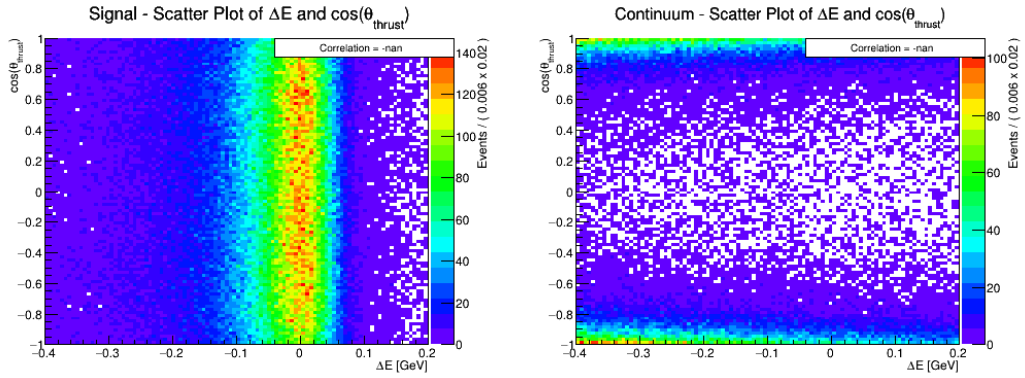

Figure C.2: Showing the signal (left) and continuum (right) scatter plots of ΔE and $cos(\theta_{thrust})$

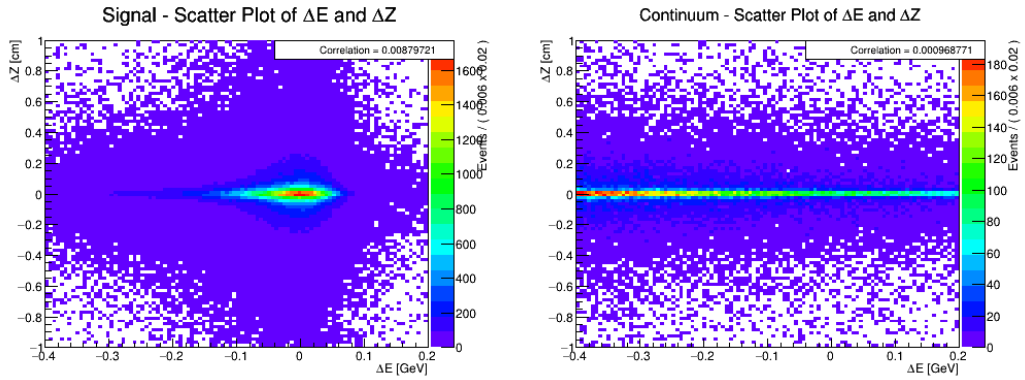

Figure C.3: Showing the signal (left) and continuum (right) scatter plots of ΔE and ΔZ

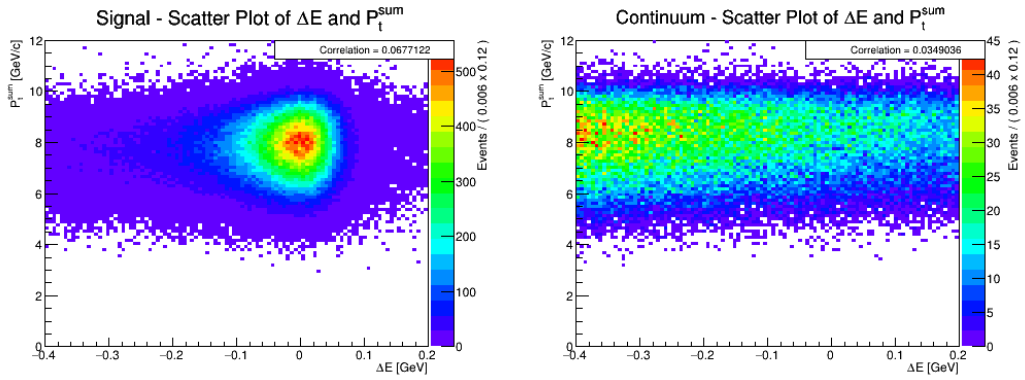

Figure C.4: Showing the signal (left) and continuum (right) scatter plots of ΔE and P_t^{sum}

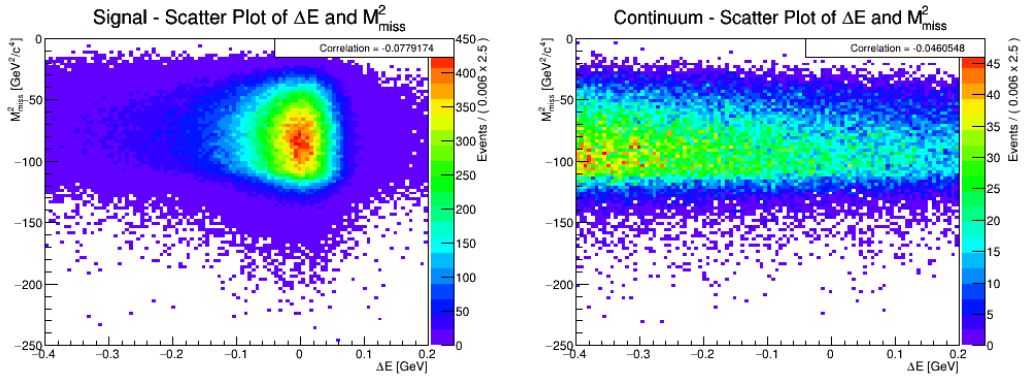

Figure C.5: Showing the signal (left) and continuum (right) scatter plots of ΔE and M^2_{miss}

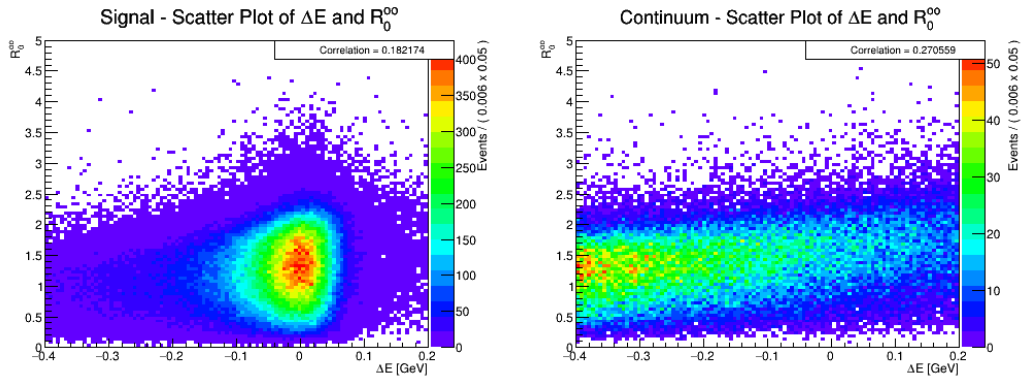

Figure C.6: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_0^{oo}

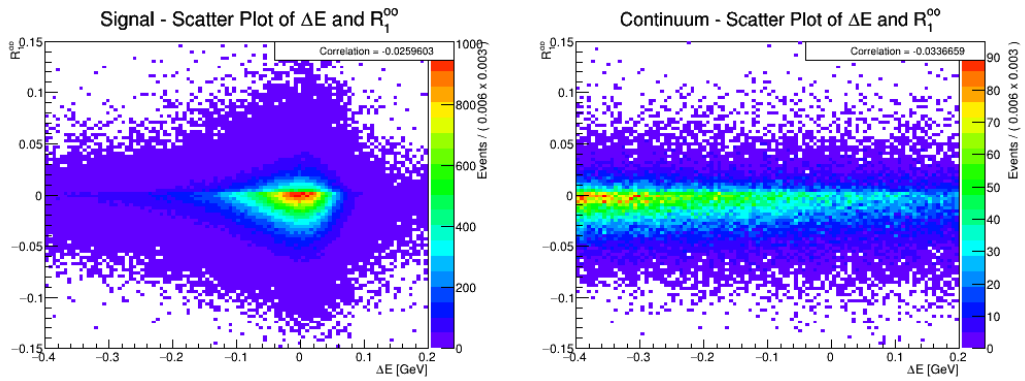

Figure C.7: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_1^{oo}

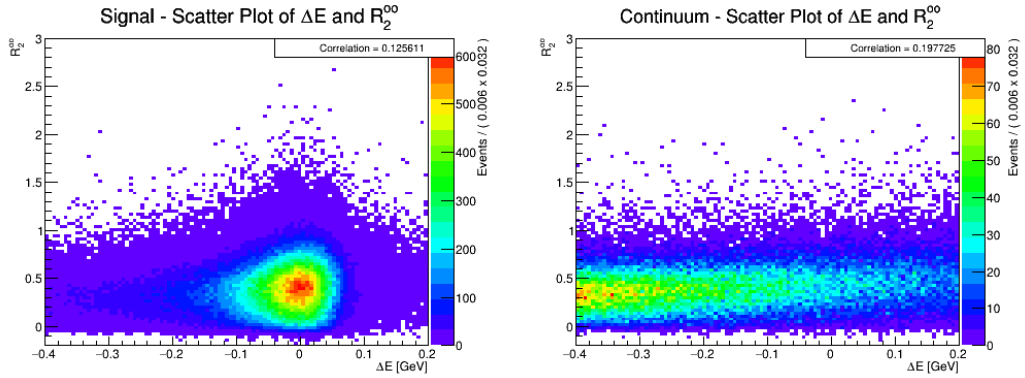

Figure C.8: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_2^{oo}

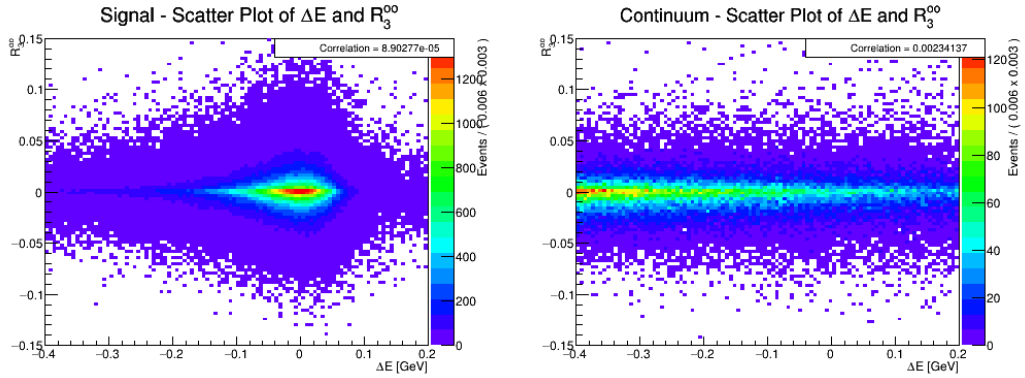

Figure C.9: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_3^{oo}

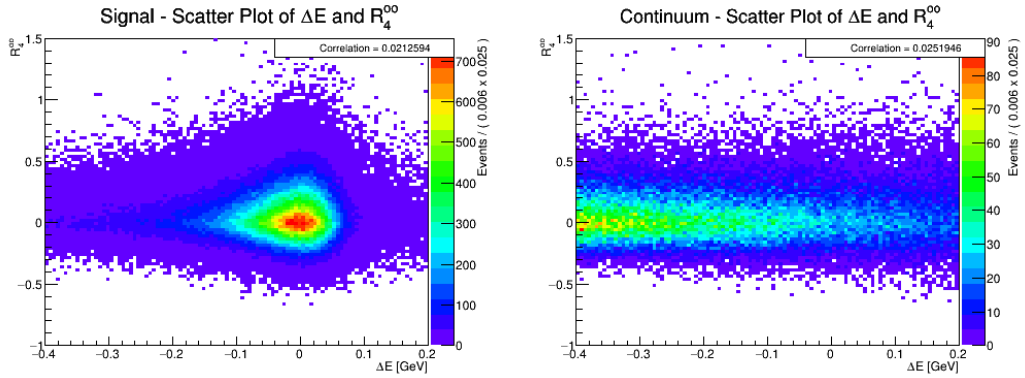

Figure C.10: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_4^{oo}

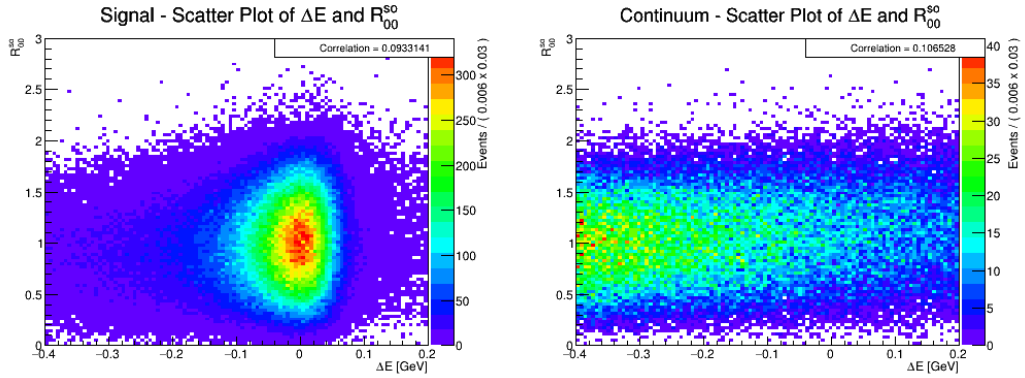

Figure C.11: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{00}^{so}

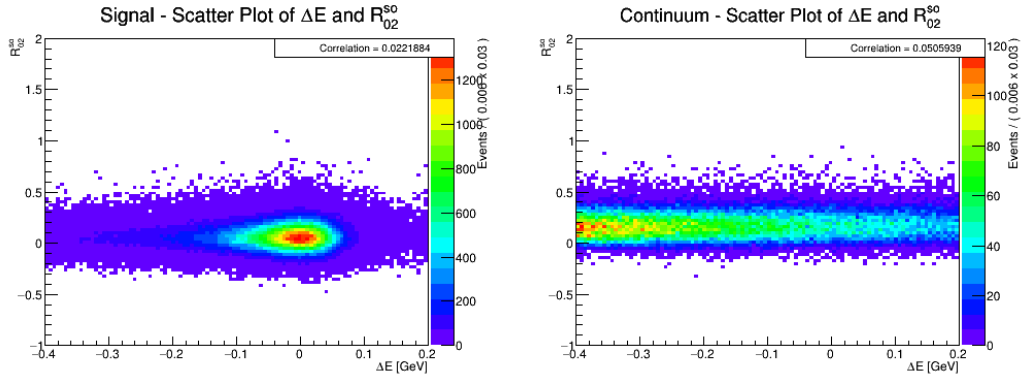

Figure C.12: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{02}^{so}

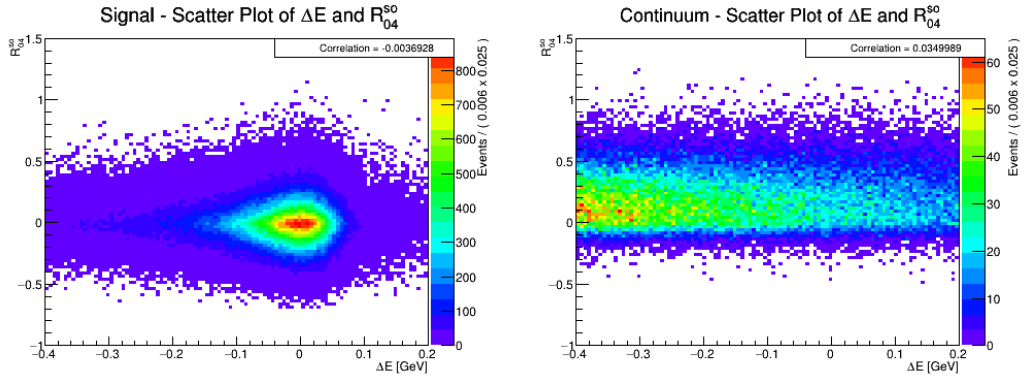

Figure C.13: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{04}^{so}

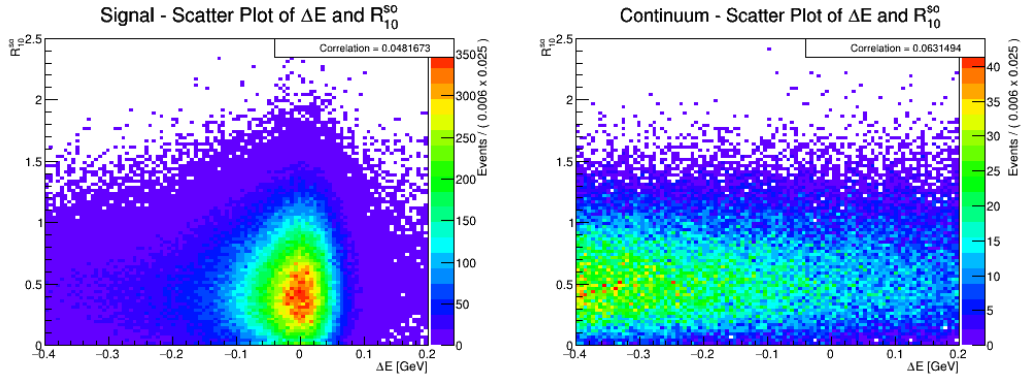

Figure C.14: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{10}^{so}

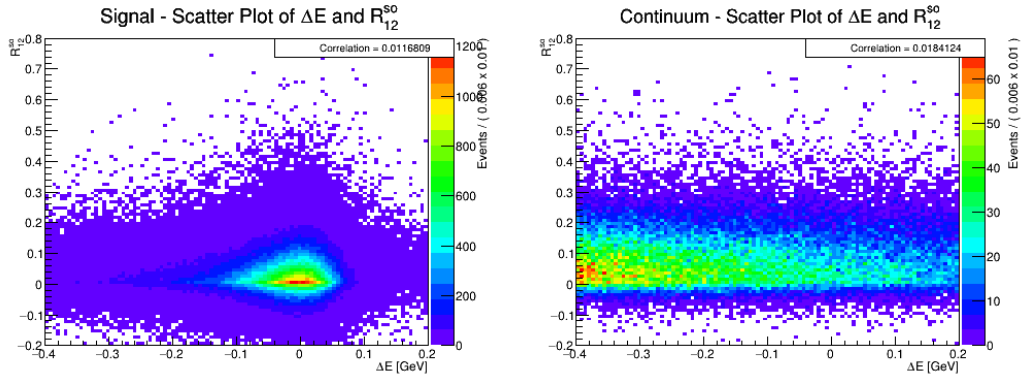

Figure C.15: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{12}^{so}

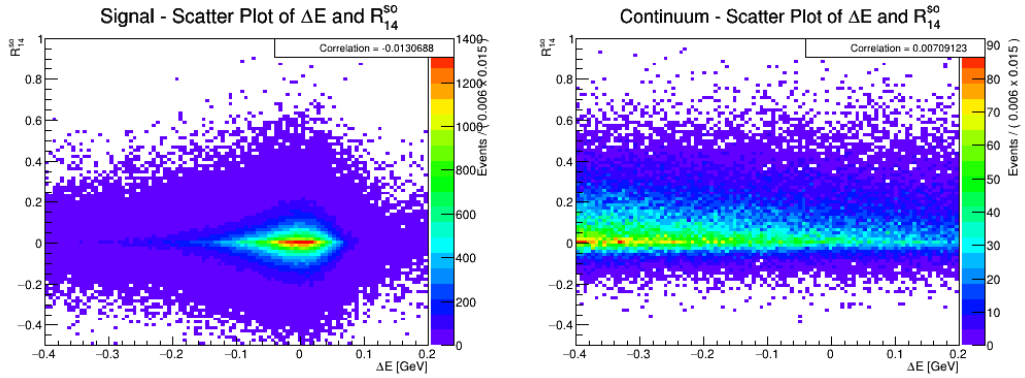

Figure C.16: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{14}^{so}

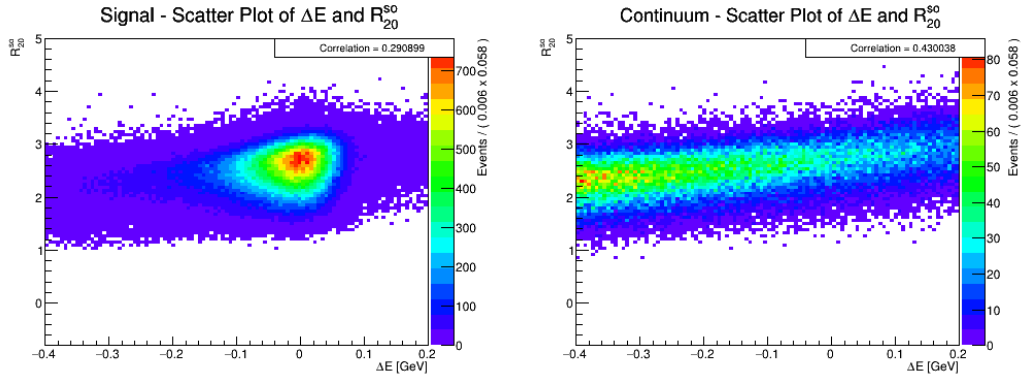

Figure C.17: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{20}^{so}

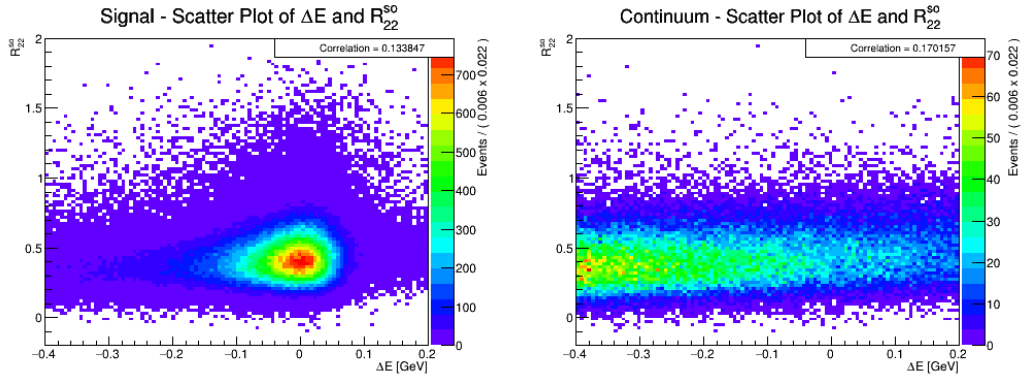

Figure C.18: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{22}^{so}

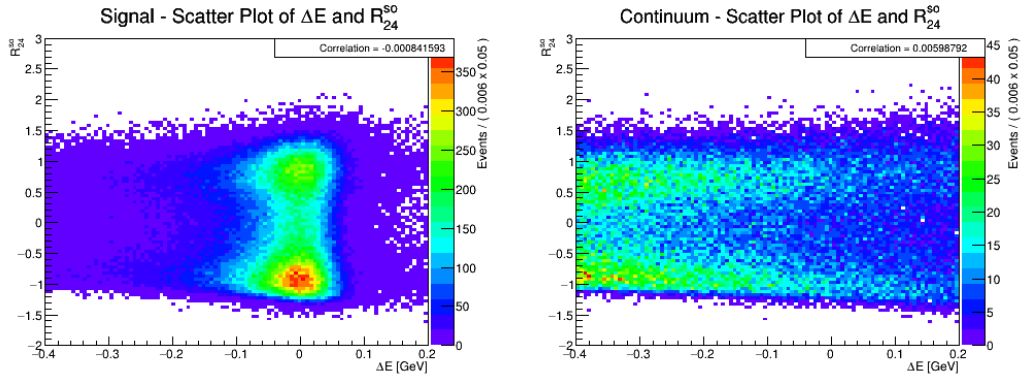

Figure C.19: Showing the signal (left) and continuum (right) scatter plots of ΔE and R_{24}^{so}

Bibliography

[1] James Kahn. ''Investigations of $B^0 \to K_S\pi^0$ decays with the Belle experiment''. MSc Thesis. The University of Melbourne, May 2015.

[2] Georges Aad et al. ''Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC''. In: *Phys. Lett.* B716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020.

[3] Y. Fukuda et al. ''Evidence for oscillation of atmospheric neutrinos''. In: *Phys. Rev. Lett.* 81 (1998), pp. 1562–1567. DOI: 10.1103/PhysRevLett.81.1562.

[4] Laurent Canetti, Marco Drewes, and Mikhail Shaposhnikov. ''Matter and Antimatter in the Universe''. In: *New J. Phys.* 14 (2012), p. 095012. DOI: 10.1088/1367-2630/14/9/095012.

[5] C. Patrignani et al. ''Review of Particle Physics''. In: *Chin. Phys.* C40.10 (2016), p. 100001. DOI: 10.1088/1674-1137/40/10/100001.

[6] Patrick Huet and Eric Sather. ''Electroweak baryogenesis and standard model CP violation''. In: *Phys. Rev.* D51 (1995), pp. 379–394. DOI: 10.1103/PhysRevD.51.379.

[7] A. D. Sakharov. ''Violation of CP Invariance, c Asymmetry, and Baryon Asymmetry of the Universe''. In: *Pisma Zh. Eksp. Teor. Fiz.* 5 (1967). [Usp. Fiz. Nauk161,61(1991)], pp. 32–35. DOI: 10.1070/PU1991v034n05ABEH002497.

[8] C. S. Wu et al. ''Experimental Test of Parity Conservation in Beta Decay''. In: *Phys. Rev.* 105 (1957), pp. 1413–1414. DOI: 10.1103/PhysRev.105.1413.

[9] J. H. Christenson et al. ''Evidence for the 2π Decay of the K_2^0 Meson''. In: *Phys. Rev. Lett.* 13 (1964), pp. 138–140. DOI: 10.1103/PhysRevLett.13.138.

[10] P. Kooijman and N. Tuning. ''Lectures on CP violation''. 2015. URL: https://www.nikhef.nl/~h71/Lectures/2015/ppII-cpviolation-29012015.pdf.

[11] Ling-Lie Chau and Wai-Yee Keung. ''Comments on the Parametrization of the Kobayashi-Maskawa Matrix''. In: *Phys. Rev. Lett.* 53 (1984), p. 1802. DOI: 10.1103/PhysRevLett.53.1802.

[12] Lincoln Wolfenstein. ''Parametrization of the Kobayashi-Maskawa Matrix''. In: *Phys. Rev. Lett.* 51 (1983), p. 1945. DOI: 10.1103/PhysRevLett.51.1945.

[13] A. J. Bevan et al. ''The Physics of the B Factories''. In: *Eur. Phys. J.* C74 (2014), p. 3026. DOI: 10.1140/epjc/s10052-014-3026-9.

[14] Colin Gay. ''B mixing''. In: *Ann. Rev. Nucl. Part. Sci.* 50 (2000), pp. 577–641. DOI: 10.1146/annurev.nucl.50.1.577.

[15] Michael Gronau. ‘‘A Precise sum rule among four $B \to K\pi$ CP asymmetries’’. In: *Phys. Lett.* B627 (2005), pp. 82–88. DOI: 10.1016/j.physletb.2005.09.014.

[16] Seungwon Baek and David London. ‘‘Is There Still a $B \to \pi K$ Puzzle?’’ In: *Phys. Lett.* B653 (2007), pp. 249–253. DOI: 10.1016/j.physletb.2007.08.001.

[17] M. Fujikawa et al. ‘‘Measurement of CP asymmetries in $B^0 \to K^0\pi^0$ decays’’. In: *Phys. Rev.* D81 (2010), p. 011101. DOI: 10.1103/PhysRevD.81.011101.

[18] Y. -T. Duh et al. ‘‘Measurements of branching fractions and direct CP asymmetries for $B \to K\pi$, $B \to \pi\pi$ and $B \to KK$ decays’’. In: *Phys. Rev.* D87 (2013), p. 031103. DOI: 10.1103/PhysRevD.87.031103.

[19] Bernard Aubert et al. ‘‘Measurement of time dependent CP asymmetry parameters in B^0 meson decays to omega K^0_S, $\eta' K^0$, and $\pi^0 K^0_S$’’. In: *Phys. Rev.* D79 (2009), p. 052003. DOI: 10.1103/PhysRevD.79.052003.

[20] J. P. Lees et al. ‘‘Measurement of CP Asymmetries and Branching Fractions in Charmless Two-Body B-Meson Decays to Pions and Kaons’’. In: *Phys. Rev.* D87 (2013), p. 052009. DOI: 10.1103/PhysRevD.87.052009.

[21] Z. Natkaniec et al. ‘‘Status of the Belle silicon vertex detector’’. In: *Nucl. Instrum. Meth.* A560 (2006), pp. 1–4. DOI: 10.1016/j.nima.2005.11.228.

[22] A. Abashian et al. ‘‘The Belle Detector’’. In: *Nucl. Instrum. Meth.* A479 (2002), pp. 117–232. DOI: 10.1016/S0168-9002(01)02013-7.

[23] R. Itoh. ‘‘BASF - BELLE AnalysiS Framework’’. In: *Proceedings, 9th International Conference on Computing in High-Energy Physics (CHEP 1997): Berlin, Germany, April 7-11, 1997*. 1997. URL: http://www.ifh.de/CHEP97/paper/244.ps.

[24] D. J. Lange. ‘‘The EvtGen particle decay simulation package’’. In: *Nucl. Instrum. Meth.* A462 (2001), pp. 152–155. DOI: 10.1016/S0168-9002(01)00089-4.

[25] Rene Brun et al. *GEANT: Detector description and simulation tool*. Tech. rep. CERN, 1993. URL: http://cds.cern.ch/record/1073159/files/cer-002728534.pdf.

[26] F Fang. ‘‘Study of $K_S \to \pi^+\pi^-$ Selection’’. In: *Belle Note 323* (2000).

[27] H. Kakuno et al. ‘‘Neutral B flavor tagging for the measurement of mixing induced CP violation at Belle’’. In: *Nucl. Instrum. Meth.* A533 (2004), pp. 516–531. DOI: 10.1016/j.nima.2004.06.159.

[28] Geoffrey C. Fox and Stephen Wolfram. ‘‘Observables for the Analysis of Event Shapes in e^+e^- Annihilation and Other Processes’’. In: *Phys. Rev. Lett.* 41 (1978), p. 1581. DOI: 10.1103/PhysRevLett.41.1581.

[29] S. H. Lee et al. ‘‘Evidence for $B^0 \to \pi^0\pi^0$’’. In: *Phys. Rev. Lett.* 91 (2003), p. 261801. DOI: 10.1103/PhysRevLett.91.261801.

[30] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. ‘‘Deep Sparse Rectifier Neural Networks’’. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*. Ed. by Geoffrey J. Gordon and David B. Dunson. Vol. 15. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011, pp. 315–323. URL: http://www.jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf.

[31] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. 2015. arXiv: 1511.07289.

[32] M. Feindt and U. Kerzel. ‘‘The NeuroBayes neural network package’’. In: *Nucl. Instrum. Meth.* A559 (2006), pp. 190–194. DOI: 10.1016/j.nima.2005.11.166.

[33] Richard H. Byrd et al. ‘‘A Limited Memory Algorithm for Bound Constrained Optimization’’. In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[34] Michael Feindt. *A Neural Bayesian Estimator for Conditional Probability Densities*. 2004. arXiv: physics/0402093 [physics.data-an].

[35] R. Brun and F. Rademakers. ‘‘ROOT: An object oriented data analysis framework’’. In: *Nucl. Instrum. Meth.* A389 (1997), pp. 81–86. DOI: 10.1016/S0168-9002(97)00048-X.

[36] Wouter Verkerke and David P. Kirkby. ‘‘The RooFit toolkit for data modeling’’. In: *eConf* C0303241 (2003). [186(2003)], MOLT007. arXiv: physics/0306116 [physics].

[37] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. ‘‘Torch7: A Matlab-like Environment for Machine Learning’’. In: *BigLearn, NIPS Workshop*. 2011. URL: http://publications.idiap.ch/downloads/papers/2011/Collobert_NIPSWORKSHOP_2011.pdf.

[38] The Theano Development Team et al. *Theano: A Python framework for fast computation of mathematical expressions*. 2016. arXiv: 1605.02688.

[39] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467.

[40] Xavier Glorot and Yoshua Bengio. ‘‘Understanding the difficulty of training deep feedforward neural networks’’. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*. Ed. by Yee W. Teh and D. M. Titterington. Vol. 9. 2010, pp. 249–256. URL: http://www.jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf.

[41] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980.

[42] Lisha Li et al. *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*. 2016. arXiv: 1603.06560.

[43] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661.

[44] Gilles Louppe, Michael Kagan, and Kyle Cranmer. *Learning to Pivot with Adversarial Networks*. 2016. arXiv: 1611.01046.