Bachelor's Thesis

Suppression of Continuum Background with Neural Networks for Belle II

Urbschat, Bela[1, 2]

[1]Max Planck Institute for Physics
[2]Technical University of Munich, Department of Physics

December 18, 2023

Commit used for generation of this PDF: eb71f35ebcb056fc65b5cd6b54d43d2ea1e3bd15

Examiner (Supervisor): Prof. Dr. Allen Caldwell
Second Examiner: Prof. Dr. Stefan Schönert
Presented On: December 19, 2023

Contents

1. Introduction

After the discoveries of new particles like the Higgs boson or top quark, in recent years there has been an ongoing drought without any discoveries of comparable importance. More importantly though there are also no real hints on where to search. There are no clear theoretical predictions of a further particle, as it was for example the case for the top quark, which could be predicted with great confidence after verification of the Cabibbo–Kobayashi–Maskawa mechanism in the standard model. The focus of most research has thus shifted towards precision tests of the standard model. Increasingly more precise measurements are pursued in the hope of finding deviations from standard model predictions that may hint on the existence of new physics. To enable the necessary precision, extremely high statistics are necessary. To accommodate this, accelerator experiments started to aim for very high luminosities in order to collect the necessary data in reasonable time. The prime example is the Belle II experiment located at the SuperKEKB collider facility in Japan, but also for example the LHC is pursuing a high luminosity upgrade.
At Belle II, a so called B factory, pairs of B mesons are produced in large numbers to study their subsequent decays. B physics offers many opportunities for precision tests of the standard model, where processes involving loop diagrams are especially interesting as they may be influenced by new particles entering the loops. A group of decays of interest here are the $\mathrm{B} \to \mathrm{K}\pi$ decays, where the tree-level amplitudes are suppressed, thus making them sensitive to loop contributions. Measurements of branching ratios and CP asymmetries of those decays are expected to satisfy certain relations predicted by the standard model, which may however be violated if new physics is involved. This allows for so called null tests of the standard model. Some of the necessary measurements are however very difficult as the decays are rare and backgrounds are high. In this situation thus the best possible background suppression is desirable. Motivated by this, here a novel approach for $q\bar{q}$ background suppression using low level variables and deep neural networks will be explored for the $\mathrm{B}^0 \to \mathrm{K}^0\pi^0$ decay.

2. Theory and Motivation

The following sections serve to give an overview of the theoretical aspects and resulting motivations important for this thesis. Some aspects which are not directly important for this theses but still worth mentioning in context will only be covered very briefly.

2.1. CP Violation

CP symmetry is the combination of charge (C) and parity (P) symmetries which was originally conceived as an extension to P symmetry which was found to be violated in weak processes. Contrary to the initial motivation, CP symmetry itself was found to also be violated. Again, violations of CP symmetry have so far only ever been observed in weak processes. All available experimental evidence suggests that the strong and electromagnetic interactions conserve CP. In practice CP violation usually manifests in the differences between some processes and those involving the corresponding antiparticles.

2.1.1. Types of CP Violation

There are different manifestations of CP violation which will be briefly introduced. For the following explanations we consider a particle X which decays to some final state f. The CP conjugated particles are denoted $\overline{X}$ and $\overline{f}$. An observed asymmetry is then defined as

$$\mathcal{A} = \frac{\Gamma(X \to f) - \Gamma(\overline{X} \to \overline{f})}{\Gamma(X \to f) + \Gamma(\overline{X} \to \overline{f})}. \tag{2.1}$$

Direct CP Violation Direct CP violation, or CP violation in decay, means a difference in the decay amplitudes of a given decay and its CP conjugate which manifests in different decay rates. This may be expressed as

$$\Gamma(X \to f) \neq \Gamma(\overline{X} \to \overline{f}), \text{ or } \left|\frac{A_f}{A_{\overline{f}}}\right| \neq 1. \tag{2.2}$$

For direct CP violation there must be multiple (at least two) amplitudes contributing to the total amplitude of the decay. We may write the total amplitudes for the decay and its CP conjugated version in terms of the individual contributing decay amplitudes magnitudes A_k, CP even, so called strong phases δ_k and CP odd, so called weak phases ϕ_k.

$$A_f = \sum_k A_k e^{i(\delta_k + \phi_k)}, \quad A_{\overline{f}} = \sum_l A_l e^{i(\delta_l - \phi_l)}. \tag{2.3}$$

CP violation then can be seen to only occur if there are at least two contributing amplitudes which have different strong and weak phases as

$$|A_f|^2 - |A_{\overline{f}}|^2 = 2 \sum_{k,l} A_k A_l \sin(\phi_k - \phi_l) \sin(\delta_k - \delta_l). \tag{2.4}$$

***CP* Violation in Mixing** CP violation in neutral meson mixing occurs when the mass eigenstates are not pure CP eigenstates. This introduces an asymmetry in the flavor changing oscillations where now the transition rate from particle to antiparticle and from antiparticle to particle differ. For example for the neutral B meson system this means

$$\left|\left\langle \mathrm{B}^0 \middle| \overline{\mathrm{B}}^0(t) \right\rangle\right| \neq \left|\left\langle \overline{\mathrm{B}}^0 \middle| \mathrm{B}^0(t) \right\rangle\right|. \tag{2.5}$$

This effect for B mesons is however very small and can not be measured directly.

***CP* Violation in the Interference of Mixing and Decay** If the final state f is chosen as an CP eigenstate, meaning $f = \overline{f}$, the decay chains $X \to \overline{X} \to f$ and $X \to f$ now may interfere. This results in an observable asymmetry which is how CP violation in the B^0 meson system was initially experimentally verified. The experimental determination here requires a time-dependent measurement of the asymmetry.

2.1.2. *CP* Violation in the SM - The CKM Matrix

CP violation in weak processes is described in the standard model. Weak phases enter the decay amplitudes through the components of the Cabibbo-Kobayashi-Maskawa (CKM) quark mixing matrix [1] which is parametrized by three rotation angles and one complex phase. Complex phases can be shown to only enter when there are at least three generations of quarks. This in turn implied that the observation of CP violation as described by the CKM formalism hinted on the existence of a complete third quark generation, which was subsequently discovered.
There are many parametrizations of the CKM matrix. One particularly useful one is the Wolfenstein parametrization, which highlights the hierarchy of the CKM matrix elements.

$$V_{\mathrm{CKM}} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} = \begin{pmatrix} 1 - \frac{\lambda^2}{2} & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{\lambda^2}{2} & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4) \tag{2.6}$$

λ here is approximately equal to 0.2. Couplings within the same generation are thus the largest and decrease the more generations are traversed.

2.2. Search for New Physics in B → Kπ Decays

The $\mathrm{B} \to \mathrm{K}\pi$ decays offer a good probe for physics beyond the standard model due to comparatively large contributions of loop diagrams to the total decay amplitudes. This is because tree level contributions are suppressed by the involved small CKM couplings. Measurements of these decays thus are sensitive to the loop contributions which in turn are expected to be sensitive to new physics, as unknown particles may enter the loops. Two examples for the involved Feynman diagrams are shown in fig. 2.1.

2.2.1. Isospin Sum Rule as a Null-Test of the Standard Model

In order to avoid large hadronic uncertainties which complicate direct measurements of loop contributions one combines measurements from flavor symmetry related decay modes. A common approach is to combine measurements of decays which are related by isospin symmetry [21]. Assuming isospin symmetry a sum rule leading to the relation

$$2\mathcal{A}_{CP}(\pi^0\mathrm{K}^+)\frac{\mathcal{B}(\pi^0\mathrm{K}^+)}{\mathcal{B}(\pi^-\mathrm{K}^+)}\frac{\tau_{\mathrm{B}^0}}{\tau_{\mathrm{B}^+}} - \mathcal{A}_{CP}(\pi^+\mathrm{K}^0)\frac{\mathcal{B}(\pi^+\mathrm{K}^0)}{\mathcal{B}(\pi^-\mathrm{K}^+)}\frac{\tau_{\mathrm{B}^0}}{\tau_{\mathrm{B}^+}} - \mathcal{A}_{CP}(\pi^-\mathrm{K}^+) + 2\mathcal{A}_{CP}(\pi^0\mathrm{K}^0)\frac{\mathcal{B}(\pi^0\mathrm{K}^0)}{\mathcal{B}(\pi^-\mathrm{K}^+)} = 0 \tag{2.7}$$

can be derived. $\mathcal{A}_{CP}$ are the direct CP asymmetries and $\mathcal{B}$ the branching fractions (averaged over b and $\overline{\text{b}}$). While initially believed to be violated by electroweak penguin diagram contributions, which do not conserve isospin, Gronau [3] argued that within the standard model such contributions are extremely small and the sum rule should hold with uncertainty much smaller than 1% [4]. Thus this sum rule gives a precise null-test of the standard model with any new physics in the contributing loops possibly leading to a violation of the sum rule.

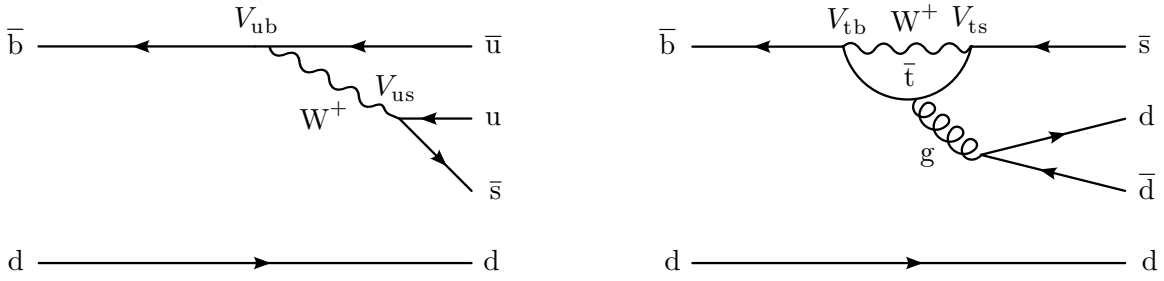


Figure 2.1.: Example for Feynman Diagrams describing the processes contributing to the decay amplitudes for the $\text{B} \to \text{K}\pi$ decays. On the left a color suppressed tree-level diagram and on the right a QCD penguin diagram is shown. In the loop is drawn with a $\overline{\text{t}}$ but also $\overline{\text{c}}$ and $\overline{\text{u}}$ can contribute.

While the current experimental results are compatible with the sum rule, some of the contributing measurements still come with large uncertainties. Thus more precise measurements are required, which is a challenging task as some of the contributing decays are especially hard to measure. Most notably $\text{B} \to \text{K}^0\pi^0$ introduces the additional difficulty of the reconstructed final state ($\text{K}^0_\text{S}\pi^0$) being a CP eigenstate and thus containing no information on the flavor of the decaying B. To still determine the corresponding CP asymmetries, this information must therefore be obtained through flavor tagging. Flavour tagging allows for deduction of the flavor of the reconstructed B by considering the decay of the other B from a produced $\text{B}\overline{\text{B}}$ pair. This however comes at a statistical cost [4]. For the decays appearing in the sum rule the current Belle II measurements of $\mathcal{B}$ and $\mathcal{A}_{CP}$ are shown in table 2.1. The uncertainties for $\text{B} \to \text{K}^0\pi^0$ can be seen to be statistically dominated, especially for the CP asymmetry.
As an attempt to reduce statistical uncertainties on those measurements, one can try to improve background suppression for the concerned decays, which forms the main motivation for this thesis. While here we will consider the decay $\text{B}^0 \to \text{K}^0_\text{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$ specifically as an example, this is not the only $\text{B} \to \text{K}\pi$ decay for which an analysis could benefit from better continuum suppression.

decay	$\mathcal{B}$ $[10^{-6}]$	$\mathcal{A}_{CP}$
$\text{B}^0 \to \text{K}^+\pi^-$	$20.67 \pm 0.37 \pm 0.62$	$-0.072 \pm 0.019 \pm 0.007$
$\text{B}^+ \to \text{K}^0\pi^+$	$24.37 \pm 0.71 \pm 0.86$	$0.046 \pm 0.029 \pm 0.007$
$\text{B}^+ \to \text{K}^+\pi^0$	$13.93 \pm 0.38 \pm 0.71$	$0.013 \pm 0.027 \pm 0.005$
$\text{B}^0 \to \text{K}^0\pi^0$	$10.40 \pm 0.66 \pm 0.60$	$-0.06 \pm 0.15 \pm 0.04$

Table 2.1.: Current Belle II measurements for branching fractions and CP asymmetries appearing in the sum rule presented in eq. (2.7). The values are taken from [24]. The first contribution to uncertainty is the statistical component and the second the systematic.

3. The Belle II Experiment

3.1. The SuperKEKB Collider

SuperKEKB[1] is an e^+e^- collider located in Tsukuba, Japan, operating mainly at the $\Upsilon(4S)$ resonance, which corresponds to a center of mass energy of $\sqrt{s} = 10.58\,\text{GeV}$. Operation at slightly different energies is however also possible, as done for so called *off-resonance* runs where no $\Upsilon(4S)$ is produced. The hadronic cross section of the e^+e^- collision as a function of the center of mass energy is illustrated in fig. 3.1 which also shows the different $b\overline{b}$ resonances and the energy region for off-resonance operation. $\Upsilon(4S)$ is chosen for normal operation of the collider as it almost exclusively (in more than 96% of cases) decays to a pair of the desired B mesons [23]. The decays of the produced $B\overline{B}$ pairs are then studied with the Belle II detector (introduced in the next section).

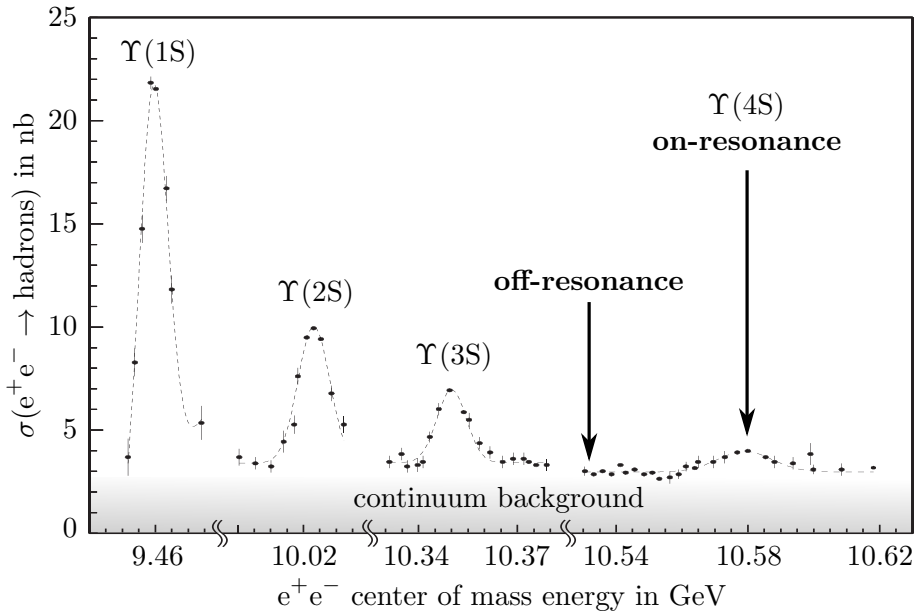


Figure 3.1.: Hadronic cross section of the e^+e^- collision as a function of the center of mass energy (adapted from [25]).

A simplified representation of the collider is shown in fig. 3.2. The collider consists of two separate rings: the high energy ring (HER) and low energy ring (LER) operating at around 7 GeV and 4 GeV respectively [18]. The center of mass frame of the collision therefore has a boost of $\beta\gamma = 0.28$ in the lab frame.

SuperKEKB together with the Belle II detector are operating at the *precision forefront*. This is complementary to the *energy forefront*, which is the regime of operations of experiments at the LHC. To enable the most precise measurements possible, large statistics are vital, which is why SuperKEKB is aiming for a very high luminosity of $6 \times 10^{35}\,\text{cm}^{-2}\,\text{s}^{-1}$, which however is yet to achieve. Nevertheless, SuperKEKB holds the world record in peak luminosity at $4.71 \times 10^{34}\,\text{cm}^{-2}\,\text{s}^{-1}$ [26]. The high peak luminosities are reached by highly focussing the beams at the interaction point. This is enabled by

[1] SuperKEKB is an upgrade of the KEKB collider which was operated in conjunction with the Belle detector.

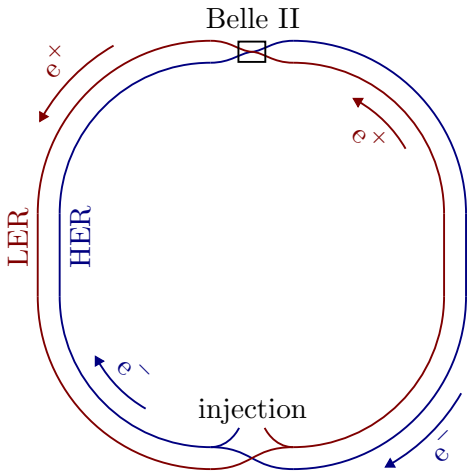


Figure 3.2.: Schematic depiction of the SuperKEKB colliders storage rings. Shown are the low energy ring (LER), high energy ring (HER) as well as location of the Belle II detector and e^+e^- injection.

the so called *nano beam scheme* [6], which was introduced as part of the upgrade from the predecessor KEKB to SuperKEKB.

3.2. The Belle II Detector

The statements in this and the following sections will be based of the Belle II Physics Book [18] if not specified otherwise.
The Belle II detector, a comprehensive upgrade of the Belle detector, is a general purpose spectrometer build around the interaction point of the SuperKEKB collider. It is composed of different sub-detectors which are build around the interaction point in a shell like structure. The detectors construction further follows the asymmetry of the collisions along the beam axis to maximize geometrical acceptance. While dependent on the different sub-detectors, in the lab frame the detector as a whole covers a polar angle (measured form the detectors symmetry axis) of around 17° to 150°, corresponding to a symmetric acceptance of around 23° to 157° in the center of mass frame.
An overview of the detector, indicating the locations of the sub-detectors, is shown in fig. 3.3. As indicated in the figure, the usual coordinate system is chosen such that the z-axis is the symmetry axis of the detector, pointing in the direction of the boost of the collisions. The x-axis is chosen to point horizontally away from the center of the colliders storage rings, which in a left handed coordinate system fixes the y-axis to point upwards. The endcap part of the detector in the direction of the z-axis is usually referred to as the *forward* region, the opposite side of the detector then as the *backward* region. Everything in between is called the *barrel* region.

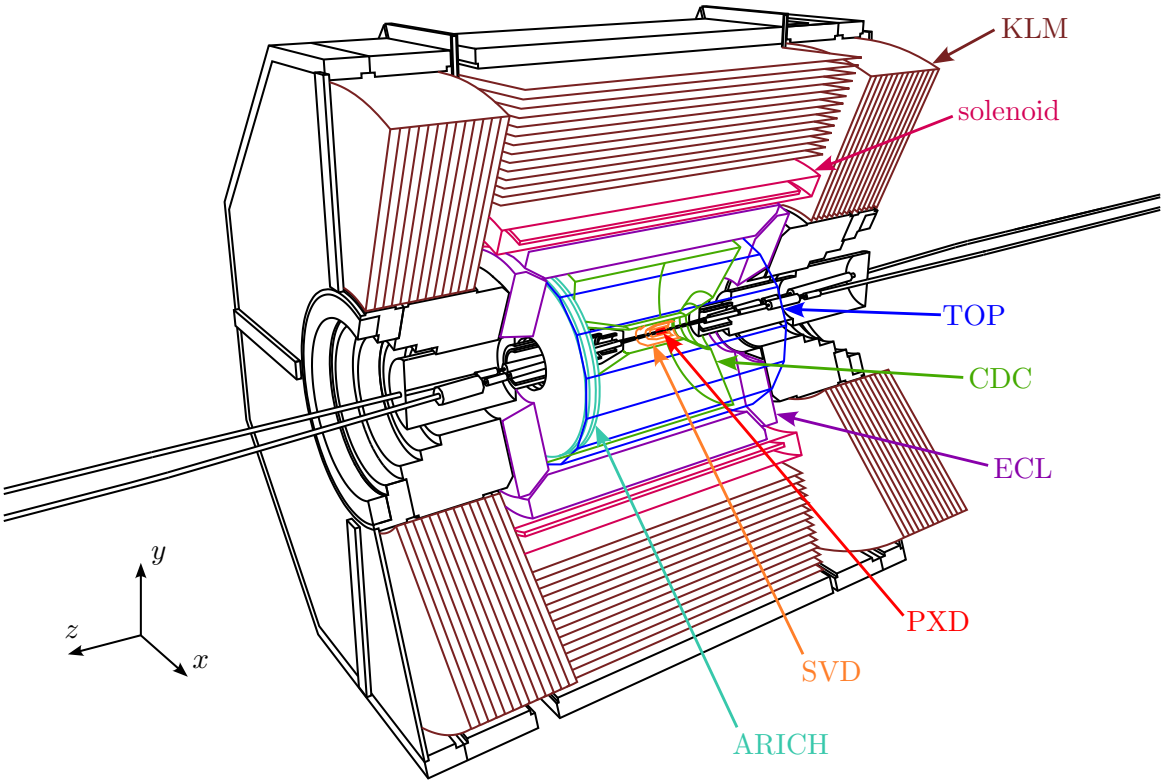


Figure 3.3.: Wireframe drawing of the Belle II detector. The different sub-detectors are shown in different colors and are indicated by the corresponding labels. The indicated coordinate system has its origin usually located at the interaction point. Here it is drawn translated next to the detector for better visibility.

3.2.1. Vertex Detector (VXD)

The innermost layer of the Belle II detector is the vertex detector (VXD). It consists of 6 layers of silicon detectors of different technologies. The innermost two layers, directly surrounding the beam

pipe, are pixelated sensors of the DEPFET type positioned at $r = 14\,\text{mm}$ and $r = 22\,\text{mm}$ from the z-axis. This part of the VXD is referred to as the *silicon pixel detector* (PXD). The remaining layers at radii of 39 mm to 135 mm are double-sided silicon strip sensors, which are together referred to as the *silicon vertex detector* (SVD). The pixelated sensors for the innermost layers are necessary to uniquely determine hit positions for a large number of tracks simultaneously, as required for measurements at the high luminosities the collider is operated at. Together the PXD and SVD allow for the precise reconstruction of tracks of charged particles near the interaction point, which in turn enables the precise determination of decay vertices.

3.2.2. Central Drift Chamber (CDC)

The central drift chamber (CDC) is the main tracking device in the Belle II detector. It is a large-volume wire chamber extending from $r = 16\,\text{cm}$ to $r = 113\,\text{cm}$, surrounding the VXD. Its volume is filled with a 50 : 50 mixture of helium and ethane and traversed by 14 336 sense wires and 42 240 field wires. The total of 56 layers of sense wires are arranged in 9 superlayers which alternate between *axial* and *stereo* orientation of the wires [15]. The wires in the axial superlayers are parallel to the z-axis while for the stereo layers they are slightly skewed relative to the z-axis. Combination of the hits from axial and stereo layers then allows for reconstruction of three dimensional tracks. As the tracks are curved due to the magnetic field of the surrounding solenoid (described in detail later), the reconstructed tracks allow for determination of particle momenta from the track curvatures. Further the CDC allows for determination of the energy loss $\frac{\mathrm{d}E}{\mathrm{d}x}$ of the traversing particles, which depends on the velocity and thus, together with information on the momenta, can be used for particle identification. Finally the CDC is used as the (only) input to the track trigger due to its fast readout.

3.2.3. Particle Identification (TOP, ARICH)

There are two sub-detectors dedicated to particle identification. While both of them fundamentally rely on Cherenkov radiation to obtain information on the velocity of the traversing particles, their construction and working principle is remarkably different. The time-of-propagation (TOP) counter is a novel Cherenkov detector utilizing total internal reflection of Cherenkov light in a rectangular quartz bar of length 2.6 m and width 45 cm. The quartz bar at the same time acts as the Cherenkov radiator and light guide to guide the Cherenkov photons to one end of the bar where they will be detected. The other end of the bar is formed into a spherical mirror. 16 such quartz bars are placed around the perimeter of the CDC. As the Cherenkov light is only detected at one end of the quartz bar, the TOP can be build to be very compact, allowing for a larger volume of the CDC.
In the TOP, the two dimensional information for a Cherenkov ring image is obtained from the signals from an array of 16-channel micro-channel plate photomultiplier tubes (MCP-PMTs) located at one end of each of the quartz bars. One spacial dimension as well as the arrival time of the photons already gives enough information to reconstruct the two dimensional Cherenkov ring image. This however requires a very high single-photon time resolution of at least around 100 ps, which is on the order of the propagation time difference for Cherenkov light from kaons and pions (the main particle types which the TOP is designed to discriminate) at 2 GeV. The required single photon time resolution is achieved with purpose build MCP-PMTs and readout electronics, reaching a resolution of around 40 ps [5]. Further this method also requires precise knowledge of the particle production time, which is provided by the other sub-detectors.
While the TOP covers the barrel region of the detector, covering polar angles from 31° to 128°, the forward endcap region is covered by an aerogel ring-imaging Cherenkov (ARICH) detector, covering the angles from 14° to 30°. There is no dedicated particle identification for the backwards region. The ARICH detector is a more traditional ring imaging Cherenkov detector. It uses 2 cm thick aerogel as the radiator to then sample the Cherenkov photons directly in a two dimensional image plane located 20 cm behind the radiator.

3.2.4. Electromagnetic Calorimeter (ECL)

The electromagnetic calorimeter (ECL) is dedicated to the detection of photons, which are not visible in the other sub-detectors. Besides this it is also used to identify electrons with the intend of separating them from hadrons like the charged pions. The calorimeter consists of a large array of a total of 8736 thallium-doped caesium iodide CsI(Tl) crystals and is subdivided into barrel, forward and backwards regions. In total, a polar angle from 12° to 155° with small gaps on the order of 1° between the three regions is covered. While most of the ECLs structure, including the crystals, was inherited from the Belle detector, the readout electronics were upgraded to accommodate wave-form-sampling. This was necessary as due to the higher luminosities at Belle II the background levels will increase to an extend where the comparatively long decay time of the scintillators can cause pulses from neighboring (background) events to overlap, which could not be resolved with the old readout electronics.

3.2.5. Superconducting Solenoid

The ECL is surrounded by a superconducting solenoid magnet with an inner radius of 1.7 m. All sub-detectors located within the solenoid are submerged in its homogeneous magnetic field of 1.5 T. Only the K_L^0 and μ detector (described in the next section) is located outside the solenoid. The purpose of the magnetic field is to curve the tracks of charged particles in the detector as required for determination of their traverse momenta from track curvatures.

3.2.6. K_L^0 and μ Detector (KLM)

The K_L^0 and μ detector (KLM) is the outermost component of the Belle II detector. It consists of a structure of alternating layers of 4.7 cm thick iron plates and active detector elements. While muons do not produce any showers in the ECL, they can be detected as tracks by the detector elements of the KLM. To identify muons, tracks from the CDC are matched with hits from the KLM under the assumption that the muon must traverse the KLM completely, which distinguishes muons from hadrons. Hadrons, in particular the K_L^0, are then identified by a cluster of hits in the KLM that cannot be matched to any charged tracks in the detector. The K_L^0 further induce showers in the ECL when traversing it. If possible information from both the KLM and ECL is combined.

3.3. Continuum Background

While the $\Upsilon(4S)$ resonance is the desired state to be produced in the collisions, it is by far not the only possible one. For example $e^-e^+ \to e^-e^+$ (Bhabha scattering) has a cross section about 300 times the one for production of a $\Upsilon(4S)$ resonance [10].

The dominating non-hadronic backgrounds, like Bhabha scattering, can be easily rejected. However, hadronic processes of the type $e^+e^- \to q\bar{q}$, where q are the quarks lighter than the b quark, result in complicated hadronic showers in the detector and are much more difficult to reject. These processes are what is referred to as *continuum background* or $q\bar{q}$ *background* in the context of Belle II. As there is a sufficient mass difference between the b quark and the lighter quarks, continuum background appears uniformly throughout the operating range of the SuperKEKB collider. This is also indicated in fig. 3.1.

Continuum background then forms the main background for many analyses, including those of the decay $B^0 \to K_S^0(\pi^+\pi^-)\pi^0(\gamma\gamma)$. For this low multiplicity decay a signal peak in the variables chosen for signal yield extraction can essentially not be identified without applying a continuum background suppression.

3.4. Common Approaches to Continuum Background Suppression

The general task for a continuum background suppression (usually just referred to as *continuum suppression*) is to find some variable that allows for differentiation between signal and background events by means of placing a cut on that variable. This becomes a non-trivial task when high background rejection has to be paired with high signal efficiencies, as required for a low multiplicity decay with high background.
Signal events can be distinguished from the continuum background by their different topologies. In background events, where quarks of invariant masses below those of the b quark are produced, the remaining energy causes the subsequent hadronic showers to be strongly aligned with the direction of momentum of the initial quark and anti-quark. In the rest frame of the collision this means that the hadronisation will be confined to two back-to-back hadronic jets. For $B\overline{B}$ events however there is almost no energy excess, implying that the decay products will exhibit an approximately isotropic angular distribution with no correlation between the directions of decay products from the two produced B mesons (illustrated in fig. 3.4).

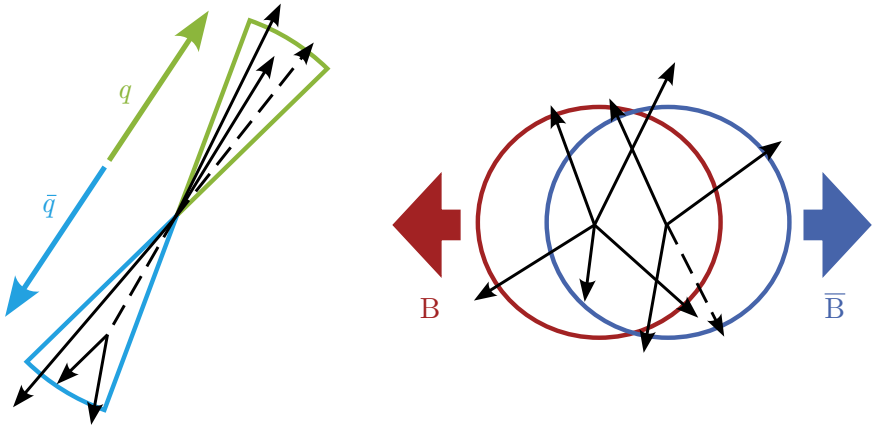


Figure 3.4.: Illustration of the different event shapes for $B\overline{B}$ decay events and $q\bar{q}$ background events (adapted from [25]). The B mesons only very slowly move apart and their decay products are approximately spherically distributed. In $q\bar{q}$ events the hadronisation is confined to two back-to-back jets.

A fundamental concept to capture information regarding the event shape are thrust frames. A thrust frame is defined as a polar coordinate system where the z-axis points in the average direction of momentum in a decay (referred to as the thrust axis). The boost is that of the rest frame of the collision. We define two thrust frames for a reconstructed event: one for the decay of the signal B, meaning the reconstructed B, and one for the rest of event, meaning everything reconstructed that was not matched to the signal B. The definition of signal B and rest of event is illustrated in fig. 3.5. For a $q\bar{q}$ event then the thrust axes will preferably align antiparallel while for $B\overline{B}$ events there is no preferred alignment.
To capture those differences, for continuum suppression commonly a set of "high level" event shape variables, like Cleo Cones, KSFW moments and further thrust related variables are used. These variables are specifically designed to be sensitive to the topological differences between continuum and signal described above. More detailed explanations of them can be found in [7].
To combine the information contained in all of those variables, one usually relies on methods from multivariate analysis. The tools at hand are boosted decision trees (BDTs), a form of recursive partitioning, or (deep) neural networks (DNNs). Those are then trained on simulated data, allowing for supervised training. The training target is a number encoding whether a simulated event is signal or background. Usually one chooses 1 for signal events and 0 for background events. As a BDT/DNN cannot be trained perfectly, its output will fall somewhere in the range between 1 and 0. If trained

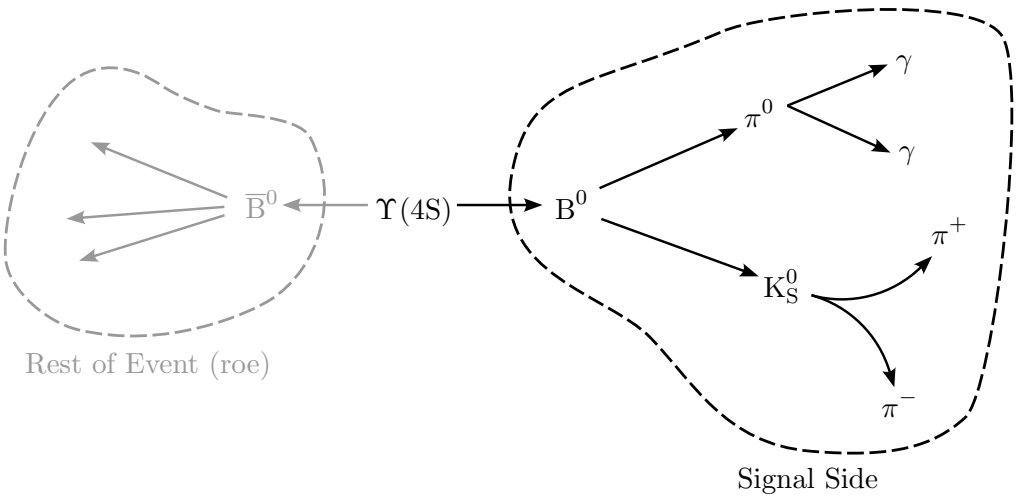


Figure 3.5.: Illustration of the decay of a $\Upsilon(4\mathrm{S})$ into a $\mathrm{B}\overline{\mathrm{B}}$ pair. The B^0 on the right is shown to decay in the signal mode for $\mathrm{B}^0 \to \mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$. This B^0 is referred to as the *signal* B. The decay products from the remaining $\overline{\mathrm{B}}^0$ are referred to as the *rest of event* (roe). As the shown signal decay is also possible for $\overline{\mathrm{B}}^0$, the B^0 and $\overline{\mathrm{B}}^0$ in the drawing could as well be swapped.

correctly, most signal (background) events are assigned numbers close to 1 (0). In essence the training is equivalent to finding a (rather complicated) function that transforms a set of input variables to a single output variable suited for a continuum suppression cut. Such a function, be it in form of a BDT or DNN, in the following we refer to as a *classifier*.

While for many measurements a continuum suppression as outlined above may be sufficient, especially for low multiplicity decays the best possible continuum suppression is desirable. There have been some past investigations on the additional suppression power gained by introducing further continuum suppression variables. Some only introduce a few, like the total traverse momenta or Δz [17], while others included a very broad set of variables, comprising for example information on the decay vertices and low level momentum variables [14].

However, usually these *new* variables are paired with a set of the common (engineered) ones, the general idea being to augment the traditional approach. The additional variables may be specific to the decay to which the continuum suppression is to be applied. Examples are the momenta of the particles in the final state or the decay vertex positions of intermediate particles. While this allows for potentially more efficient continuum suppression, at the same time some generality is lost. This can complicate for example estimation of systematic uncertainties.

In this thesis a slightly different approach is pursued, where the engineered variables are discarded completely in favor of a new set based purely on low level variables, which will be introduced in section 4.2. The philosophy being that if one already attempts to augment the conventional variables by introducing a few "new" low level variables to capture information not available through the conventional ones, one may as well choose a set of exclusively low level variables. The event shape information should of course still be extractable from the chosen variables. This could be thought of as as a more direct approach as it is attempted to find a function to *directly* calculate a highly effective continuum suppression variable from exclusively fundamental variables.

In this thesis we demonstrate that such an approach does function, but also address and discuss the difficulties and problems introduced by it.

4. Continuum Background Suppression for $B^0 \to K^0_S(\pi^+\pi^-)\pi^0(\gamma\gamma)$

4.1. Reconstruction and Selection for $B^0 \to K^0_S(\pi^+\pi^-)\pi^0(\gamma\gamma)$

For the reconstruction we follow a present approach [22] which will be outlined here.

Reconstruction and preliminary selection The $K^0_S \to \pi^+\pi^-$ candidates are reconstructed from $\pi^+\pi^-$ candidates, which in turn have been reconstructed as charged tracks in the detector. The dipion invariant mass is required to lie in the region between 450 MeV and 550 MeV. The $\pi^0 \to \gamma\gamma$ candidates are reconstructed from the detected photons, which are required to have a minimal energy depending on the calorimeter region they have been detected in. We require the energies to be larger than 22.5 MeV for the forward region and larger than 20 MeV for the barrel and backwards regions. Further the π^0 mass is required to lie between 105 MeV and 150 MeV and the absolute value of the cosine of the helicity angle of the π^0 is required to be less than 0.98. These criteria are imposed in order to suppress contributions from misreconstructed π^0 candidates. A mass-constrained fit is applied to the π^0 after the initial reconstruction.
The B meson candidate is reconstructed by combination of the reconstructed K^0_S and π^0 candidates. Here we impose constraints on two kinematic variables, the beam constrained mass, denoted M_{bc} and the difference between the reconstructed energy and half the center of mass energy, denoted ΔE. The definitions are as follows:

$$M_{\mathrm{bc}} = \sqrt{E^2_{\mathrm{beam}} - \vec{p}^{\,2}_{\mathrm{B}}}, \quad \Delta E = E_{\mathrm{B}} - E_{\mathrm{beam}}, \tag{4.1}$$

where E_{beam} is half the energy of the collision in the center of mass system of the collision. We require $5.2\,\mathrm{GeV} < M_{\mathrm{bc}} < 5.3\,\mathrm{GeV}$ and $|\Delta E| < 0.3\,\mathrm{GeV}$. Further a vertex fit is applied to the complete decay.

Final selection For the final selection we re-compute the kinematic variable M_{bc} to have no direct dependency on the magnitude of the π^0 momentum. The π^0 momentum is reconstructed purely from the photons detected in the ECL and is usually measured rather poorly compared to the track momenta[1].

$$M'_{\mathrm{bc}} = \sqrt{E^2_{\mathrm{beam}} - \left(\vec{p}_{K^0_S} + \frac{\vec{p}_{\pi^0}}{\left|\vec{p}_{\pi^0}\right|} \sqrt{\left(E_{\mathrm{beam}} - E_{K^0_S}\right)^2 - m^2_{\pi^0}} \right)^2}. \tag{4.2}$$

For the final selection then the following criteria are applied:

$$5.24\,\mathrm{GeV} < M'_{\mathrm{bc}} < 5.3\,\mathrm{GeV}, \quad |\Delta E| < 0.3\,\mathrm{GeV}, \tag{4.3}$$

$$482\,\mathrm{MeV} < m_{K^0_S} < 513\,\mathrm{MeV}, \quad 120\,\mathrm{MeV} < m_{\pi^0} < 145\,\mathrm{MeV}. \tag{4.4}$$

Off-resonance Where off-resonance data will be used, the same reconstruction and requirements are applied, except for M'_{bc} where $5.2\,\mathrm{GeV} < M'_{\mathrm{bc}} < 5.26\,\mathrm{GeV}$ is required instead to account for the shift introduced by the different beam energy.

[1] The default version of M_{bc} is also known to enhance correlations with ΔE, which is a further common reason for re-calculation. However this is of no further concern for this thesis as here M_{bc} will not be used for a signal yield fit.

4.1.1. Topologically Similar Control Channel

$\mathrm{B}^0 \to \overline{\mathrm{D}}^0(\mathrm{K}^+\pi^-)\pi^0(\gamma\gamma)$ was selected as a topologically similar control channel to be used in the verification of MC modeling, as well as for some studies with the trained classifiers. The structure of two charged tracks and two photons is the same as for the signal channel. The reconstruction applied is exactly the same as for $\mathrm{B}^0 \to \mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$ up to the replacement of K^0_S by $\overline{\mathrm{D}}^0$.
The most notable difference is the, compared to K^0_S, much shorter lifetime of the $\overline{\mathrm{D}}^0$. This implies that the decay vertex positions are different to those of the signal channel, which one has to keep in mind when comparing evaluations of classifier performance on this control channel with the signal channel.
Another candidate for a topologically similar control channel would be $\mathrm{B}^+ \to \overline{\mathrm{D}}^0(\mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma))\pi^+$. While this decay is of higher multiplicity than the above introduced control channel, there are some difficulties with it due to the intermediate $\overline{\mathrm{D}}^0$ state. For example it is not clear how to define the rest of event to make it comparable to that of the signal channel. While the $\mathrm{B}\overline{\mathrm{B}}$ pair is almost at rest in the center of mass frame for the signal channel, in this control channel the rest of event would originate from a B still approximately at rest in the center of mass frame, while the K^0_S and π^0 originate from a boosted $\overline{\mathrm{D}}^0$. The question then is in which frame to represent the variables used for the continuum suppression. The consequences of whichever choice are not immediately clear.

4.1.2. Data Samples Used

The following are the data samples used throughout this thesis:

- Generic MC[2] ($q\bar{q}$ where $q = \mathrm{u}, \mathrm{d}, \mathrm{s}, \mathrm{c}$ & $\mathrm{B}\overline{\mathrm{B}}$): $1\,\mathrm{ab}^{-1}$
- Pure signal MC for signal channel and control channel: 4×10^6 and 2×10^6 events produced resulting in 1 019 638 and 523 183 reconstructed events respectively
- Physics data: $361.65\,\mathrm{fb}^{-1}$
- Off-resonance generic MC ($q\bar{q}$ where $q = \mathrm{u}, \mathrm{d}, \mathrm{s}, \mathrm{c}$): $169.328\,\mathrm{fb}^{-1}$
- Off-resonance data: $42.28\,\mathrm{fb}^{-1}$

The above listed MC samples do not contain any $\tau^-\tau^+$ contributions. While $\tau^-\tau^+$ pairs are produced in the collisions, for analyses of the decays of B mesons they are usually already rejected by a pre-selection (skim) of the data that happens prior to reconstruction. Therefore $\tau^-\tau^+$ was expected to be a negligible background for the reconstructed decay and was not included in the reconstruction for the MC samples. Despite initial assumption, the data available for this thesis eventually turned out to not have the skim applied as intended. This as well as a further problem with not correctly applied momentum and energy corrections for the physics data is discussed in detail in appendix A.1. In essence the MC samples and thus also trainings and evaluations of the classifiers on MC are unaffected. Comparisons of physics or off-resonance to MC however may be affected. This will be discussed in context where necessary.
For off-resonance only $q\bar{q}$ events for quarks lighter than b are possible, which is why for off-resonance only those components of MC were reconstructed.
Generic MC and data were reconstructed for both, the signal channel $\mathrm{B}^0 \to \mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$ as well as the control channel $\mathrm{B}^0 \to \overline{\mathrm{D}}^0(\mathrm{K}^+\pi^-)\pi^0(\gamma\gamma)$. Off-resonance MC and off-resonance physics data was only reconstructed for the signal channel.

[2] Generic MC in the context of this thesis always means run-independent MC. For future studies one could e.g. use run independent MC for the training and run-dependent MC for later studies using the classifiers.

4.2. Continuum Suppression Variables

4.2.1. Introduction of Variables Used

In the following the chosen continuum suppression variables will be introduced and motivated. Some examples of their distributions are given and finally the used naming scheme is introduced. As the naming scheme can only be introduced sensibly after the variables have been introduced conceptually, it is explained last. For the plots shown prior to introduction of the naming scheme, the corresponding caption will explain the shown variables sufficiently to make sense of them without knowledge of the naming scheme.

Momentum Variables In the case of the decay $B^0 \to K^0_S(\pi^+\pi^-)\pi^0(\gamma\gamma)$, the final state particles measured by the detector are two charged pions and two photons, which are detected in the VXD and CDC as tracks or ECL as clusters respectively. The reconstruction gives for each event at least two charged tracks, one positive and one negative (on the signal side for the two pions), as well as two clusters (on the signal side for the two photons). All reconstructed tracks and clusters, for signal side and rest of event separately, are ranked by momentum in the center of mass frame to then select the highest ranking ones for each type to be used for the continuum suppression variables.
The numbers of reconstructed tracks and clusters chosen for continuum suppression from the rest of event were chosen to be the same as for the signal side. This means one positively charged track, one negatively charged track and two clusters. Chosen are the tracks and clusters of highest momentum as they are the most significant. Attempts to include more than only the highest ranking ones showed that the suppression power gained is very limited. For decays different than the one considered here however the effectiveness of inclusion of more tracks and clusters from the rest of event should be reevaluated.
For each of the tracks and clusters then a **representation of their momenta in a thrust frame** is computed to be used for the continuum suppression. Two different thrust frames are used: the signal side thrust frame, defined by the momenta of the signal decay, and the rest of event thrust frame, defined by the momenta of *all* reconstructed tracks and clusters in the rest of event.
Some suppression power is obvious to be inherent in this representation: For example, a signal side track momentum in the corresponding rest of event frame will for signal events have no real preferred direction relative to the thrust axis. For background events however, due to the jet like structure the relative orientation of signal side thrust frame and rest of event thrust frame is essentially fixed (with the thrust axes oriented antiparallel). In this case the angle between thrust axis and momentum vectors will be generally small. This implies that the cosine of the angle between the signal side tracks momenta and the rest of event thrust axis will be distributed almost flat for signal events while for background events the distribution is peaked at ± 1. An example for this is shown in fig. 4.1 (upper left). While for this example the different distributions can be intuitively explained, for example the distributions of the azimutal angle of the signal track momenta in the rest of event thrust frames[3] (example shown in fig. 4.1, upper right) are more difficult to interpret. Anyhow, the distributions for signal and background events are clearly different, suggesting the suppression power of the variables, which is sufficient here. Correlations between variables may also contribute some suppression power. Thus some variables with seemingly very similar distributions for signal and background events may also be useful for the background suppression.

Vertex Variables Further we introduce, this time only in the thrust frame of the signal side[4], the **vertices corresponding to the two charged tracks** (for both signal side and rest of event) as

[3] The oscillating component for the azimuthal angle is most likely related to the center of mass frames z-axis being the detectors z-axis which ends up not exactly aligned with the direction of the boost.

[4] The variables are not introduced for the rest of event thrust frame as by doing so the only information gained should be the same as already conveyed by the momentum variables being represented in both thrust frames.

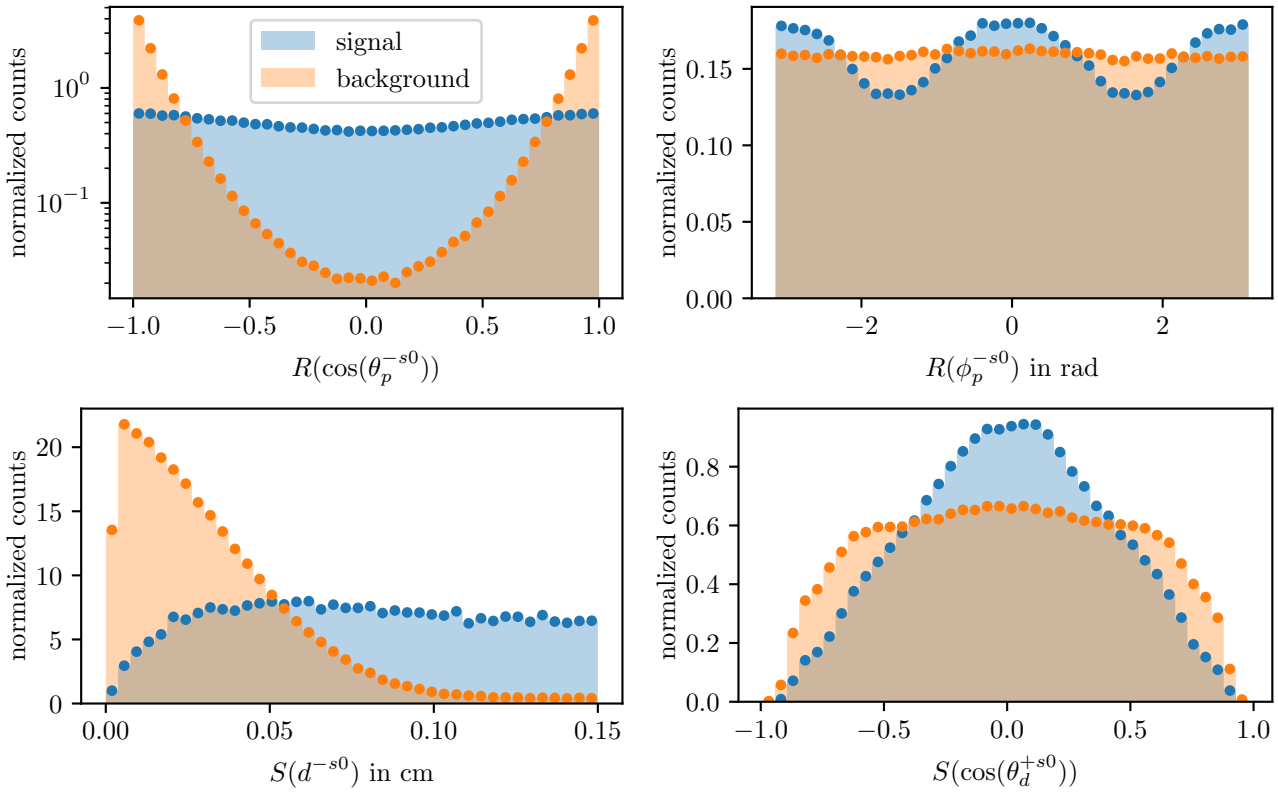


Figure 4.1.: Examples for different distributions of signal/background of used continuum suppression variables. Shown are polar and azimutal angle of momentum of the negative track of the signal side (π^-) in the thrust frame of the rest of event (upper left and right) as well as length and polar angle of the displacement vector corresponding to the decay vertex assigned to the negative track of the signal side (lower left and right). The variable naming is explained in detail in section 4.2.1. All distributions are normalized.

input variables. On the signal side, for signal events the vertices assigned to the tracks then correspond to the decay vertex of the K^0_S. As for the photons in the final state the ECL cannot determine their directions accurately, reliable vertex reconstruction is not possible. Thus vertex variables are only used for the charged tracks.
For signal events, on the signal side the vertices will be displaced from the interaction point by the amount the K^0_S traveled. For $q\bar{q}$ background, while per the reconstruction the tracks should still be matched preferably to the K^0_S contained in the jets, those now may be of different boost. Due to the large number of tracks in the jets, there also is the possibility for reconstruction of some fake K^0_S. Considering the corresponding distribution of distance traveled by the reconstructed K^0_S (shown in fig. 4.1, lower left), clearly the K^0_S in signal decays tend to travel farther.
Further a vertex position is assigned separately to each of the two tracks as a result of the vertex fit. For signal events the vertices associated with the positive and negative track should be exactly the same. Per the reconstruction however small differences are possible. Especially for background events non-coinciding vertex positions may be more frequent as the matched particles may be secondary particles in the hadronic shower from a $q\bar{q}$ event. This is of course already suppressed by the nature of the reconstruction where tracks are required to originate from approximately the same point.
While, as outlined above, there is some intuitive motivation for introduction of the decay vertex variables, this is also motivated by the work of Weyland [14], where such variables were already shown to be effective for improvement of continuum suppression. Their suppression power is further

suggested by the indeed different distributions for signal and background for for example the angular components of the vertex vector. An example for an angular component is shown in fig. 4.1 (lower right). Finally also the traveled distance of the signal B projected onto the z-axis (Δz) is added to represent its decay vertex[5].

Frame Orientation Variables In order to also capture the relative orientations of the thrust frames as well as the absolute orientation of those in the detector, (the cosine of) the angle between the thrust axes as well as between signal thrust axis and z-axis are used as further variables.

Fit Variables The classifiers are trained for the purpose of being applied to a sample from which then, by means of a fit, a signal yield is to be extracted. One has to decide which variables to use for the fit, the canonical choice being ΔE and M_{bc}. A popular replacement for M_{bc} is the *probability integral transform* of the classifier output. The transform is computed by evaluating the cumulative distribution function of the classifier output distribution for only signal events. This can be denoted as $F_X(Z)$, where X is the classifier output sample for only signal events and Z the classifier output sample for all events. As it can be shown that $F_X(X)$ is of standard uniform distribution, in the transformed distribution $F_X(Z)$, the signal part will appear as a constant component. The background part usually takes an approximately exponential form. If the background part can be sufficiently modeled, the signal yield can be simply extracted as a constant component.

If M_{bc} is not used for the fit, it is customary to place a cut on the variable. An alternative is to use M_{bc} as an additional input for the continuum suppression. The classifier used will then learn what is roughly equivalent to placing a cut, while also being able to utilize any further information possibly encoded in M_{bc}. Here it is chosen to use M_{bc} (or more precisely M'_{bc}) as a further input variable for the classifiers.

Variable Naming

For the kinematic and vertex variables a naming scheme is used that encodes which frame they are represented in, which track or cluster they belong to as well as whether they correspond to the decay of the signal B or the rest of event. Superscripts are assigned to the variables indicating the corresponding track or cluster. The superscripts are three symbols each, where the first indicates if the variable corresponds to a positive track (+), negative track (−) or cluster (0). The second symbol is either s or r indicating correspondence to the signal side or rest of event. Finally the appended number indicates the order of the track or cluster (as sorted by momentum). As in the context of this thesis we chose to only use one track of each type and two clusters, for tracks this number is always 0 and for clusters it is either 0 or 1. The used symbols to which the superscripts are assigned are listed and explained in table 4.1. For the polar angles usually the cosine of the angles are used. To further express in which frame the variable is represented in, we write $S(\cdot)$ or $R(\cdot)$ for the signal side and rest of event thrust frames respectively (the dot is to be replaced by the corresponding variable).

For example the momentum p of the zeroth positively charged track from the rest of event represented in the thrust frame of the signal side would be denoted as $S(p^{+r0})$.

The remaining variables, for which the above notation is not sensible are Δz, M'_{bc}, $\cos(\theta_{SR})$ and $\cos(\theta_{Sz})$. The latter two are denoting the cosine of the angle between the z-axes of signal and rest of event thrust frames and the cosine of the angle between the z-axis of the signal thrust frame and the z-axis of the detector respectively.

[5] There are further variables describing the B decay vertex. They were not available with the data samples used here but should be considered for future studies.

variable name	explanation
p	magnitude of momentum vector
θ_p	polar angle of momentum vector
ϕ_p	azimuthal angle of momentum vector
d	magnitude of vertex vector
θ_d	polar angle vertex vector
ϕ_d	azimuthal angle of vertex vector

Table 4.1.: Explanations of variable names used for the naming scheme.

Correlations and Final Variable Choice

The momentum magnitude variables are by definition the same in the different frames and thus always 100% correlated. As there is no information gained by keeping both of them, one will be discarded. It is chosen to discard the representations in the rest of event thrust frame, but the choice is arbitrary. The final set of variables is shown in table 4.2. The distributions for signal and background events for all of the chosen variables (for MC samples) are shown in appendix A.2.1 for the signal channel and appendix A.2.2 for the control channel.

Δz	$R(\cos(\theta_p^{+s0}))$	$R(\phi_p^{0s0})$	$S(\cos(\theta_p^{-s0}))$	$S(\phi_d^{-r0})$	$S(p^{+s0})$
$\cos(\theta_{SR})$	$R(\cos(\theta_d^{-r0}))$	$R(\phi_p^{0s1})$	$S(\cos(\theta_p^{+r0}))$	$S(\phi_d^{-s0})$	$S(\phi_p^{0r0})$
$\cos(\theta_{Sz})$	$R(\cos(\theta_d^{-s0}))$	$R(\phi_p^{-r0})$	$S(\cos(\theta_p^{+s0}))$	$S(\phi_d^{+r0})$	$S(\phi_p^{0r1})$
M'_{bc}	$R(\cos(\theta_d^{+r0}))$	$R(\phi_p^{-s0})$	$S(\cos(\theta_d^{-r0}))$	$S(\phi_d^{+s0})$	$S(\phi_p^{0s0})$
$R(\cos(\theta_p^{0r0}))$	$R(\cos(\theta_d^{+s0}))$	$R(\phi_p^{+r0})$	$S(\cos(\theta_d^{-s0}))$	$S(p^{0r0})$	$S(\phi_p^{0s1})$
$R(\cos(\theta_p^{0r1}))$	$R(\phi_d^{-r0})$	$R(\phi_p^{+s0})$	$S(\cos(\theta_d^{+r0}))$	$S(p^{0r1})$	$S(\phi_p^{-r0})$
$R(\cos(\theta_p^{0s0}))$	$R(\phi_d^{-s0})$	$S(\cos(\theta_p^{0r0}))$	$S(\cos(\theta_d^{+s0}))$	$S(p^{0s0})$	$S(\phi_p^{-s0})$
$R(\cos(\theta_p^{0s1}))$	$R(\phi_d^{+r0})$	$S(\cos(\theta_p^{0r1}))$	$S(d^{-r0})$	$S(p^{0s1})$	$S(\phi_p^{+r0})$
$R(\cos(\theta_p^{-r0}))$	$R(\phi_d^{+s0})$	$S(\cos(\theta_p^{0s0}))$	$S(d^{-s0})$	$S(p^{-r0})$	$S(\phi_p^{+s0})$
$R(\cos(\theta_p^{-s0}))$	$R(\phi_p^{0r0})$	$S(\cos(\theta_p^{0s1}))$	$S(d^{+r0})$	$S(p^{-s0})$	
$R(\cos(\theta_p^{+r0}))$	$R(\phi_p^{0r1})$	$S(\cos(\theta_p^{-r0}))$	$S(d^{+s0})$	$S(p^{+r0})$	

Table 4.2.: The final set of variables. The names follow the conventions explained in section 4.2.1.

4.2.2. MC Modeling of Variables Used

As the variables introduced are not very commonly used for continuum suppression, care must be taken to verify their MC modeling. The classifiers will be trained on a MC sample and if this sample does not model the physics data sufficiently well, the classifiers may behave in unexpected ways when applied to physics data. As analyses at Belle II (and also in HEP in general) are usually conducted blind, the MC modeling in the signal channel cannot be verified by comparing an MC sample to physics data in that same channel. To work with this constraint, off-resonance data and the control channel (as introduced in section 4.1.1) are employed. The following comparisons are considered:

- Signal channel: off-resonance MC $\leftrightarrow$ off-resonance data
  - Allows for verification of MC modeling for $q\bar{q}$ background events
- On-resonance MC (with off-resonance M'_{bc} cuts) $\leftrightarrow$ off-resonance MC
  - Shows to which extend the MC modeling conclusions from the off-resonance comparison can be translated to on-resonance (physics) data

- Signal channel: side-band MC $\leftrightarrow$ side-band data (physics data where the usual M'_{bc} cut is replaced by $5.2\,\mathrm{GeV} < M'_{\mathrm{bc}} < 5.27\,\mathrm{GeV}$)
  - Allows for verification of MC modeling for $q\bar{q}$ and $\mathrm{B\overline{B}}$ background events together, also higher statistics than for off-resonance
- Topologically similar control channel: MC $\leftrightarrow$ data
  - Approximates complete set of events to be modeled, however signal events are only present in a very small fraction
- High purity control channel MC $\leftrightarrow$ data (prospective, not part of this thesis)
  - Possible candidate: $\mathrm{B}^+ \to \overline{\mathrm{D}}^0(\mathrm{K}^+\pi^-)\pi^+$
  - Could allow for verification of MC modeling (of only the track related variables, if above candidate is used) of signal events

No single comparison is suited for a judgment of all aspects of the MC modeling. Thus, the comparisons presented above are supposed to complement each other and should always be considered together.

Comparison of the Distributions

Here for each of the considered comparisons a series of plots was created, one for each variable, overlaying the distributions for the two compared types of data. $\sqrt{N}$ is taken as the uncertainty for each bin, where N is the number of events in that bin. Using these uncertainties, the pull is computed and also shown.
The distributions for the $\cos(\theta)$ like variables tend to be extremely peaked at ± 1. This makes it difficult to visualize them using a histogram, as possibly a large region between the peaks will be almost empty. To work around this, an invertible transformation is employed that flips the distributions in the regions $[-1, 0)$ and $[0, 1]$. This causes the peaks near ± 1 to now be located just above and below zero respectively. In mathematical terms the transformation is

$$T(x) := \begin{cases} -x + 1 & \text{for } x \geq 0 \\ -x - 1 & \text{for } x < 0 \end{cases} = -x + (2H(x) - 1)\,, \tag{4.5}$$

where $H(x)$ is the Heaviside step function. If the transformation is applied for a variable, the corresponding label of the plot will be $T(x)$, where x is the variable. As the transformation only makes sense for the cosine variables which take values exclusively between -1 and 1, the transformation will only ever be applied to those. Also note that the transformation will only be used for the plots, *not* as a general processing of the data.
All the plots can be found in appendix A.3, appendix A.4, appendix A.5 and appendix A.6 for the above listed comparisons respectively.

Discussion of MC Modeling

Below some of the most prominent discrepancies observed will be highlighted, referencing some example distributions. Definitive conclusions are difficult due to the known problems with the available data samples explained in appendix A.1.

Off-Resonance Data vs MC For most of the variables most of the bins of the plotted histograms agree within 2.5 sigma. Notable exceptions are the momentum magnitudes, which for both tracks and clusters appear shifted. Some examples for the affected variables are shown in fig. 4.2. The observed disagreements here are expected to be related to the not applied momentum and energy corrections as mentioned in appendix A.1. Whether the disagreements for the track variables can be resolved by the very small track momentum corrections remains to be seen.

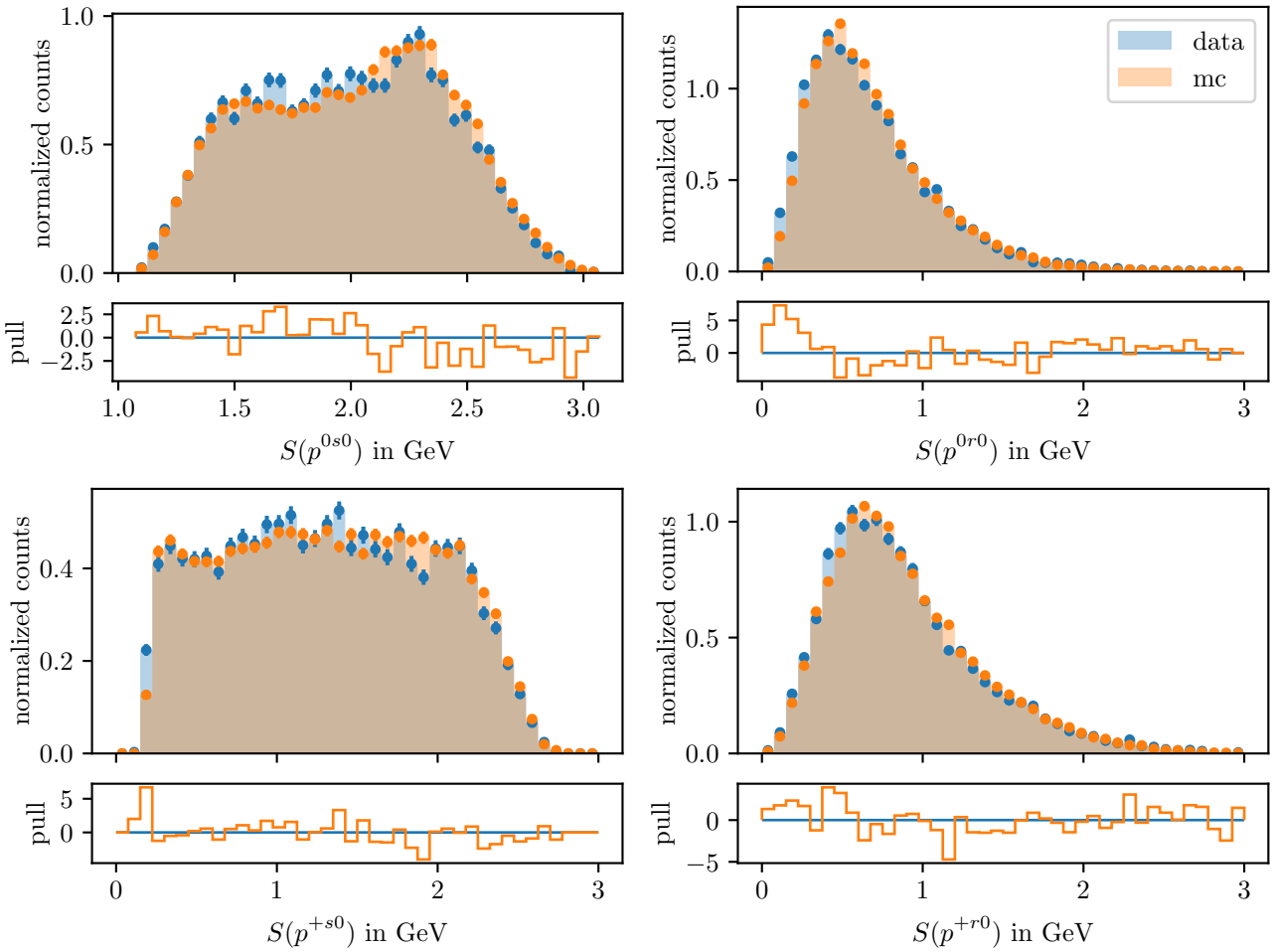


Figure 4.2.: Selection of distribution comparisons of off-resonance MC to off-resonance data where significant discrepancies could be observed.

Off-Resonance MC vs On-Resonance MC Overall agreement is surprisingly good considering that the two samples correspond to different beam energies. For a good fraction of the variables agreement is within 2 or less sigma. However, a few of the distributions are entirely off. For M'_{bc} a disagreement is expected as the variable is directly connected to the beam energy. For others like $S(\cos(\theta_p^{0s0}))$ a connection to the beam energy is not obvious but nevertheless large disagreements are observed. As the disagreements for a given variables are always either acceptable (meaning within 2.5 sigma) or extremely pronounced, we assume that the clear disagreements are an effect of the differing beam energies. Thus we conclude that for the variables used here $q\bar{q}$ background for off-resonance behaves essentially the same as for on-resonance up to some disagreements directly tied to the different beam energies. This however ideally should be further investigated in future studies.

Side-Band MC vs Data For the sideband statistics are higher than for off-resonance which highlights some of the disagreements that were hard to observe from the off-resonance comparison. Most notably a clear shift in M'_{bc} and $\cos(\theta_{Sz})$ is observed. This is a clear sign of a problem with the data as those are very common variables which are expected to be modeled well. Thus most likely the disagreements can be traced back to the not applied corrections. Notably now also some disagreements in the vertex variables become obvious. Whether the discrepancies are again caused by the not applied corrections is unclear and cannot be judged until a data sample with applied corrections

is available. Further the disagreements for momentum variables are now due to the higher statistics more pronounced. Examples for the above explicitly mentioned variables as well as an example for a vertex and momentum variable each are shown in fig. 4.3.

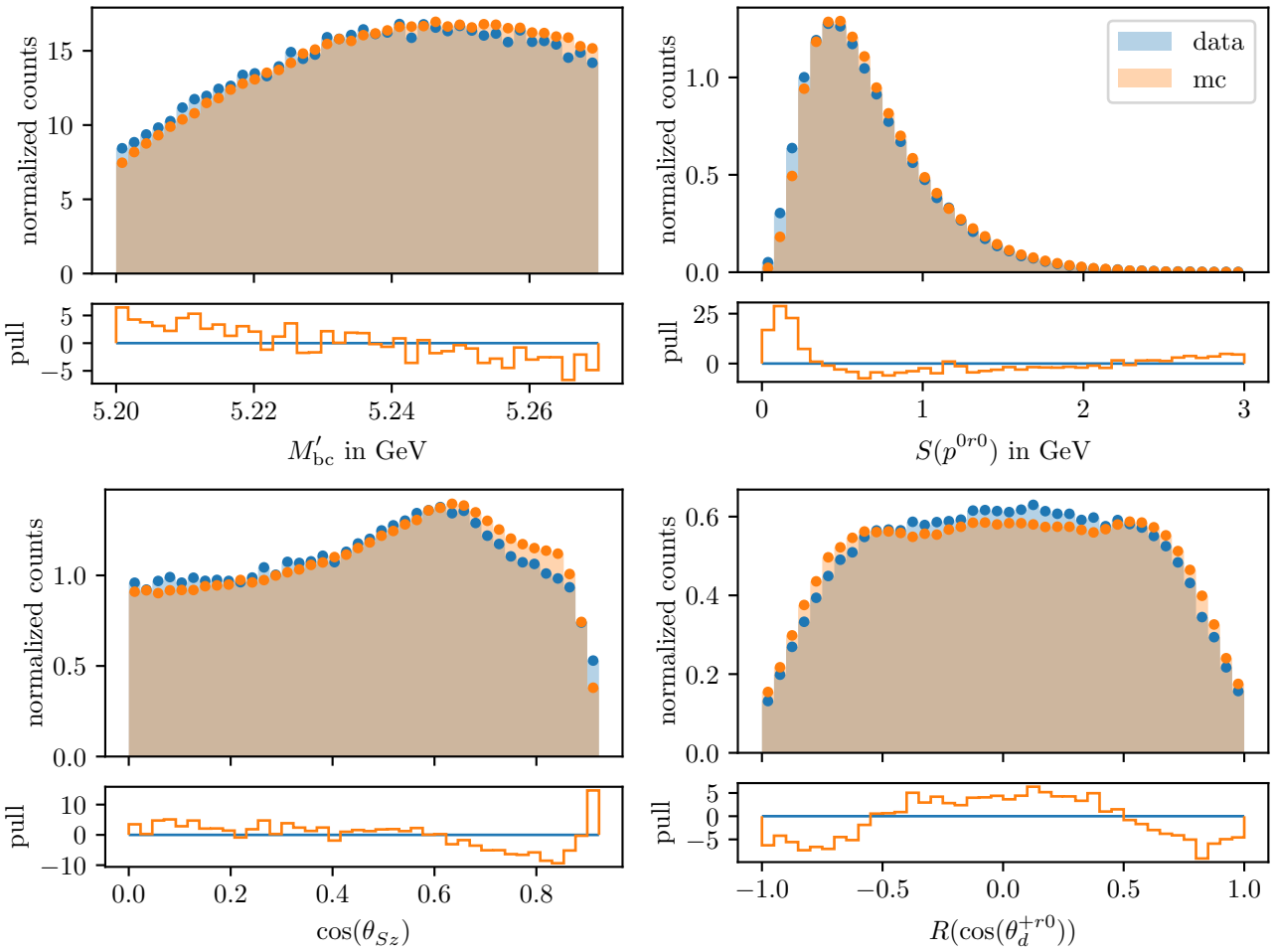


Figure 4.3.: Selection of example distribution comparisons of side-band MC to side-band data where significant discrepancies could be observed. For the momentum variables (here shown is $S(p^{0r0})$) the same disagreement as also observed for the off-resonance comparison (shown in fig. 4.2) can now seen to be more pronounced due to higher statistics.

Control Channel MC vs Data The same discrepancies as observed for the sideband comparison are also observed for the control channel. The only notable exception are the $S(\cos(\theta^{\pm\cdots}))$ like variables where some bins near the peaks at ± 1 disagree by around 10 sigma.

We conclude that given the known issues with the data MC modeling appears reasonable but note that a definitive conclusion is not possible with the available data samples.

4.3. Training of Classifiers

4.3.1. Data Samples for Training

There are two requirements for the data samples used for training and to some extend also the performance evaluation of the classifiers:

1. Each sample should contain the same number of signal and background events to avoid bias towards either of the categories.

2. The samples for training and evaluation of classifier performance during and after training should be completely disjoint.

The samples of same number of signal events are also generally needed for performance evaluation, as the fraction of signal events in the generic MC sample is very small.
With the available data samples, the background MC events for training must be taken from the generic MC sample. The signal events will be taken from the dedicated signal MC sample. To assure the disjoint samples for training and evaluation, the generic MC sample as well as the signal MC sample are first split into three parts. 60% of events will be dedicated to training (*training sample*), 10% to performance evaluation during training (*test sample*) and the remaining 30% to performance evaluation and studies of the classifiers after training (*validation sample*). For each of the training, test and validation samples from the generic MC only the $q\bar{q}$ background events are taken and combined with the same number of signal events from the corresponding part of signal MC. This then results in three samples of same number of signal and background events[6]. The above described process is illustrated in fig. 4.4. The designations training, test and validation sample will be used for both, the corresponding generic MC samples as well as the samples of equal number of signal and background events. Which of the two is used should either be clear from context or otherwise will be mentioned explicitly.

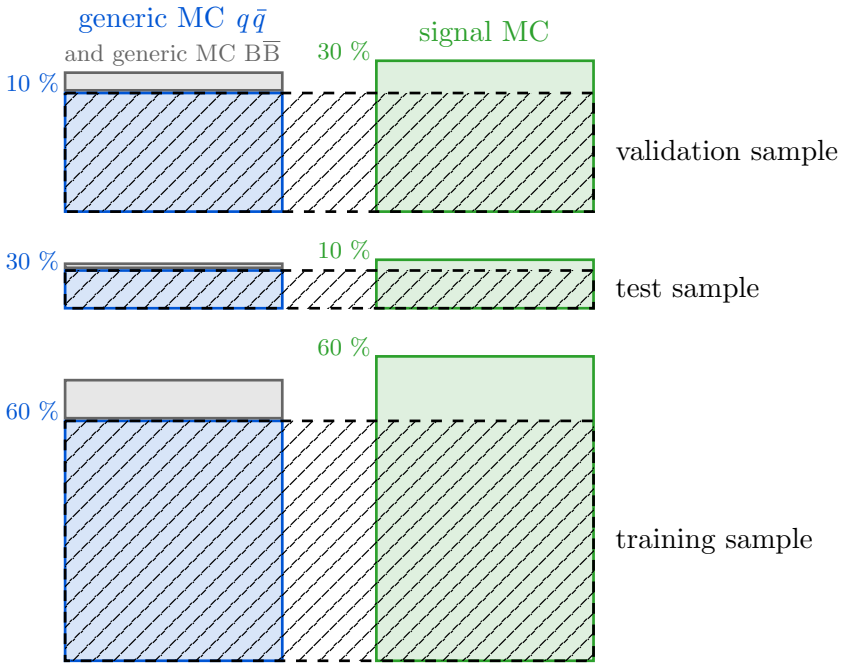


Figure 4.4.: Visualization of the division of the generic and signal MC samples to form training, test and validations samples of equal number of signal and background events.

[6]There happen to be more events in the signal MC sample than there are background events in the generic MC sample. This results in some portion of the signal MC sample remaining intentionally unused.

4.3.2. Base Loss Function and Performance Metrics

As the problem of continuum suppression is a binary classification problem, the base loss function for the classifier training is chosen as the *binary cross-entropy*. Mathematically the binary cross-entropy over some set of events (a batch) can be formulated as

$$\mathcal{L} = -\sum_{i \in \text{ batch}} \Big[y^i_{\text{true}} \log(y^i) + (1 - y^i_{\text{true}}) \log(1 - y^i) \Big]. \tag{4.6}$$

y^i_{true} are the labels which are 1 for signal and 0 for background and y^i are the predictions. During the training the index i then usually runs over a batch of events, meaning the events considered for one training step.
While the binary cross-entropy is used as the quantity to be minimized during training, for a measure of the performance of a trained classifier we rather choose the area under the *receiver operating characteristic* (ROC) curve, usually denoted as AUC. In the case of a continuum suppression, the ROC curve is the curve characterizing a classifier through its signal efficiency (fractions of signal events retained after continuum suppression) as a function of background rejection (fraction of background events rejected by the continuum suppression)[7]. The AUC can maximally reach 1 in the ideal case of rejecting all background events but not a single signal event. As this metric is much more computationally intensive, it is not suited for maximization during the training and is only ever computed after a completed epoch. An epoch meaning a complete iteration over the whole training sample corresponding to some number of optimization steps, depending on the chosen size of the batches. The AUC computed on the test sample is what is used to measure classifier performance during the training (at each epoch).

4.3.3. Introduction of Classifiers Used

Boosted Decision Trees (BTDs)

BDTs are known to produce very robust classifiers and are frequently applied for continuum suppression. Thus they are chosen as a reference to compare with the deep neural networks (DNNs, introduced in the next section) for this thesis. For the training of the BDTs the *LightGBM* implementation [12] was chosen. Compared to the DNNs, there are fewer hyperparameters to adjust for BDT training. As BDTs are known to be very robust, an in-depth tuning of hyperparameters was deemed unnecessary. The chosen hyperparameters have been selected manually, where care was taken to avoid overtraining (which can be regulated by choosing a sufficiently small number of leaves). The number of leaves was set to 20 and the learning rate to 0.01. Finally the parameter `min_sum_hessian_in_leaf` was used and set to 100 to make the BDT more robust against overtraining. Otherwise the defaults of the LightGBM implementation were used.

Deep Neural Networks (DNNs)

The focus in this thesis is on deep neural networks (DNNs) as classifiers. The initial motivation was that they were believed to possibly be more capable than the BDTs, especially when it comes to extracting information from the set of low level continuum suppression variables used for this thesis. DNNs however turn out to be much more delicate and difficult to handle than BDTs. The problems encountered in the application for continuum suppression will be discussed thoroughly in the following sections.
All neural networks used were implemented using *Tensorflow* [9] together with the *Keras* API [8] (which as of now is bundled with Tensorflow).

[7] ROC curves for the final classifiers are shown in fig. 4.12.

Neural Network Architecture In this thesis exclusively multilayer perceptron architectures with batch normalization layers between the dense layers are used. The batch normalization layers will learn the mean and variance of each input during the training and use this information to transform the inputs such that the corresponding output distributions are of mean zero and variance one. While the number of layers and neurons per layer are treated as tunable hyperparameters, the activation functions as well as batch normalization layers between the dense layers are always the same. The overall structure of data flow is illustrated in fig. 4.5. The very first layer is also a batch normalization layer which serves to normalize the raw input variables. This is followed by blocks of one dense layer, one activation layer and finally a batch normalization layer. For the activation functions *leaky rectified linear unit* (leaky ReLU) activation was chosen. This was found to speed up the training by orders of magnitude when compared to hyperbolic tangent activations[8] as used in the study by Weyland [14]. The in-between batch normalization layers prevent possible problems with very small or very large activations by re-scaling them before they are fed into the next dense layer. The number of blocks as well as the number of nodes in each dense layer is left as a tunable hyperparameter. In the following a network of n blocks will always be referred to as a network of n layers. After the last block one further dense layer reduces the activations to a single value which is finally mapped to the range $[0, 1]$ by a sigmoid function.

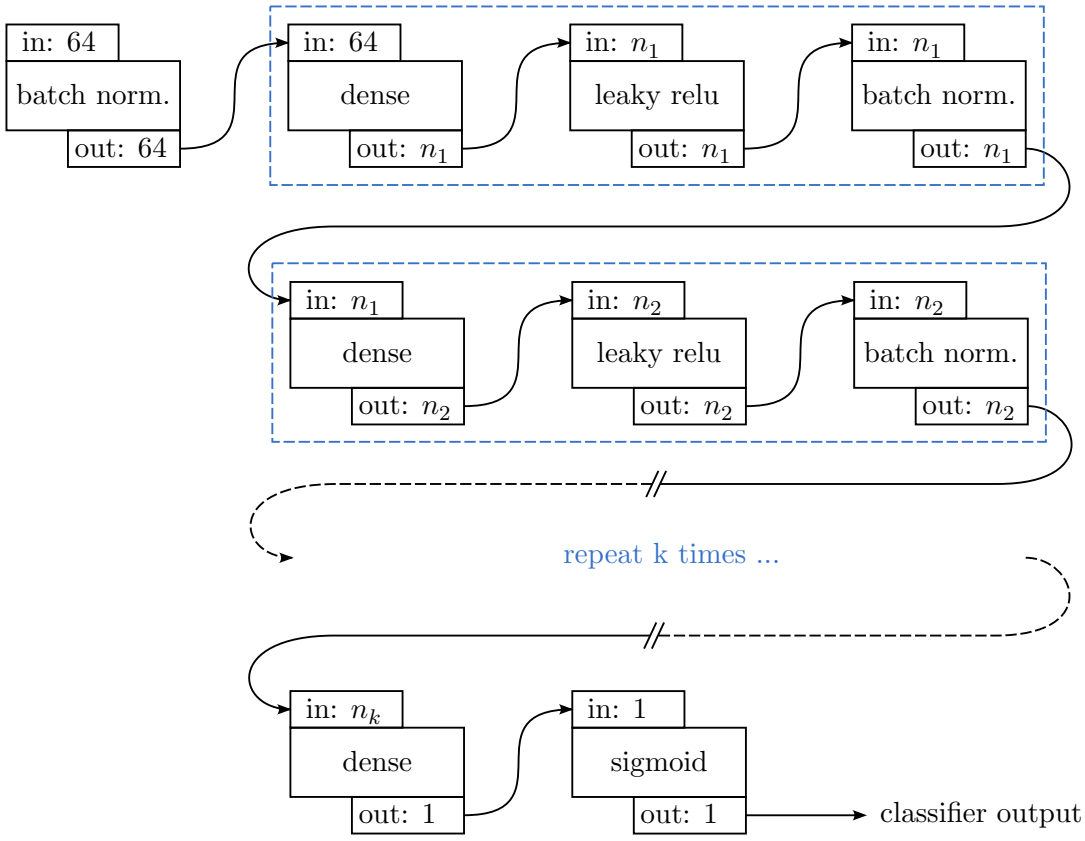


Figure 4.5.: Visualization of the structure of the used neural network architectures. The first layer is a batch normalization layer to normalize the raw input variables which is followed by a variable number of blocks of one dense layer, activation function layer and batch normalization layer. The last layer reduces everything to one output which is passed through a sigmoid function.

[8] Hyperbolic tangent activations are known to introduce problems with (almost) vanishing gradients in the optimization.

Optimizer and Training For the optimizer *AdamW* was chosen, which is an implementation of the popular Adam optimizer that also correctly incorporates *weight decay*, a way of regularization which serves to prevent overtraining [13, 20]. This optimizer has besides the learning rate one further hyperparameter for the weight decay. The learning rate can be controlled by means of a *learning rate schedule*. This is a common technique that can help with (faster) convergence. For this thesis however the increased complexity due to the additionally introduced hyperparameters was found to outweigh the benefits. Therefore a constant learning rate was used.

4.3.4. Decorrelation

The DNNs, if trained without any countermeasures essentially *always* evolve in a direction where the classifier output is strongly correlated with ΔE. This means that depending on the cut chosen for the classifier output, the shape of the distribution of ΔE for continuum background is sculpted to an extend where the signal peak is not clearly distinguishable from the background anymore[9]. A fit separating signal and background of such a distribution is of course not reliable. The sculpting issue must be addressed, as for an analysis here the final goal would be a signal yield fit, which happens after continuum suppression. If this cannot be achieved, any studies of the classifiers for the continuum suppression are of limited significance.
An example for the distribution of ΔE after a continuum suppression cut using both a DNN and BDT classifier is shown in fig. 4.6. The classifiers are applied to the generic MC (validation) sample. As a reference for the expected shape of the distribution after continuum suppression without any sculpting, the distribution for a sample with equal fraction of signal and background events is shown (on the left). The applied DNN is without any decorrelation measures, causing the corresponding ΔE distribution after the continuum suppression to be sculpted significantly. As long as no decorrelation measures are applied, the sculpting in ΔE occurred for all tested network architectures.
The observed sculpting behaviour is understood to be favoured in the training as it allows for many background events to be discarded relatively easily (those next to the signal peak) once the sculpting has been learned. Interestingly however sculpting for the BDT appears suppressed. Apparently a BDT is immune to such strong sculpting by the nature of its working principle.
Sculpting being connected to generally better background rejection also implies that performance scores cannot directly be compared if one of the classifiers introduces sculpting but the other does not.
Two methods for decorrelation of the classifier output and ΔE have been considered for this thesis and will be introduced below.

Adversarial Networks

Adversarial networks have been shown to be applicable for training neural networks such that their output is independent of a given set of nuisance parameters [11]. For the continuum suppression here the nuisance parameter corresponds to ΔE. Training with an adversarial network is set up in the following way: A predictive model outputs a prediction which then is fed into a second predictive model, the adversary model. The adversary model is supposed to predict *the distribution* of certain nuisance parameters from only the output of the first predictive model.
To incorporate the adversary into the training, a metric quantifying its performance in predicting the nuisance parameters distribution(s) from the first predictive models outputs must be chosen. The chosen metric is then incorporated into the loss function for the first predictive model. This must be done such that the model is encouraged to evolve into a direction where the adversary performs as bad as possible. The adversary model however also must be trained, where the loss is chosen such, that the adversary learns to better predict the nuisance parameter distribution(s) in order to catch up with the evolution of the other predictive model.

[9] The sculpting issue was already pointed out and addressed in [17], although there different continuum suppression variables were used. The taken decorrelation measures are also different to those taken here.

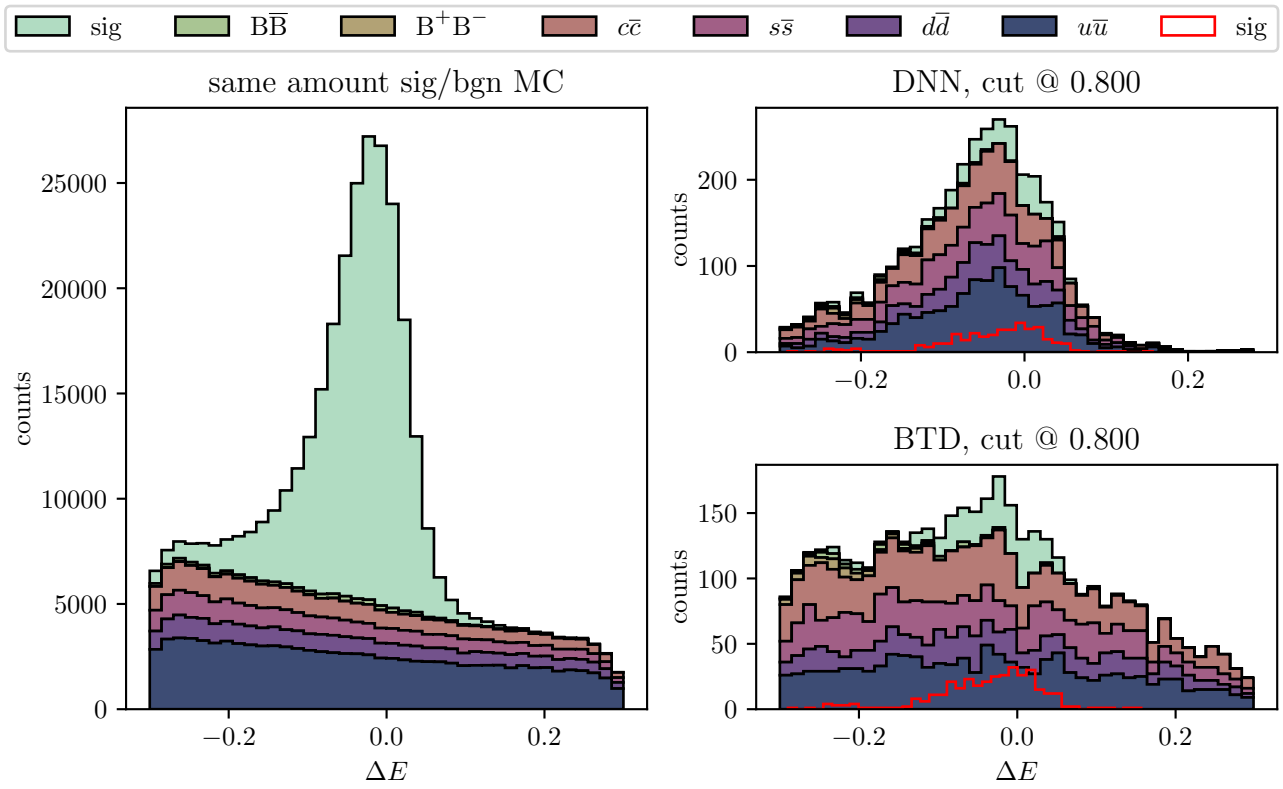


Figure 4.6.: ΔE distribution for MC sample with equal amount of signal/background events as example for signal/background distribution shapes (left) and distributions of generic MC sample after continuum suppression with DNN without decorrelation (upper right) and BDT (lower right). The continuum suppression using the DNN introduces significant sculpting of the background distribution of ΔE. The left plot shall serve as a reference of the expected (not sculpted) signal and background shapes after continuum suppression.

For the application for decorrelation here the first predictive model is the DNN used for continuum suppression. The adversary model is a second DNN producing parameters for a Gaussian mixture model to approximate the distribution of ΔE. A more detailed explanation as well as formulation of an adversary loss function can be found in [17].
Unfortunately there are some drawbacks with this method. First, for each training step of the main predictive model the adversary model must be trained for a few steps, significantly slowing down the overall training process. Besides this there is an inherent difficulty in the tuning of hyperparameters of which a great number is additionally introduced. This is because the overall loss function to be minimized is now significantly more complicated as it contains a component involving the output from the adversary model. If the many hyperparameters for the adversary and its contribution to the overall loss are not chosen correctly, it will turn out to "weak", and the classifier model may decent into the same global minimum as for training without adversary. On the other hand if the adversary is to "strong", its contribution to the loss will outweighing the classifier loss part, causing the classifier training to be unstable or completely fail. One has to find just the right configuration to balance the contributions to remain stable throughout the whole training. This is difficult because the balance is influenced by many hyperparameters.
Nevertheless using adversarial networks can be, and has been shown to be effective for decorrelation, even for the specific case of ΔE as the nuisance parameter in $\mathrm{B}^0 \to \mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$ (but with other input variables as used here) [14, 17]. Following the same approach, training with adversarial networks

was implemented during research for this thesis.
Due to the difficulties with tuning already encountered with the distance correlation method (introduced in the next section), which is supposed to be easier to tune, adversarial networks could not be pursued much beyond a proof of concept for this thesis. However the lessons learned from the hyperparameter tuning with the distance correlation decorrelation method are believed to be also applicable to adversarial networks. As the hyperparameter tuning is beyond the scale of this thesis and thus no properly decorrelated classifiers with adversarial networks could be reached, the results will generally always use the distance correlation method.

Distance Correlation

As an alternative to adversarial networks, decorrelation using distance correlation (referred to as *DisCo* in the following) can be used. This method has been shown to perform very similar to adversarial networks for decorrelation [19]. The central idea is to use an efficiently estimable correlation metric that is able to capture non-linear correlations, namely distance correlation, to punish the classifier the more its output is correlated with a given nuisance parameter. To do so, the correlation metric is simply scaled and added to the classifier training loss.
This addresses the main issue that comes with the use of adversarial networks: the number of hyperparameters. With DisCo there is only one hyperparameter, the scale of the distance correlation when added to the classifier loss. Besides the reduced number of hyperparameters, training times are also not increased as much as with adversarial networks, where usually more time is spend training the adversarial network than the actual classifier. The loss for a classifier with DisCo becomes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classifier}}(\vec{y}, \vec{y}_{\text{true}}) + \lambda \cdot \text{dCorr}(\vec{z}, \vec{y}), \tag{4.7}$$

where $\vec{y}$ is the vector of predictions for a batch, $\vec{y}_{\text{true}}$ the corresponding labels and $\vec{z}$ the corresponding nuisance parameters. $\text{dCorr}(\vec{z}, \vec{y})$ is the distance correlation[10] and λ the hyperparameter scaling it[11].
One can choose to compute dCorr on only the background events in each batch. This is valid here as the problematic sculpting only really occurs for the background part of the distribution of ΔE. Which choice should be favoured is however not immediately obvious. To archive the same effectiveness of the decorrelation, λ had to be chosen larger by a factor of around 7.5 when dCorr is computed on only background events. It is known that the used estimator for dCorr introduces a bias scaling with $\frac{1}{n}$, where n is the number of events in a sample [19]. This however would imply that for the now smaller subset of only background events in a batch the bias should be larger. dCorr however has to be scaled *up* for decorrelation to be effective. Thus some other effect must be at play. Possibly correlations for the signal part are inherently larger. This however was not further investigated here.
While there now is only one extra hyperparameter, the training was found to be still very sensitive to it. The classifiers would quickly evolve to produce still significantly correlated output if λ is not chosen correctly. Simply choosing a large value for λ is also not possible as this will impact final performance significantly. The process and results of the tuning are described in the next section.

Hyperparameter Tuning

Systematically tuning hyperparameters for a classifier with correlation countermeasures, here DisCo, is difficult as there are two conflicting objectives: the best possible classification performance (as for example measured by an AUC score) and sufficiently low correlation of the classifier output and ΔE. The problem is that a correlated classifier usually performs better for signal/background separation. Thus only tuning for classifier performance is not possible as λ would always be tuned to zero. Tuning

[10] For computation of the estimator for dCorr the implementation (for Tensorflow) from [19] was used.

[11] As pointed out in [19], technically the exponent of dCorr is another parameter to chose. However as it was found that here λ alone is already difficult to tune and as there further is no direct motivation for adjustments of the exponent, the exponent is always set to 1.

for a joint metric, like the total loss (including the DisCo term as shown in eq. (4.7)) over the test sample, introduces the difficulty of having to chose such a metric correctly in order to not have the performance or the correlation measure be overrepresented, essentially introducing hyperparameters of the hyperparameter tuning. This defeats the purpose of a systematic hyperparameter tuning. While there are ways to optimize hyperparameters for multiple objectives with the hyperparameter optimization framework *Optuna* [16], they are expected be rather inefficient as they require a significant number of trainings to be run[12]. An alternative approach could be to attempt to automate the fitting of the distributions after the continuum suppression and then optimize for lowest statistical uncertainty of the signal yield extracted with the fit. While this would be the ideal optimization objective, many technical difficulties are expected. Thus an implementation is far beyond the scale of this thesis.
The tuning is further complicated here as even with decorrelation measures, the DNNs would in most cases after a sufficient number of epochs still suddenly introduce significant sculpting in ΔE, making the final training result very unforeseeable. This will be further elaborated on in the following paragraphs.
As, outlined above, the decision of whether a distribution of ΔE is sculpted too much or not is highly non-trivial making it difficult to implement a tuning entirely in code. More sophisticated systematic tuning attempts are expected to introduce further (technical) difficulties in their own right and are thus not pursued in this thesis. Instead here the tuning was chosen to be conducted manually, verifying the quality of the distributions by eye. While this introduces some subjectivity, the differences between effective and insufficient decorrelation are pronounced enough to make the decision straight forward. During the manual tuning the evolution of the distribution of ΔE (after application of continuum suppression) would be monitored at each epoch, from which with some experience it was immediately obvious whether a distribution may be usable for a fit or not. This worked well as the sculpting in most cases was either only very slight or obviously too strong and transitions happened very quickly. To reach a configuration where the decorrelation using DisCo would remain stable throughout the whole training, first some studies with a preliminary set of hyperparameters were conducted. The preliminary hyperparameters were determined through manual tuning with the goal of reaching a usable configuration for preliminary studies. There was no real strategy besides simply attempting different parameters based on experience until something deemed usable was found. The main objective was to reach a configuration where sculpting is not immediately introduced. This had to be paired with reasonable training times and general stability of the training. Some of the hyperparameter choices here were also based on experience from prior attempts of tuning a classifier without decorrelation, as was done prior to acknowledging the severity of the sculpting in ΔE. The tuned hyperparameters are listed in table 4.3 with their preliminary values in the corresponding column.
One key observation during this preliminary tuning was that introduction of a bottleneck in the neural network significantly helps with suppression of sculpting. While not completely suppressed in most cases, the start of sculpting could generally be delayed much further compared to an architecture without bottleneck. Something similar was archived by choosing very small networks (two layers, less than 40 neurons each), which however also significantly impacted the final classification performance.

Evolution of ΔE Distribution To study the evolution of the sculpting behaviour with DNNs in order to better understand and eventually completely suppress it, the background distribution of ΔE was recorded at each epoch for a set of trainings. Here we show three examples for the different sculpting behaviours that were observed. The chosen examples are:

1. A DNN without any decorrelation measures (equivalent to $\lambda = 0$)

2. A DNN with slightly weakened DisCo-decorrelation to highlight start of correlations after sufficient epochs ($\lambda = 1$)

[12] This is also influenced by no support for pruning of trials when optimizing for multiple objectives.

	prelim. value	final value	description
n_{layers}	5	5	number of layers
$n_{\text{neurons},0}$	100	100	1st dense layer neurons
$n_{\text{neurons},1}$	100	100	2nd dense layer neurons
$n_{\text{neurons},2}$	4	6	3rd dense layer neurons
$n_{\text{neurons},3}$	100	100	4th dense layer neurons
$n_{\text{neurons},4}$	100	100	5th dense layer neurons
weight decay	0.000142	0.000142	Weight decay for AdamW
learning rate	0.002	0.015	learning rate
dCorr on bgn	True	True	choice to compute dCorr on only background events
λ	1.8	2	scale of dCorr in total loss
s_λ	7.5	7.5	scale factor for λ when dCorr computed on bgn only
batch size	2048	16384	number of events in a minibatch

Table 4.3.: Preliminary and final choice of the tuned hyperparameters for the DNN.

3. DisCo-decorrelated DNN with preliminary hyperparameters ($\lambda = 1.8$)

All three trainings used the preliminary hyperparameters as shown in table 4.3, except for λ which was adjusted accordingly.
The evolutions of the background distributions of ΔE after a continuum suppression cut (here chosen at 0.9) for the three trials are shown in fig. 4.7. The distributions have been normalized (meaning separately at each epoch) to highlight the changes in their shape rather than absolute scale.
If DisCo is not used, already after around 5 epochs the distribution is highly sculpted and stays sculpted throughout the training. If DisCo is applied, but the training is continued for sufficiently many epochs (with small enough λ), eventually sculpting suddenly starts. For the concrete example here this happens after around 120 epochs. The exact value however may vary depending on the hyperparameters, including λ. While less severe compared to the case without decorrelation, the sculpting already is to an extend where the signal peak is not clearly separable from the background anymore.
If further λ is tuned (i.e. preliminary hyperparameters are used), the start of significant sculpting can be further delayed and sculpting at the end of training can be slightly more suppressed.
It appears as if the networks, given they are large enough, are capable of evolution in a direction where the DisCo term has almost no effect. A possible interpretation could be the following: There may be enough freedom to choose a path through the parameter space to the global minimum (of the total loss) where at some point the gradients of the DisCo term are negligible compared to those of the binary cross-entropy term. Even if this imbalance occurs only temporarily, once the barrier introduced by the DisCo term has been overcome (or avoided) there is usually no way back to a less correlated classifier, as even if the loss in now biased by the DisCo term, the gradients may not be influenced if the bias is constant. This fits in with the classifiers usually training without significant correlations until a certain point where they then very quickly evolve, within a few epochs (of course also depending on exact hyperparameters) into a correlated classifier, and stay this way thereafter.
While the point of suddenly increasing sculpting can be generally delayed by adjusting the hyperparameters, complete suppression until convergence[13] is difficult. To delay the start of sculpting (possibly beyond the defined end of training) λ can be increased. However, too large values for λ were found to at some point cause the loss to be overwhelmed by the DisCo term, leading to unstable or suboptimal training. This effect might be significant especially later in the training where gradients of

[13]The definition of convergence of a DNN may vary as this depends on for example how many epochs are run as well as possibly an early stopping policy, if applied.

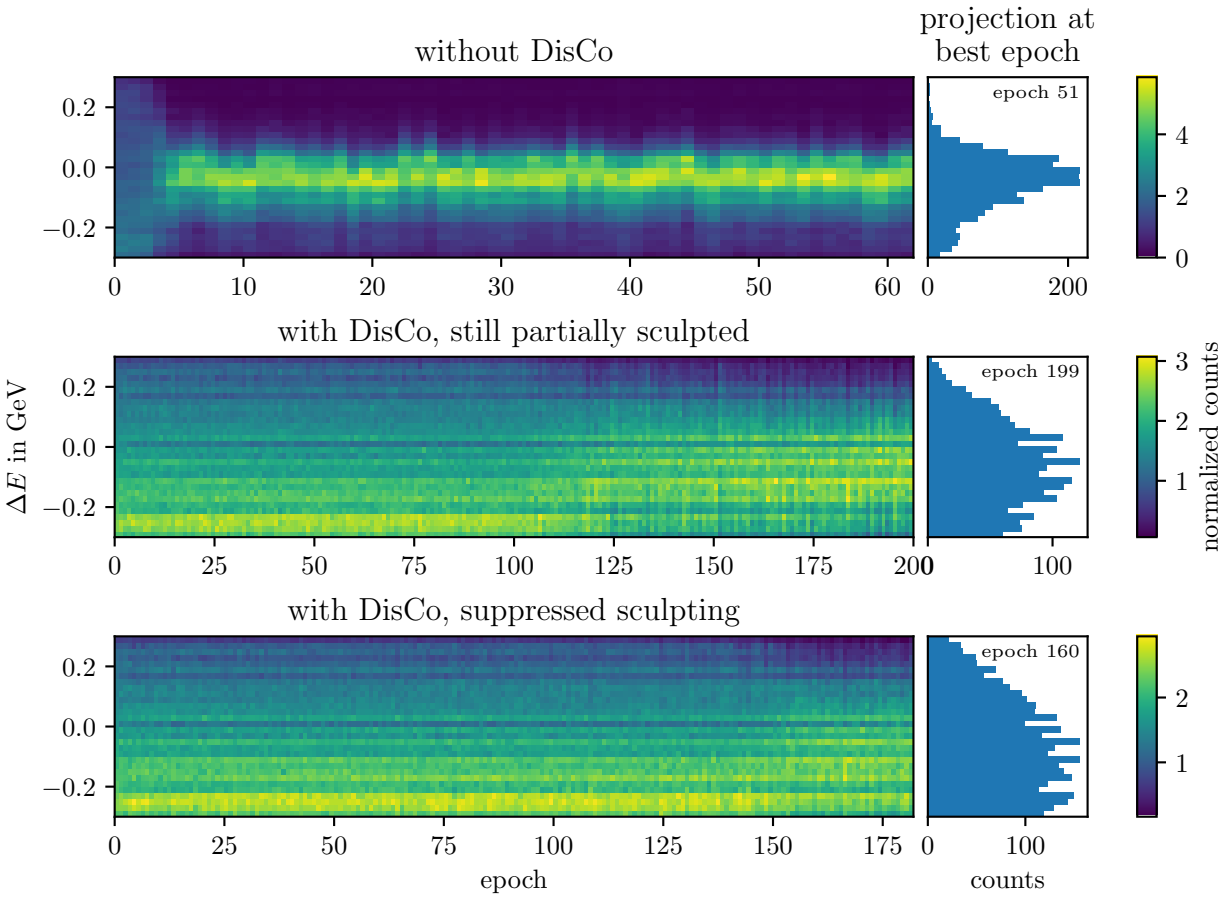


Figure 4.7.: Evolution of the distribution of ΔE (only background) throughout the training for three DNN examples: without DisCo, with DisCo but $\lambda = 1$ and with DisCo and $\lambda = 1.8$ (in that order). The remaining hyperparameters are the preliminary hyperparameters as listed in table 2.1.

the cross-entropy term are expected to be small[14]. Thus generally the smallest λ that still sufficiently suppresses sculpting throughout the entirety of the training is desirable. This is further assisted by the here chosen bottleneck architecture (as already mentioned above).

Coincidence of dCorr Increase and Sculpting Further also the coincidence of start of sculpting and an (expected) increase of dCorr was verified. For this a DNN was trained with three different batch sizes (16 384, 2048, and 512 events per batch), as if the batches are too small the distance correlation is expected to not be sufficiently representative of the whole training sample anymore. The resulting training histories are shown in fig. 4.8. The hyperparameters besides batch size and λ were the preliminary hyperparameters from table 4.3. λ was again set to 1 to reduce training times until sculpting starts. As the value of dCorr does fluctuate between steps to an extend where it can be hard to resolve the overall trend from the plots, the average over 100 steps each is also drawn (in red). For the smallest batch size any trend of dCorr appears to get lost in the fluctuations. For the larger batch sizes a clear trend is visible. The increase of dCorr around the epochs where sculpting begins is for both cases approximately 0.0006. Thus the quantification of the degree of sculpting by means of dCorr appears to be approximately independent of the sample size used for computation of

[14]Here one could attempt to decrease λ over the course of the training, which however was not further pursued here.

dCorr. This of course only holds as long as the overall trend is not overwhelmed by the fluctuations. dCorr can be seen to be biased by an amount depending on the sample size, as expected. As already mentioned above, the bias is an artifact of the used estimator for dCorr and scales with $\frac{1}{n}$, where n is the size of the sample (here a batch).
The takeaway from this study is that a larger batch size is desirable in order to ensure better numerical stability of the DisCo term. As evident from fig. 4.8, the preliminary batch size of 2048 appears to be smaller than optimal. Thus for the final hyperparameters 16 384 is chosen instead. However note that a too large batch size may also not be desirable as using too many events for a single optimization step will encourage overfitting. Here one could attempt to implement a training loop where dCorr is computed on more events than used for the classifier loss at each step. This could allow for even better numerical stability of dCorr while also retaining the effect of small batch sizes to prevent overfitting.

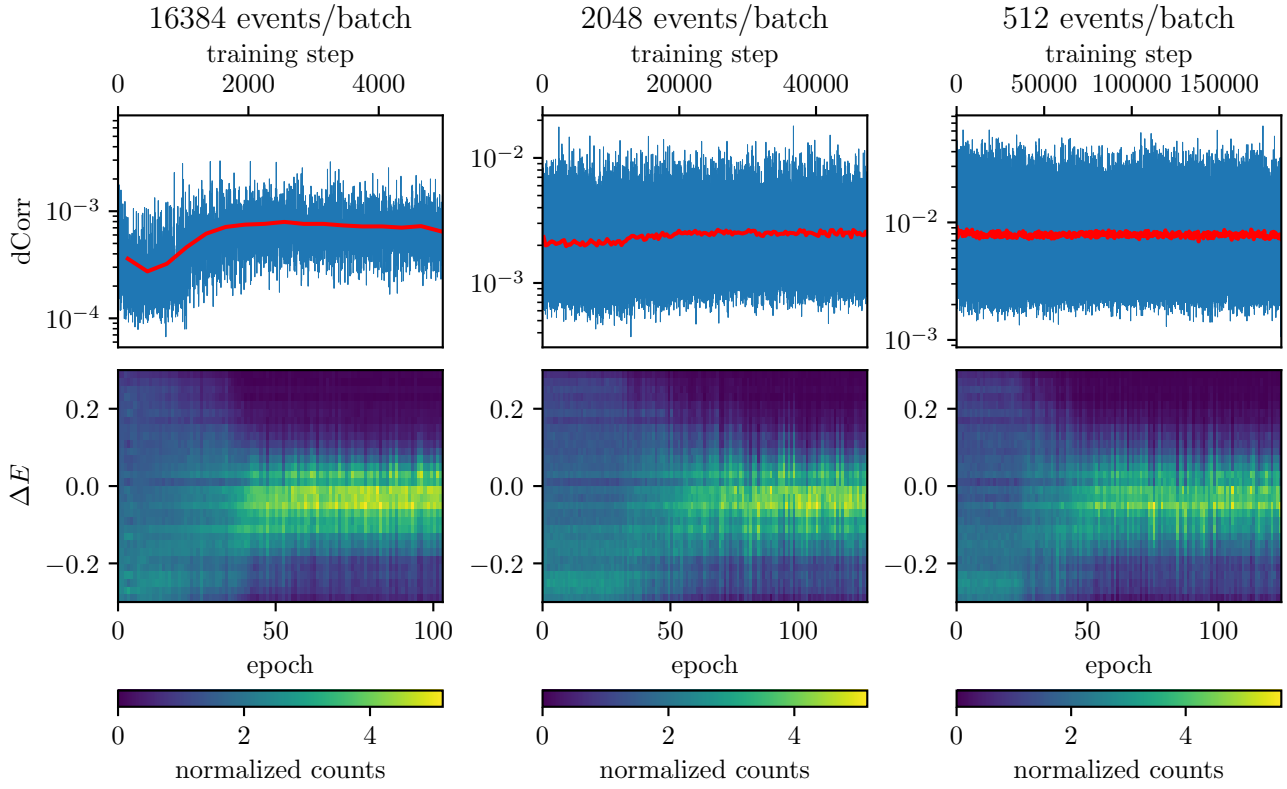


Figure 4.8.: Evolution of dCorr and distribution of ΔE throughout the training for three different batch sizes. The start of sculpting coincides with an increase in dCorr where however for too small batch sizes the increase is overwhelmed by the numerical instability of dCorr.

dCorr on test sample There still remains the concern of dCorr computed on the training sample possibly not sufficiently representing dCorr in general (similar to overtraining). For the trial with batch size 2048 the average of dCorr computed on the test sample was additionally recorded. For the technical reason of insufficient GPU memory on the available hardware but also to match the bias with dCorr computed at each training step, the computation was again done in batches of the same size as used for the training (meaning 2048). The results for dCorr for all batches corresponding to an epoch are then averaged. The resulting history of dCorr is shown in fig. 4.9 (upper plot). dCorr computed on the training sample appears to overall represent the correlations for the test sample reasonably well. Thus the conclusions deduced from results related to dCorr on only the training sample are assumed

to be sufficiently generalizable.

Total and Classifier Loss Next to dCorr at each epoch also the classifier loss (meaning only the binary cross-entropy) as well as total loss (see eq. (4.7)) are considered to assess the change in the total loss expected to coincide with the step in dCorr. This is expected as the classifier is not stably decorrelated yet. Total loss and classifier loss are shown in fig. 4.8 (lower plot).
It can be seen that during the transition to a sculpted distribution (around step 11 500, marked in the plot), the drop of the classifier loss is not entirely compensated for by the DisCo term as a (smaller) drop can still be observed for the total loss. This means that despite of the decorrelation measure, the total loss as a function of the parameter space still exhibits regions where a decrease is associated with the behaviour of introduction of sculpting. The overall takeaway is, that the DisCo term appears effective but not sufficiently strong throughout the whole parameter space. To address this, for the final hyperparameters we increase λ to 2. This could be later verified to result in stably decorrelated trainings.

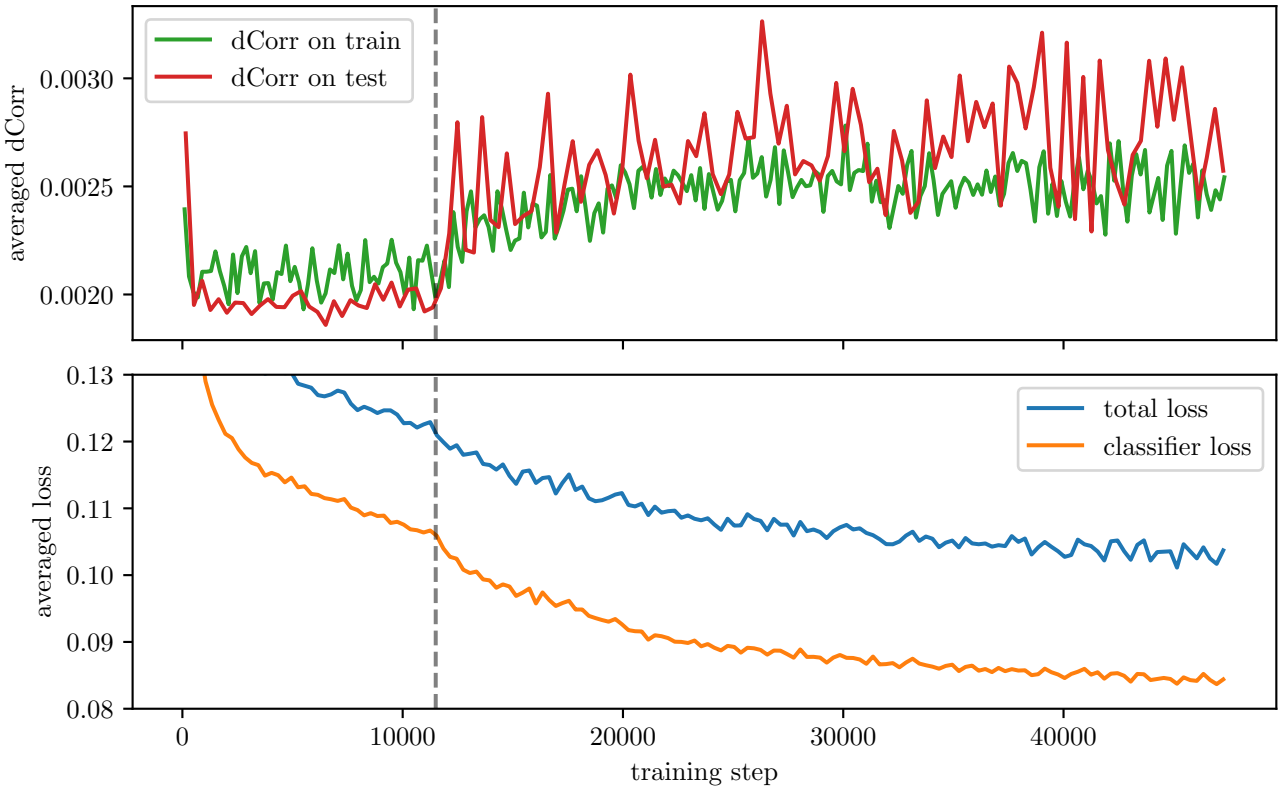


Figure 4.9.: History of dCorr computed on both the training and test sample (upper plot) as well as corresponding total and classifier losses (lower plot). Evidently the contribution of dCorr to the total loss is still to weak as a clear step in the total loss remains. The shown data is from the same training as used for the corresponding plot in fig. 4.8 (batch size 2048).

Final Hyperparameter Choice Incorporating the changes already mentioned in the discussions above, the final hyperparameters for the DNN in this thesis are fixed. The learning rate was increased to 0.015, as this would delay overtraining and thus yield overall better final performance. For the lower learning rates overtraining started to become an issue. Further the bottleneck was widened from only four nodes to six, as this was still sufficient for stable decorrelation. The complete set of final

hyperparameters is shown in the corresponding column of table 4.3. With the final hyperparameters five full trainings (with an early stopping after 20 epochs of no improvement) were run. From those the model with overall best AUC score was chosen to be used for the following studies. In the following this will be referred to as the *DisCoDNN*.

4.3.5. Handling of `NaN` Values

Reconstruction of some of the final state particles can fail for some events, which results in a value of `NaN` in some of the variables. A overview of the `NaN` occurrences relative to the total number of events in a sample is shown in fig. 4.10. Shown are 10 variables each for which the highest occurrences of `NaN` values in the corresponding sample was found. Further we consider this for generic MC signal/background separately as well as off-resonance physics data and MC. From the generic MC occurrences it is apparent that for background `NaN` occurrences are higher. Comparing occurrences for off-resonance MC and physics data shows that the occurrences in MC and physics data are approximately on the same order of magnitude. As no alarming differences are observable, we assume in the following that `NaN` values can be treated the same for all samples[15]. In most of the cases the variables related to the second photon in the final state are most probable to contain `NaN` values. Here the `NaN` occurrences are almost the same for all of them, indicating that the `NaN` values occur when reconstruction of the second photon fails completely.
As the neural network can only operate on floating point numbers, we must decide how to handle `NaN` values. Unfortunately there seems to be no single right procedure. A common thing to do is to replace them with some fixed value. For this thesis they are always chosen to be set to 0. Completely removing events with at least one `NaN` value would discard too many events (here up to 15%). It was found that classification is not influenced by setting `NaN` to zero in any significant way. The final classifiers were applied to a subset of events including at least one `NaN` value in one of the input variables as well as the subset of events without any `NaN` values. Interestingly the AUC score dropped by approximately 0.0005 (0.0003) when evaluated on only events without `NaN` values and increased by approximately 0.002 (0.001) when evaluated on only events with at least one `NaN` value, using the final DisCoDNN (BDT). This indicates that the classifiers are able to use the information conveyed by a variable not being defined for an event (which now is encoded in the value of exactly zero). The higher occurrences of `NaN` values for background events are probably a decent criterion which can be considered for classification. Thus the slight performance increase here is not unexpected. In any case, the performance differences are small, which shows that setting all `NaN` values to zero should not introduce any unexpected instabilities.

[15] Optimally a high multiplicity control channel (not available here) should also be checked to verify the occurrences of `NaN` values for physics data signal events.

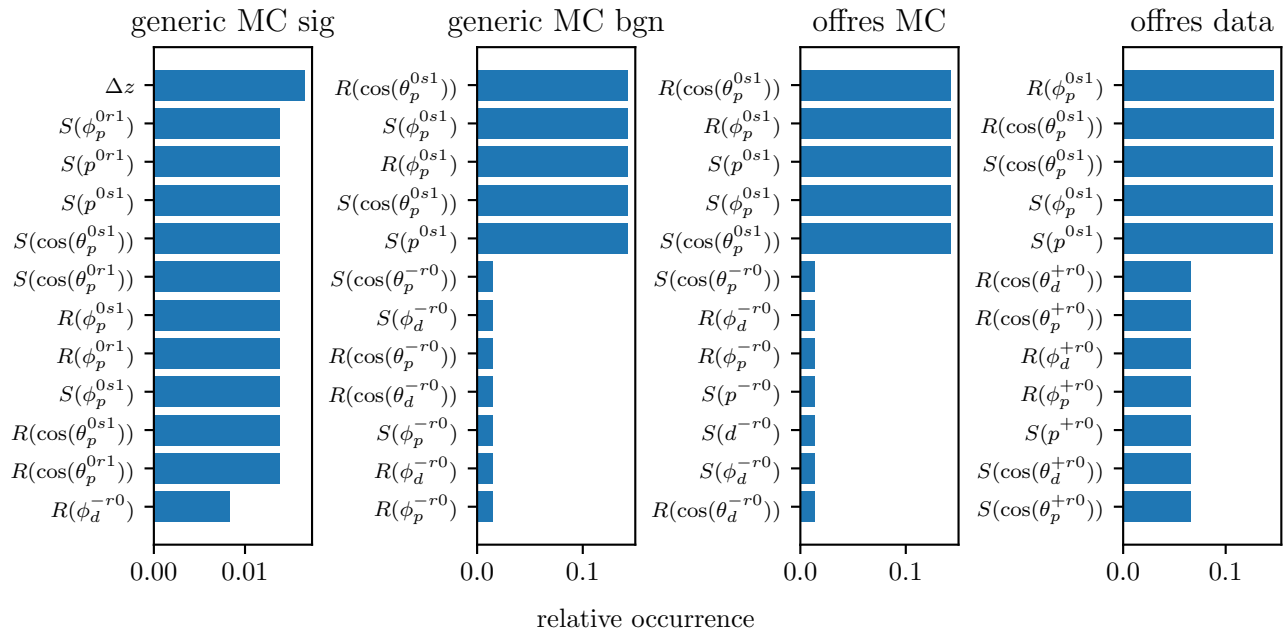


Figure 4.10.: Occurrences of `NaN` values in each variable (shown are the top 10) for generic MC signal/background and off-resonance MC/data. The occurrences are normalized to the total number of events in the respective samples.

4.4. Classifier Evaluation

4.4.1. Classification Performance

In the following the DisCoDNN and BDT are compared while also considering a DNN without decorrelation (generally labeled just *DNN*) as a reference.

Classifier Output Distributions First we consider the distribution of classifier outputs computed over the validation sample (with same number amount of signal and background events). Any anomalous shapes here may indicate incomplete or suboptimal training. The distributions are shown overlaid for DisCoDNN, DNN without DisCo, DisCoDNN with preliminary hyperparameters and BDT (separately for signal and background events) in fig. 4.11. The DNN with preliminary hyperparameters shall serve as an example for anomalies in the distribution. Those are most likely connected to the training not having converged properly as training here was stopped just around the epoch where sculpting starts (the classifier here is the same as for the ΔE evolution shown in fig. 4.7, bottom plot). All classifier output distributions but the one of the preliminary DNN follow the expected shape. They are strongly peaked near zero (one) for background (signal) events. No bumps in the middle of the distributions are observed as are present in the distribution produced by the preliminary DNN. For background events notably the DisCoDNN classifier output distribution is slightly more peaked for values near 1 than the distributions of BDT and DNN without DisCo. Signal events are however assigned very similar classifier outputs for all three classifiers. Only near zero the DNNs with and without DisCo appear slightly more inclined to misclassify some signal events. Overall thus the DisCoDNN is expected to perform similar to BDT and DNN without DisCo for signal extraction but worse for background rejection. This makes sense as the DisCo term specifically imposes the condition to only reject background events such that the background distribution of ΔE retains its shape. This extra constraint then may make it overall harder to reject some of the background events, resulting in slightly worse background rejection.

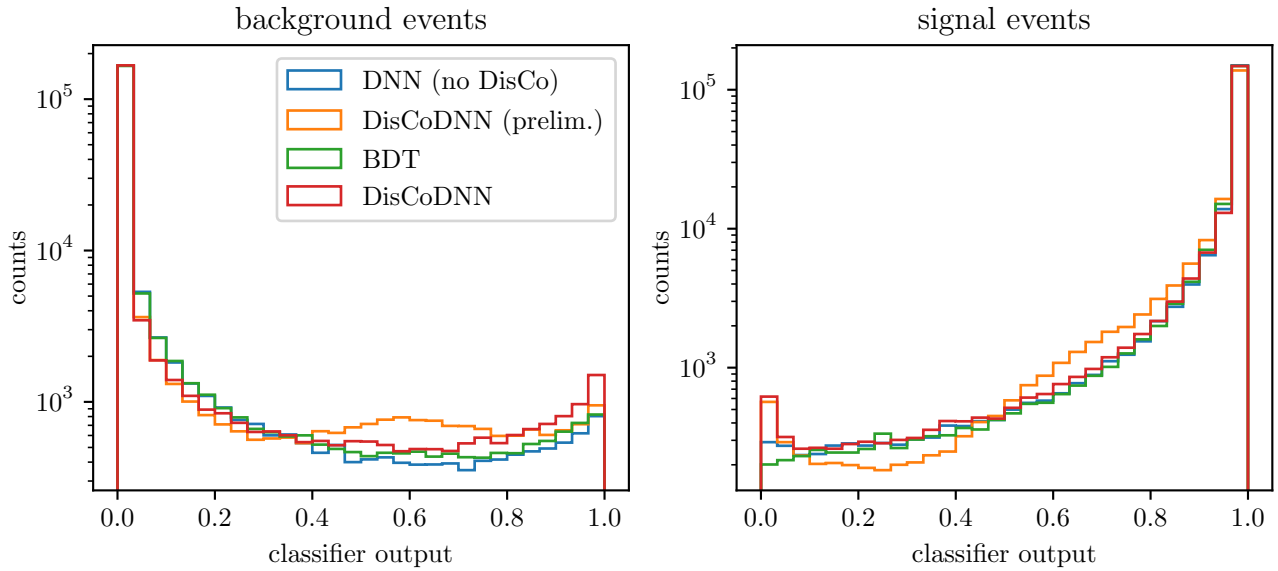


Figure 4.11.: Distributions of classifier output for DisCoDNN, DNN without DisCo, DisCoDNN with preliminary hyperparameters and BDT. The DisCoDNN with preliminary hyperparameters shall serve as an example of anomalies in the distribution. The remaining distributions approximately follow the expected shapes, indicating healthy classifiers.

ROC Curves When applying the continuum suppression there is the freedom to choose the continuum suppression cut depending on the desired amount of signal events to be retained or background events to be rejected. The signal efficiencies and background rejections for all the possible choices are visualized by the ROC curves shown in fig. 4.12. Overall when compared to the BDT, the DisCoDNN provides worse background rejection at the same signal efficiency. The DNN without disco can be seen to be of slightly better performance than the BDT. The performance gap can be attributed to the inferior background rejection capabilities of the DisCoDNN, as already indicated by the classifier output distributions.

Classifier Cut Positions For later comparisons of results using DisCoDNN and BDT, the continuum suppression cuts have to be chosen such that a fair comparison is possible. The condition imposed here will be equal fixed signal efficiency for the compared classifiers. Depending on the circumstances however a different condition may be preferred. The optimal choice here would be to choose the continuum suppression cut such that the uncertainties from a final signal yield fit (on MC) are minimal. This is however a non-trivial condition. As the concerned decay is very rare, it is generally desirable to not discard too many signal events when applying the continuum suppression in order to retain sufficient statistics for the signal. Thus as a compromise the condition of fixed signal efficiency is chosen. In the following discussion always 90% signal efficiency will be required.
While ROC curves visualize the performance for all possible continuum suppression cut positions, they do not directly indicate where to place the cut in order to reach a desired signal efficiency or background rejection. To compute the continuum suppression cut position for a given signal efficiency, the signal efficiency as a function of the cut position is sampled. This, together with the background rejection as a function of the cut position, is shown in fig. 4.13. The dashed lines indicate the required signal efficiency as well as corresponding cut positions and background rejections. The cut positions are determined by numerically inverting the sampled signal efficiency function. Comparing the curves for BDT and DisCoDNN it again becomes obvious that the DisCoDNN performs only slightly worse for signal extraction and the main contribution to the overall performance difference stems from worse

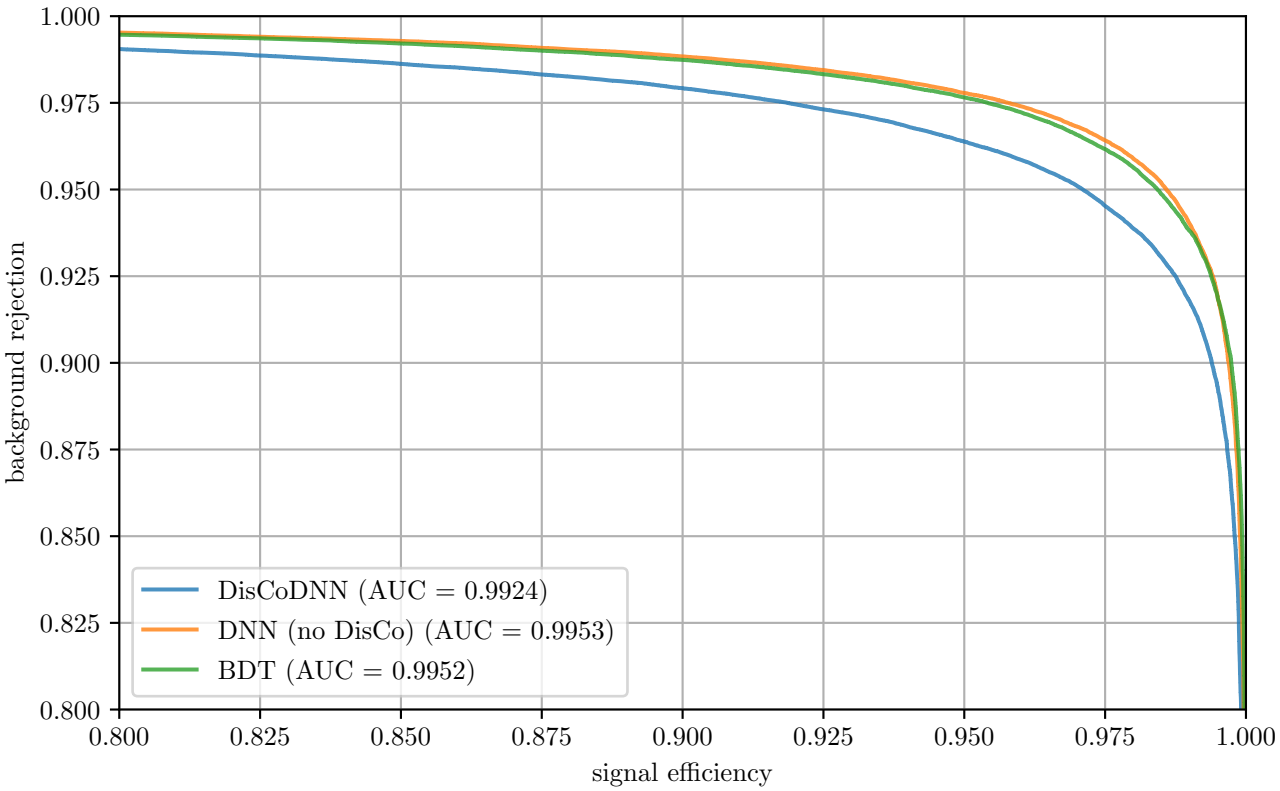


Figure 4.12.: ROC curves for DisCoDNN, DNN without DisCo and BDT. The DNN without DisCo barely outperforms the BDT whereas the DisCoDNN appears to perform noticeably worse.

background rejection.
The DNN without DisCo can be seen to perform slightly better for both signal efficiency and background rejection when compared to the BDT. The latter may be related to the extra events that the correlated DNN is able to discard by learning the correlation with ΔE.

4.4.2. Fit Variable Distributions After Continuum Suppression

To also assess suitability for signal yield fits, examples for the distributions of ΔE as well as the probability integral transform will be discussed here. The distributions of ΔE for the generic MC sample after continuum suppression with both the DisCoDNN and BDT are shown in fig. 4.14. As mentioned above, the continuum suppression cut was chosen such that 90% of signal events are retained. The background distribution for the DisCoDNN can be seen to follow the same shape as before continuum suppression (as shown in fig. 4.6, left plot), indicating effective decorrelation of classifier output and ΔE. For the BDT the distribution is sculpted, but can still be modeled well enough to be suitable for a fit. Comparing the two background distribution shapes, it appears that indeed the inferior background rejection of the DisCoDNN may at least partially be induced by the condition to keep the shape of the distribution in ΔE untouched. The BDT was able to reject many events that fall left (towards negative values) of the signal peak in ΔE. The DisCoDNN however cannot (if the decorrelation is effective) reject background events that fall only into a specific region in ΔE. Only ever background rejection over the whole spectrum of ΔE at once can be improved. Otherwise sculpting would be introduced. Comparing the number of remaining background events in the region under the signal peak, the BDT can be seen to only slightly excel in background suppression there. Thus there is no large

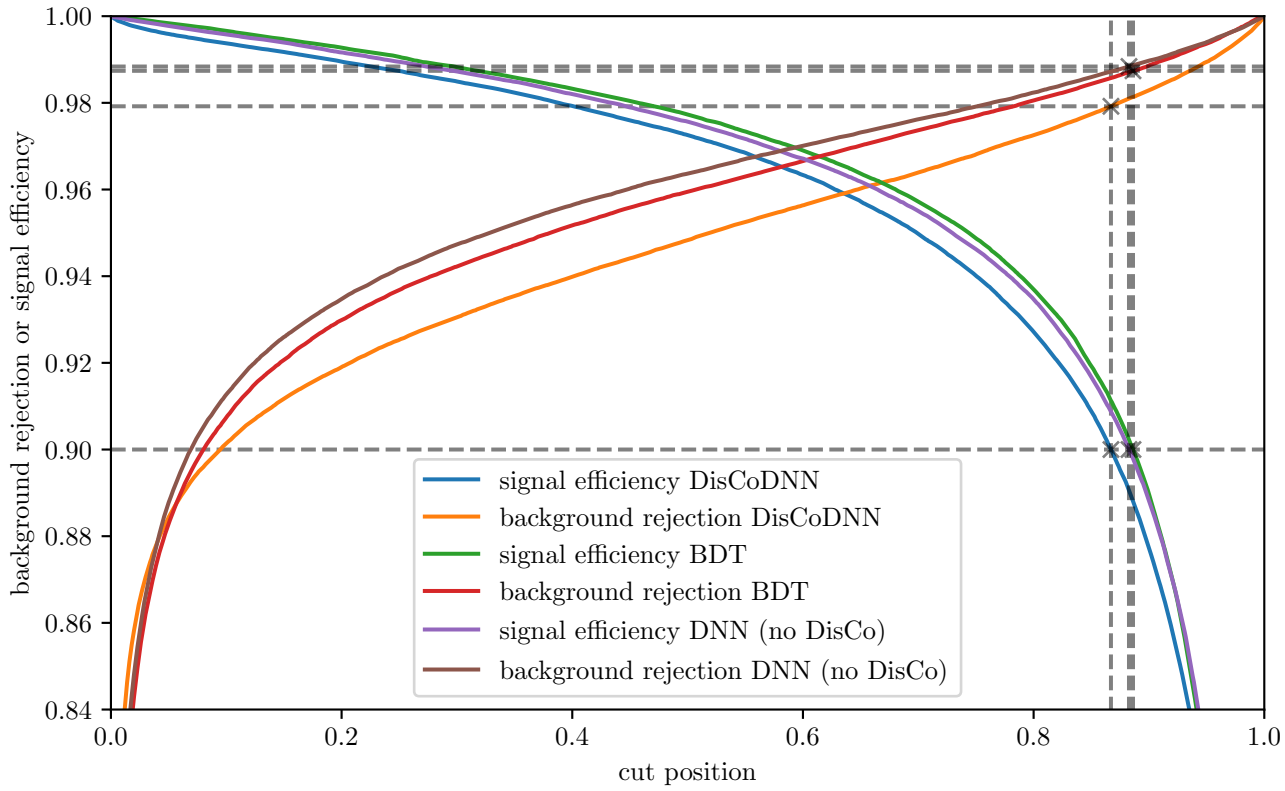


Figure 4.13.: Signal efficiency and background rejection as a function of the continuum suppression cut position. The performance gap between DisCoDNN and BDT/DNN without DisCo can be seen to be rooted mainly in inferior background rejection.

difference in classification capability between DisCoDNN and BDT for events that closely resemble signal events, at least in their corresponding value of ΔE. The DisCoDNN therefore appears to be limited in overall performance by its ability to suppress the hardest to suppress background events (i.e. those that the closest resemble signal events). If performance for those events is not improved, the decorrelation condition prohibits performance for also the technically easier to classify events to improve.
To further assess this, preferably a BDT with decorrelation measures should be prepared (not done here). This would give further insights on whether the DisCoDNN and BDT are of similar overall capability for the continuum suppression.
The distributions of the probability integral transformation (denoted μ here) are shown in fig. 4.15. The true signal distribution for the transform was taken to be that of signal MC. For all distributions the signal portion is reasonably flat, as expected. The logarithmic plot reveals the background part to take an approximately exponential shape. Next to the DisCoDNN and BDT we also consider the distribution for a DNN without decorrelation to show that μ is not affected even if ΔE is strongly sculpted. Further also technically a fit in only a single variable is possible. The accuracy of such a fit however was found to be much worse when compared to fits in two variables and is thus not further considered here.

4.4.3. Test Fits

To further judge the suitability of the distributions for signal yield fits, two simple fits have been set up. Note that the fits have not been particularly optimized and shall only serve as a reference to roughly

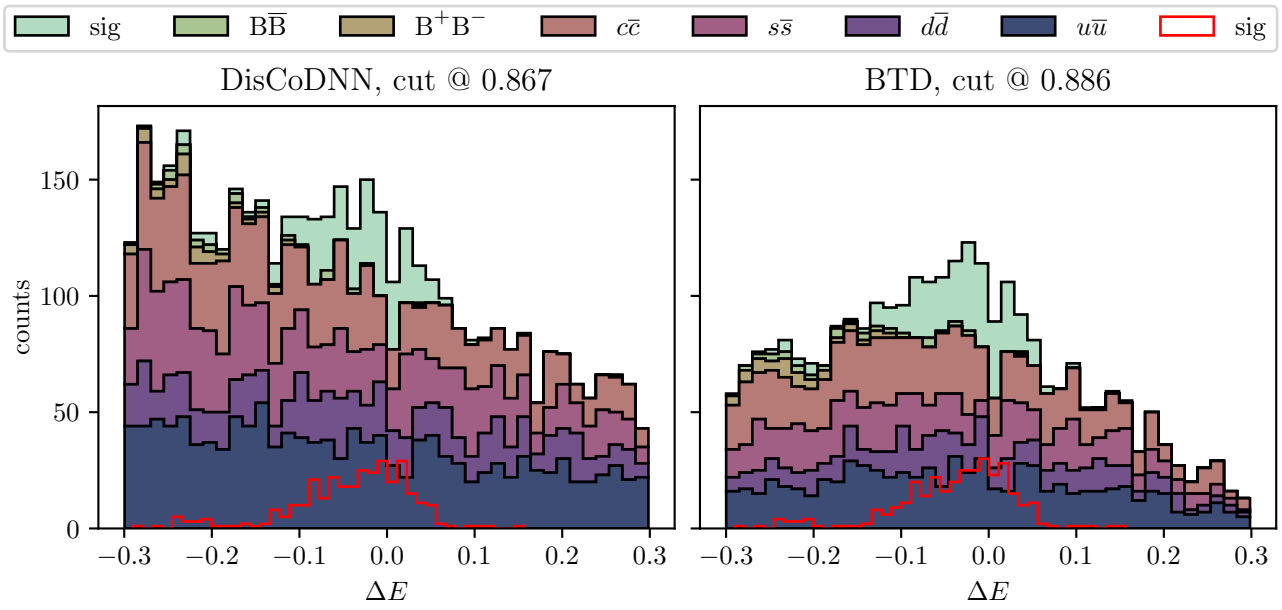


Figure 4.14.: Distribution of ΔE after continuum suppression cut for DisCoDNN and BDT. The different MC components are shown stacked. Sculpting of the background distribution can be seen to be successfully suppressed for the DisCoDNN. For the BDT sculpting occurs but appears inherently limited. The continuum suppression cuts are chosen to retain 90% of signal events.

assess if continuum suppression with either the DisCoDNN or BDT is obviously favored. As already indicated, the fits are done in two variables: ΔE and μ. The fits for continuum suppression using the DisCoDNN and BDT are shown in fig. 4.16 and fig. 4.17 respectively. The continuum suppression cuts are again chosen for 90% signal efficiency. Further the slightly tighter cut $|\Delta E| < 0.25$ is applied to remove some of the rounding of the ΔE distribution towards the edges which is believed to be introduced by the vertex fit during the reconstruction.

The background is split into $q\bar{q}$ (continuum) and $B\overline{B}$ background which are separately modeled. The signal peak in ΔE for both DisCoDNN and BDT is modeled as the sum of a Johnson's S_U-distribution and normal distribution. For μ the signal is by definition flat and thus always modeled by a uniform distribution. $q\bar{q}$ background in ΔE is modeled by a first order Chebyshev polynomial (essentially a straight line) for the DisCoDNN and a normal distribution for the BDT. In μ exponential distributions of the form $\exp[p_n(x)]$ where $p_n(x)$ is an nth order polynomial of the form $ax + bx^2 + cx^3 \ldots$ are used for both $q\bar{q}$ and $B\overline{B}$ background. For $q\bar{q}$, 4th and 3rd order polynomials were found to give decent agreement for DisCoDNN and BDT respectively. For $B\overline{B}$ 2nd order polynomials are used. Finally $B\overline{B}$ in ΔE is also modeled by an exponential with 2nd order polynomial in the exponent.

The shapes of the above named distributions are fixed by a fitting them to the generic MC validation sample for $q\bar{q}$ background and to the signal MC sample for the signal shape. For the final fit only the yields are left floating. The resulting yields for both fits as well as the true yields (known for the MC sample) are listed in table 4.4. Uncertainties are determined by the "Hesse" method as provided by the used minimizer (`minuit`). Almost all of the resulting pulls for the yields can be seen to fall within one sigma.

The uncertainties on the fit results differ slightly. When calculated relative to the true yields, the fit for continuum suppression with the BDT shows a slight advantage of around 0.76% smaller uncertainty. Despite the smaller errors on the fitted yields, for the BDT the difference of the fitted and true yield relative to the true yield is around 2.8% larger than for the BDT. As the uncertainties here are however

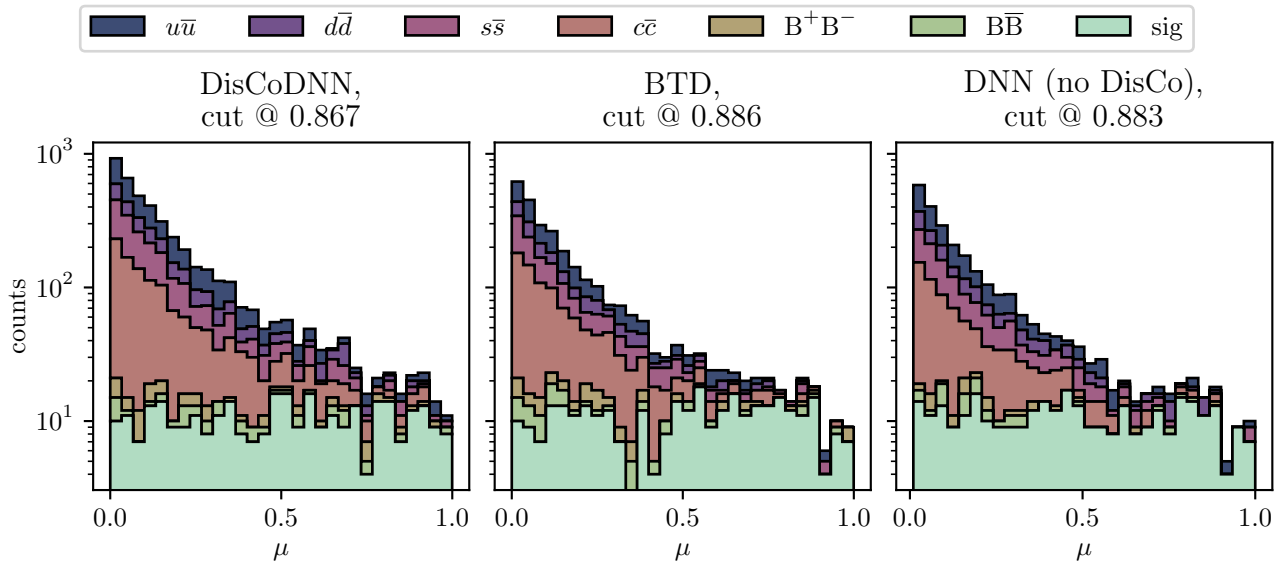


Figure 4.15.: Distribution of the probability integral transform (μ) for continuum suppression with the DNN with DisCo BDT and DNN without DisCo. The signal portion in all cases is reasonably flat, as expected. Continuum suppression cuts are chosen to retain 90% of signal events.

on the order of 8%, this may be considered a fluctuation.

Thus evidently for continuum suppression with the DisCoDNN the extracted signal yield is slightly less precise than for continuum suppression with the BDT. Overall however precision is on the same order of magnitude making it difficult to rule out the DNNs immediately without further optimization of the fits. One factor that may influence the signal yield precision is the chosen continuum suppression cut, of which here only a single choice was used.

While the BDT performs slightly better for the fits attempted here, both BDT and DisCoDNN may be of interest in their own right depending on the details of an analysis.

	signal	$q\bar{q}$	$B\overline{B}$
true yield DisCoDNN	318	3313	71
true yield BDT	321	2134	75
yield DisCoDNN	310.6 ± 28.3	3343 ± 39	49.30 ± 31.28
yield BDT	337.5 ± 26.1	2149 ± 35	43.52 ± 27.83
rel. fit error DisCoDNN in %	8.902	1.178	44.06
rel. fit error BDT in %	8.144	1.626	37.1
rel. true error DisCoDNN in %	2.335 ± 8.902	0.897 ± 1.178	30.57 ± 44.06
rel. true error BDT in %	5.133 ± 8.144	0.710 ± 1.626	41.97 ± 37.10
pull DisCoDNN in σ	-0.2623	0.7619	-0.6937
pull BDT in σ	0.6302	0.4367	-1.131

Table 4.4.: Results from test fits shown in fig. 4.16 and fig. 4.17. Note that the true MC yields are slighly different due to numerical error introduced by the procedure to determine the cut position.

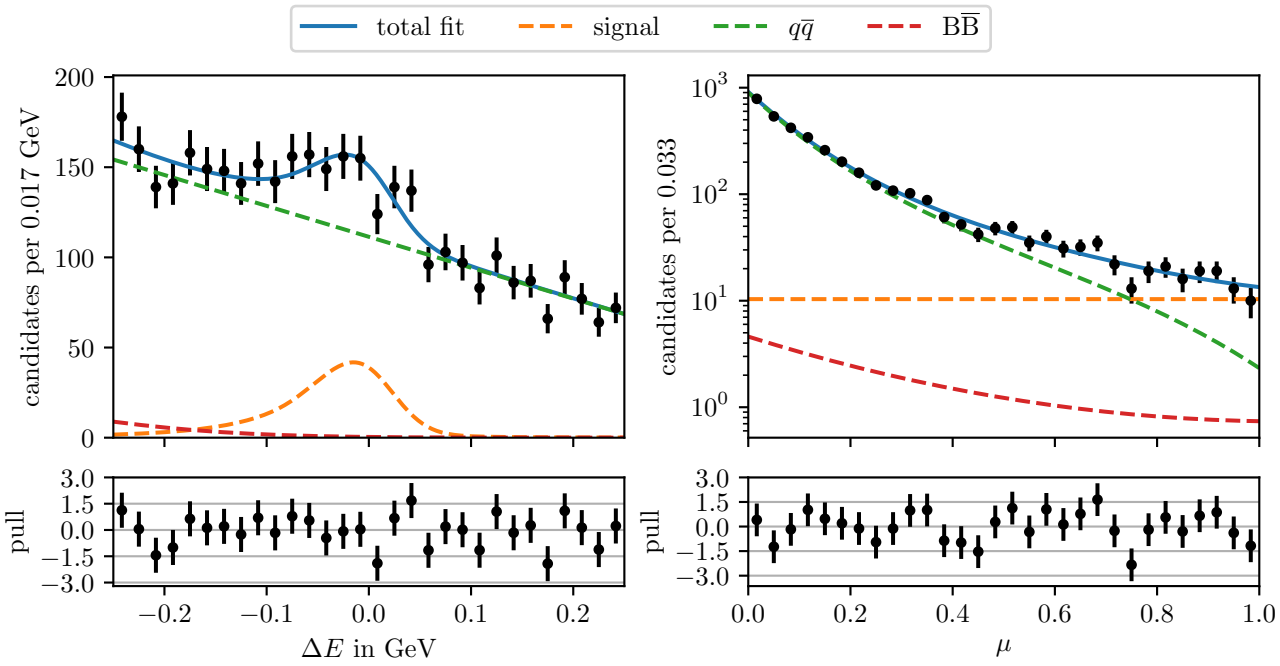


Figure 4.16.: Test signal yield fit in ΔE and μ (probability integral transform) after continuum suppression with the DisCoDNN. The resulting yields are shown in table 4.4.

4.4.4. Classifier Stability

To assess the stability of the classifiers using our new continuum suppression variables, two studies of the stability of the classifier output under under fluctuations of the input data were conducted. The difficulty here is to obtain a large number of data samples to evaluate the classifiers performances on. As the available MC data[16] is only a single sample and generating more is very expensive, one has to resort to methods to generate data samples mimicking the available sample. The two different attempts made are described in the sections below.

Bootstrap Method

The bootstrap method is a procedure to select subsets of events from a given sample to generate further samples [2]. The procedure is the following: The number of events for a sample is sampled from a Poisson distribution for which the mean is set to the number of events in the original sample. The obtained number of events is then drawn from the original sample, where the same event can be drawn multiple times. This procedure is equivalent to generating samples under the assumption that there is a known finite set of possible events.
The obvious flaw is that each event stays exactly the same and the fluctuations of the numerical values of the variables corresponding to an event are not captured. The classifiers however should be generalized to an extend where these fluctuations have negligible effect. This is also partially addressed with the uncorrelated toy samples which have been generated and will be discussed in the next section. The more important fluctuation modeled here is the one of the number of events of a given class of events included in the sample. Even if not clearly defined, intuitively one expects certain classes of a given abstract property to exist. An example may be a class of events that resemble signal events very closely and are thus hard to classify correctly. Depending on how many events of such a class

[16] The studies here could technically also be done on physics data, which here however is not permitted as the analyses are done blind.

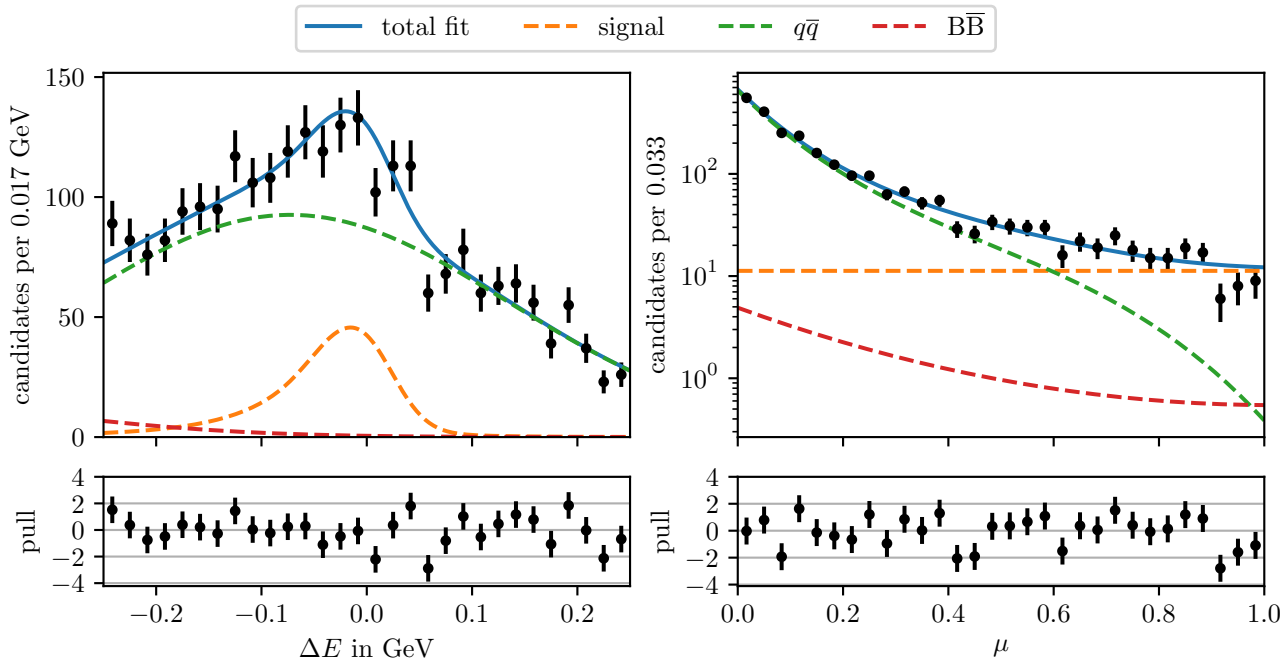


Figure 4.17.: Test signal yield fit in ΔE and μ (probability integral transform) after continuum suppression with the BDT. The resulting yields are shown in table 4.4.

are present in a sample, the classifier may perform slightly better or worse. Evaluating the classifier performance on many samples generated by the bootstrap method then shows to which degree the classifier performance remains stable under the modeled fluctuations.

2000 samples have been generated from the generic MC sample by the bootstrap method for DisCoDNN, BDT and DNN without DisCo. For each generated sample and classifier the AUC score was computed and recorded. The generic MC sample was chosen (as opposed to the sample with equal amount of background and signal events) in order to obtain a representation of classifier performance in a realistic setting. Realistic meaning the correct proportions of signal and background events. The distributions of the AUC scores for all generated samples for both DisCoDNN and BDT are shown in fig. 4.18. The distributions are of reasonable spread, indicating that the classifiers remain stable under the modeled fluctuations. Further shown in the same plots is a fit to the distribution using an asymmetric Gaussian. The parameters obtained from the fit are also shown where σ_l and σ_r are the left and right side standard deviations respectively. Asymmetry of the distribution is expected when a classifier is trained close to its theoretical performance limit. If most of the samples already result in scores near the performance limit, it becomes increasingly unlikely for the fluctuations to induce a performance score higher than the mean. The asymmetry is slightly more pronounced for the BDT and DNN without DisCo than for the DisCoDNN. This is probably related to the inherently worse performance of the DisCoDNN due to the imposed decorrelation. Overall the distributions for the BDT and DNN without DisCo are also slightly less spread.

The means of the distributions are overall close to the AUC scores obtained on the validation sample as shown with the ROC curves in fig. 4.12. This is expected as the samples have all been generated from the same generic MC sample and thus for the most part contain the same background events as have been used for the ROC curves.

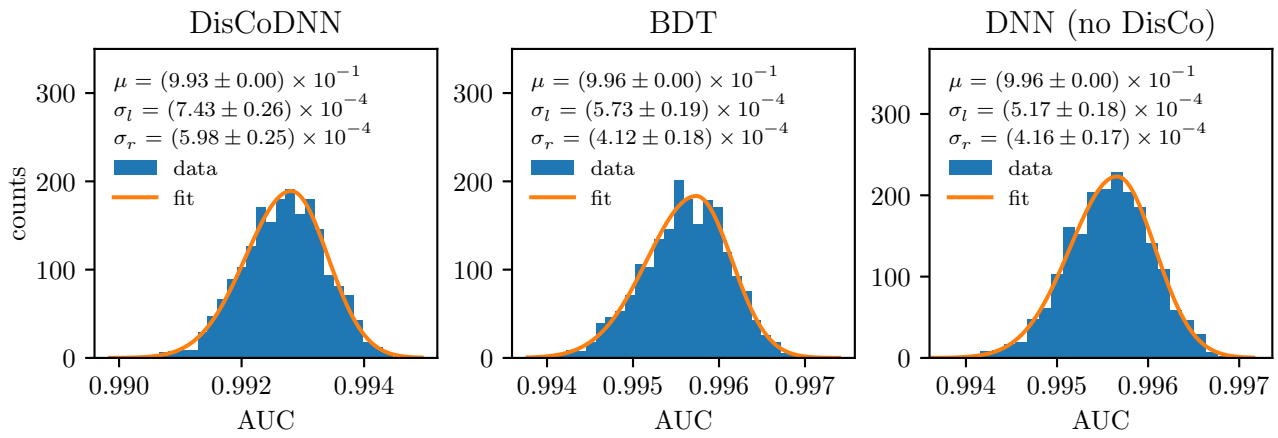


Figure 4.18.: Distribution of the AUC scores from 2000 samples generated from generic MC by the bootstrap method. The distributions are fit with an asymmetric Gaussian also shown in the plots.

Uncorrelated Toys

To generate a number of toy samples, events were generated by sampling the classifier input variables from their corresponding distributions. The distributions were taken for background from the generic MC validation sample and for signal from the dedicated signal MC sample. The number of signal and background events is taken to be the same as in the generic MC sample. All of the variables were sampled completely independently, meaning that *no correlations are modeled*.
It was attempted to model the correlations by sampling from higher dimensional histograms for groups of significantly intercorrelated variables. However those groups were found to contain too many members to fill a histogram of the same dimension. Even when correlations with Pearson correlation coefficient smaller than 0.3 were neglected, groups of up to 8 intercorrelated variables were found. While in any case there is insufficient data to fill an 8 dimensional histogram, it also requires unrealistic amounts of memory. Thus correlations were chosen to be disregarded completely. This however happens to give an interesting perspective on their importance for classification performance.
Correlations are known to be significant for the chosen set of input variables. Most notably they also differ significantly for signal and background events. The correlations between the chosen variables, as well as the difference in correlations between signal and background events are illustrated in fig. 4.19. Below the diagonal the Pearson correlations coefficients computed over the whole sample (here the union of training, test and validation sample as introduced in section 4.3.1) are shown. Above the diagonal the magnitude of the difference of the Pearson correlation coefficients for only signal and only background events is shown[17]. This illustrates that there are significant differences in correlations between signal and background events, which is information the classifiers can utilize. Thus, if evaluated on samples without any modeled correlations, the classifiers are expected to perform worse, given they properly utilize the correlations.
As for the bootstrap method, again 2000 samples were generated. The distributions of the corresponding AUC scores for DisCoDNN, BDT and DNN without DisCo are shown in fig. 4.20. Again, the distributions are well shaped and of reasonable spread, indicating overall stability of the classifiers under the modeled fluctuations. Opposed to the bootstrap study, here all events are now actually unique in terms of the exact numerical values of the corresponding variables. Compared to the distribution from the bootstrap method, the means are obviously lower and the spread is higher. The

[17] One should keep in mind that the Pearson correlation coefficients shown only capture linear correlations, the classifiers though are expected to be able to also learn more general correlations.

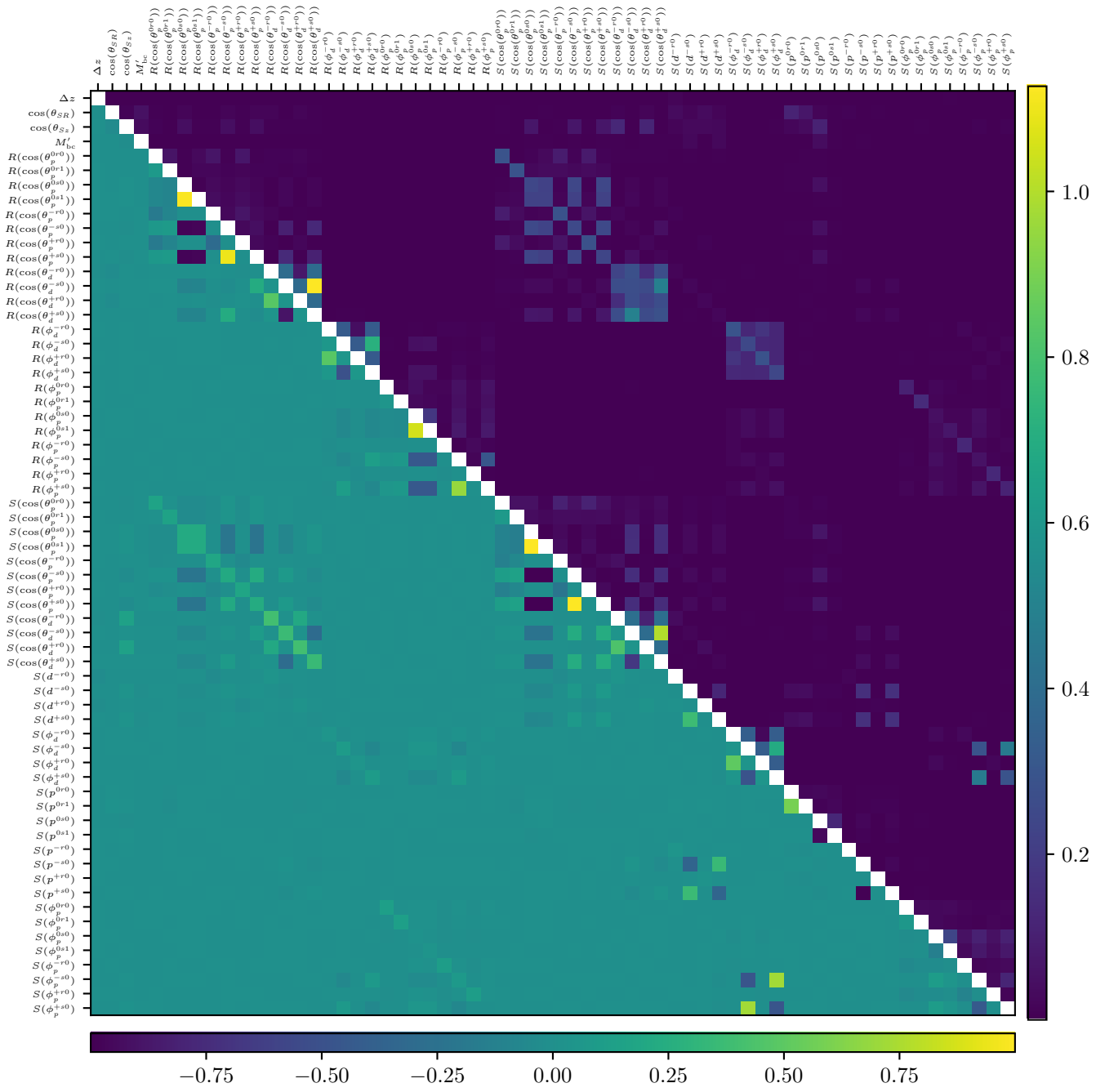


Figure 4.19.: Pearson correlation coefficients for the variables used. Below the diagonal the correlations for only background events are shown. Above the diagonal the absolute values of the differences between the correlation coefficients for only signal and only background are shown. This highlights that correlations between the variables are indeed different for signal and background events.

Note: For unknown reason some PDF viewers display this figure blurred. It is not supposed to be blurred.

distributions remain slightly skewed. The distribution for DisCoDNN is further skewed than with the bootstrap samples while for the BDT the asymmetry remains similar. This does not fit in with the interpretation of classifiers near the performance limit from above and is not entirely understood. Notably the mean AUC for the DisCoDNN dropped only very slightly by 0.004, while for the BDT the difference is 0.014. For the DNN without DisCo the performance drop is the largest with a difference of 0.067. Further the spread of the distribution also increased the most for the DNN without DisCo when compared to the other two classifiers. Thus the DNN without DisCo can be seen to most heavily rely on the correlations. Interestingly however the DisCoDNN appears to rely on correlations less than the BDT. Considering the heavy reliance of the DNN without DisCo on correlations, presumably the decorrelation keeps the DisCoDNN from learning too much of the correlations. This may also be a contributing factor to the overall slightly worse performance of the DisCoDNN when compared to the BDT.

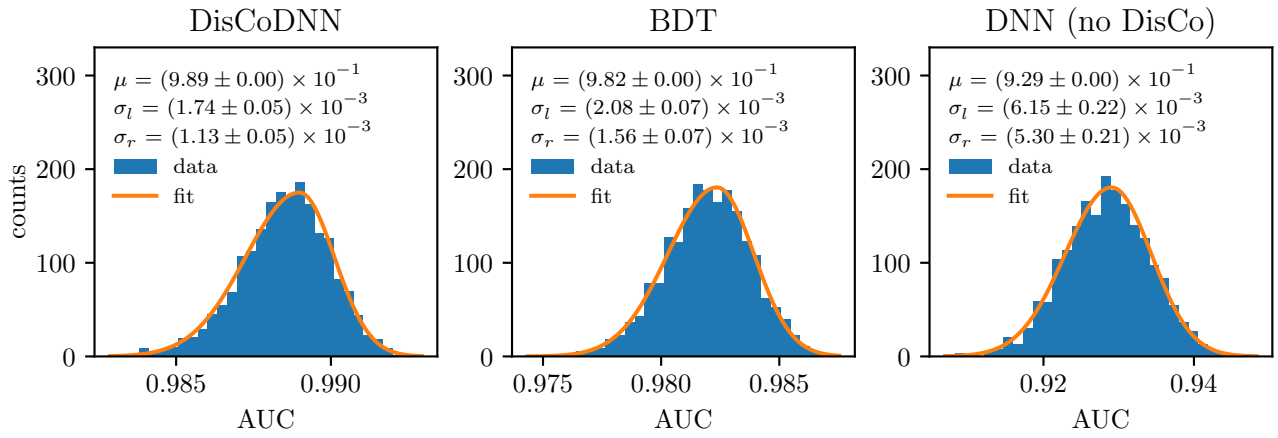


Figure 4.20.: Distribution of the AUC scores from 2000 samples generated from generic MC by sampling from the distributions of the input variables. This neglects any correlations between the input variables, resulting in inferior classifier performance. The distributions are fit with an asymmetric Gaussian also shown in the plots.

4.4.5. Classifier Generalizability

We apply the classifiers trained on the signal channel to the topologically similar control channel. This will give insight on the extend to which the learned characteristics of the signal channel are generally applicable for continuum suppression. Studied are the DisCoDNN, BDT and DNN without DisCo. The resulting classifier output distributions are shown in fig. 4.21. Interestingly all classifiers tend to misclassify fewer background events for the control channel. Thus apparently for the control channel, background events are easier to identify. The DNN without DisCo behaves slightly different as its output distribution shows overall worse performance when compared to the other two classifiers applied to the control channel. As here the correlations with ΔE are not suppressed and thus used for the classification, this indicates that the exact details of those correlations are specific to the signal channel.
While showing expected behaviour for background events, the classifiers are essentially useless for classification of signal events of the control channel. The classifier output distribution now is almost symmetric. This is not entirely unexpected, as the background events for both channels are expected to be similar, signal events however not necessarily are. Despite of the control channel being chosen to resemble the topology of the signal channel, the decays appear different enough for all of the classifiers to almost completely fail to identify any signal. This highlights that the chosen continuum suppression

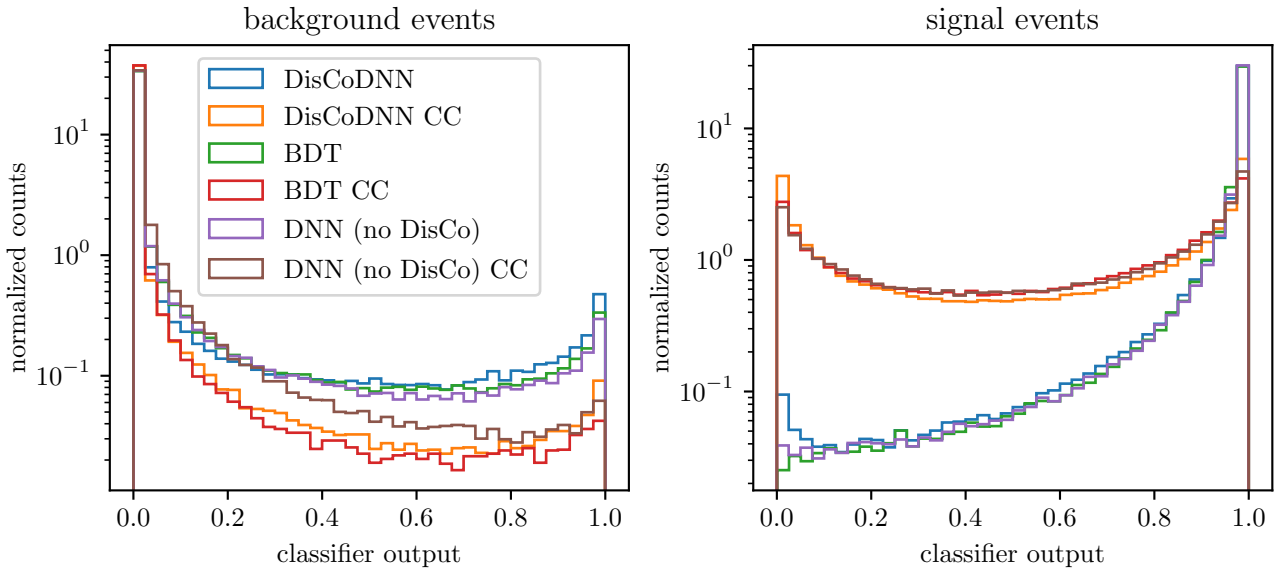


Figure 4.21.: Classifier outputs for DisCoDNN, BDT and DNN without DisCo applied to the signal and control channel. While similar performance can be archived for background rejection, all classifiers almost completely fail to identify signal events for the control channel.

variables are indeed very specific to a given decay. The overall performance difference is also evident from the ROC curves for the classifiers on the control channel sample. They are shown, together with the ROC curves for the signal channel sample for reference, in fig. 4.22. Differences in the input variable distributions between control channel and signal channel are also directly visible from the corresponding distributions. Plots of all the distributions are shown in appendix A.2.1 for the signal channel and appendix A.2.2 for the control channel.
To better interpret the above observations, we consider the distributions of ΔE after continuum suppression for the control channel. The continuum suppression cuts are again chosen for 90% signal efficiency. As the signal efficiency of the classifiers is very poor, the cuts must be chosen extremely loose. This can best be seen from the signal efficiency and background rejection as a function of the cut position as shown in fig. 4.23. Here again the background rejection can be seen to be overall better than for the signal channel. To understand the cause for this further studies of the related differences between signal and control channel are required. Interestingly the pattern for signal efficiency of similar performance of BDT and DNN without disco but noticeably worse performance of the DisCoDNN can be observed also for the control channel. For the background rejections however the DNN without DisCo now performs much worse than the other two classifiers. The reason for this, which is the same as for the differences in the classifier output distributions discussed above, is related to the sculpting in ΔE. The distributions of ΔE for the control channel after the continuum suppression with the determined cuts are shown in fig. 4.24. The DNN without DisCo can be seen to fail spectacularly. This indicates that much of the performance of this DNN indeed stems from the introduced sculpting. However, the distribution now is sculpted into a peak that is no longer centered around zero. Apparently the method by which the DNN reconstructs ΔE from the input variables introduces a bias in such a way that the distribution (as reconstructed by the DNN) ends up slightly shifted towards positive values for the control channel. For DisCoDNN and BDT the continuum suppression however is, despite the very loose cuts, surprisingly effective. Compared to the DisCoDNN the BDT obviously suppresses background better at the same signal efficiency, indicating better generalizability of the BDT. The observed performance differences between DisCoDNN and BDT already observed for the signal channel appear overall amplified for the control channel. Notably

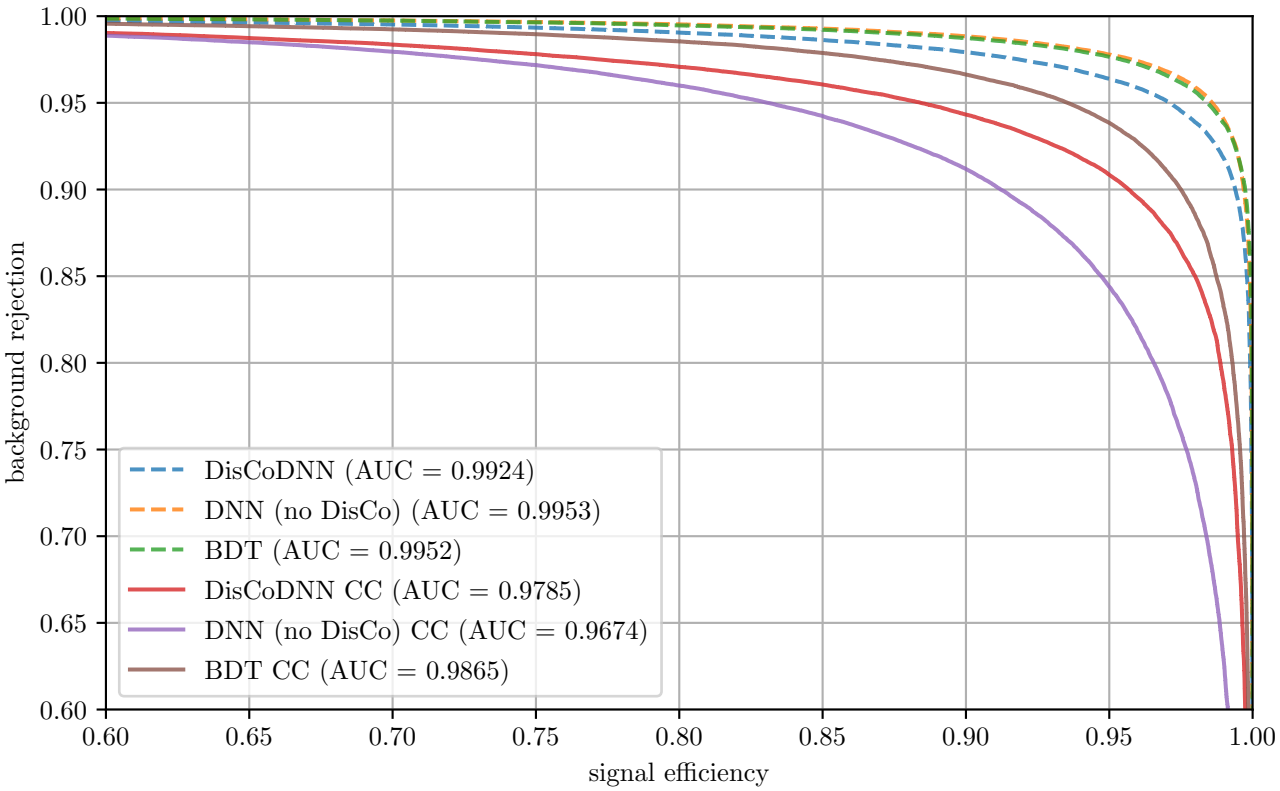


Figure 4.22.: ROC curves for the studied classifiers applied to control channel MC. The ROC curves for the same classifiers applied to the signal channel data are shown for reference (dashed lines). Evidently all classifiers perform much worse when applied to control channel MC where depending on the classifier the performance difference however differs significantly.

now also a large B^+B^- background appears, which is believed to be a trait of the control channel. For completeness the distributions in μ after continuum suppression are also shown in fig. 4.25.
We conclude that the variables used exhibit distributions which for signal events are indeed very specific to a given decay as the classifiers generalize poorly for signal identification. While for the considered control channel even with very loose cuts, the continuum suppression was found to be surprisingly effective, it is hard to judge if this is generally the case. Preferably further control channels would have to be considered.

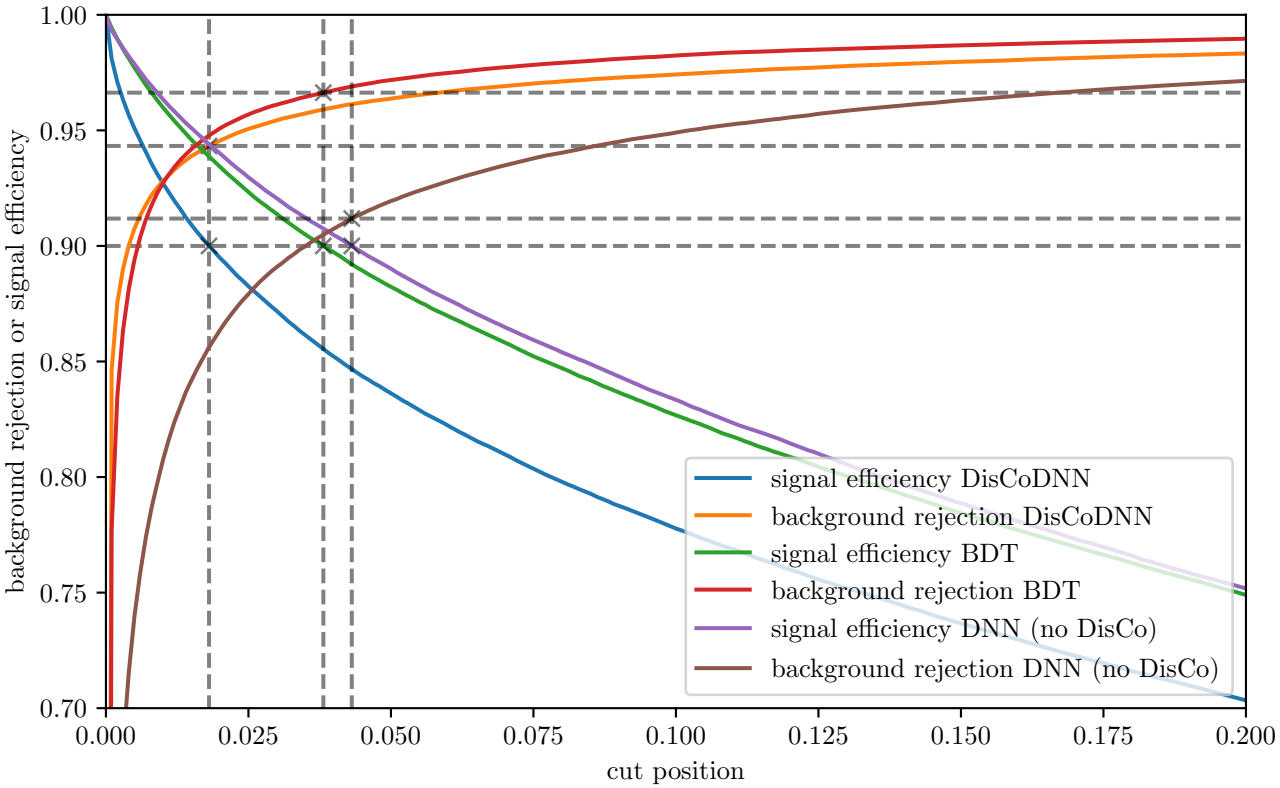


Figure 4.23.: Signal efficiency and background rejection as a function of the continuum suppression cut position for the classifiers trained on the signal channel but applied to the control channel.

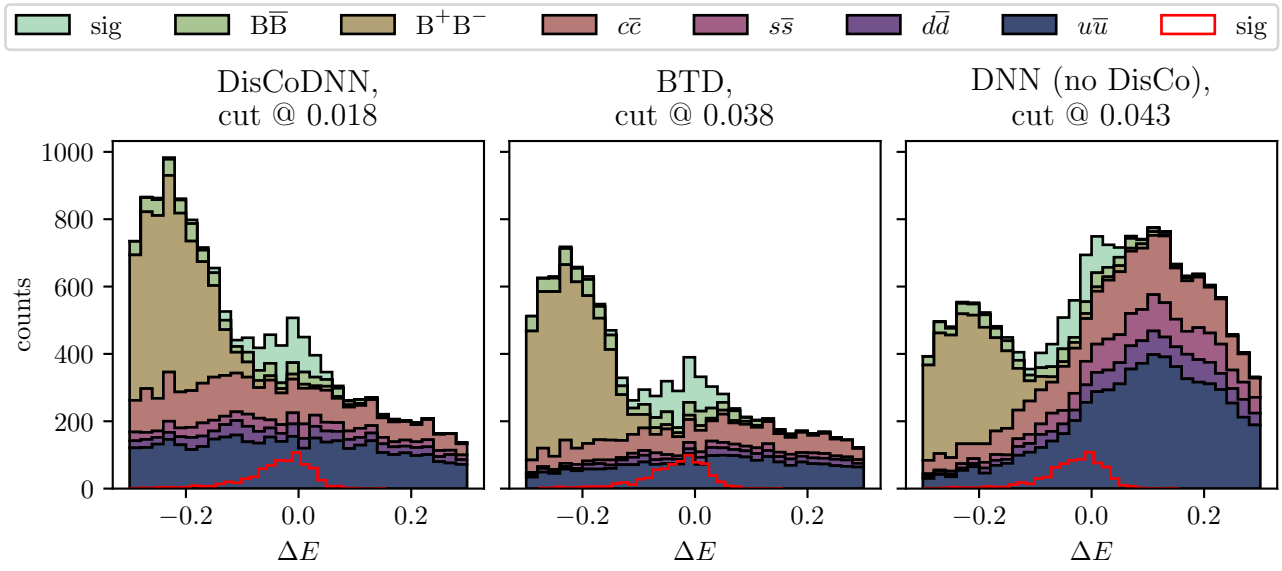


Figure 4.24.: Distribution of ΔE after continuum suppression with the classifiers trained on the signal channel but applied to the control channel. Again the cut positions are chosen to result in 90% signal efficiency.

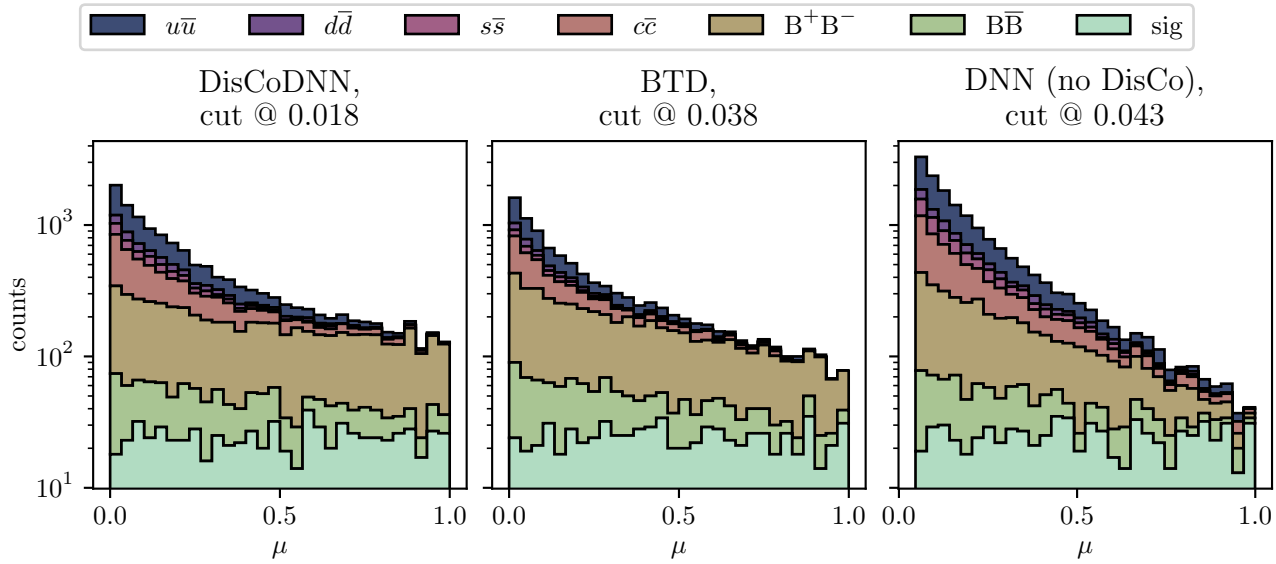


Figure 4.25.: Distribution of the probability integral transform (μ) after continuum suppression with the classifiers trained on the signal channel but applied to the control channel. Again the cut positions are chosen to result in 90% signal efficiency.

5. Conclusion and Outlook

This thesis introduced a novel set of low-level continuum suppression variables and demonstrated their applicability for continuum suppression in $B^0 \to K^0_S(\pi^+\pi^-)\pi^0(\gamma\gamma)$. As the first step for this the MC modeling of the introduced variables was verified. Decent agreement, given the known problems with the data samples available for this thesis, was found for background events by considering off-resonance, sideband as well as control channel data. MC modeling for signal events could not be directly verified as the considered control channel is of rather low multiplicity. In any case, verification of MC modeling for signal events is expected to be problematic as it was found that the chosen continuum suppression variables for signal events are highly dependent on a given decay mode. This raises the question to which extend conclusions drawn for a control channel decay can be extended to the signal channel considered here. In a future study one or multiple control channels allowing for verification of MC modeling of the used variables for signal events should be considered.
For the continuum suppression two classifiers, a BDT and a DNN, were prepared, with the initial motivation being that the DNN might be more capable then a BDT when it comes to extraction of information from the used low level variables. The training of the DNN however was significantly complicated as large correlations between the classifier output and ΔE had to be avoided in order to enable the final goal of a signal yield fit. The BDT interestingly appeared (to some extend) inherently immune to this problem. To accomplish training of the DNN, trainings with adversarial networks as well as well as the DisCo decorrelation method were implemented. The tuning of hyperparameters for those training methods turned out to be difficult, partially because systematic approaches are hard to apply. A configuration resulting in stable decorrelation was eventually reached with only the DisCo method, resulting in a decorrelated DNN, here called the DisCoDNN. Tuning of the training with adversarial networks is expected to be even more difficult and was thus deemed beyond the scale of this thesis. Nevertheless we demonstrated that effective decorrelation, even though very laborious, is possible.
The prepared DisCoDNN as well as the BDT were applied to MC samples to evaluate and compare their performances. The BDT was found to achieve overall better background rejection at a given signal efficiency. Assessment of the distribution of ΔE however showed that the performance difference may be at least be partially traceable to the decorrelation, which was only applied for the DNN. To further assess this, a BDT with similar decorrelation measures should be prepared for future studies. From the results presented in this thesis we conclude that neither the BDT nor the DisCoDNN are the obviously favored method for continuum suppression with the goal of a signal yield fit. Thus both options may be considered for an analysis. The overall trend however seems to be that the DisCoDNN is limited in performance due to the decorrelation measure and is unlikely to outperform the BDT with the current setup, even if further tuning is applied.
Stability of the classifiers using the newly introduced continuum suppression variables was addressed through studies using a large number of samples generated from the available MC sample. This showed that both classifiers remain reasonably stable under the modeled fluctuations. As a side effect of this study the dependence of the classifiers on the correlations between the input variables could be assessed. This showed that compared to the BDT, the DisCoDNN appears to only very slightly rely on the correlations. Whether in case of the DisCoDNN this is an effect of the applied decorrelation could be further studied by repeating the studies for a BDT with applied decorrelation.
Finally to study the generalizability of the trained classifiers, they were applied to the control channel $B^0 \to \overline{D}^0(K^+\pi^-)\pi^0(\gamma\gamma)$. This showed that both DNN and BDT almost entirely fail to identify signal events, which highlights the issue of generalizability of the used continuum suppression variables. To

still achieve sufficient signal efficiency, cuts had to be chosen very loosely, which however resulted in still surprisingly good continuum suppression. Whether this is specific to the considered control channel remains to be investigated. The lack of generalizability is expected to complicate for example the estimation of systematic uncertainties through the use of control channels. Thus this issue is likely to have to be further addressed before sensible application of the proposed continuum suppression for an analysis is possible.
The initial motivation for application of the DNNs was that they may excel when it comes to utilizing the more subtile features hidden in the data. In the current state, any possible gain appears thwarted by the additional constraints that come with the decorrelation methods, which must be applied to make the DNNs usable for continuum suppression in the first place. Judging from the results presented in this thesis, a significant gain in accuracy of measurements of the concerned branching ratios or CP asymmetries *with the current setup for decorrelation* is not expected. Whether the decorrelation can be further improved to reduce the impact on classification performance should however be investigated. Possible things to cover in future research would be the following:

- Studies to determine which variables influence the sculpting the most (to possibly exclude them).
- Tuning of an adversarial network based decorrelation to see if performance is impacted similarly to what was observed for the DisCo decorrelation method.
- Application of DisCo decorrelation to a BDT to verify whether the observed performance drop is directly connected to the decorrelation.
- Investigation of further neural network architectures and their effect on the correlation induced sculpting.

A. Appendix

A.1. Known Issues with Data Samples Used

$\tau^-\tau^+$ Background The data samples prepared for this thesis were eventually found to not have the usual skim for analyses of decays of B mesons applied, even though they were supposed to. Applying it manually is only partially possible, as not all variables required for the cuts are included in the available data samples. The only way to apply the skim correctly would be to re-process all of the data. As the above described problem was discovered fairly late in the research for this thesis, re-processing was not feasible. Thus in the used physics data samples a portion of what is believed to be $\tau^-\tau^+$ background[1] is contained. It can however not be verified if this contribution really can be attributed to $\tau^-\tau^+$. Further investigation would require to either re-process while applying the correct skims or study of the effects of $\tau^-\tau^+$ using the corresponding MC samples, which however were not available (in time for this thesis). Despite of it not being entirely clear if the observed background component can be assigned to $\tau^-\tau^+$, below it will be referred to as $\tau^-\tau^+$ for simplicity.
Approximating the skim by rejecting events with a low number of matched tracks (as also done as part of the skim), was considered but turned out to be problematic. For the skim events with less than three matched tracks are rejected, but even choosing four was not sufficient to resolve even some of the largest disagreements between MC and data. Choosing more than four already rejects a significant portion of events that are clearly not part of the $\tau^-\tau^+$ background.
Two of the available variables were found to separate the $\tau^-\tau^+$ contributions rather well: the ratio of the second to the zeroth Fox-Wolfram Moment (denoted R_2) and the angle between the thrust frame of the signal side and the z-axis (denoted $\cos(\theta_{Sz})$). Both of those are peaked strongly near 1, which is unexpected. The distribution for the named variables for off-resonance data are shown in fig. A.1 Further the distribution of one continuum suppression variable (introduced in detail in section 4.2) where significant deviations were observed is shown. Eventually $\cos(\theta_{Sz})$ was chosen for an additional cut to reject the $\tau^-\tau^+$ contribution. We require $\cos(\theta_{Sz}) < 0.923$. The contributions that are assigned to the $\tau^-\tau^+$ and thus discarded are highlighted in fig. A.1. This also shows that the anomalous regions are caused by the same group of events, which justifies the compromise of choosing a cut on $\cos(\theta_{Sz})$ to reject them. While here only examples are shown, anomalies observed in the distributions of many other used variables were simultaneously resolved by placing the cut. The above mentioned cut will be applied consistently to all samples. We note that similar anomalies were observed for the sideband[2] and in smaller extend for the control channel $\mathrm{B}^0 \to \overline{\mathrm{D}}^0(\mathrm{K}^+\pi^-)\pi^0(\gamma\gamma)$. The exact reason as to why anomalies were less pronounced for the control channel is unclear.

Missing Momentum and Energy Corrections It was further discovered that for the available physics data corrections on the track momenta and photon energies were not applied, even though they should have been. The track momentum correction is known to be small, the photon energy correction however may be significant. Correctly applying those corrections also requires a re-processing of the data. Possible MC modeling disagreements introduced by this will be discussed if encountered.
Further it is believed that this may be related to distorted reconstruction efficiencies on physics data, which also have been found for the available samples. When compared to the MC samples, the physics

[1] Anomalous parts in distributions of the used variables were found to be correlated with low numbers of tracks, as expected for τ decays.
[2] Sideband means signal channel samples with the M'_{bc} replaced by $5.2\,\mathrm{GeV} < M'_{\mathrm{bc}} < 5.27\,\mathrm{GeV}$ to exclude signal events.

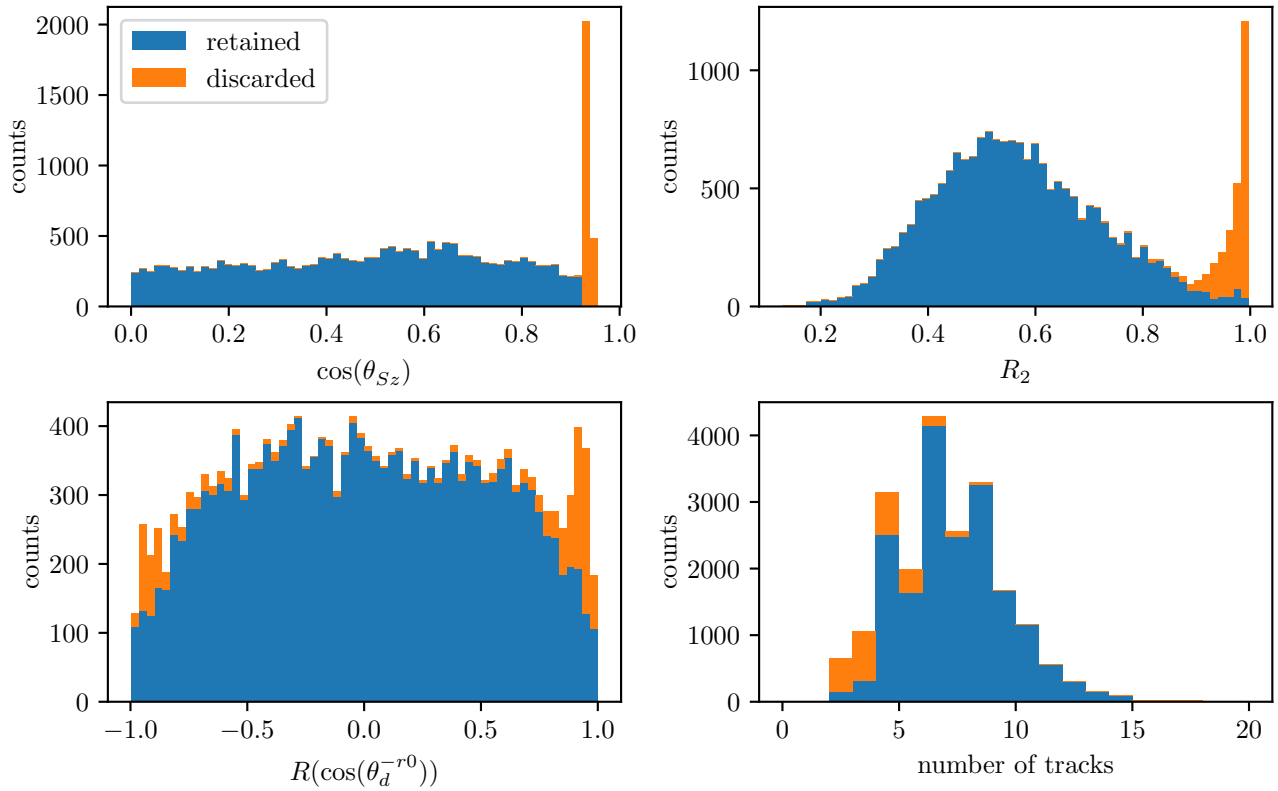
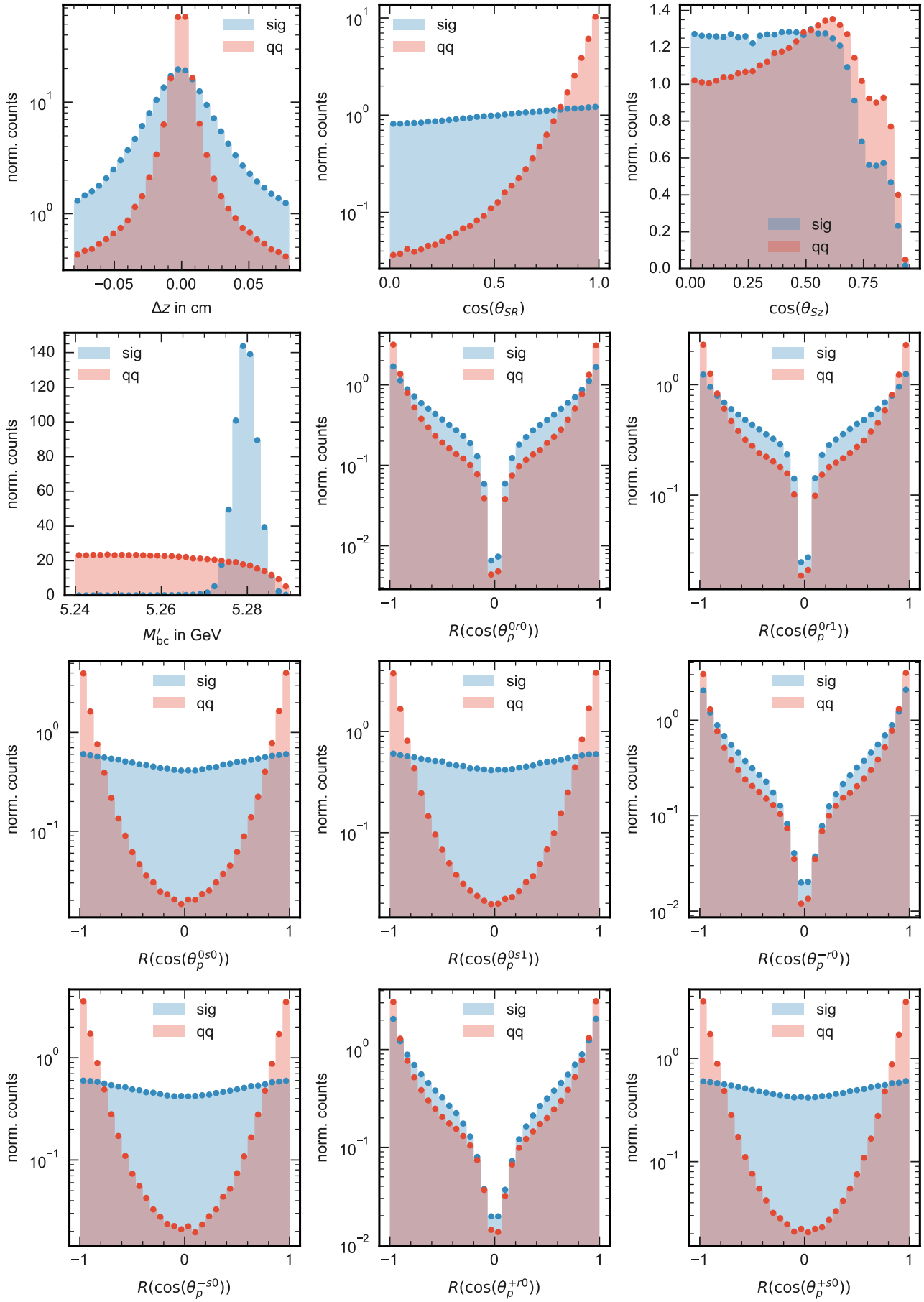


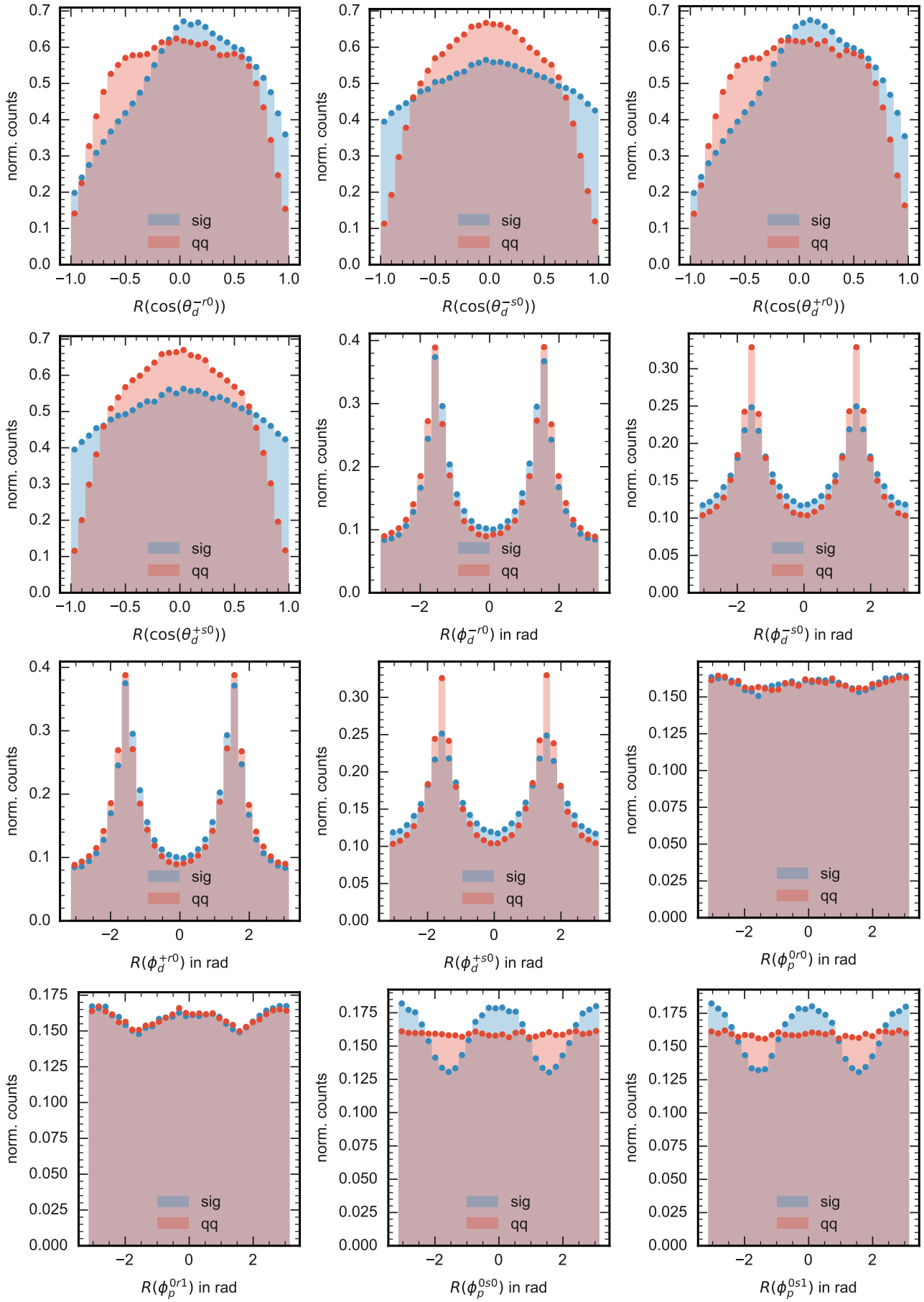
Figure A.1.: Visualizations of the portion of events that are removed from off-resonance data by applying the additional cut $\cos(\theta_{Sz}) < 0.923$. The cut at the same time removes large anomalies in $\cos(\theta_{Sz})$ itself, R_2, as well as some of the continuum suppression variables (introduced in section 4.2) of which one example ($R(\cos(\theta_d^{-r0}))$) is shown. Further the number of tracks in an event for the removed events can seen to be generally low, indicating the anomalous parts to be possibly caused by $\tau^-\tau^+$ contributions.

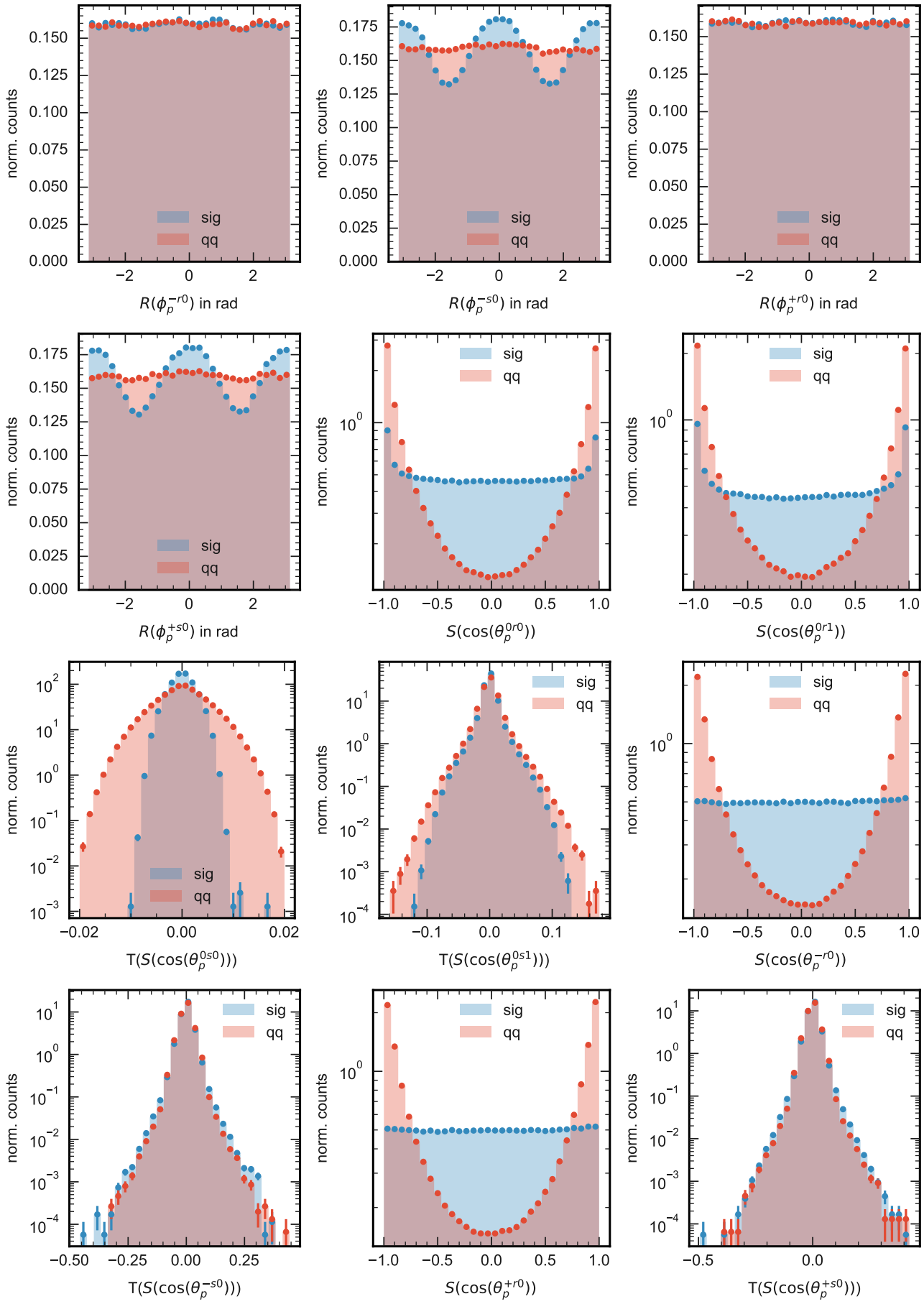
data samples are found to contain fewer events than expected from the integrated luminosities (which are known for each sample). If computed relative to the $1\,\mathrm{ab}^{-1}$ generic MC sample, the data sample is found to correspond to only $294\,\mathrm{fb}^{-1}$, where as $362\,\mathrm{fb}^{-1}$ would have been expected. Therefore either the generic MC sample contains too many or the physics data too few reconstructed events. The latter may be explained in relation to the not applied energy corrections: The distributions of the variables used for cuts in the reconstruction have been found to have some discrepancies between data and MC which are such that usually in the tails, which are discarded by the cuts, relative to the integrated luminosity more events are contained for physics data than for MC. Thus applying the cuts may have the effect of removing relatively more events for physics data than for MC. The result is a too low reconstruction efficiency for the physics data. The too low efficiencies have been observed for both the control channel and signal channel. The mentioned discrepancies in the distributions were observed for off-resonance, sideband as well as the control channel. An example for the discussed above is M'_{bc} for which the distribution for the sideband is shown in fig. 4.3 (upper left). Here the shift of the distribution towards lower energies for physics data may be one manifestation of the described above. Considering all of the above, any studies involving physics data should definitely be repeated with the corrected samples before any final conclusions are drawn. Further the reconstruction efficiencies should be checked once the corrections are applied, as there is no guarantee that the explanation given here is correct or accounts for the whole of the difference. As the problems outlined here only affect physics data, the training of classifiers as well as their evaluation on MC data are not directly affected.

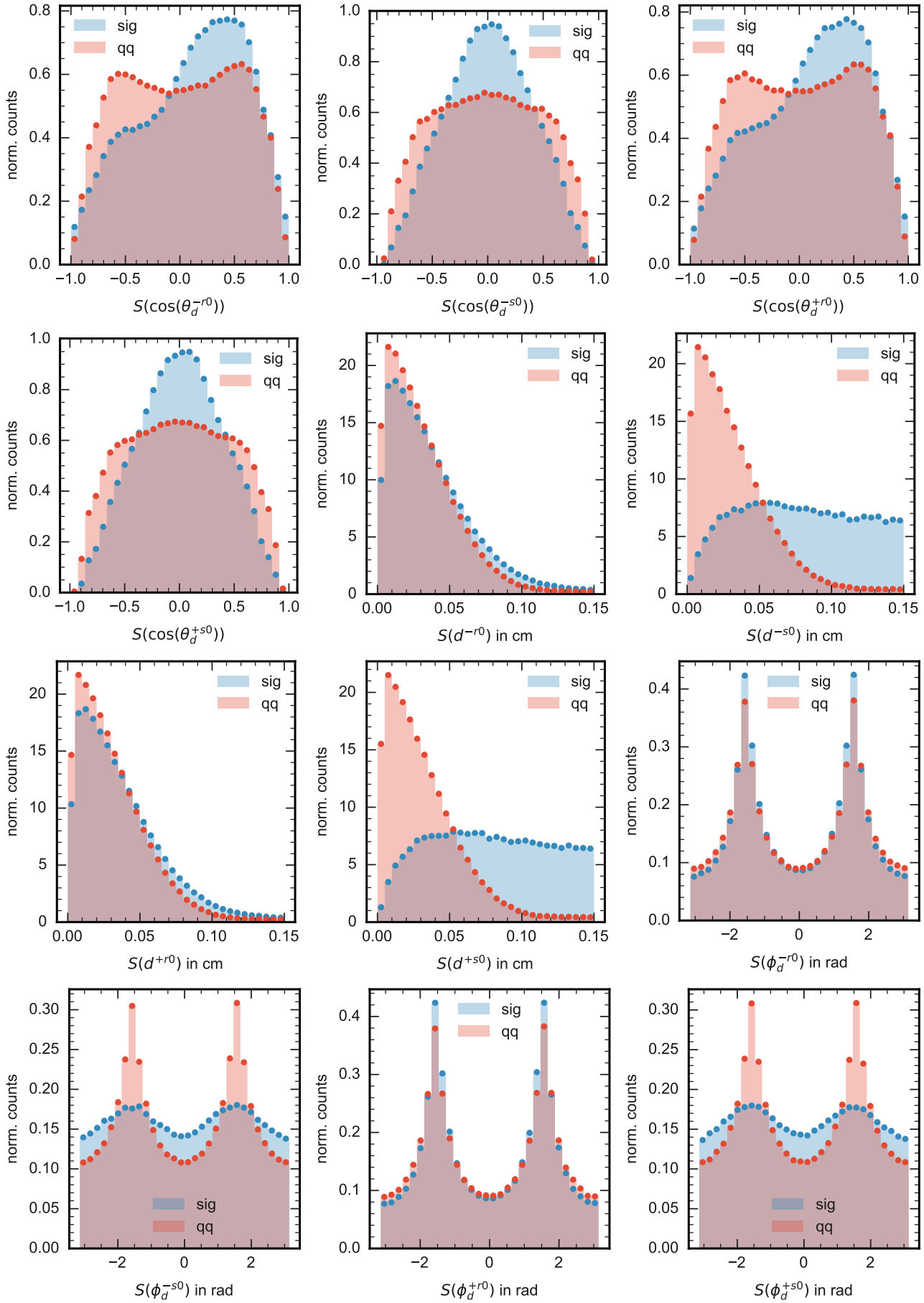
A.2. MC Signal vs Background Plots

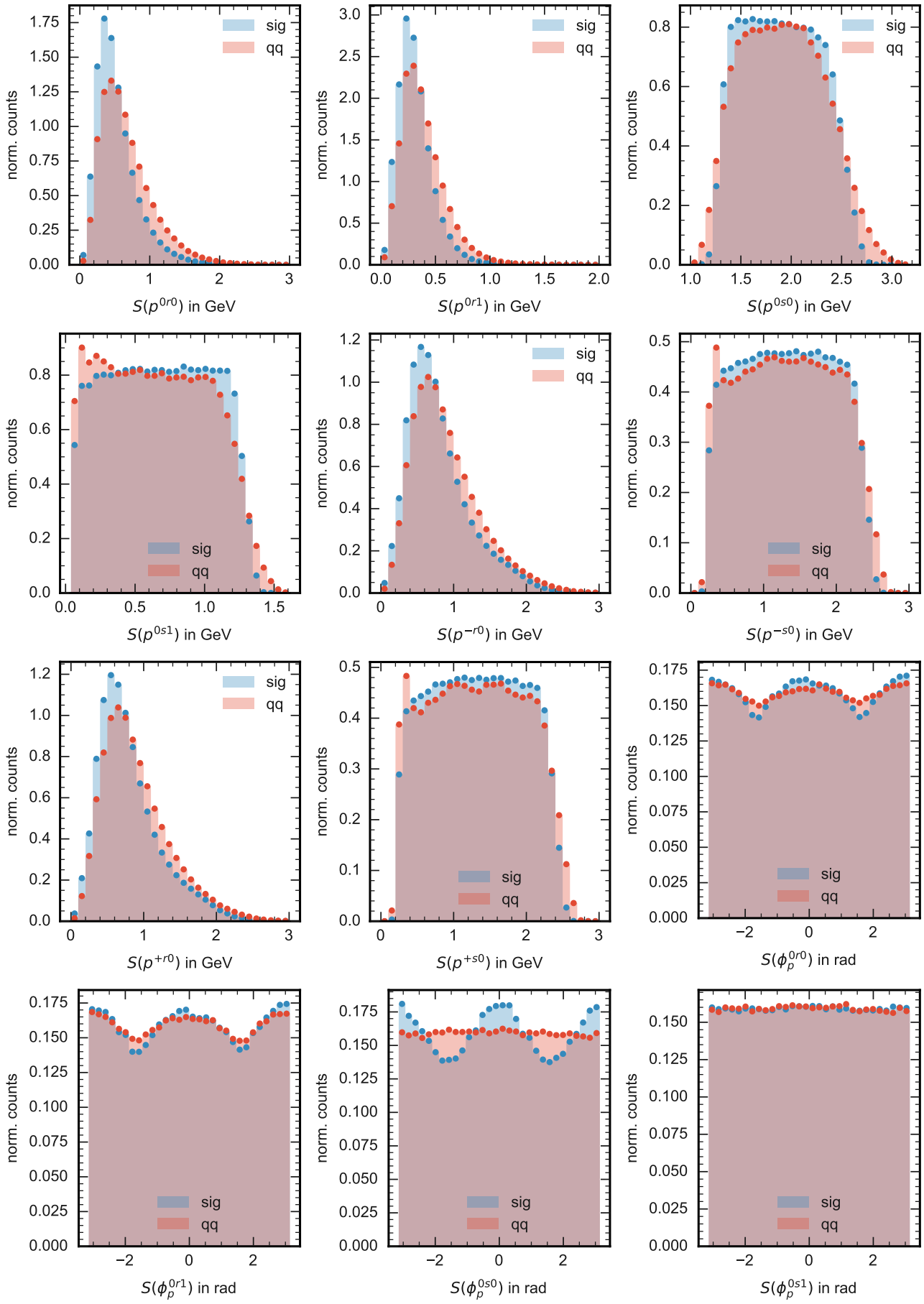
A.2.1. Signal Channel

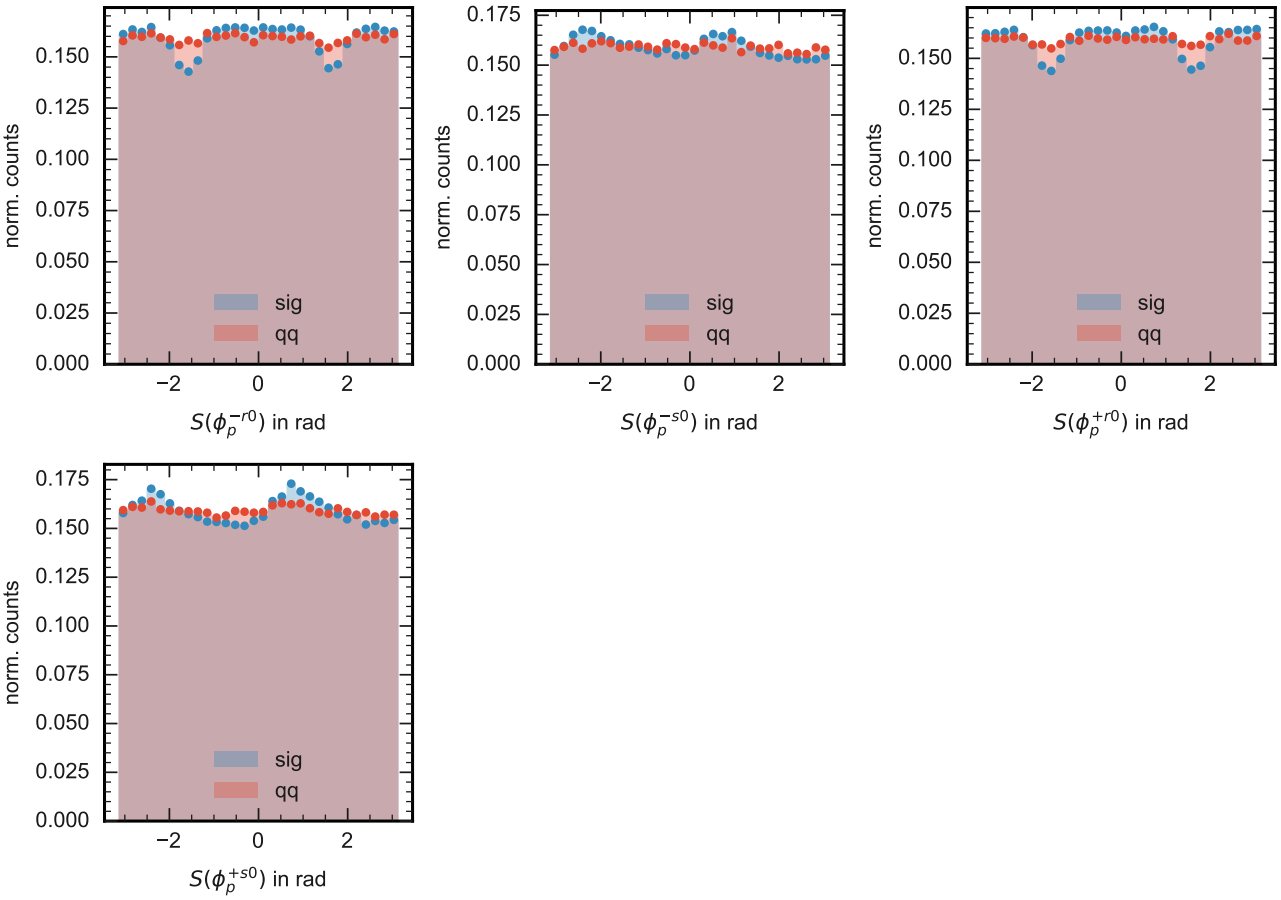



norm. counts
sig
qq
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad

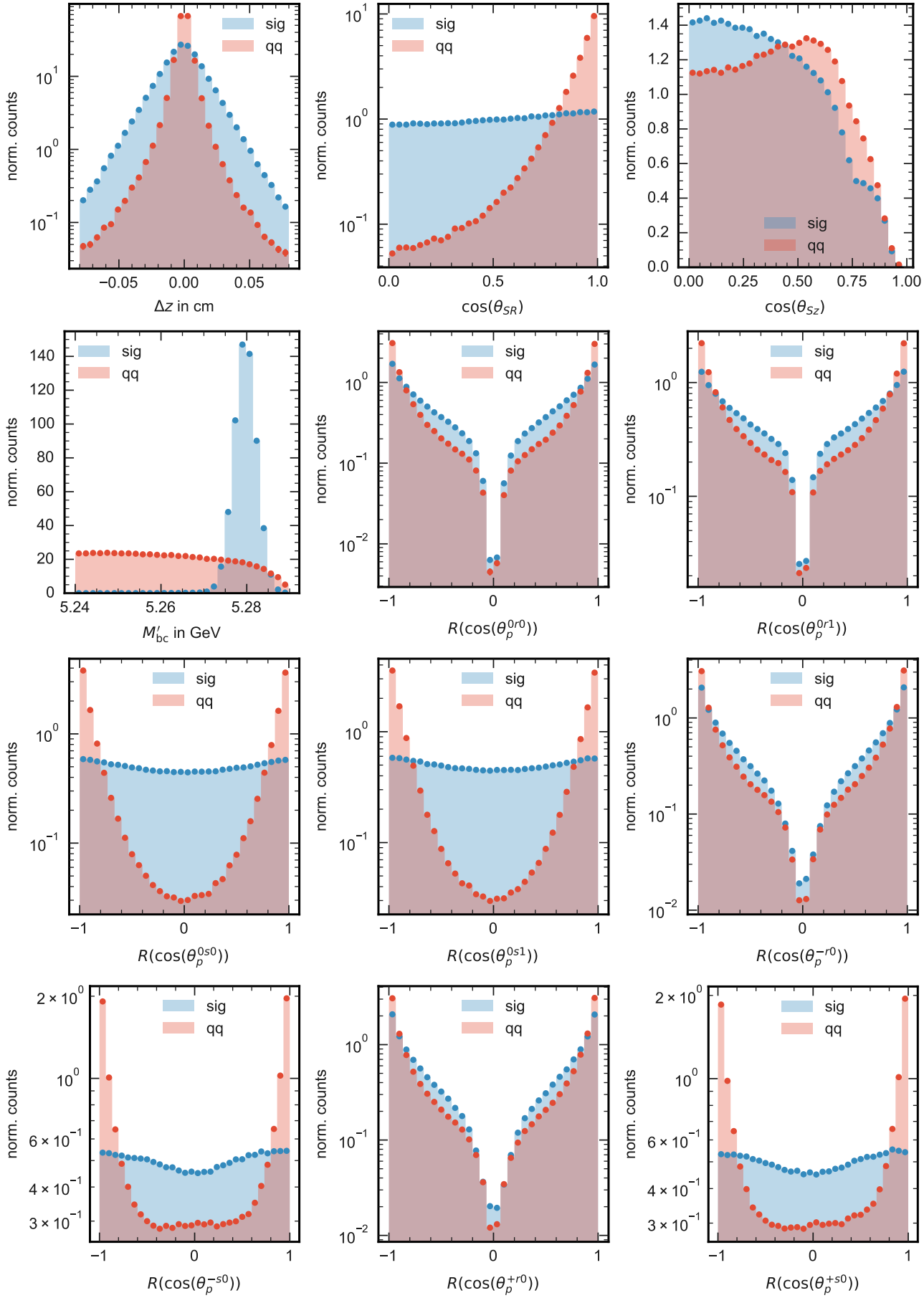

norm. counts
sig
qq
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
$S(\cos(\theta_p^{0r0}))$
$S(\cos(\theta_p^{0r1}))$
$\mathrm{T}(S(\cos(\theta_p^{0s0})))$
$\mathrm{T}(S(\cos(\theta_p^{0s1})))$
$S(\cos(\theta_p^{-r0}))$
$\mathrm{T}(S(\cos(\theta_p^{-s0})))$
$S(\cos(\theta_p^{+r0}))$
$\mathrm{T}(S(\cos(\theta_p^{+s0})))$

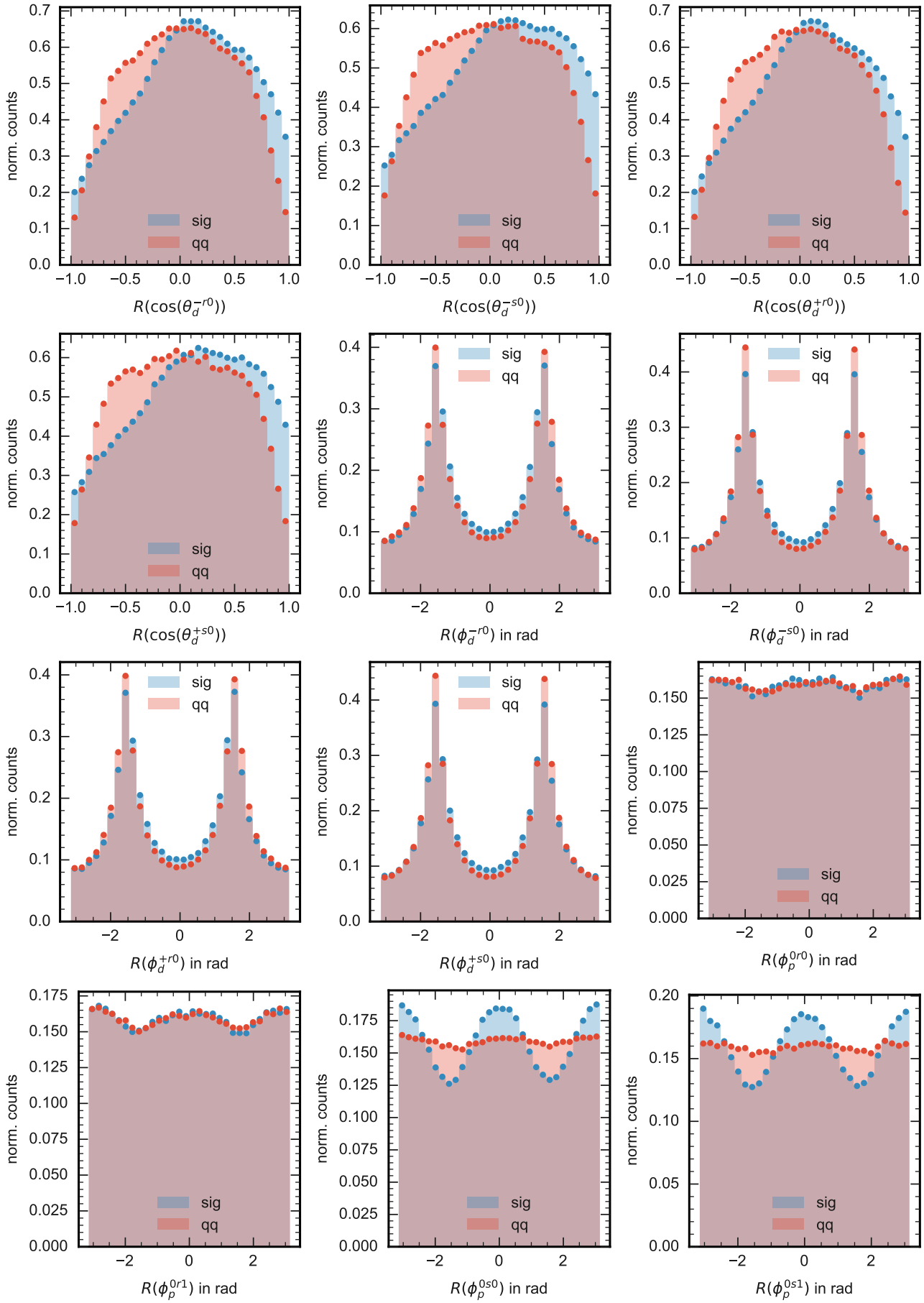

sig
qq
norm. counts
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad

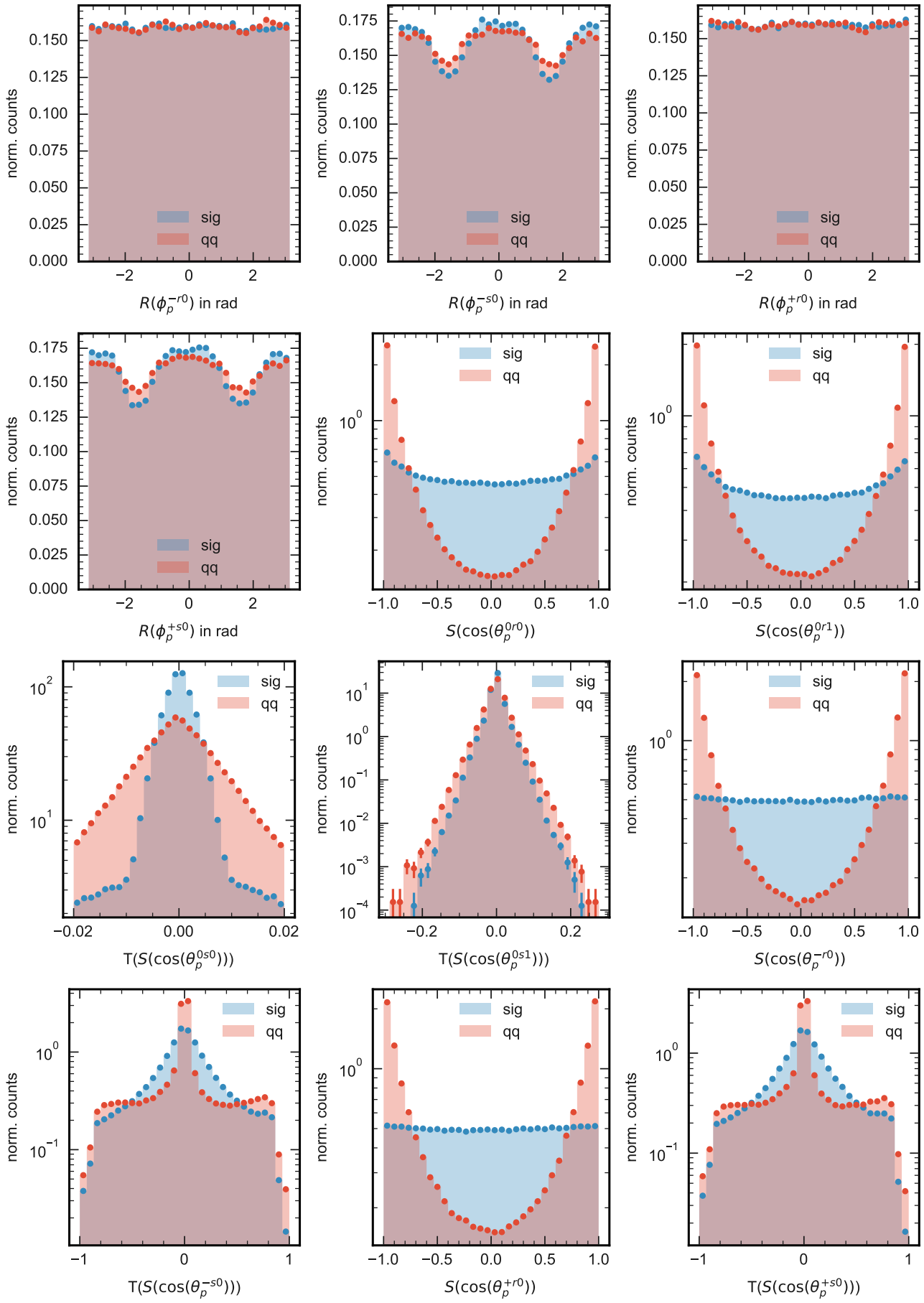

norm. counts
sig
qq
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad

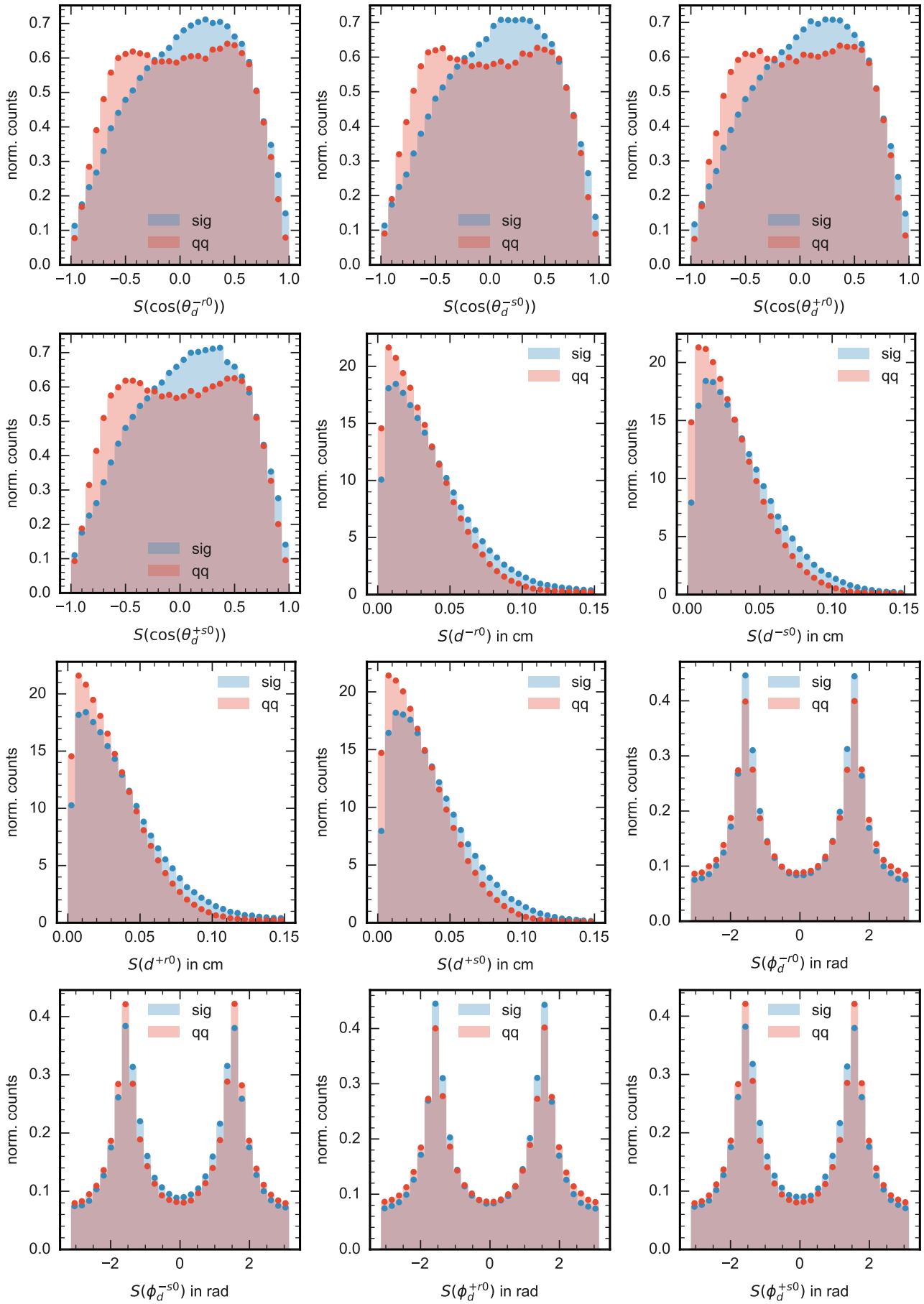

norm. counts
sig
qq
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad
$S(\phi_p^{+s0})$ in rad

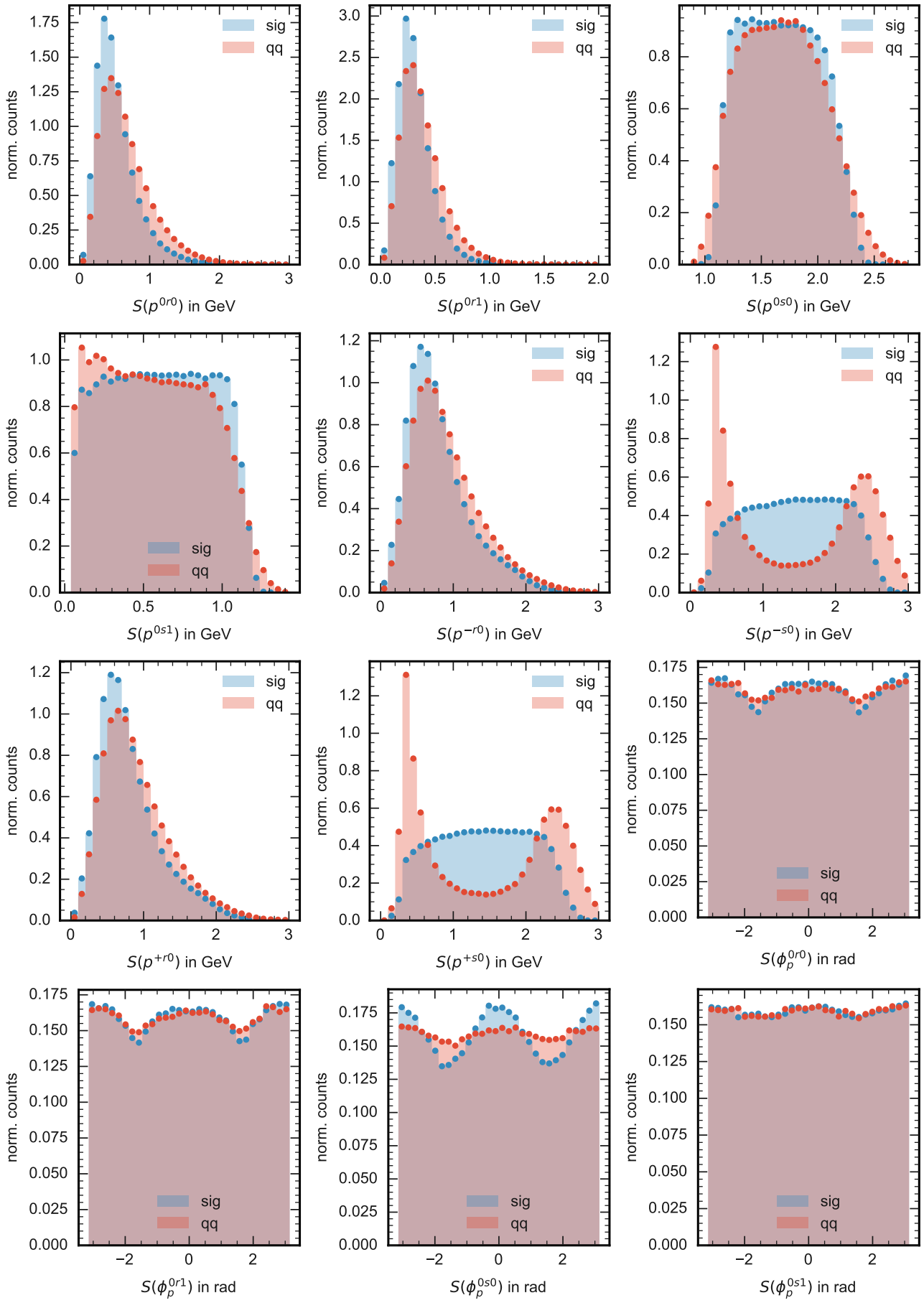
A.2.2. Topologically Similar Control Channel

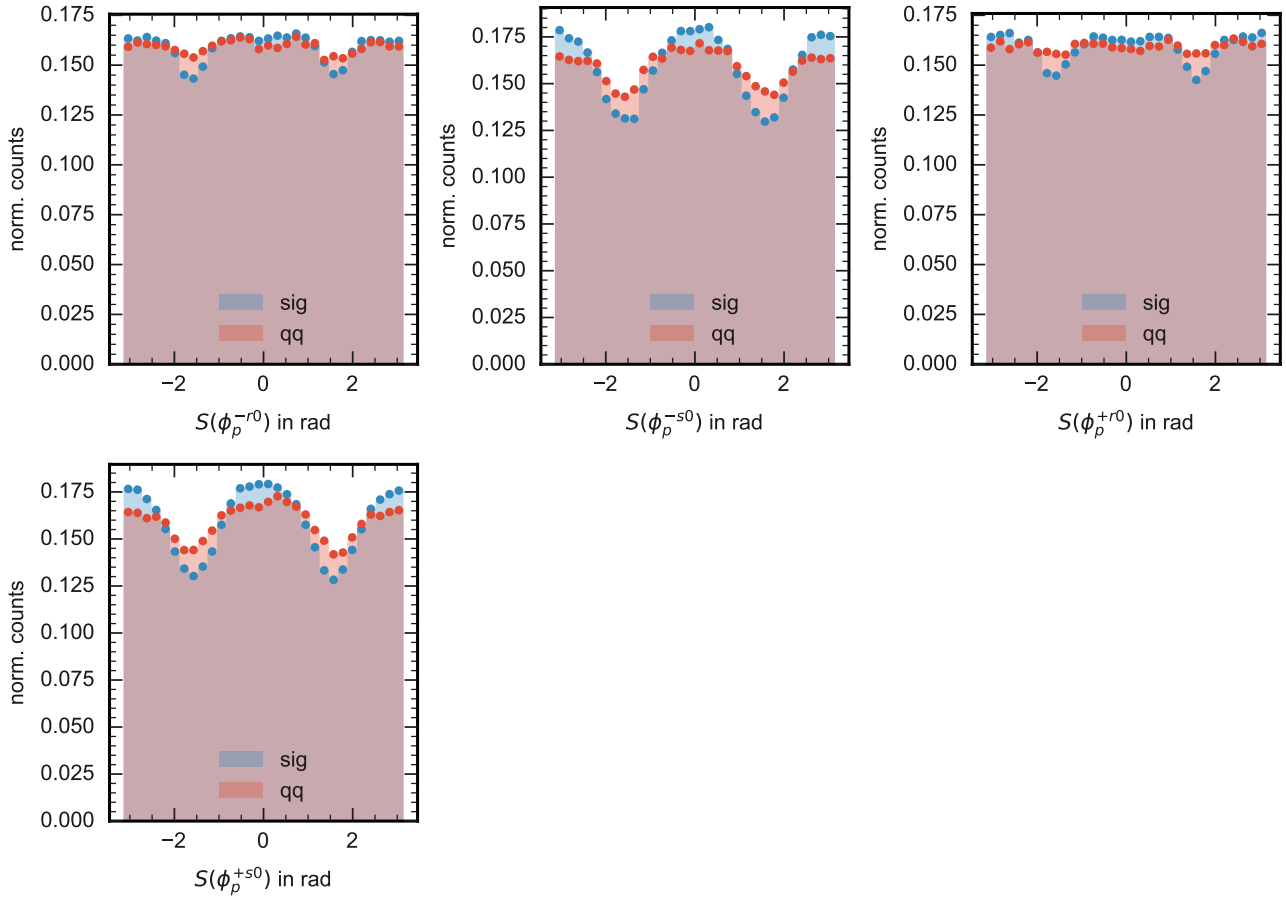



norm. counts
sig
qq
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad

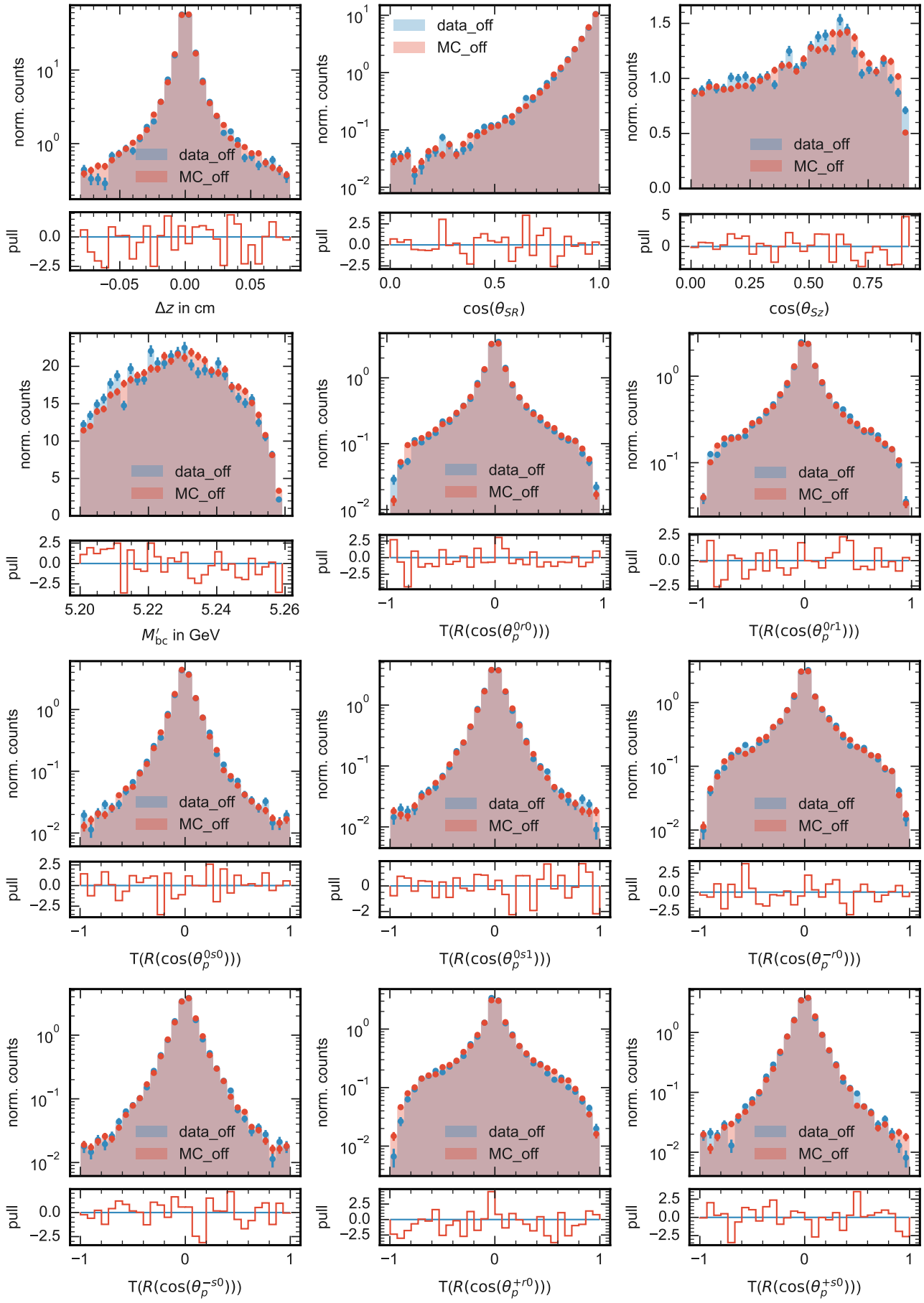

norm. counts
sig
qq
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
$S(\cos(\theta_p^{0r0}))$
$S(\cos(\theta_p^{0r1}))$
$T(S(\cos(\theta_p^{0s0})))$
$T(S(\cos(\theta_p^{0s1})))$
$S(\cos(\theta_p^{-r0}))$
$T(S(\cos(\theta_p^{-s0})))$
$S(\cos(\theta_p^{+r0}))$
$T(S(\cos(\theta_p^{+s0})))$

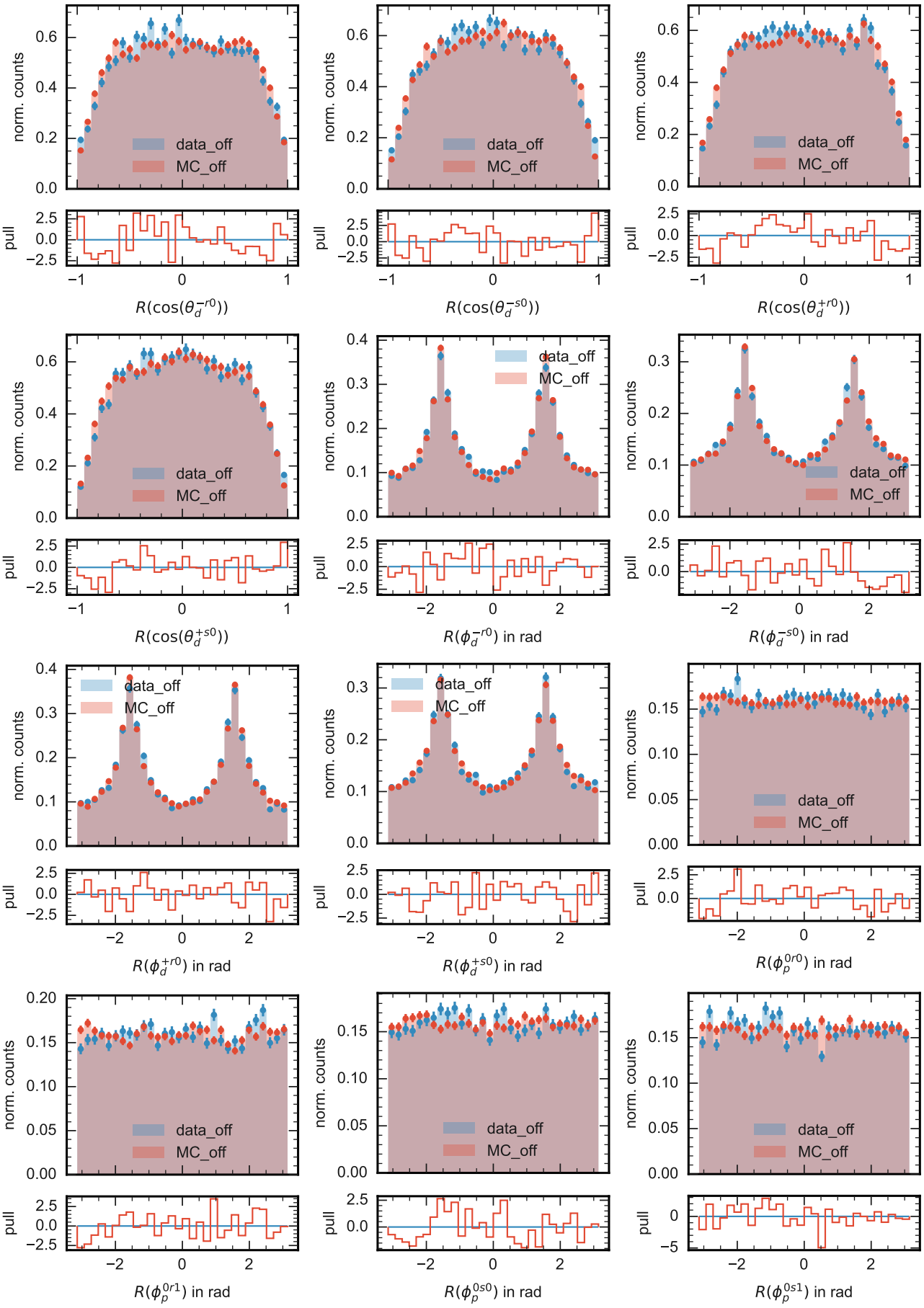

norm. counts
sig
qq
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad

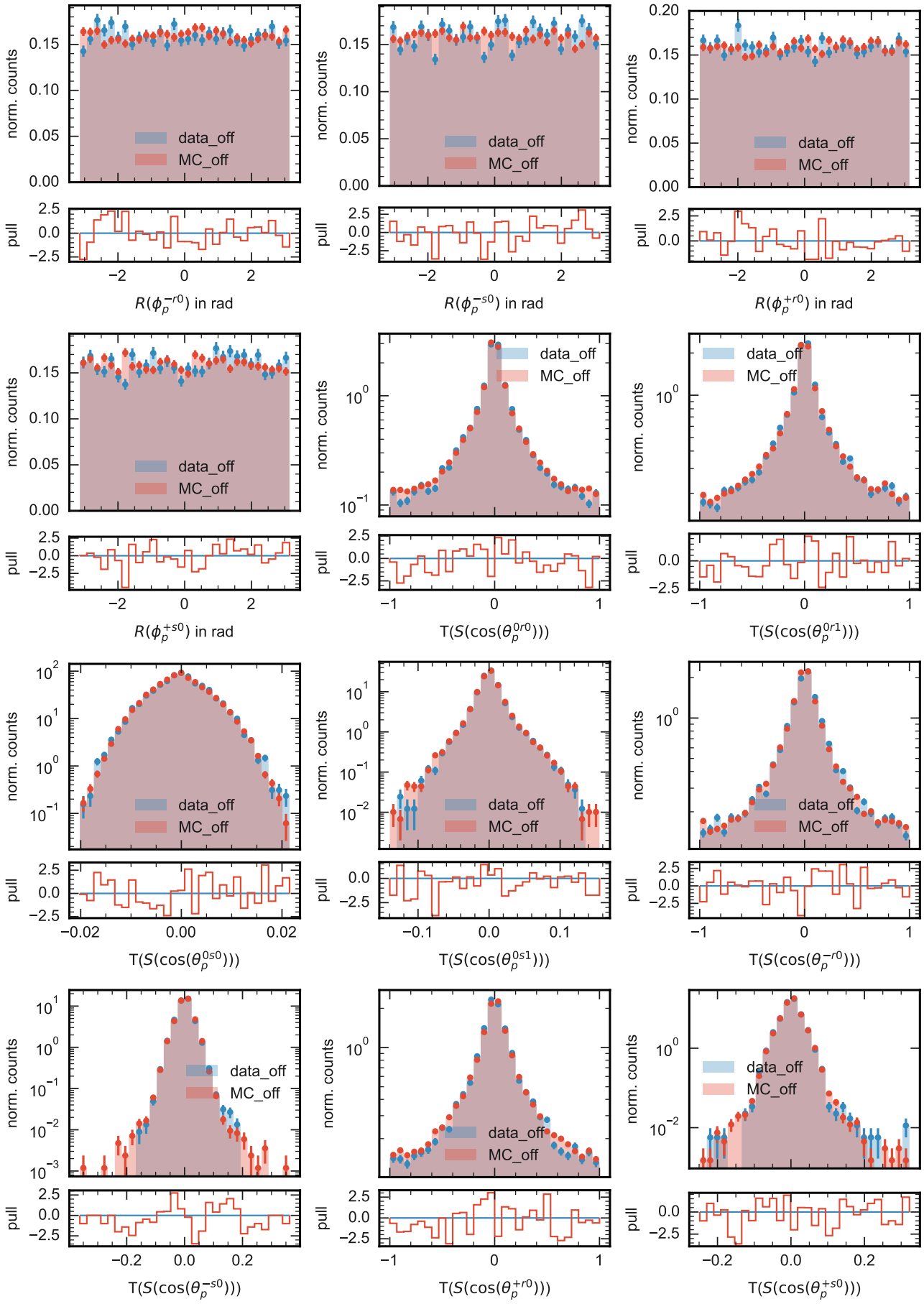

norm. counts
sig
qq
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad

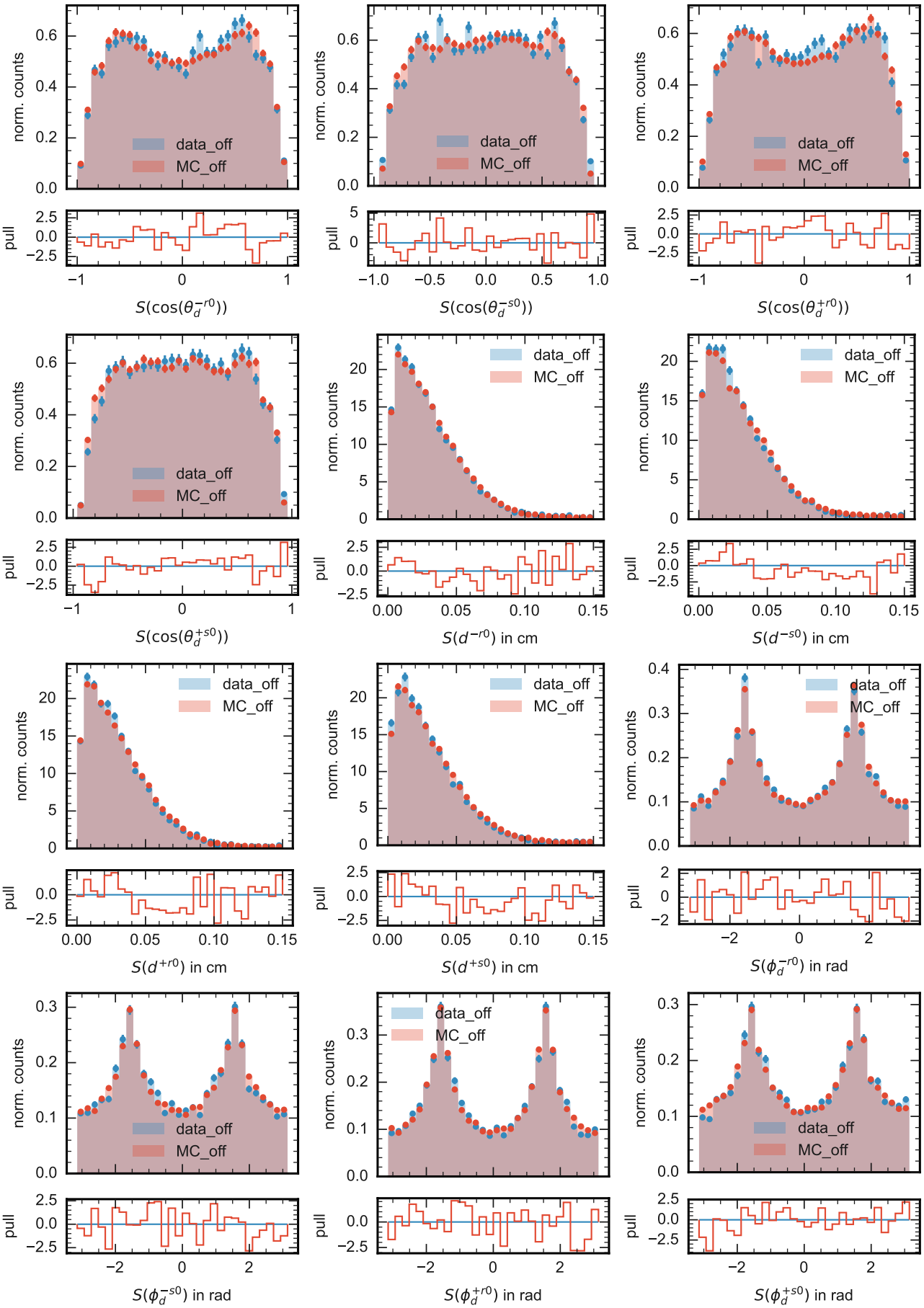

norm. counts
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−2
0
2
sig
qq
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad
$S(\phi_p^{+s0})$ in rad

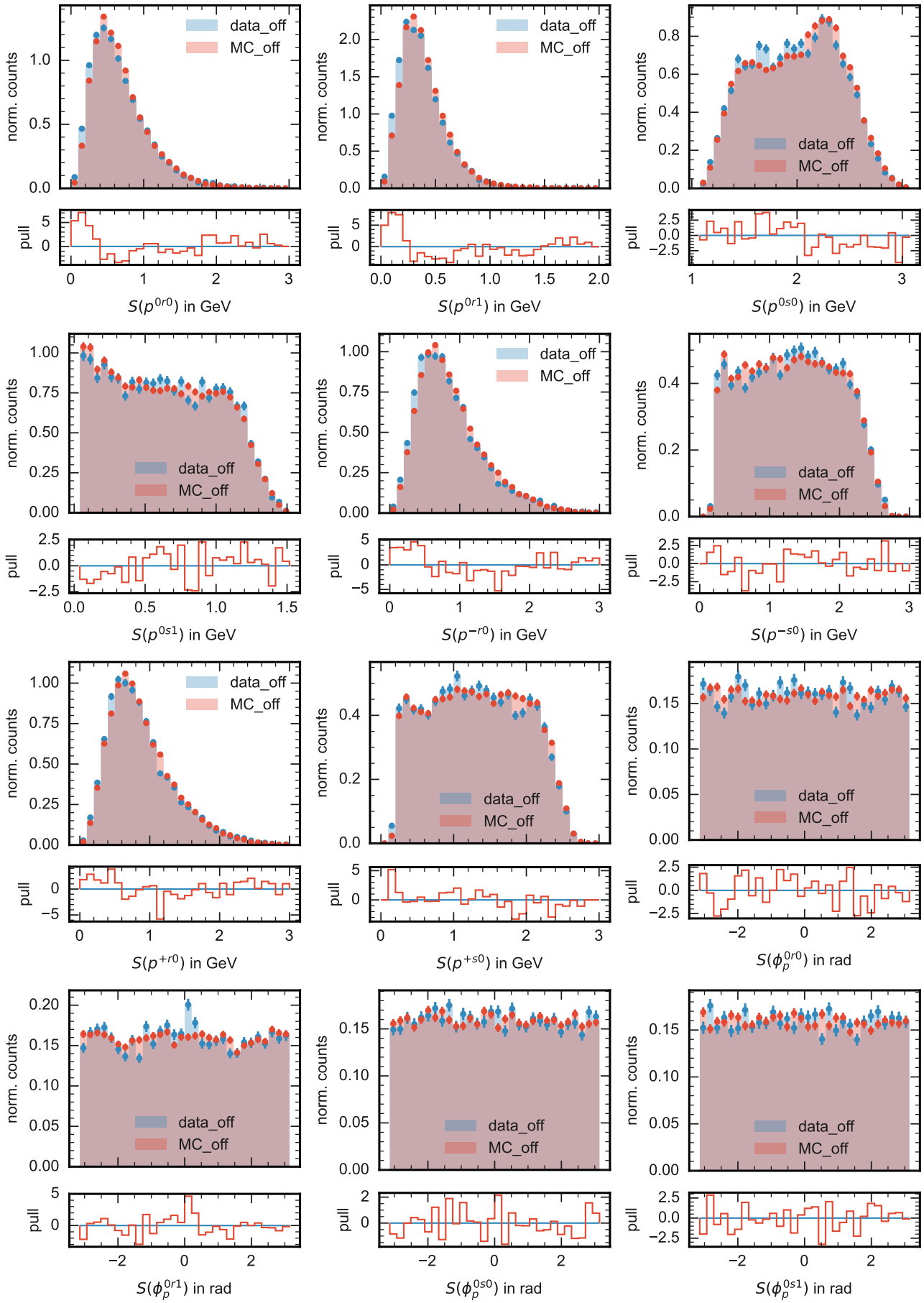
A.3. Off-Resonance Data vs Off-Resonance MC Plots

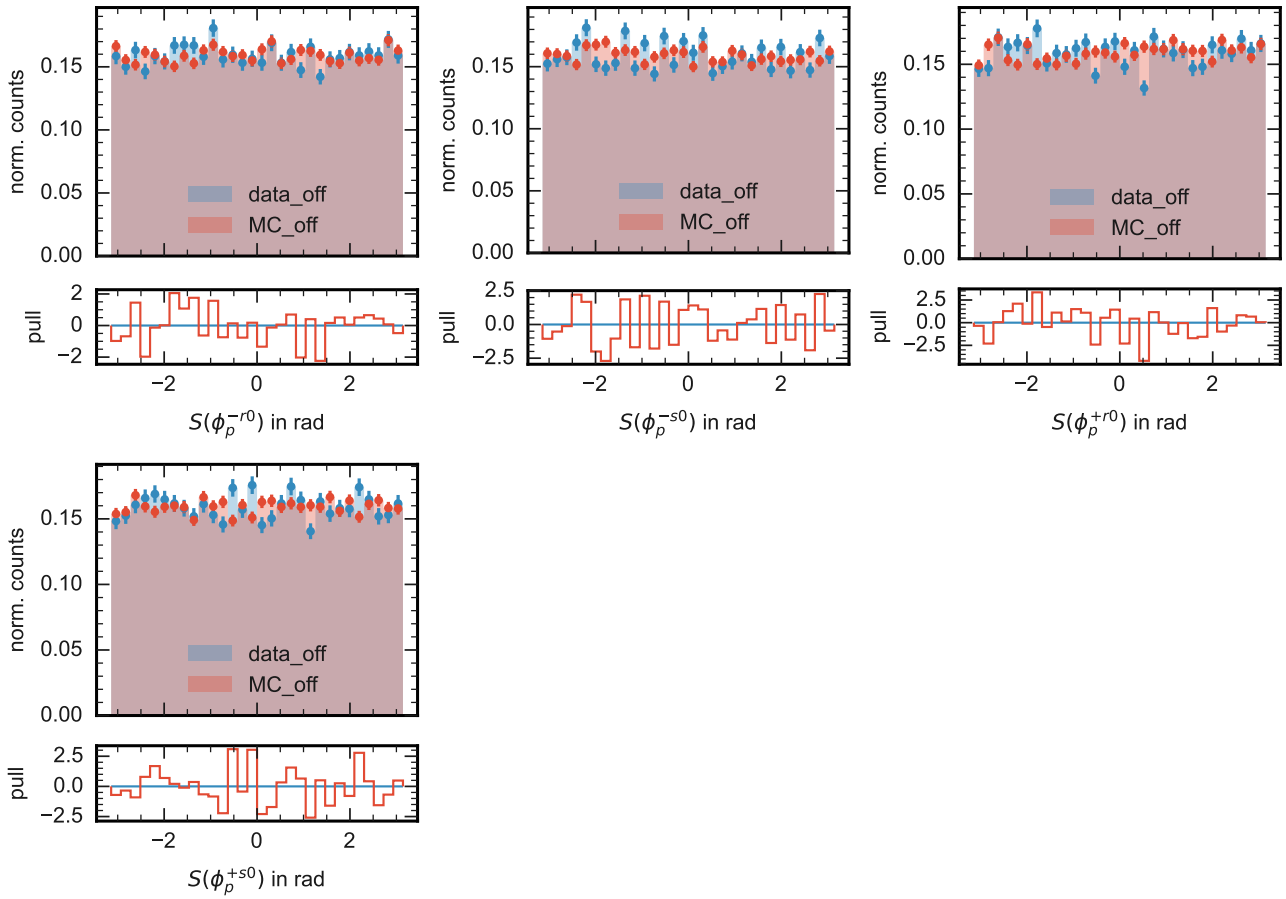



norm. counts
data_off
MC_off
pull
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad

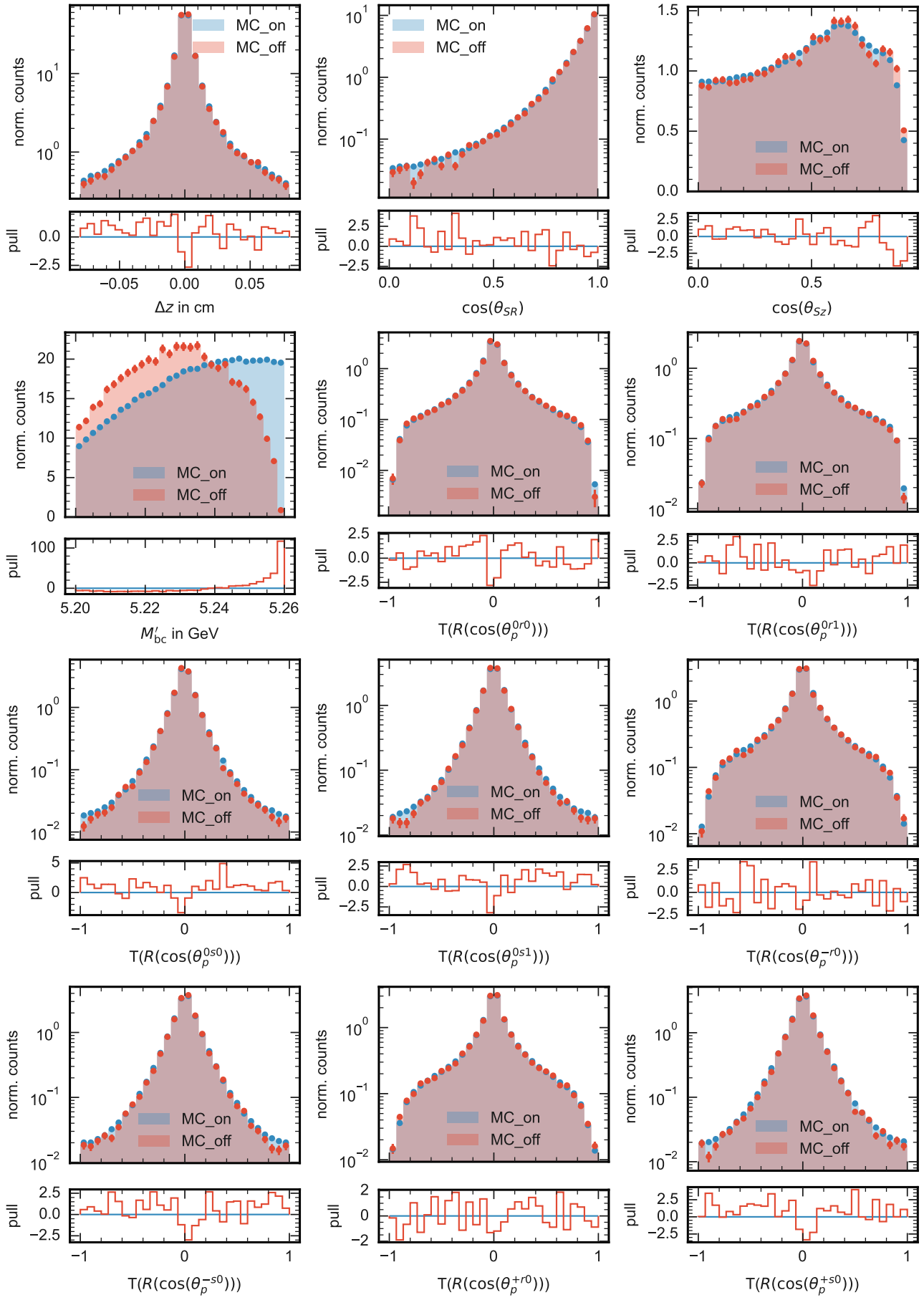

norm. counts
data_off
MC_off
pull
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
T(S(cos(θ_p^{0r0})))
T(S(cos(θ_p^{0r1})))
T(S(cos(θ_p^{0s0})))
T(S(cos(θ_p^{0s1})))
T(S(cos(θ_p^{-r0})))
T(S(cos(θ_p^{-s0})))
T(S(cos(θ_p^{+r0})))
T(S(cos(θ_p^{+s0})))

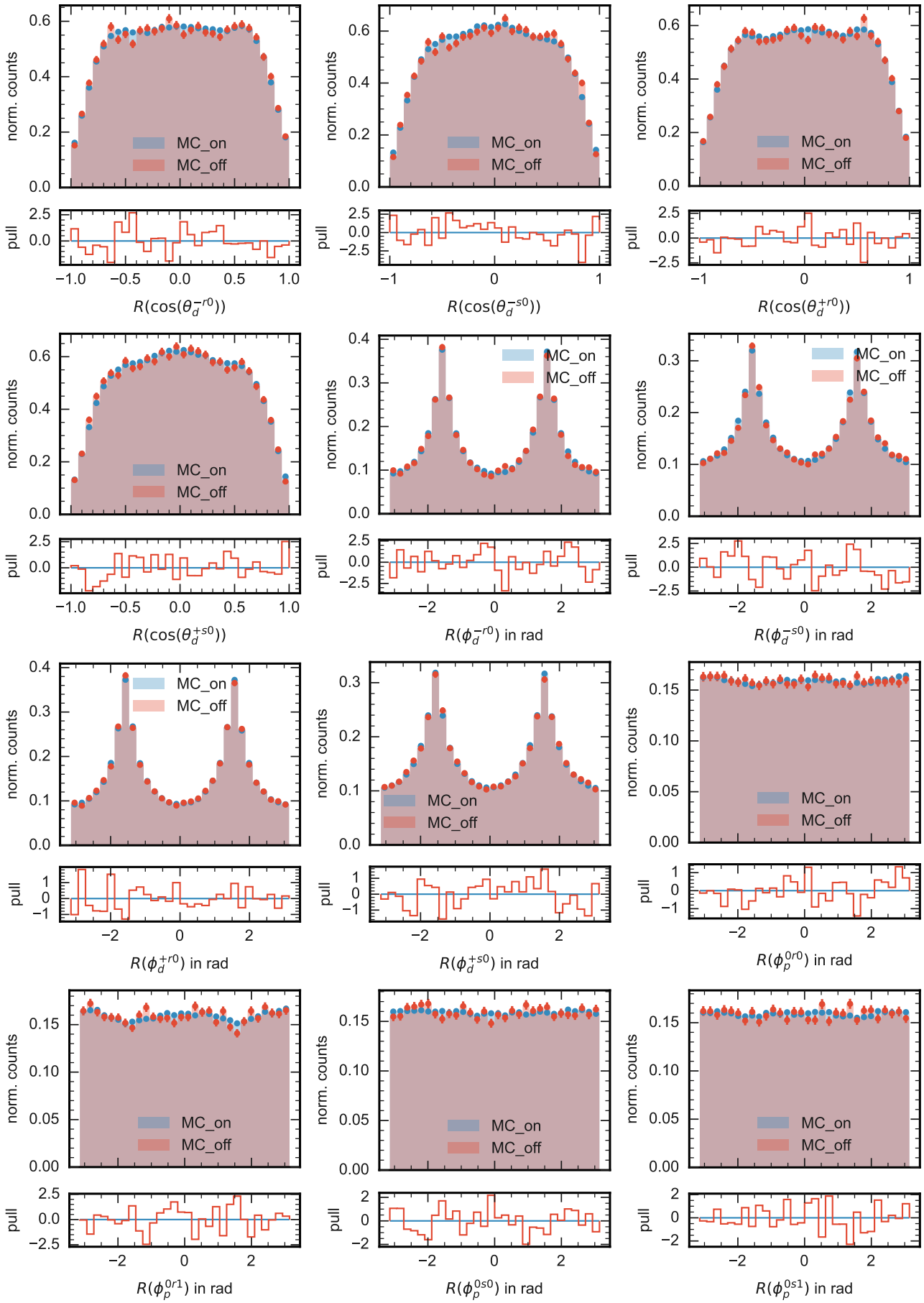

norm. counts
data_off
MC_off
pull
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad

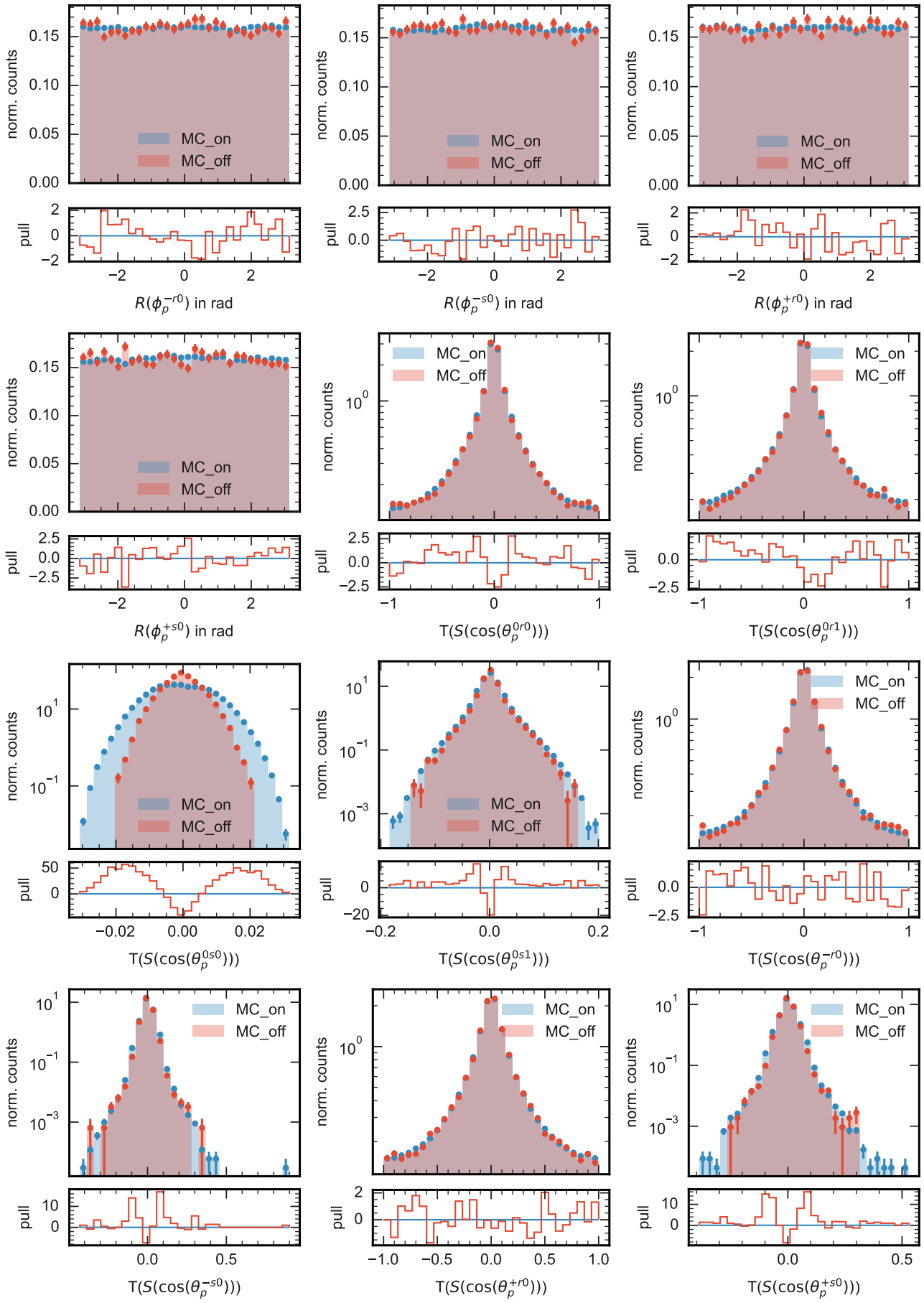

data_off
MC_off
norm. counts
pull
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad

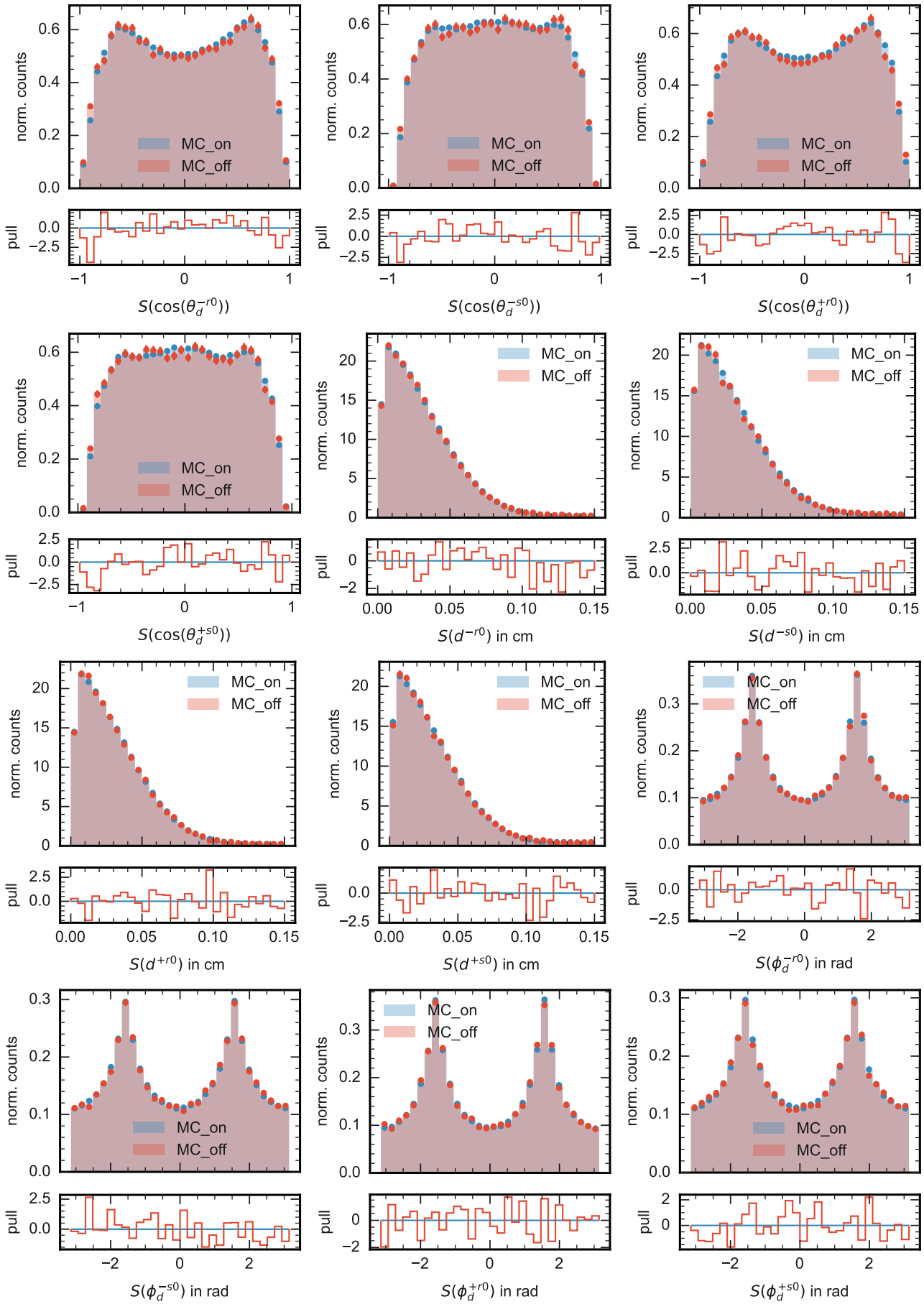

norm. counts
data_off
MC_off
pull
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad
$S(\phi_p^{+s0})$ in rad

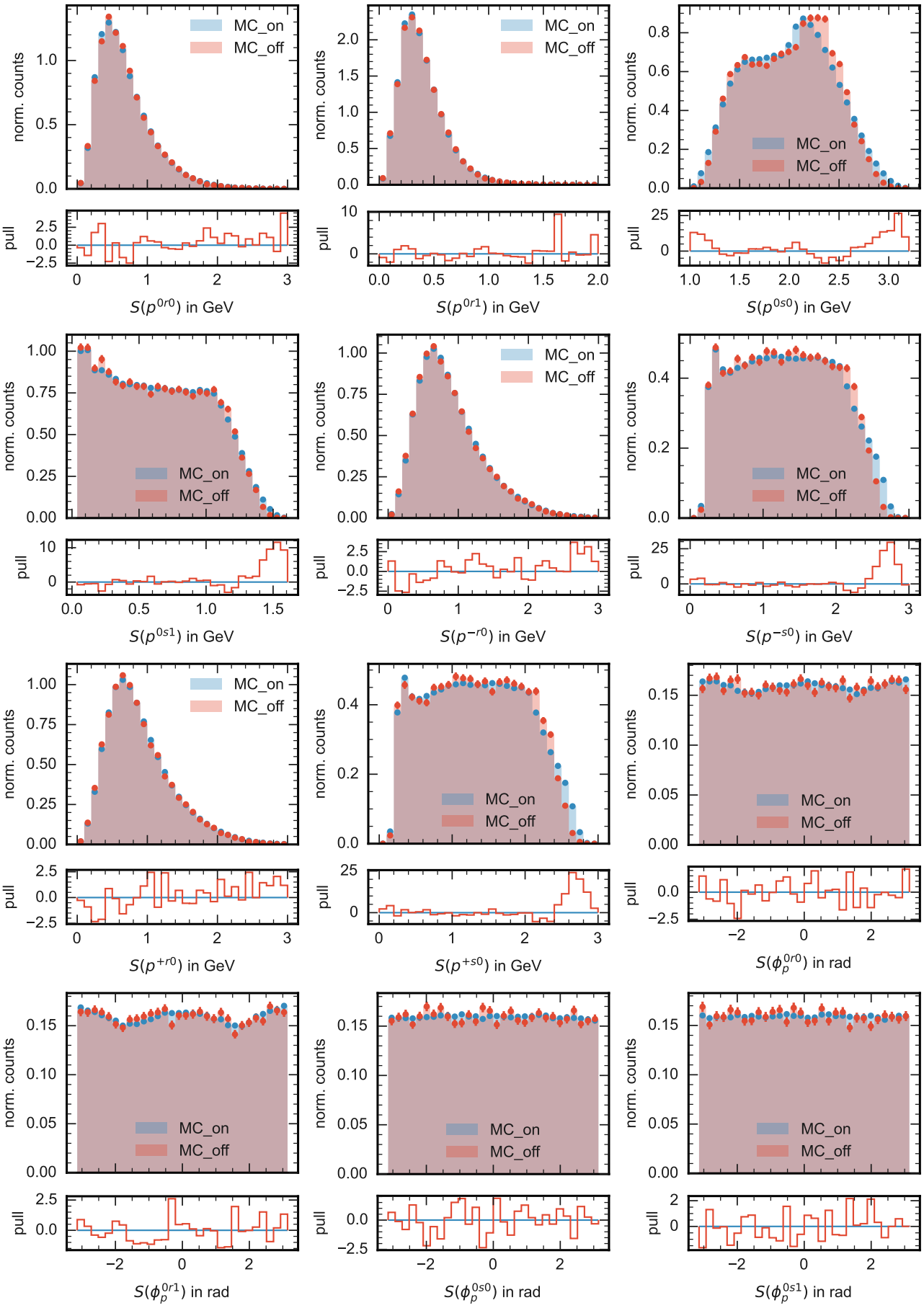
A.4. On-Resonance MC vs Off-Resonance MC Plots

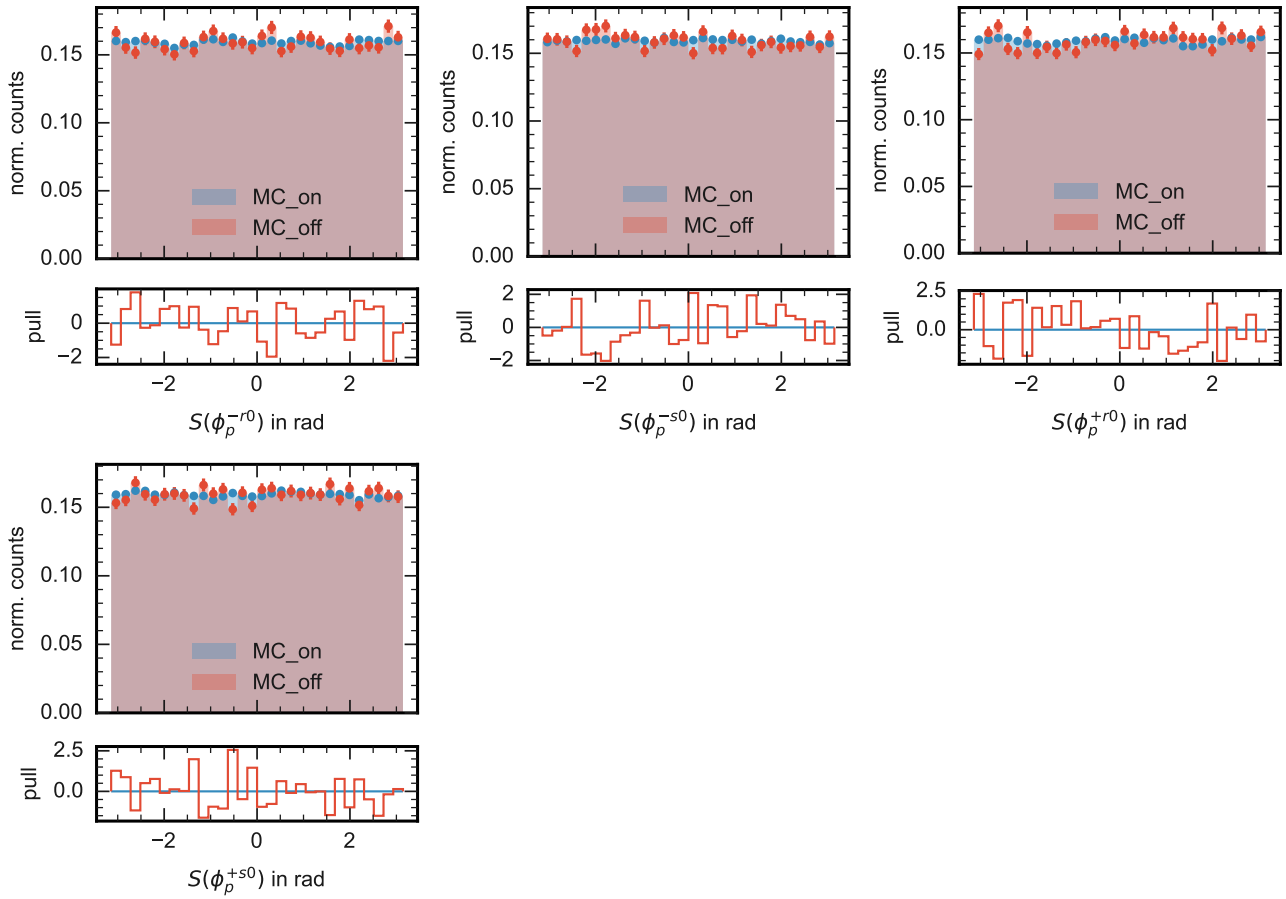



norm. counts
MC_on
MC_off
pull
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad


norm. counts
MC_on
MC_off
pull
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
$T(S(\cos(\theta_p^{0r0})))$
$T(S(\cos(\theta_p^{0r1})))$
$T(S(\cos(\theta_p^{0s0})))$
$T(S(\cos(\theta_p^{0s1})))$
$T(S(\cos(\theta_p^{-r0})))$
$T(S(\cos(\theta_p^{-s0})))$
$T(S(\cos(\theta_p^{+r0})))$
$T(S(\cos(\theta_p^{+s0})))$

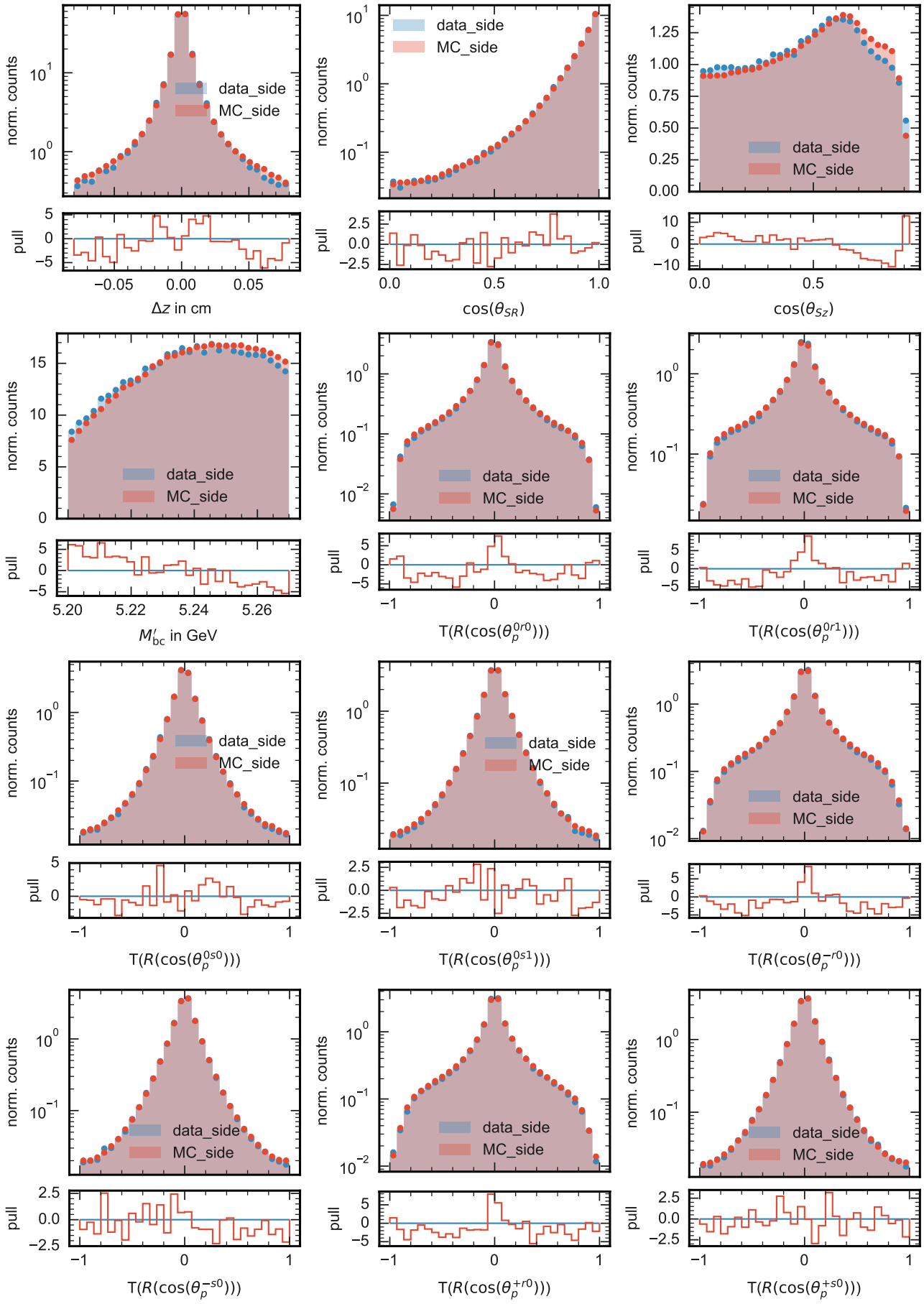

norm. counts
MC_on
MC_off
pull
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad

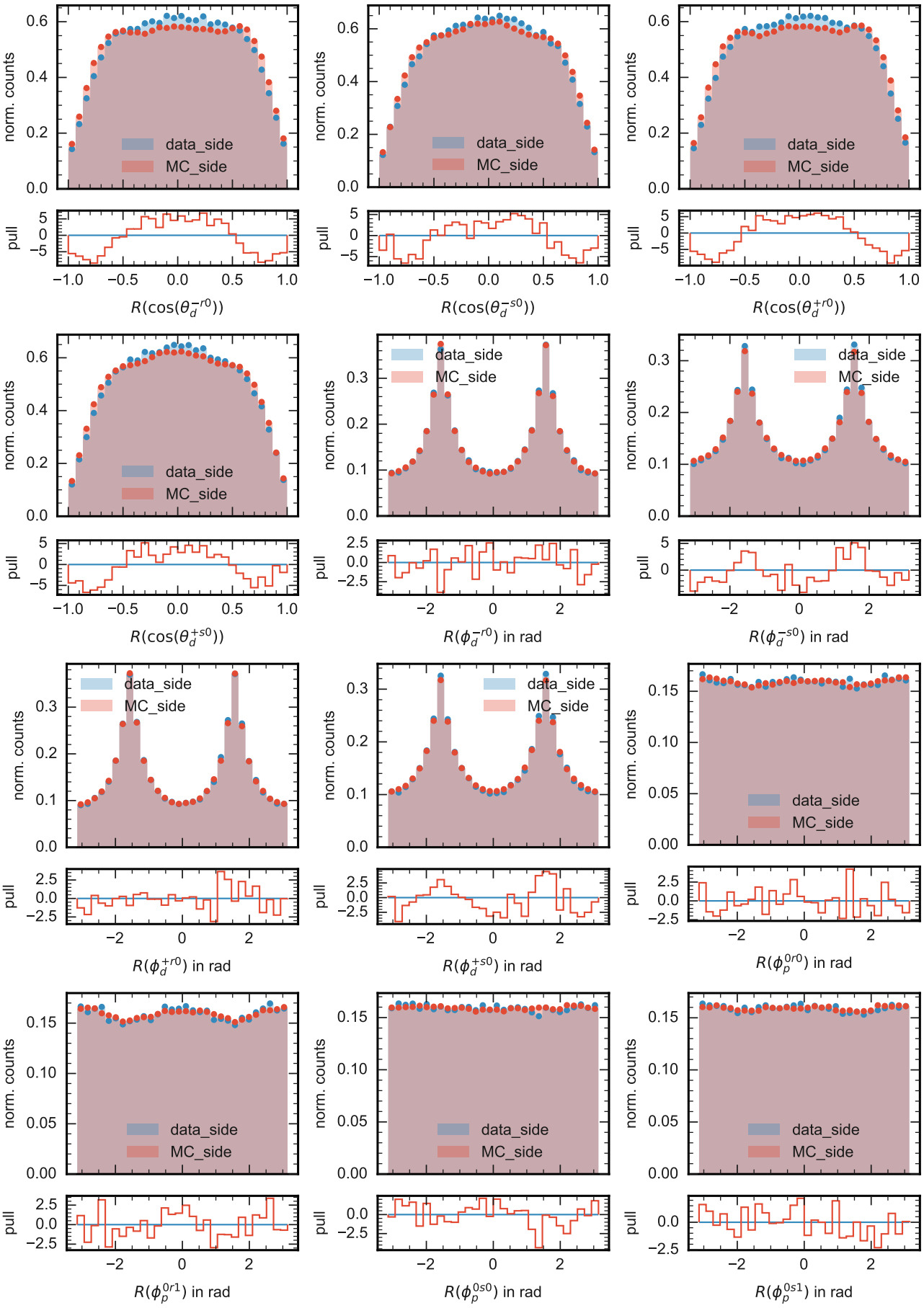

MC_on
MC_off
norm. counts
pull
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad

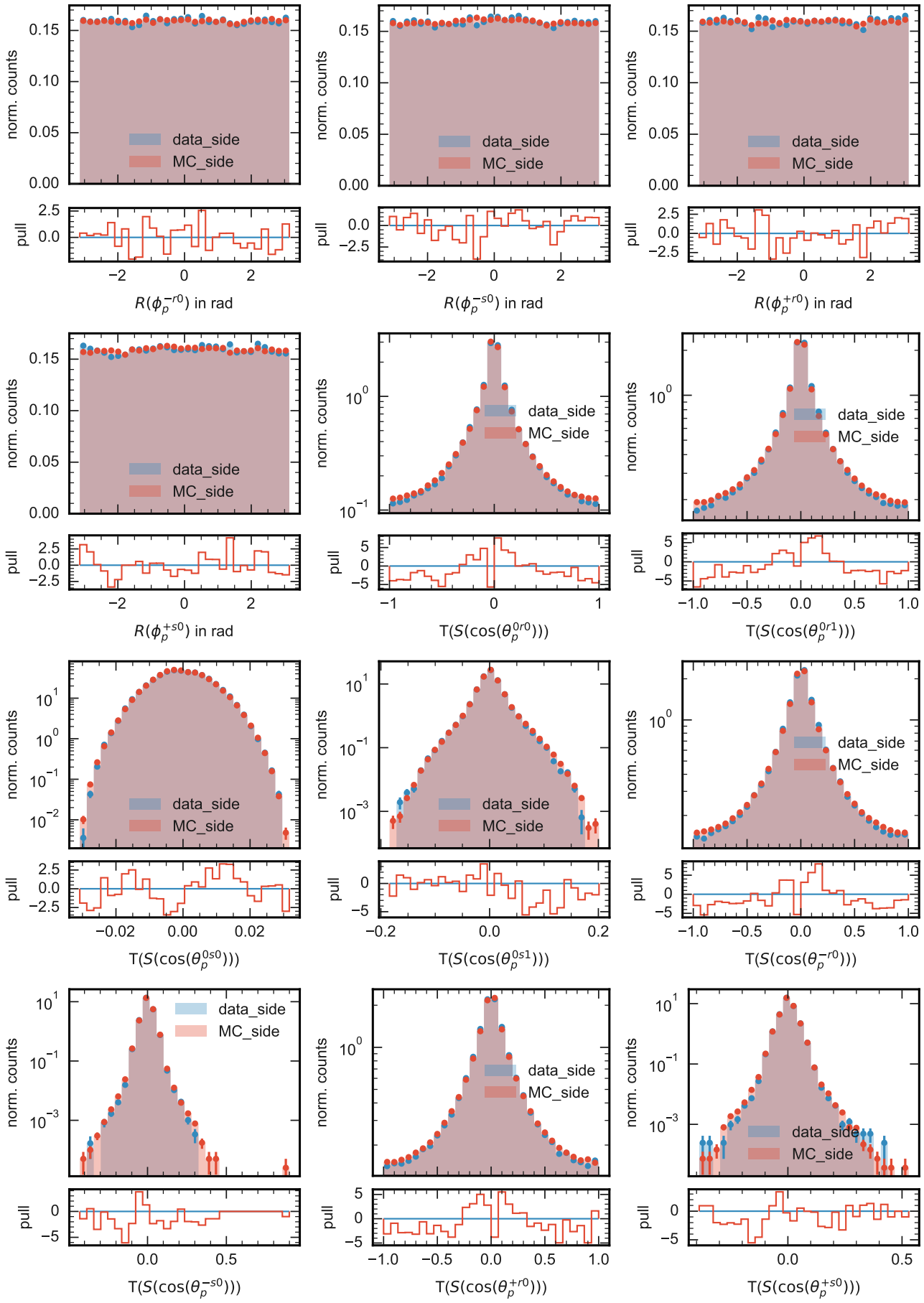

norm. counts
0.15
0.10
0.05
0.00
MC_on
MC_off
pull
2
0
−2
2.5
0.0
−2
0
2
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad

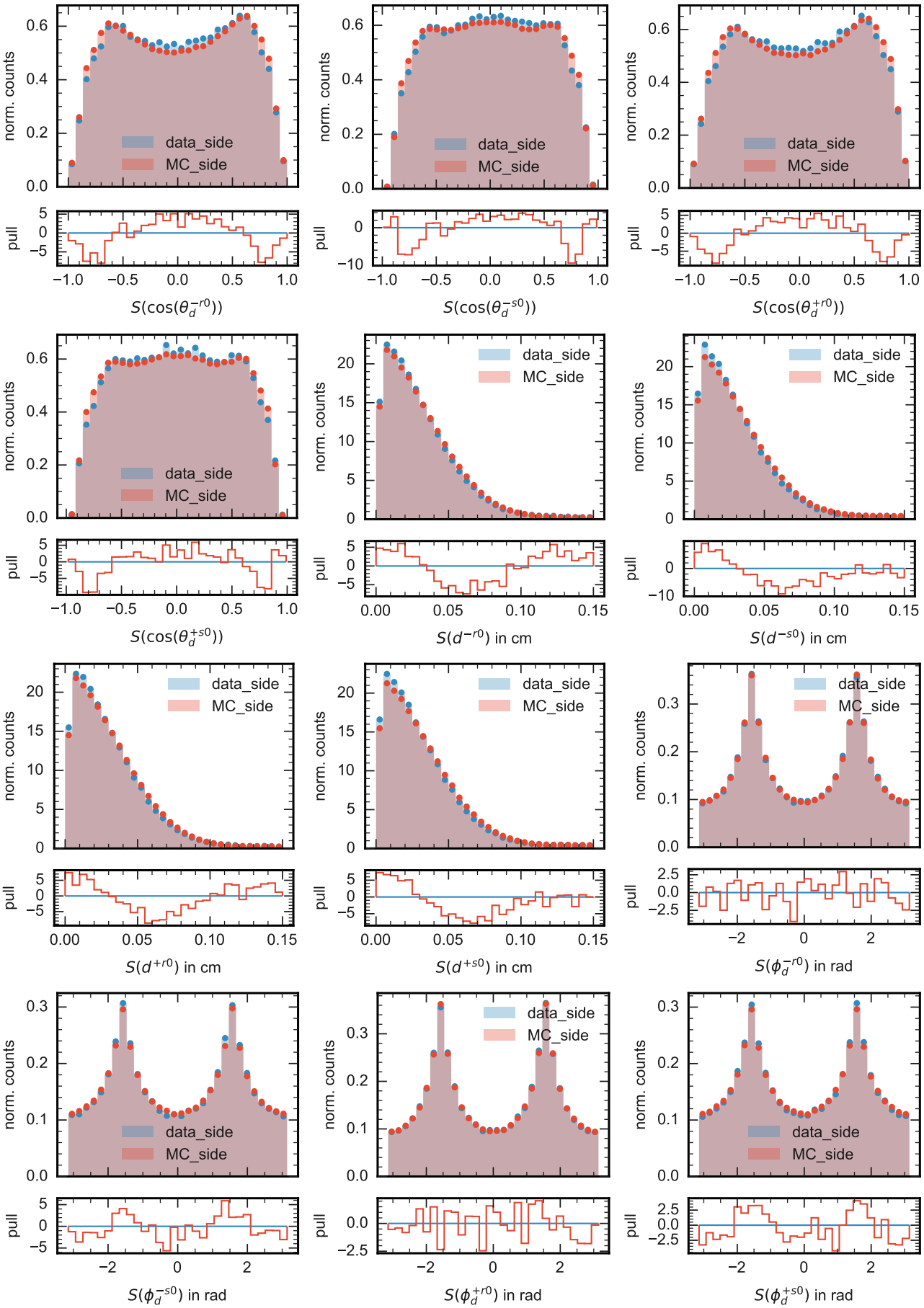
norm. counts
0.15
0.10
0.05
0.00
MC_on
MC_off
pull
2.5
0.0
−2
0
2
$S(\phi_p^{+s0})$ in rad

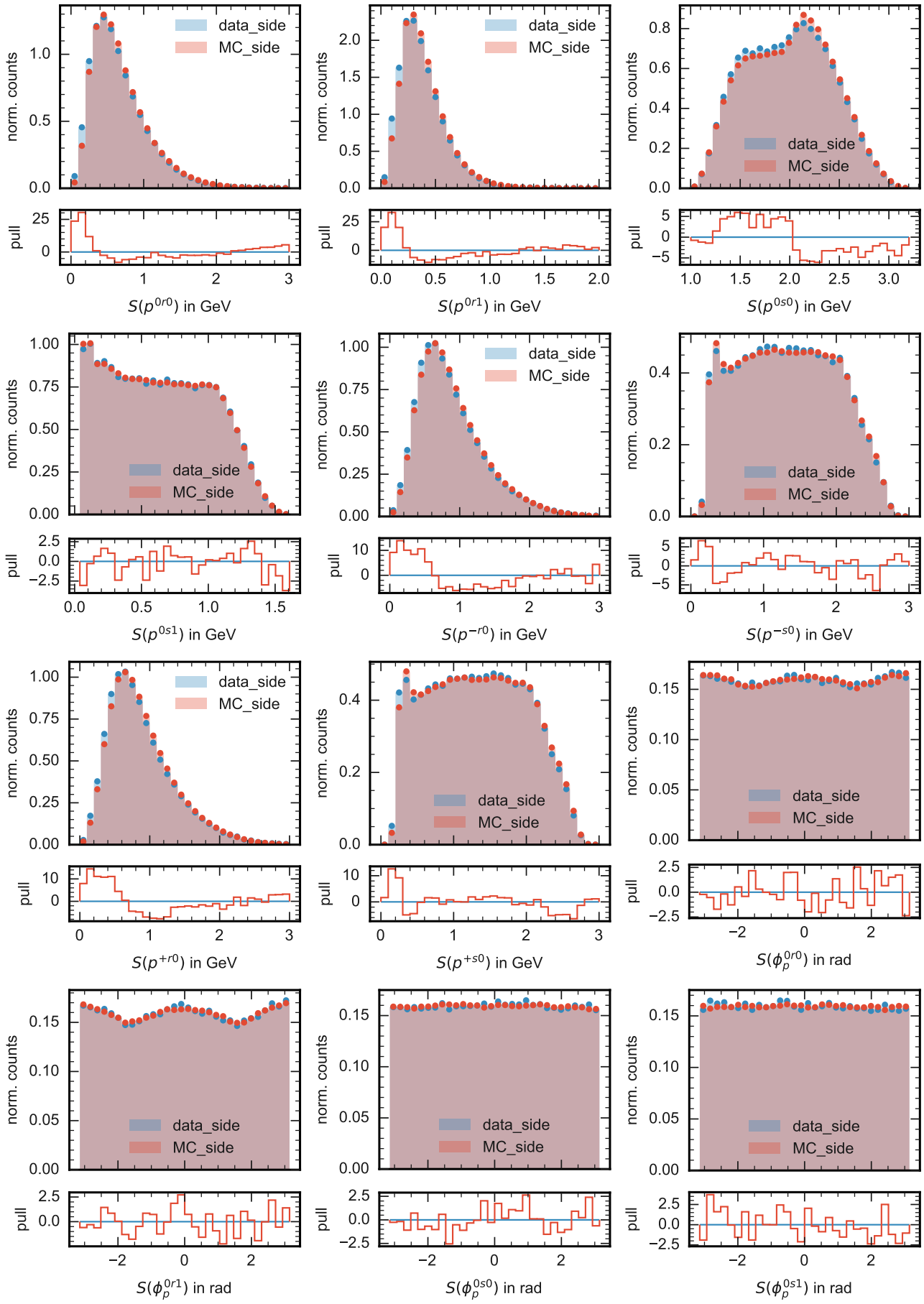
A.5. Sideband Data vs Sideband MC Plots

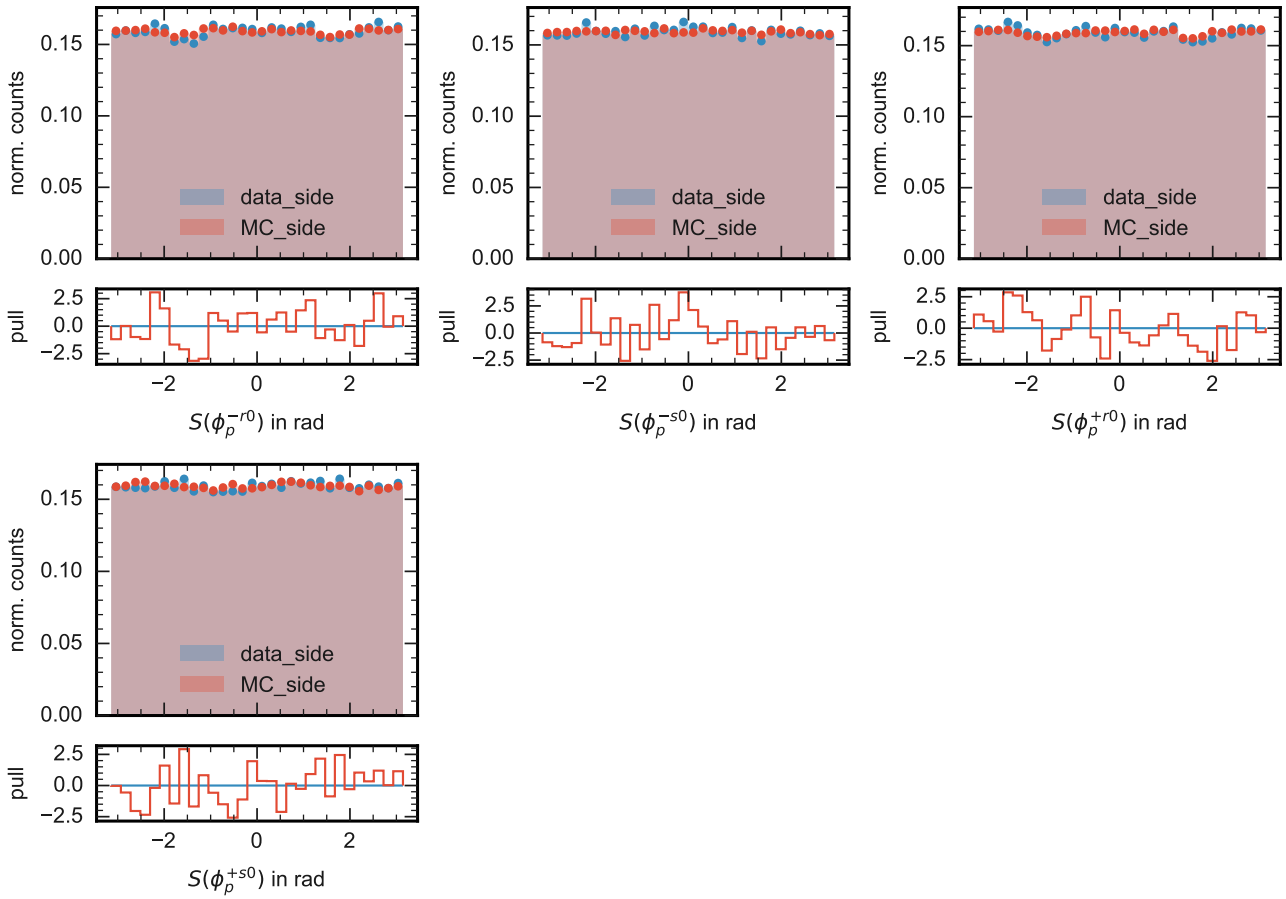



norm. counts
data_side
MC_side
pull
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad

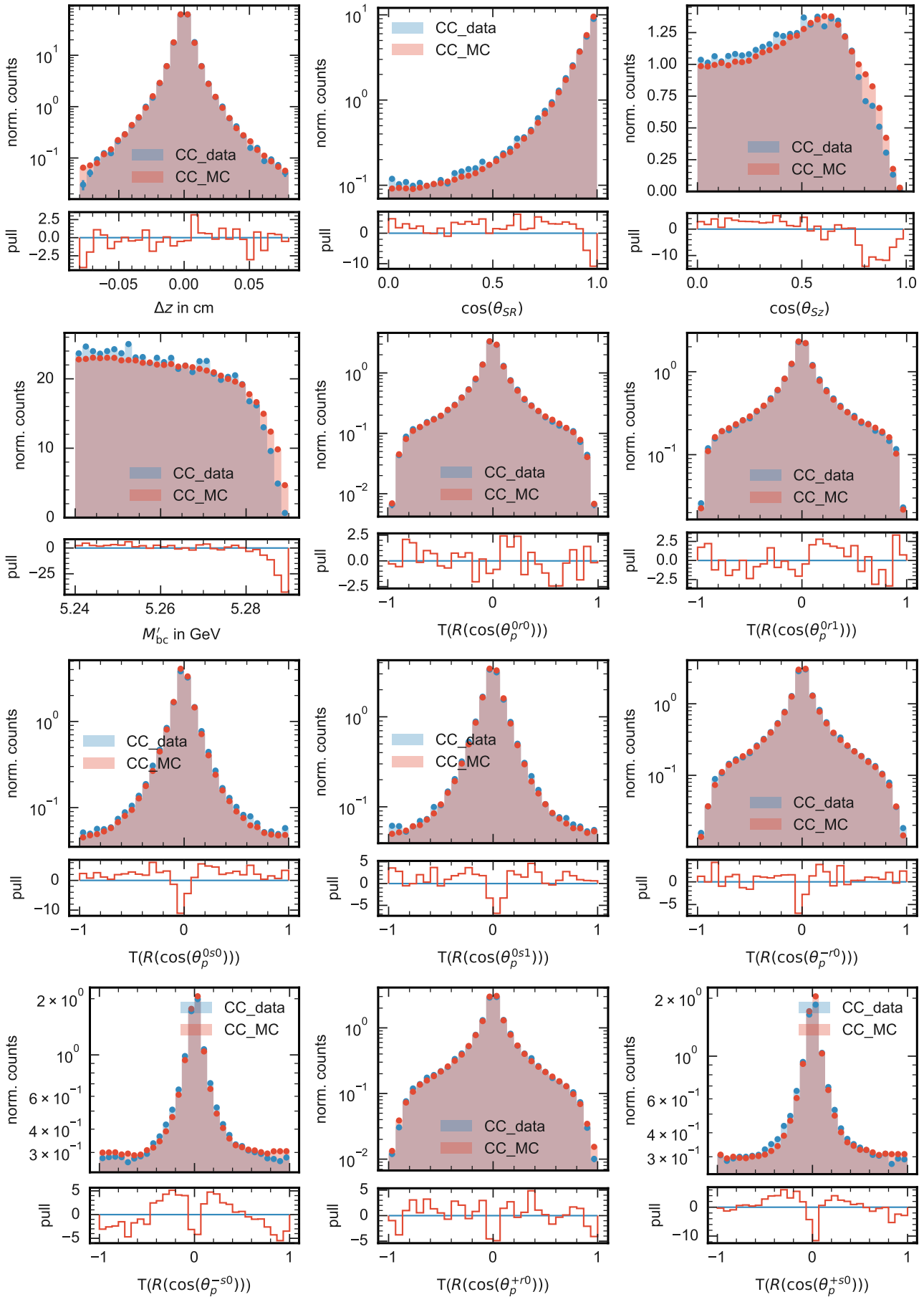

norm. counts
data_side
MC_side
pull
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
$T(S(\cos(\theta_p^{0r0})))$
$T(S(\cos(\theta_p^{0r1})))$
$T(S(\cos(\theta_p^{0s0})))$
$T(S(\cos(\theta_p^{0s1})))$
$T(S(\cos(\theta_p^{-r0})))$
$T(S(\cos(\theta_p^{-s0})))$
$T(S(\cos(\theta_p^{+r0})))$
$T(S(\cos(\theta_p^{+s0})))$

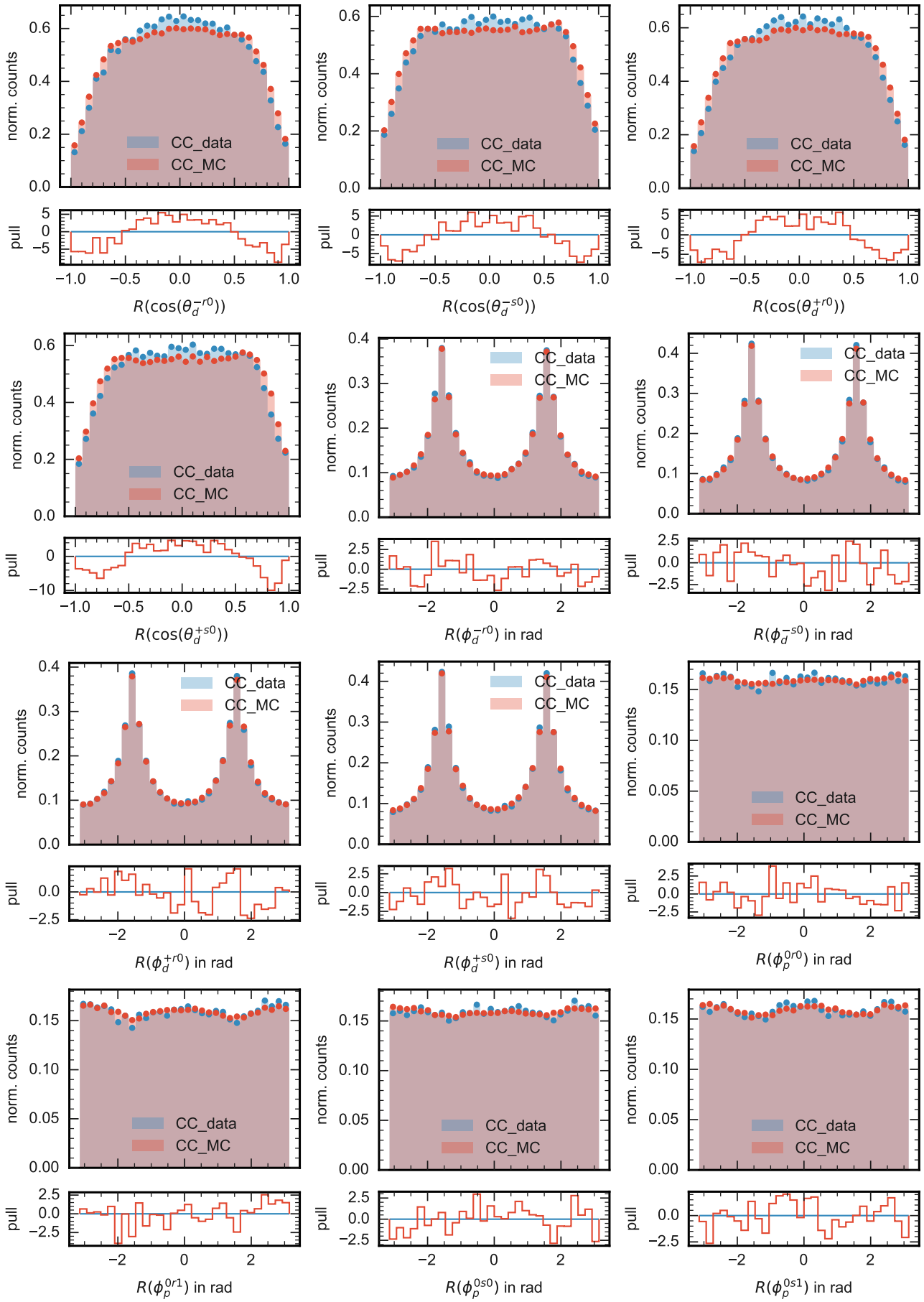

norm. counts
data_side
MC_side
pull
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad

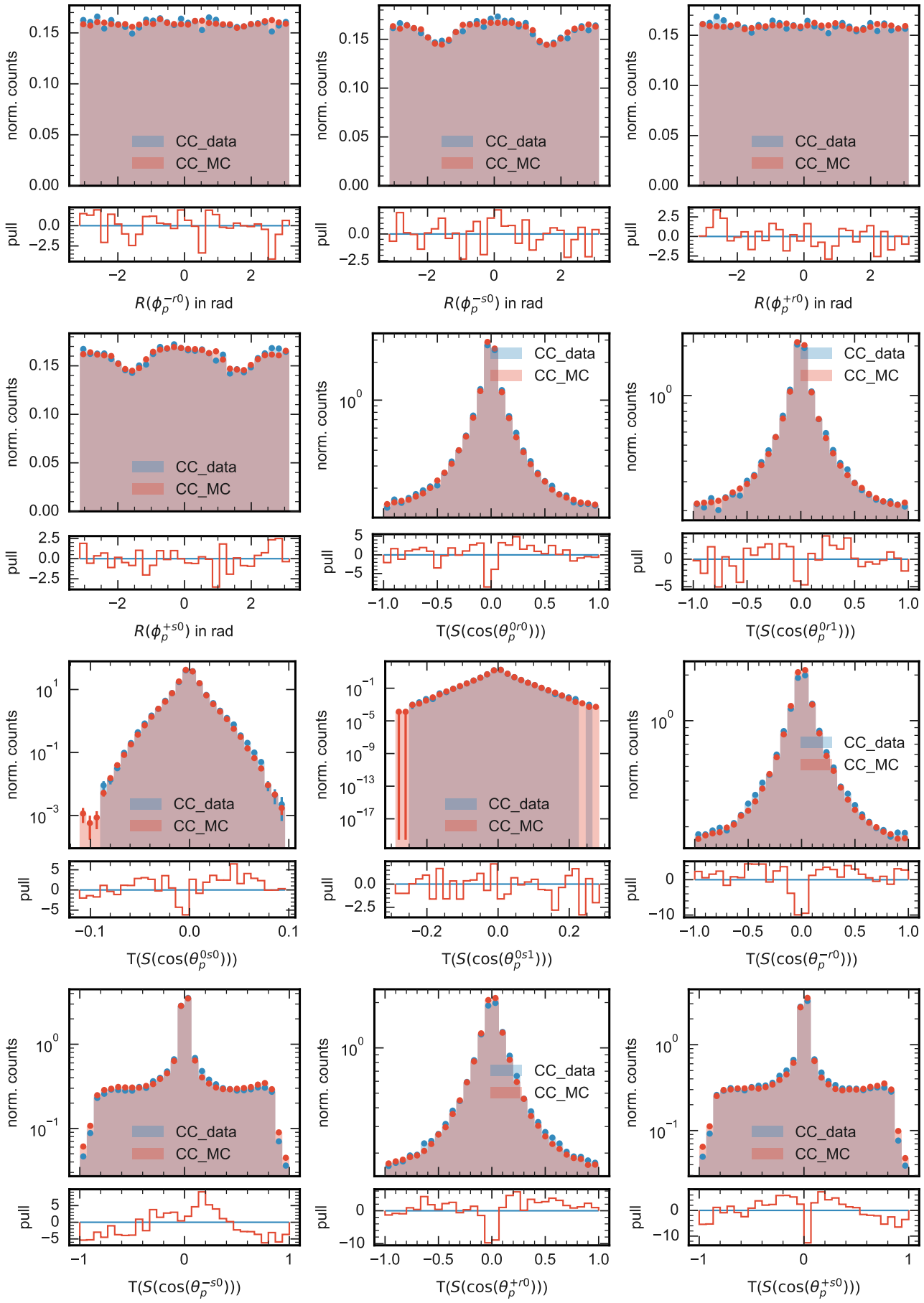

data_side
MC_side
norm. counts
pull
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad

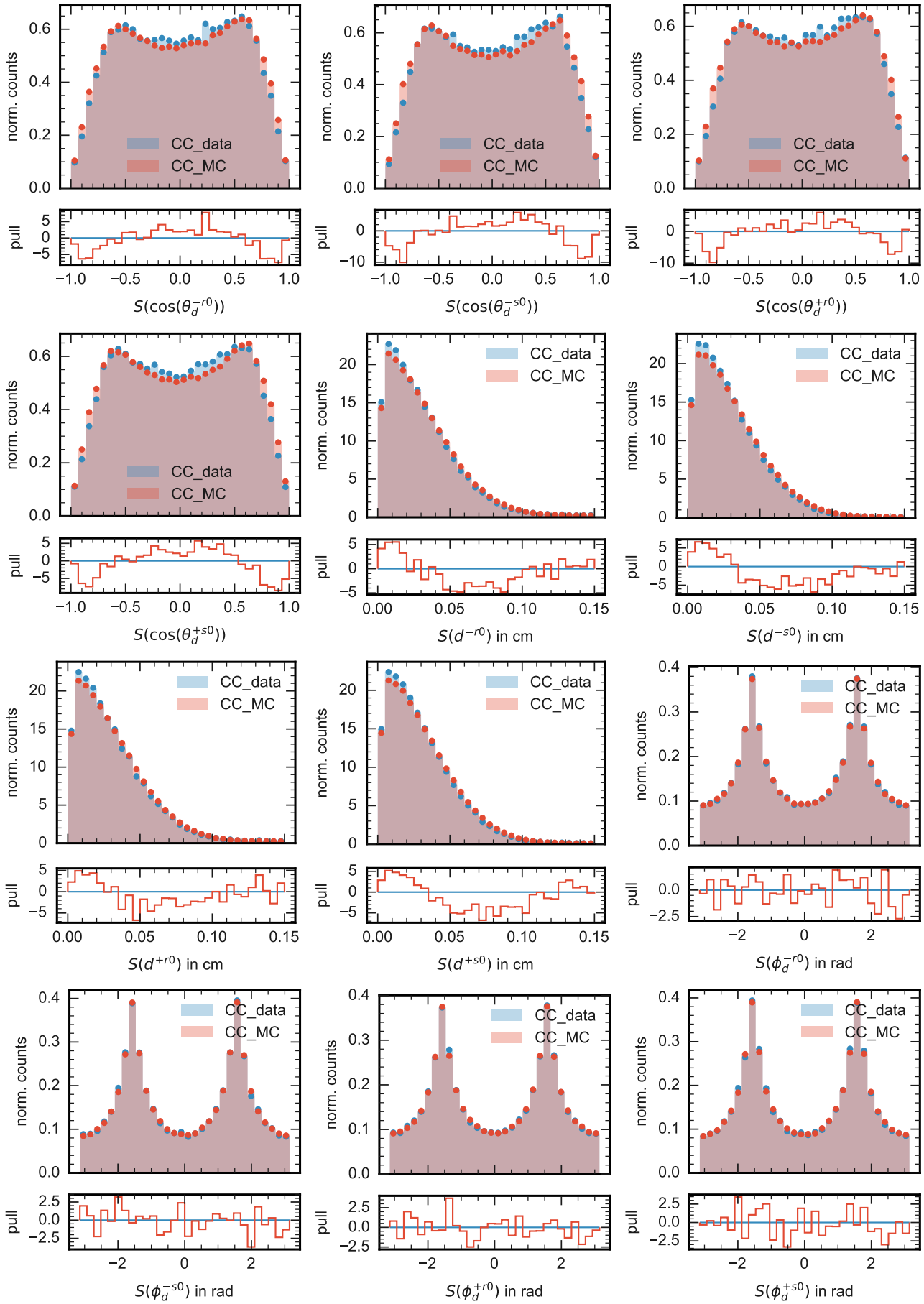

norm. counts
0.15
0.10
0.05
0.00
data_side
MC_side
pull
2.5
0.0
−2.5
−2
0
2
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad
$S(\phi_p^{+s0})$ in rad

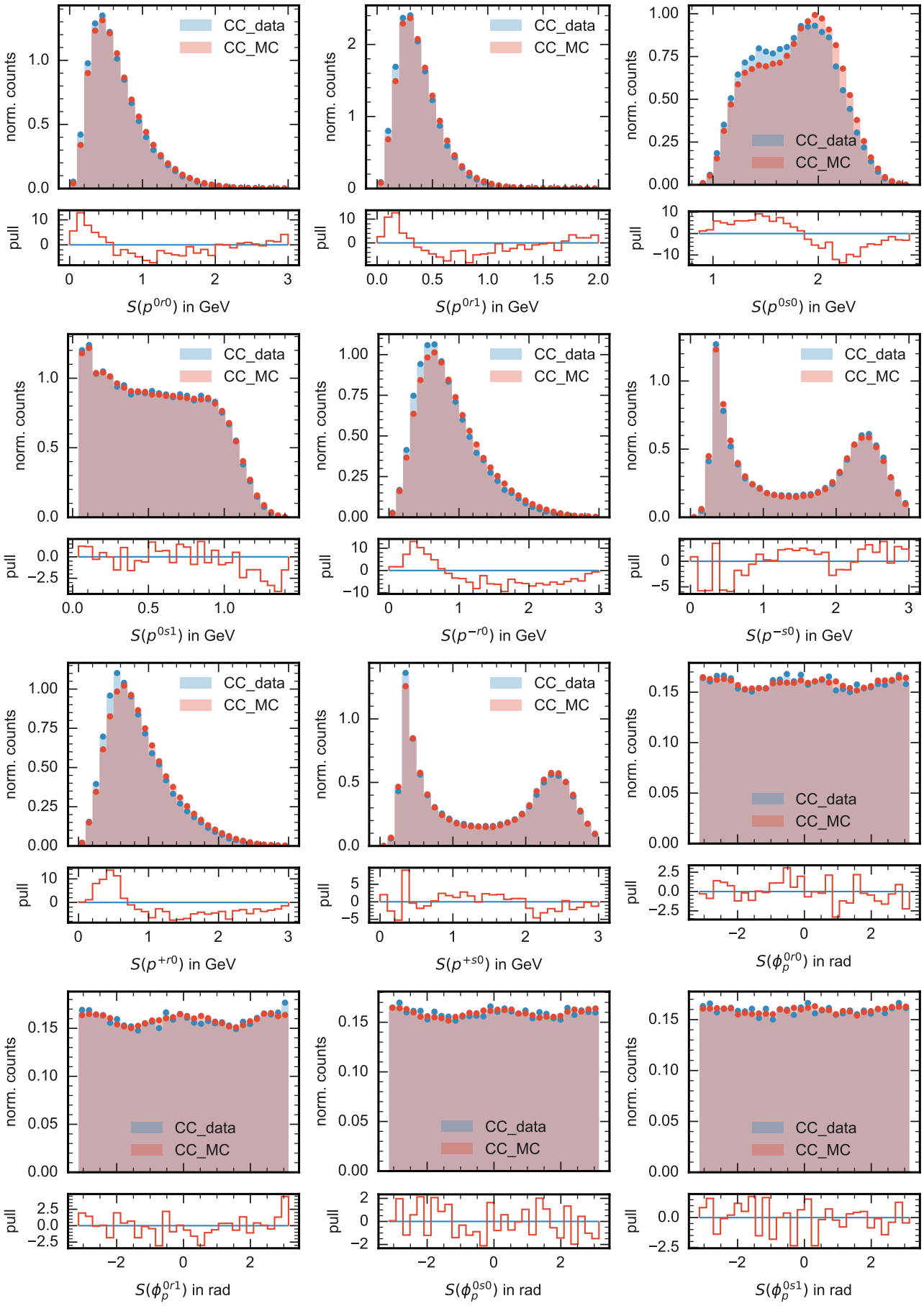
A.6. Topologically Similar Control Channel Data vs MC Plots

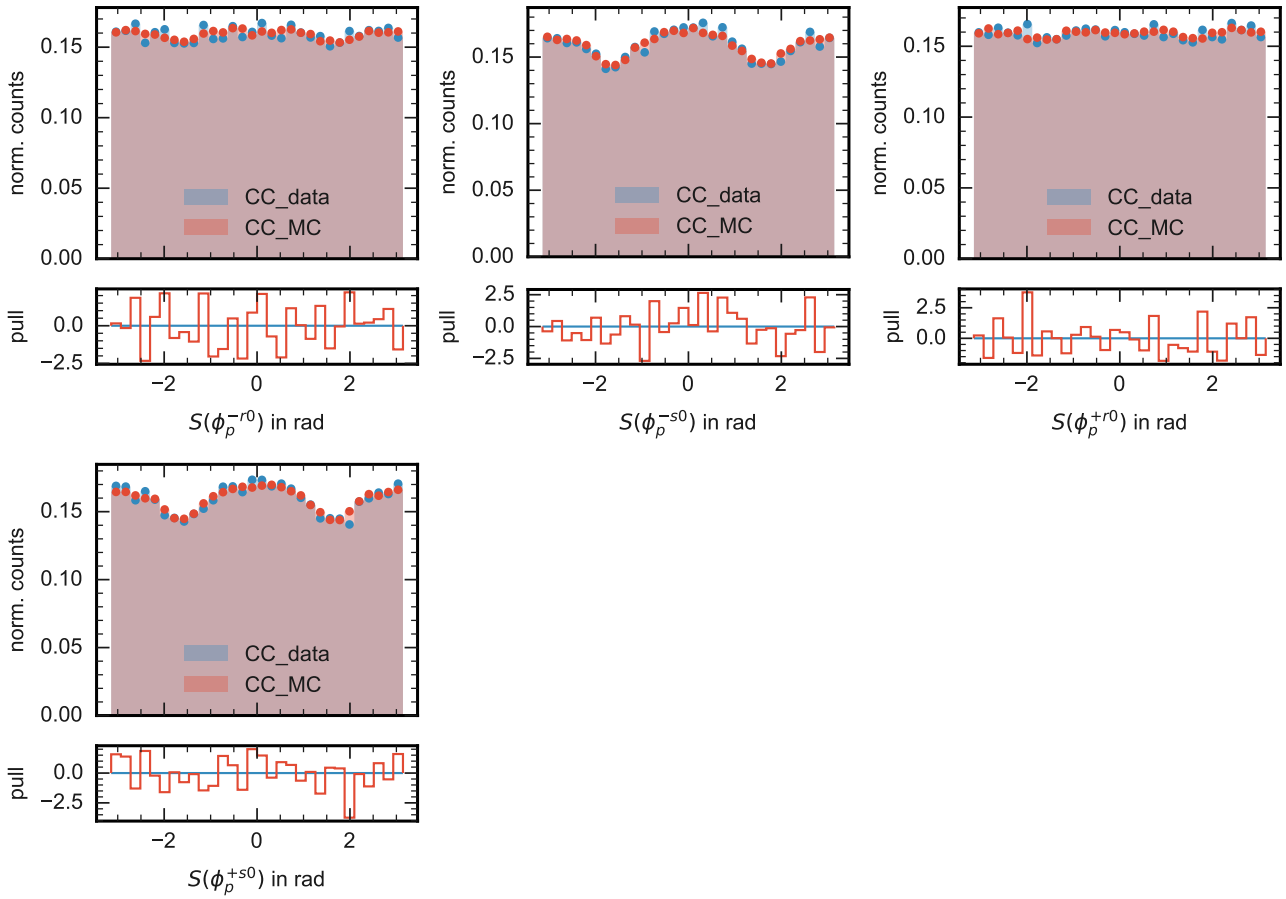



norm. counts
CC_data
CC_MC
pull
$R(\cos(\theta_d^{-r0}))$
$R(\cos(\theta_d^{-s0}))$
$R(\cos(\theta_d^{+r0}))$
$R(\cos(\theta_d^{+s0}))$
$R(\phi_d^{-r0})$ in rad
$R(\phi_d^{-s0})$ in rad
$R(\phi_d^{+r0})$ in rad
$R(\phi_d^{+s0})$ in rad
$R(\phi_p^{0r0})$ in rad
$R(\phi_p^{0r1})$ in rad
$R(\phi_p^{0s0})$ in rad
$R(\phi_p^{0s1})$ in rad


norm. counts
CC_data
CC_MC
pull
$R(\phi_p^{-r0})$ in rad
$R(\phi_p^{-s0})$ in rad
$R(\phi_p^{+r0})$ in rad
$R(\phi_p^{+s0})$ in rad
$T(S(\cos(\theta_p^{0r0})))$
$T(S(\cos(\theta_p^{0r1})))$
$T(S(\cos(\theta_p^{0s0})))$
$T(S(\cos(\theta_p^{0s1})))$
$T(S(\cos(\theta_p^{-r0})))$
$T(S(\cos(\theta_p^{-s0})))$
$T(S(\cos(\theta_p^{+r0})))$
$T(S(\cos(\theta_p^{+s0})))$


norm. counts
CC_data
CC_MC
pull
$S(\cos(\theta_d^{-r0}))$
$S(\cos(\theta_d^{-s0}))$
$S(\cos(\theta_d^{+r0}))$
$S(\cos(\theta_d^{+s0}))$
$S(d^{-r0})$ in cm
$S(d^{-s0})$ in cm
$S(d^{+r0})$ in cm
$S(d^{+s0})$ in cm
$S(\phi_d^{-r0})$ in rad
$S(\phi_d^{-s0})$ in rad
$S(\phi_d^{+r0})$ in rad
$S(\phi_d^{+s0})$ in rad


CC_data
CC_MC
norm. counts
pull
$S(p^{0r0})$ in GeV
$S(p^{0r1})$ in GeV
$S(p^{0s0})$ in GeV
$S(p^{0s1})$ in GeV
$S(p^{-r0})$ in GeV
$S(p^{-s0})$ in GeV
$S(p^{+r0})$ in GeV
$S(p^{+s0})$ in GeV
$S(\phi_p^{0r0})$ in rad
$S(\phi_p^{0r1})$ in rad
$S(\phi_p^{0s0})$ in rad
$S(\phi_p^{0s1})$ in rad


norm. counts
CC_data
CC_MC
pull
$S(\phi_p^{-r0})$ in rad
$S(\phi_p^{-s0})$ in rad
$S(\phi_p^{+r0})$ in rad
$S(\phi_p^{+s0})$ in rad

Bibliography

[1] Makoto Kobayashi and Toshihide Maskawa. "CP-Violation in the Renormalizable Theory of Weak Interaction". In: *Progress of Theoretical Physics* 49.2 (Feb. 1973), pp. 652–657. ISSN: 0033-068X. DOI: 10.1143/PTP.49.652. eprint: https://academic.oup.com/ptp/article-pdf/49/2/652/5257692/49-2-652.pdf. URL: https://doi.org/10.1143/PTP.49.652.

[2] B. Efron. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1 (1979), pp. 1–26. DOI: 10.1214/aos/1176344552. URL: https://doi.org/10.1214/aos/1176344552.

[3] Michael Gronau. "A precise sum rule among four B $\to$ Kπ CP asymmetries". In: *Physics Letters B* 627.1 (2005), pp. 82–88. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2005.09.014. URL: https://www.sciencedirect.com/science/article/pii/S0370269305013274.

[4] Tim Gershon and Amarjit Soni. "Null tests of the Standard Model at an International Super B Factory". In: *Journal of Physics G: Nuclear and Particle Physics* 34.3 (Jan. 2007), p. 479. DOI: 10.1088/0954-3899/34/3/006. URL: https://dx.doi.org/10.1088/0954-3899/34/3/006.

[5] K. Nishimura. "The time-of-propagation counter for BelleII". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 639.1 (2011). Proceedings of the Seventh International Workshop on Ring Imaging Cherenkov Detectors, pp. 177–180. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2010.09.164. URL: https://www.sciencedirect.com/science/article/pii/S0168900210021984.

[6] Yukiyoshi Ohnishi et al. "Accelerator design at SuperKEKB". In: *Progress of Theoretical and Experimental Physics* 2013.3 (Mar. 2013), 03A011. ISSN: 2050-3911. DOI: 10.1093/ptep/pts083. eprint: https://academic.oup.com/ptep/article-pdf/2013/3/03A011/4439973/pts083.pdf. URL: https://doi.org/10.1093/ptep/pts083.

[7] A. J. Bevan et al. "The Physics of the B Factories". In: *The European Physical Journal C* 74.11 (Nov. 2014). DOI: 10.1140/epjc/s10052-014-3026-9. URL: https://doi.org/10.1140%2Fepjc%2Fs10052-014-3026-9.

[8] François Chollet et al. *Keras*. https://keras.io. 2015.

[9] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[10] Torben Ferber and Phillip Urquijo. "Overview of the Belle II Physics Generators". Version 2. In: (Jan. 2016). URL: https://docs.belle2.org/record/282/files/BELLE2-NOTE-PH-2015-006-v2.pdf.

[11] G. Louppe, M. Kagan, and K. Cranmer. "Learning to Pivot with Adversarial Networks". In: *ArXiv e-prints* (Nov. 2016). arXiv: 1611.01046 [stat.ML].

[12] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[13] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam". In: *CoRR* abs/1711.05101 (2017). arXiv: 1711.05101. URL: http://arxiv.org/abs/1711.05101.

[14] Dennis Weyland. "Continuum Suppression with Deep Learning techniques for the Belle II Experiment". MA thesis. Karlsruhe Institute of Technology (KIT), 2017. URL: https://publish.etp.kit.edu/record/21416/files/0_EKP-2017-00061.pdf.

[15] Sara Pohl and Christian Kiesling. "Track Reconstruction at the First Level Trigger of the Belle II Experiment". Presented on 11 04 2018. PhD thesis. Munich: Munich, Ludwig-Maximilians-Universität, 2018. URL: https://docs.belle2.org/record/823.

[16] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *CoRR* abs/1907.10902 (2019). arXiv: 1907.10902. URL: http://arxiv.org/abs/1907.10902.

[17] Anton Hawthorne-Gonzalvez and Martin Sevior. "The use of adversaries for optimal neural network training". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 913 (2019), pp. 54–64. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2018.10.043. URL: https://www.sciencedirect.com/science/article/pii/S0168900218313573.

[18] E Kou et al. "The Belle II Physics Book". In: *Progress of Theoretical and Experimental Physics* 2019.12 (Dec. 2019), p. 123C01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptz106. eprint: https://academic.oup.com/ptep/article-pdf/2019/12/123C01/32693980/ptz106.pdf. URL: https://doi.org/10.1093/ptep/ptz106.

[19] Gregor Kasieczka and David Shih. "Robust Jet Classifiers through Distance Correlation". In: *Phys. Rev. Lett.* 125 (12 Sept. 2020). (corresponding code repository: https://github.com/gkasieczka/DisCo), p. 122001. DOI: 10.1103/PhysRevLett.125.122001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.122001.

[20] Zeke Xie, Issei Sato, and Masashi Sugiyama. "Stable Weight Decay Regularization". In: *CoRR* abs/2011.11152 (2020). arXiv: 2011.11152. URL: https://arxiv.org/abs/2011.11152.

[21] Belle II Collaboration et al. *First search for direct CP-violating asymmetry in* $B^0 \to K^0\pi^0$ *decays at Belle II*. 2021. arXiv: 2104.14871 [hep-ex].

[22] S. Hazra, A. B. Kaliyar, and G. B. Mohanty. "Measurements of Branching Fraction and CP Asymmetry in $B^0 \to K^0_S\pi^0$ Decays at Belle II". In: (Feb. 2022). URL: https://docs.belle2.org/record/2780/files/BELLE2-NOTE-PH-2021-053_v2.5.pdf.

[23] R. L. Workman et al. "Review of Particle Physics". In: *PTEP* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.

[24] Belle II Collaboration et al. *Measurement of branching fractions and direct CP asymmetries for* $B \to K\pi$ *and* $B \to \pi\pi$ *decays at Belle II*. 2023. arXiv: 2310.06381 [hep-ex].

[25] Belle II Software Group. *Belle II Software Documentation*. Version light-2309-munchkin, commit f91695aba. 2023. URL: https://software.belle2.org/light-2309-munchkin/sphinx/ (visited on 10/10/2023).

[26] Demin Zhou et al. *Luminosity performance of SuperKEKB*. 2023. arXiv: 2306.02692 [physics.acc-ph].