

Optimizing Neural Networks for Reconstruction of Neutral Pions in Tau Decays with the ATLAS Experiment at CERN

Florian Alexander Sagolla

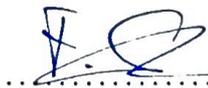
Masterarbeit in Physik
angefertigt im Physikalischen Institut

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
Bonn

April 2023

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.

Bonn, 18.04.2023
Date


Signature

1. Gutachter: Prof. Dr. Klaus Desch
2. Gutachter: Prof. Dr. Ian Brock

Abstract

Since the development of the Standard Model of Particle Physics in the second half of the 20th century, experimental physicists have put great effort into probing the theory for consistency. Even though our universe shows clear signs for insufficiencies in the Standard Model, all experimental results have been compatible with it so far. The operational start of the Large Hadron Collider (LHC) in 2009 was an essential milestone in the process of validating the Standard Model as it enabled the collaborations ATLAS and CMS to discover the Higgs boson in 2012. Since then, extensive measures have been taken to investigate the newly found particle's properties. The behavior of the Higgs boson under Charge-Parity (CP) is one of them. Significant deviations from the Standard Model value could indicate an explanation for the observed matter-anti matter imbalance in the universe. Nevertheless, the Higgs boson CP has not been measured with sufficient precision to accept or reject the Standard Model hypothesis.

Not only in the context of Higgs CP measurements, Higgs boson decays to τ leptons are interesting, as they allow to probe, how the Higgs boson couples to fermions. However, reconstructing hadronic τ decays is especially challenging. To address this issue, the PanTau algorithm had been developed within ATLAS and served well in the latest CP measurements. It provided the τ decay mode as well as the decay product four-vectors. Lately, the PanTau decay mode reconstruction was significantly outperformed by a deep neural network. This however, did not include a reconstruction of the decay product four-vectors that are essential for Higgs CP measurements.

This thesis studies the updated decay mode reconstruction and an additional neural network is optimized that provides reconstruction of the decay product kinematics in the $1p1n$ decay mode. Here, the essential part is reconstructing the neutral pion. The analysis in different figures of merit of the reconstruction yields a result competitive with the PanTau algorithm, even without any information on the associations between Photon Shots and Neutral PFOs.

The structure of the thesis is as follows: Chapter 1 provides an overview of the Standard Model with an emphasis on the τ lepton and Higgs CP measurements. Chapter 2 explains the ATLAS event reconstruction, followed by an introduction into the used machine learning techniques. The PanTau algorithm and its performance are described in Chapter 4 and the neural network decay mode reconstruction is investigated in Chapter 5. Subsequently, a neural network for reconstruction of the neutral pion in the $1p1n$ decay mode is optimized. Finally, the reconstruction is assessed in a comparison between Monte Carlo data and experimental data.

Contents

1	The Standard Model of Particle Physics	1
1.1	The τ Lepton	2
1.2	Electroweak Unification and the Higgs Mechanism	3
1.3	Shortcomings of the Standard Model	4
1.4	Towards an Improved Higgs CP Measurement	4
2	The Large Hadron Collider (LHC) and the ATLAS Detector	7
2.1	The LHC	7
2.2	The ATLAS Coordinate System	7
2.3	Components of the ATLAS Detector	8
2.4	Particle Reconstruction	11
2.5	Event Generation and Simulation	12
3	Machine Learning Techniques	13
3.1	Fundamental Concepts	13
3.2	The Loss Function	14
3.3	Gradient-Based Learning	14
3.4	Artificial Neural Networks (ANN)	15
3.5	Comparison between Deep Learning and Classical ML Approaches	18
3.6	Hyperparameter Optimization	19
4	Reconstruction of τ Leptons with the PanTau Algorithm	20
4.1	Identification of τ Leptons	20
4.2	CellBased Particle Flow Algorithm	21
4.3	Decay Mode Reconstruction	21
4.4	Prediction of the Decay Product Kinematics	22
5	Decay Mode Reconstruction with a Recurrent Neural Network (RNN)	30
5.1	Input Data	30
5.2	RNN Architecture	32
5.3	Preprocessing and Training	32
5.4	RNN Classification Performance	33
5.5	Correction Using the Number of Charged Tracks	35

6	Reconstruction of the Neutral Pion in the 1p1n Decay Mode with a Neural Network	38
6.1	Setup	38
6.2	Performance of the Recurrent Neural Network	39
6.3	Target Transformation and Batch Normalization	39
6.4	Implementing the DeepSet NN	42
6.5	Assessing the Loss Function	42
6.6	Hyperparameter Optimization	45
6.7	Chapter Summary: Links between Photon Shots and Neutral PFOs	49
7	Validation on LHC Data	52
7.1	Cut Selection	52
7.2	Overall modelling	53
7.3	Observables Based on the Neural Network Output	56
7.4	Modelling of Exclusive Decay Modes	57
8	Conclusion	60
A	Neutral Pion Reconstruction: Additional Figures	61
A.1	PanTau algorithm performance compared to the DeepSet NN with Optimized Hyperparameters	62
A.2	Learning Curves for All Calls of the Hyperparameter Optimization	112
B	Validation on LHC Data: Additional Figures	117
B.1	Decay Mode: All	118
B.2	Decay Mode: 1p0n	121
B.3	Decay Mode: 1p1n	124
B.4	Decay Mode: 1pXn	127
B.5	Decay Mode: 3p0n	130
B.6	Decay Mode: 3pXn	133
C	Full Workflow Chart of the PanTau Algorithm	136
	Bibliography	137
	List of Figures	142
	List of Tables	150
	Acknowledgements	151

The Standard Model of Particle Physics

The Standard Model describes the fundamental particles and their interactions among each other to the highest-known level. Developed in the second half of the 20th century, it has been tested numerous times without finding significant deviations from its predictions. The Standard Model is summarized in Fig. 1.1.

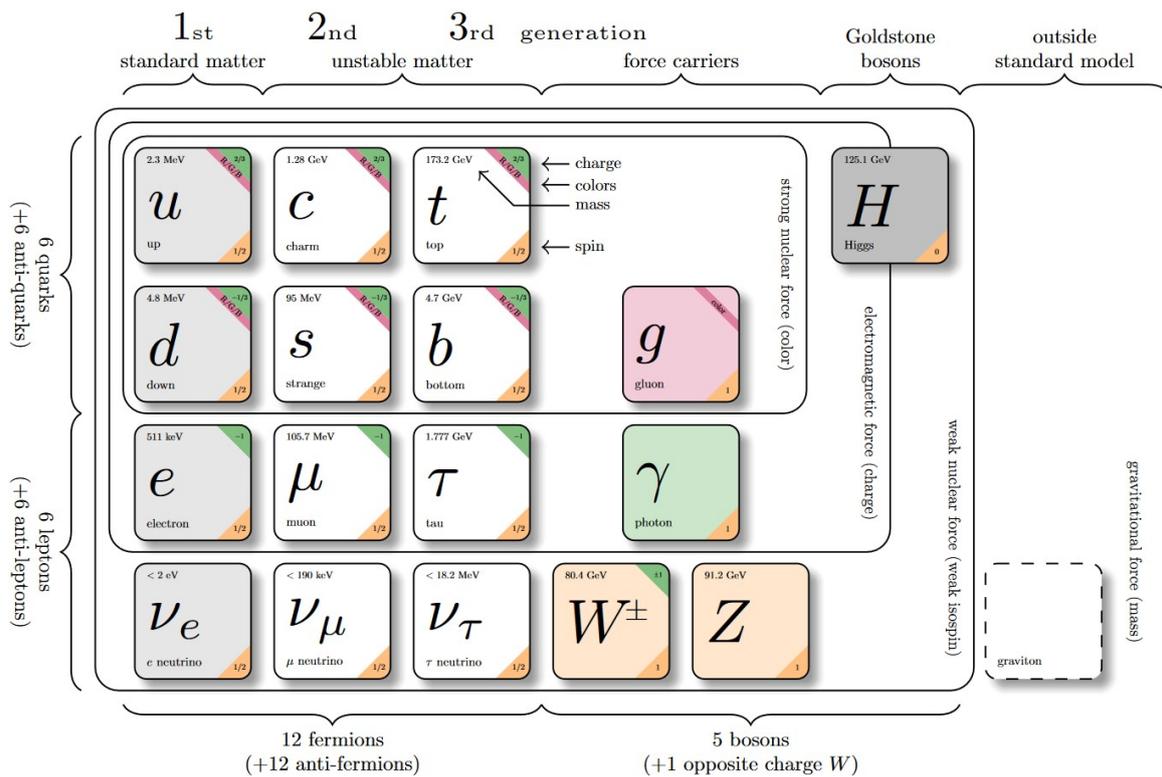


Figure 1.1: Overview of the Standard Model of particle physics [1].

One can distinguish between fermions and bosons. Fermions have spin- $\frac{1}{2}$ and can be further divided into quarks and leptons. Quarks make up the constituents of hadrons and have either a charge of $+\frac{2}{3}$ or $-\frac{1}{3}$. Leptons have charge ± 1 , except for the neutrinos, which are neutral. Both, hadrons and leptons,

can be divided into three generations which, due to lepton universality, only differ in their mass. For each particle, a corresponding anti-particle with opposite charge exists. Interactions between fermions are described by the five bosons. The photon mediates the electromagnetic interaction. It is massless and does not carry electric charge. For the Gluon, which mediates the strong interaction, the same accounts. However, it does carry color charge. The W^\pm and Z^0 bosons mediate the weak interaction and carry either no charge (Z^0) or ± 1 (W^\pm). Their respective rest masses are (91.1876 ± 0.0021) GeV and (80.377 ± 0.012) GeV [2]. The photon, the gluon and the W and Z bosons all have spin-1. The Higgs boson is the only particle in the Standard Model with spin-0. Its rest mass is (125.25 ± 0.17) GeV [2].

1.1 The τ Lepton

The τ lepton has a mass of (1776.86 ± 0.12) MeV and is therefore the heaviest lepton. It was discovered in 1975 at the Stanford Linear Accelerator Center (SLAC) [3] and is particularly interesting as it can decay to leptons and hadrons due to its high mass. The lowest order Feynman diagram of the τ lepton decay can be seen in Fig. 1.2. After a mean lifetime of only (290.3 ± 0.5) s⁻¹⁵ it decays to a τ neutrino

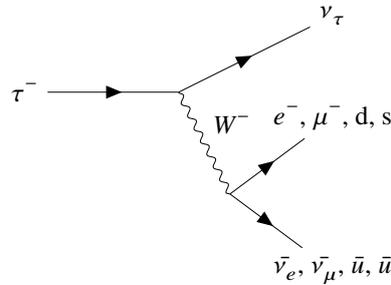


Figure 1.2: Lowest order Feynman diagram of a τ lepton decay.

and a W boson first. Subsequently, the W boson decays either to a light lepton and the corresponding neutrino (leptonic decay) or to a light quark and its anti-quark (hadronic decay). In this analysis, in the dominant hadronic decay modes are particularly interesting. They are listed in Table 1.1 with their corresponding branching ratio predicted by the Standard Model and can be distinguished by their number of charged hadrons (either pions or kaons) and neutral pions. It makes sense to assign labels to the

Table 1.1: Notation for the dominant hadronic τ lepton decay modes with the corresponding branching ratios \mathcal{B} . h^\pm indicates a charged hadron. Neutrinos are omitted [4].

Decay mode	Label	\mathcal{B} (w.r.t total τ decays)
h^\pm	1p0n	11.5%
$h^\pm \pi^0$	1p1n	25.9%
$h^\pm \geq 2\pi^0$	1pXn	10.6%
$3h^\pm$	1p0n	9.46%
$3h^\pm \geq 1\pi^0$	3pXn	5.09%

different decay modes accordingly (e.g. 1p1n denotes $h^\pm \pi^0$). As the neutrino cannot be detected with

the ATLAS detector (see Chapter 2), it is omitted. The neutral pions from the τ decay further decay into two photons (98.823 ± 0.034) % of the time. The photons can further decay via pair production into e^+e^- [2].

1.2 Electroweak Unification and the Higgs Mechanism

There has always been the goal of unifying all forces into one theory. Therefore, a main achievement was the electroweak unification developed by Glashow, Salam and Weinberg (GSW), who received the Nobel Prize 1979 for their work [5, 6]. The theory of the weak interaction is represented by an $SU(2)_L$ group. The index L denotes exclusive coupling to left-handed particles or right-handed anti-particles. To be self-consistent, the theory would require gauge invariance. Accordingly, three gauge fields are introduced for the weak interaction: $W^{(1)}$, $W^{(2)}$ and $W^{(3)}$. $W^{(1)}$ and $W^{(2)}$ can be combined to the charged W bosons:

$$W_\mu^\pm = \frac{1}{\sqrt{2}} \left(W_\mu^1 \mp iW_\mu^2 \right) \quad (1.1)$$

On the other hand, the Z^0 boson as mediator of neutral currents cannot be associated with the $W^{(3)}$ as experiments showed that the Z^0 couples to both, left and right-handed particles. According to Quantum Electrodynamics (QED), the photon can serve as a neutral boson. With this, a local $U(1)_Y$ gauge symmetry can be introduced. The index Y is the coupling of B_μ , the new gauge field to the weak hypercharge. The GSW theory states that the photon field A_μ as well as the Z_μ field are a mixture of $W_\mu^{(3)}$ and B_μ :

$$A_\mu = +B_\mu \cos(\theta_W) + W_\mu^{(3)} \sin(\theta_W) \quad (1.2)$$

$$Z_\mu = -B_\mu \sin(\theta_W) + W_\mu^{(3)} \cos(\theta_W) \quad (1.3)$$

θ_W denotes the weak mixing angle [7].

So far, these bosons are massless, which was not experienced in experiments. The Higgs mechanism developed by R. Brought, F. Englert and P. Higgs was proposed in 1964 [8–10]. It solves this problem by introducing a scalar isospin doublet Φ :

$$\Phi = \begin{pmatrix} \Phi^+ \\ \Phi^0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \Phi_1^+ + i\Phi_2^+ \\ \Phi_3^0 + i\Phi_4^0 \end{pmatrix}. \quad (1.4)$$

The Higgs potential is defined by

$$V(\Phi) = \mu^2 \Phi^* \Phi + \lambda (\Phi^* \Phi)^2 \quad (1.5)$$

with $\mu < 0$ and $\lambda > 0$. As observable in Fig. 1.3, the potential has an infinite number of minima. When choosing one of them, the global symmetry is broken and this field excitation is associated with the Higgs boson [7]. Furthermore, the weak gauge bosons now acquire masses. Same accounts for the fermions.

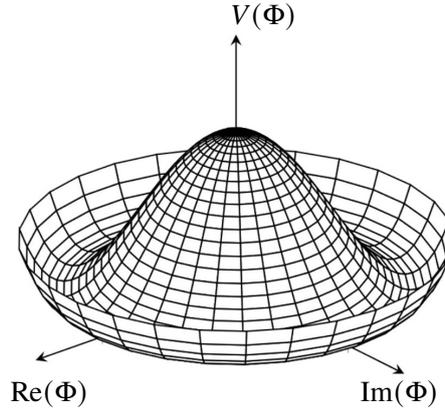


Figure 1.3: Sketch of the Higgs potential $V(\Phi)$ from Eq. (1.5) adapted from [11]. $\text{Re}(\Phi)$ and $\text{Im}(\Phi)$ indicate real and imaginary part.

1.3 Shortcomings of the Standard Model

Even though the Standard Model very accurately describes the physics observed in experiments so far, there are still a number of shortcomings that require an extension of the model. The one most relevant for this analysis is an insufficient source for CP violation in the Standard Model. Under CP transformations charge (C) and parity (P) are inverted, which means that all particles are transformed into their corresponding anti-particles and the orientation of space is reversed. Hence, left-handed particles are transferred to right-handed anti-particles and vice versa. The sources of CP violation that were found beyond that were not sufficient to explain the strong matter-antimatter asymmetry that can be observed in the universe.

Another shortcoming of the Standard Model is its inability to explain Dark Matter. When exploring the movement of galaxies inside a cluster, it was observed that the mass of a galaxy must be higher than the sum of masses of the luminous stars it contains, which leads to the assumption that there must be another source for gravitational force [12]. This cannot be explained by the Standard Model. In addition, the Standard Model does not describe the gravitational force. Gravitation continues to be described with General Relativity. A way to unite the two theories has not been found yet. A further shortcoming is the so-called *hierarchy problem*. The large discrepancy between the energy scales of the gravitational force and the electroweak force leads to large Higgs masses at high energy scales. The Higgs mass would have to be fine-tuned at high precision for the Standard Model to be applicable with these high mass scales. Lastly, the Standard Model predicts Neutrinos to be massless. However, it has been observed that Neutrinos have a mass difference, which implies that at least two of them do have a mass.

1.4 Towards an Improved Higgs CP Measurement

The Higgs boson predicted by the Standard Model predicts one purely CP -even Higgs Boson. If this is not the case, it could be an indicator for the missing source of CP violation mentioned in the previous section. Therefore, it is of high interest to measure the Higgs CP precisely. When considering $H \rightarrow \tau\tau$ decays, CP violation is described by the single mixing angle parameter $\phi_\tau \in [-90^\circ, 90^\circ]$ in

the generalized Yukawa coupling term in the Lagrangian

$$\mathcal{L}_Y = -\frac{m_\tau}{\nu} \kappa_\tau (\cos \phi_\tau \bar{\tau} \tau + \sin \phi_\tau \bar{\tau} i \gamma_5 \tau) h \quad (1.6)$$

with m_τ denoting the τ lepton mass, κ_τ the reduced Yukawa coupling strength and ν the Standard Model VEV [13]. A value of $\phi_\tau = 0^\circ$ would realize the Standard Model hypothesis; $\phi_\tau = \pm 90^\circ$ would realize the pure CP-odd hypothesis. All other values of ϕ_τ would indicate a mixing of CP-even and odd, which would violate CP. The ATLAS and CMS collaborations have put large efforts into measuring this parameter. The results are compatible with the Standard Model, but a CP-odd contribution could not be excluded by either of them. The ATLAS result $\phi_\tau = 9^\circ \pm 16^\circ$ rejected the CP-odd hypothesis by 3.4 standard deviations, while the CMS value $\phi_\tau = 4^\circ \pm 17^\circ$ disfavored CP-odd by 3.2 standard deviations [14, 15].

The angle ϕ_τ can be measured by exploiting the correlation between the τ leptons' transverse spin components. For different decay modes of the τ lepton, different approaches can be used to receive ϕ_τ . The $1p1n$ decay mode is well suited due to its high cross-section. For this decay mode, the ρ decay plane method was developed in [16–18]: The signed acoplanarity angle between the two τ decay planes φ_{CP}^* (Fig. 1.4) is measured in order to compare its distribution to the expected distributions for a CP-even and CP-odd Higgs. In order to extract φ_{CP}^* , the angular directions of the two pions from

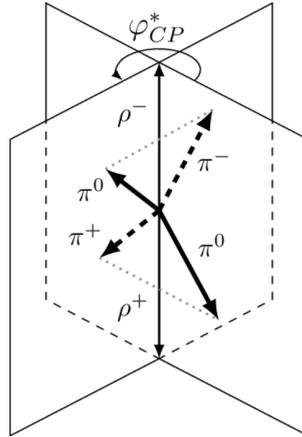


Figure 1.4: Graphical display of the decay planes of a di- τ decay via the ρ resonance [14].

the ρ decay have to be measured. Another observable that needs to be considered is the energy difference between the two pions from the τ decay:

$$Y_\pm = \frac{E(\pi^\pm) - E(\pi^0)}{E(\pi^\pm) + E(\pi^0)} \quad (1.7)$$

When splitting the phase space in $Y_+ Y_- > 0$ and $Y_+ Y_- < 0$, one receives the distributions for φ_{CP}^* . The distributions predicted by the Standard Model are shown in 1.5. Any measured phase shift from these distributions would result in a non-zero value for ϕ_τ .

Obviously, a higher angular resolution and energy resolution directly leads to an improved Higgs CP measurement. This thesis aims to improve reconstruction of the τ decay products in order to achieve this.

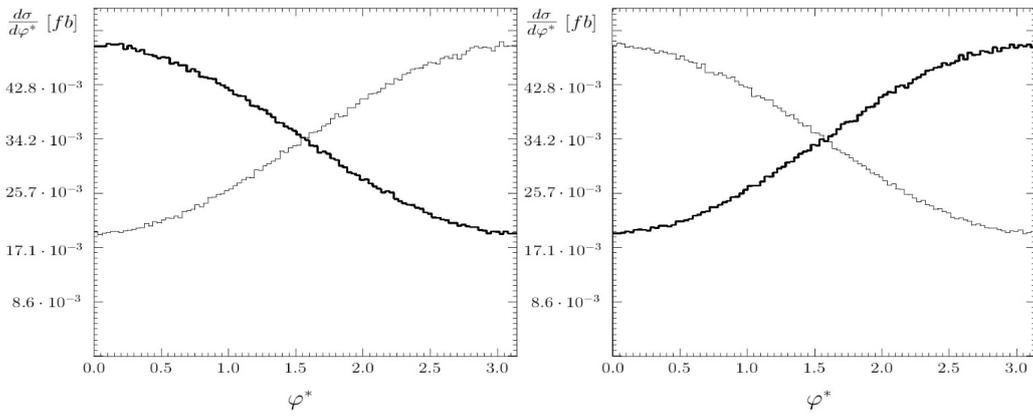


Figure 1.5: Dependence of the differential cross-section on φ_{CP}^* for $Y_+Y_- > 0$ (left) and $Y_+Y_- < 0$ (right) [16].

The Large Hadron Collider (LHC) and the ATLAS Detector

In the following, the experimental design, the process of data taking as well as reconstruction and simulation of particle collisions is described. For further reading, consider [19].

2.1 The LHC

The LHC is a particle collider located at CERN laboratory in Geneva, Switzerland. With a circumference of approximately 27 km, the LHC is the largest particle collider in the world. The LHC started operating in 2009 and accelerates either protons or lead ions. In this thesis, proton-proton collisions from LHC run 2 (2015-2018) are considered. During this data taking period, a center of mass energy of 13 TeV was reached. In order to force collisions, the protons are extracted from hydrogen atoms, pre-accelerated and fed into the circular LHC beam pipes. Here, two proton beams, each consisting of *bunches* of 10^{11} particles separated by 25 ns, run in opposite directions. At the four main experiments, ATLAS, CMS, LHCb and ALICE, the two beams are forced to collide and data is recorded.

2.2 The ATLAS Coordinate System

Position and motion of a particle in the ATLAS detector is described in the right-handed coordinate system illustrated in Fig. 2.1. Its center is the nominal interaction point. The positive x -axis points towards the accelerator ring's center, while the y -axis points upwards. The z -axis points along the beam pipe. The azimuthal angle ϕ is the angular distance between a particle's momentum vector and the x -axis in the x - y -plane. The polar angle θ denotes the angle between the z -axis and the momentum vector of the particle. As this observable is not Lorentz invariant, the pseudorapidity $\eta = -\ln \tan(\theta/2)$ is mostly used. The transverse momentum p_T is the component of the momentum in the x - y -plane. The angular distance is defined as

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}. \quad (2.1)$$

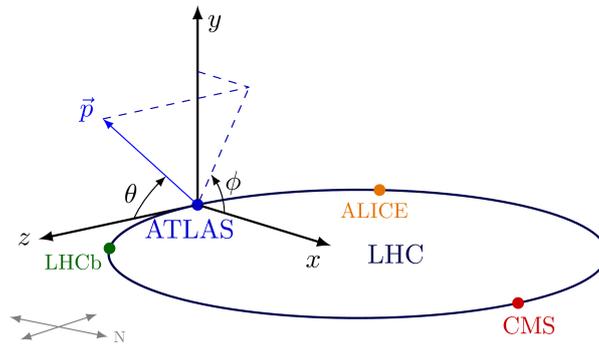


Figure 2.1: Sketch of the LHC and the ATLAS coordinate system. The other three major experiments at LHC, namely ALICE, CMS and LHCb are depicted as well [20].

2.3 Components of the ATLAS Detector

The ATLAS (A Toroidal LHC ApparatuS) experiment is one of the four major experiments at the LHC. In this section, the main components of the ATLAS detector will be described with an emphasis on the parts that enable the detection of τ leptons. The full detector can be seen in Fig. 2.2. With a length of 44 m and a diameter of 25 m it is the largest volume particle detector ever constructed.

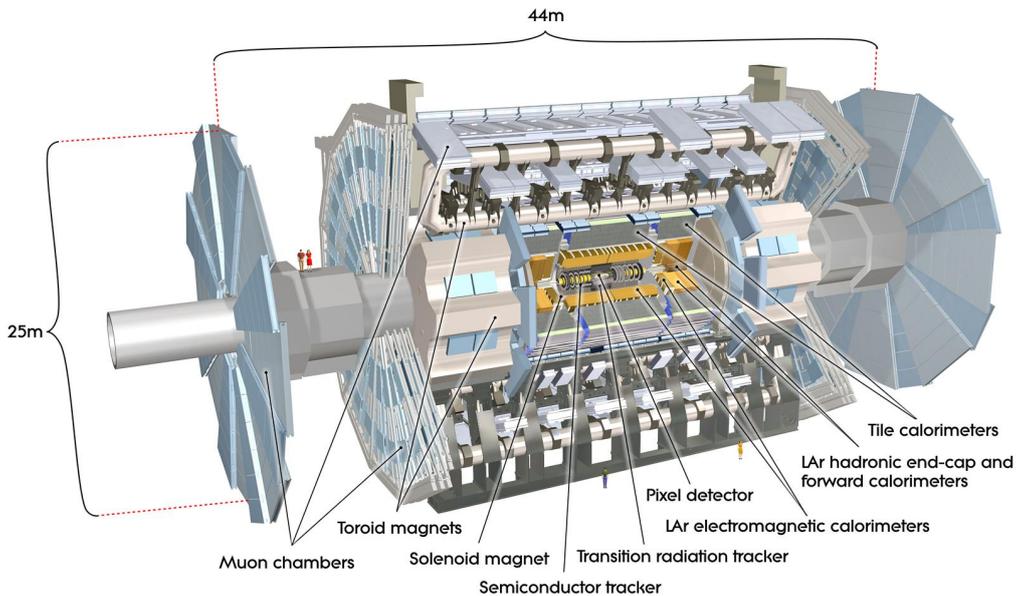


Figure 2.2: Sketch of the ATLAS detector [21].

2.3.1 The Inner Detector

In the Inner Detector, the trajectory of charged particles is determined. A magnetic field generated by a solenoid bends these trajectories to enable charge and momentum measurement. The Inner Detector

consists of three layers, represented in Fig. 2.3. The layer closest to the beam axis is the Pixel Detector

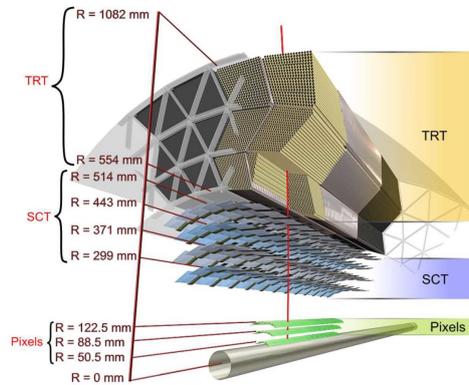


Figure 2.3: Sketch of the components of the ATLAS Inner Detector [19].

followed by the Semiconductor Tracker (SCT) modules. They enable precise vertex measurements within a range of $|\eta| < 2.5$. The outermost part is the Transition Radiation Tracker (TRT) that covers $|\eta| < 2.0$. It plays a main role in momentum measurement and particle identification.

2.3.2 The Calorimeter System

The calorimeter system is responsible for energy measurements especially for neutral particles as they do not leave a track in the Inner Detector. It consists of the electromagnetic (ECAL) and the hadronic calorimeter (HCAL) as displayed in Fig. 2.4.

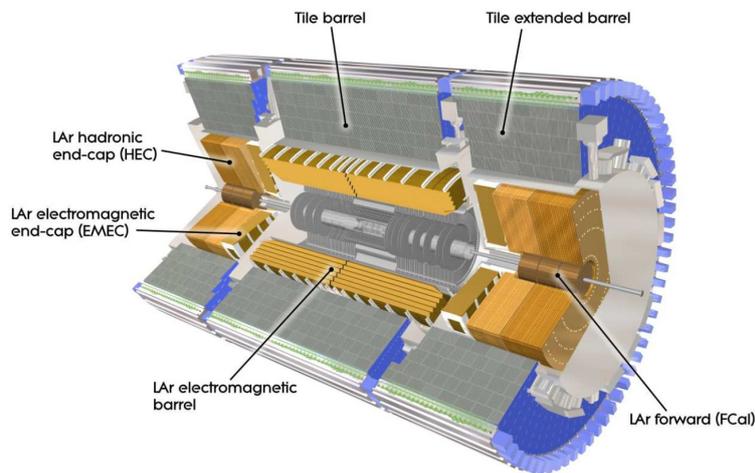


Figure 2.4: Sketch of the calorimeter system of the ATLAS detector [19].

The Electromagnetic Calorimeter

The electromagnetic calorimeter (ECAL) is a liquid argon (LAr) sampling calorimeter that is responsible for the energy measurement of electrons and photons. It is divided into a barrel and the end-caps. This way, the ECAL covers the regions $|\eta| < 1.3$ and $1.5 < |\eta| < 2.5$. As represented in Fig. 2.5, the ECAL consists of three layers. The innermost part is the strip layer (EM1), which was designed specifically to be able to separate closely located photons that originated from neutral pion decays as mentioned in Section 1.1. The second layer (EM2) is responsible for the most part of an electromagnetic shower, while the third layer (EM3) contains the rest of the shower [22].

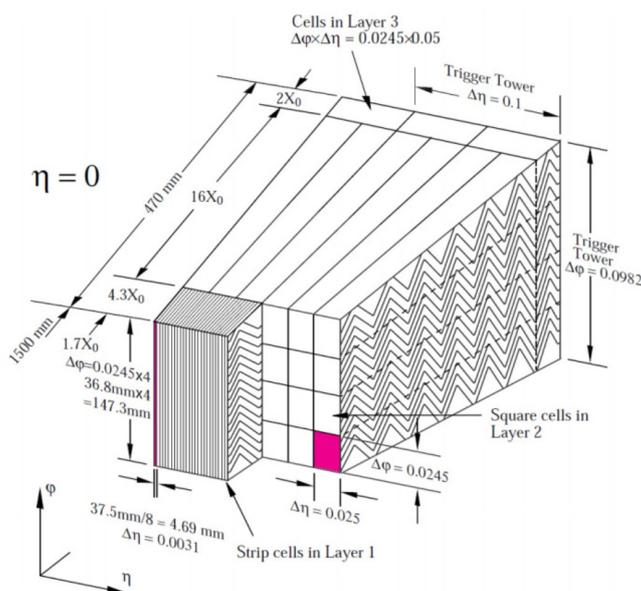


Figure 2.5: Sketch of the electromagnetic calorimeter of the ATLAS detector [19].

The Hadronic Calorimeter

In this part of the detector, the energies of hadrons that were not stopped in the ECAL are measured. Again, the detector is divided in a barrel ($|\eta| < 1.7$) and an end-cap region ($1.5 < |\eta| < 3.2$).

2.3.3 The Muon System

The Muon System is the outermost part of the detector and is responsible for measuring the tracks and momenta of muons.

2.3.4 The Trigger System

In LCH run 2, a bunch crossing rate of 40 MHz was reached, which corresponds to the aforementioned bunch-spacing of 25 ns. At this rate, it is not possible to record all data measured by the detector, so two trigger levels are imposed. The Level-1 (L1) trigger considers a subset of the detector information to identify events containing physics worth analyzing. In this process, specific signatures, like the one

of muon are recognized. The L1 reduces the data frequency to approximately 100 kHz. Furthermore, it defines *regions of interest*. The High Level Trigger (HLT) exploits this information to refine the reconstruction. Selection cuts are added. This decreases the data taking frequency to around 100 kHz, which can be used for offline physics analyses [23].

2.4 Particle Reconstruction

During data processing, the detector responses of the different detector components have to be mapped to physics particles. Most of the particles mentioned here build, together with additional detector information, the basis for the reconstruction of hadronically decaying τ leptons that will be explained and analyzed in the next chapters.

Electrons and Photons

Electrons and photons leave similar signatures in the ECAL. The charged electron, however, leaves an additional track in the Inner Detector. Therefore, both are reconstructed from energy deposits in the ECAL using the Sliding Window Algorithm [24]. If an additional track pointing to the ECAL cluster is measured by the tracking system, the particle is identified with an electron, otherwise a photon.

For the reconstruction of τ leptons, two more extensions to this algorithm are essential: The measurement of two matched tracks associated with an ECAL cluster is identified with a $\gamma \rightarrow e^+e^-$ conversion (Conversion Track). Additionally, Photon Shots are local maxima in the ECAL strip layer as explained in Section 2.3.2.

Jets

Jets are collimated sprays of particles produced by quarks or gluons. They are reconstructed using the anti- k_r algorithm. The algorithm exploits clusters in the HCAL that exceed noise level [25]. In τ reconstruction, it is essential to separate ordinary jets from τ candidates, as will be explained in Chapter 4.

Muons

Muons only interact minimally with most components of the ATLAS detector. They reach the Muon System, where most of them are stopped. Together with the Inner Detector, trajectory, momentum and energy can be measure [26]. As muons can originate from leptonic τ decays, and because of their straight forward identification, they can serve well as trigger for decays to $Z\tau_\mu\tau_{had}$, that will be analyzed in later chapters.

Missing Transverse Energy

To determine the missing transverse energy, momentum conservation in the transverse plane can be utilized:

$$E_T^{\text{miss}} = \sum_{\text{selected electrons}} p_T^e + \sum_{\text{accepted photons}} p_T^\gamma + \sum_{\text{accepted } \tau \text{ leptons}} p_T^{\tau_{had}} + \sum_{\text{selected muons}} p_T^\mu + \sum_{\text{accepted jets}} p_T^{\text{jets}} + \sum_{\text{unused tracks}} p_T^{\text{track}} \quad (2.2)$$

This is possible as the colliding protons initially only have a momentum component parallel to the z -axis. Missing transverse energy can result from weakly interacting particles, like neutrinos, or from particles outside the detector acceptance [27]. As there is always a neutrino involved in the τ decay, it is highly relevant to reconstruct E_T^{miss} accurately.

Pile-Up

Particle reconstruction is significantly impeded by the fact that per bunch crossing, usually more than one particle collision happens. This results in a contamination of the signals of the detector. Therefore, it is important, to consider the average interactions per bunch crossing.

2.5 Event Generation and Simulation

In order to test the Standard Model, not only the actual data needs to be taken. In addition, simulated data is necessary to be able to compare theory and experimental data. In this process, Monte Carlo generators play an important role. Event generators like Sherpa, Pythia and Powheg are able to randomly generate particle collisions according to the Standard Model [28–30]. This includes a whole chain of steps as illustrated in Fig. 2.6: The hard process is calculated perturbatively on matrix element level (red). Additionally, initial state radiation (ISR) and final state radiation (FSR) during the hard process are considered (blue). In the further decay, parton showers are formed via fragmentation and hadronization (green). Moreover, QED bremsstrahlung during the process (yellow) and the underlying event (purple) are simulated.

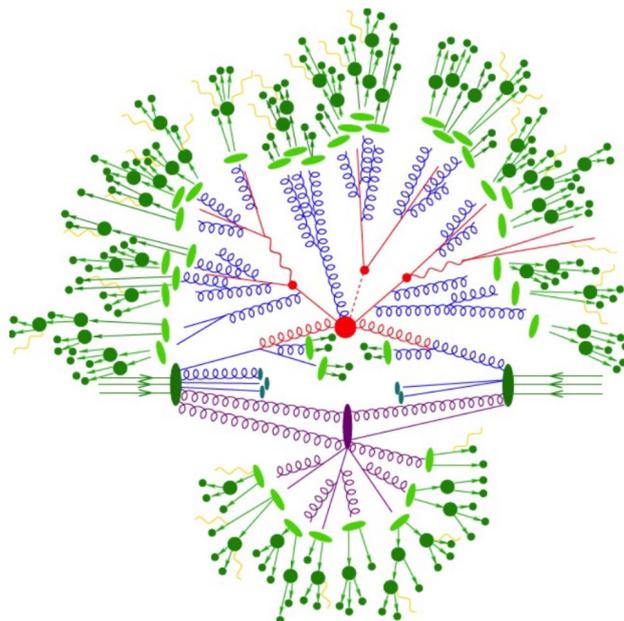


Figure 2.6: Exemplary event generated with the Sherpa Monte Carlo generator. Red: Hard scattering. Blue: ISR and FSR. Green: Fragmentation and hadronization. Yellow: QED bremsstrahlung. Purple: Underlying event [31].

Machine Learning Techniques

The surge of Machine Learning (ML) techniques has influenced large parts of science, business and our daily lives. ML is a subfield of artificial intelligence that focuses on the development of algorithms that enable computer systems to learn and make predictions or decisions without being explicitly programmed. Since the deep learning revolution about ten years ago, it has been possible to train significantly more complex ML models on much larger datasets. With the ever-increasing volume of data produced by particle physics experiments such as the LHC, ML has become a valuable tool for extracting meaningful information from the data [32]. τ lepton reconstruction in the ATLAS experiment is no exception from this: Over the last years, classic ML methods for identification and decay mode classification were replaced by deep learning methods in order to achieve significant advances in performance [33, 34].

In this chapter, the basics of ML will be covered to introduce the concepts applied in the following chapters. More detailed information can be found in [35], [36] and [37].

3.1 Fundamental Concepts

To solve a general ML problem, the steps summarized in Fig. 3.1 are performed repeatedly, until the model reaches acceptable performance. The input data for an ML model contains a number of samples.

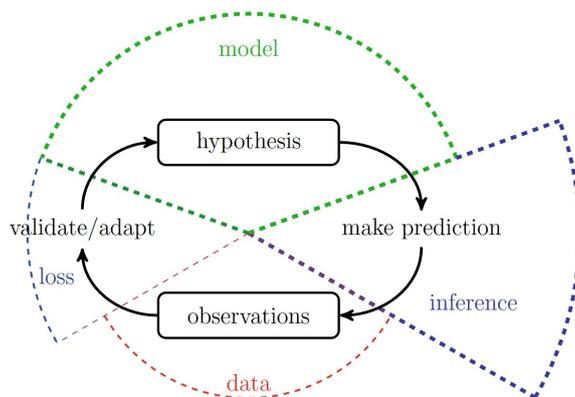


Figure 3.1: Graphical Display of a general machine learning workflow [35].

Each sample contains features $\mathbf{x} \in \mathcal{X}$ and labels $\mathbf{y} \in \mathcal{Y}$. Usually, the number of features m is high and there is only one label ($n = 1$). However, in this analysis, the number of labels varies. The ML algorithm aims to choose a model h from the hypothesis space \mathcal{H} that predicts most accurately the labels from the features. After making an initial choice on h , the chosen model is used to predict $h(\mathbf{x}) = \hat{\mathbf{y}}$ for each sample. The loss function $L((\mathbf{x}, \mathbf{y}), h)$ (see Section 3.2) describes the quality of the prediction and is used to adapt h . As feature and label space vary in their structure and elements from them might be hard to generate (especially the labels), different strategies of ML are used. Besides *supervised learning*, there is *unsupervised learning* and *reinforcement learning*. In supervised learning, which this thesis focuses on, a training dataset with known labels is used to train the model. In unsupervised learning, ML algorithms like *k-means clustering* are trained to find correlations within unlabeled data. Reinforcement learning includes methods that do not require training data, but learn during prediction via trial-and-error. One can further divide the ML problem into classification and regression tasks. In classification tasks, \mathcal{Y} is discrete; in a regression task, it is continuous.

3.2 The Loss Function

As already mentioned, the loss function

$$L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \rightarrow \mathbb{R}_+ : ((\mathbf{x}, \mathbf{y}), h) \mapsto L((\mathbf{x}, \mathbf{y}), h) \quad (3.1)$$

is used to quantify the quality of the model prediction. Depending on the ML problem, different loss functions are feasible. For classification tasks with unordered \mathcal{Y} , *categorical cross-entropy* can be used. It is defined as

$$L((\mathbf{x}, \mathbf{y}), h) = L(\mathbf{y}, \hat{\mathbf{y}}) := - \sum_{i=1}^n y'_i \cdot \log \hat{y}_i \quad (3.2)$$

with \mathbf{y}' one-hot encoded and $\hat{\mathbf{y}}$ containing the predicted probabilities for each possible label. For regression tasks, the *mean squared error* (MSE) loss is widely used:

$$L(\mathbf{y}, \hat{\mathbf{y}}) := \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (3.3)$$

If instances of \mathcal{Y} vary in their importance, one can extend the function to

$$L(\mathbf{y}, \hat{\mathbf{y}}) := \sum_{i=1}^n a_i \cdot (y_i - \hat{y}_i)^2 \quad \text{with} \quad a_i \in \mathbb{R}^n \quad (3.4)$$

and assign large a_i to labels with higher relevance.

3.3 Gradient-Based Learning

In the context of deep learning, gradient-based methods became essential. They require each hypothesis $h^{(\mathbf{w})} \in \mathcal{H}$ to be characterized by a specific weight vector $\mathbf{w} \in \mathbb{R}^n$. In addition, the loss function has to depend smoothly on \mathbf{w} . For simplicity, we define the loss function as $L(\mathbf{w})$. Accordingly, the ML

problem becomes

$$\min_{\mathbf{w} \in \mathbb{R}^n} L(\mathbf{w}). \quad (3.5)$$

As $L(\mathbf{w})$ is smooth, its gradient $\Delta L(\mathbf{w})$ (in a sufficiently small neighborhood) around the current guess $\mathbf{w}^{(r)}$ can be determined. Accordingly, the *gradient descent step* can be performed by evaluating

$$\mathbf{w}^{(r+1)} = \mathbf{w}^{(r)} - \alpha \Delta L(\mathbf{w}^{(r)}) \quad (3.6)$$

with $\alpha > 0$ denoting the *learning rate*. This step can be performed repeatedly until a minimum in $L(\mathbf{w})$ is reached. The algorithm is named *gradient descent*. This however, only works, if the learning rate is appropriately chosen. Too large α may lead to an oscillation around the minimum and too small α may lead to unfeasible training durations. This can be tackled by using the *stochastic gradient descent* (SGD) including a variable learning rate. Furthermore, in this algorithm the local derivative is approximated by different randomization strategies. The *adam* optimizer facilitates this as described in more detail in [38]. Helpful for better convergence is standardizing the features to make computations more efficient. Smoother convergence can be achieved by passing not only one sample, but several at a time into the algorithm. The *batch size* is the number of samples per step in this *mini-batch* SGD. An *epoch* is finished after all samples have been considered.

To investigate if the training was successful, it is worth considering the development of the loss during training. In order to avoid exploiting statistical effects in the training sample, a separate validation dataset is defined that is independent of the training data. Now, the *training error* (total error on the training dataset) and the *validation error* (total error on the validation dataset) can be observed against the number of epochs. This relation is known as the *learning curve*. If both errors are still decreasing at the end of the training it is a sign over *underfitting*. If the validation error reached its minimum, but the training loss is still decreasing, it is a sign of *overfitting*.

3.4 Artificial Neural Networks (ANN)

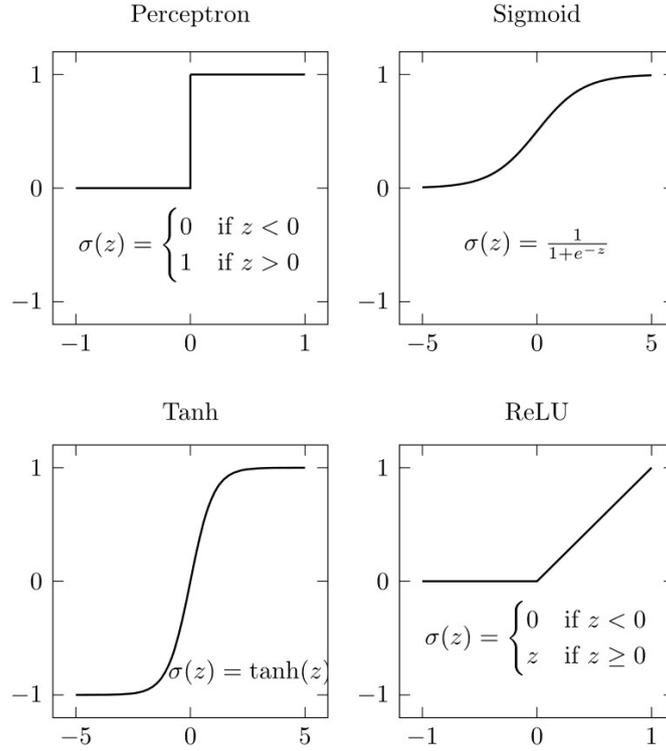
ANNs are ML models inspired by the biological neural networks of the human brain. They consist of interconnected nodes that are organized into layers and are capable of modeling complex non-linear relationships between inputs and outputs with high performance. We speak of *deep learning* for large numbers of layers.

3.4.1 Feed Forward Neural Networks

The most basic type of ANN is the *feed forward neural network*. The output of a single node, called *perceptron* is

$$a(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + b) = \sigma(z) \quad (3.7)$$

with w being the weight vector, x the input and b the bias. The *activation function* σ can have different shapes (Fig. 3.2) and can be used to implement non-linearity into the ANN. The final output of the ANN


 Figure 3.2: Graphical Display of different activation functions $\sigma(z)$ [39].

can be calculated via *forward propagation* using the relation

$$a_j^{(l)} = \sigma \left(\sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)} \right) \quad (3.8)$$

where $a_k^{(l-1)}$ denotes the output of the k^{th} neuron from the previous layer and $w_{jk}^{(l)}$ denotes the weight of the k^{th} neuron to the current neuron j . An example feed forward neural network can be observed in Fig. 3.3. Its input vector has size three, the two intermediate layers (*hidden layers*) have four nodes each and the output has size two.

As the criteria for gradient-based learning mentioned in Section 3.3 are fulfilled for ANNs, the concept of gradient descent can be applied. From Eq. (3.6) follows that

$$\Delta L(\mathbf{w}) = \frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}}. \quad (3.9)$$

needs to be determined. The single terms can be simplified to

$$\frac{\partial L}{\partial a_j} = \sum_{l=1}^N \frac{\partial L}{\partial a_l} \frac{\partial a_l}{\partial z_j} \frac{\partial z_l}{\partial a_j} = \sum_{l=1}^N \delta_j w_{jl}, \quad \frac{\partial a_j}{\partial z_j} = \sigma'(z_j), \quad \text{and} \quad \frac{\partial z_j}{\partial w_{ij}} = a_i. \quad (3.10)$$

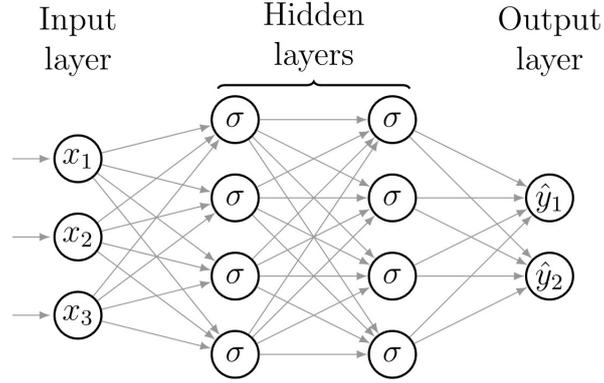


Figure 3.3: Graphical Display of a feed forward neural network with x_i being the feature input, σ the activation function and \hat{y}_i the output [39].

Starting at the output layer, δ_j can be computed recursively with high efficiency. This process is known as *back propagation*.

A problem when training ANNs is often that the input distribution for each layer shifts during training. This can limit training speed and convergence. One can normalize the layer inputs via *batch normalization* to address this issue [40].

3.4.2 Recurrent Neural Networks (RNN)

The *simple recurrent neural network* (sRNN) introduces feedback loops into the structure of the feed forward neural network. Therefore, it is able to process not only vectors, but ordered sequences of vectors. This is achieved by adapting Eq. (3.7) to

$$\mathbf{s}_t(\mathbf{x}_t, \mathbf{s}_{t-1}) = g_t(\mathbf{V} \cdot \sigma_t(\mathbf{W}\mathbf{s}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}) + c) \quad \text{with} \quad t = 0, \dots, N. \quad (3.11)$$

Here, \mathbf{U} , \mathbf{V} and \mathbf{W} are weight matrices, \mathbf{x} is the input vector, \mathbf{s} the internal state and N the length of the sequence. g is a non-linear activation function. A sketch of the sRNN node is displayed in Fig. 3.4. Adjusting the weights is done similarly to a feed forward neural network by performing *back propagation through time* [42].

A common problem with sRNNs is that with longer sequences, the gradients often vanish or explode. This can be tackled by considering a *long short-term memory* (LSTM) RNN, which is considered the “dominant workhorse in sequence processing” [42]. This *gated* RNN introduces the input gate i_t , the forget gate f_t and the output gate o_t via

$$\tilde{c}_t = \sigma(\mathbf{W}_c \mathbf{s}_{t-1} + \mathbf{U}_c \mathbf{x}_t + b_c) \quad (3.12)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (3.13)$$

$$s_t = o_t \cdot \sigma(c_t). \quad (3.14)$$

It effectively avoids the problem of vanishing or exploding gradients.

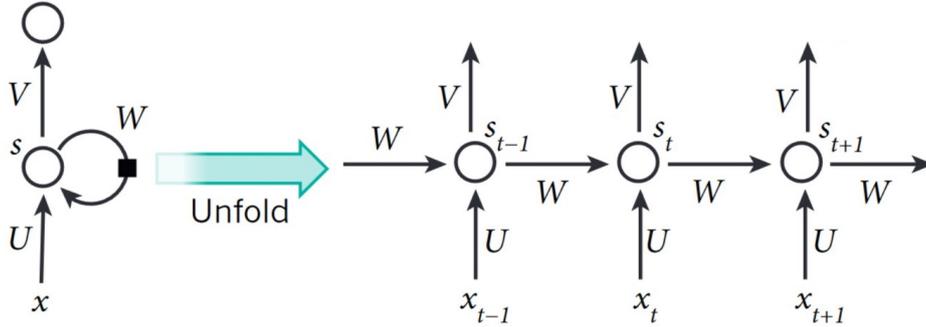


Figure 3.4: Graphical Display of an RNN node adapted from [41]. U , V and W denote weight matrices, x the input vector, s the interval state and t the time step.

3.4.3 DeepSet Neural Networks

The DeepSet neural network (DeepSet NN) implements a neural network architecture that enables processing unordered sequences with variable length. If the sequence order has no meaning in the context of the ML problem, this can have advantages over RNNs as time-consuming loops within the nodes can be avoided. The formal definition of the DeepSet NN can be summarized as

$$\hat{\mathbf{y}}(\mathbf{x}) = \lambda \left(\sum_i \kappa_i(\mathbf{x}) \right). \quad (3.15)$$

The input vector \mathbf{x} is transformed into some representation $\kappa(\mathbf{x})$, whose components are added up and fed into a feed forward neural network λ . κ is usually a feed forward neural network as well. As the summation is commutative, it marks the step at which the ordering of the input elements becomes irrelevant [43].

3.5 Comparison between Deep Learning and Classical ML Approaches

Apart from ANNs, a variety of classical ML algorithms have been used long before the ascent of deep learning. These include linear regression, support vector machines (SVM), nearest neighbor algorithms and boosted decision trees (BDT). Most of them require less computational power, but do require a more elaborated extraction of the input features. As BDTs are widely used in ATLAS τ reconstruction, they will be compared to ANNs in more detail in the following. Just like an ANN, a BDT can perform supervised classification and regression tasks. The tree is trained iteratively. In a simple classification task with $\mathcal{Y} = \{0, 1\}$. During the first step, an input feature is chosen, and the samples are ordered according to this variable. Now, for different thresholds of the input feature, the dataset is split into two. In each subset, the purity of labels is calculated. The same is repeated for the other input features. The input feature and threshold that produces the highest purity is chosen. Subsequently, two branches exist. For each of them, the procedure is repeated until all branches have reached an acceptable level of purity. These *leaves* acquire the label of the majority of samples that they contain. To classify a new sample with unknown label, the sample is passed through the tree and at every threshold it takes

its according path until it reaches a leaf. The sample is labeled with the leaf's label. The BDT is *boosted* by giving higher weights to wrongly classified samples during the training process. Usually, BDTs require higher-level representations of the data, but therefore training is faster. Furthermore, if a BDT is predicting poorly, its tree structure can give hints on what the problem might be. This does not account for deep ANNs that act more like a black box. However, the advantages of BDTs only apply, if it is possible to extract all information inside the data into a manageable amount of input features and the training dataset is not too large. Otherwise, it can drastically increase training duration due to the iterative training approach. The gradient-based ANN training does not have this problem. Another disadvantage of BDTs is the fact, that they can only represent linear decision boundaries. As already mentioned, ANNs can learn much more complex representations [44].

3.6 Hyperparameter Optimization

Until now, the mentioned parameters of the ML algorithm, like the weights in an ANN, were all adjusted during training. Hyperparameters, however, are higher-level parameters that are set manually before training. There are two types of hyperparameters: The first type only concerns the model, like the number of layers of an ANN or the activation function. The second type concerns the optimization process, like learning rate and batch size [45]. There are various ways of determining the hyperparameter configuration that suits best the specific ML problem. They vary in complexity from trying out random hyperparameter values (*random search*) to training artificial neural networks that predict the optimal hyperparameter values. Tuning hyperparameters is usually consuming a high amount of computational resources, so it is advisable to choose the tuning algorithm thoughtfully.

3.6.1 Grid Search

When performing a grid search, range and step size for each hyperparameter to be optimized are defined. A loop over all hyperparameter combinations is run and the hyperparameter configuration that produces the lowest final validation loss is chosen. An advantage of this method is that by choosing large ranges and small step sizes, the hyperparameter space can be entirely scanned. On the other hand, computational costs rise quickly with larger ranges and smaller step sizes.

3.6.2 Bayesian Optimization

A problem shared among the random and grid search approach is the fact that the algorithms do not consider previous observations. *Bayesian optimization* addresses this problem by keeping history over the previously tested hyperparameter configurations and their performance. On every *call*, they choose a hyperparameter configuration that is in the region of former configurations with high performance. If the loss landscape with respect to the hyperparameter values is not too complex, this can lead to quick convergence.

Reconstruction of τ Leptons with the PanTau Algorithm

As highlighted in Section 1.4, an accurate reconstruction of τ leptons is an essential part of Higgs CP measurements at ATLAS. However, the reconstruction can be quite challenging: Due to its low lifetime, the τ decays already before reaching the detector. The neutrino generated by the decay cannot be directly detected by the ATLAS detector (Chapter 2). Therefore, the visible part of the decay (τ_{vis}) has to be reconstructed with high accuracy. Reconstruction of leptonic τ decays is rather simple as the resulting muon or electron can be directly detected. For hadronic τ decays, more elaborate techniques are required to reconstruct the hadronic remnants ($\tau_{\text{had-vis}}$) as their signature is similar to the one of an ordinary jet. For simplicity, $\tau_{\text{had-vis}}$ will be referred to as τ_{vis} from now on.

In previous ATLAS analyses like [46], the PanTau algorithm used a particle flow algorithm (see Section 4.2) and various BDTs (see Section 3.5) to provide τ decay mode reconstruction and reconstruction of the decay products' kinematics. A full diagram of the PanTau workflow can be found in Appendix C. The aspects relevant for this analysis are described in this chapter.

4.1 Identification of τ Leptons

As already mentioned, before reconstructing the τ_{vis} candidate, accurate τ identification has to be performed to discriminate from regular jets. Fig. 4.1 demonstrates common differences between jets and τ_{vis} : In general, the jet is fewer collimated and contains more hadrons than the τ_{vis} [47]. Since Run 2, τ_{vis} identification is achieved using an ANN initially developed by C. Deutsch [48, 49].

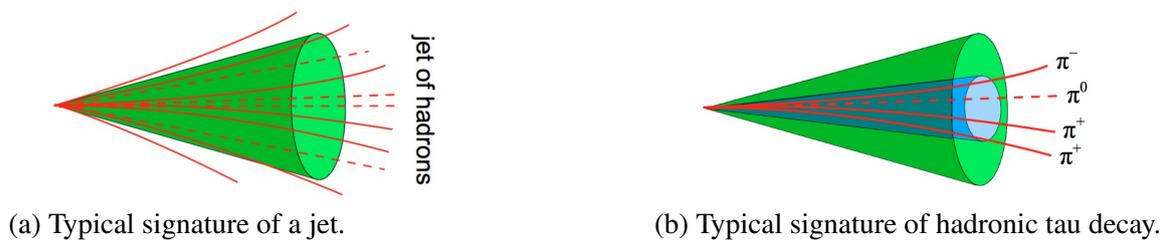


Figure 4.1: Graphical display of common differences between a jet (a) and a τ_{vis} (b) [47].

4.2 CellBased Particle Flow Algorithm

CellBased is an algorithm on calorimeter cell level developed to reconstruct objects stemming from hadronic τ decays. It utilizes various properties of the ATLAS detector described in Chapter 2. Charged particle flow objects (Charged PFOs), mostly originating from charged pions, can easily be identified by the corresponding track in the pixel detector. Their total energy can be estimated from the energy deposited in the electromagnetic calorimeter. The remaining energy deposits are reclustered to receive the neutral pion candidates (Neutral PFOs). These however, can also originate from other sources like pile-up (Section 2.4). Therefore, a BDT is used to identify the clusters from real neutral pions [50]. Furthermore, a custom algorithm developed by S. Yuen provides information on the connections between Neutral PFOs and Photon Shots [51].

A schematic display of the signature of a sample $1p1n$ decay (Section 1.1) in the η - ϕ -plane can be seen in Fig. 4.2. Conversion Tracks from electron pair production are depicted as well. One can observe well the charged PFO originating from the charged pion as well as two groups of objects originating from the neutral pion decay to two photons.

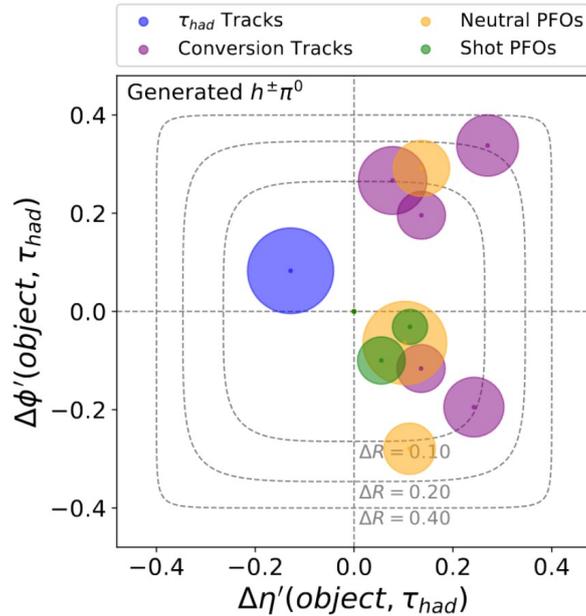


Figure 4.2: Graphical display of the reconstructed objects in a generated τ decay. Each circles' area represents the object's energy [52].

4.3 Decay Mode Reconstruction

The τ decay mode is determined by a combination of various BDTs that use the kinematics of all PFOs found by the CellBased algorithm. The performance of this classification task can be quantified using a confusion matrix [35]. The two axes of the matrix represent true (x-axis) and predicted decay mode (y-axis). Here, column normalization is used to obtain the classification efficiency for each decay mode (migration matrix). The overall efficiency is the weighted average of the values on the diagonal from

bottom left to top right. The resulting migration matrix for the PanTau classification on a $\gamma^* \rightarrow \tau\tau$ dataset can be observed in Fig. 4.3. The diagonal efficiency is 73.1%.

Reco Tau Decay Mode	ATLAS Simulation Internal					Diagonal Efficiency
	1p0n	1p1n	1pXn	3p0n	3pXn	
3pXn	0.0	0.5	0.5	5.4	58.0	73.1%
3p0n	0.1	0.1	0.1	91.4	36.6	
1pXn	2.0	11.2	40.4	0.5	1.7	
1p1n	16.6	77.2	56.1	1.4	3.3	
1p0n	81.3	11.0	2.9	1.2	0.4	

Figure 4.3: Migration matrix of the decay mode classification performance of the PanTau algorithm on a $\gamma^* \rightarrow \tau\tau$ sample [34].

4.4 Prediction of the Decay Product Kinematics

In order to perform the ρ decay plane method described in Section 1.4, one needs to reconstruct the four-vectors of the charged and neutral pion from the 1p1n decay. The charged pion four-vector can be obtained with high accuracy from the tracking system, so the crucial part is reconstructing the neutral pion. It does not leave a track in the Inner Detector (only in the case pair production before entering the ECAL) and its signature in the calorimeter can be difficult to interpret. Therefore, PanTau exploits information from various detector parts with a custom algorithm. Consider Appendix C for further details.

The performance of the prediction can be measured in different figures of merit that describe the trajectory of the neutral pion from the 1p1n decay. To isolate these events, only events true decay mode 1p1n will be considered in the following. Furthermore, to be consistent with the performance measurement in Chapter 6, only events with 1p1n predicted by the ANN in Chapter 5 are considered. Due to the large p_T spectrum, the unphysical $\gamma^* \rightarrow \tau\tau$ dataset serves well for developing a general τ reconstruction. However, in this thesis, reconstructing physical decays that enable Higgs CP measurements is of interest. Therefore, the neural network is evaluated on a $Z \rightarrow \tau\tau$ sample that was generated by using the *tag-and-probe* method. As introduced in Section 2.4, a single muon is used as a trigger (tag). If a τ_{had} is found in the same event (probe), the event is accepted [53].

As describes in Section 1.4, an important figure of merit is the energy resolution of the neutral pion. We define the $E(\pi^0)$ residual as

$$\Delta E(\pi^0) = \frac{E(\pi_{\text{PanTau}}^0) - E(\pi_{\text{true}}^0)}{E(\pi_{\text{true}}^0)}. \quad (4.1)$$

The residual distribution is shown for different binnings of $p_T(\tau_{\text{vis,true}})$ in Fig. 4.4. The outliers at -100% result from events for which PanTau predicts $E(\pi^0) = 0$. As we use the decay mode selection predicted by the ANN in Chapter 5, these events were reconstructed as $1p0n$ by PanTau. By considering the

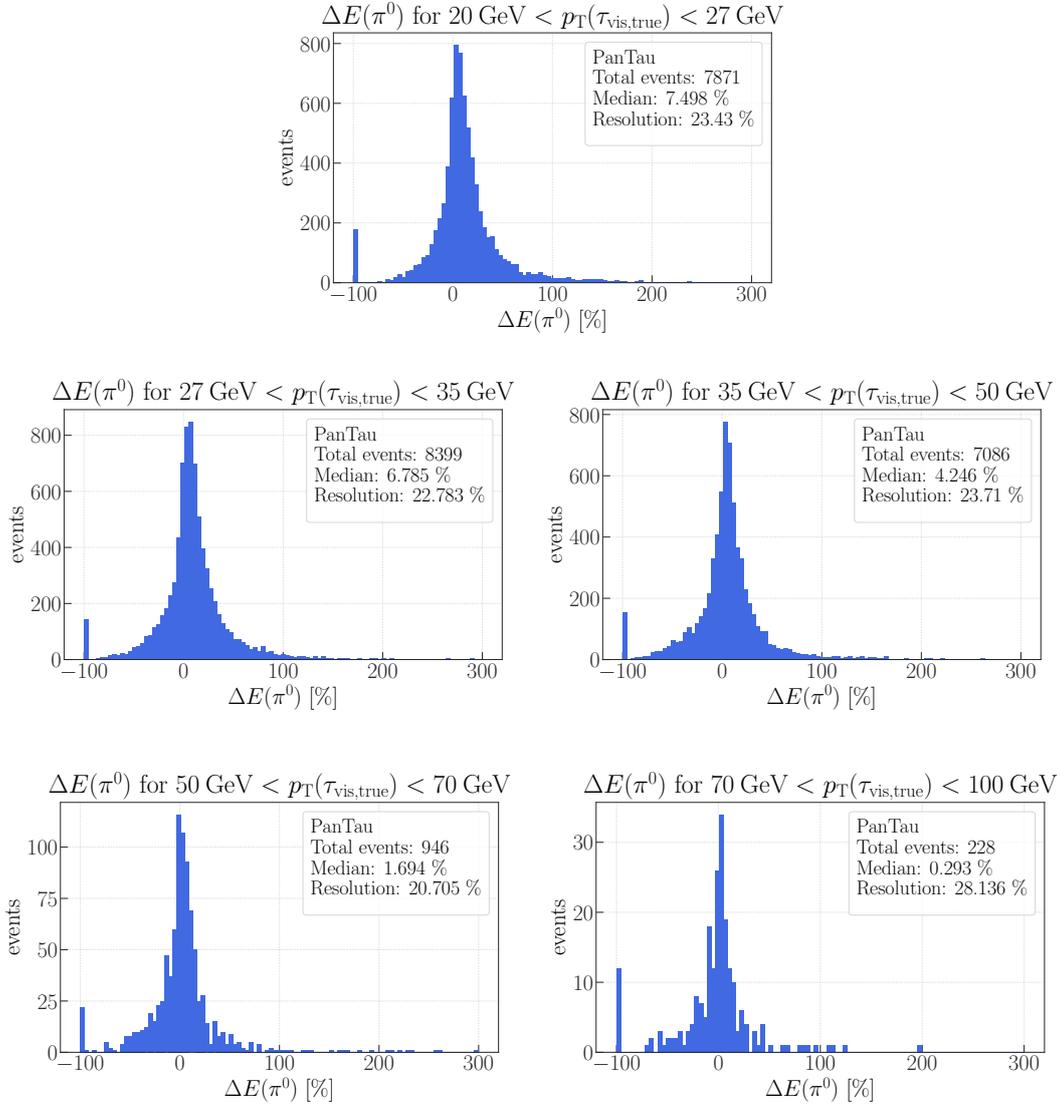


Figure 4.4: Distributions for different intervals of $p_T(\tau_{\text{vis,true}})$ of the $E(\pi^0)$ residual for the PanTau algorithm.

quantiles of these distributions, the behavior can be summarized (Fig. 4.5). Another way to quantify the performance is by defining the overall resolution as the average deviation of all values within the 90% quantile and the bias as the median of the distribution. The behavior of these two observables is depicted in Fig. 4.6. One can observe that bias and resolution both decrease for higher $p_T(\tau_{\text{vis,true}})$. An explanation for this could be the uncertainty on the compensation for pile-up in the calorimeter. Here, a constant value is subtracted from all energy deposits. With higher $p_T(\tau_{\text{vis,true}})$ and therefore on average

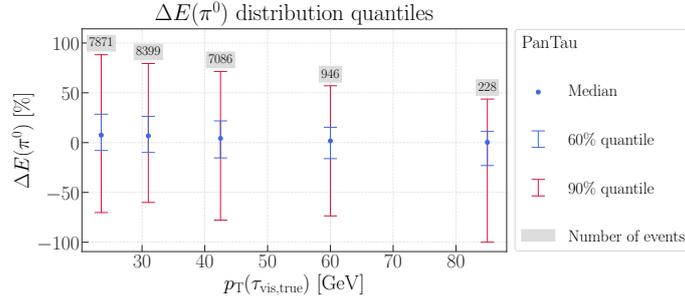


Figure 4.5: Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$.

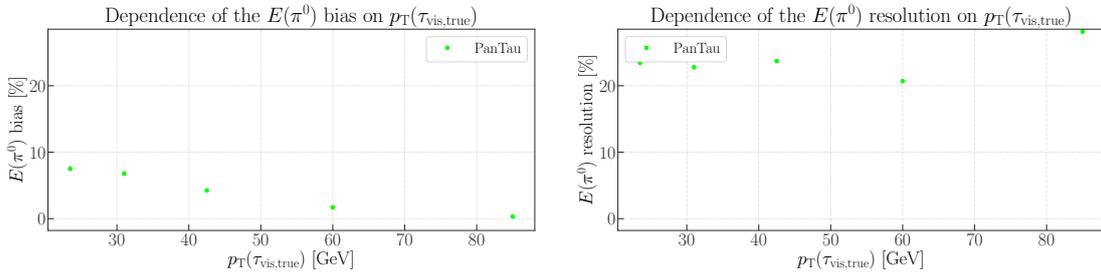


Figure 4.6: Bias and resolution of the $E(\pi^0)$ prediction for the PanTau algorithm.

higher $E(\pi^0)$ as well, this constant error has a lower impact on the overall uncertainty. To understand the energy resolution in more detail, it is helpful to consider the angular distance between the two pions. According to Eq. (2.1), it is defined as

$$d(\pi^0, \pi^\pm) = \sqrt{(\phi(\pi^0) - \phi(\pi^\pm))^2 + (\eta(\pi^0) - \eta(\pi^\pm))^2}. \quad (4.2)$$

The dataset is split into high and low $d(\pi^0, \pi^\pm)$ and the quantiles of each residual distribution are shown in Fig. 4.7. This time against $E(\pi_{\text{true}}^0)$. A small angular distance should make assigning the different objects to the right one of the two pions harder, which should result in a worse energy resolution. However, a small distance between the two pions is more likely with larger $p_T(\tau_{\text{vis}})$. In the plot, one can observe this effect: Apart from the lowest energy bin, the $E(\pi^0)$ resolution is better for higher angular distances.

Another measure of interest is the angular orientation of the τ axis in the η - ϕ -plane. In the 1p1n decay, it is defined by the orientation of the straight line passing through the two pions. $\alpha(\pi^0, \pi^\pm) \in [0, \pi]$ is the unsigned angular direction deviation from the true τ axis. To consider the effects of high and low energy neutral pions, similarly to before, a cut on $E(\pi_{\text{true}}^0)$ is performed to visualize the angular direction resolution (Fig. 4.8). One can observe that not only the energy resolution improves with higher energies, but also the angular direction resolution. In the low energy bin, it remains more or less constant with higher angular distances between the two pions. In the high energy bin, the resolution improves. This is reasonable due to the already mentioned effect that with low angular distances, the objects are harder to assign correctly to the pions.

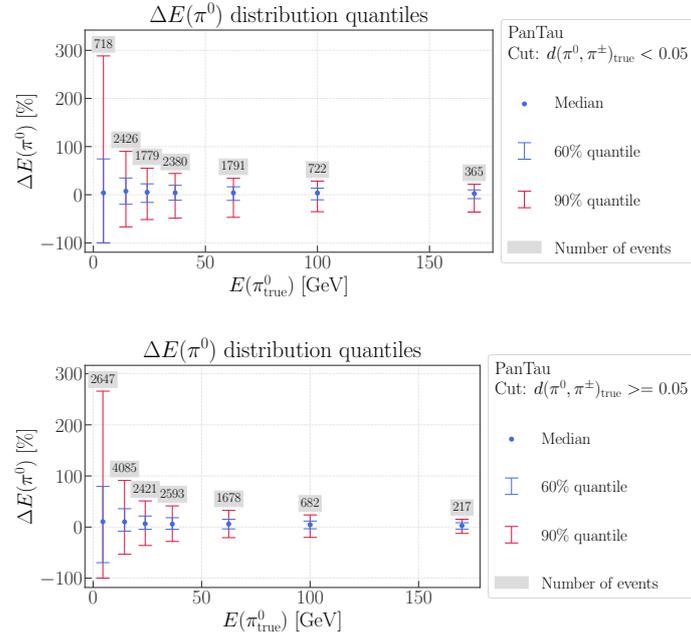


Figure 4.7: Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $E(\pi^0_{\text{true}})$ with a cut on $d(\pi^0, \pi^\pm)$.

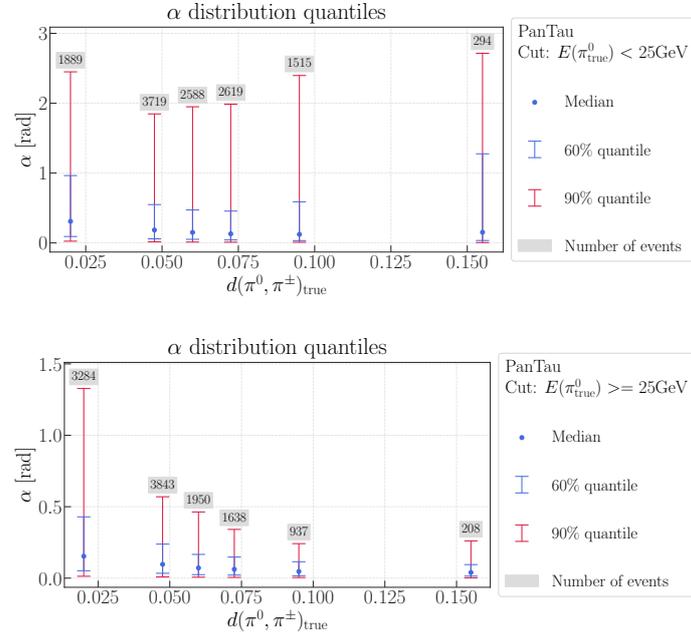


Figure 4.8: Display of the quantiles of the $\alpha(\pi^0, \pi^\pm)$ residual distribution for the PanTau algorithm against $d(\pi^0, \pi^\pm)$ with a cut on $E(\pi^0_{\text{true}})$.

As explained in Section 1.4, the energy asymmetry Y is the more physics oriented measure in this analysis than $E(\pi^0)$. The corresponding performance of PanTau is shown in Fig. 4.9. The behavior of a stronger performance with larger $p_T(\tau_{\text{vis,true}})$ can be seen as not as strong as for $E(\pi^0)$.

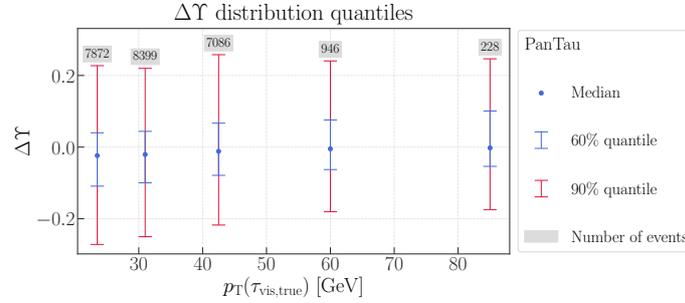


Figure 4.9: Display of the quantiles of the Y residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$.

Further figures of merit are the angular deviation in ϕ and η (Figs. 4.10 and 4.11). Worth noting are the long tails of the distributions in the highest p_T bin. Apart from that, like for the other observables the resolution gets better with higher pt .

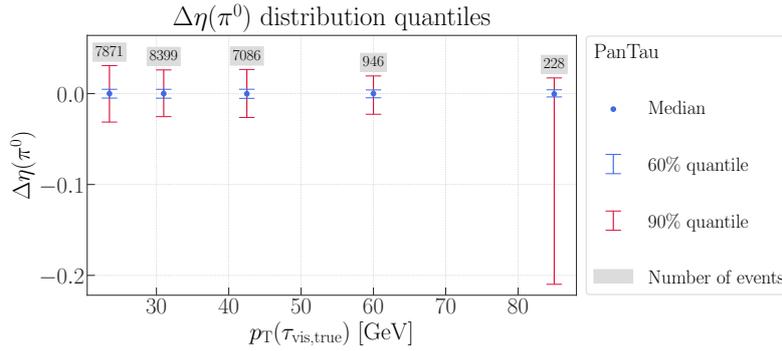


Figure 4.10: Display of the quantiles of the $\eta(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$.

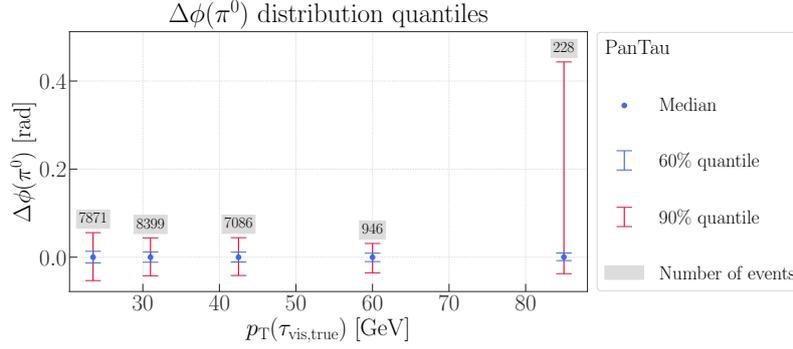


Figure 4.11: Display of the quantiles of the $\phi(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$.

In addition, the resolution on $d(\pi^0, \pi^\pm)$ is of interest, as it shows, if it is possible to resolve the locations of the two pions. The performance is displayed against the true angular distance in Fig. 4.12. For easier visualization, two different y-axis ranges are shown. The plot shows that in the in most cases,

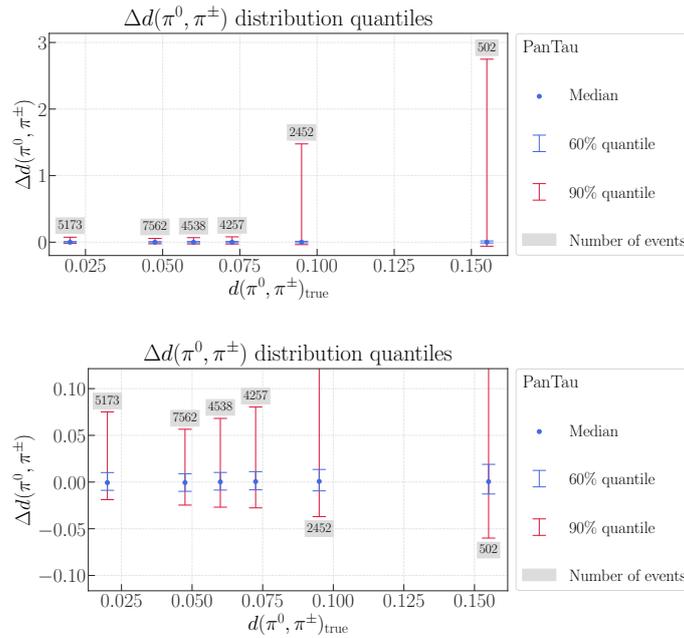


Figure 4.12: Display of the quantiles of the $d(\pi^0, \pi^\pm)$ residual distribution for the PanTau algorithm against $d(\pi^0, \pi^\pm)$ for two different ranges on the y-axis.

the distance resolution is significantly smaller than the true distance. Only for small distances this is not the case.

Furthermore, it is helpful to consider dependence of the performance of PanTau on the angular direction of the neutral pion. The $E(\pi^0)$ resolution is displayed against $\phi(\pi_{\text{true}}^0)$ and $\eta(\pi_{\text{true}}^0)$ in Fig. 4.13. As expected, the resolution does not deviate strongly with ϕ and η .

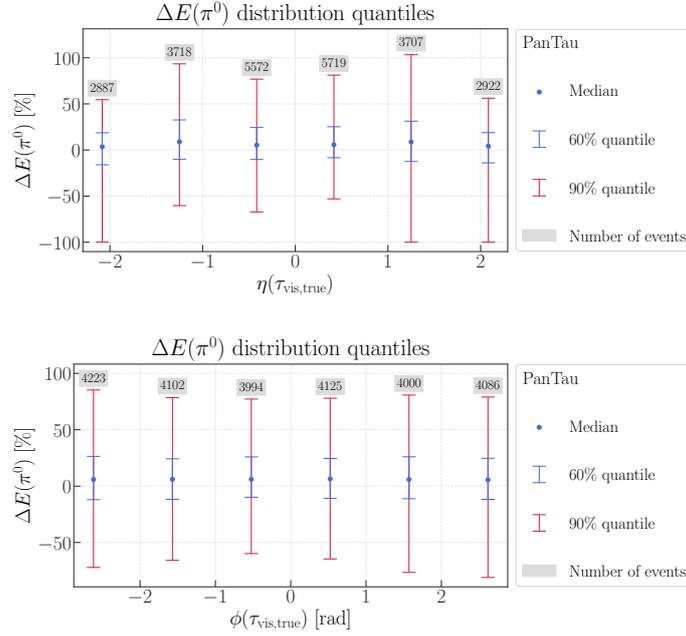


Figure 4.13: Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $\eta(\pi_{\text{true}}^0)$ and $\phi(\pi_{\text{true}}^0)$.

Another factor that can impair performance is pile-up as explained in Section 2.4. The corresponding plot can be seen in Fig. 4.14. As expected, with more interactions per bunch crossing, the resolution

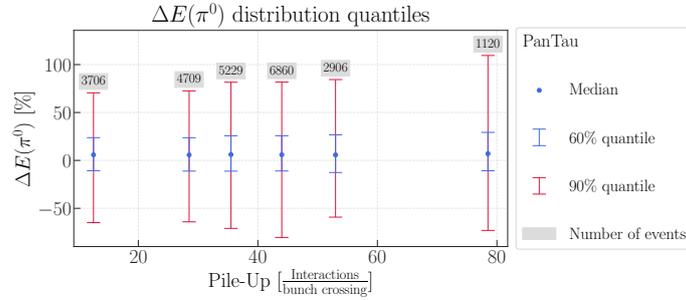


Figure 4.14: Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against pile-up.

worsens.

As the $1p1n$ decay is mediated by the ρ particle (Section 1.4), the invariant mass of charged and neutral pion should result in the ρ mass. The predicted distribution of the ρ mass and the corresponding distribution on truth level is shown in Fig. 4.15. It is clear to see that PanTau has a bias towards higher masses. The origin of the outliers around the π^\pm mass (140.0 GeV [2]) can be explained by the fact that, if no π^0 is detected, the ρ mass is just the mass of the reconstructed π^\pm . This is the same effect as for the outliers in Fig. 4.4.

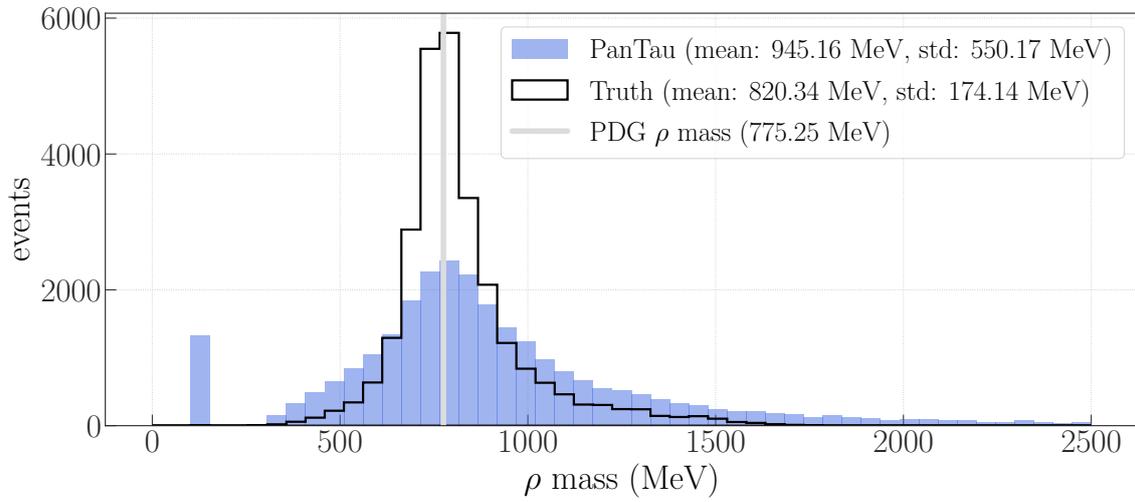


Figure 4.15: Display of the ρ meson mass distribution from the $1p1n$ decay for PanTau and truth data. The Particle Data Group (PDG) average is depicted as well [2].

For more displays of the PanTau reconstruction performance consider Appendix A.

Decay Mode Reconstruction with a Recurrent Neural Network (RNN)

Before improving the reconstruction of the τ decay product kinematics, it is worth considering an updated method of the decay mode reconstruction. Inspired by the previously cited work from C. Deutsch, an RNN (Section 3.4.2) was developed by H. Nguyen in [34], which significantly outperformed the PanTau decay mode classification by exploiting the advantages of deep neural networks over BDTs described in Section 3.5.

In this section, especially input data, architecture and performance on different data samples will be discussed. Even though the developed RNN does not (like the PanTau algorithm) deliver the decay product kinematics, it serves as a basis for a second neural network that will be described and optimized in the following chapter to reconstruct the decay products' four vectors.

5.1 Input Data

The RNN is trained on a Monte Carlo data sample (Section 2.5) simulating the unphysical $\gamma^* \rightarrow \tau\tau$ decay.

5.1.1 Input Features

Just like for PanTau, the RNN input features contain the kinematics of Charged and Neutral PFOs: The signed angular distance to the τ axis in η and ϕ and the 10th logarithm of the PFO's transverse momentum. As especially the correct reconstruction of neutral pions is challenging, more information on the Neutral PFOs is added. In addition, the network receives the kinematic information for the Photon Shots and Conversion Tracks. The full list of input features with explanations is shown in Table 5.1.

5.1.2 Cut Selection

The cuts listed in Table 5.2 were applied to speed up training and avoid that the neural networks learns common reconstruction errors: The cut applied on $p_T(\tau_{\text{vis}})$ avoids that the model learns reconstruction errors as the Particle Flow algorithm was developed for $p_T(\tau_{\text{vis}})$ in this interval. The cut on η avoids events where the τ is located too close to the beam axis to be properly detected by the tracking system and rejects τ candidates between barrel and end-cap of the calorimeter.

Table 5.1: RNN input features

Object	Variable	Description
Charged PFO	$\Delta\phi(\tau, \text{Charged PFO})$	Distance in ϕ between Charged PFO and τ axis
	$\Delta\eta(\tau, \text{Charged PFO})$	Distance in η between Charged PFO and τ axis
	$\log(p_T(\text{Charged PFO}))$	logarithm of Charged PFO transverse momentum
	$\log(p_T(\tau))$	logarithm of reconstructed τ transverse momentum
Neutral PFO	$\Delta\phi(\tau, \text{Neutral PFO})$	Distance in ϕ between Neutral PFO and τ axis
	$\Delta\eta(\tau, \text{Neutral PFO})$	Distance in η between Neutral PFO and τ axis
	$\log(p_T(\text{Neutral PFO}))$	logarithm of Neutral PFO transverse momentum
	$\log(p_{T,\tau})$	logarithm of reconstructed τ transverse momentum
	π^0 BDT score	likeliness of the Neutral PFO stemming from a π^0
	NHitsInEM1	Number of Photons associated with the Neutral PFO
	NPosECells_EM1	Number of cells with positive energy in EM1
	ENG_FRAC_CORE	Fraction of total cluster energy contained in the highest energy cell
	SECOND_R	Second moment of the radial distance between cluster cells and shower axis
	ptSubRatio	Fractional transverse momentum overlap between Neutral and Charged PFOs
Photon Shot	$\Delta\phi(\tau, \text{Shot})$	Distance in ϕ between Photon Shot and τ axis
	$\Delta\eta(\tau, \text{Shot})$	Distance in η between Photon Shot and τ axis
	$\log(p_T(\text{Shot}))$	logarithm of Photon Shot transverse momentum
	$\log(p_T(\tau))$	logarithm of reconstructed τ transverse momentum
Conversion Track	$\Delta\phi(\tau, \text{ConvTrack})$	Distance in ϕ between Conversion Track and reconstructed τ axis
	$\Delta\eta(\tau, \text{ConvTrack})$	Distance in η between Conversion Track and reconstructed τ axis
	$\log(p_T(\text{ConvTrack}))$	logarithm of Conversion Track transverse momentum
	$\log(p_T(\tau))$	logarithm of reconstructed τ transverse momentum

Table 5.2: Cuts applied on the training data.

- $20 \text{ GeV} > p_T(\tau_{\text{vis}}) > 100 \text{ GeV}$ on truth and reconstruction level
- $|\eta| < 2.5$ AND ($|\eta| > 1.52$ OR $|\eta| < 1.37$) on truth and reconstruction level
- number of charged tracks of one or three on truth and reconstruction level
- truth decay mode is one out of the modes listed in Table 1.1

5.1.3 Target

The target is the truth matched decay mode.

5.2 RNN Architecture

The recurrent form of the neural network is used to be able to pass a series of vectors to the network instead of only one vector. This is necessary, as there is e.g. up to three charged tracks per event. As the RNN additionally requires an ordering of the vectors, they are passed ordered by p_T even though this has no physical meaning.

The RNN architecture is depicted in Fig. 5.1. The data for the four different types of objects is received by the RNN in four different branches. Per event, up to three Charged PFOs, ten Neutral PFOs, six Photon Shots and four Conversion Tracks are passed as a sequence to the network. Each branch consists of a dense layer and an LSTM layer. The number of nodes is shown in brackets in the figure. After merging the four branches, three dense layers are added before a 5-vector with the probabilities for the five chosen decay modes is returned through the softmax activation function. The internal activation function is always the hyperbolic tangent.

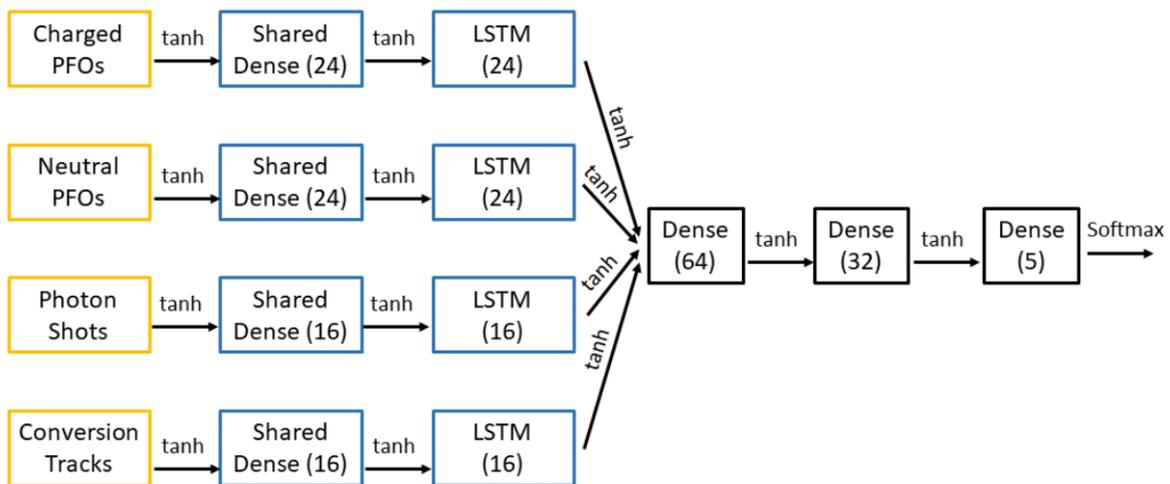


Figure 5.1: Architecture of the used RNN for decay mode reconstruction [34].

5.3 Preprocessing and Training

The Keras framework with TensorFlow as backend is used for training and evaluation. For proper separation of training, test and validation data, the dataset is split randomly according to the ratio 4:5:1. K-fold cross-validation is not chosen as it is computationally expensive and due to the large size of the dataset [35]. In addition, the input features are standardized. In each epoch, the neural network is evaluated on the training data and the validation error is calculated on the validation data. By using the Adam optimizer with an initial learning rate of 0.001 and a batch size of 256 the weights of the RNN are adjusted continuously. A maximum of 300 epochs is chosen. An early stopping criterion assures that

training stops, if the validation loss does not improve any further. This should prevent overtraining as the validation loss is calculated on the validation set (independent of the training set).

5.4 RNN Classification Performance

The classification performance of the RNN on the $\gamma^* \rightarrow \tau\tau$ sample is shown in Fig. 5.2. The improvement compared to PanTau (Fig. 4.3) can be clearly observed. The diagonal efficiency improves to 80.0%. Anyway, the deviation that produces the 2x2 submatrix in the right top corner and the 3x3 submatrix in the left bottom corner can still be observed, which can be explained by the fact that the neural network still has problems determining the number of neutral hadrons.

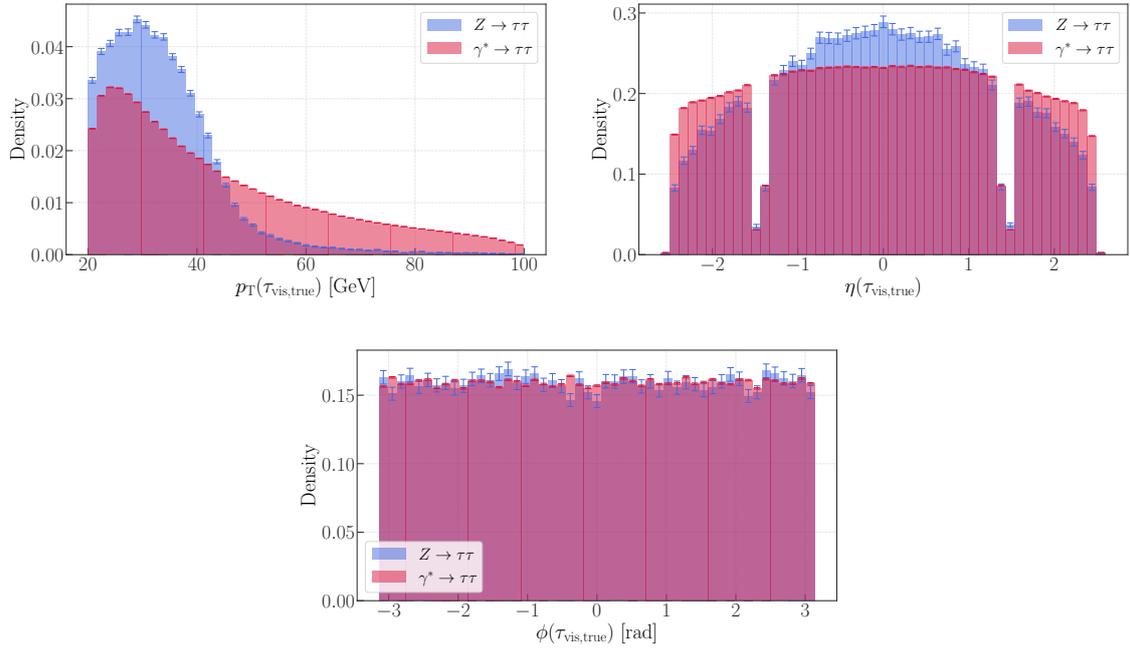
Reconstructed decay mode	Column norm.					Diagonal efficiency:
	h^\pm	$h^\pm\pi^0$	$h^\pm \geq 2\pi^0$	$3h^\pm$	$3h^\pm \geq 1\pi^0$	80.0
$3h^\pm \geq 1\pi^0$	0.2	0.5	0.6	5.1	71.1	
$3h^\pm$	0.7	0.2	0.1	90.4	19.5	
$h^\pm \geq 2\pi^0$	1.2	7.3	60.3	0.5	3.5	
$h^\pm\pi^0$	12.6	84.5	37.9	2.1	5.4	
h^\pm	85.3	7.5	1.2	1.9	0.4	

Figure 5.2: Migration matrix of the decay mode classification with the RNN setup from H. Nguyen in [34] on a $\gamma^* \rightarrow \tau\tau$ sample.

As already explained in Section 4.4, the especially the performance on the physical $Z \rightarrow \tau\tau$ sample is of interest. Therefore, the model is evaluated once again on this dataset. The classification efficiency for the 1p1n decay increases from 84.5% to 88.7%, while the diagonal efficiency drops to 72.0%, which is not trivial. This can be explained by considering the distribution of the decay modes predicted by the standard model (Table 1.1). As the number of 1p1n decays is much larger than the number of occurrences of the other decay modes, a higher efficiency for 1p1n influences the diagonal efficiency disproportionately high. This does not explain the deviation in efficiency for the single decay modes, though. A trivial reason for this could be overtraining. If the evaluation set for $\gamma^* \rightarrow \tau\tau$ is not independent of the training set, then the performance on the obviously independent $Z \rightarrow \tau\tau$ sample could be worse. However, overtraining is not likely due to the already explained early stopping criterion. Furthermore, training and test set were split randomly to assure independence.

Another reason could be that different observables are differently distributed in the two datasets. This can be observed in Fig. 5.4. As already mentioned, the $p_T(\tau_{\text{vis,true}})$ spectrum of the $\gamma^* \rightarrow \tau\tau$ process is much wider than for $Z \rightarrow \tau\tau$. A similar effect can be observed for $\eta(\tau_{\text{vis,true}})$. As expected, the $\phi(\tau_{\text{vis,true}})$ distributions do not differ by much. The behavior of the prediction efficiency against different observables for each decay mode is shown in Fig. 5.5 to 5.8. There is no strong dependency on any variables. Only the efficiency for the 1pXn decay mode increases with higher $p_T(\tau_{\text{vis,true}})$.

Reconstructed decay mode	Column norm.					Diagonal efficiency: 72.0
	h^\pm	$h^\pm\pi^0$	$h^\pm \geq 2\pi^0$	$3h^\pm$	$3h^\pm \geq 1\pi^0$	
$3h^\pm \geq 1\pi^0$	0.2	0.4	0.3	9.8	66.3	
$3h^\pm$	0.6	0.1	0.1	86.1	17.3	
$h^\pm \geq 2\pi^0$	2.5	6.9	39.5	0.9	3.7	
$h^\pm\pi^0$	35.7	88.7	59.6	2.7	12.7	
h^\pm	61.1	3.9	0.5	0.5	0.0	

 Figure 5.3: Migration matrix of the decay mode classification with the RNN on a $Z \rightarrow \tau\tau$ sample.

 Figure 5.4: Distributions of different observables in the $Z \rightarrow \tau\tau$ and $\gamma^* \rightarrow \tau\tau$ sample. The error bars denote the Poisson error.

The unclear effects could be further understood by assessing the performance against other observables that are differently distributed in the different datasets.

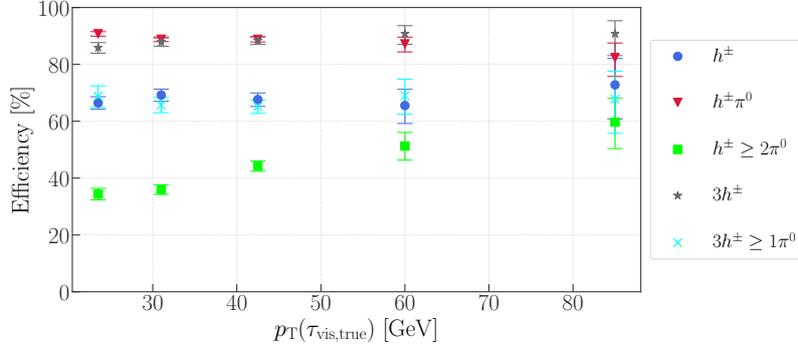


Figure 5.5: Dependence of the classification efficiency for different decay modes on $p_T(\tau_{\text{vis,true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].

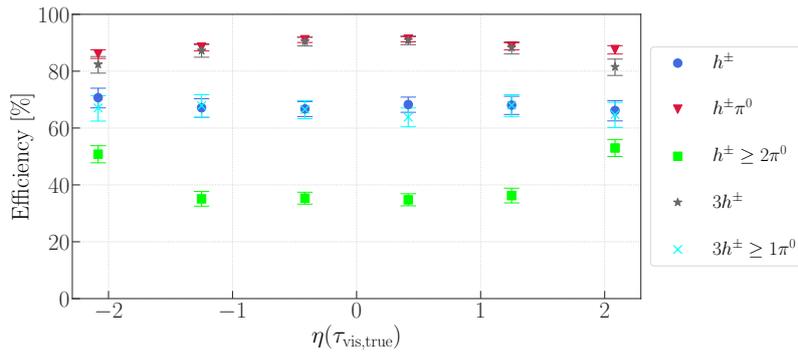


Figure 5.6: Dependence of the classification efficiency for different decay modes on $\eta(\tau_{\text{vis,true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].

5.5 Correction Using the Number of Charged Tracks

While the already mentioned error counting the number of neutral hadrons can hardly be resolved, mistakes while counting the number of charged hadrons can be corrected by the number of charged tracks delivered with high accuracy from Tracking System. Applying this correction results in the migration matrix shown in Fig. 5.9. It is obvious, how this correction helps to improve the efficiency for the $3pXn$ decay mode. Even though the efficiency for the $1p1n$ decay worsens slightly, an overall more consistent τ reconstruction is received by applying this correction.

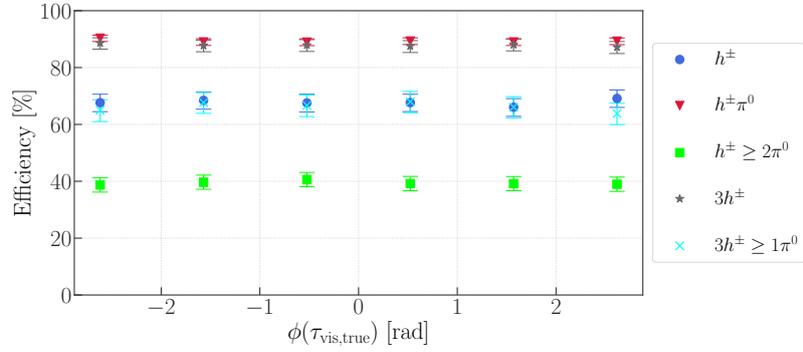


Figure 5.7: Dependence of the classification efficiency for different decay modes on $\phi(\tau_{\text{vis, true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].

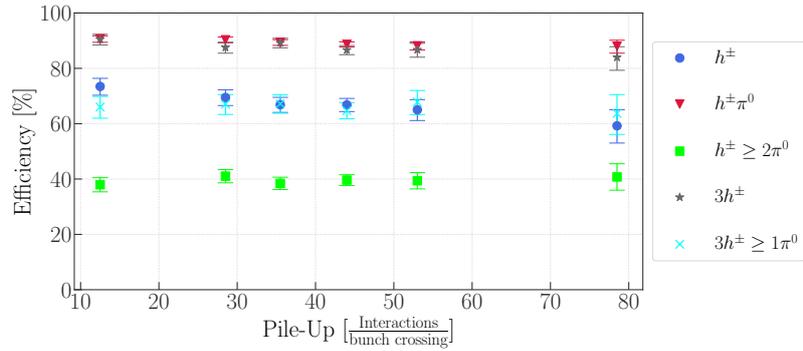


Figure 5.8: Dependence of the classification efficiency for different decay modes on the average number of interactions per bunch crossing. The error bars denote the 99% Clopper-Pearson intervals [54].

Reconstructed decay mode	Column norm.					Diagonal efficiency:
	h^\pm	$h^\pm\pi^0$	$h^\pm \geq 2\pi^0$	$3h^\pm$	$3h^\pm \geq 1\pi^0$	72.5
$3h^\pm \geq 1\pi^0$	0.3	1.2	1.4	10.1	77.1	
$3h^\pm$	0.6	0.2	0.1	86.2	17.5	
$h^\pm \geq 2\pi^0$	2.5	6.8	38.9	0.7	1.9	
$h^\pm\pi^0$	35.6	87.9	59.1	2.3	3.5	
h^\pm	61.0	3.9	0.5	0.5	0.0	
	True decay mode					

Figure 5.9: Migration matrix of the decay mode classification with the RNN on a $Z \rightarrow \tau\tau$ sample with a post-processing correction using the number of tracks.

Reconstruction of the Neutral Pion in the 1p1n Decay Mode with a Neural Network

A prototype of an RNN (Section 3.4.2) that predicts the neutral pion kinematics was developed by Y. Kinoshita in [55] based on H. Nguyen's work. The setup that predicts the pion kinematics is very similar, but various changes are imposed to adapt the ML algorithm from a classification to a regression task. In this chapter, the architecture, preprocessing and training configuration are introduced and optimized. Then, the performance is assessed in the figures of merit introduced in Chapter 4.

6.1 Setup

As described in the previous chapter, the performance of the reconstruction on the physical $Z \rightarrow \tau\tau$ dataset is of interest. Therefore, training and evaluation are performed on this sample. The input features stay the same as in Table 5.1. A major change is the target. Instead of using the truth decay mode for the optimization, the four-vector of the true neutral energy (E, p_T, η, ϕ) is used. It is transformed to the following three variables to achieve similar orders of magnitude: The transverse momentum is divided by the charged transverse momentum and the angles η and ϕ are replaced by the angular distances between charged and neutral pion. The mass of the neutral pion is well known to be 135.0 GeV [2], so three components of the four-vector are sufficient to describe the neutral pion kinematics. The target vector is therefore

$$\left(p_T(\pi^0)/p_T(\pi^\pm), \Delta\phi(\pi^0, \pi^\pm), \Delta\eta(\pi^0, \pi^\pm) \right). \quad (6.1)$$

As predicting a vector with numerical values is a regression task and not a classification task. The network has to be adjusted: The final activation function is changed to a linear function and the loss function is changed to mean squared error Eq. (3.4). The selection cuts are mostly adopted from the decay mode network. The only addition is requiring the truth decay mode to be 1p1n. All other configurations on training, preprocessing and architecture stay the same. The updated regression RNN architecture can be seen in Fig. 6.1.

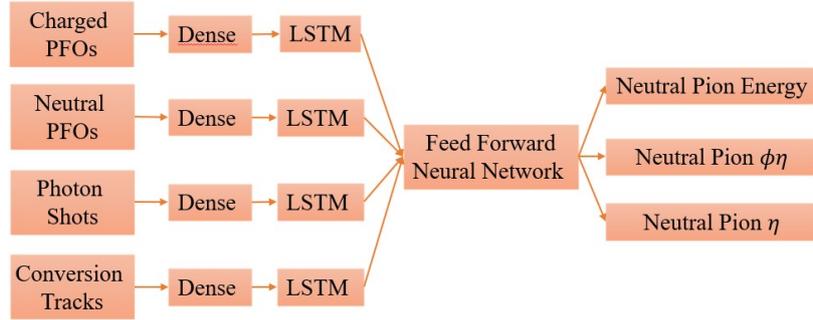


Figure 6.1: Architecture of the regression RNN.

6.2 Performance of the Recurrent Neural Network

The figures of merits are calculated as introduced for the PanTau algorithm in Chapter 4. To estimate the stability of the ANN performance, the network is trained several times with different Tensorflow seeds (10, 11, 12). The value for resolution and bias are the average of the three runs; the uncertainty is the standard deviation. As computational resources are limited, training only three times is chosen. Training multiple times is not trivial to do for the PanTau algorithm with this setup, so no uncertainty is shown here. The performance of the regression neural network can be visualized with several details, as already shown in Section 4.4. To simplify, in this section, the focus will mostly lie on energy and angular resolution. The full selection of plots can be found in Appendix A.

In Fig. 6.2 and 6.3 the performance of the RNN is summarized. One can observe that the $E(\pi^0)$ resolution is similar for the RNN and PanTau. Nevertheless, there is still a bias for the RNN. A bias on the $E(\pi^0)$ prediction, however, can be calibrated, so in the following, just the energy resolution will be considered. The behavior does not differ much for the two bins of $d(\pi^0, \pi^\pm)$ ($E(\pi^0) > 25$ GeV and $E(\pi^0) \geq 25$ GeV). The angular resolution is significantly worse for the RNN. Especially in the high energy bin, PanTau outperforms the RNN. This is not surprising as several aspects of the RNN training pipeline were not optimized yet.

6.3 Target Transformation and Batch Normalization

As described in Chapter 3, batch normalization and a proper transformation of the target and feature distributions are highly recommended when training a neural network. Therefore, batch normalization is added after the merge of the four branches of the RNN. The feature distributions are already being standardized as described in Section 5.3. When considering the target distributions, there is still room for improvement, though, as visible in Fig. 6.4. In comparison to the angular distributions, the distribution of the p_T ratio is not only very unlike a Gaussian, but also has a obviously much higher median and has a significant amount of outliers.

This can be corrected by using a *quantile transformer* [56]. This transforms any input distribution into a Gaussian using its quantiles. The resulting target distributions can be observed in Fig. 6.5. As can be seen in Fig. 6.6 and 6.7, the changes significantly improve the performance. The RNN energy resolution is now significantly better than PanTau and the RNN angular resolution now matches the performance of PanTau. In the low energy bin, it outperforms PanTau.

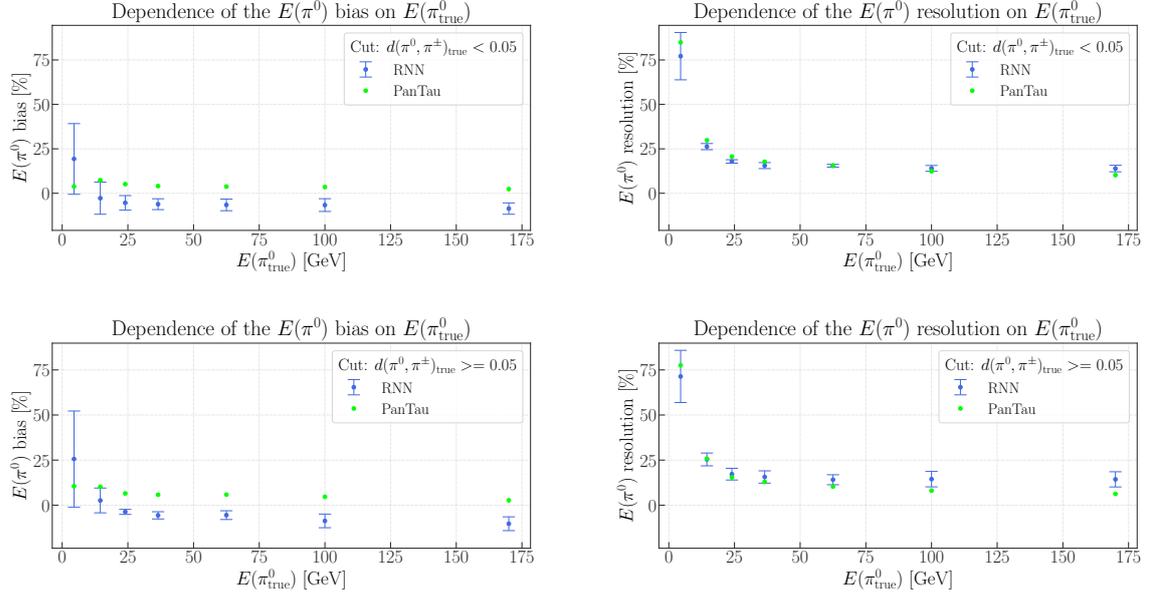


Figure 6.2: Energy resolution and bias for the RNN and PanTau in two bins of $d(\pi^0, \pi^\pm)$. The error bars denote the average of three training runs with different Tensorflow seeds.

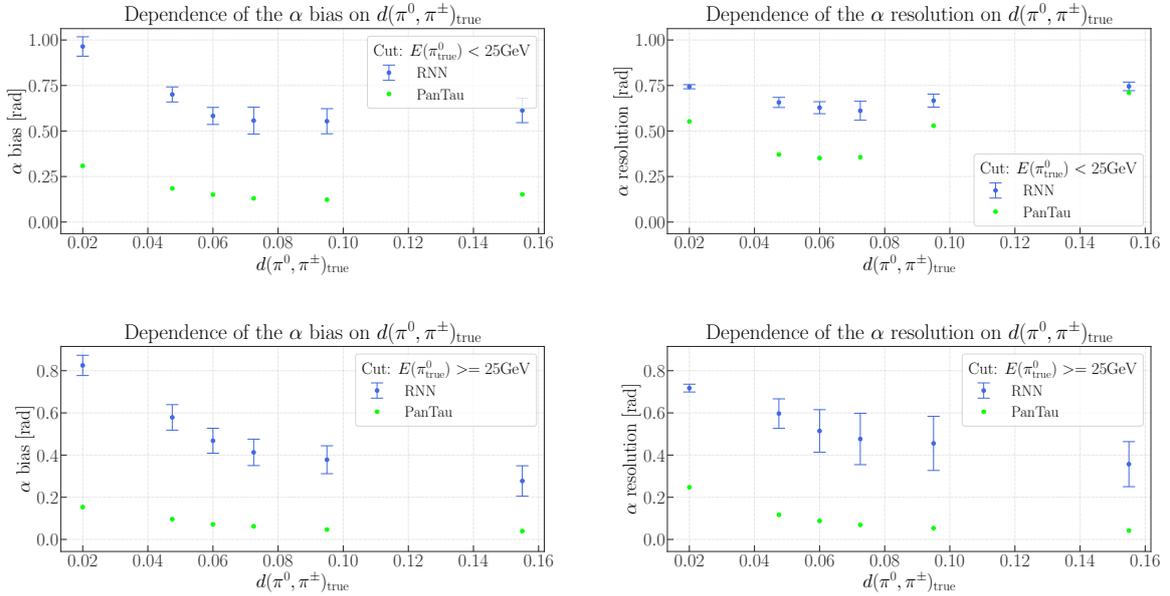


Figure 6.3: $\alpha(\pi^0, \pi^\pm)$ resolution and bias for the RNN and PanTau in two bins of $E(\pi_{\text{true}}^0)$. The error bars denote the average of three training runs with different Tensorflow seeds.

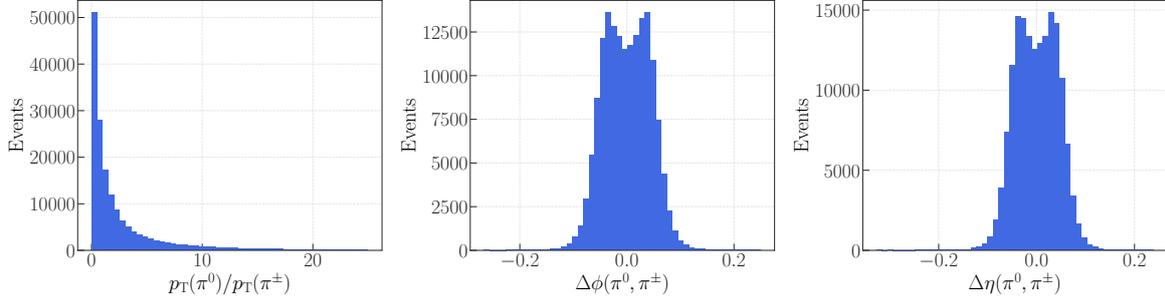


Figure 6.4: Distribution of the targets of the neural network.

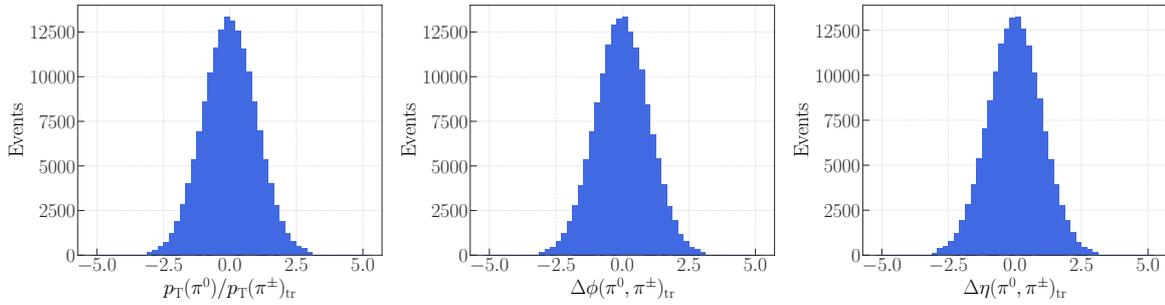


Figure 6.5: Distribution of the targets of the neural network after applying the quantile transformation.

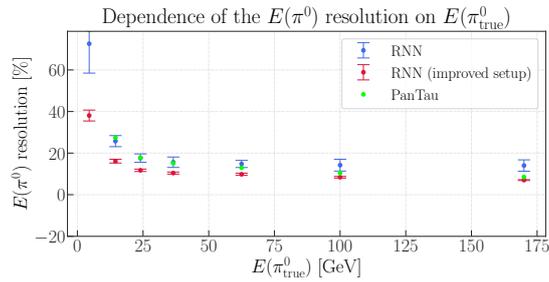


Figure 6.6: Energy resolution for PanTau and the RNN with improved and basic setup in two bins. Improved setup means the addition of batch normalization and a target Gaussian transformation. The error bars denote the average of three training runs with different Tensorflow seeds.

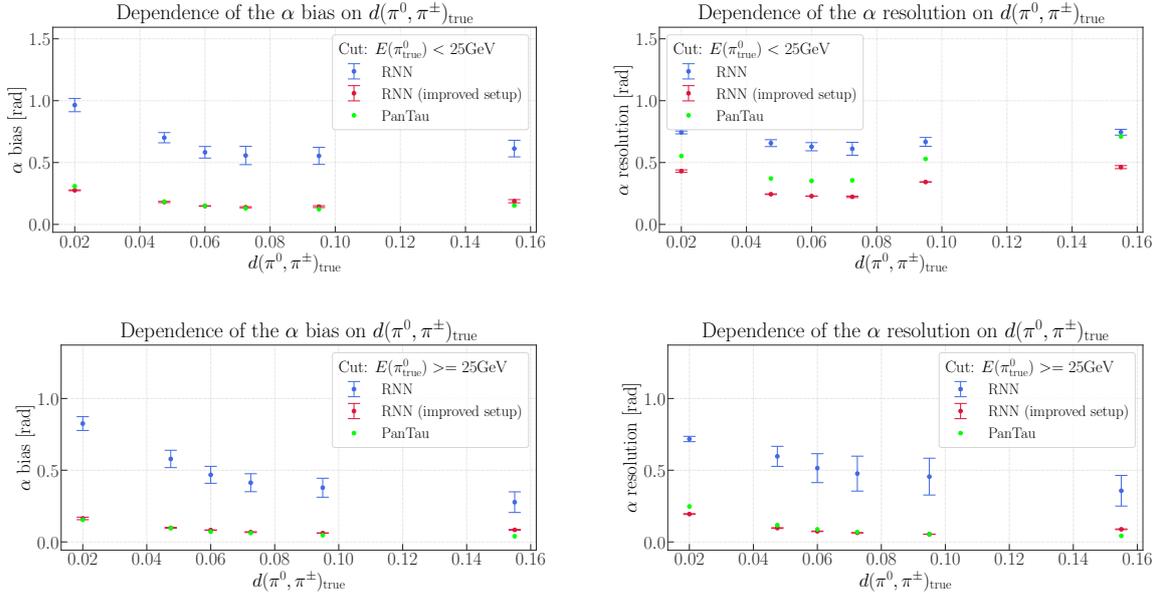


Figure 6.7: $\alpha(\pi^0, \pi^\pm)$ resolution and bias for PanTau and the RNN with improved setup in two bins of $E(\pi^0_{\text{true}})$. Improved setup means the addition of batch normalization and a target transformation. The error bars denote the average of three training runs with different TensorFlow seeds.

6.4 Implementing the DeepSet NN

In Section 5.2, it was explained that from the physics perspective, considering the order of elements passed to the neural network is not necessary. As the RNN still considers the order, computational performance can be improved by using a different approach that allows passing sets of vectors (instead of ordered sequences) to the network. In [43] the DeepSet architecture was developed that fulfills this requirement. In [14] it is used to improve τ decay mode reconstruction. A significantly lower training time was achieved through this. The same is now implemented for the regression neural network in this analysis. The new architecture can be seen in Fig. 6.8. Applying the concept from Section 3.4.3, the DeepSet functionality is implemented by replacing the LSTM nodes with two separate feed forward neural networks (κ and λ) and a summation in between in each branch. The prediction performance of the DeepSet NN is summarized in Fig. 6.9 and 6.10. It does not differ significantly from the RNN. Just in the high energy bin, the resolution is slightly worse.

A major improvement, however, is achieved in terms of computational performance: Fig. 6.11 shows the validation loss against training time. It is obvious that the minimum loss is reached already after less than half of the duration. This could be very helpful in future analyses with much larger datasets.

6.5 Assessing the Loss Function

When predicting the neutral pion kinematics, there is a trade-off between optimizing energy and angular resolution. Both are required for CP measurements as described in Section 1.4. To understand, if there

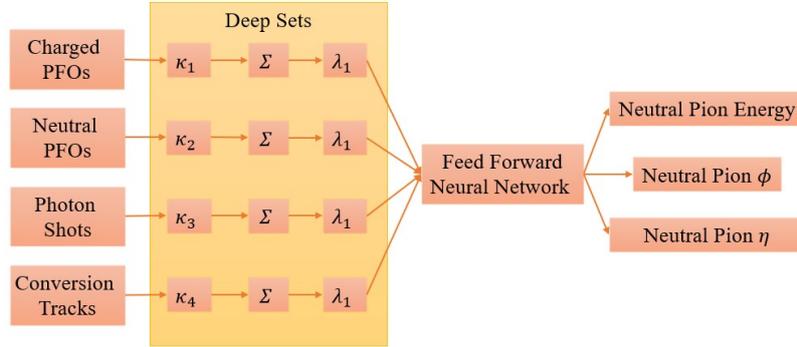


Figure 6.8: Architecture of the used DeepSet NN. κ_i and λ_i are feed forward neural networks. Σ denotes a summation over the outputs of κ_i .

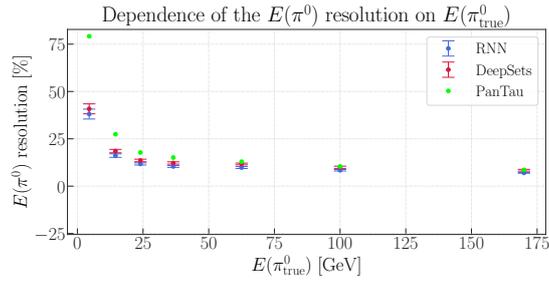


Figure 6.9: Energy resolution for the DeepSet NN, the RNN and PanTau. The error bars denote the average of three training runs with different TensorFlow seeds.

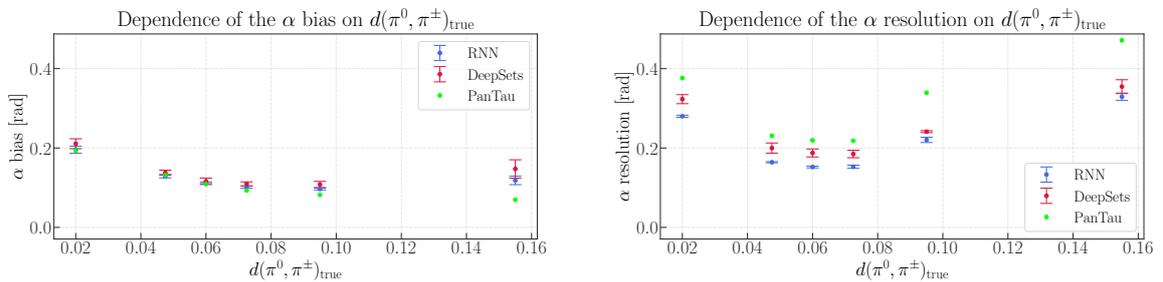


Figure 6.10: $\alpha(\pi^0, \pi^\pm)$ resolution and bias for the DeepSet NN, the RNN and PanTau in two bins of $E(\pi^0_{\text{true}})$. The error bars denote the average of three training runs with different TensorFlow seeds.

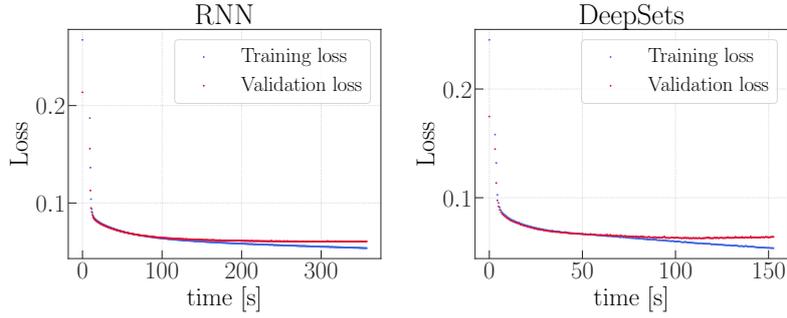


Figure 6.11: Validation and training loss against training time for the RNN and the DeepSet NN.

is room for improvement here, the mean squared error loss function is altered via Eq. (3.4) to

$$\text{Loss} = a \cdot (\Delta p_T)^2 + b \cdot (\Delta\phi)^2 + c \cdot (\Delta\eta)^2 \quad (6.2)$$

with the adjustable constants a , b and c . In Fig. 6.12 and 6.13, scenarios with a very high weight for the angles or for the transverse momentum are shown.

The expected behavior that if the angular error is weighted stronger, the energy resolution worsens and the angular resolution improves and vice versa is only partly fulfilled. The effect can only be observed for the energy resolution that worsens for larger b and c . In conclusion, changing the loss function

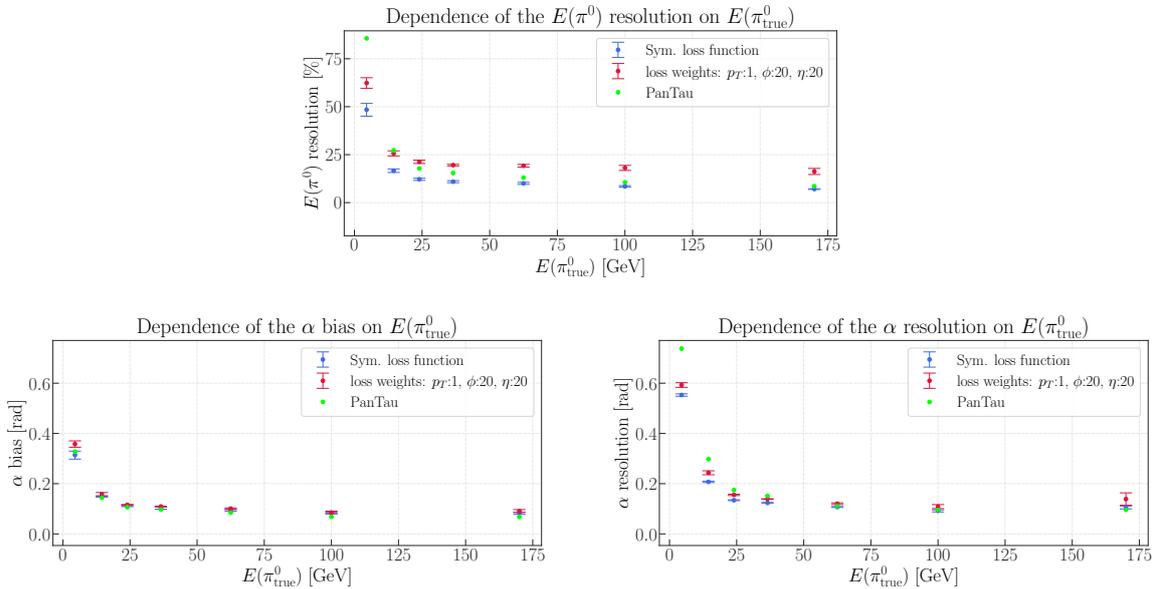


Figure 6.12: Performance of the ANN for a large weight of the angles in the loss function.

does not affect the performance strongly. This could be due to the fact that the architecture needs to be optimized first or that the data does not contain any more information. If this is solved, it could be worth

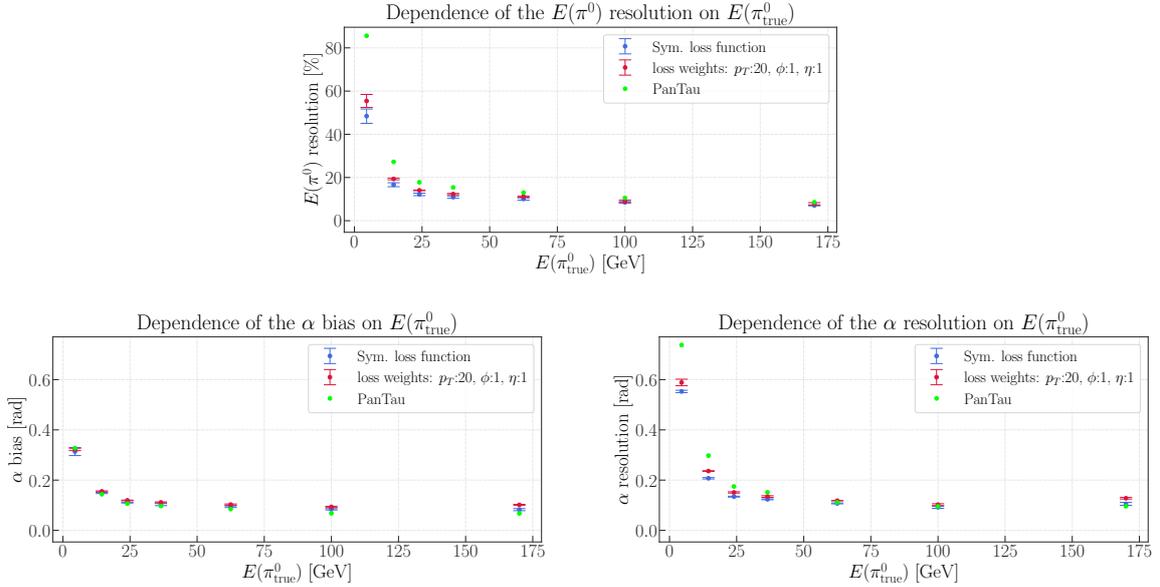


Figure 6.13: Performance of the ANN for a large weight of the transverse momentum in the loss function.

including physical requirements into the loss function. For example, the 1p1n decay is mediated by the ρ particle. Therefore, the distribution of the invariant mass of the neutral and charged pion should be centered around the Standard Model value of the ρ mass. Adding a term that penalizes a deviation from this value could help to improve the performance.

6.6 Hyperparameter Optimization

The overall structure of a neural network can largely influence its performance. A model with low complexity might not be able to learn all information that the data contains, and a too complex model can lead to overtraining and long training durations. In this section, the architecture of the neural network as well as the training parameters are assessed.

Hyperparameter optimization is computationally expensive. The choices on the considered parameter ranges in this chapter are all a compromise of considering all reasonable parameter options and the availability of computational resources. Furthermore, ideally, the network architecture and the training configuration would be optimized together. However, especially considering wide ranges of batch sizes and learning rates can lead quickly to very long training durations and large RAM allocation. Therefore, the optimization is split into a grid search (Section 3.6.1) on the training parameters and a Bayesian optimization (Section 3.6.2) of the architecture.

6.6.1 Grid search on Training Parameters

The training configuration is optimized by performing a grid search on batch size and learning rate. The choices are listed in Table 6.1. The result can be observed in Fig. 6.14, in which for each of the nine combinations the validation loss and the training loss are considered. For all combinations a minimum

Table 6.1: Parameters and choices considered during the training parameter grid search.

training parameter	search options
Batch size	$[2^8, 2^{12}, 2^{14}]$
Learning rate	$[10^{-5}, 10^{-4}, 10^{-3}]$

Table 6.2: Architecture parameters on the corresponding ranges considered in the Bayesian optimization.

hyperparameter	search range
Number of layers in κ	[1, 8]
Number of units in κ	[8, 64]
Number of layers in λ	[1, 8]
Number of units in λ	[8, 64]
Number of final dense layers	[1, 8]
Number of units in each final dense layer	[8, 256]
Internal activation function	tanh, ReLu, sigmoid

in validation loss around is reached. The number of epochs is chosen high enough that this is the case for all combinations. Of course there is the possibility, that there exist further minima at a larger number of epochs, but testing this would require large amounts of computational resources, so this compromise is chosen. For all combinations, the minimum is located around 0.06. With the assumption that with a lower validation loss, the performance in energy and angular resolution improves too, it can be concluded that with all configurations optimal results in the reconstruction can be reached. To not use unnecessary computing resources, it makes sense to consider the training duration until the minimum is reached. As expected, this is the case for large batch sizes, so a batch size of 2^4 is chosen. The training converges most smoothly with low learning rates, which was expected as well. Accordingly, a learning rate of 0.001 is selected.

6.6.2 Bayesian Optimization

The optimized hyperparameters and their corresponding ranges are listed in Table 6.2. The number of calls was set to 100. The maximum number of epochs per call is set to 500 to leave enough time for convergence. The development of the minimum validation loss can be observed in Fig. 6.15. No convergence can be observed, just a lower threshold with some outliers above.

The lowest validation loss was reached already in call number 4. The architecture configuration is summarized in Table 6.3 As it was already observed, that the performance significantly changes with different TensorFlow seeds, the possibility exists that it is a coincidence that this configuration yields the lowest validation loss. The result is compared to the baseline in Fig. 6.16 and 6.17. No significant improvement can be observed.

To exclude that the number of epochs was set to low (undertraining), a criterion on convergence is imposed: If the validation loss after 500 epochs does not deviate more than one percent from the validation loss 20 epochs before, it is considered converged. If it is higher, it is considered diverged. According to this definition, for all calls the training either diverged or converged, so choosing more

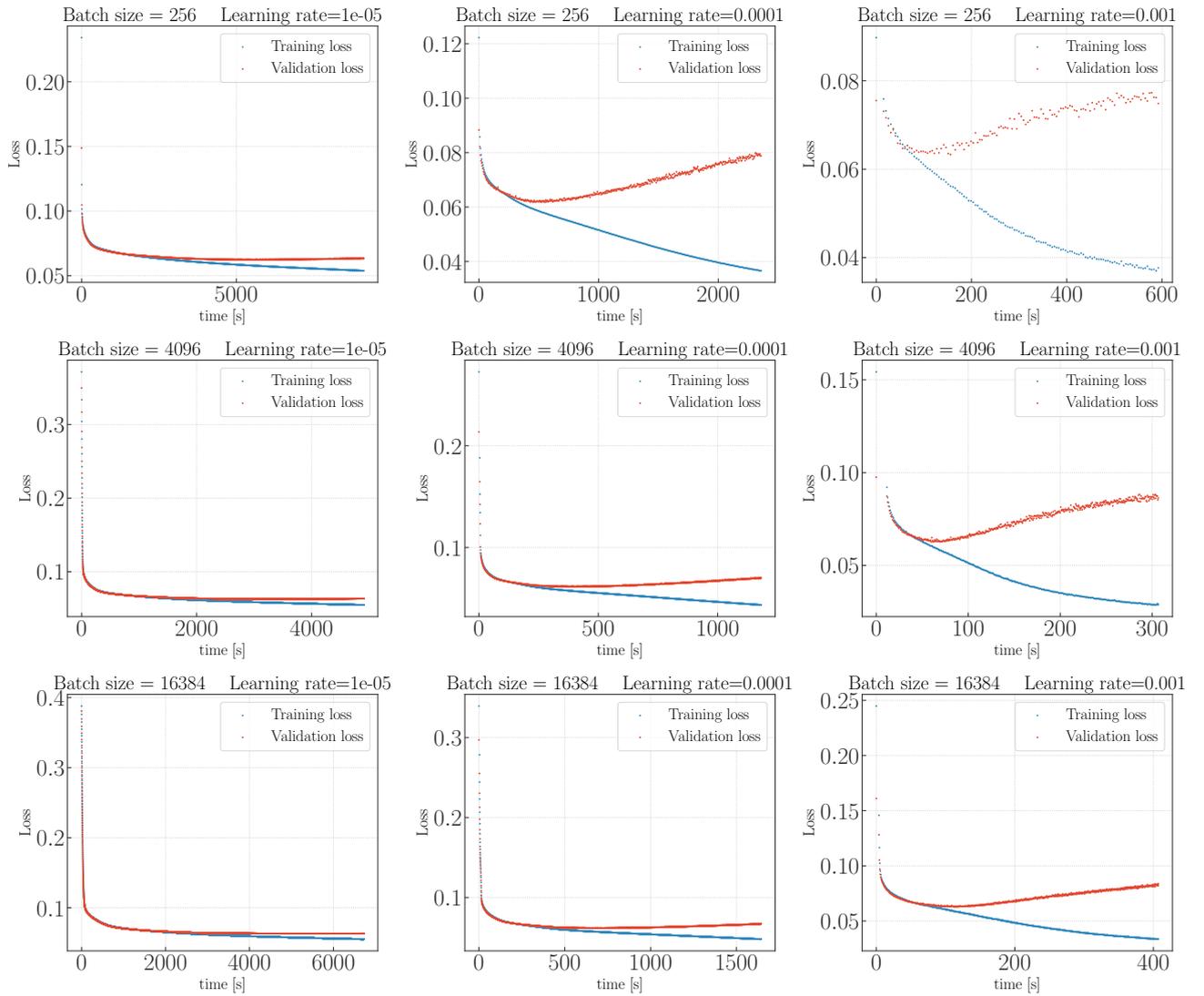


Figure 6.14: Learning curves for different configurations of batch size and learning rate. The x-axis is displayed in units of time instead of number of epochs

Table 6.3: Best performing configuration of architecture parameters received from the Bayesian optimization.

Hyperparameter	Best value
Number of layers in κ	7
Number of units in κ	22
Number of layers in λ	2
Number of units in λ	9
Number of final dense layers	3
Number of units in each final dense layer	37
Internal activation function	tanh

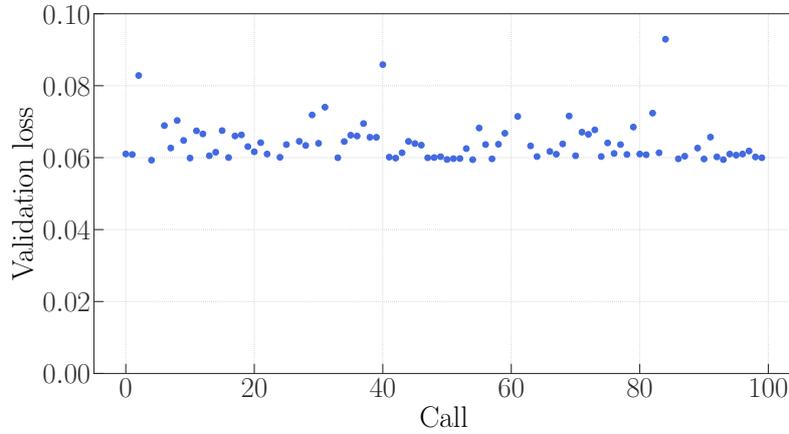


Figure 6.15: Minimum validation loss against call of the Bayesian hyperparameter optimization.

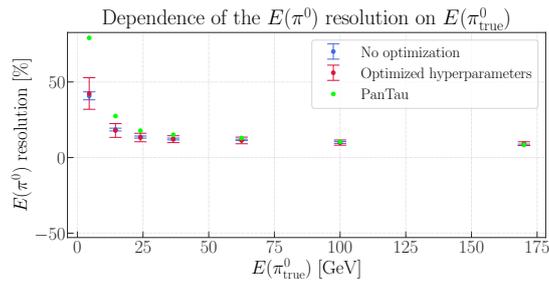


Figure 6.16: Energy resolution for the ANN with and without optimized hyperparameters. The error bars denote the average of three training runs with different TensorFlow seeds.

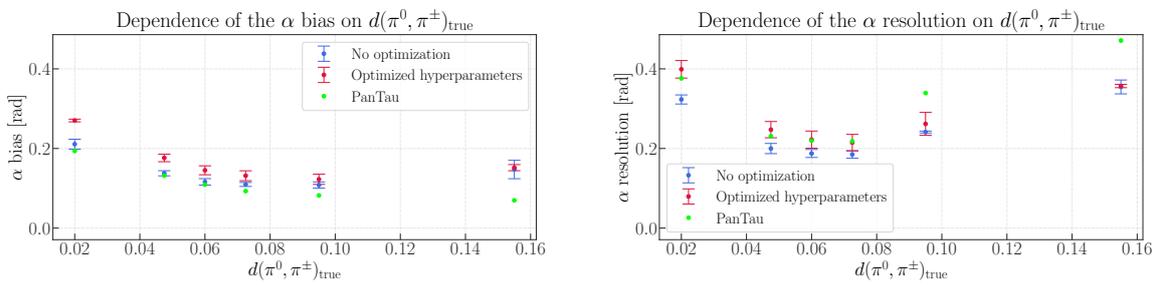


Figure 6.17: $\alpha(\pi^0, \pi^\pm)$ resolution and bias for the ANN with and without optimized hyperparameters. The error bars denote the average of three training runs with different TensorFlow seeds.

epochs would not improve the result. As an example, the learning curve (validation loss against epoch) for call one can be observed in Fig. 6.18. Consider Appendix A for all of them.

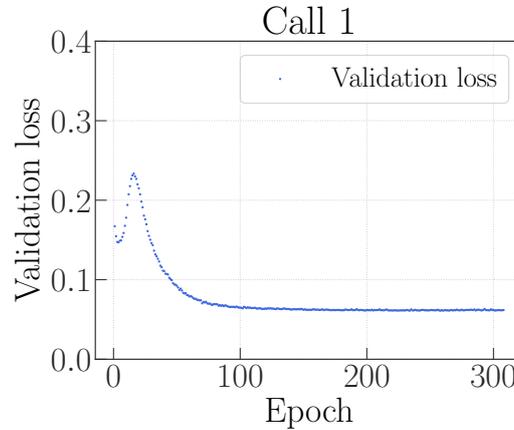


Figure 6.18: Learning curve for call one of the hyperparameter optimization.

Another reason for no further improvement during the optimization could be the high complexity of the parameter space and a not sufficiently high number of calls. One could investigate correlations between architecture parameters and minimum validation loss and perform more calls to address this issue.

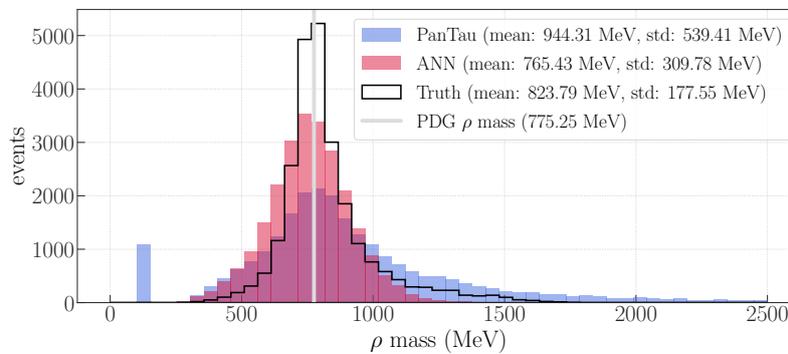


Figure 6.19: Display of the ρ meson mass distribution from the 1p1n decay for PanTau, the optimized neural network and truth data. The Particle Data Group (PDG) average is depicted as well [2].

6.7 Chapter Summary: Links between Photon Shots and Neutral PFOs

To summarize, it is worth comparing the predicted ρ mass distribution from Section 4.4 to the one predicted by the neural network. This can be seen in Fig. 6.19. As already discussed, the neural network outperforms PanTau in terms of energy resolution. As the invariant mass of the two pions depends

strongly on their energies, it is not surprising that the distribution for the neural network is more narrow. Especially at the tail to high energies.

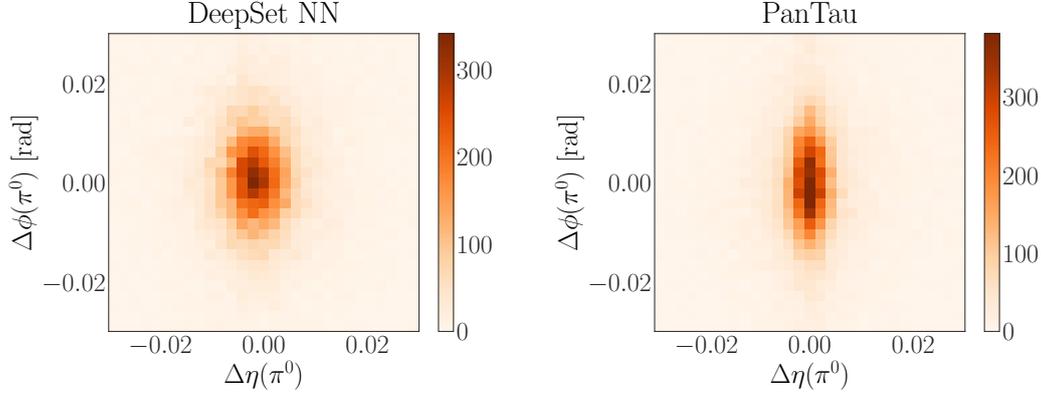


Figure 6.20: 2D residual distribution plots PanTau (right) and the optimized neural network (left).

A comparison of the final performance in terms of angular resolution is illustrated in the 2D residual plots in Fig. 6.20. The performance of the neural network in ϕ resolution is a bit stronger. However, the shape of the η residual distribution appears to be much more narrow for PanTau. To further investigate this effect, the η residual distribution is depicted in Fig. 6.21 with a logarithmic y-axis. It is obvious that the performance of the neural network is worse around the center of the distribution. Towards the tails, the opposite accounts. The stronger performance of PanTau for most of the events in η resolution could possibly result from more information in the strip layer of the ECAL. As shown in Fig. 2.5, the strip layer has high granularity in the η direction. This effect can be explained as follows: As already

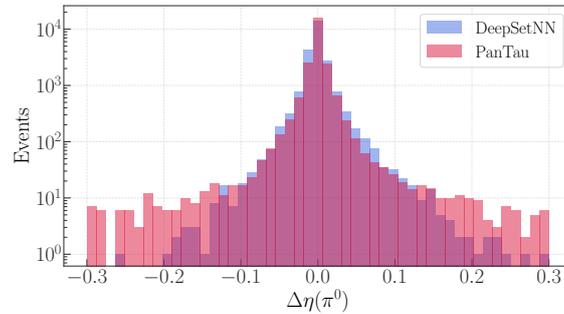


Figure 6.21: Graphical display of the $\eta(\pi^0)$ residual distribution for PanTau and the neural network with a logarithmic y-axis.

described in Section 4.4, the PanTau algorithm exploits information on ECAL cell level to link photon shots and individual Neutral PFOs. The neural network input features do not contain this information. The locations in η and ϕ of the Neutral PFOs do not include the structure of the clusters, so only with this information, it can not be said clear, if a Photon Shot can be associated with a Neutral PFO. Extracting this information again from upstream datasets could help increase especially the η resolution

in future analyses. Furthermore, it is quite promising that the neural network is able to reach an overall performance competitive with the PanTau algorithm, although it uses significantly less information.

Validation on LHC Data

Up to this chapter of the thesis, the verification of the prediction performance has been assessed on data generated by Monte Carlo simulations (Section 2.5) according to the Standard Model. This had the great advantage that the labels have always been provided: True decay mode and true neutral pion four-vector were obtained from the simulation as well. It is essential, however, to verify this simulation on real data taken by the ATLAS detector. In order to achieve this, once again, data obtained via the tag-and-probe method is used. The accordance of real data and simulation is compared in the following with the framework `sif` that runs on top of the `HAPPY` framework. As estimating systematic uncertainties would require a much larger effort, they are not considered in the following.

7.1 Cut Selection

The cut selection for this chapter is listed in Table 7.1 and matches the signal region from the 2017 τ reconstruction analysis described in [53].

Table 7.1: Cuts applied in the comparison of data and simulation. Consider [53] for further explanations.

-
- trigger: Single-muon with $p_T > 20$ GeV
 - at least one medium $\tau_{\text{had-vis}}$ with one or three tracks
 - muon and tau lepton have opposite charge
 - no b -tagged jets
 - $m_T(\mu, E_T^{\text{miss}}) < 50$ GeV
 - $\sum \cos \Delta\phi(\tau_{\text{had-vis}}, E_T^{\text{miss}}) > -0.15$
 - $45 \text{ GeV} < m_{\text{vis}}(\mu, \tau_{\text{had-vis}}) < 85 \text{ GeV}$
 - $\Delta\phi(\mu, \tau_{\text{had-vis}}) > 2.4$
-

7.2 Overall modelling

7.2.1 General Observables

To start with, general observables concerning the event and the reconstructed τ are investigated as can be observed in Fig. 7.1. Each variable is plotted in a double plot: The upper plot represents a histogram of the real data (black marker with a striped area representing the statistical error) as well as different processes generated via simulation. These processes mostly include true $Z \rightarrow \tau\tau$ decays, Z decays to two leptons and decays to a W boson with additional jets. An estimate for processes that were falsely reconstructed ditau decays (both τ lepton have *Same Sign* charge) are displayed as well. Other processes with rather low incidence in this region are reconstructed Z decays from fake muons, processes involving top quarks and processes involving two vector bosons. In the lower plot, the ratio of the data and the sum of the simulated events in each bin is displayed. For ϕ , η and the transverse momentum of the τ lepton, data and simulation do not deviate by more than 10%. One can nicely observe the lower selection cut on the transverse momentum at $p_T(\tau_{\text{vis}}) = 20$ GeV. The number of average interactions per bunch crossing is mostly modelled accurately. Some deviations are visible towards higher values.

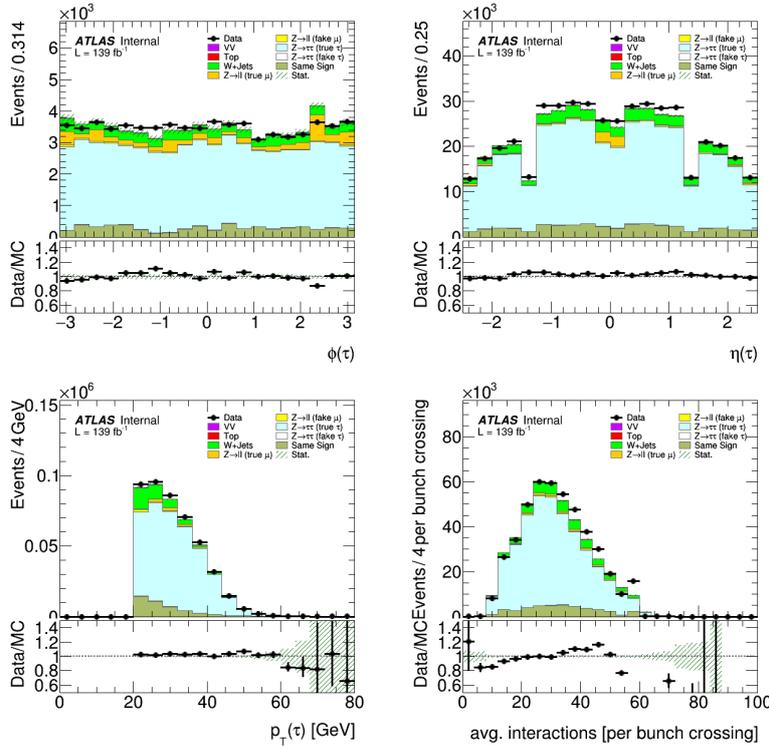


Figure 7.1: Comparison plots between data and simulation for general observables. All decay modes included.

7.2.2 Neural Network Features

In a next step, the modelling of the neural network input features is assessed. They were introduced in Section 5.1.1. This marks an important step as missmodelling on these variables can negatively

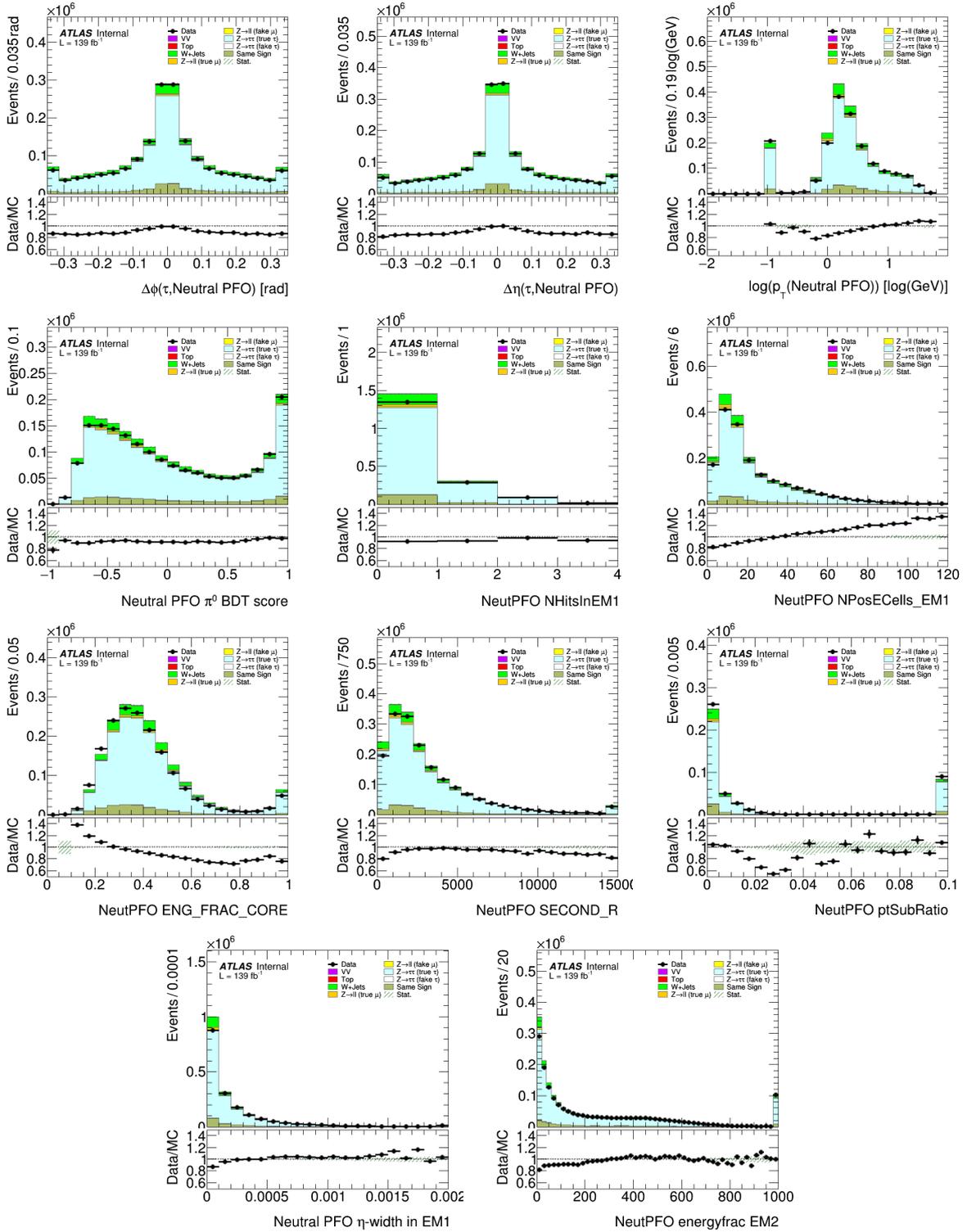


Figure 7.2: Comparison plots between data and simulation for Neutral PFO variables. All decay modes included.

influence the neural network performance. In Fig. 7.2, it can be observed that most variables concerning the Neutral PFOs are modelled accurately. The logarithm of the transverse momentum is not ideally modelled towards lower values. However, this only applies for transverse momenta of a few GeV. For the number of positive cells in the strip layer of the ECAL (NPosECells_EM1), the calibration is slightly false. The simulated distribution is shifted towards lower values. The opposite accounts for the fraction of the energy deposit in the ECAL contained in the center cell of the cluster (ENG_FRAC_CORE). Further slight missmodelling occurs for low values of the width in η in the ECAL strip layer (η -width in EM1) and the energy fraction in the second layer of the ECAL (energyfrac EM2). When observed

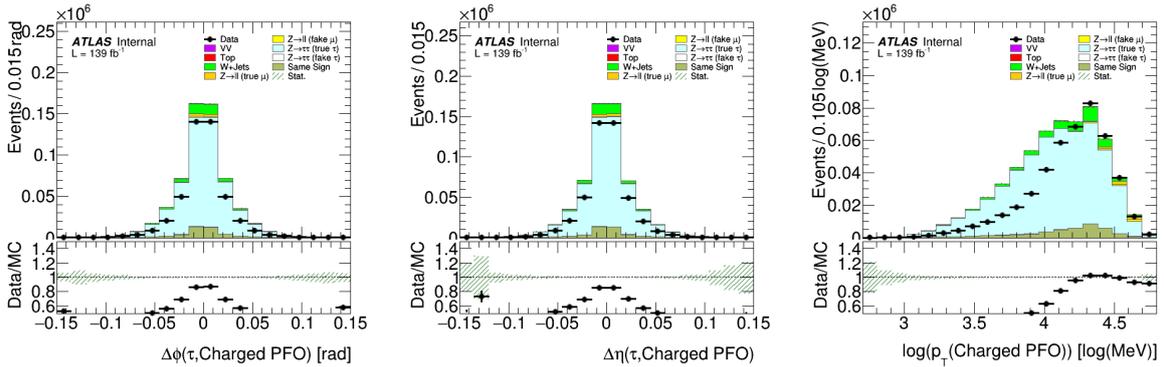


Figure 7.3: Comparison plots between data and simulation for Charged PFO variables. All decay modes included.

the modelling of the variables concerning the Charged PFOs in Fig. 7.3, it is obvious that there is an issue with the angular distances in η and ϕ . The same accounts for the logarithm of the transverse momentum. The modelling is inaccurate for transverse momenta lower than approximately 12 GeV. A

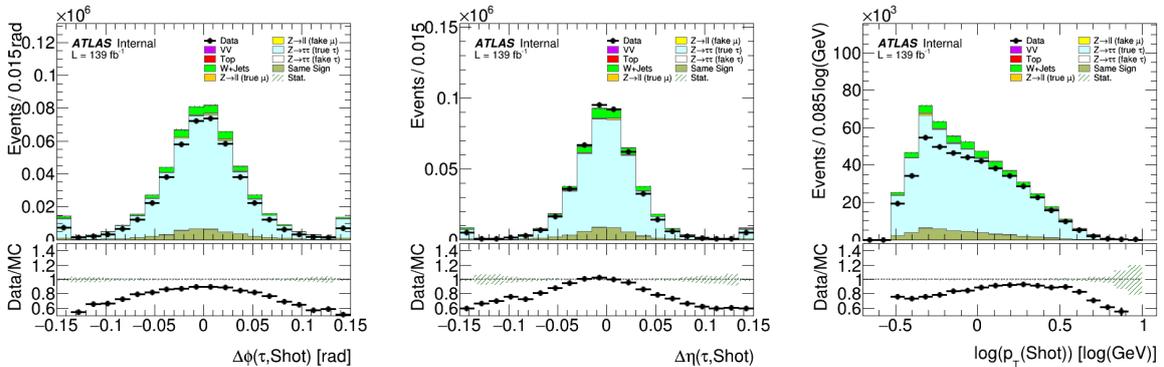


Figure 7.4: Comparison plots between data and simulation for the Photon Shot variables. All decay modes included.

similar effect can be observed for the logarithm of the transverse momentum of Photon Shots (Fig. 7.4) and Conversion Tracks (Fig. 7.5). Here, the data deviates from simulation in η and ϕ as well.

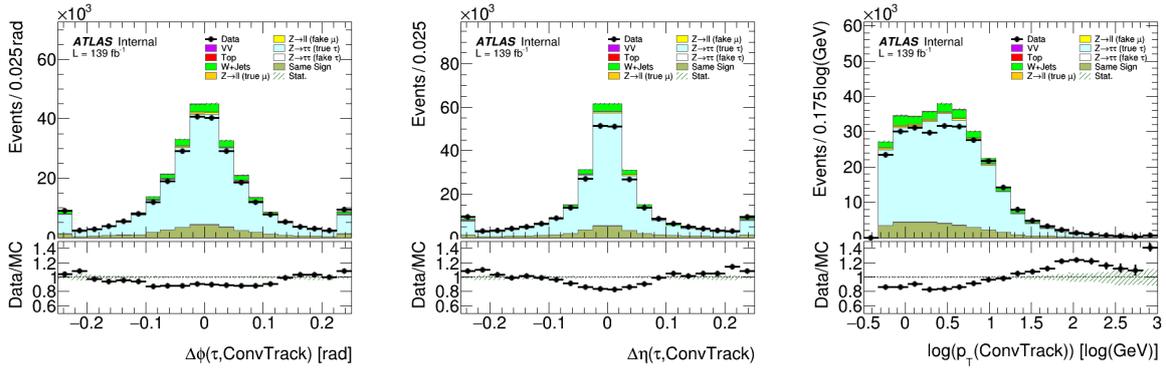


Figure 7.5: Comparison plots between data and simulation for the Conversion Track variables. All decay modes included.

7.3 Observables Based on the Neural Network Output

After having assessed the neural network input features, the output variables of the two neural networks considered in this thesis are investigated.

7.3.1 Decay Mode

The comparison of data and simulation for the decay mode predicted by the neural network described in Chapter 5 can be seen in Fig. 7.6. The modelling for reconstructed decay modes containing up to one neutral pion are accurately modelled. This does not account for the 1pXn and 3pXn decay mode, though.

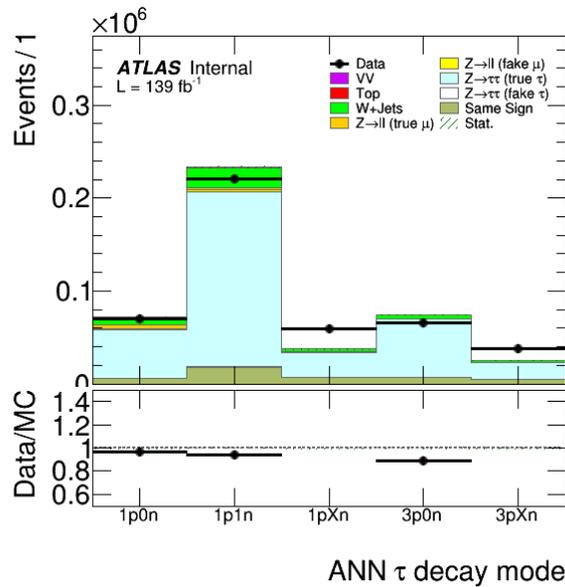


Figure 7.6: Comparison plots between data and simulation of the decay μ mode classification ANN. All decay modes included.

7.3.2 Neutral Pion Kinematics

In this section, variables related to the neutral pion four-vector predicted by the neural network optimized in Chapter 6 are discussed. The comparison plots for data and simulation can be observed in Fig. 7.7. The modelling for $\phi(\pi^0)$, the transverse momentum and the energy match well. The distributions for $\eta(\pi^0)$ and the angular distances between neutral and charged pion show missmodelling towards the center of the distribution.

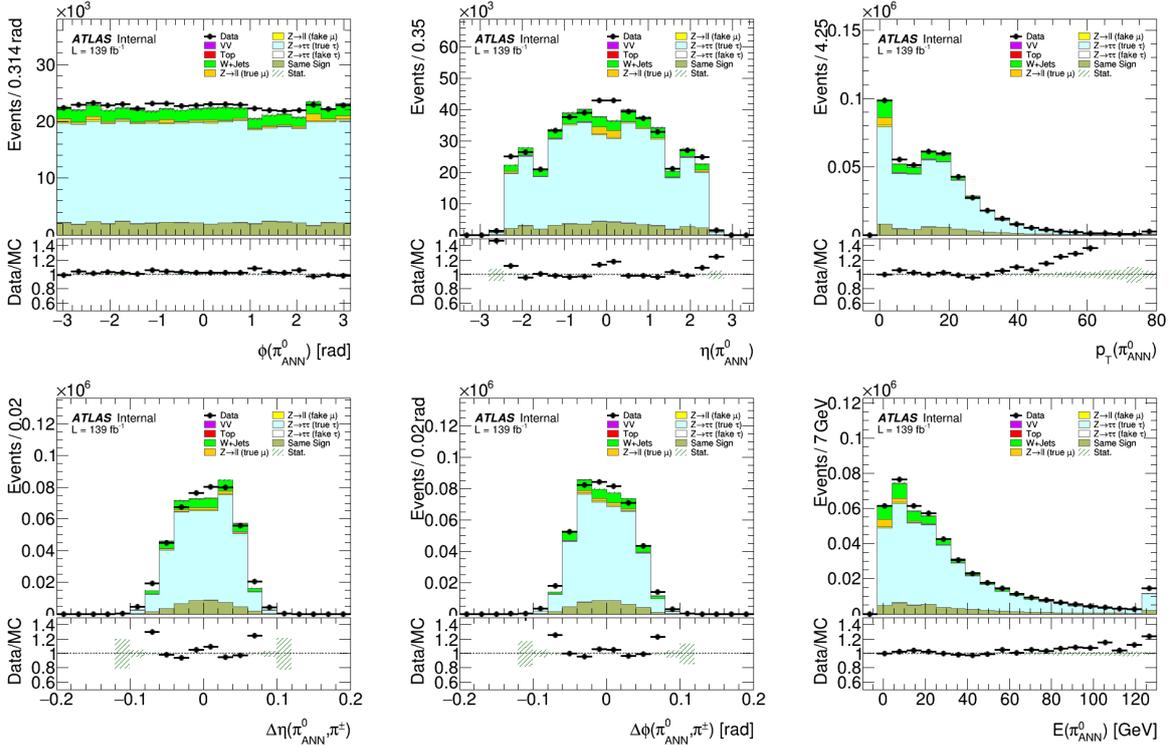


Figure 7.7: Comparison plots between data and simulation of variables predicted by the ANN. All decay modes included.

In Fig. 7.8, observables derived from the neutral pion four-vector are assessed. Here, strong deviations from the data can be observed for the energy asymmetry at values close to -1 . These are events with low energy of the charged pion compared to the neutral pion. The simulated distribution of the angular distance between the two pion $d(\pi^0, \pi^\pm)$ is too wide compared to the distribution of the data. The shape of the simulated distribution of the ρ mass matches the data, however, it is shifted towards higher masses.

7.4 Modelling of Exclusive Decay Modes

In a next step, the possibility of investigating the modelling of the introduced observables for exclusive decay modes predicted by the neural network was examined. This did not yield satisfactory results most likely due to software-related issues. The reason for error could not be found, however it was narrowed down as described in the following.

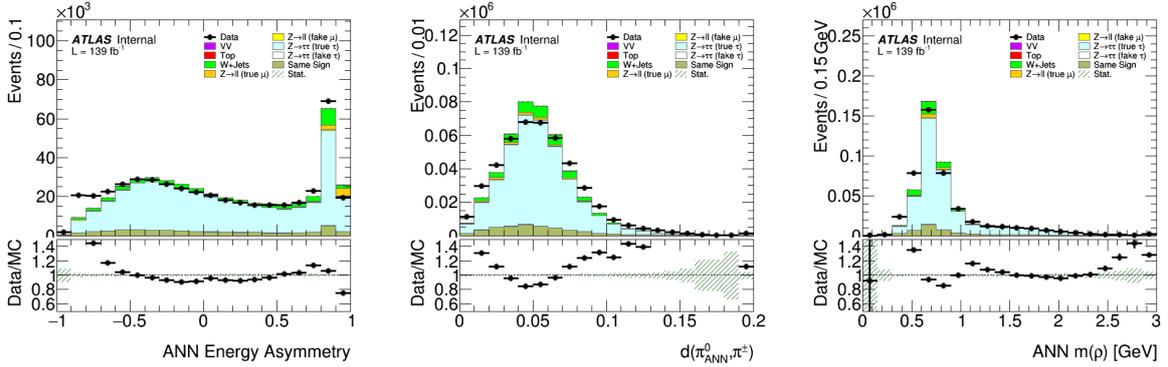


Figure 7.8: Comparison plots between data and simulation of observables derived from the ANN prediction for the π^0 kinematics. All decay modes included.

When plotting different variables exclusively for the $1p0n$ decay mode (Fig. 7.9), the disagreement between simulation and data is obvious. For this crosscheck, the decay mode predicted by PanTau and the number of charged tracks plotted. The simulated distributions show the expected behavior. E.g., there are only events with one τ track. This was assured by the correction introduced in Section 5.5. This does not account for the data and the same sign estimate, as the same sign estimate is determined by using the data distribution in the same sign region (not from the Monte Carlo simulation). The same plots for

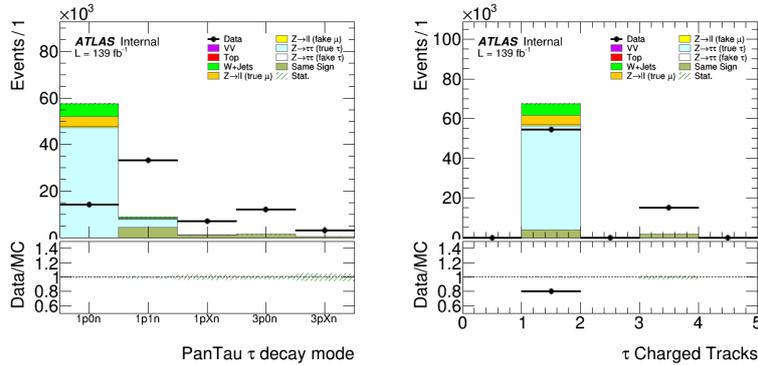


Figure 7.9: Comparison plots between data and simulation for additional observables for crosschecking. Exclusively events with decay mode $1p0n$ predicted by the ANN.

inclusive plotting are displayed in Fig. 7.10. One can observe in these plots that the shapes match the shapes of the data in the exclusive plots. This could be a sign that in the exclusive plots, random events are shown for the data. This is supported by the fact that observables that should behave the same way for all decay modes are not affected by the missmodelling, as to be seen in Fig. 7.11. A reason for the problem is most likely caused by a software-related error in the bookkeeping of the different datasets. The comparison plots for all decay modes and variables can be found in Appendix B.

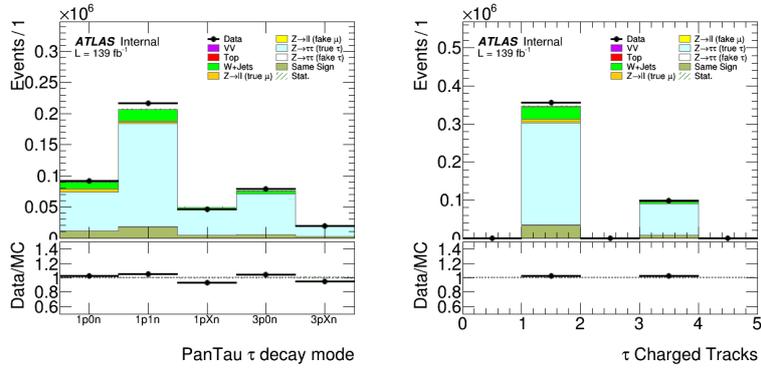


Figure 7.10: Comparison plots between data and simulation for additional observables for crosschecking. All decay modes selected.

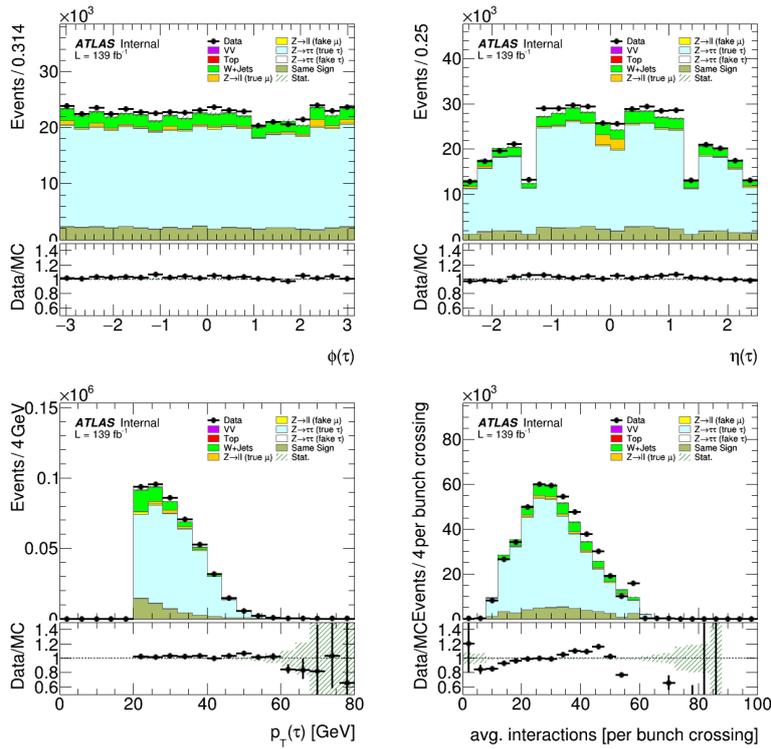


Figure 7.11: Comparison plots between data and simulation for different observables. Exclusively events with decay mode 1p0n predicted by the ANN.

Conclusion

In this thesis, the performance of a newly developed RNN predicting the τ decay mode was investigated and a separate artificial neural network for the prediction of τ decay product kinematics was optimized. The result was compared to the PanTau algorithm that was used previously to perform both tasks.

In order to achieve this, the structure of the PanTau algorithm was analyzed as a basis for estimating its performance compared to the two neural networks. In the next step, the newly developed RNN for the decay mode classification was analyzed. Its performance was compared to the performance of PanTau, once more validating the superiority of the RNN. The structure of this RNN was used in the following as a basis to develop a second artificial neural network that predicts the four-vector of the neutral pion in the $1p1n$ decay mode. Already before conducting extensive optimization steps, an energy resolution close to the performance of PanTau was reached. The strongest improvement in prediction performance was enabled by a Gaussian transformation of the target distributions. Apart from that, training duration was significantly reduced by implementing the DeepSet architecture into the network. Various methods were investigated to optimize the loss function, training configuration and architecture of the network, which, however, yielded no significant improvement in performance. An indicator for too little information in the training dataset. Moreover, by investigating the performance of the neural network and PanTau in terms of η resolution, evidence was found that the neural network input features are missing crucial information from the ECAL strip layer. Including this information could significantly improve the performance in the future. The performance of the optimized network was extensively compared to the PanTau result in different figures of merit, most importantly energy and angular resolution. Both were proven to be competitive with the PanTau reconstruction. Finally, a comparison between simulation and real data measured by the ATLAS detector yielded satisfactory agreement. Further improvements could be achieved by considering promising developments in other parts of the ATLAS τ reconstruction: A graph neural network was developed that improved τ identification as well as a convolution neural network on calorimeter cell level to improve the decay mode classification [57, 58].

The results of this thesis support the assumption that the neural network's prediction of the neutral pion four-vector could have a positive impact on future Higgs CP measurements in the ATLAS collaboration, an important step in the further validating the Standard Model or possibly discover an explanation for the matter-anti matter asymmetry in the universe. Moreover, this thesis proves once more the ability of deep learning methods to replace or even outperform classic machine learning algorithms. Together with the increasing amount of data in particle physics experiments and growing computational capacities, this insight could be used to significantly improve future particle physics analyses.

Neutral Pion Reconstruction: Additional Figures

A.1 PanTau algorithm performance compared to the DeepSet NN with Optimized Hyperparameters

Figure of Merit: $\Delta E(\pi^0)$

Binning: $p_T(\tau_{\text{vis,true}})$

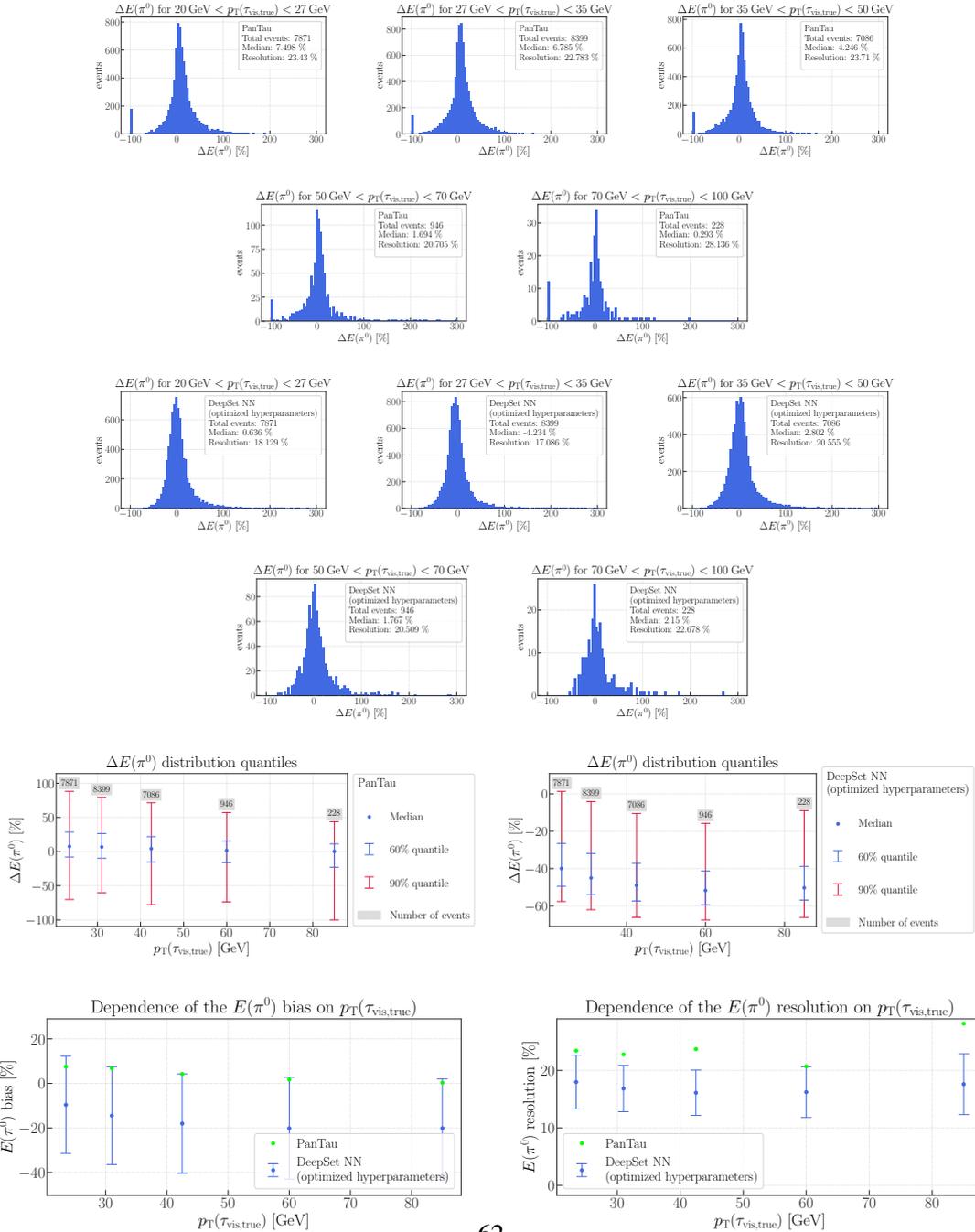


Figure A.1: $\Delta E(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi^0_{\text{true}})$, Cut: $d(\pi^0, \pi^\pm)_{\text{true}} \geq 0.05$

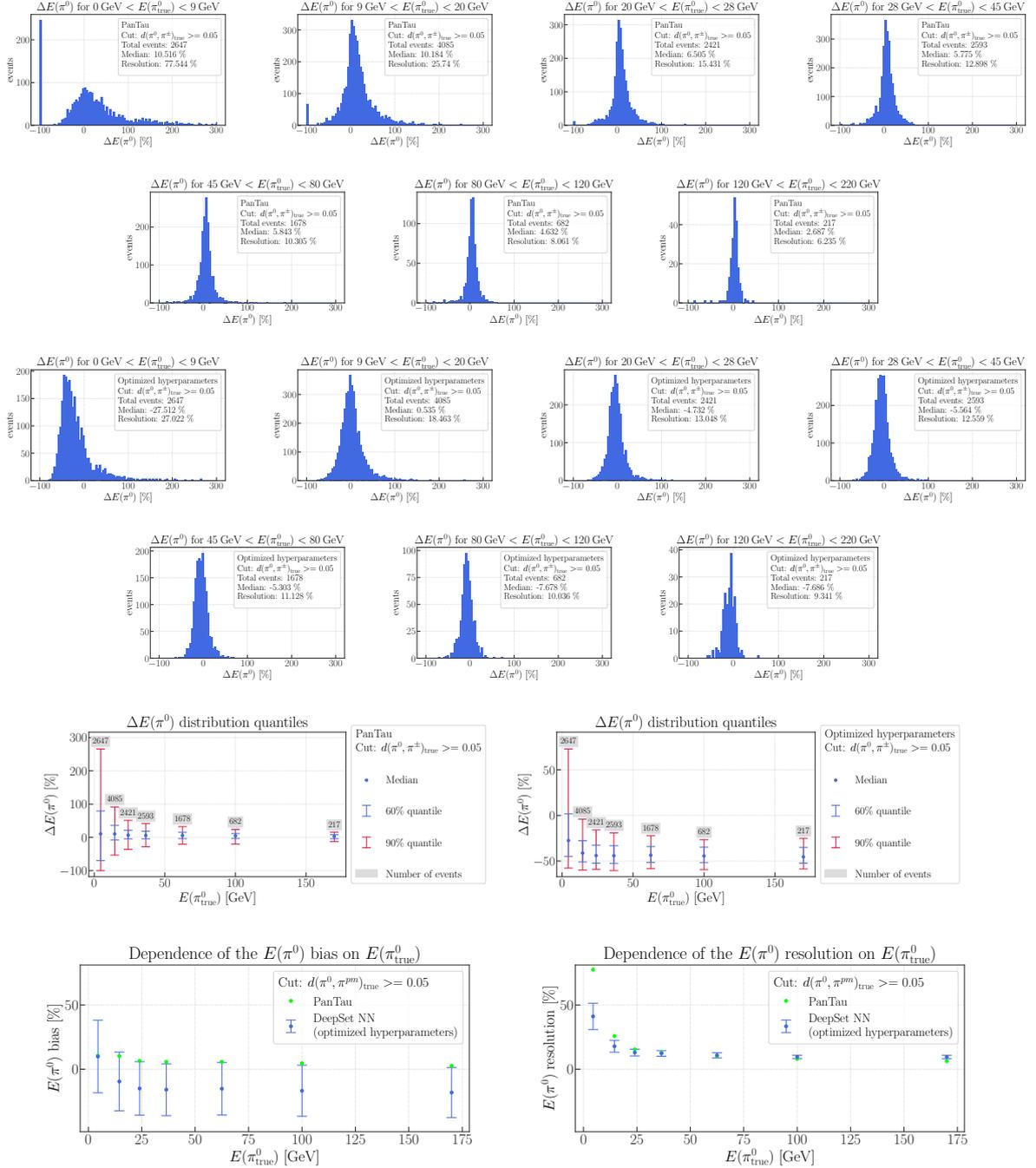


Figure A.2: $\Delta E(\pi^0)$ plots for different intervals of $E(\pi^0_{\text{true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $d(\pi^0, \pi^\pm)_{\text{true}} \geq 0.05$ is applied.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi^0_{\text{true}})$, Cut: $d(\pi^0, \pi^\pm)_{\text{true}} < 0.05$

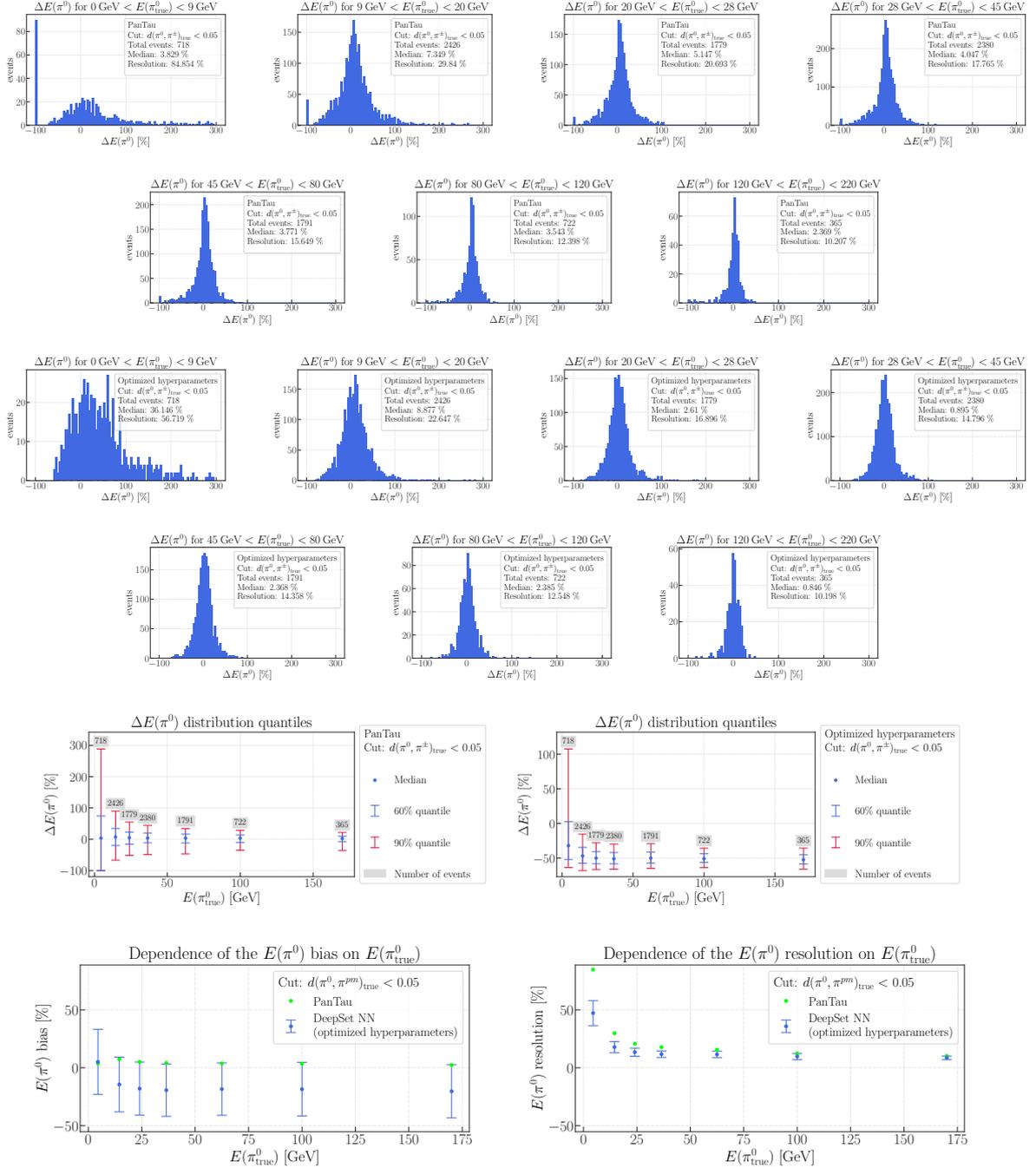


Figure A.3: $\Delta E(\pi^0)$ plots for different intervals of $E(\pi^0_{\text{true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $d(\pi^0, \pi^\pm)_{\text{true}} < 0.05$ is applied.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

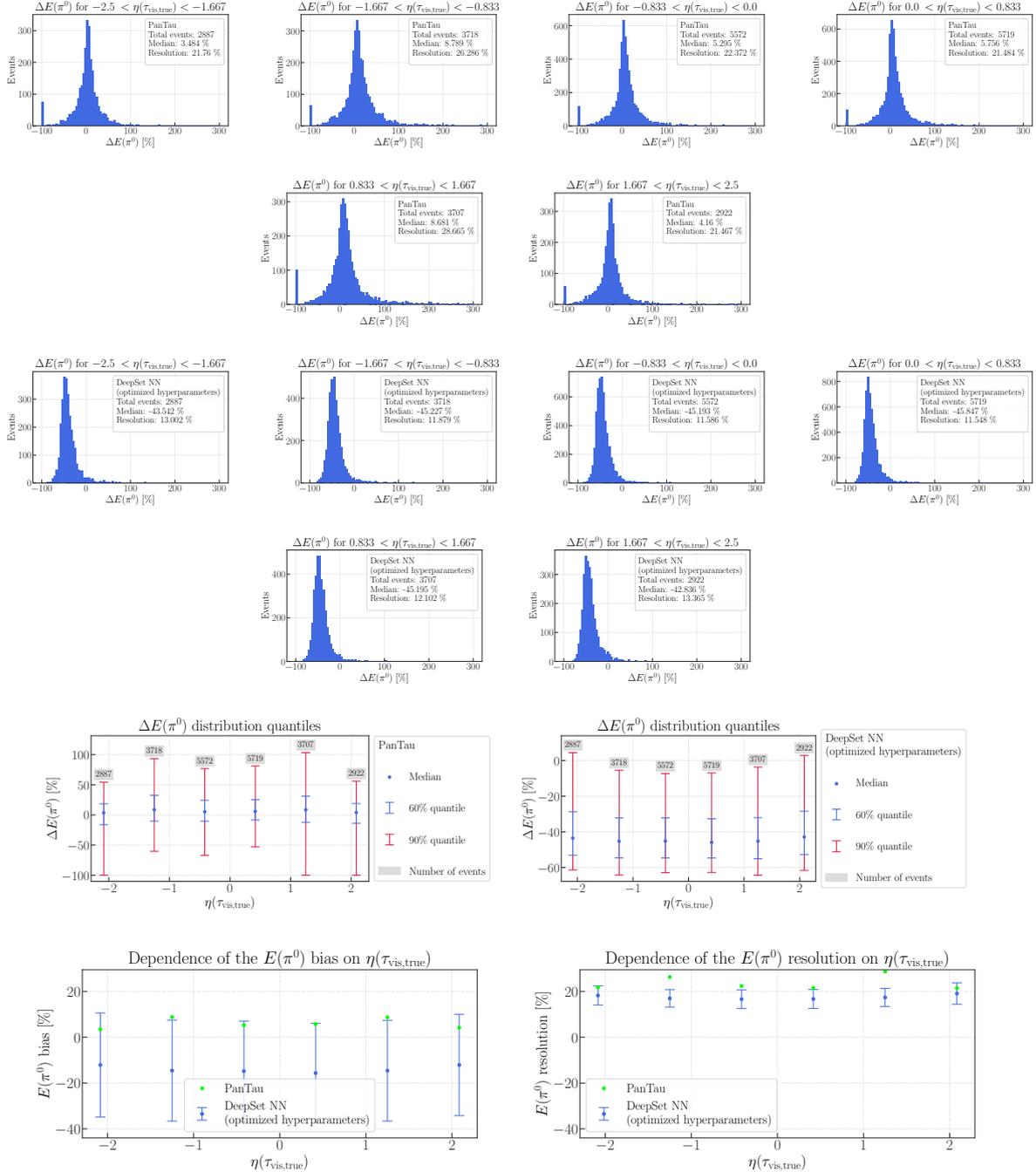


Figure A.4: $\Delta E(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

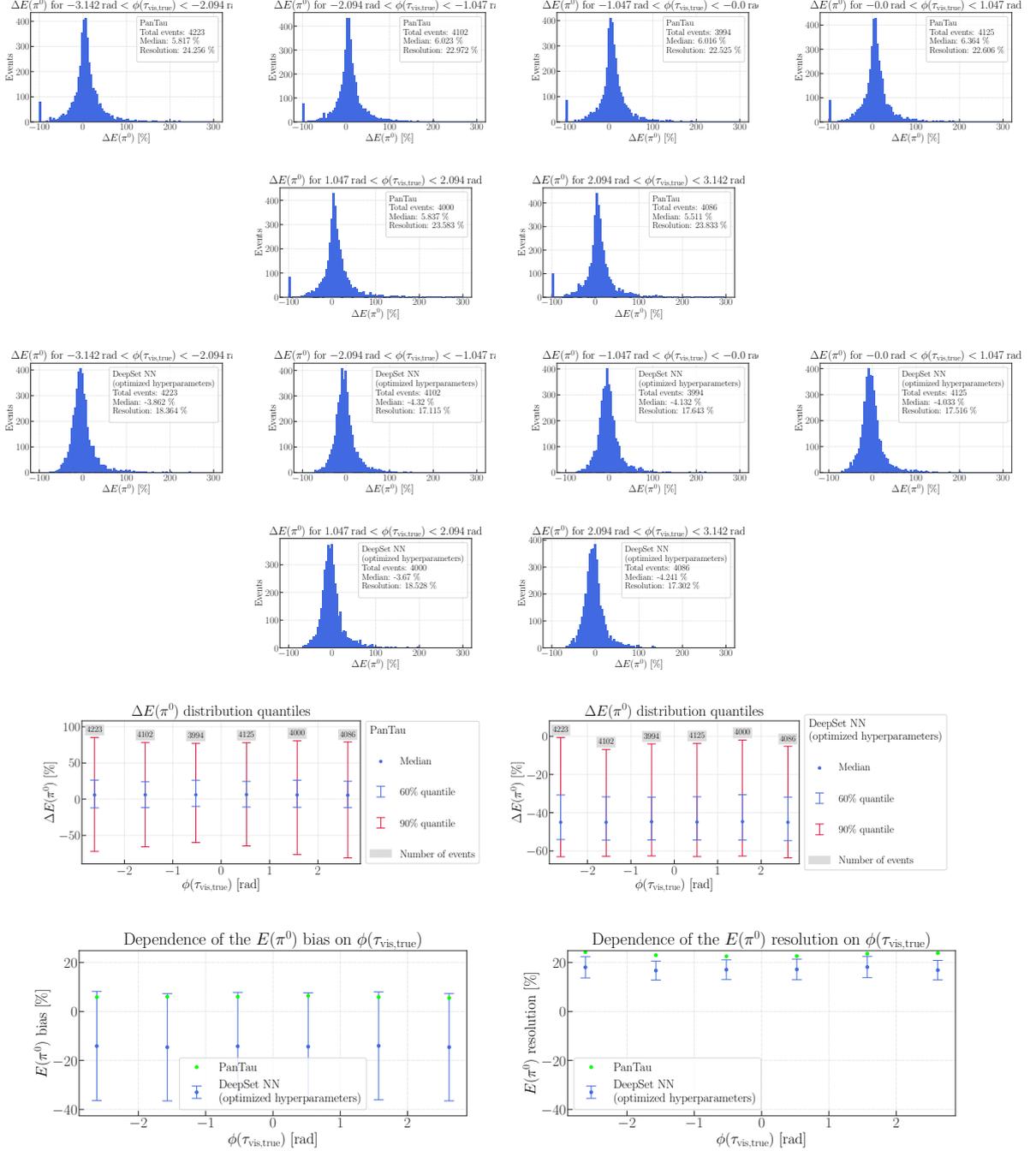
Binning: $\phi(\pi_{\text{true}}^0)$


Figure A.5: $\Delta E(\pi^0)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

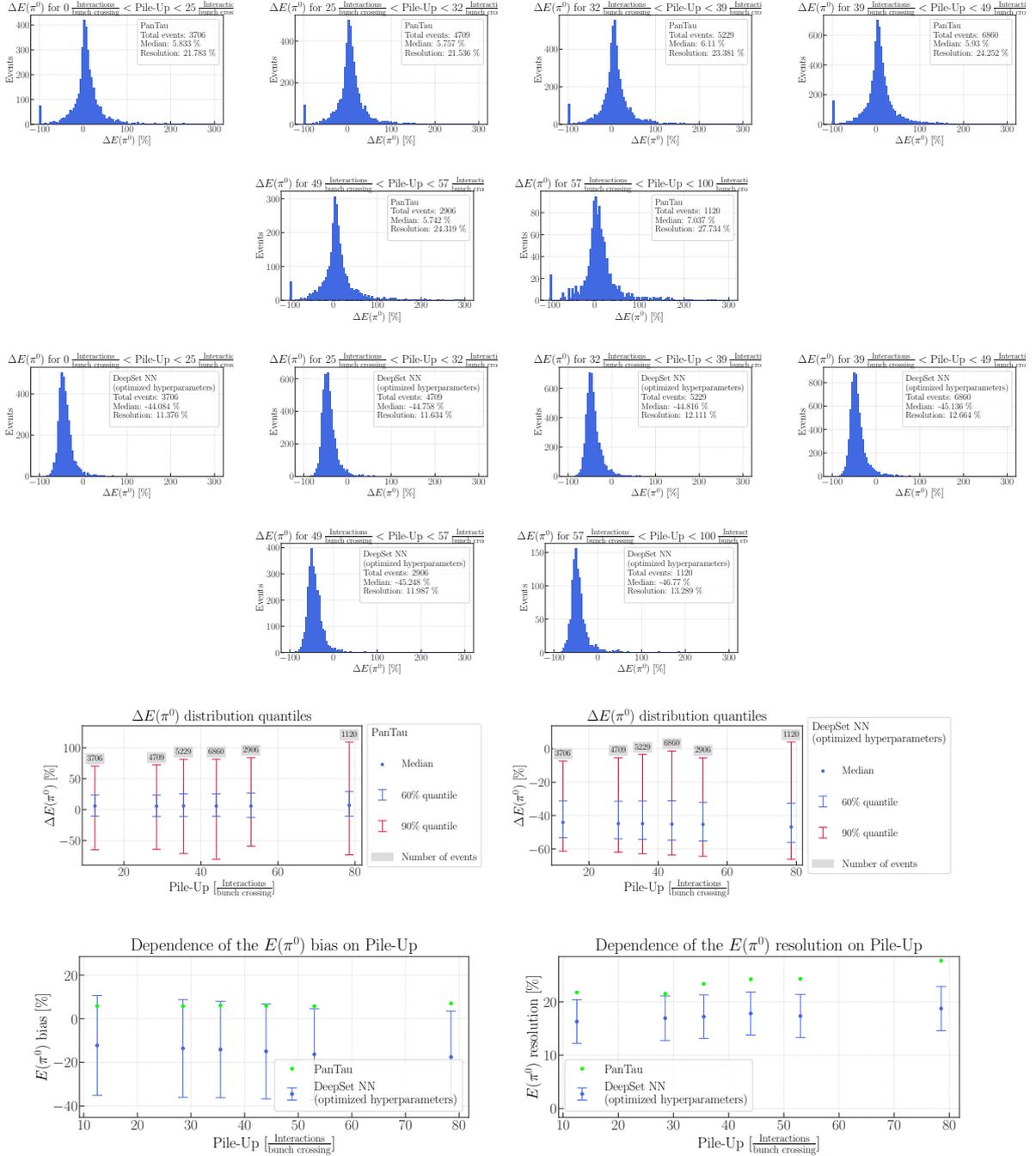


Figure A.6: $\Delta E(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

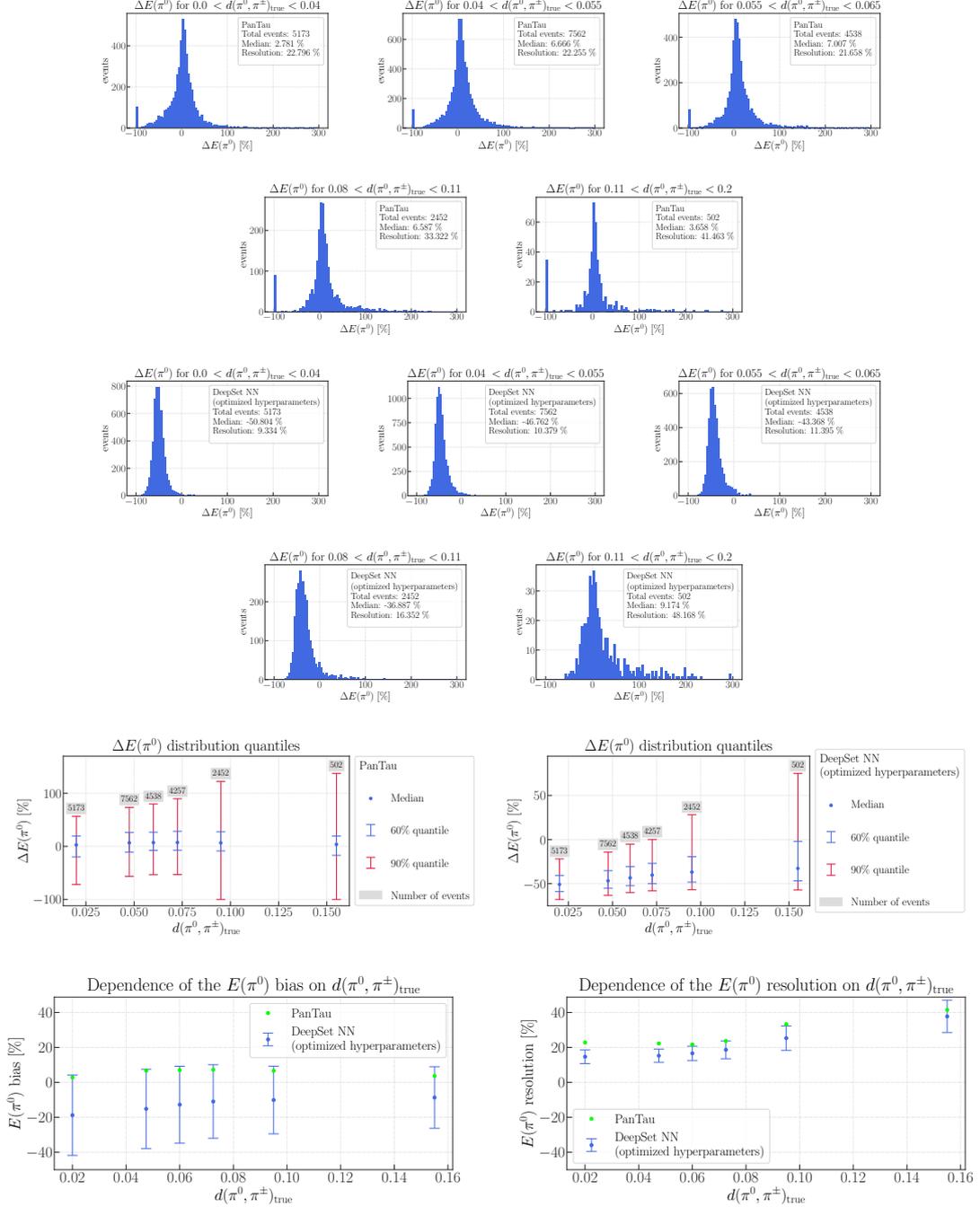
Binning: $d(\pi^0, \pi^\pm)$


Figure A.7: $\Delta E(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

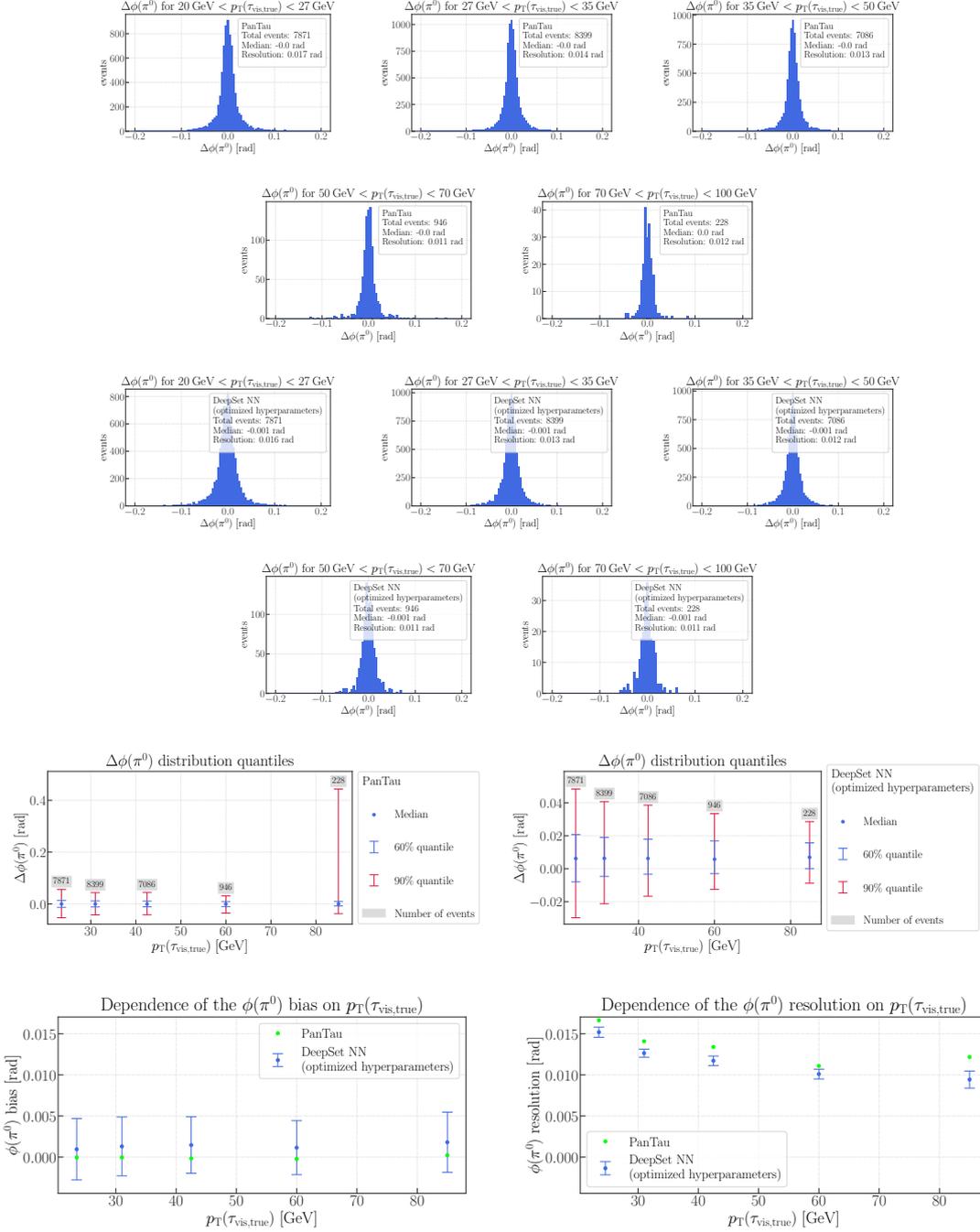
Figure of Merit: $\Delta\phi(\pi^0)$
Binning: $p_T(\tau_{\text{vis,true}})$


Figure A.8: $\Delta\phi(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

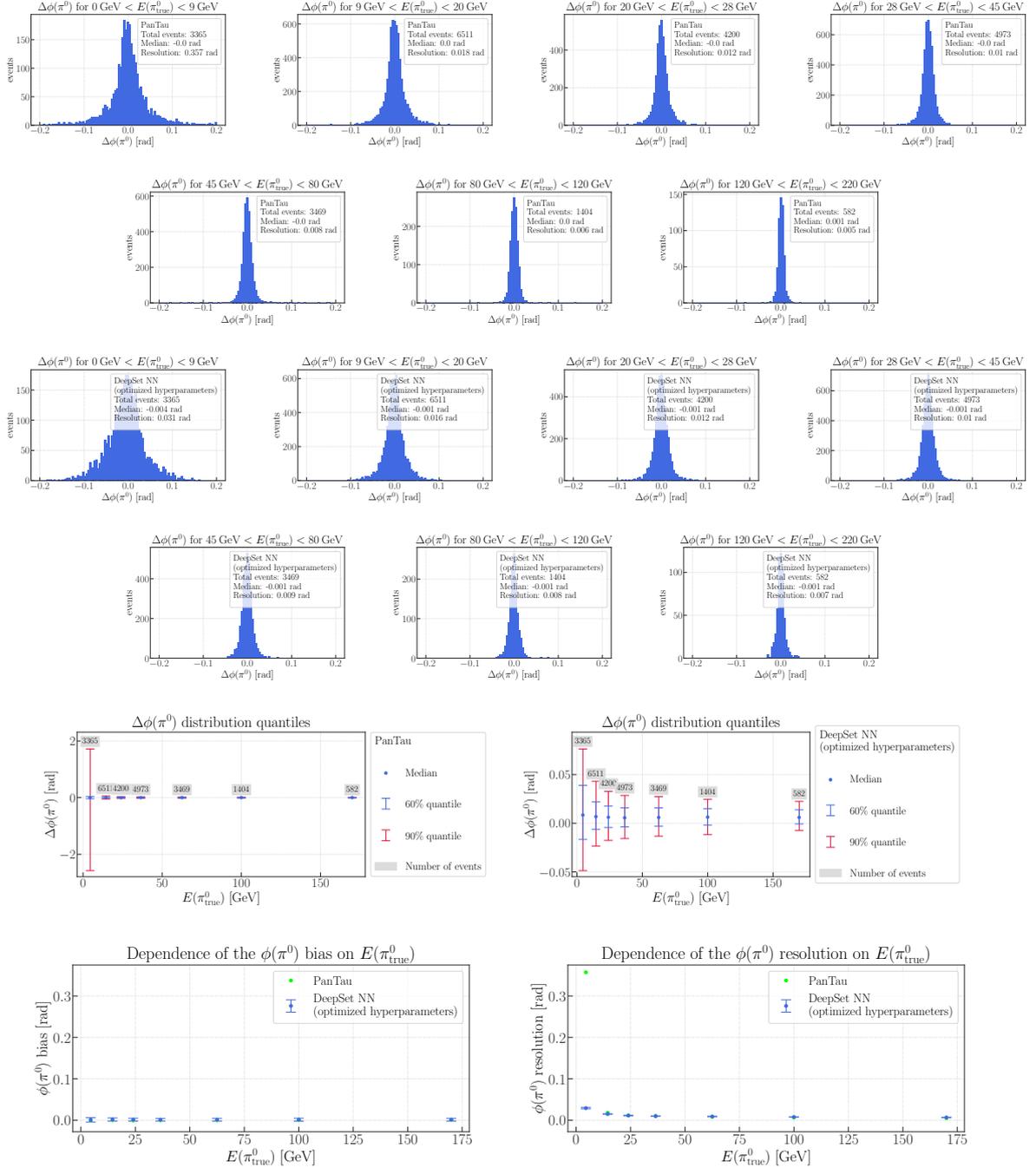


Figure A.9: $\Delta\phi(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

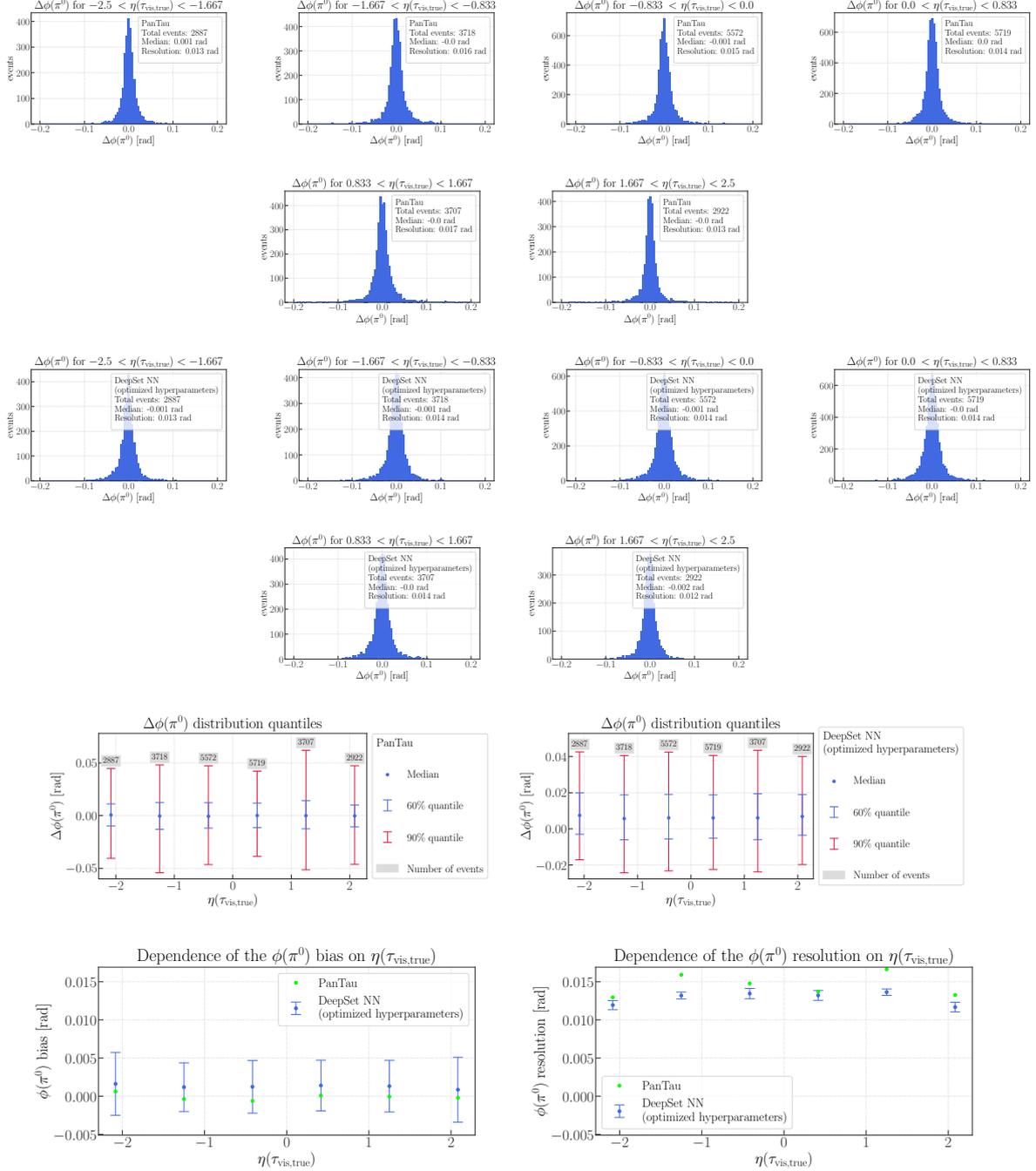


Figure A.10: $\Delta\phi(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi^0_{\text{true}})$

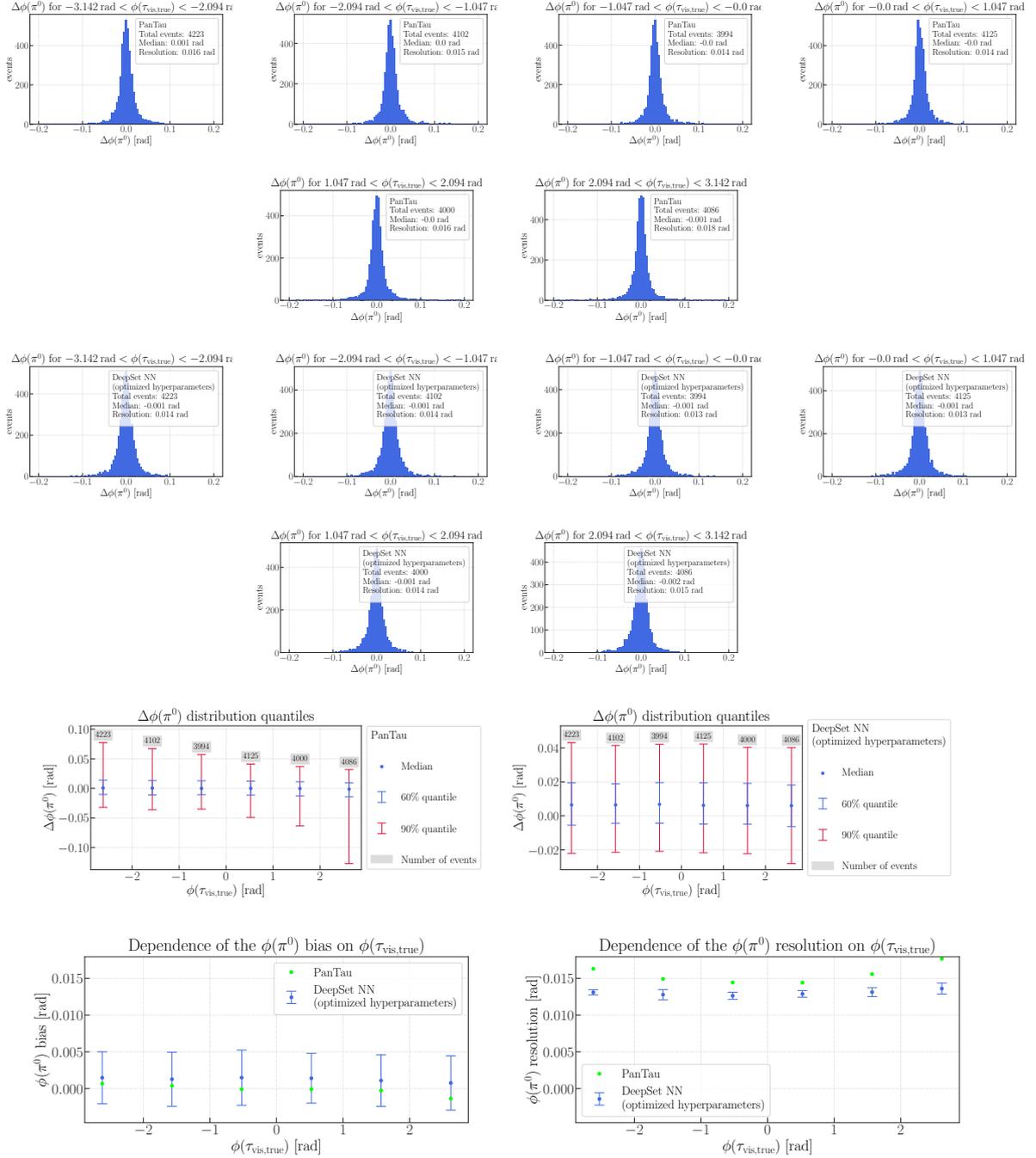


Figure A.11: $\Delta\phi(\pi^0)$ plots for different intervals of $\phi(\pi^0_{\text{true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

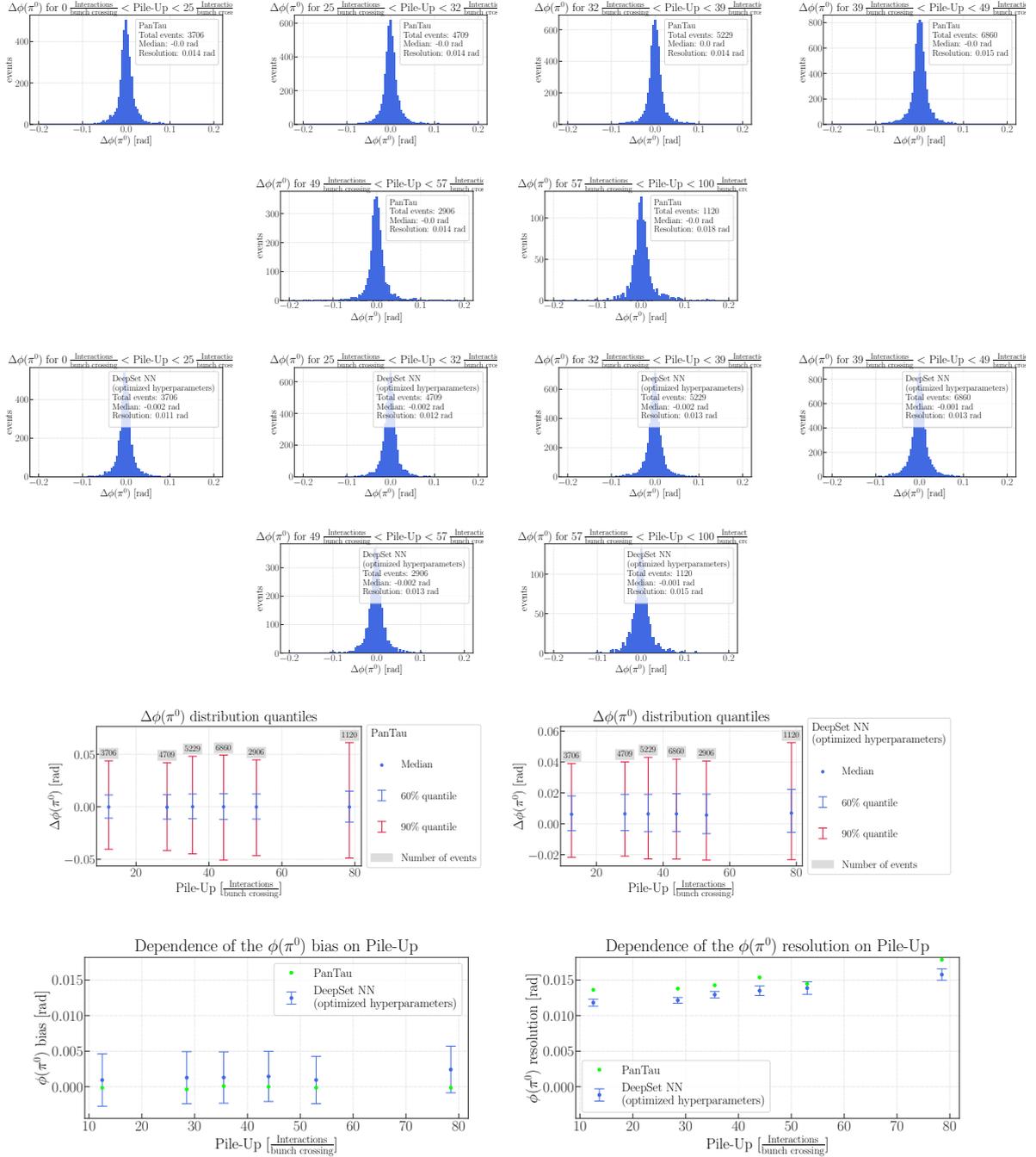


Figure A.12: $\Delta\phi(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$

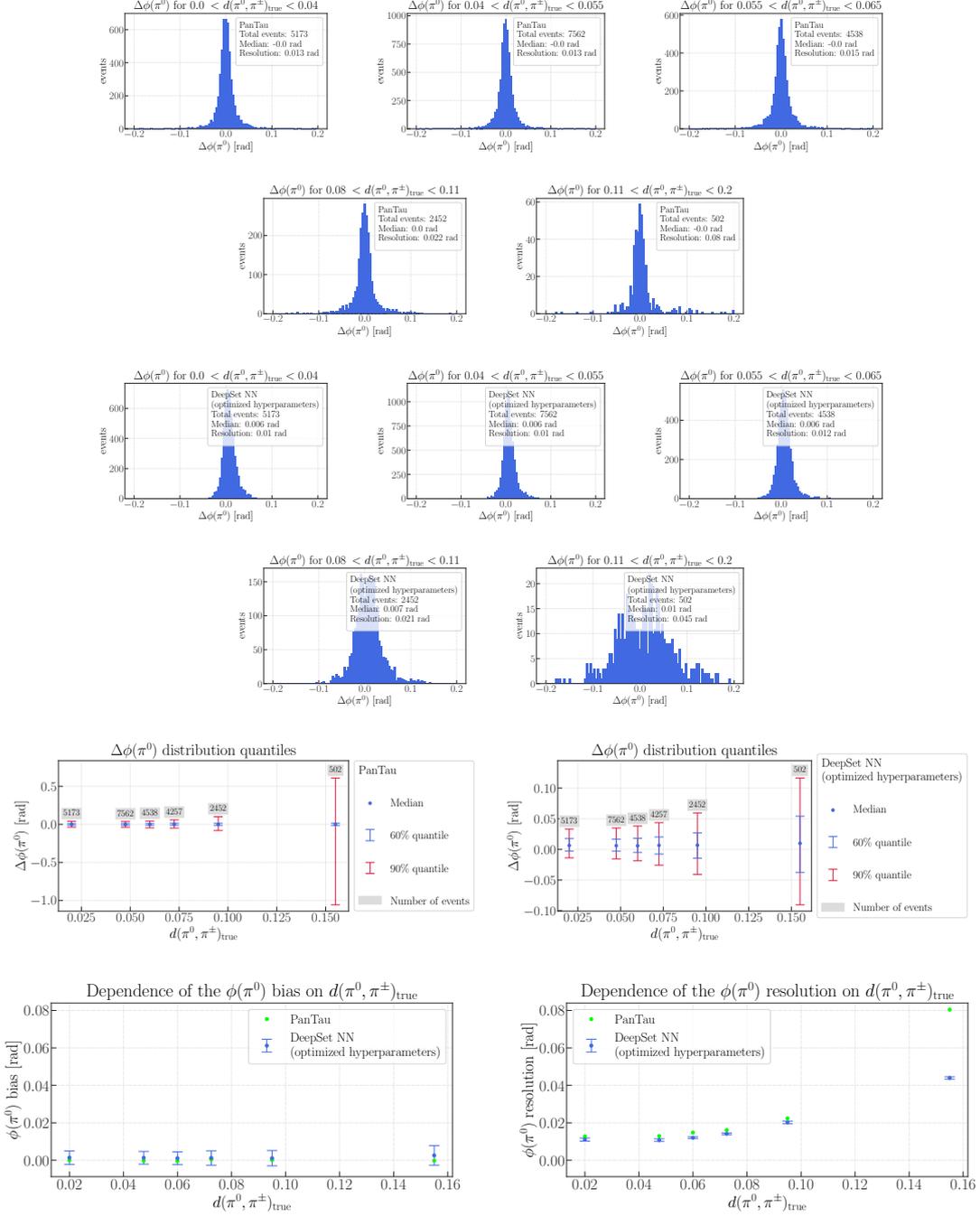


Figure A.13: $\Delta\phi(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

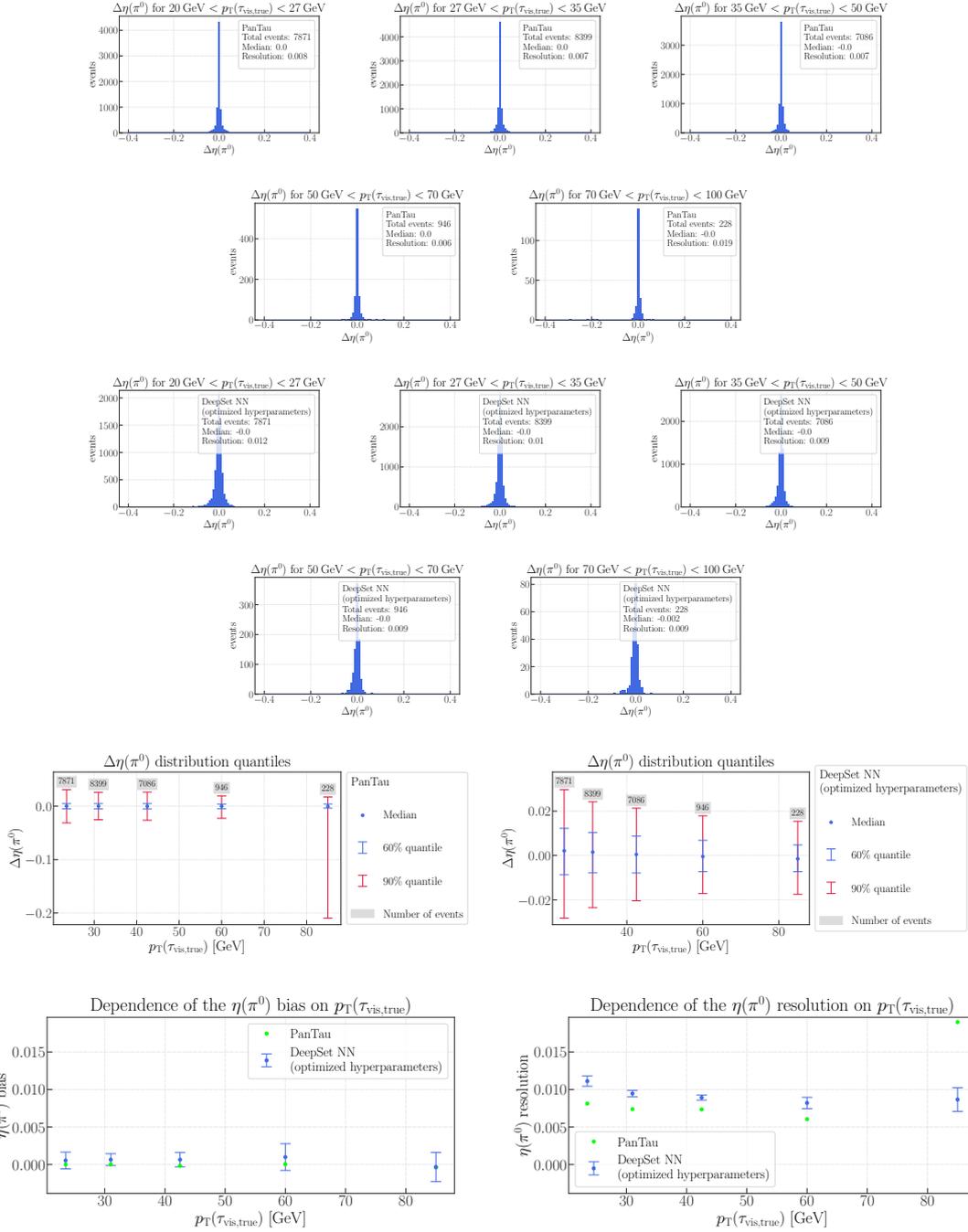
Figure of Merit: $\Delta\eta(\pi^0)$
Binning: $p_T(\tau_{\text{vis,true}})$


Figure A.14: $\Delta\eta(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

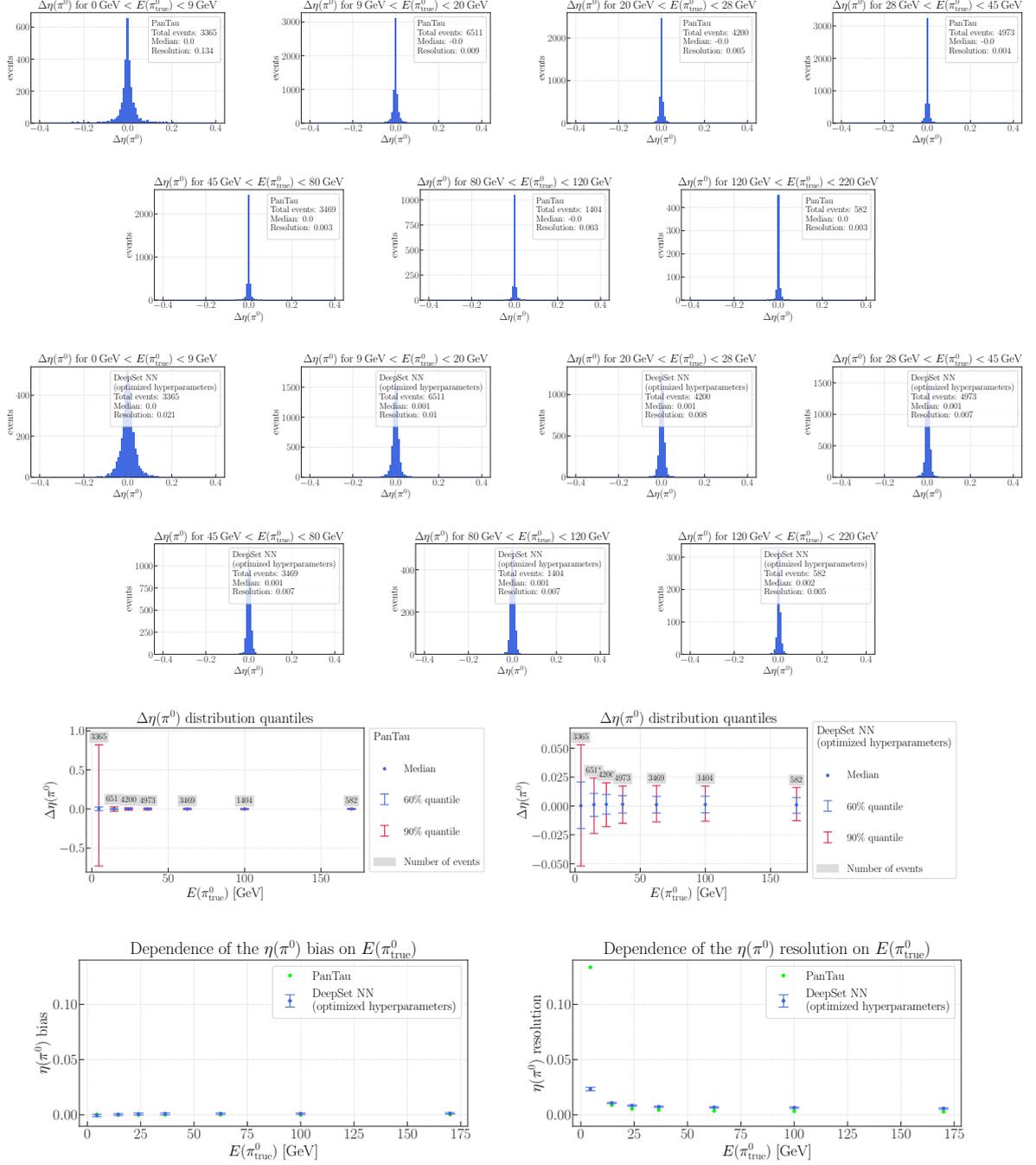


Figure A.15: $\Delta\eta(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

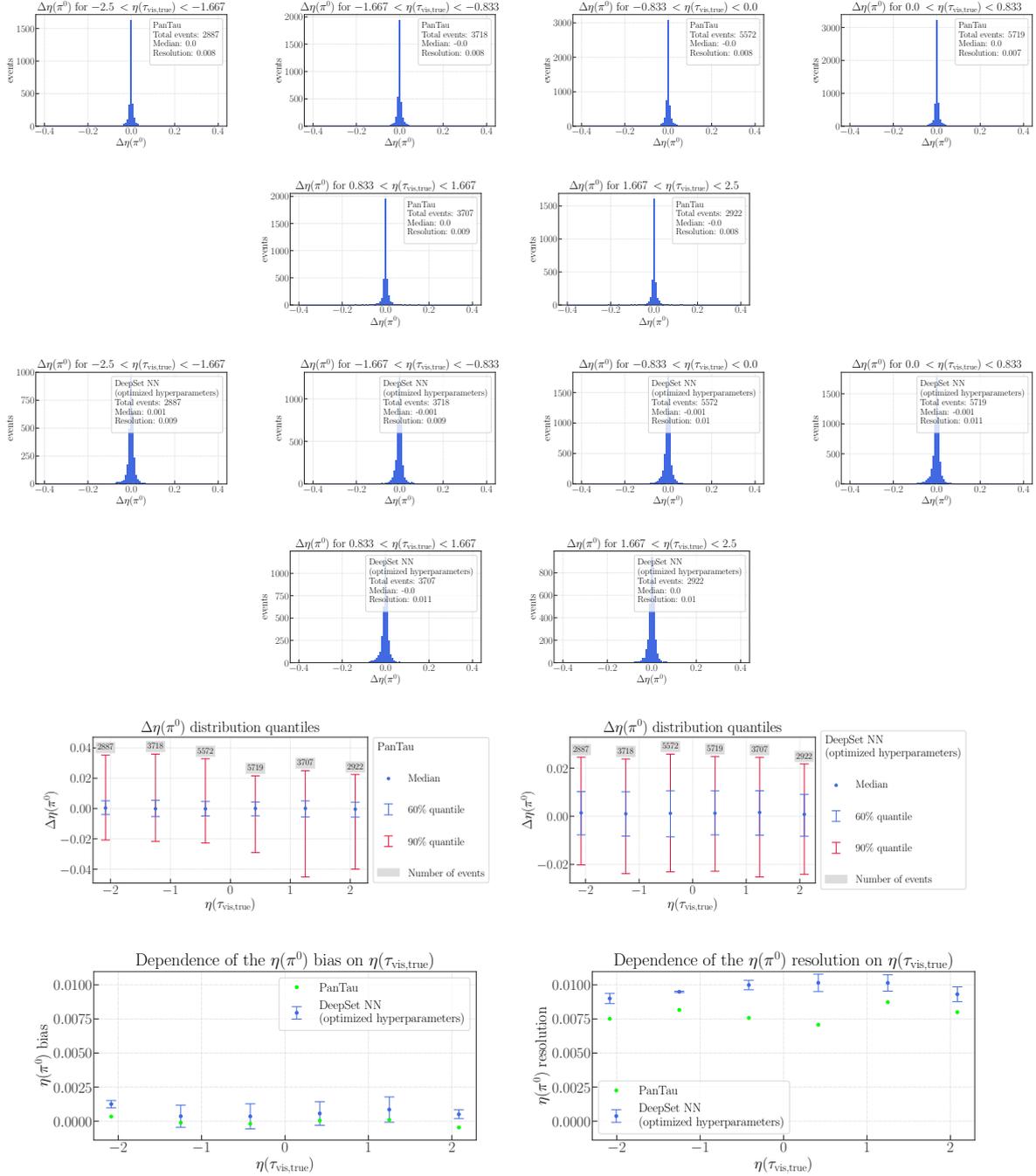


Figure A.16: $\Delta\eta(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi^0_{\text{true}})$

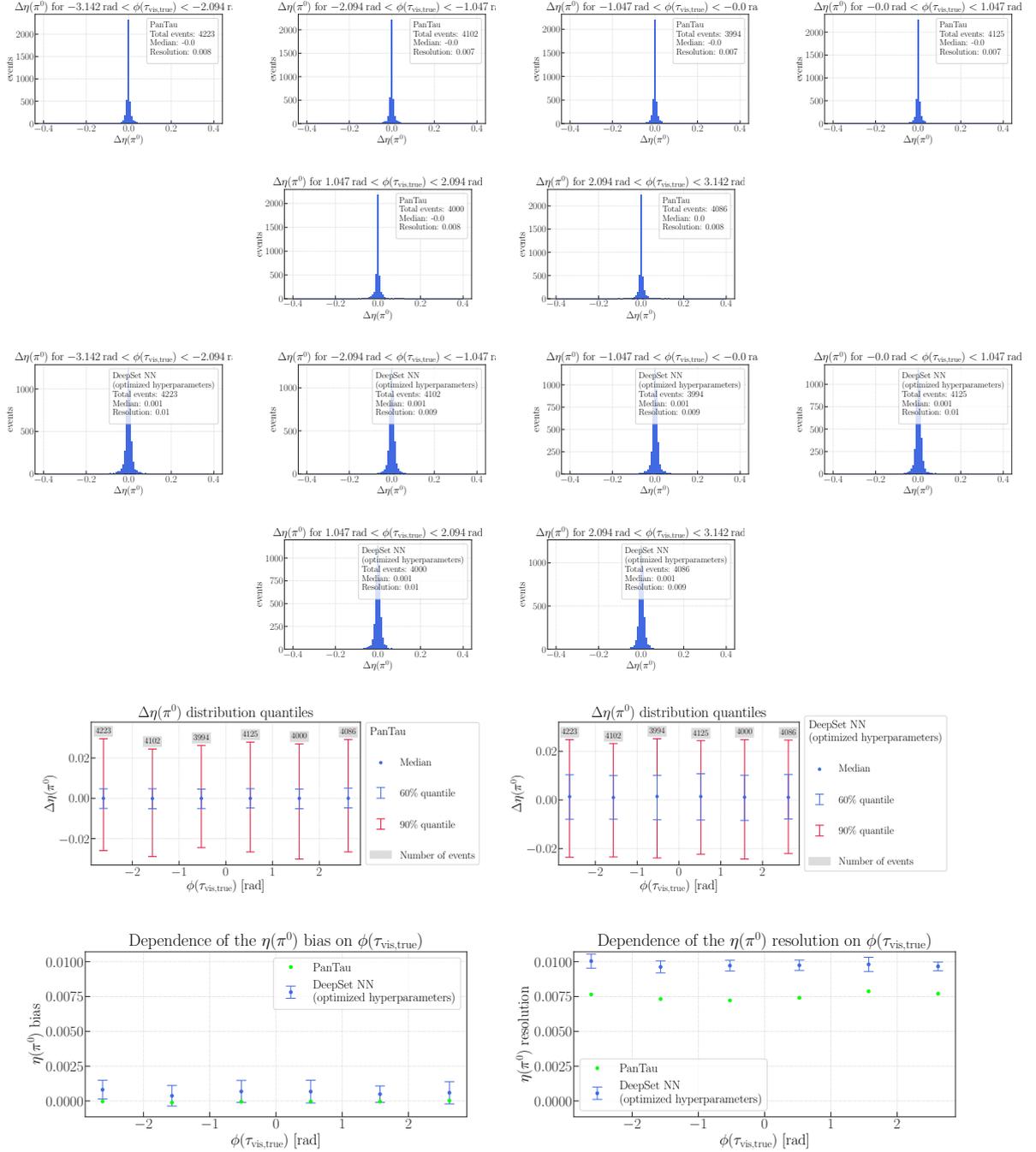


Figure A.17: $\Delta\eta(\pi^0)$ plots for different intervals of $\phi(\pi^0_{\text{true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

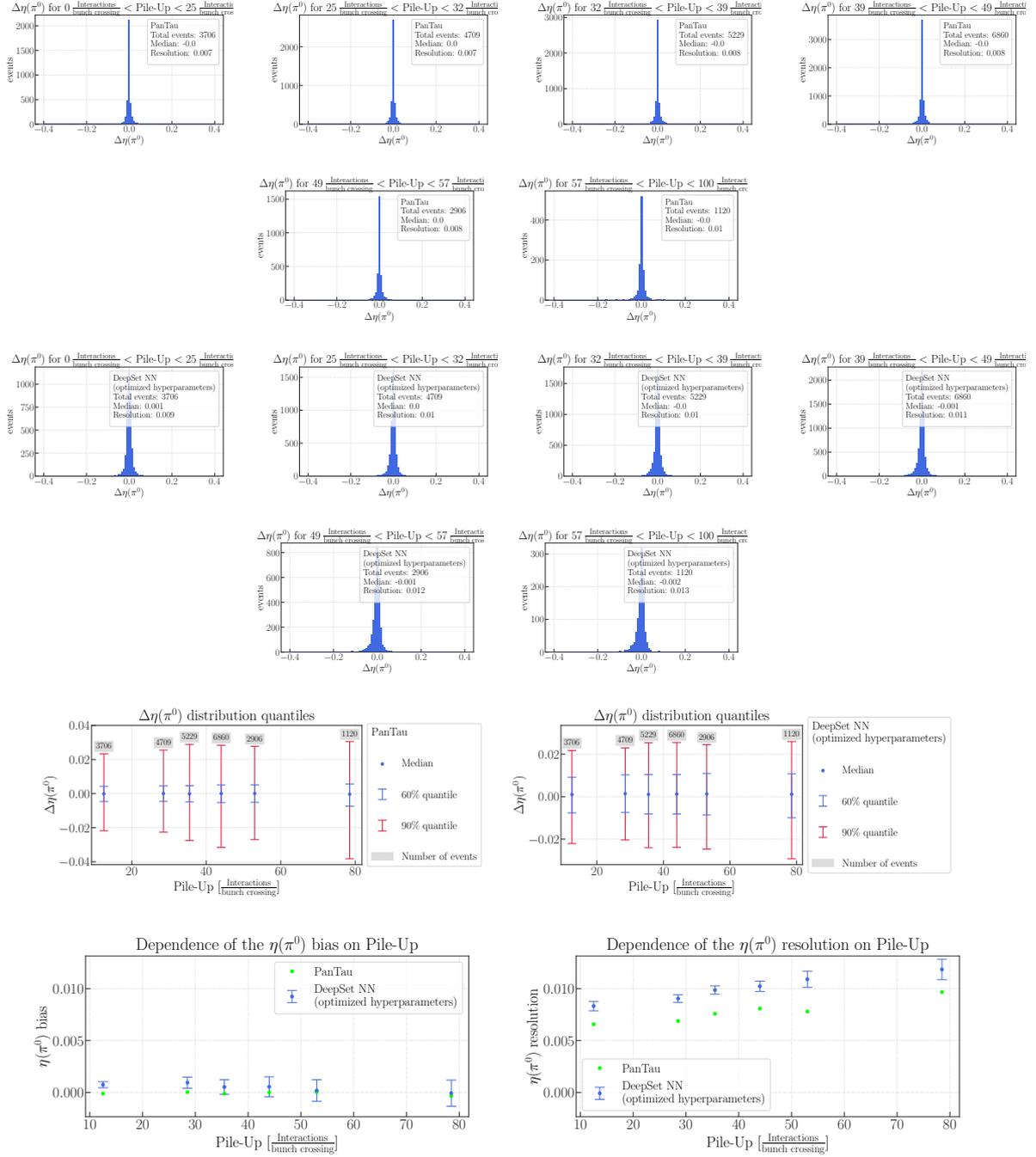


Figure A.18: $\Delta\eta(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$

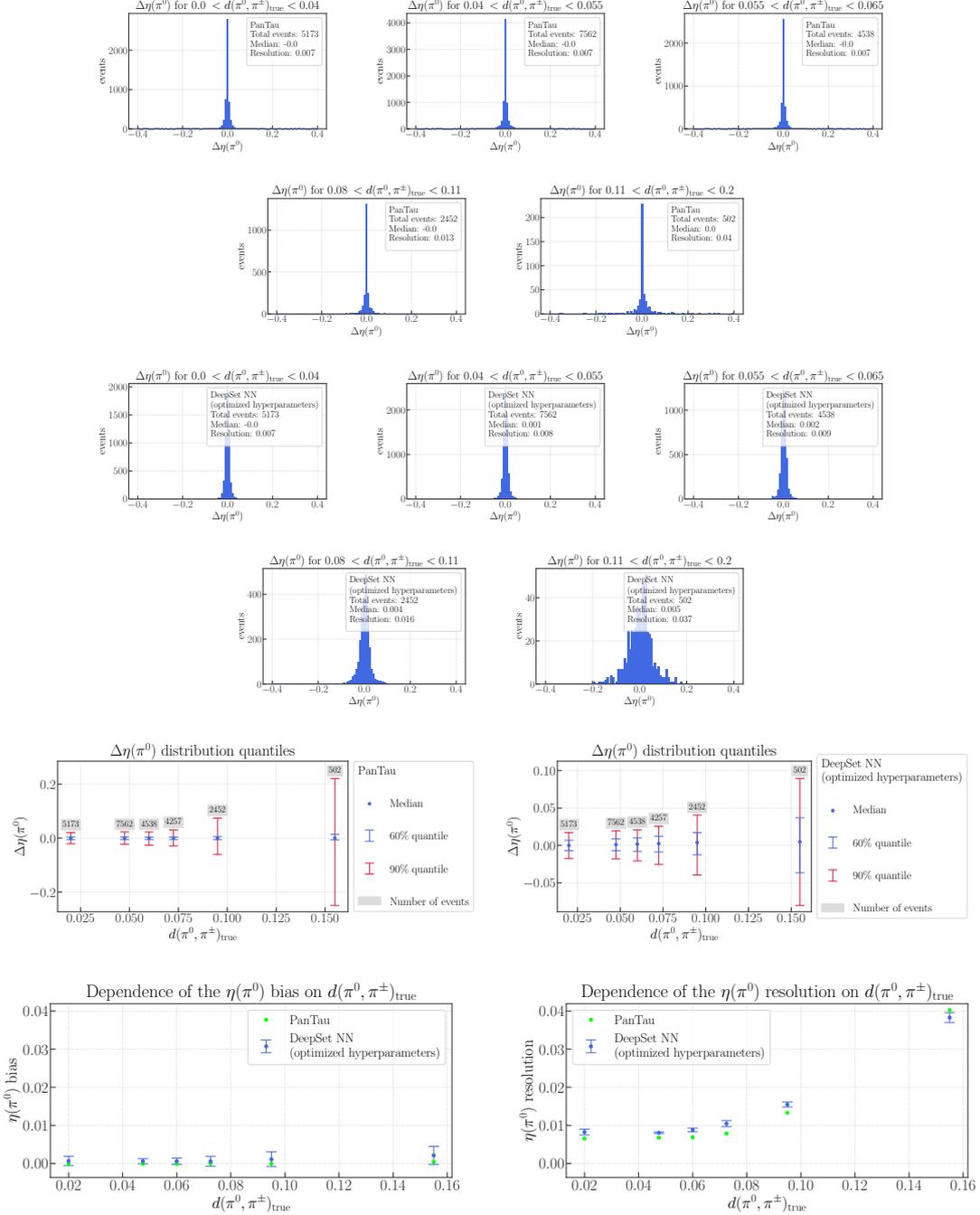


Figure A.19: $\Delta\eta(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

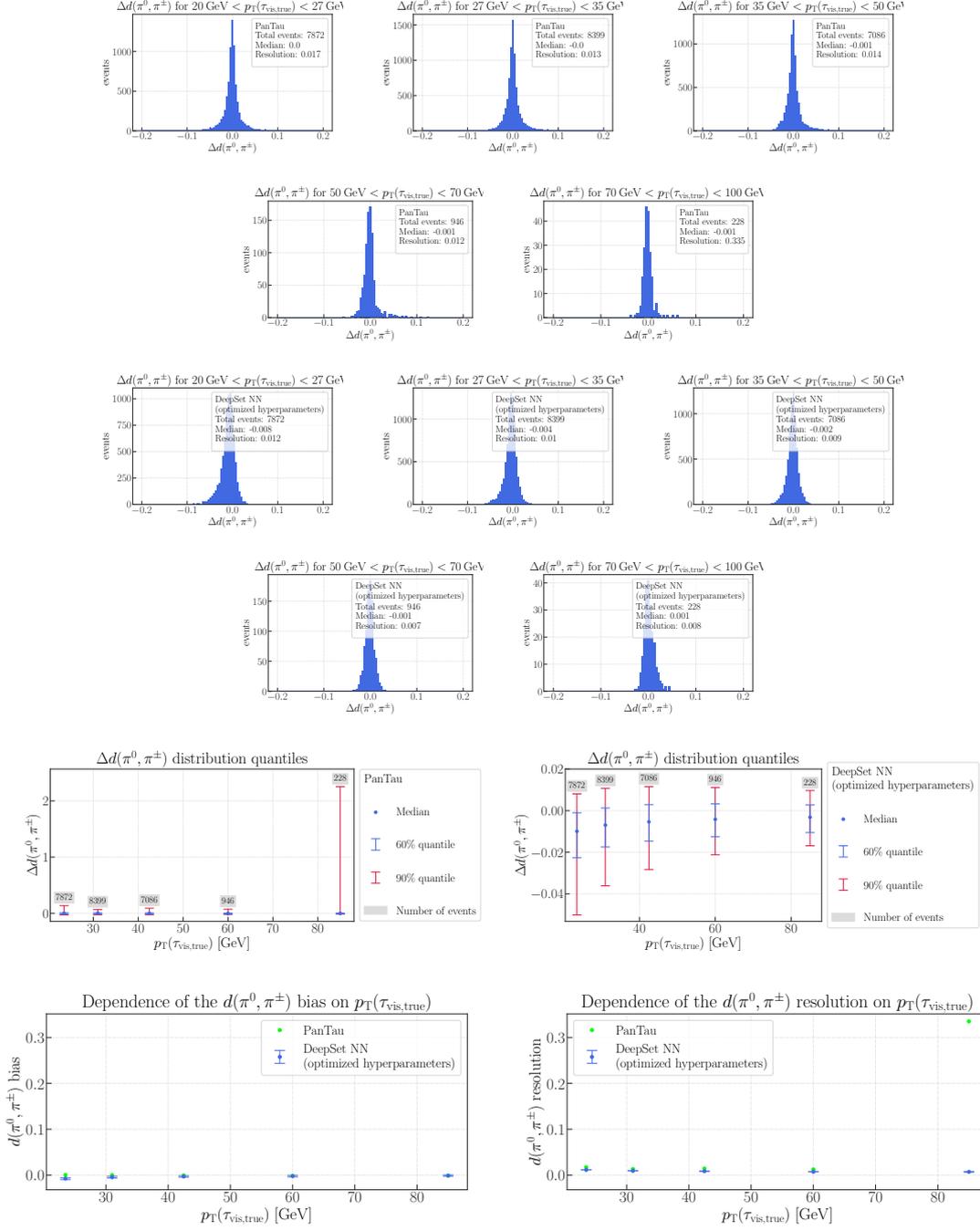
Figure of Merit: $\Delta d(\pi^0, \pi^\pm)$
Binning: $p_T(\tau_{\text{vis,true}})$


Figure A.20: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

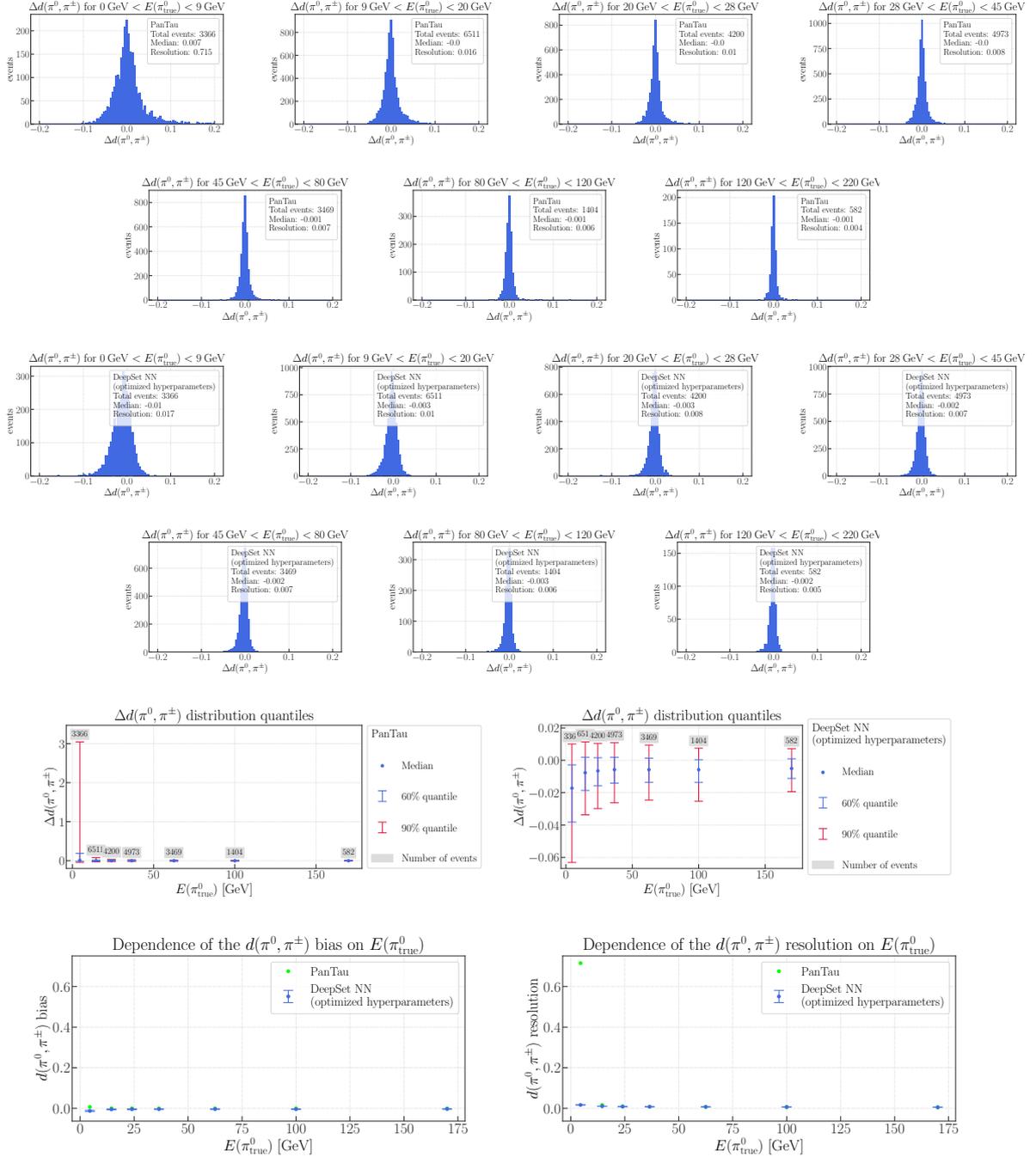


Figure A.21: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

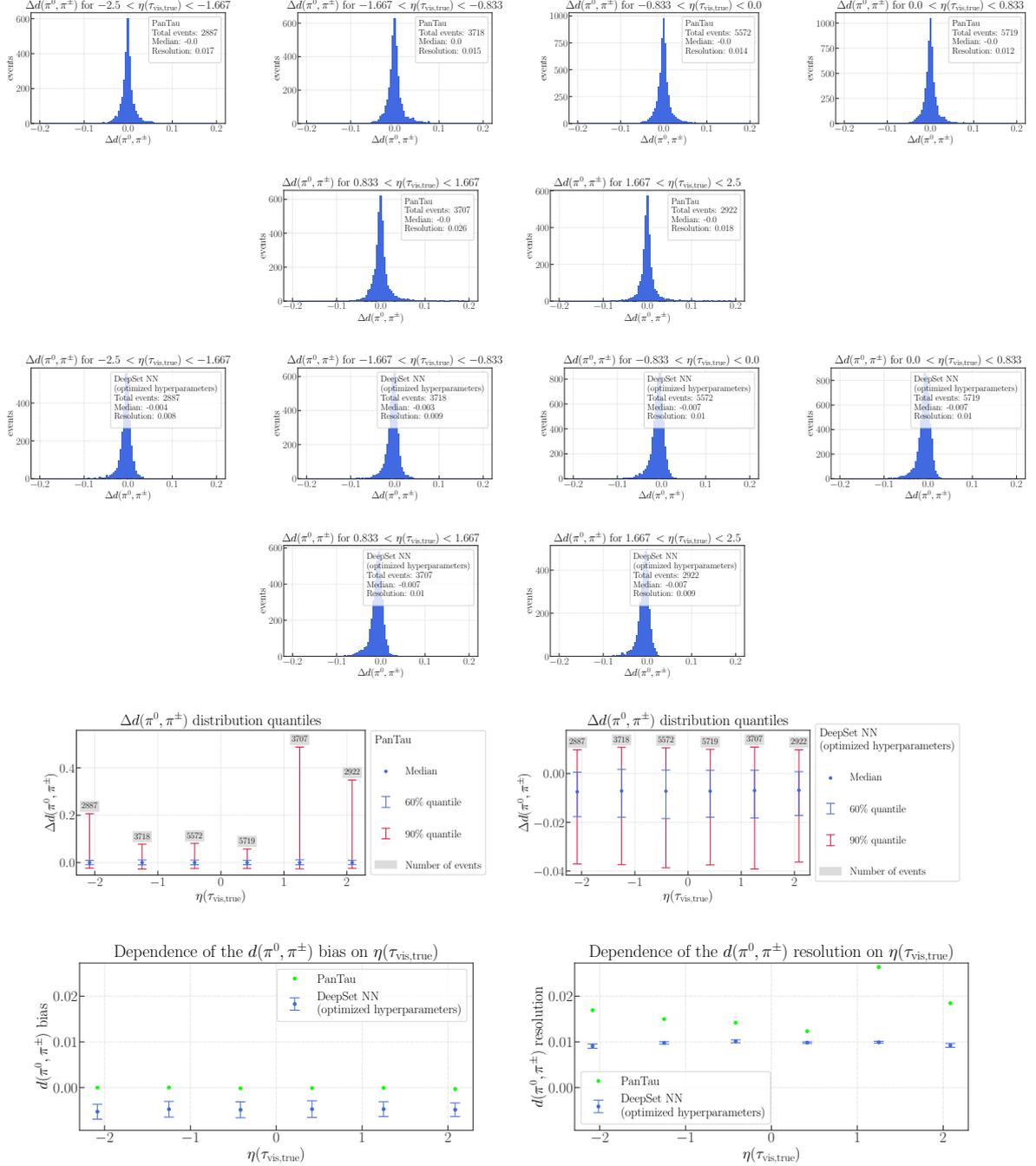


Figure A.22: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi_{\text{true}}^0)$

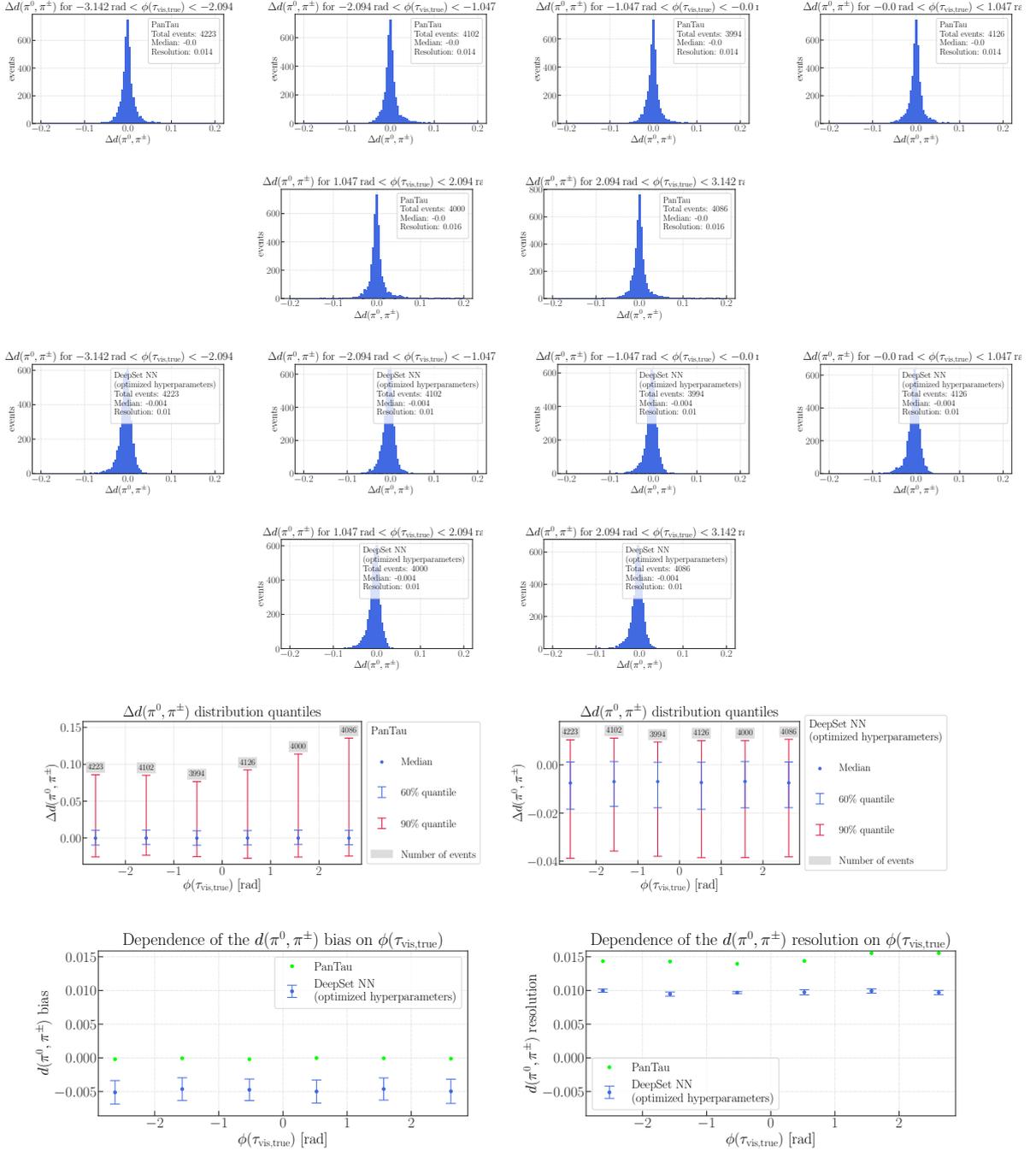


Figure A.23: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

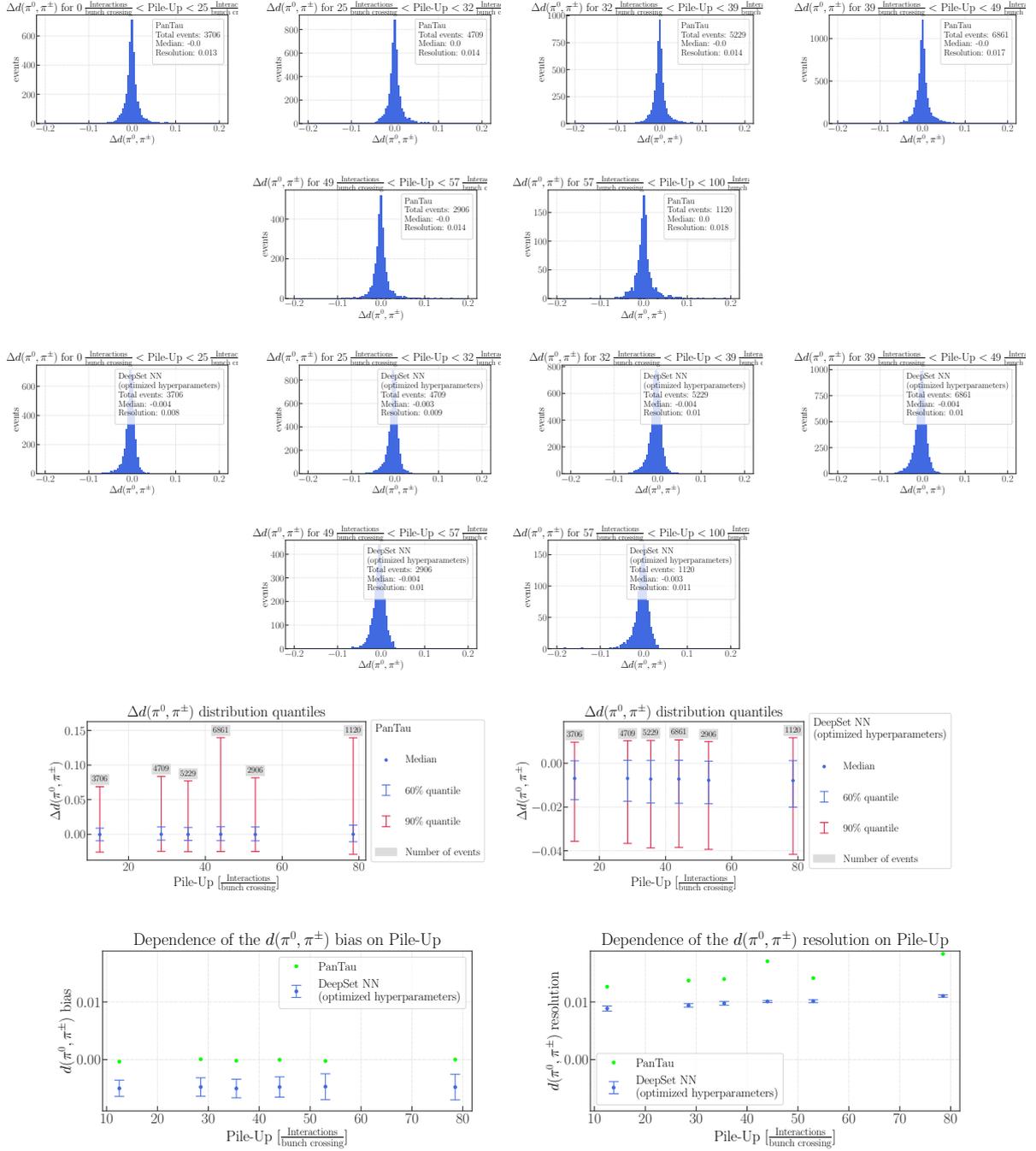


Figure A.24: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$

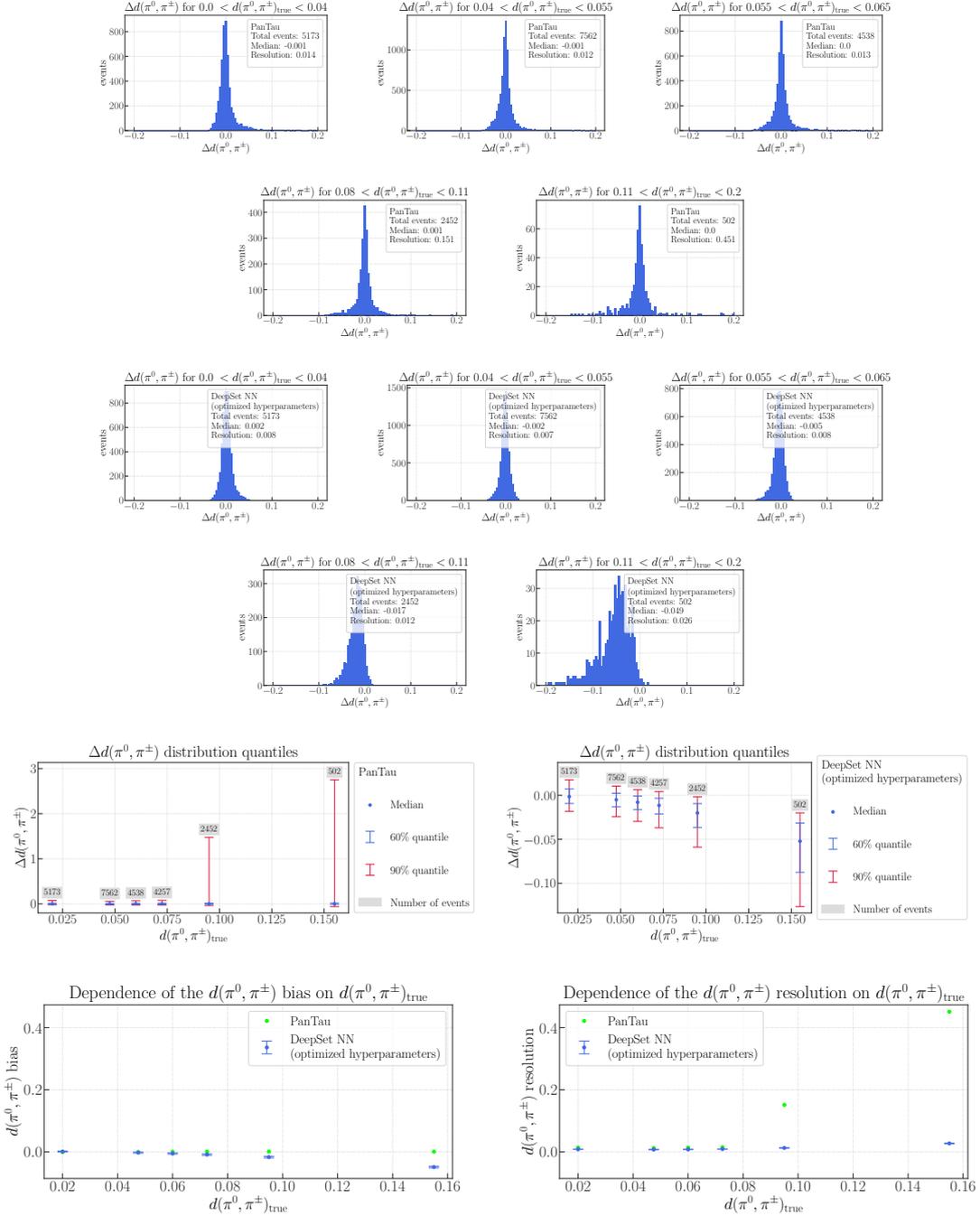


Figure A.25: $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Figure of Merit: ΔY

Binning: $p_T(\tau_{\text{vis,true}})$

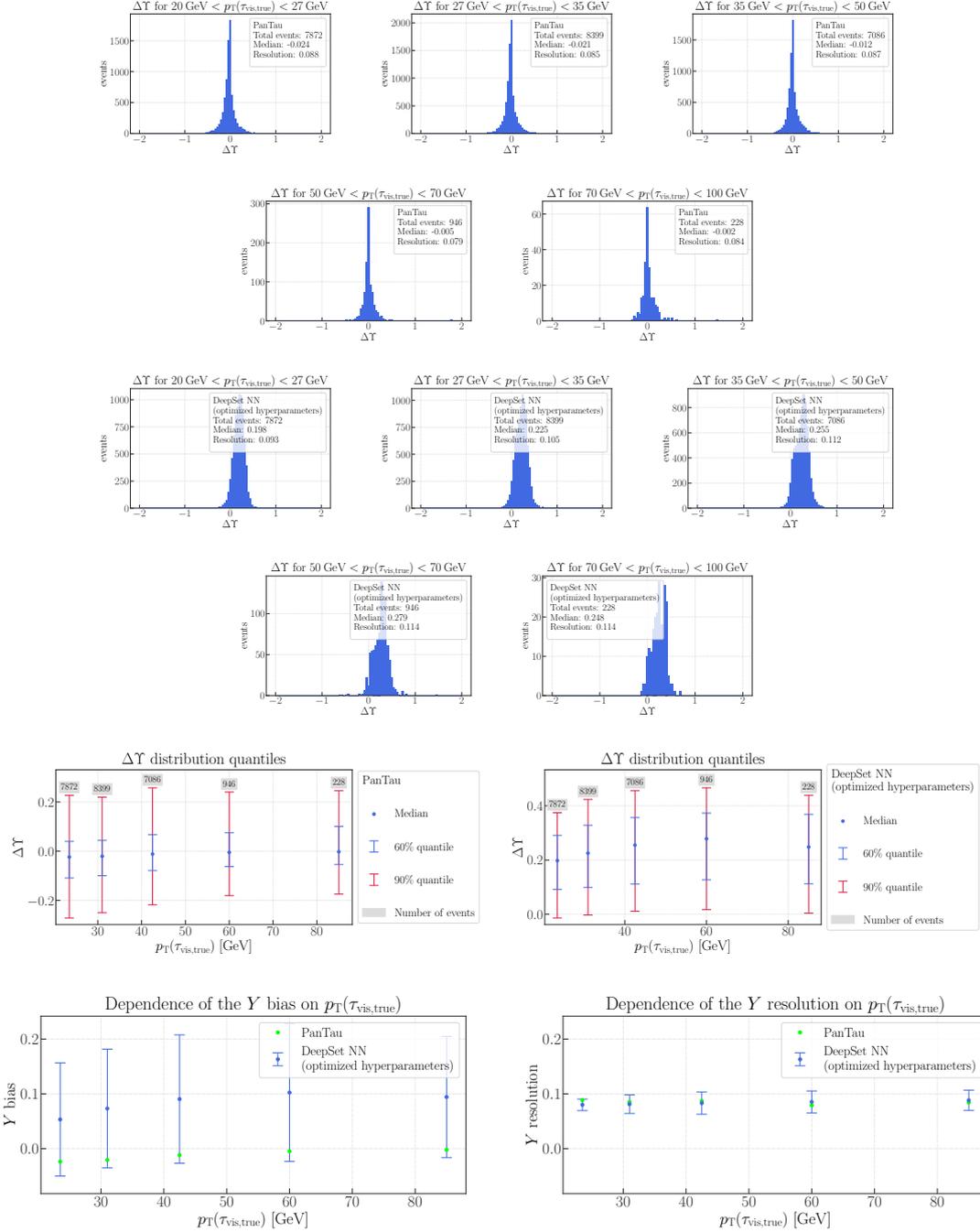


Figure A.26: ΔY plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

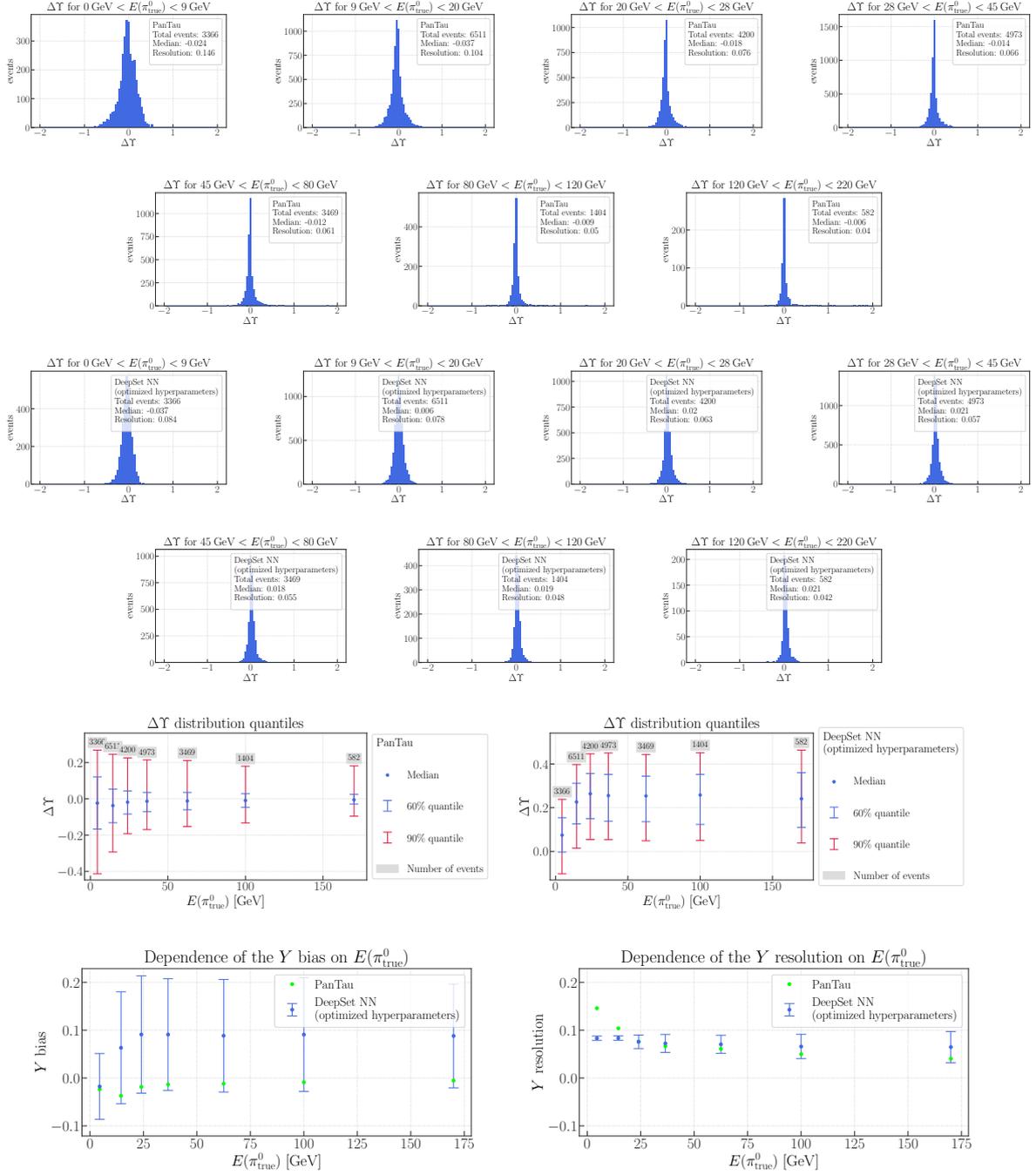


Figure A.27: ΔY plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

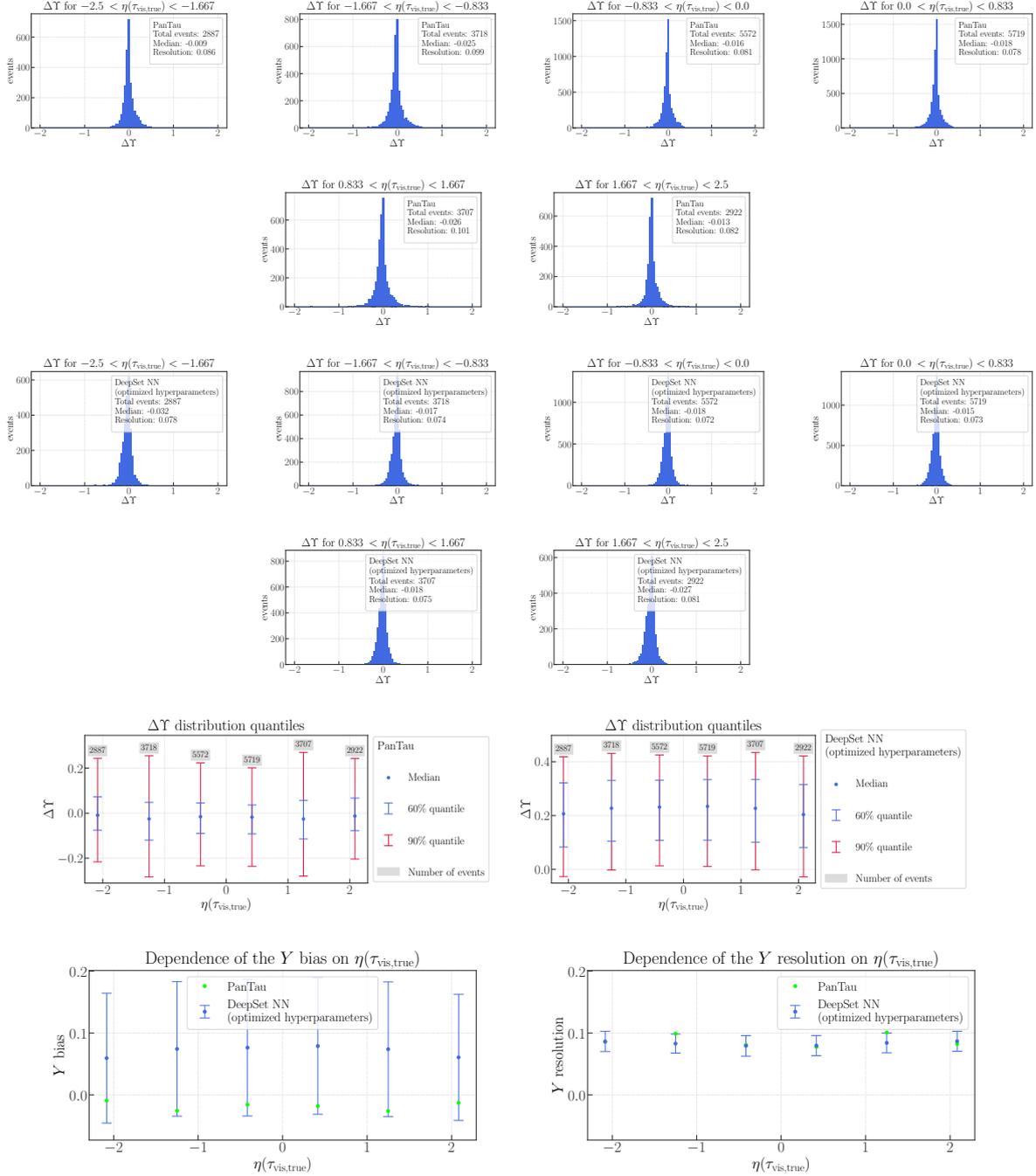


Figure A.28: ΔY plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi_{\text{true}}^0)$

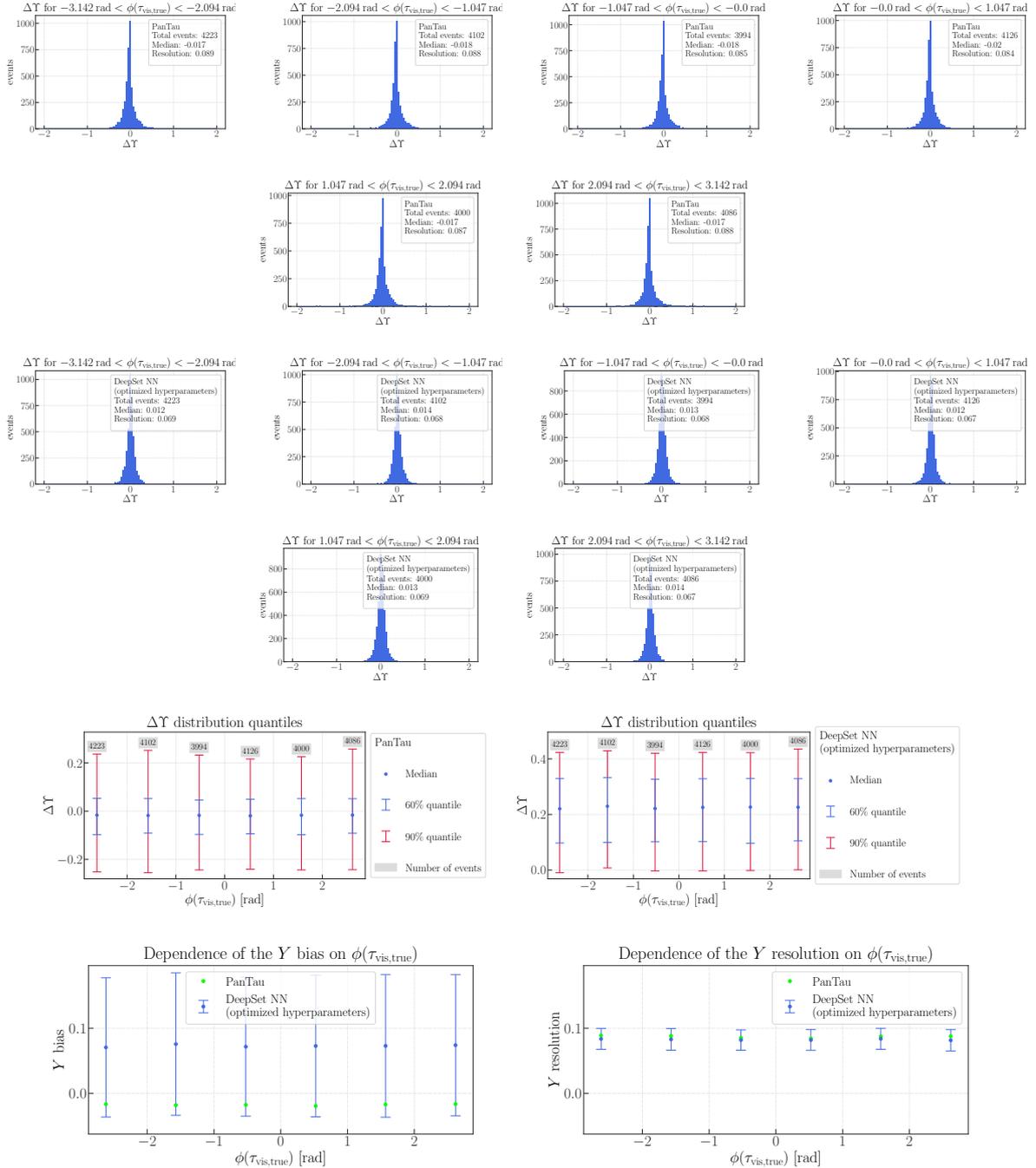


Figure A.29: ΔY plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

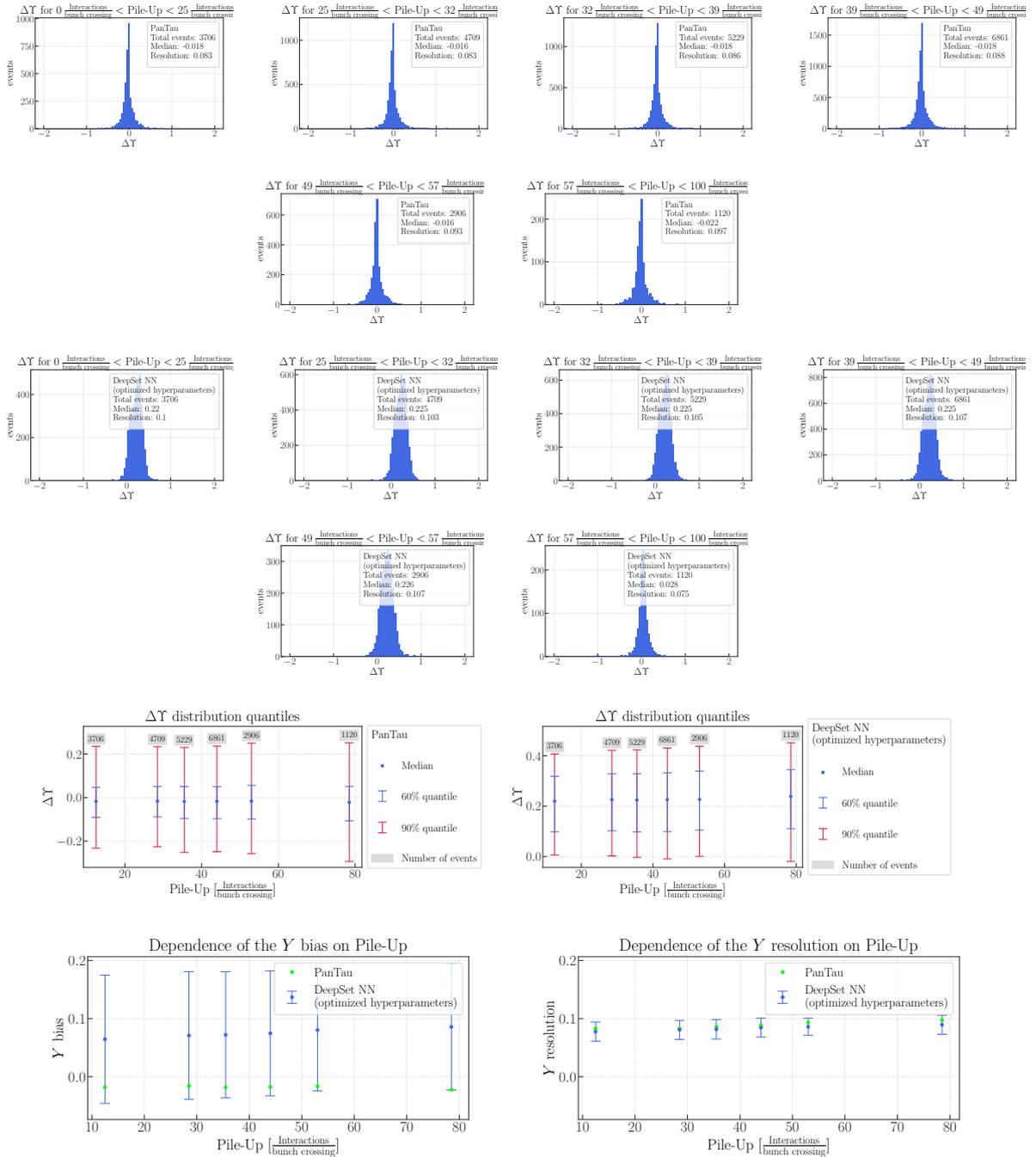


Figure A.30: ΔY plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$

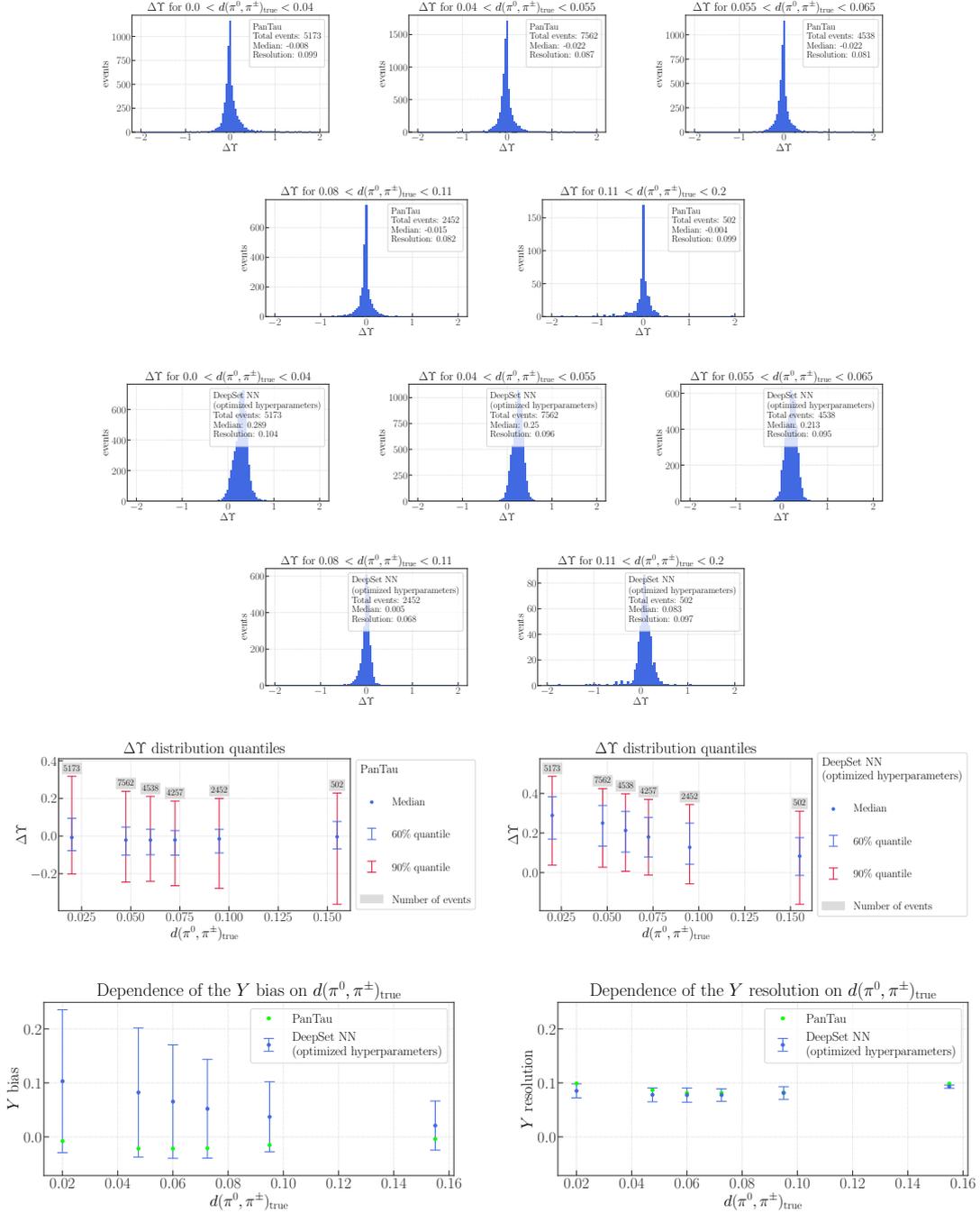


Figure A.31: ΔY plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

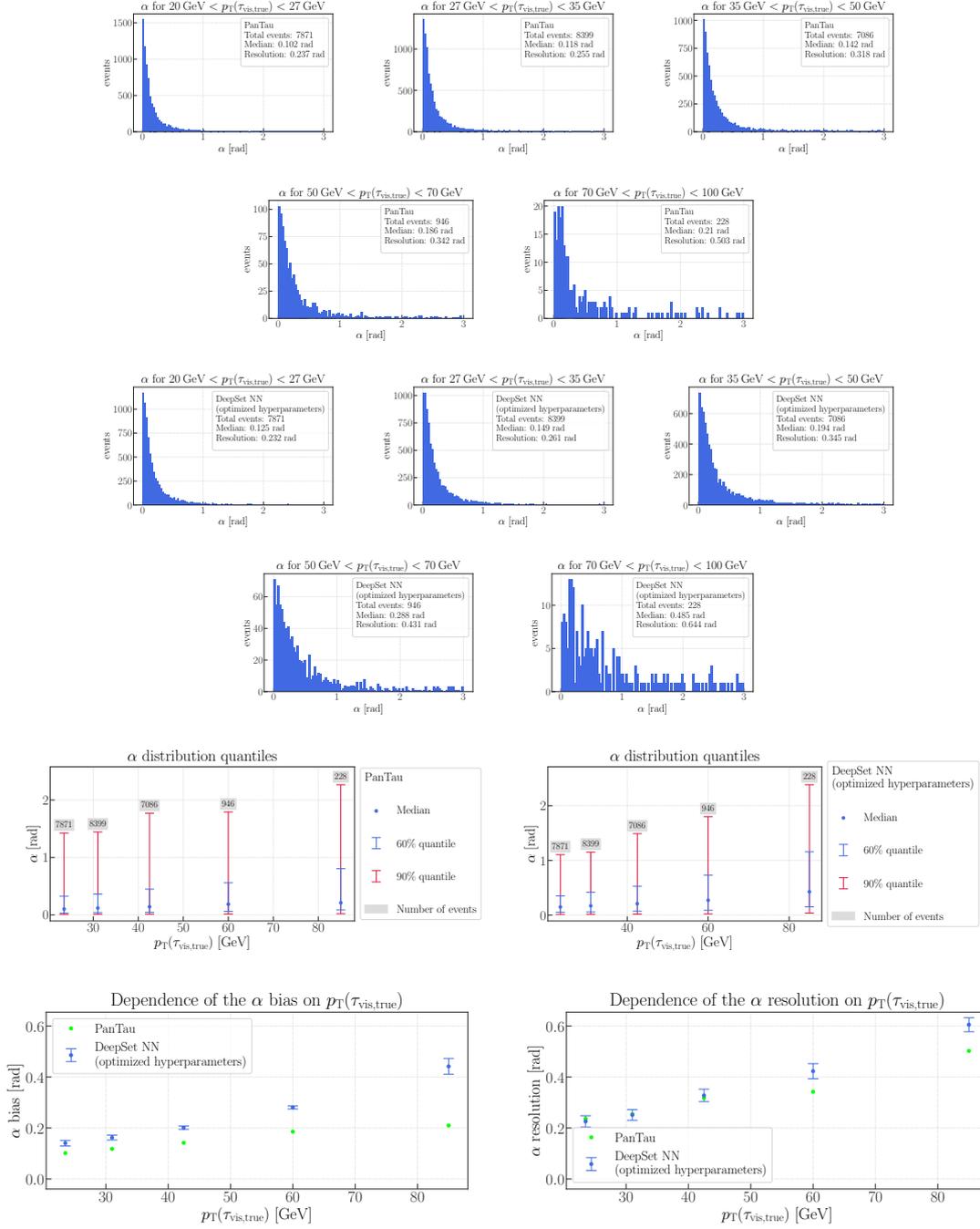
Figure of Merit: α
Binning: $p_T(\tau_{\text{vis,true}})$


Figure A.32: α plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

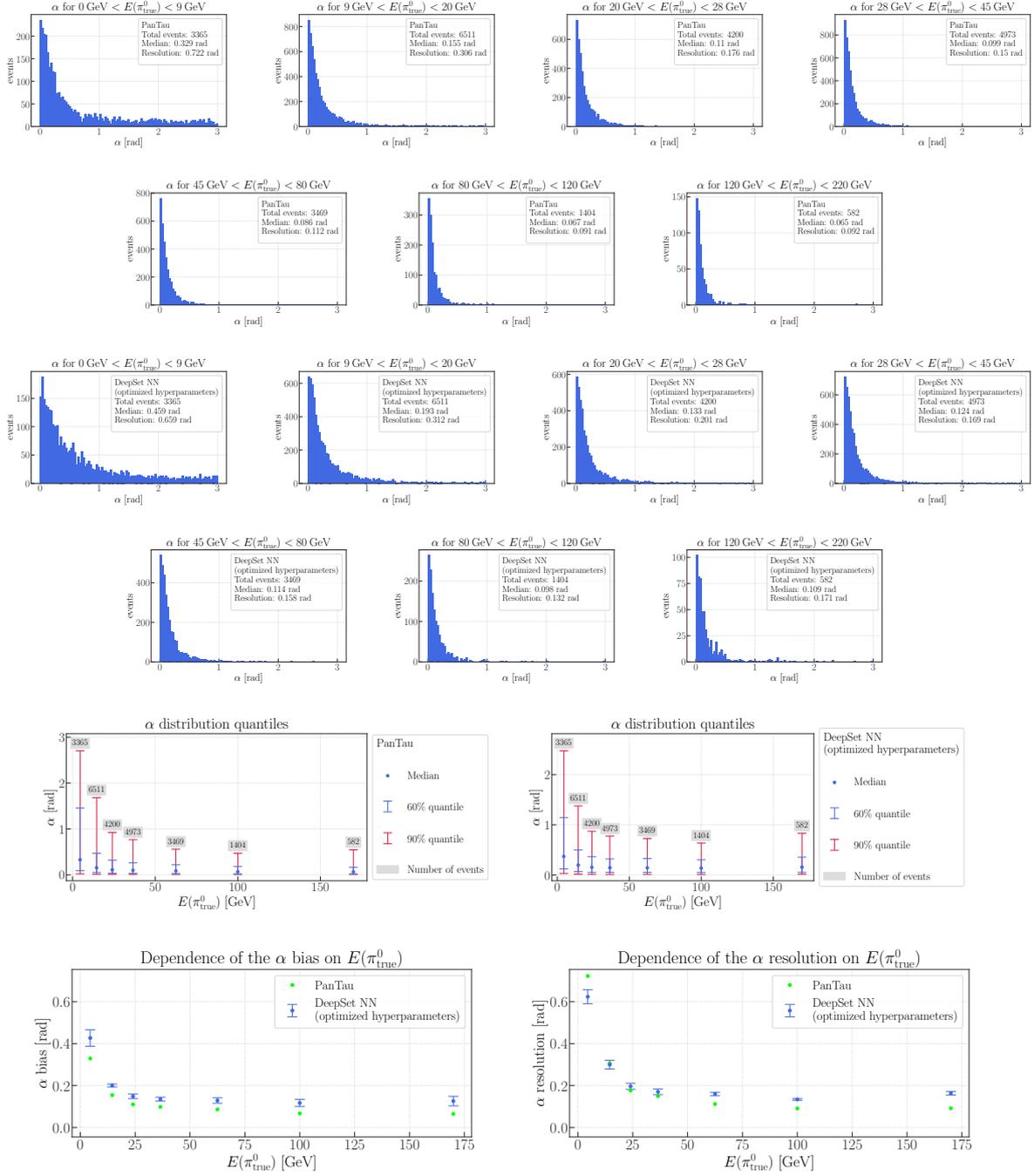


Figure A.33: α plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

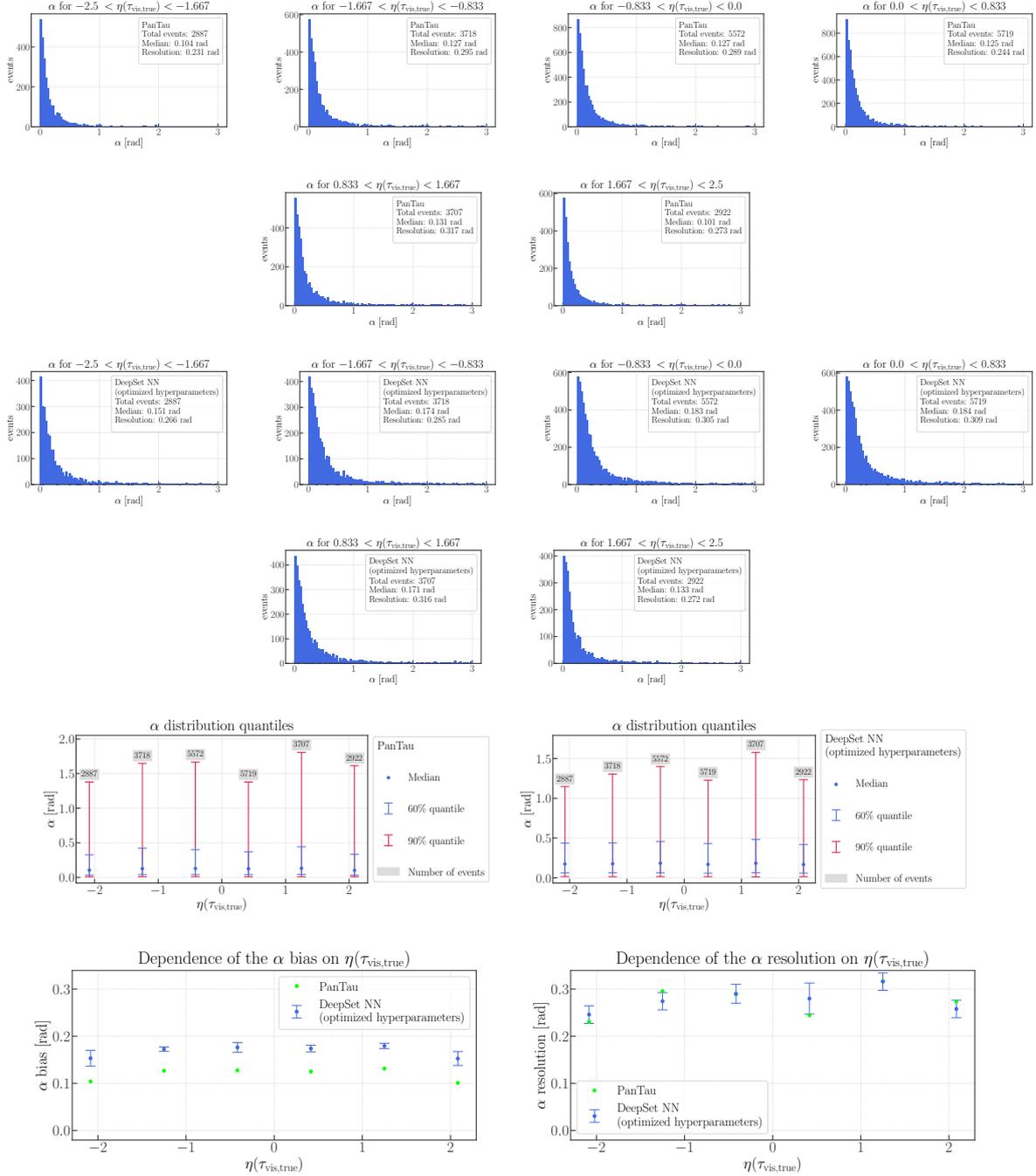


Figure A.34: α plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi_{\text{true}}^0)$

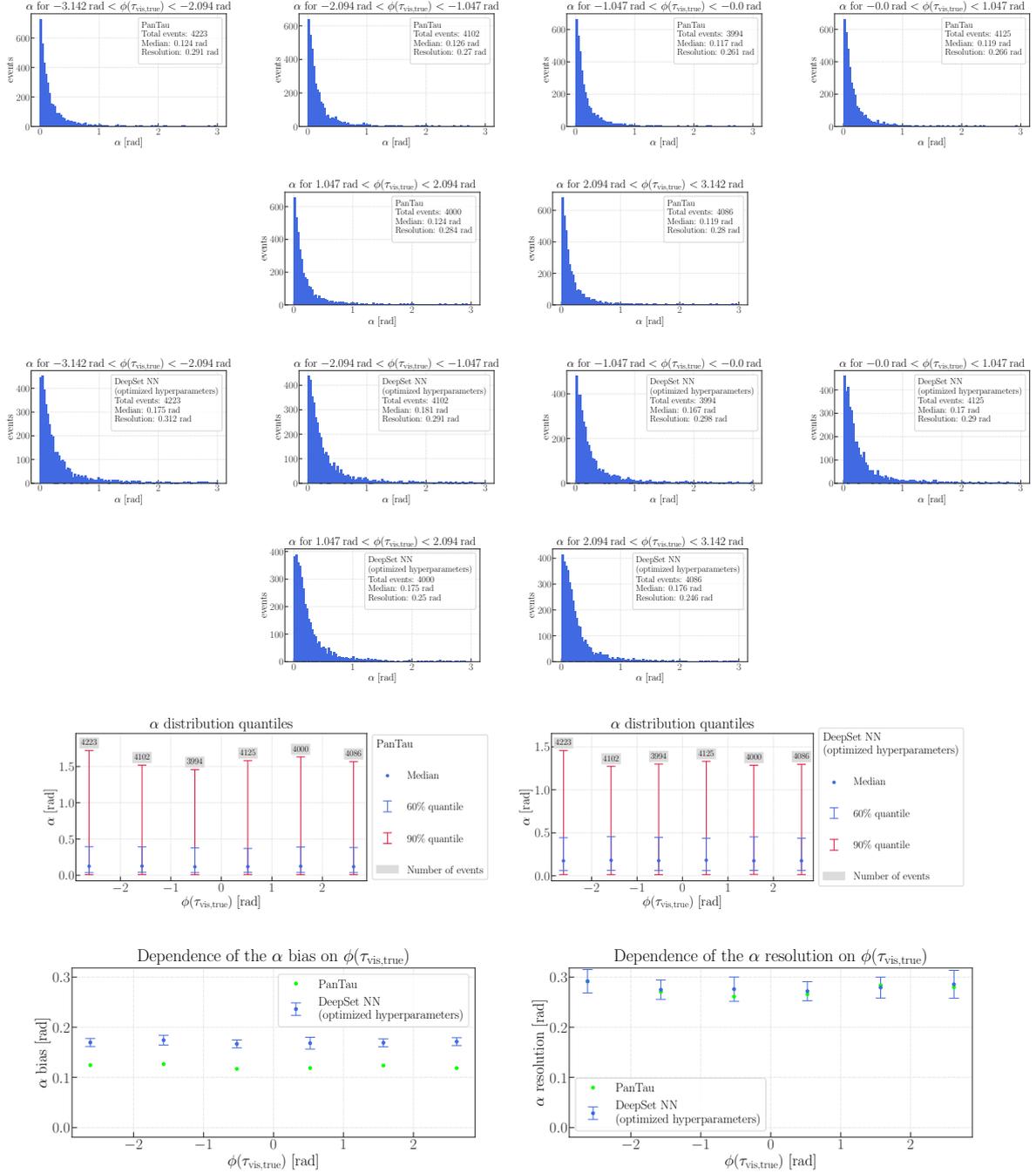


Figure A.35: α plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

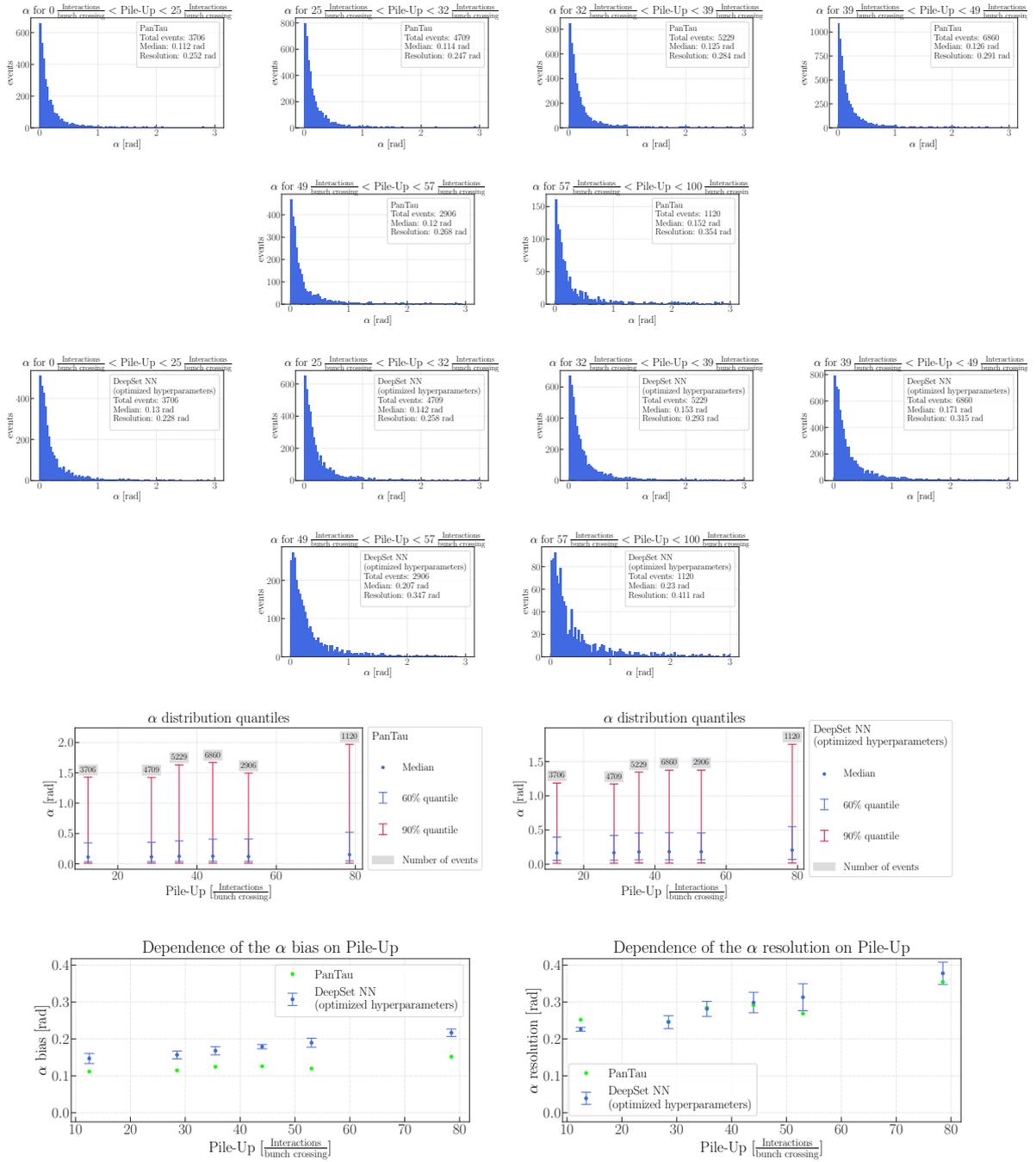


Figure A.36: α plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$, Cut: $E(\pi_{\text{true}}^0) \geq 25\text{GeV}$

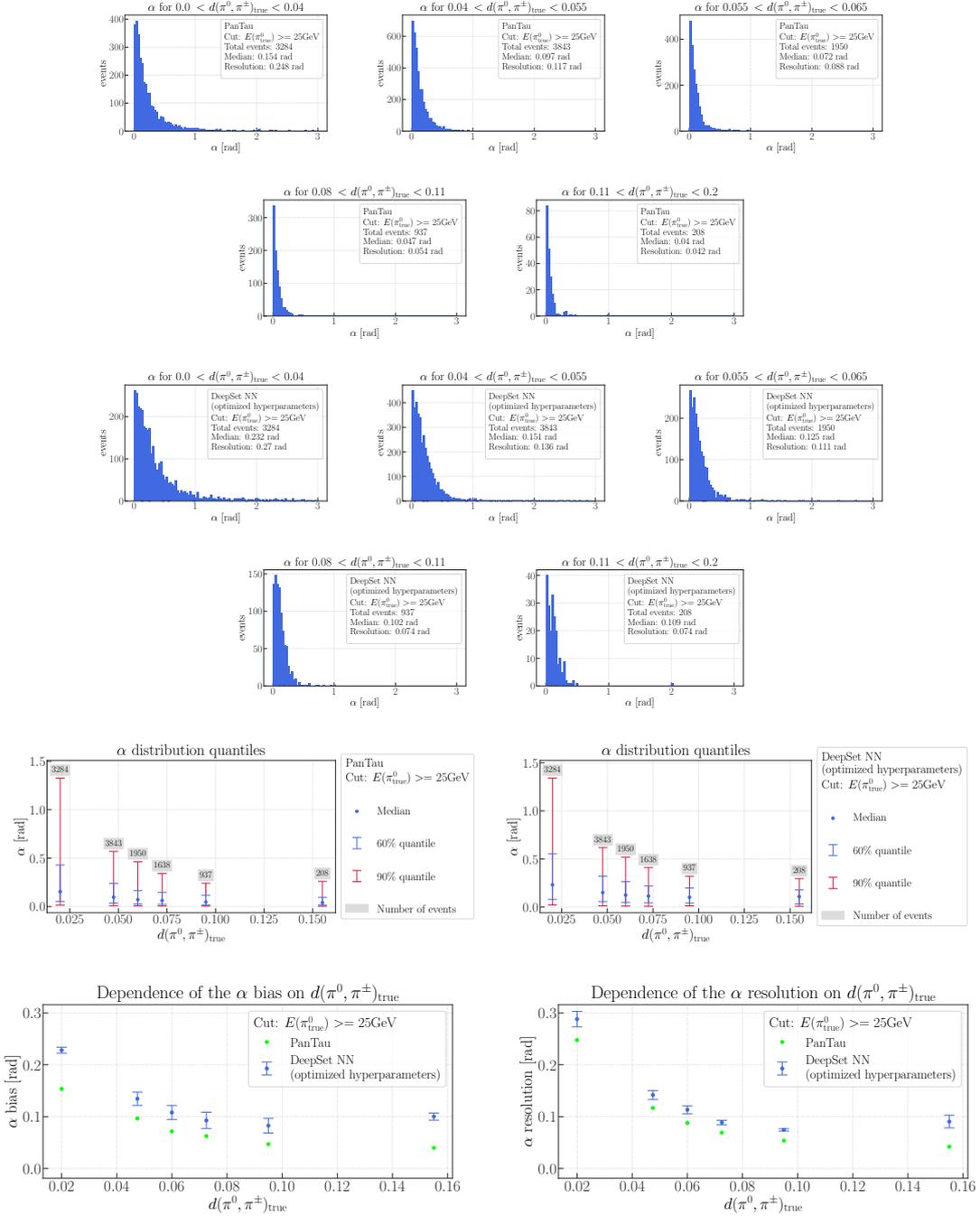


Figure A.37: α plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $E(\pi_{\text{true}}^0) \geq 25\text{GeV}$ is applied.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$, Cut: $E(\pi_{\text{true}}^0) < 25\text{GeV}$

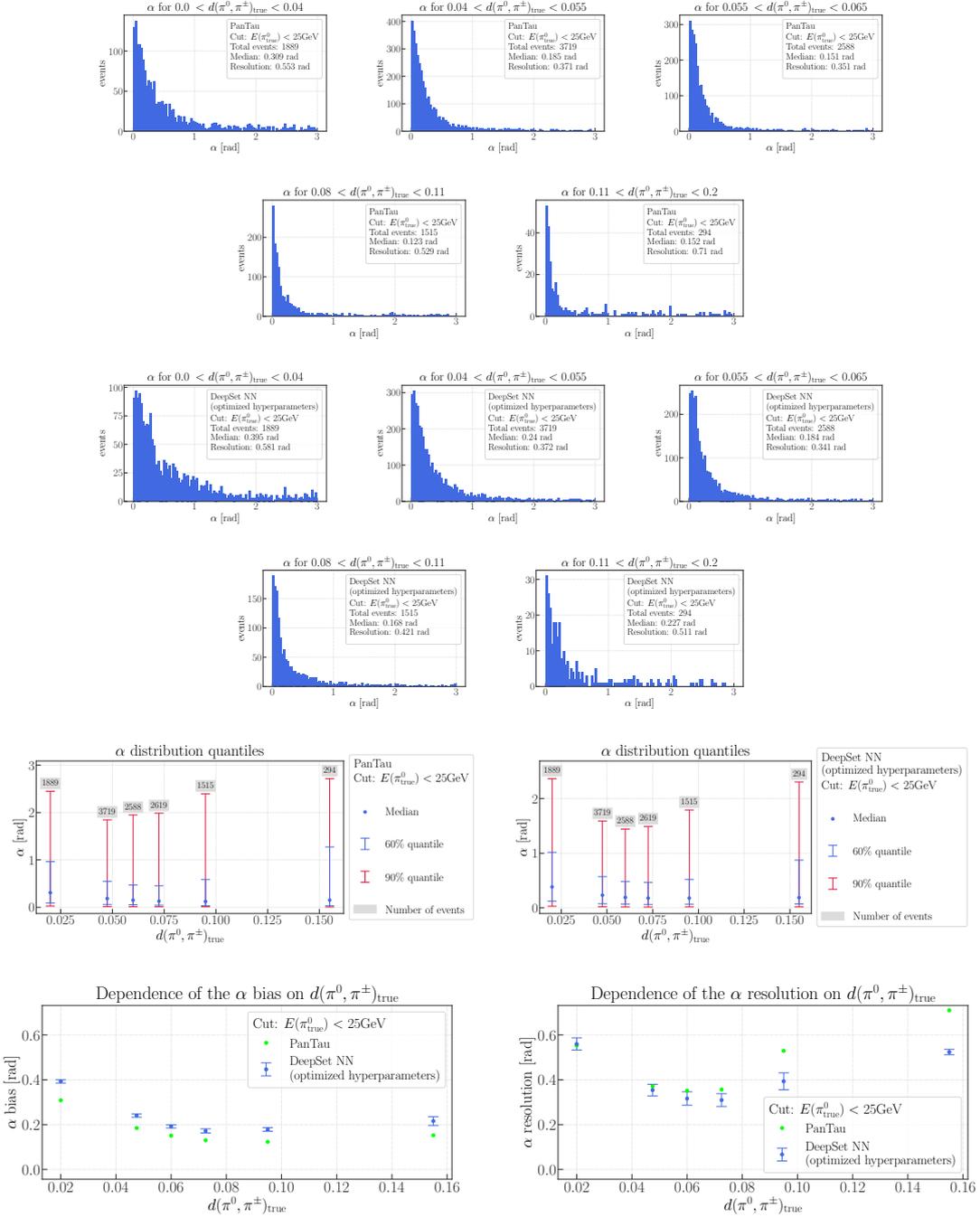


Figure A.38: α plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $E(\pi_{\text{true}}^0) < 25\text{GeV}$ is applied.

Figure of Merit: $\Delta\rho$ mass

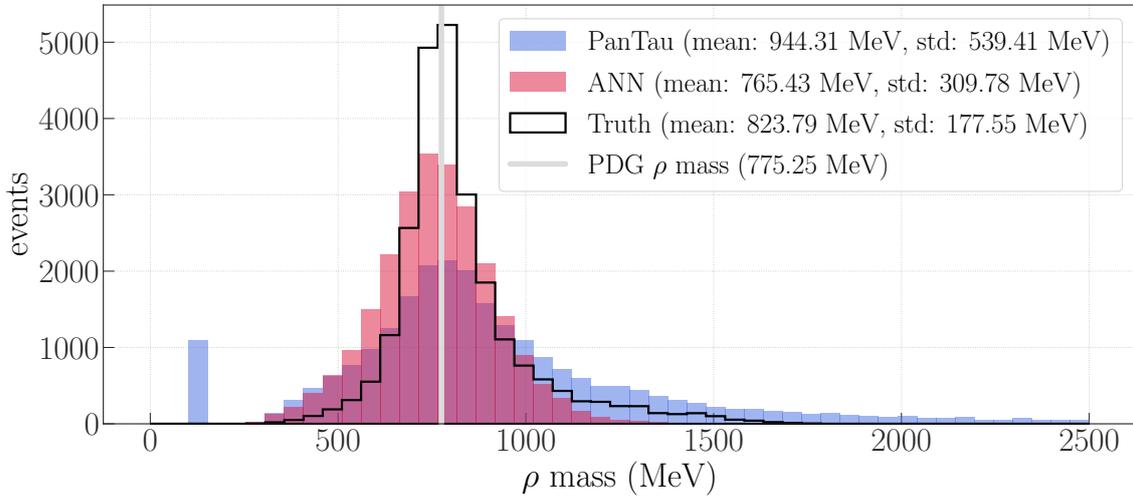


Figure A.39: ρ mass distribution plot on truth level as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $p_T(\tau_{\text{vis,true}})$

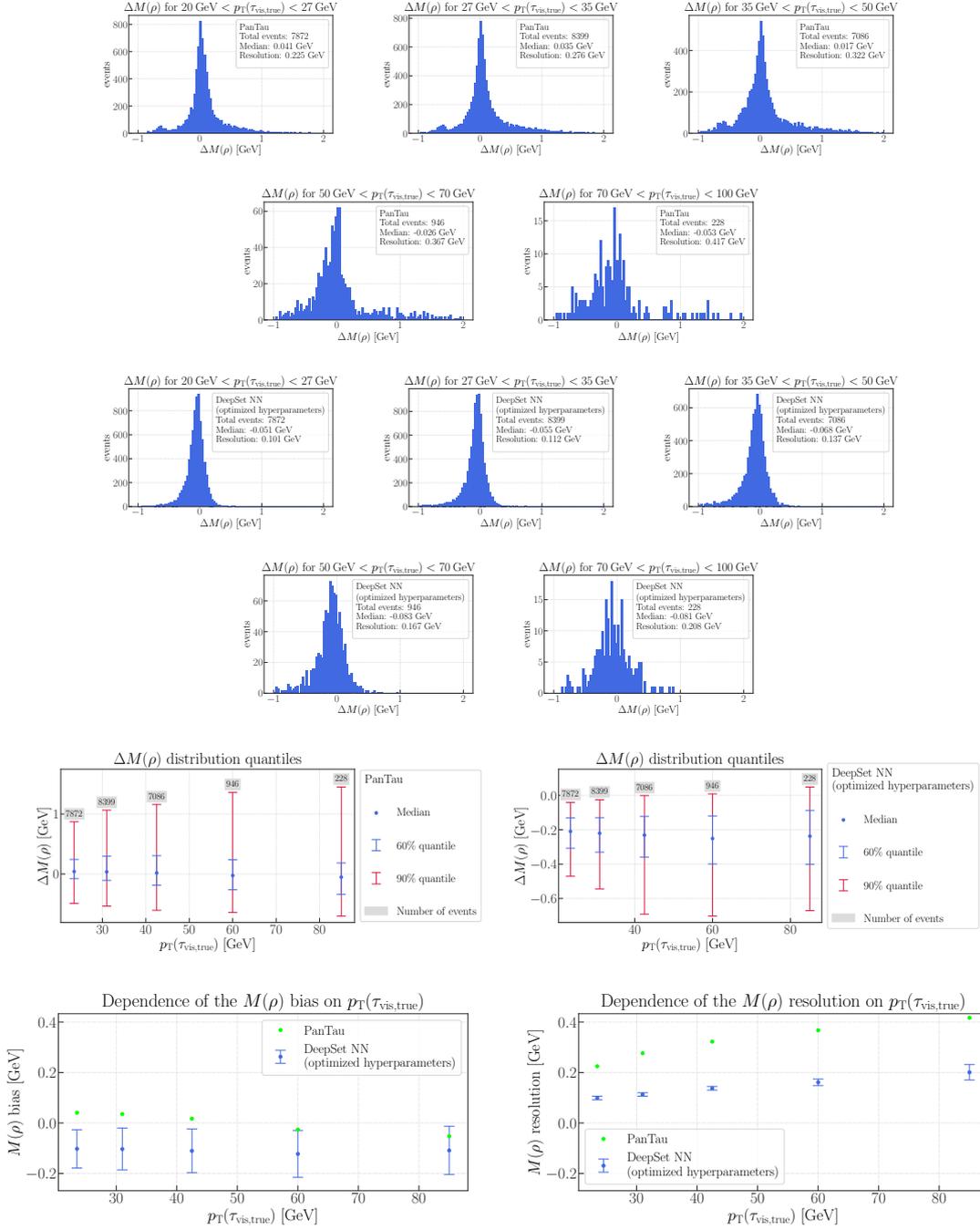


Figure A.40: $\Delta\rho$ mass plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $E(\pi_{\text{true}}^0)$

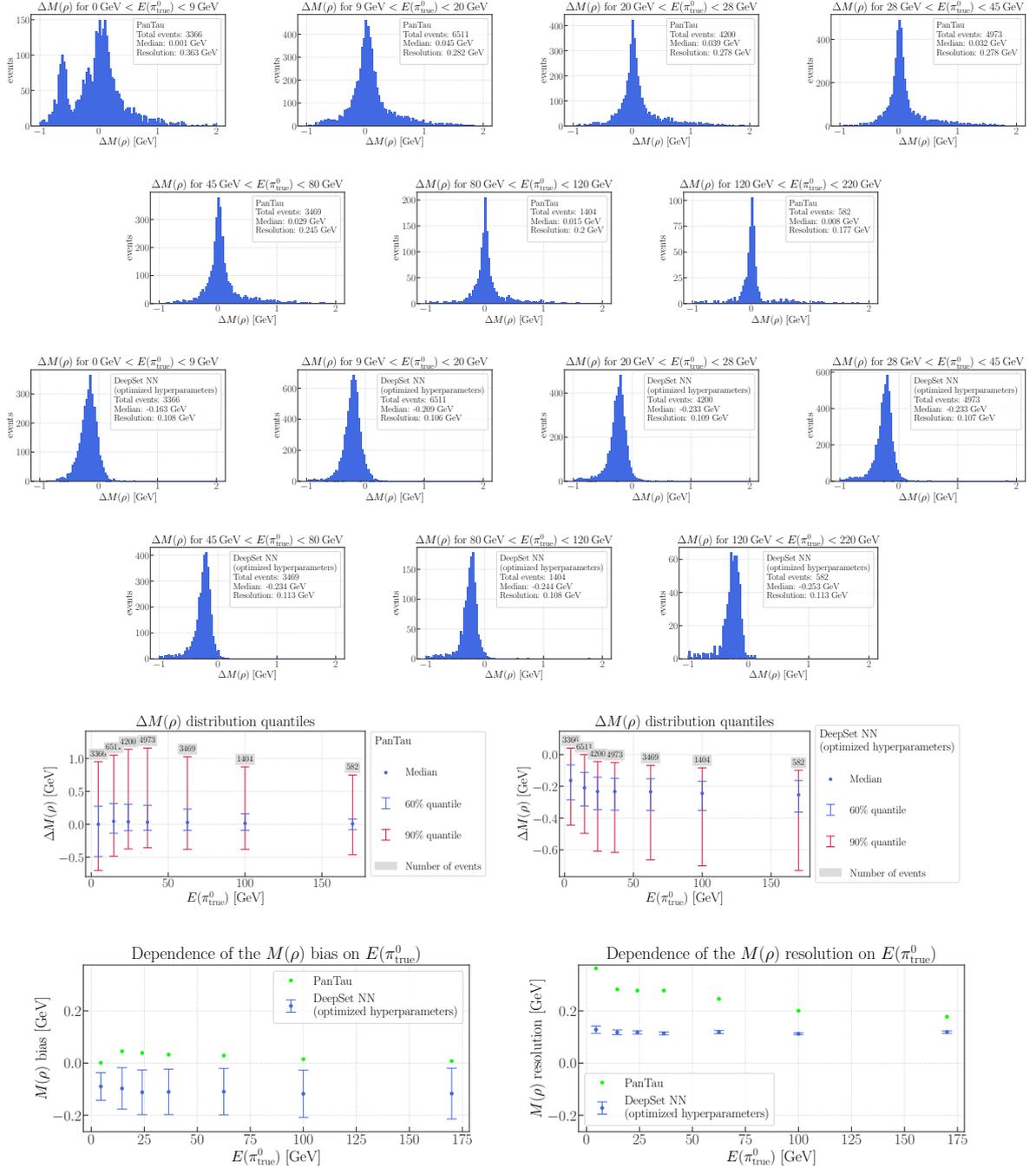


Figure A.41: $\Delta\rho$ mass plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\eta(\pi_{\text{true}}^0)$

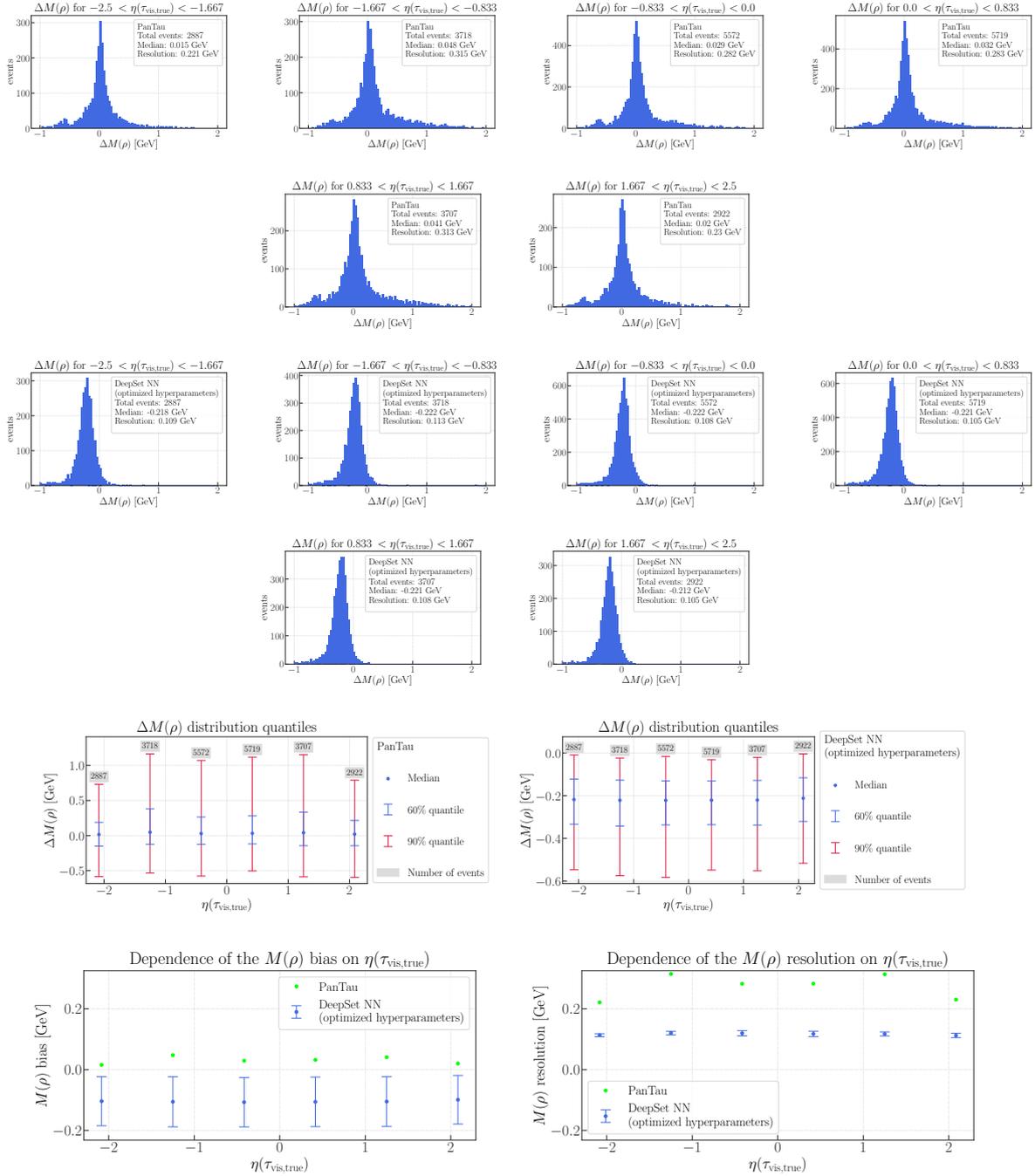


Figure A.42: $\Delta\rho$ mass plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $\phi(\pi_{\text{true}}^0)$

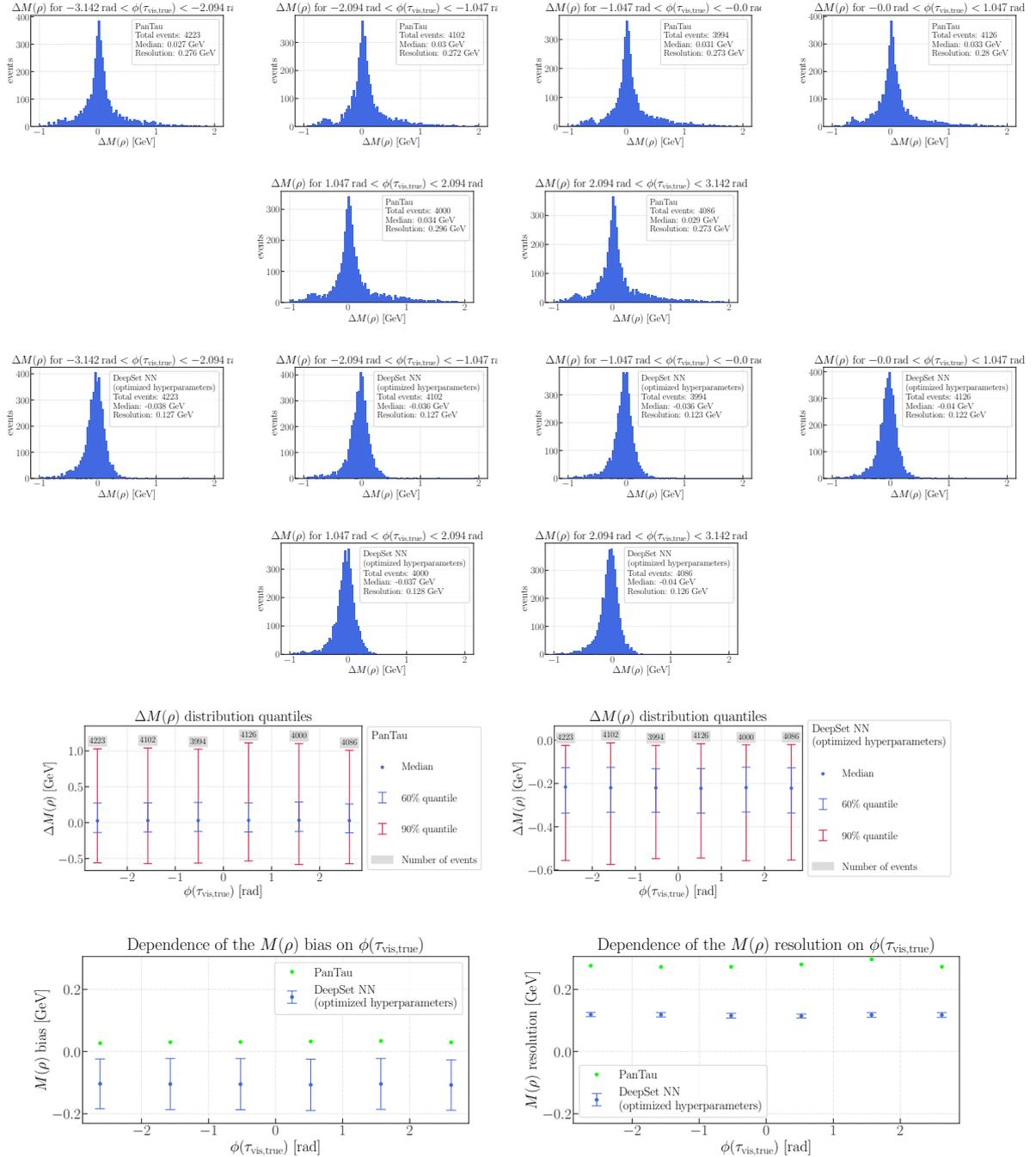


Figure A.43: $\Delta\rho$ mass plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: Pile-up

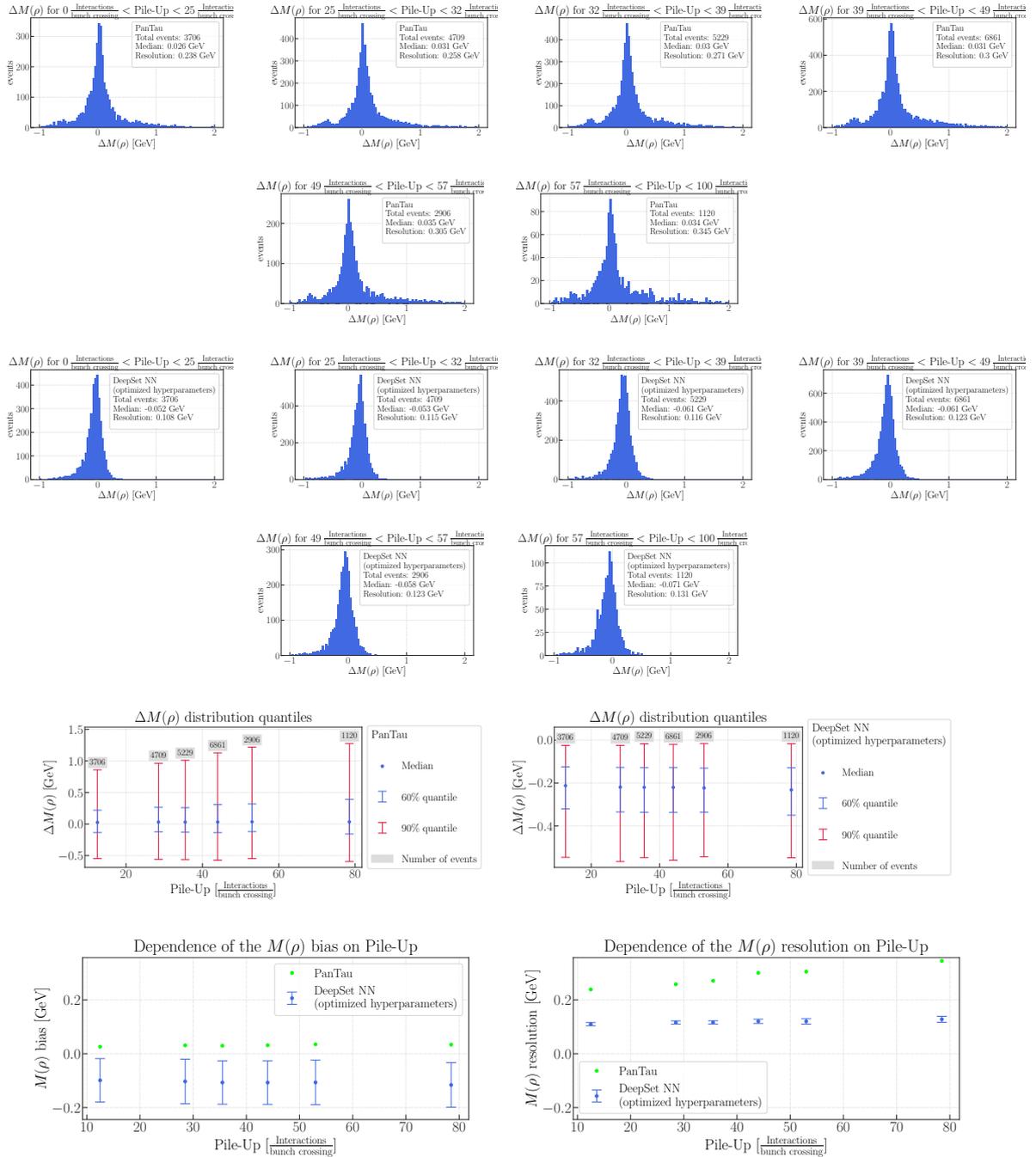


Figure A.44: $\Delta\rho$ mass plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

Appendix A Neutral Pion Reconstruction: Additional Figures

Binning: $d(\pi^0, \pi^\pm)$

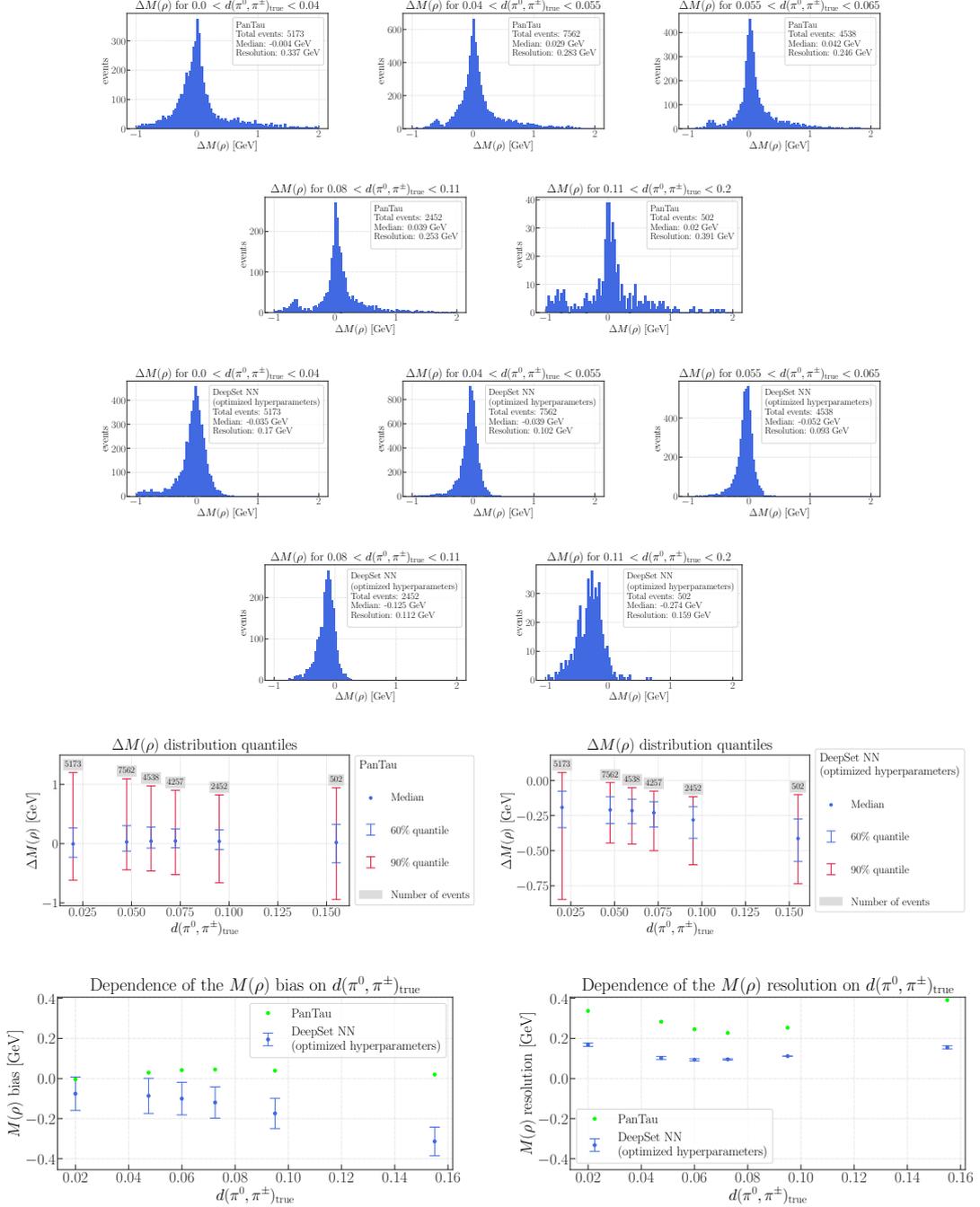


Figure A.45: $\Delta\rho$ mass plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.

A.2 Learning Curves for All Calls of the Hyperparameter Optimization

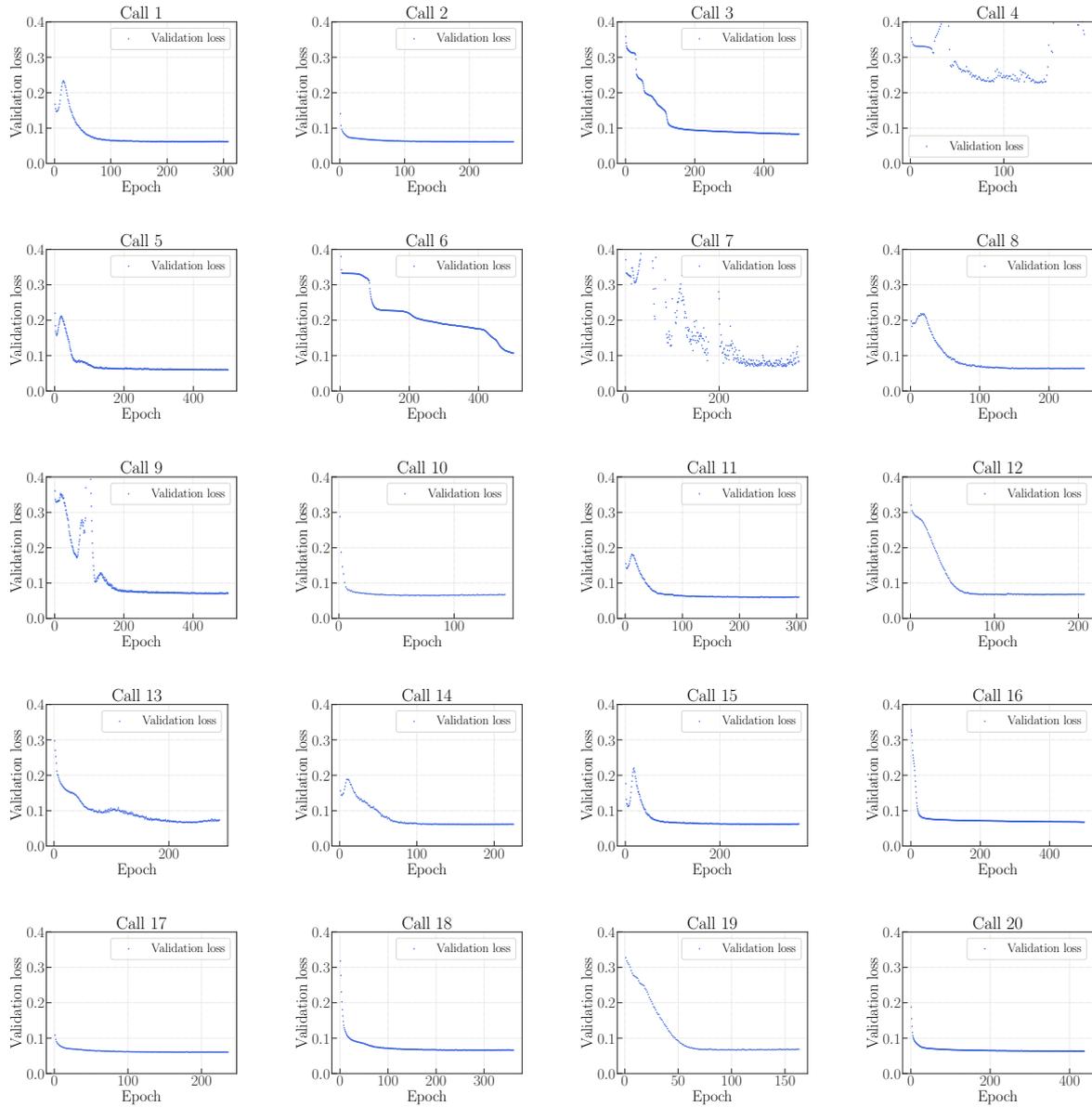


Figure A.46: Learning curves for calls 1 through 20 of the hyperparameter optimization.

Appendix A Neutral Pion Reconstruction: Additional Figures

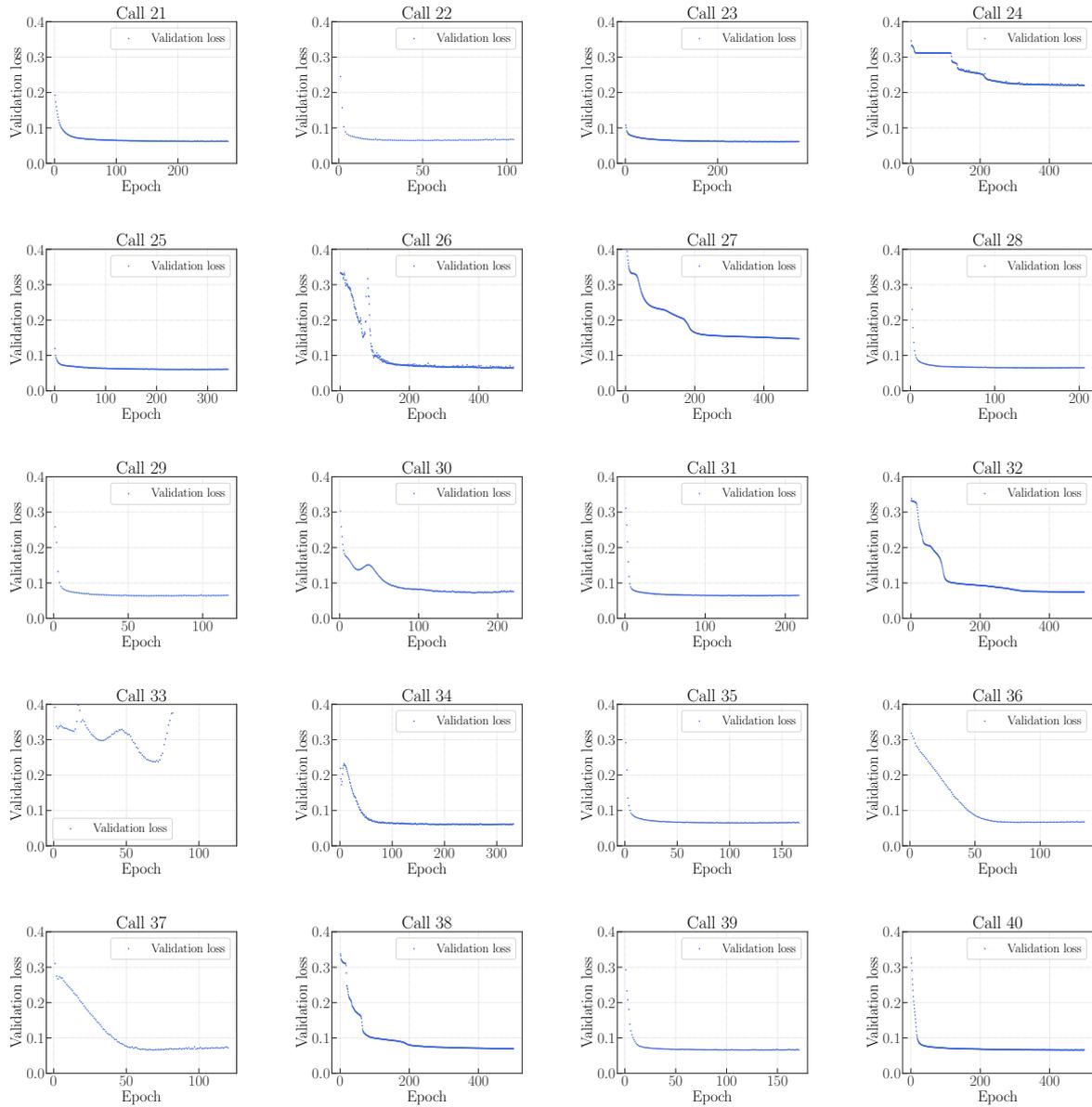


Figure A.47: Learning curves for calls 21 through 40 of the hyperparameter optimization.

Appendix A Neutral Pion Reconstruction: Additional Figures

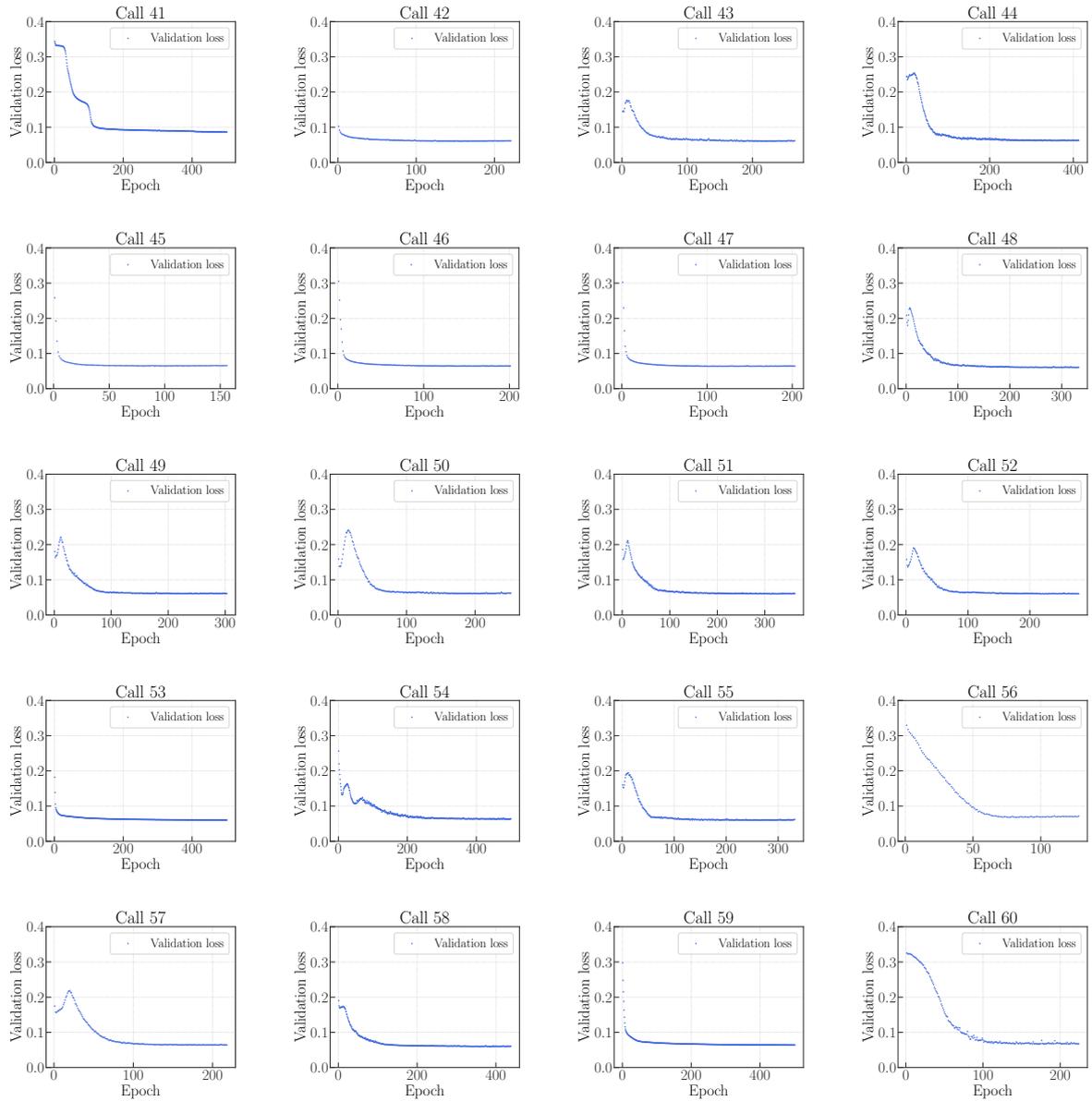


Figure A.48: Learning curves for calls 41 through 60 of the hyperparameter optimization.

Appendix A Neutral Pion Reconstruction: Additional Figures

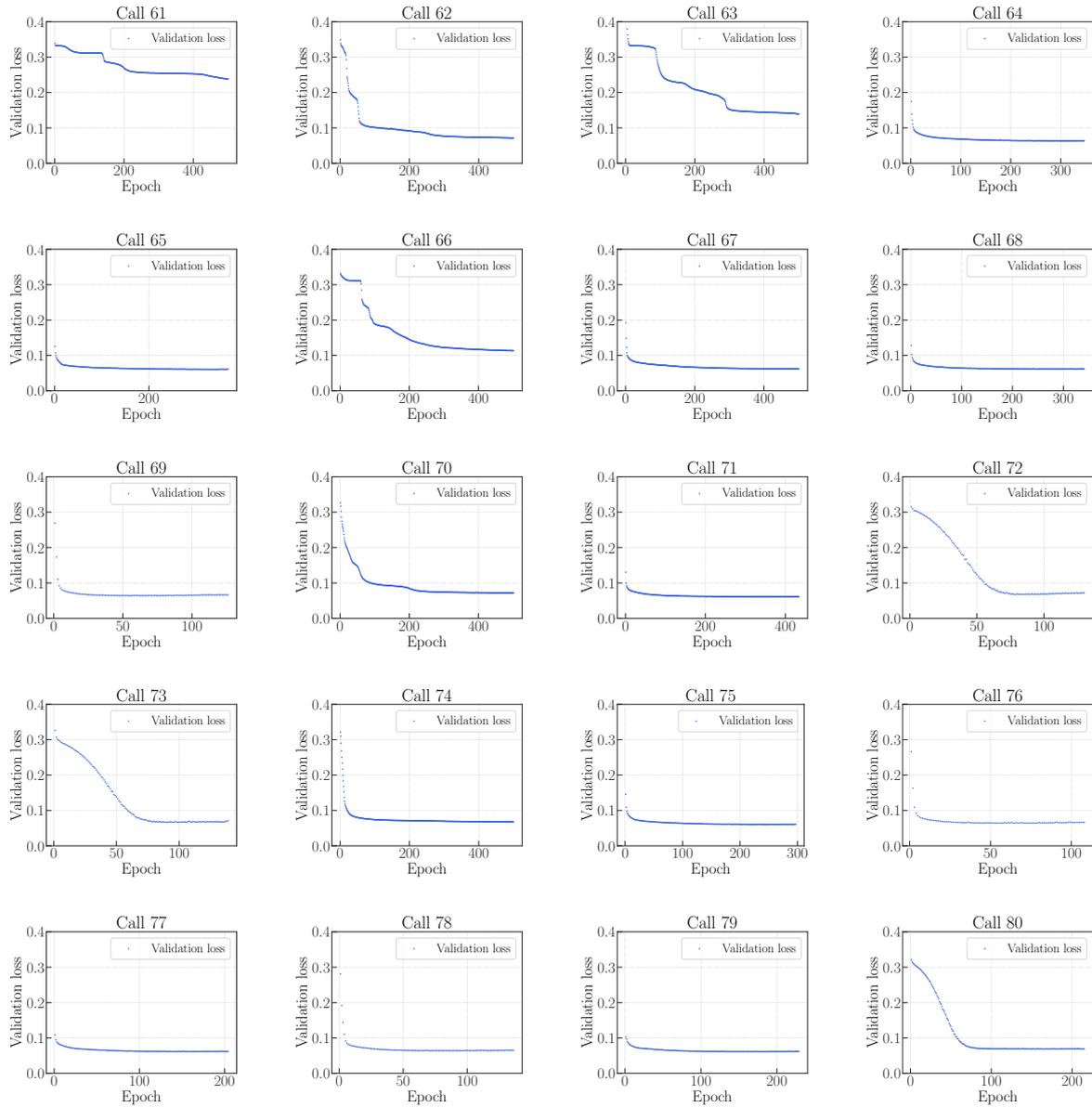


Figure A.49: Learning curves for calls 61 through 80 of the hyperparameter optimization.

Appendix A Neutral Pion Reconstruction: Additional Figures

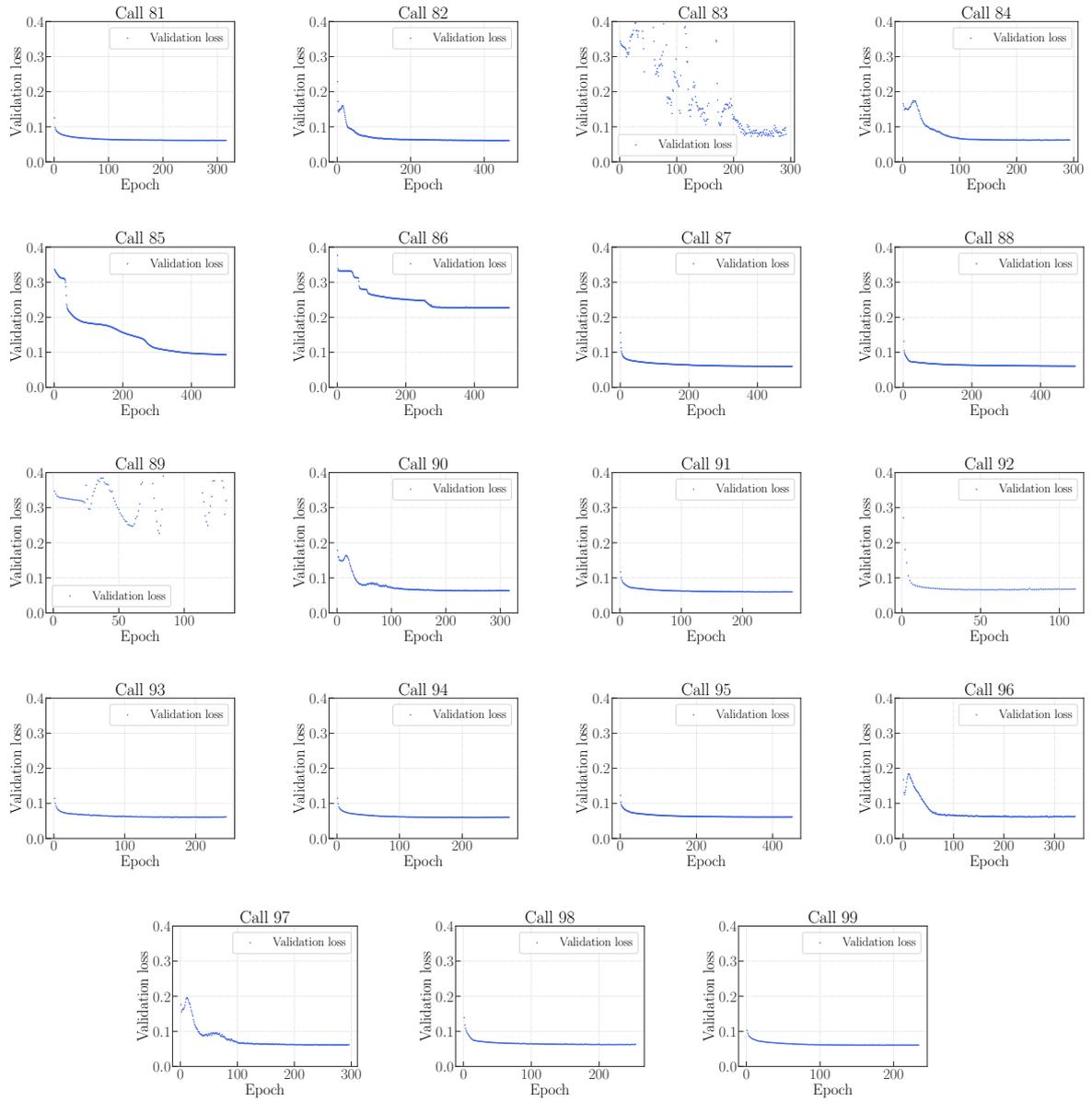


Figure A.50: Learning curves for calls 81 through 100 of the hyperparameter optimization.

Validation on LHC Data: Additional Figures

B.1 Decay Mode: All

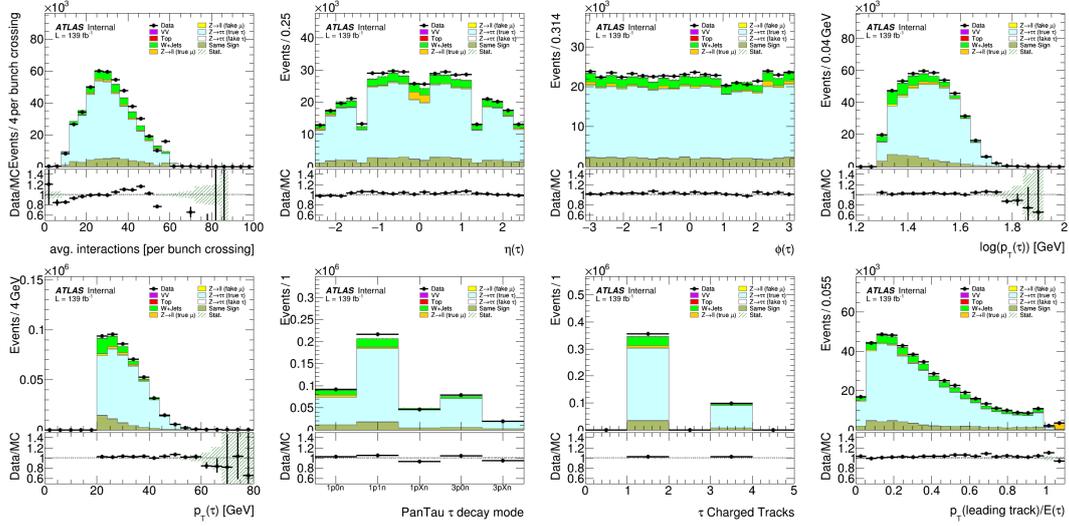


Figure B.1: Data Monte Carlo Comparison plots for all decay modes for General Observables.

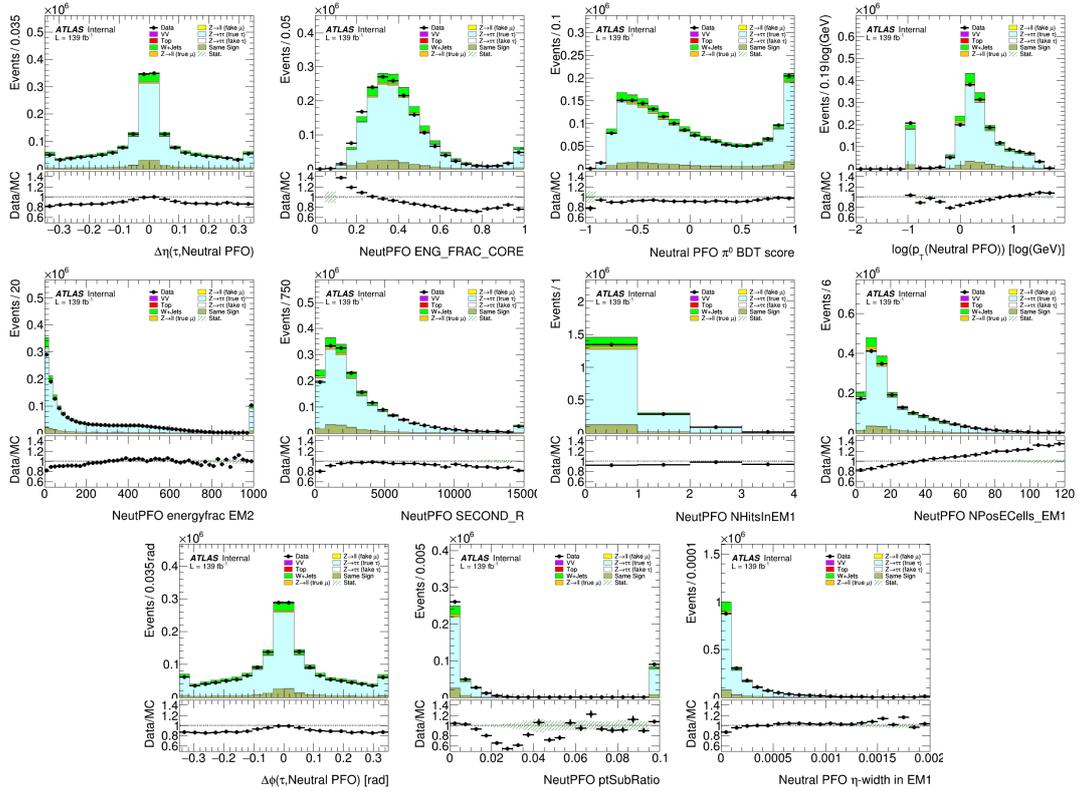


Figure B.2: Data Monte Carlo Comparison plots for all decay modes for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

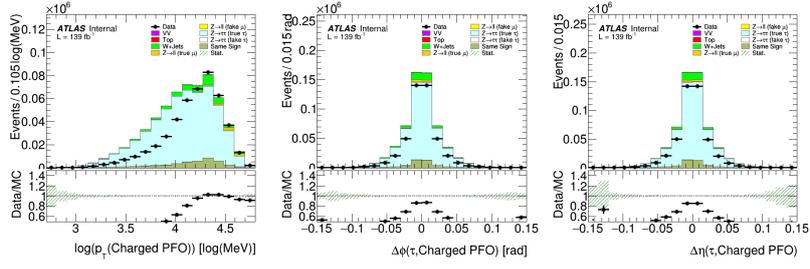


Figure B.3: Data Monte Carlo Comparison plots for all decay modes for Charged PFO Variables.

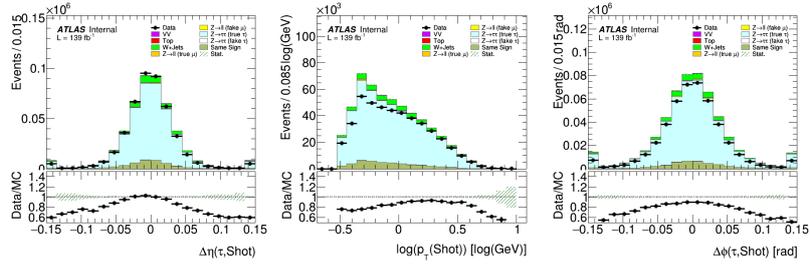


Figure B.4: Data Monte Carlo Comparison plots for all decay modes for Photon Shot Variables.

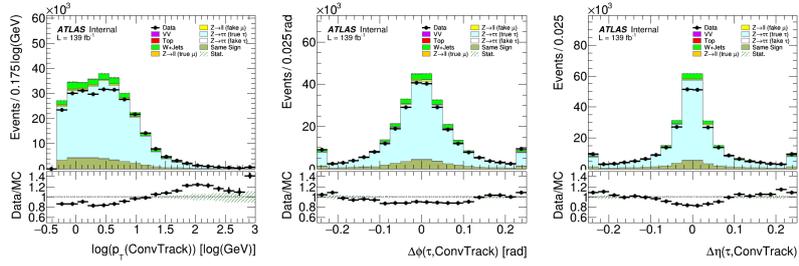


Figure B.5: Data Monte Carlo Comparison plots for all decay modes for Conversion Track Variables.

Appendix B Validation on LHC Data: Additional Figures

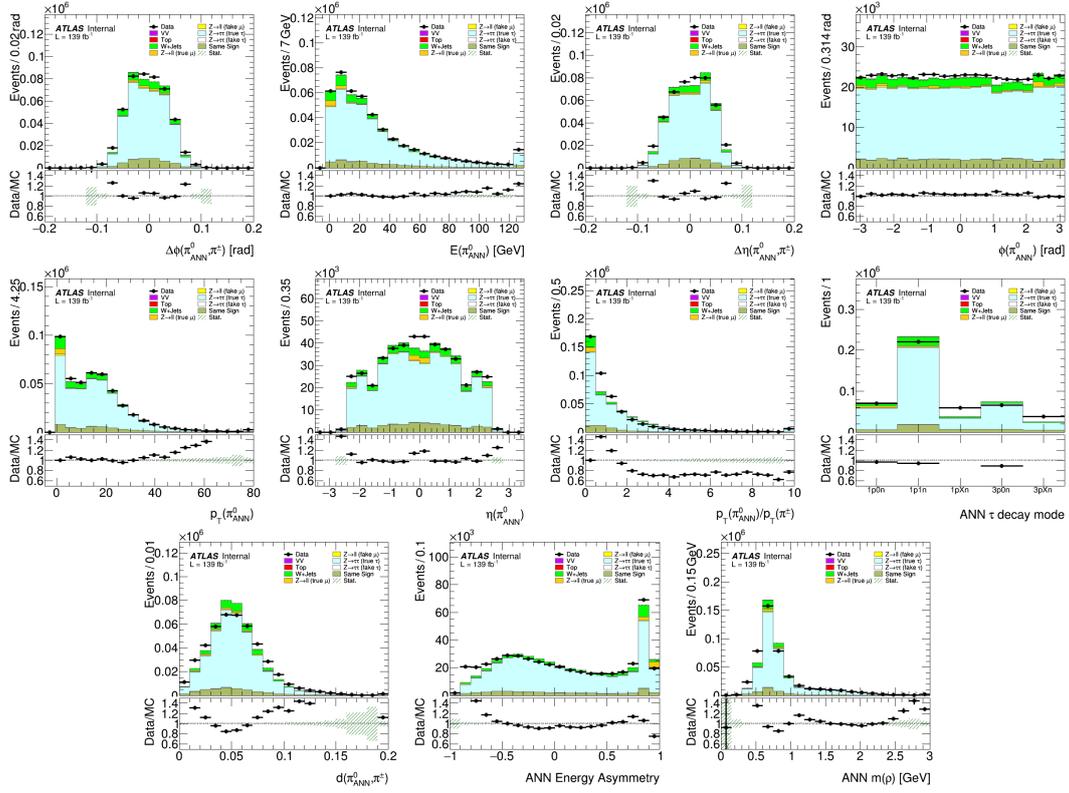
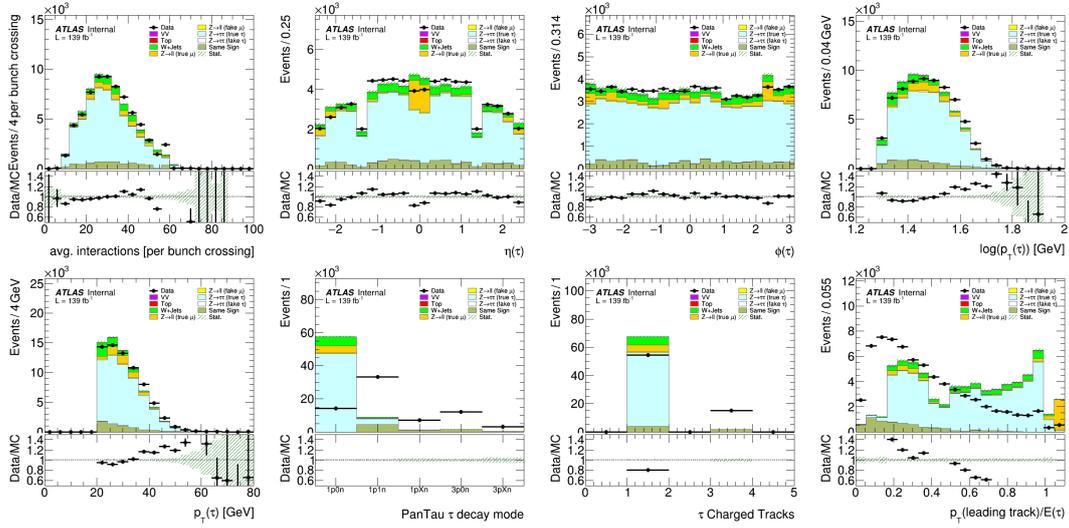
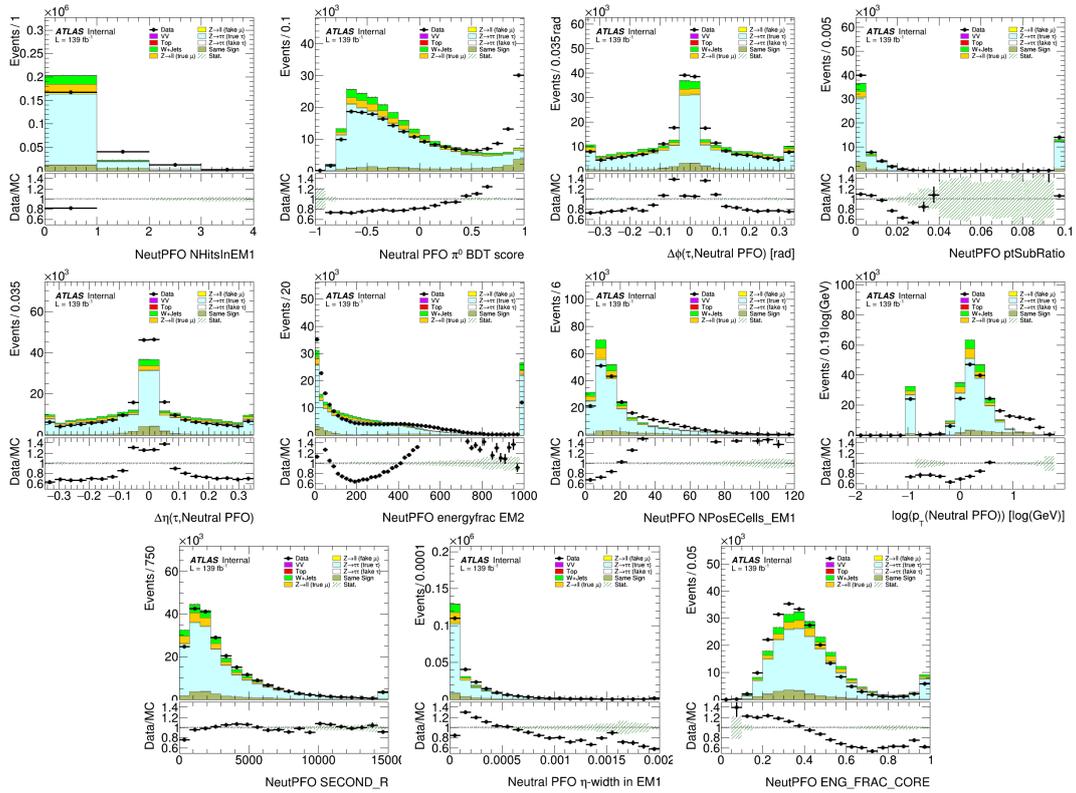


Figure B.6: Data Monte Carlo Comparison plots for all decay modes for Neural Network Output Variables.

B.2 Decay Mode: $1p0n$

 Figure B.7: Data Monte Carlo Comparison plots for the decay mode $1p0n$ for General Observables.

 Figure B.8: Data Monte Carlo Comparison plots for the decay mode $1p0n$ for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

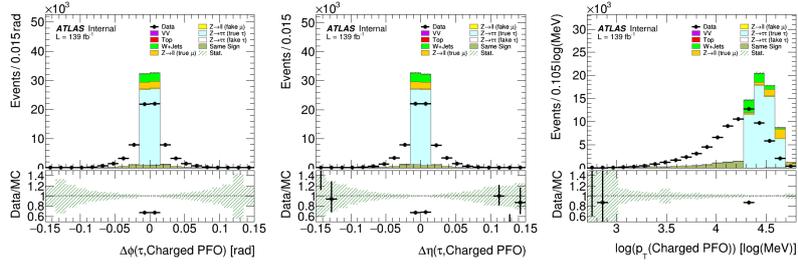


Figure B.9: Data Monte Carlo Comparison plots for the decay mode $1p0n$ for Charged PFO Variables.

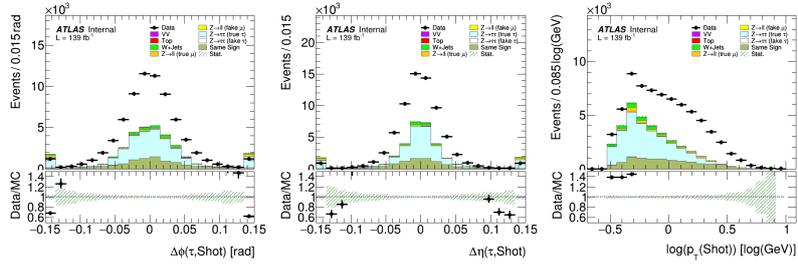


Figure B.10: Data Monte Carlo Comparison plots for the decay mode $1p0n$ for Photon Shot Variables.

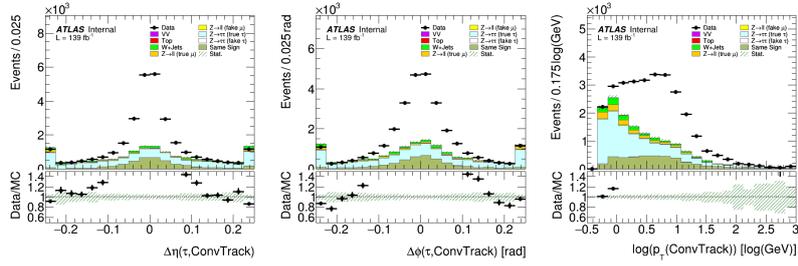


Figure B.11: Data Monte Carlo Comparison plots for the decay mode $1p0n$ for Conversion Track Variables.

Appendix B Validation on LHC Data: Additional Figures

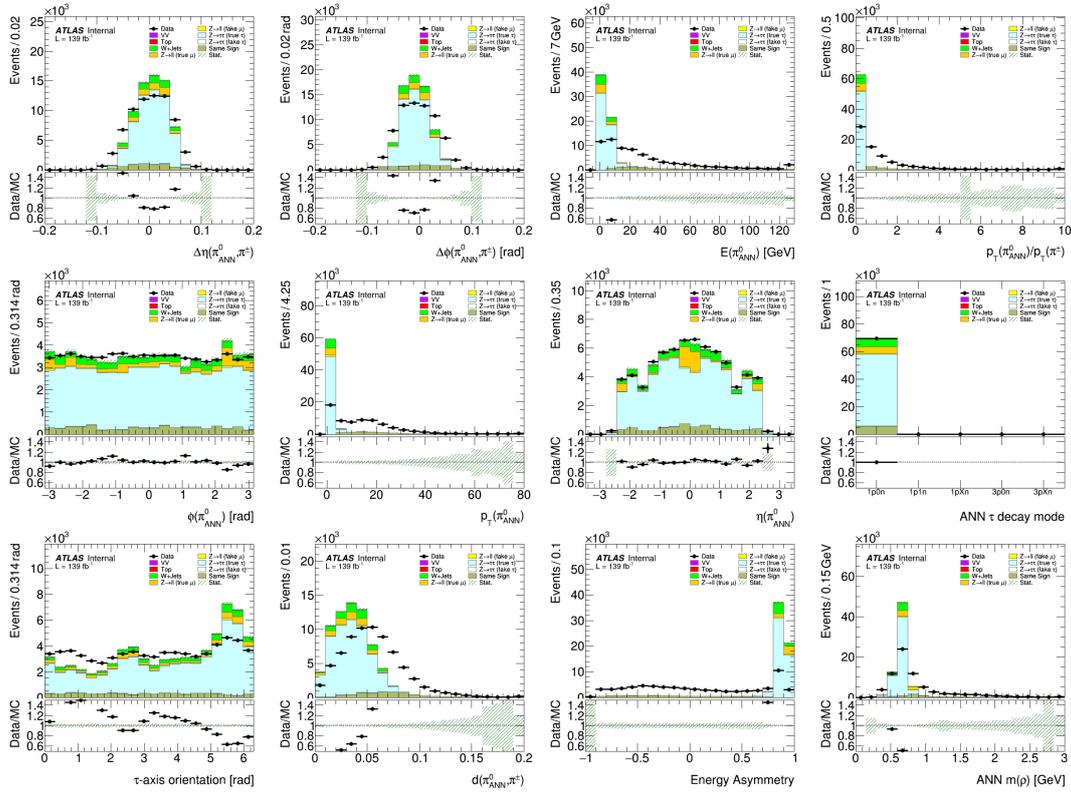


Figure B.12: Data Monte Carlo Comparison plots for the decay mode 1p0n for Neural Network Output Variables.

B.3 Decay Mode: 1p1n

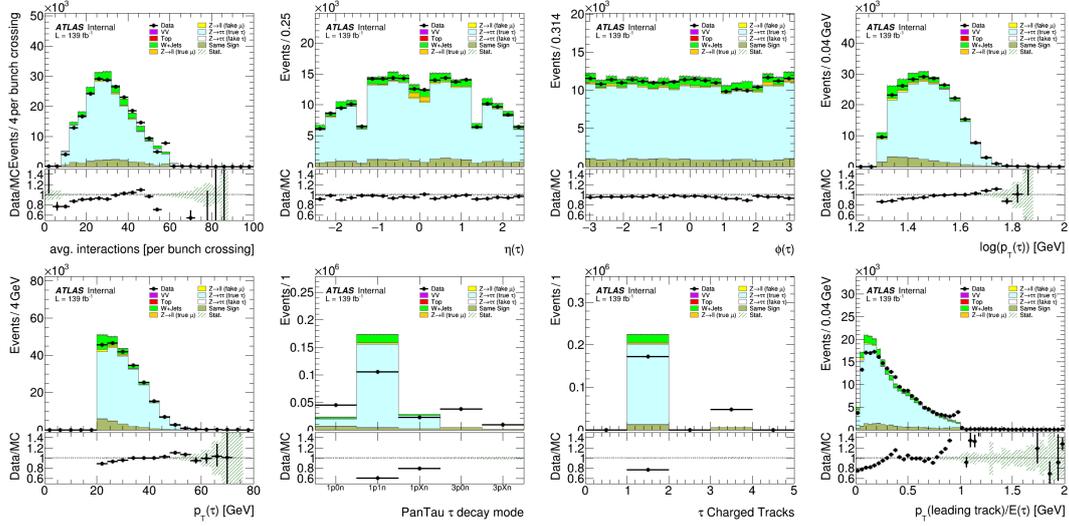


Figure B.13: Data Monte Carlo Comparison plots for the decay mode 1p1n for General Observables.

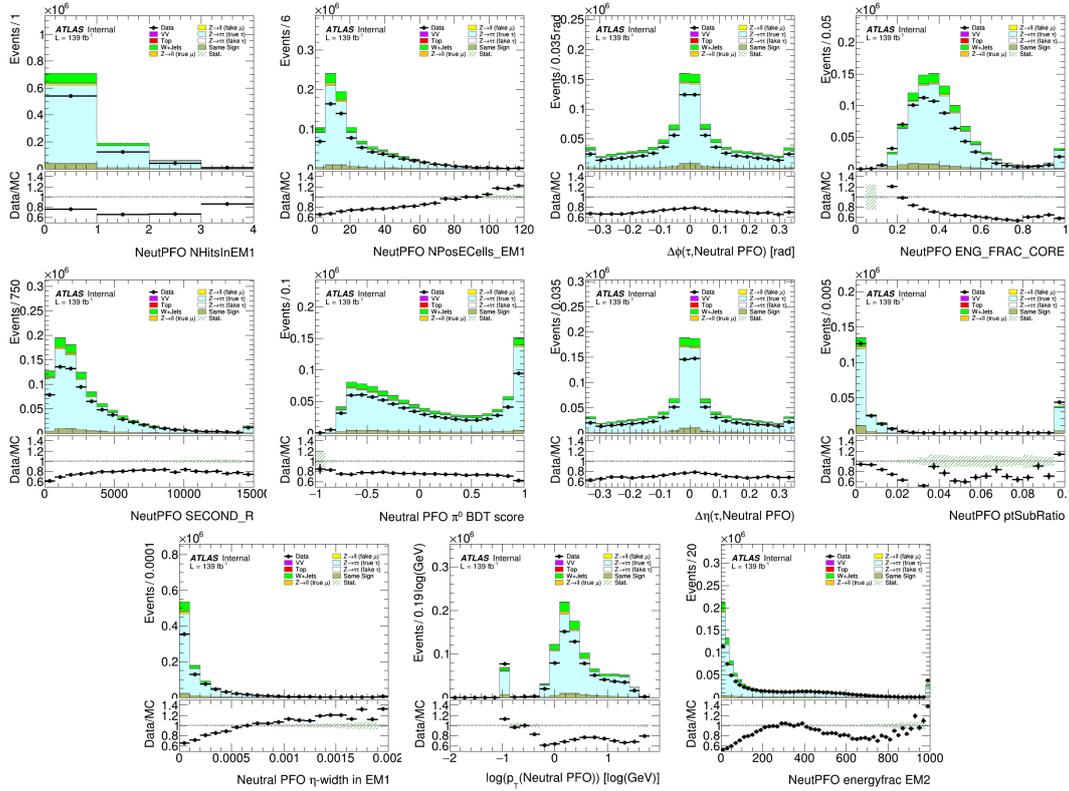


Figure B.14: Data Monte Carlo Comparison plots for the decay mode 1p1n for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

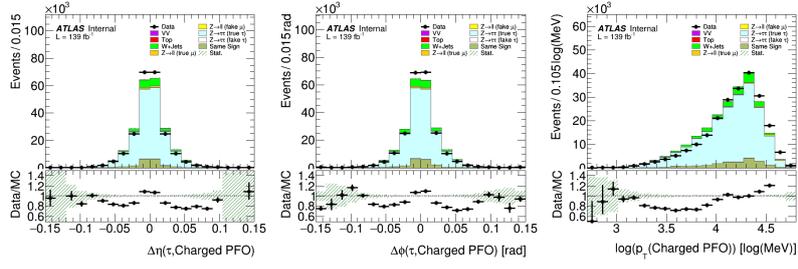


Figure B.15: Data Monte Carlo Comparison plots for the decay mode 1p1n for Charged PFO Variables.

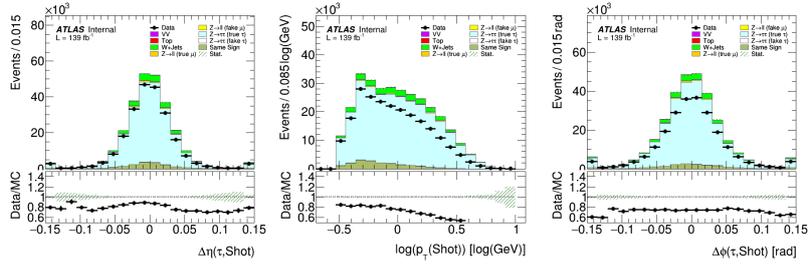


Figure B.16: Data Monte Carlo Comparison plots for the decay mode 1p1n for Photon Shot Variables.

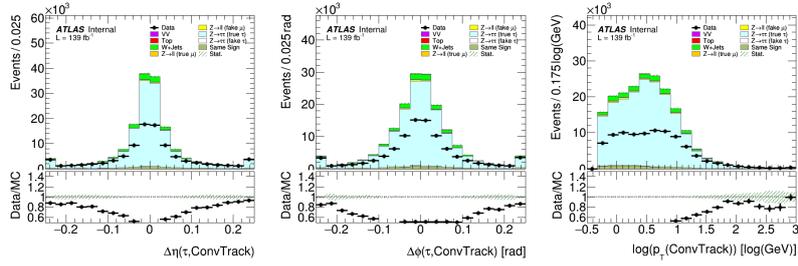


Figure B.17: Data Monte Carlo Comparison plots for the decay mode 1p1n for Conversion Track Variables.

Appendix B Validation on LHC Data: Additional Figures

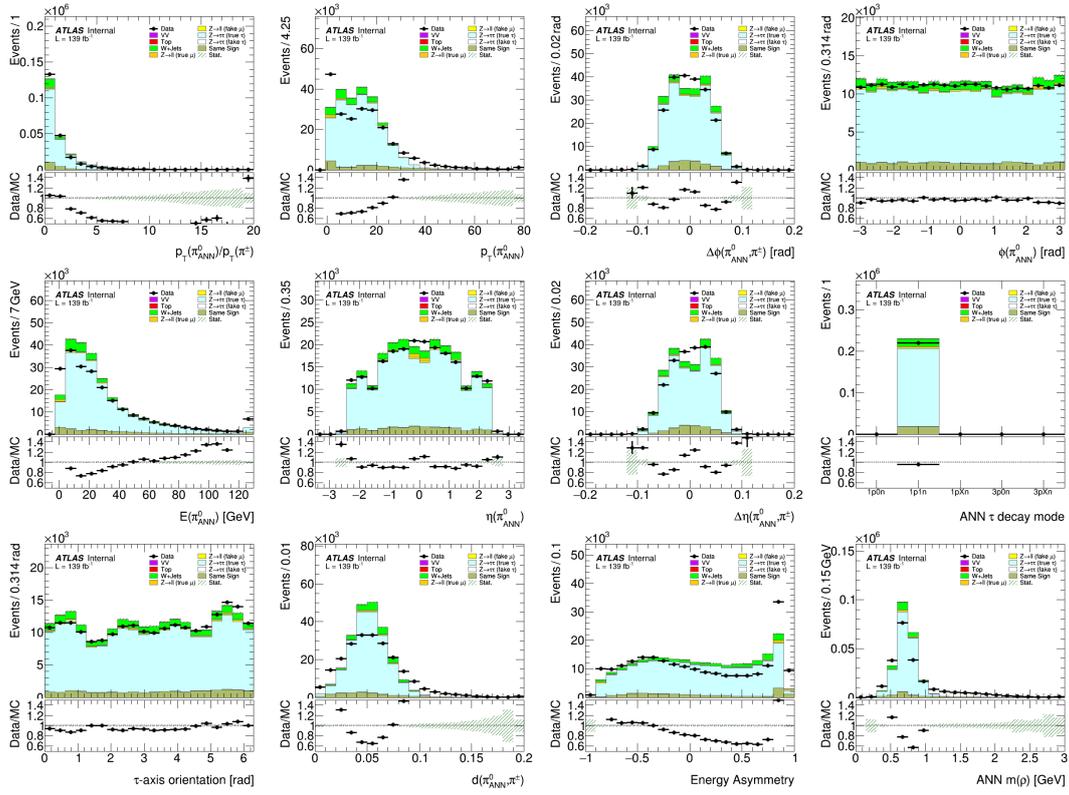
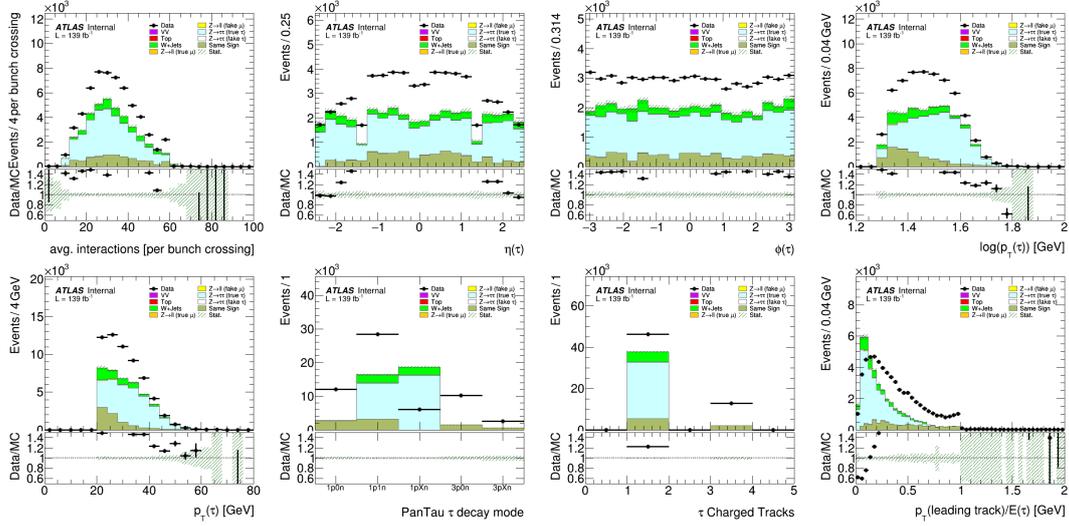
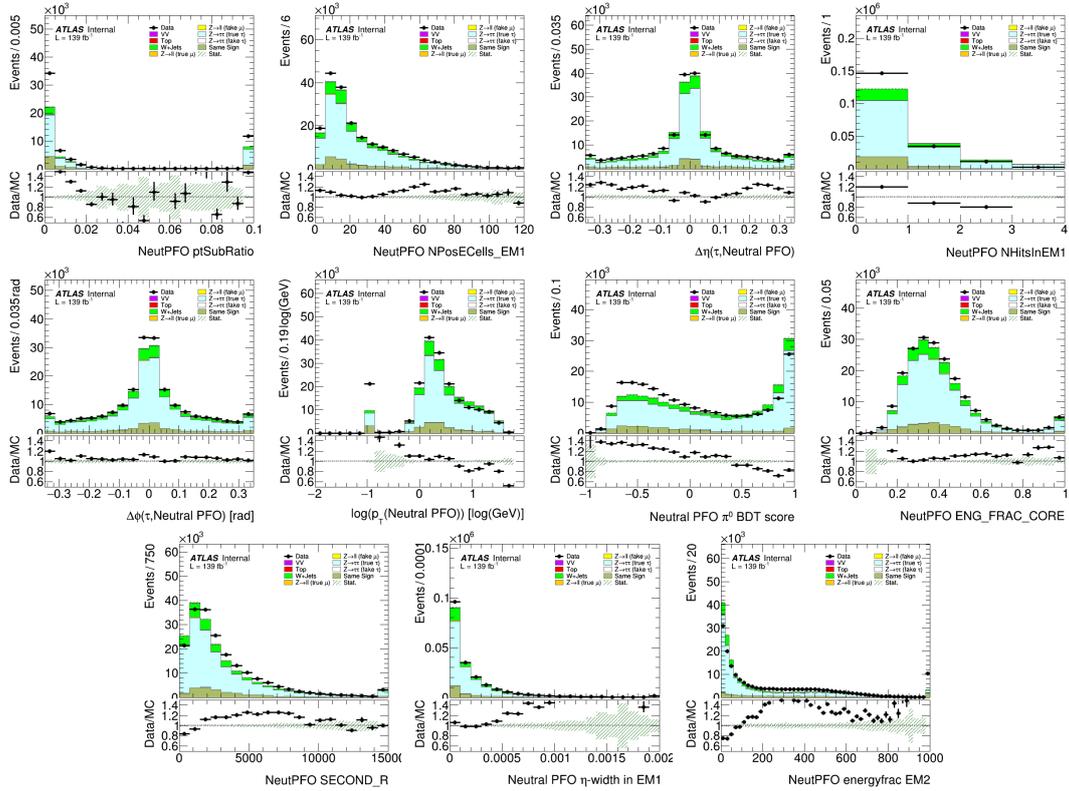


Figure B.18: Data Monte Carlo Comparison plots for the decay mode 1p1n for Neural Network Output Variables.

B.4 Decay Mode: $1pXn$

 Figure B.19: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for General Observables.

 Figure B.20: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

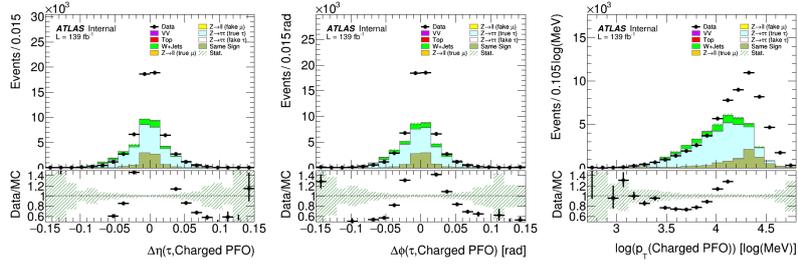


Figure B.21: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for Charged PFO Variables.

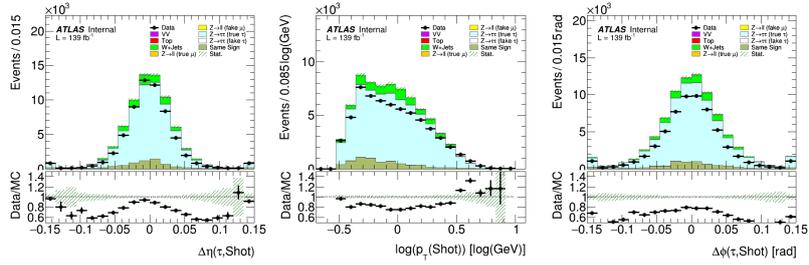


Figure B.22: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for Photon Shot Variables.

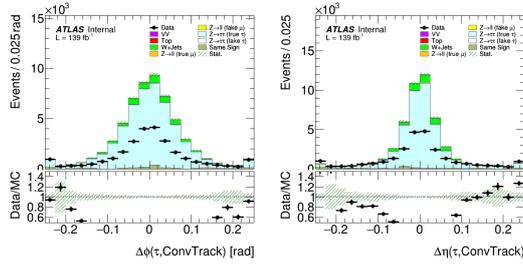


Figure B.23: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for Conversion Track Variables.

Appendix B Validation on LHC Data: Additional Figures

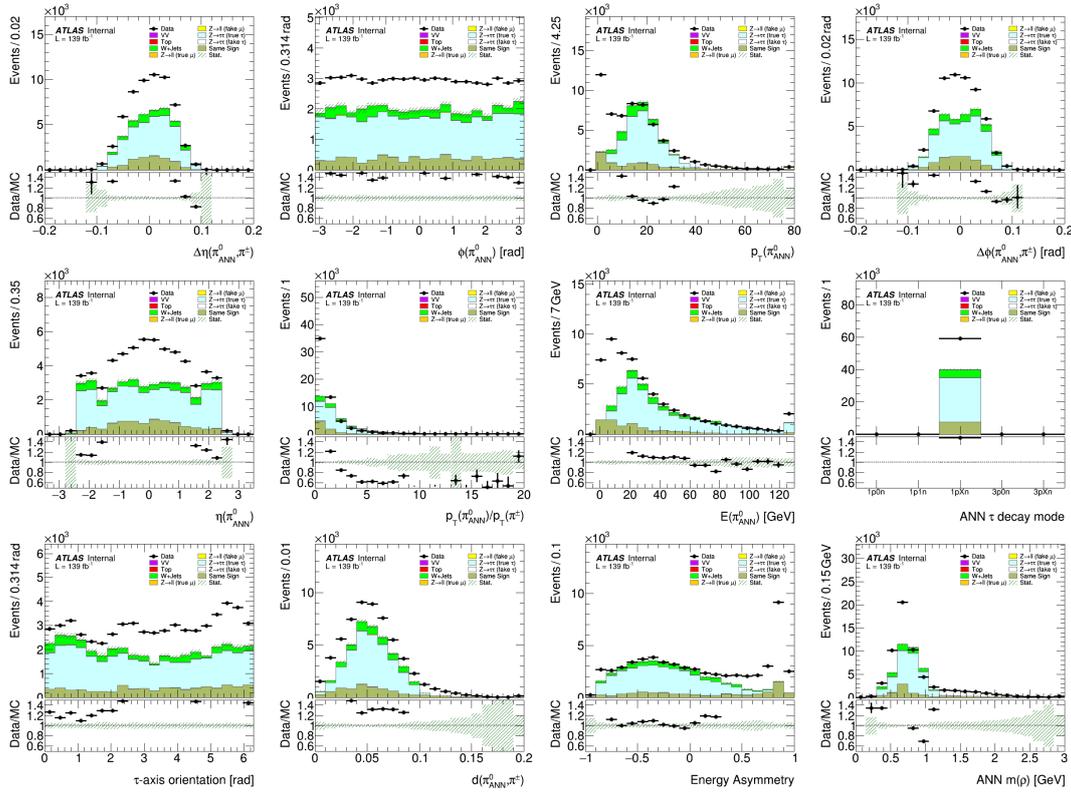
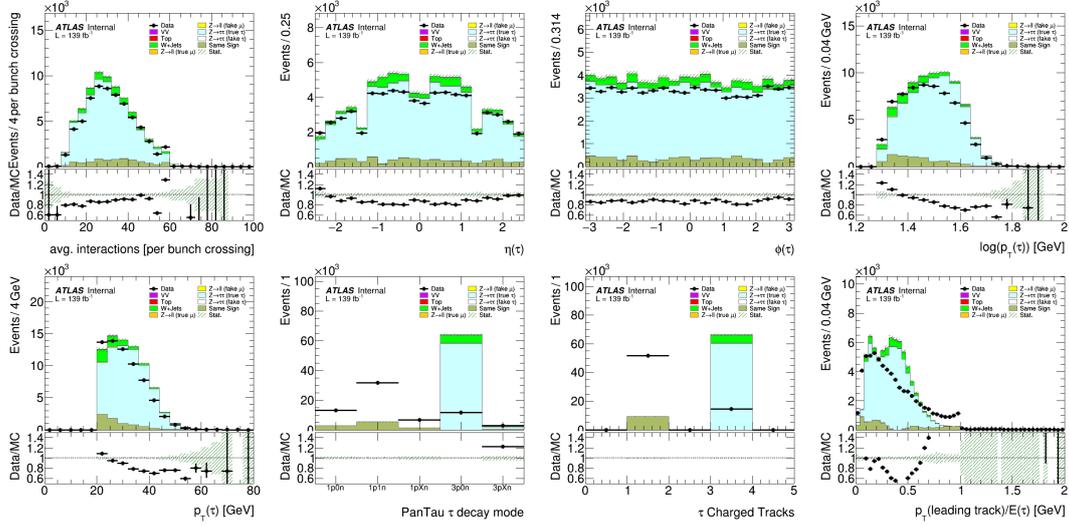
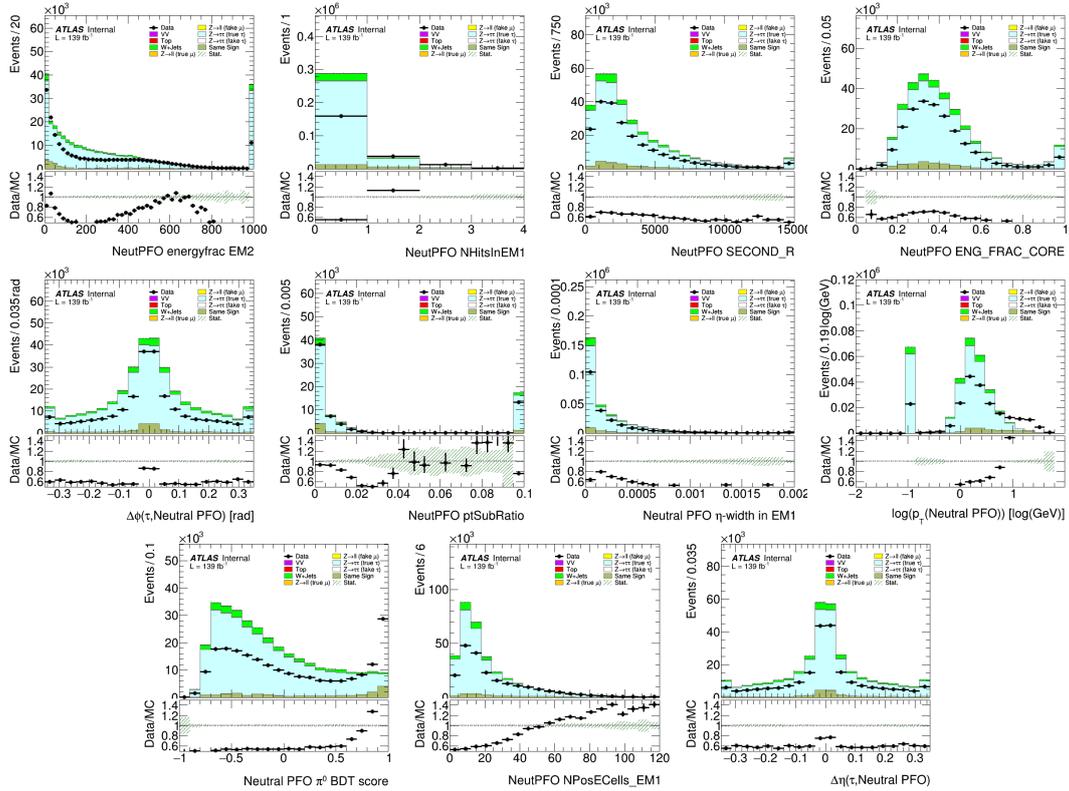


Figure B.24: Data Monte Carlo Comparison plots for the decay mode $1pXn$ for Neural Network Output Variables.

B.5 Decay Mode: $3p0n$

 Figure B.25: Data Monte Carlo Comparison plots for the decay mode $3p0n$ for General Observables.

 Figure B.26: Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

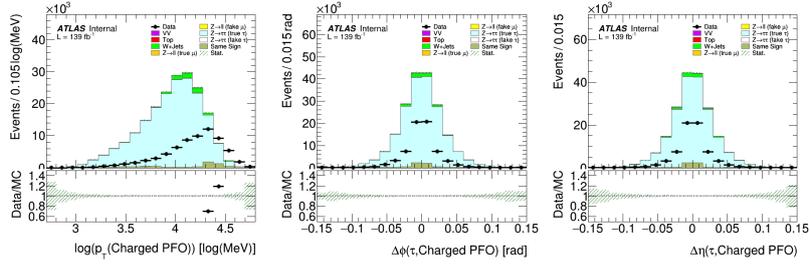


Figure B.27: Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Charged PFO Variables.

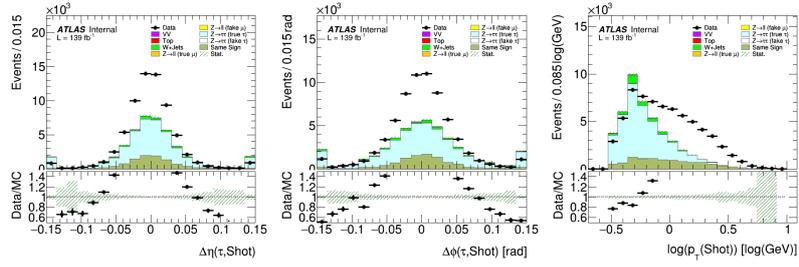


Figure B.28: Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Photon Shot Variables.

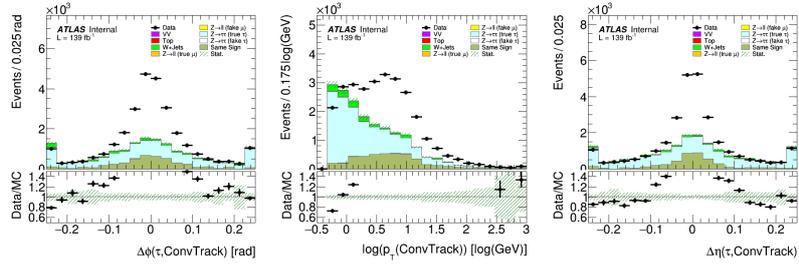
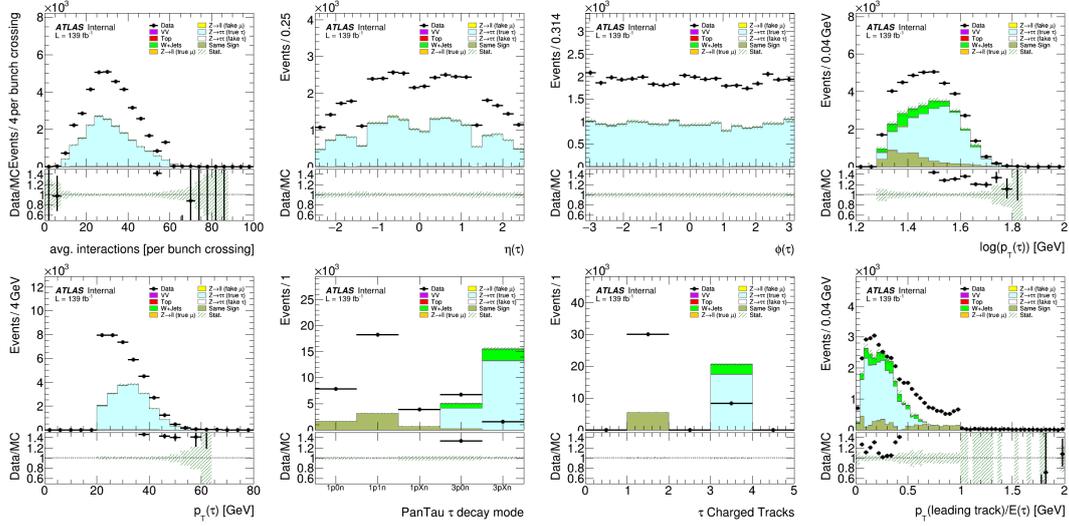
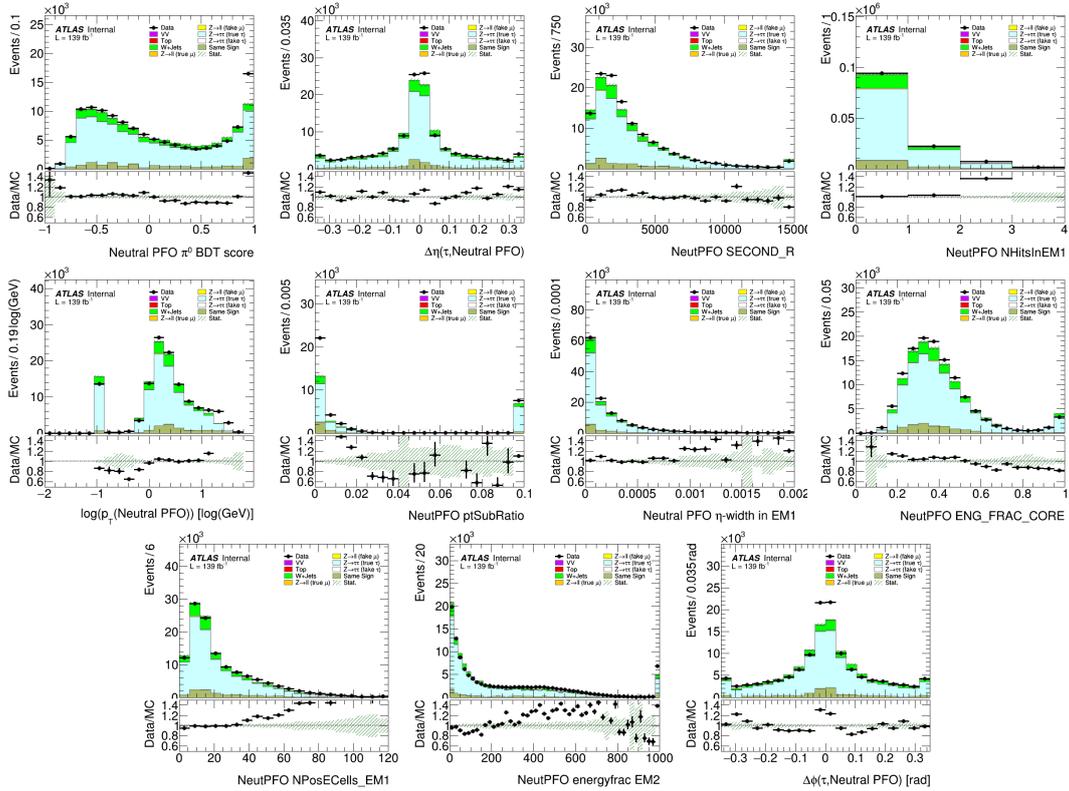


Figure B.29: Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Conversion Track Variables.

B.6 Decay Mode: $3pXn$

 Figure B.31: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for General Observables.

 Figure B.32: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Neutral PFO Variables.

Appendix B Validation on LHC Data: Additional Figures

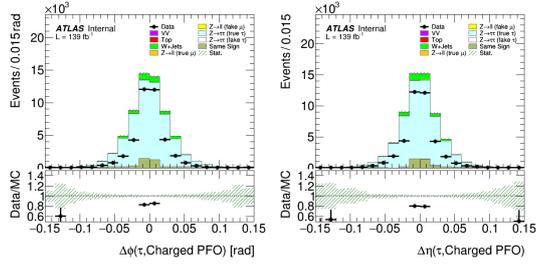


Figure B.33: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Charged PFO Variables.

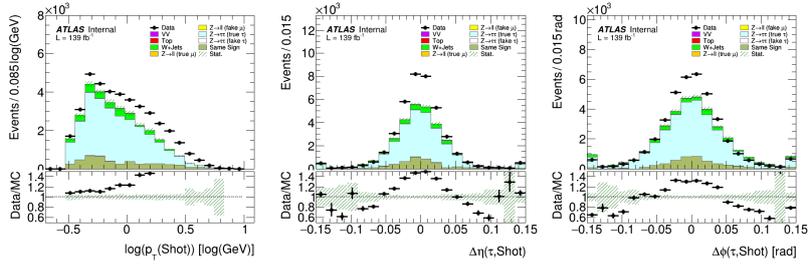


Figure B.34: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Photon Shot Variables.

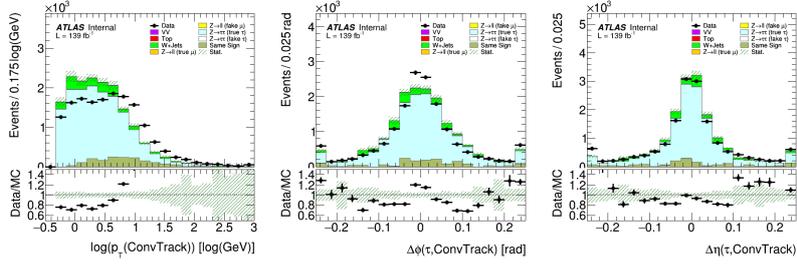


Figure B.35: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Conversion Track Variables.

Appendix B Validation on LHC Data: Additional Figures

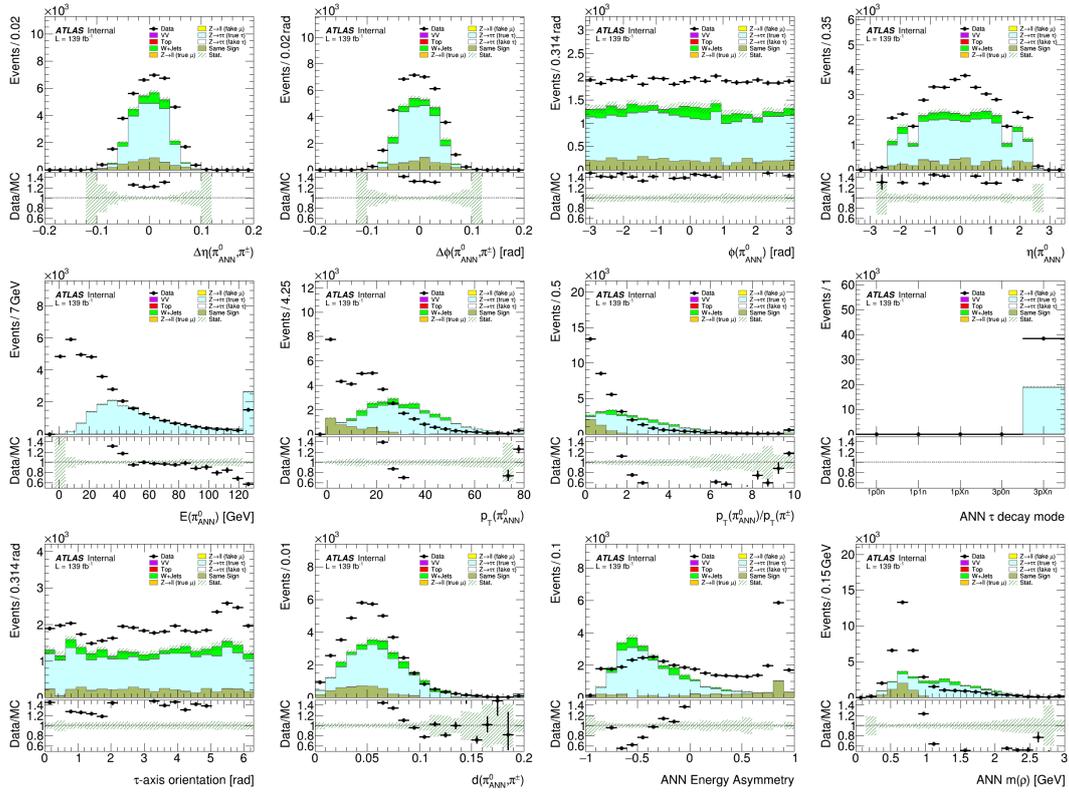


Figure B.36: Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Neural Network Output Variables.

Full Workflow Chart of the PanTau Algorithm

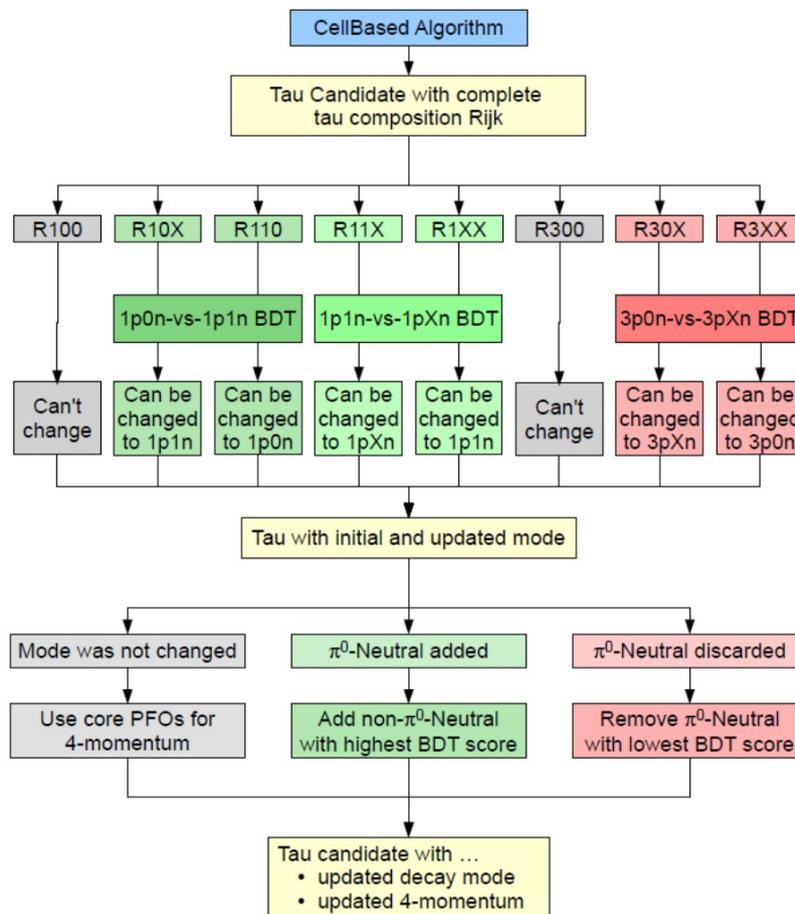


Figure C.1: Full workflow diagram of the PanTau algorithm [34].

Bibliography

- [1] S. Kottwitz, *TEXample*, 2023,
URL: <https://texample.net/tikz/examples/model-physics/> (cit. on p. 1).
- [2] Particle Data Group, K. Nakamura et al., *Review of Particle Physics*, *J. Phys. G* **37** (2022) 075021,
URL: <http://pdg.lbl.gov> (cit. on pp. 2, 3, 28, 29, 38, 49).
- [3] M. L. Perl et al., *Evidence for Anomalous Lepton Production in $e^+ - e^-$ Annihilation*,
Phys. Rev. Lett. **35** (22 1975) 1489,
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.35.1489> (cit. on p. 2).
- [4] P. D. Group, *Review of Particle Physics*,
Progress of Theoretical and Experimental Physics **2022** (2022), ISSN: 2050-3911,
URL: <https://doi.org/10.1093/ptep/ptac097> (cit. on p. 2).
- [5] S. Weinberg,
Conceptual foundation of the unified theory of weak and electromagnetic interactions,
URL: <https://www.nobelprize.org/uploads/2018/06/weinberg-lecture.pdf>
(visited on 16/03/2023) (cit. on p. 3).
- [6] A. Salam and J. C. Ward, *Electromagnetic and weak interactions*, *Phys. Lett.* **13** (1964) 168
(cit. on p. 3).
- [7] M. Thomson, *Modern Particle Physics*, Cambridge University Press, 2013 (cit. on p. 3).
- [8] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*,
Phys. Rev. Lett. **13** (9 1964) 321,
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.321> (cit. on p. 3).
- [9] P. Higgs, *Broken symmetries, massless particles and gauge fields*, *Physics Letters* **12** (1964) 132,
ISSN: 0031-9163,
URL: <https://www.sciencedirect.com/science/article/pii/0031916364911369>
(cit. on p. 3).
- [10] P. W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*,
Phys. Rev. Lett. **13** (16 1964) 508,
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.508> (cit. on p. 3).
- [11] Maike Christina Hansen, *Studies into measuring the Higgs CP-state in $H \rightarrow \tau\tau$ decays at ATLAS*,
PhD thesis: Rheinische Friedrich-Wilhelms-Universität Bonn, 2020,
URL: <https://hdl.handle.net/20.500.11811/8456> (cit. on p. 4).

- [12] F. Zwicky, *Die Rotverschiebung von extragalaktischen Nebeln*, *Helvetica Physica Acta* **6** (1933) 110 (cit. on p. 4).
- [13] S. Berge, W. Bernreuther and S. Kirchner, *Prospects of constraining the Higgs boson's nature in the tau decay channel at the LHC*, *Physical Review D* **92** (2015), URL: <https://doi.org/10.1103/PhysRevD.92.096012> (cit. on p. 5).
- [14] The ATLAS Collaboration, *Measurement of the CP properties of Higgs boson interactions with τ -leptons with the ATLAS detector*, 2022, URL: <https://arxiv.org/abs/2212.05833> (cit. on pp. 5, 42).
- [15] CMS, *Measurement of the CP properties of the Higgs boson in its decays to τ leptons with the CMS experiment*, 2021, URL: <https://bib-pubdb1.desy.de/record/462769> (cit. on p. 5).
- [16] G. Bower, T. Pierzcha, Z. Was and M. Worek, *Measuring the Higgs boson parity using Tau \rightarrow rho nu*, *Physics Letters B* **543** (2002) 227 (cit. on pp. 5, 6).
- [17] Z. Was and M. Worek, *Transverse spin effects*, (2002), URL: <https://arxiv.org/abs/hep-ph/0202007> (cit. on p. 5).
- [18] K. Desch, A. Imhof, Z. Was and M. Worek, *Probing the CP nature of the Higgs boson at linear colliders with Tau spin correlations in the case of mixed scalar and pseudoscalar couplings*, *Physics Letters B* **579** (2004) 157, URL: <https://doi.org/10.1016/j.physletb.2003.10.074> (cit. on p. 5).
- [19] The ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, *Journal of Instrumentation* **3** (2008) S08003, URL: <https://dx.doi.org/10.1088/1748-0221/3/08/S08003> (cit. on pp. 7, 9, 10).
- [20] I. Neutelings, *CMS coordinate system*, URL: https://tikz.net/axis3d_cms/ (visited on 16/01/2023) (cit. on p. 8).
- [21] J. Pequeno, *Computer generated image of the whole ATLAS detector*, 2008, URL: <https://cds.cern.ch/record/1095924#> (cit. on p. 8).
- [22] S. P. Y. Yuen, *Improving the Reconstruction of Neutral Pions in Tau Decays Using the Strip Layer of the ATLAS Electromagnetic Calorimeter*, Master Thesis: University of Bonn, 2013, URL: <http://cds.cern.ch/record/2306444> (cit. on p. 10).
- [23] The ATLAS Collaboration, *Performance of the ATLAS trigger system in 2015*, *The European Physical Journal C* **77** (2017), URL: <https://doi.org/10.1140/epjrc/article/10052-017-4852-3> (cit. on p. 11).
- [24] W. Lampl et al., *Calorimeter Clustering Algorithms: Description and Performance*, tech. rep., All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-LARG-PUB-2008-002>: CERN, 2008, URL: <https://cds.cern.ch/record/1099735> (cit. on p. 11).
- [25] M. Cacciari, G. P. Salam and G. Soyez, *The anti- k jet clustering algorithm*, *Journal of High Energy Physics* (2008) (cit. on p. 11).

- [26] The ATLAS Collaboration, *Muon reconstruction performance of the ATLAS detector in proton-proton collision data at 13TeV*, *The European Physical Journal C* **76** (2016) (cit. on p. 11).
- [27] The ATLAS Collaboration, *Performance of missing transverse momentum reconstruction with the ATLAS detector using proton-proton collisions at 13 TeV*, *The European Physical Journal C* (2018) (cit. on p. 12).
- [28] E. Bothmann et al., *Event generation with Sherpa 2.2*, *SciPost Physics* **7** (2019), URL: <https://doi.org/10.21468/SciPostPhys.7.3.034> (cit. on p. 12).
- [29] C. Oleari, *The POWHEG BOX*, *Nuclear Physics B - Proceedings Supplements* **205-206** (2010) 36, URL: <https://doi.org/10.1016/2Fj.nuclphysbps.2010.08.016> (cit. on p. 12).
- [30] T. Sjöstrand et al., *An introduction to PYTHIA 8.2*, *Computer Physics Communications* **191** (2015) 159, URL: <https://doi.org/10.1016/2Fj.cpc.2015.01.024> (cit. on p. 12).
- [31] S. Habib et al., *ASCR/HEP Exascale Requirements Review Report*, 2016, arXiv: [1603.09303](https://arxiv.org/abs/1603.09303) [physics.comp-ph] (cit. on p. 12).
- [32] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman and D. Shih, *Machine Learning in the Search for New Fundamental Physics*, 2021, arXiv: [2112.03769](https://arxiv.org/abs/2112.03769) [hep-ph] (cit. on p. 13).
- [33] C. Deutsch, *Identification and Classification of Hadronic Tau Lepton Decays in the ATLAS Experiment*, MA thesis: Bonn U., 2017 (cit. on p. 13).
- [34] H. Nguyen, *Improving hadronic tau decay mode reconstruction at ATLAS using neural networks*, Master Thesis: University of Bonn, 2020, URL: <https://web.physik.uni-bonn.de/group/view.php?&group=1&lang=en&c=t&id=123> (cit. on pp. 13, 22, 30, 32, 33, 136).
- [35] A. Jung, *Machine Learning: The Basics*, Springer Singapore, 2022, URL: <https://doi.org/10.1007/978-981-16-8193-6> (cit. on pp. 13, 21, 32).
- [36] L. Estève et al., *INRIA/scikit-learn-mooc: Third MOOC session*, version session-3, 2022, URL: <https://doi.org/10.5281/zenodo.7220307> (cit. on p. 13).
- [37] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, <http://www.deeplearningbook.org>, MIT Press, 2016 (cit. on p. 13).
- [38] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2017, arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG], URL: <https://arxiv.org/abs/1412.6980> (cit. on p. 15).
- [39] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning in Computational Mechanics*, <https://link.springer.com/book/10.1007/978-3-030-76587-3>, MIT Press, 2022 (cit. on pp. 16, 17).
- [40] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015, URL: <https://arxiv.org/abs/1502.03167> (cit. on p. 17).
- [41] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, *Nature* **521** (2015) 436 (cit. on p. 18).

- [42] F. M. Salem, *Recurrent Neural Networks*, Springer Cham, 2022, URL: <https://doi.org/10.1007/978-3-030-89929-5> (cit. on p. 17).
- [43] M. Zaheer et al., *Deep Sets*, CoRR **1703.06114** (2017), arXiv: [1703.06114](https://arxiv.org/abs/1703.06114) (cit. on pp. 18, 42).
- [44] B. P. Roe, *Probability and Statistics in the Physical Sciences*, Springer Cham, 2020, URL: <https://doi.org/10.1007/978-3-030-53694-7> (cit. on p. 19).
- [45] T. Agrawal, *Hyperparameter Optimization in Machine Learning*, Imprint: Apress, 2021, URL: <https://doi.org/10.1007/978-1-4842-6579-6> (cit. on p. 19).
- [46] The ATLAS Collaboration, *Reconstruction of hadronic decay products of tau leptons with the ATLAS experiment.*, *Eur. Phys. J. C* **76** (2016) 295, arXiv: [1512.05955](https://arxiv.org/abs/1512.05955), URL: <https://cds.cern.ch/record/2116436> (cit. on p. 20).
- [47] J. Mahlstedt, *The ATLAS Hadronic Tau Trigger*, *Journal of Physics: Conference Series* **513** (2014) 012021 (cit. on p. 20).
- [48] *Identification of hadronic tau lepton decays using neural networks in the ATLAS experiment*, tech. rep., CERN, 2019, URL: <https://cds.cern.ch/record/2688062> (cit. on p. 20).
- [49] C. Deutsch, *Identification and Classification of Hadronic Tau Lepton Decays in the ATLAS Experiment*, Master Thesis: University of Bonn, 2017, URL: <https://inspirehep.net/literature/1658715> (cit. on p. 20).
- [50] B. T. Winter, *Reconstruction of neutral pions in hadronic tau lepton decays in the ATLAS detector*, Presented on Jan 2014, 2013, URL: <http://cds.cern.ch/record/2318236> (cit. on p. 21).
- [51] S. P. Y. Yuen, *Analysis of the Higgs boson decay in the $H \rightarrow \tau_{had}\tau_{had}$ channel and CP properties with 13 TeV collisions at the ATLAS detector*, PhD Thesis: University of Bonn, 2019, URL: <https://bonndoc.ulb.uni-bonn.de/xmlui/handle/20.500.11811/8109?locale-attribute=de> (cit. on p. 21).
- [52] *Reconstruction, Identification, and Calibration of hadronically decaying tau leptons with the ATLAS detector for the LHC Run 3 and reprocessed Run 2 data*, (2022), URL: <https://inspirehep.net/literature/2153975> (cit. on p. 21).
- [53] *Measurement of the tau lepton reconstruction and identification performance in the ATLAS experiment using pp collisions at 13 TeV*, tech. rep., CERN, 2017, URL: <http://cds.cern.ch/record/2261772> (cit. on pp. 22, 52).
- [54] S. Kim, M. P. Fay and M. A. Proschan, “A Practical Guide to Exact Confidence Intervals for a Distribution of Current Status Data Using the Binomial Approach”, *Emerging Topics in Modeling Interval-Censored Survival Data*, ed. by J. Sun and D.-G. Chen, Springer International Publishing, 2022 51, ISBN: 978-3-031-12366-5, URL: https://doi.org/10.1007/978-3-031-12366-5_4 (cit. on pp. 35, 36).
- [55] Y. Kinoshita, *Identifikation von Neutralen Pion - Teilchen mit einem Neuronalen Netzwerk*, Bachelor Thesis: University of Bonn, 2022, URL: <https://web.physik.uni-bonn.de/group/view.php?&group=1&lang=de&c=t&id=140> (cit. on p. 38).

Bibliography

- [56] A. Gramfort et al., *scikit-learn QuantileTransformer*,
URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html> (visited on 05/04/2023) (cit. on p. 39).
- [57] H. Qu and L. Gouskos, *Jet tagging via particle clouds*, *Physical Review D* **101** (2020),
URL: <https://doi.org/10.1103/PhysRevD.101.056019> (cit. on p. 60).
- [58] J. Pampel, *MLPions: Tau Decay Mode Classification Using CNNs*,
TAU CP workshop, CERN, 2023 (cit. on p. 60).

List of Figures

1.1	Overview of the Standard Model.	1
1.2	Feynman diagram of a τ lepton decay.	2
1.3	Sketch of the Higgs potential.	4
1.4	Sketch of the τ decay planes.	5
1.5	Dependence of the differential cross-section on φ_{CP}^*	6
2.1	Sketch of the LHC and the ATLAS coordinate system.	8
2.2	Sketch of the ATLAS detector.	8
2.3	Sketch of the ATLAS Inner Detector.	9
2.4	Sketch of the calorimeter system of the ATLAS detector.	9
2.5	Sketch of the ECAL.	10
2.6	With Sherpa generated event.	12
3.1	Graphical Display of a general machine learning workflow.	13
3.2	Graphical Display of different activation functions.	16
3.3	Graphical Display of a feed forward neural network.	17
3.4	Graphical Display of an RNN node.	18
4.1	Illustrations of the difference between a jet and τ_{vis}	20
4.2	$1p1n$ η - ϕ -display	21
4.3	Migration matrix for the PanTau algorithm on a $\gamma^* \rightarrow \tau\tau$ sample	22
4.4	PanTau algorithm residual distributions for different intervals of $p_T(\tau_{\text{vis,true}})$ of $E(\pi^0)$	23
4.5	Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$	24
4.6	Bias and resolution of the $E(\pi^0)$ prediction for the PanTau algorithm.	24
4.7	Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $E(\pi_{\text{true}}^0)$ with a cut on $d(\pi^0, \pi^\pm)$	25
4.8	Display of the quantiles of the $\alpha(\pi^0, \pi^\pm)$ residual distribution for the PanTau algorithm against $d(\pi^0, \pi^\pm)$ with a cut on $E(\pi_{\text{true}}^0)$	25
4.9	Display of the quantiles of the Y residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$	26
4.10	Display of the quantiles of the $\eta(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$	26

List of Figures

4.11	Display of the quantiles of the $\phi(\pi^0)$ residual distribution for the PanTau algorithm against $p_T(\tau_{\text{vis,true}})$	27
4.12	Display of the quantiles of the $d(\pi^0, \pi^\pm)$ residual distribution for the PanTau algorithm against $d(\pi^0, \pi^\pm)$ for two different ranges on the y-axis.	27
4.13	Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against $\eta(\pi_{\text{true}}^0)$ and $\phi(\pi_{\text{true}}^0)$	28
4.14	Display of the quantiles of the $E(\pi^0)$ residual distribution for the PanTau algorithm against pile-up.	28
4.15	Display of the ρ meson mass distribution from the 1p1n decay for PanTau and truth data. The Particle Data Group (PDG) average is depicted as well [2].	29
5.1	Architecture of the used RNN for decay mode reconstruction.	32
5.2	Migration matrix of the decay mode classification with the RNN setup from H. Nguyen on a $\gamma^* \rightarrow \tau\tau$ sample.	33
5.3	Migration matrix of the decay mode classification with the RNN on a $Z \rightarrow \tau\tau$ sample.	34
5.4	Distributions of different observables in the $Z \rightarrow \tau\tau$ and $\gamma^* \rightarrow \tau\tau$ sample. The error bars denote the Poisson error.	34
5.5	Dependence of the classification efficiency for different decay modes on $p_T(\tau_{\text{vis,true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].	35
5.6	Dependence of the classification efficiency for different decay modes on $\eta(\tau_{\text{vis,true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].	35
5.7	Dependence of the classification efficiency for different decay modes on $\phi(\tau_{\text{vis,true}})$. The error bars denote the 99% Clopper-Pearson intervals [54].	36
5.8	Dependence of the classification efficiency for different decay modes on the average number of interactions per bunch crossing. The error bars denote the 99% Clopper-Pearson intervals [54].	36
5.9	Migration matrix of the decay mode classification with the RNN on a $Z \rightarrow \tau\tau$ sample with a post-processing correction using the number of tracks.	37
6.1	Architecture of the regression RNN.	39
6.2	Energy resolution and bias for the RNN and PanTau.	40
6.3	$\alpha(\pi^0, \pi^\pm)$ resolution and bias for the RNN and PanTau.	40
6.4	Distribution of the targets of the neural network.	41
6.5	Distribution of the targets of the neural network after applying the quantile transformation.	41
6.6	Energy resolution for PanTau and the RNN with improved and basic setup in two bins. Improved setup means the addition of batch normalization and a target Gaussian transformation. The error bars denote the average of three training runs with different Tensorflow seeds.	41
6.7	$\alpha(\pi^0, \pi^\pm)$ resolution and bias for PanTau and the RNN with improved setup.	42
6.8	DeepSet neural network architecture.	43
6.9	Energy resolution for the DeepSet NN.	43
6.10	$\alpha(\pi^0, \pi^\pm)$ resolution and bias for the DeepSet NN.	43
6.11	Validation and training loss against training time for the RNN and the DeepSet NN.	44
6.12	Performance of the ANN for a large weight of the angles in the loss function.	44

List of Figures

6.13	Performance of the ANN for a large weight of the transverse momentum in the loss function.	45
6.14	Learning curves for different configurations of batch size and learning rate. The x-axis is displayed in units of time instead of number of epochs	47
6.15	Minimum validation loss against call of the Bayesian hyperparameter optimization.	48
6.16	Energy resolution for the ANN with and without optimized hyperparameters.	48
6.17	$\alpha(\pi^0, \pi^\pm)$ resolution and bias for the ANN with and without optimized hyperparameters.	48
6.18	Learning curve for call one of the hyperparameter optimization.	49
6.19	Display of the ρ mass distribution for PanTau, the optimized neural network and truth data.	49
6.20	2D residual distribution plots PanTau and the optimized neural network.	50
6.21	$\eta(\pi^0)$ residual distribution for PanTau and the neural network with a logarithmic y-axis.	50
7.1	Comparison plots between data and simulation for general observables. All decay modes included.	53
7.2	Comparison plots between data and simulation for Neutral PFO variables. All decay modes included.	54
7.3	Comparison plots between data and simulation for Charged PFO variables. All decay modes included.	55
7.4	Comparison plots between data and simulation for the Photon Shot variables. All decay modes included.	55
7.5	Comparison plots between data and simulation for the Conversion Track variables. All decay modes included.	56
7.6	Comparison plots between data and simulation of the decay mode classification ANN. All decay modes included.	56
7.7	Comparison plots between data and simulation of variables predicted by the ANN. All decay modes included.	57
7.8	Comparison plots between data and simulation of observables derived from the ANN prediction for the π^0 kinematics. All decay modes included.	58
7.9	Comparison plots between data and simulation for additional observables for crosschecking. Exclusively events with decay mode 1p0n predicted by the ANN.	58
7.10	Comparison plots between data and simulation for additional observables for crosschecking. All decay modes selected.	59
7.11	Comparison plots between data and simulation for different observables. Exclusively events with decay mode 1p0n predicted by the ANN.	59
A.1	$\Delta E(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis, true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	62
A.2	$\Delta E(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $d(\pi^0, \pi^\pm)_{\text{true}} \geq 0.05$ is applied.	63

List of Figures

A.3 $\Delta E(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $d(\pi^0, \pi^\pm)_{\text{true}} < 0.05$ is applied.	64
A.4 $\Delta E(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	65
A.5 $\Delta E(\pi^0)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	66
A.6 $\Delta E(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	67
A.7 $\Delta E(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	68
A.8 $\Delta\phi(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	70
A.9 $\Delta\phi(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	71
A.10 $\Delta\phi(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	72
A.11 $\Delta\phi(\pi^0)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	73
A.12 $\Delta\phi(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	74
A.13 $\Delta\phi(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	75
A.14 $\Delta\eta(\pi^0)$ plots for different intervals of $p_T(\tau_{\text{vis,true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	77
A.15 $\Delta\eta(\pi^0)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	78
A.16 $\Delta\eta(\pi^0)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	79

A.17 $\Delta\eta(\pi^0)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	80
A.18 $\Delta\eta(\pi^0)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	81
A.19 $\Delta\eta(\pi^0)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	82
A.20 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $p_T(\tau_{\text{vis, true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	84
A.21 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	85
A.22 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	86
A.23 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	87
A.24 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	88
A.25 $\Delta d(\pi^0, \pi^\pm)$ plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	89
A.26 ΔY plots for different intervals of $p_T(\tau_{\text{vis, true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	91
A.27 ΔY plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	92
A.28 ΔY plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	93
A.29 ΔY plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	94

List of Figures

A.30	ΔY plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	95
A.31	ΔY plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	96
A.32	α plots for different intervals of $p_T(\tau_{\text{vis, true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	98
A.33	α plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	99
A.34	α plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	100
A.35	α plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	101
A.36	α plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	102
A.37	α plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $E(\pi_{\text{true}}^0) \geq 25\text{GeV}$ is applied.	103
A.38	α plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution. The cut $E(\pi_{\text{true}}^0) < 25\text{GeV}$ is applied.	104
A.39	ρ mass distribution plot on truth level as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters.	105
A.40	$\Delta\rho$ mass plots for different intervals of $p_T(\tau_{\text{vis, true}})$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	106
A.41	$\Delta\rho$ mass plots for different intervals of $E(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	107
A.42	$\Delta\rho$ mass plots for different intervals of $\eta(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	108
A.43	$\Delta\rho$ mass plots for different intervals of $\phi(\pi_{\text{true}}^0)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	109

List of Figures

A.44	$\Delta\rho$ mass plots for different intervals of Pile-up for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	110
A.45	$\Delta\rho$ mass plots for different intervals of $d(\pi^0, \pi^\pm)$ for the PanTau algorithm as well as for the PanTau algorithm and the DeepSet NN with optimized hyperparameters: Residual distributions and a summary of their quantiles as well as displays of bias and resolution.	111
A.46	Learning curves for calls 1 through 20 of the hyperparameter optimization.	112
A.47	Learning curves for calls 21 through 40 of the hyperparameter optimization.	113
A.48	Learning curves for calls 41 through 60 of the hyperparameter optimization.	114
A.49	Learning curves for calls 61 through 80 of the hyperparameter optimization.	115
A.50	Learning curves for calls 81 through 100 of the hyperparameter optimization.	116
B.1	Data Monte Carlo Comparison plots for all decay modes for General Observables. . . .	118
B.2	Data Monte Carlo Comparison plots for all decay modes for Neutral PFO Variables. . .	118
B.3	Data Monte Carlo Comparison plots for all decay modes for Charged PFO Variables. . .	119
B.4	Data Monte Carlo Comparison plots for all decay modes for Photon Shot Variables. . .	119
B.5	Data Monte Carlo Comparison plots for all decay modes for Conversion Track Variables. .	119
B.6	Data Monte Carlo Comparison plots for all decay modes for Neural Network Output Variables.	120
B.7	Data Monte Carlo Comparison plots for the decay mode 1p0n for General Observables. .	121
B.8	Data Monte Carlo Comparison plots for the decay mode 1p0n for Neutral PFO Variables. .	121
B.9	Data Monte Carlo Comparison plots for the decay mode 1p0n for Charged PFO Variables. .	122
B.10	Data Monte Carlo Comparison plots for the decay mode 1p0n for Photon Shot Variables. .	122
B.11	Data Monte Carlo Comparison plots for the decay mode 1p0n for Conversion Track Variables.	122
B.12	Data Monte Carlo Comparison plots for the decay mode 1p0n for Neural Network Output Variables.	123
B.13	Data Monte Carlo Comparison plots for the decay mode 1p1n for General Observables. . .	124
B.14	Data Monte Carlo Comparison plots for the decay mode 1p1n for Neutral PFO Variables. .	124
B.15	Data Monte Carlo Comparison plots for the decay mode 1p1n for Charged PFO Variables. .	125
B.16	Data Monte Carlo Comparison plots for the decay mode 1p1n for Photon Shot Variables. .	125
B.17	Data Monte Carlo Comparison plots for the decay mode 1p1n for Conversion Track Variables.	125
B.18	Data Monte Carlo Comparison plots for the decay mode 1p1n for Neural Network Output Variables.	126
B.19	Data Monte Carlo Comparison plots for the decay mode 1pXn for General Observables. . .	127
B.20	Data Monte Carlo Comparison plots for the decay mode 1pXn for Neutral PFO Variables. .	127
B.21	Data Monte Carlo Comparison plots for the decay mode 1pXn for Charged PFO Variables. .	128
B.22	Data Monte Carlo Comparison plots for the decay mode 1pXn for Photon Shot Variables. .	128
B.23	Data Monte Carlo Comparison plots for the decay mode 1pXn for Conversion Track Variables.	128
B.24	Data Monte Carlo Comparison plots for the decay mode 1pXn for Neural Network Output Variables.	129
B.25	Data Monte Carlo Comparison plots for the decay mode 3p0n for General Observables. . .	130
B.26	Data Monte Carlo Comparison plots for the decay mode 3p0n for Neutral PFO Variables. .	130

List of Figures

B.27	Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Charged PFO Variables.	131
B.28	Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Photon Shot Variables.	131
B.29	Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Conversion Track Variables.	131
B.30	Data Monte Carlo Comparison plots for the decay mode $3p0n$ for Neural Network Output Variables.	132
B.31	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for General Observables.	133
B.32	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Neutral PFO Variables.	133
B.33	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Charged PFO Variables.	134
B.34	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Photon Shot Variables.	134
B.35	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Conversion Track Variables.	134
B.36	Data Monte Carlo Comparison plots for the decay mode $3pXn$ for Neural Network Output Variables.	135
C.1	Full PanTau workflow diagram	136

List of Tables

1.1	Notation for the dominant hadronic τ lepton decay modes.	2
5.1	RNN input features	31
5.2	Cuts applied on the training data.	31
6.1	Parameters and choices considered during the training parameter grid search.	46
6.2	Architecture parameters on the corresponding ranges considered in the Bayesian optimization.	46
6.3	Best performing configuration of architecture parameters received from the Bayesian optimization.	47
7.1	Cuts applied in the comparison of data and simulation. Consider [53] for further explanations.	52

Acknowledgements

I would like to express my gratitude to Prof. Dr. Klaus Desch for the opportunity to work on this interesting topic in his group and for providing me the experience of scientific work within the ATLAS collaboration. I would also like to thank Prof. Dr. Ian Brock for being the second referee on this thesis. To Priv. Doz. Dr. Philip Bechtle and Dr. Christian Grefe I am thankful for all advice on physics-related challenges as well as helpful advice concerning scientific research. Their availability in daily meetings enabled quick problem solving and greatly added to progress throughout this project. Let me thank Lena Herrmann, Jan-Eric Heinrichs, Simon Thiele and Marco Menen for test reading this thesis and offering advice for improvements. Moreover, thanks to all members of the working group for the support during the last year. Very helpful was also advice from Oliver Freyermuth. Thank you for the quick and patient advice on all IT-related questions. I would like to extend my gratitude to my friends for all the encouragement. Lastly, I would like to thank my family for their endless support and advice during my entire studies.