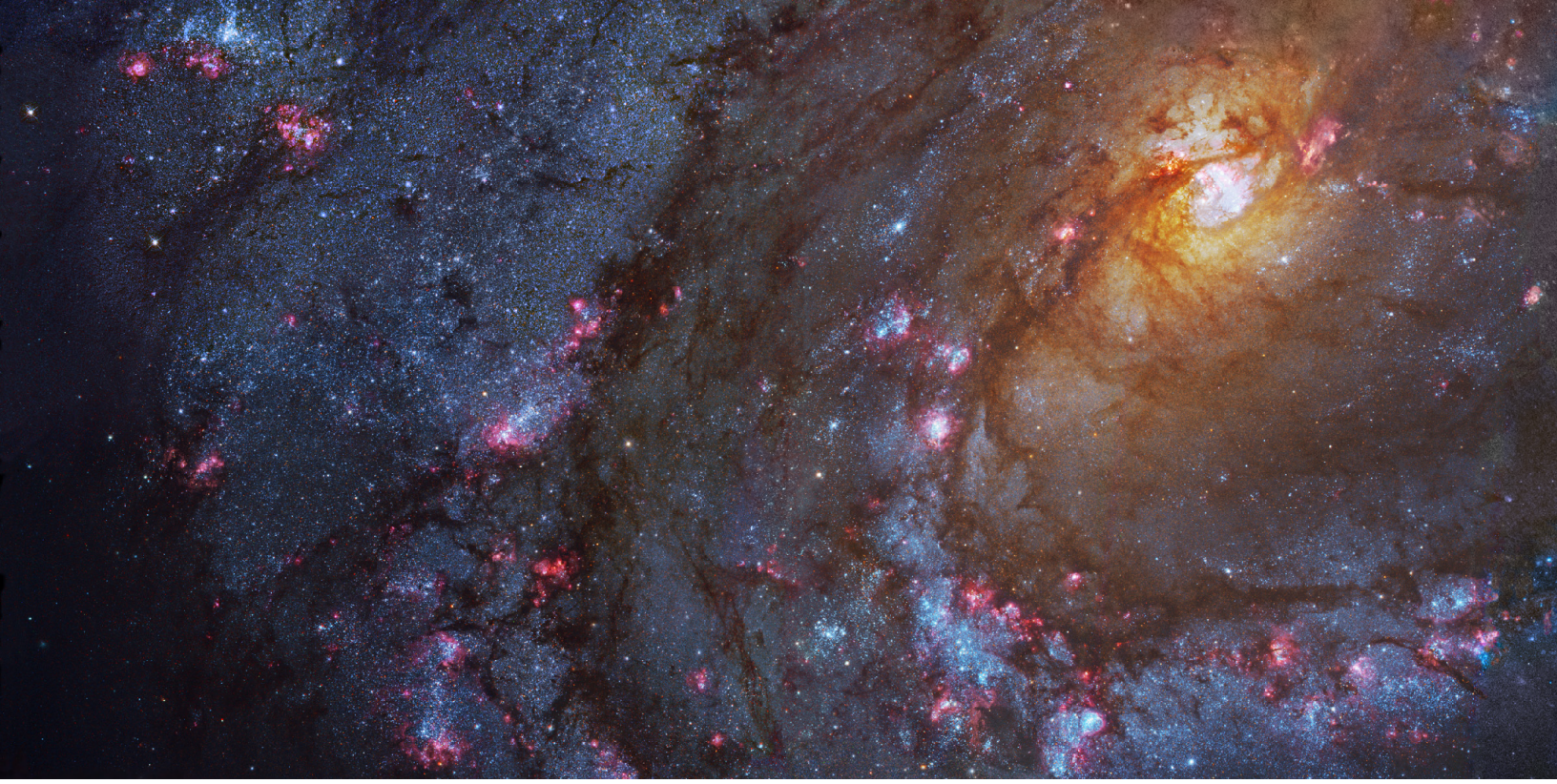

4.1 Introduction

The first Large Size Telescope (LST) was inaugurated in October 2018 and is currently taking its first data. It is the first Cherenkov Telescope Array (CTA) telescope installed on site (in La Palma island), and it is expected to be operating on its own until LST 2-4 are built. This mean that LST1 will need its own analysis chain in *mono* mode, which differs in several aspects from the stereo analysis included in the benchmark analysis tools of CTA (see section 3.4.2). Although single telescope observations present a big challenge, specially regarding source position reconstruction and γ-hadron separation, it is expected that LST1 performance, thanks to its size and camera design, will be competitive enough to offer scientific results in the time it will be operating alone.
This chapter will present a considerable amount of the work done during this thesis, which includes the development of the code for the single telescope analysis for LST1, the calculation of LST1 sensitivity based on Monte Carlo (MC) simulations, the development of a new technique for Hillas Parameters calculation without cleaning using the Expectation-Maximization algorithm and the application of the analysis chain to real LST1 data.

4.2 The LST1 analysis chain overview

The software for the single telescope analysis of LST1, named *cta-lstchain*, has been developed under the necessity of specific tools for single telescope analysis, which are not included in *ctapipe*. It has been written as a python package which heavily relies on *ctapipe*, and it is structured in several modules containing functions destined to the different parts of the analysis. The first version of *cta-lstchain* was written by the

author of this thesis and many contributors have joined the project over the last two years to improve and optimize the repository to its current version, which is able to perform every analysis step both for simulated MC data and real data. The analysis chain is divided in several steps, each of which can be executed through a python script which requires certain inputs and calls for the appropriate functions. All the configuration parameters of the different elements of the analysis are given through configuration files, which can be edited by the user or else a standard configuration will be used. The input of the analysis chain are the raw data files of LST1 events, which for MC data are the output files from *sim_telarray* (see section 3.4.1), and for real data are *zfits* files with a similar internal format. The files contain the full information available per pixel, known as *waveform* and which is the digitized signal amplitude vs. time sample for every triggered event, together with MC information in the case of a simulated file (such as the true energy, source position, number of simulated events, and so on), or recorded information from the different telescope subsystems in the case of real data (such as time, pointing, trigger type...). Throughout the analysis chain, the data including the images and image parameters, is stored in containers designed in *cta-lstchain* specifically for LST1, which can be dumped into *pandas* dataframes and saved in *hdf5* files.
The main steps of the analysis chain can be summarized as:

- **Calibration:** The waveforms of each pixel in the camera should be integrated after pedestal subtraction, and converted to number of photoelectrons. Also, the timing information of the signal is obtained.
- **Image cleaning and parameterization:** The images in the camera contain pixels with light not related to the Cherenkov event, so a cleaning must be applied to remove them. Afterwards, the distribution of photons in the image is used to calculate the Hillas parameters.
- **Energy and direction reconstruction:** The energy and direction reconstruction of the triggered events are performed using a multidimensional regression technique based on Random Forests (RFs). A set of simulated diffuse γ events are used to train the RF.
- **γ-hadron separation:** For the γ-hadron separation, a multidimensional RF classifier is used. Sets of simulated γ and proton events, which energies and directions have been reconstructed in the previous step, are used to train the classifier.

4.2.1 Calibration

In the calibration phase, the raw signal known as *waveform* is integrated after background subtraction, to obtain a total number of counts per pixel, which afterwards is multiplied by a factor to be converted in number of photoelectrons. Depending on if MC data or real data is being analyzed, the following steps vary and will be explained for each case.

Signal extraction

For every triggered event, the signal in each pixel is recorded in a 40 samples window from the 4096 samples of the Domino Ring Sampler version 4 (DSR4). This signal contains information not only from the Cherenkov light, but also from background light from Night Sky Background (NSB) and from the intrinsic noise induced by the readout chain. Before integrating the signal, it is necessary to subtract this baseline signal, known as *pedestal*. For simulated MC events the pedestal value for each pixel in the camera is already stored, but for real data it is necessary to take special pedestal runs, with randomly activated trigger. Pedestal events are used to calculate the mean pedestal value for each DSR4, Typically, around 1000 events are needed to fill the ring. In *cta-lstchain* a specific script is used to extract the pedestal values from pedestal runs and they are stored in a *hdf5* file to be used later in the calibration.
Once the pedestal is subtracted from the signal, the signal peak can be integrated. Typically a smaller window of a few samples around the maximum is used for the integration, which can be performed with one of the several integrators implemented in *ctapipe*. By default, the integrator used in *cta-lstchain* is the *NeighborPeakWindowSum*, which sums the signal in a window around the peak defined by the waveform in neighbouring pixels. This allow to avoid integrating peaks which can arise from fluctuations. The default width of the integration window for this integrator is of 7 samples, but it can be configure by the user. These calibration steps are performed for the two gain channels of LST1 camera.

Conversion to photoelectrons

Once the signal amplitude is extracted, it must be converted from DC counts to photoelectrons through a calibration factor, which is different for each pixel and channel. For simulated MC data, these factors are stored in the simulated file and the conversion can be done simply by multiplying the image in DC counts by the factor of each pixel. For real data, special calibration runs must be taken to calculate these factors. Calibration events are taken using Ultraviolet light pulses fired from the CaliBox [177], [178] located in the mirror dish. The calculation of the calibration coefficients is made using the F-factor method [179], which assumes that the distribution of photoelectrons in a Photomultiplier Tube (PMT) follows a poissonian statistics with a mean N and a root mean square (RMS) $\sqrt{N}$. The signal distribution, however, will be deviated from a poissonian statistics with a wider RMS due to an excess-noise factor F, which is different for each PMT and is measured in the laboratory. The relation between the relative widths of the two distributions can be written as:

$$F \cdot \frac{1}{\sqrt{N}} = \frac{\sigma_Q}{< Q >} \tag{4.1}$$

Where $< Q >$ is the mean value of the pixel signal and σ_Q its RMS. The calibration coefficients will come from the relation between the number of photoelectrons N and

the mean pixel signal $< Q >$. From 4.1, and taking into account that the signal must be corrected from pedestal, the relation between number of photoelectrons N and the pixel charge $< Q >$ is:

$$\frac{N}{< Q >} = F^2 \frac{< Q > - < ped >}{\sigma_Q^2 - \sigma_{ped}^2} \tag{4.2}$$

The values of $< Q >$, σ_Q, $< ped >$ and σ_{ped} are calculated from a sufficiently large number (~ 1000) of calibration and pedestal events, using a specific script from *cta-lstchain*, which stores the resulting calibration coefficients in an *hdf5* file to be used later in the calibration of data runs. For the LST1, the value of F factor used is the mean value for all PMT which is 1.2.

4.2.2 Image cleaning and parameterization

In order to extract information from the image of Cherenkov light recorded by the camera, it is necessary to get rid of the background light not belonging to the event, which has arrived to all pixels. This background light is usually related to fluctuations in the NSB. A process named cleaning is used to eliminate all pixels which presumably do not contain light from the shower. The *clean* image is used later to perform the Hillas parameterization. Besides there exist several proposed cleaning algorithms in the literature [180], [181], [182], [183], for the time being in *cta-lstchain*, a classical two-level tailcuts cleaning is being applied (like the one used in [184]). This method requires only the information of the amount of light in pixels (already converted to number of photoelectrons) and compares it to two levels of thresholds in the following way:

- Pixels with a number of photoelectrons over the highest threshold level Th_{high} are selected.
- If pixels selected in the previous step have at least one neighbour also above the highest level threshold, those pixels are marked as core pixels from the shower.
- Neighbours of the core pixels with number of photoelectrons above the lower level threshold Th_{low} are selected as boundary pixels. For the rest of them not selected, their charge is set to zero.

The standard values for the two levels used in *cta-lstchain* are $Th_{high} = 6$ phes and $Th_{low} = 3$ phe. Note that these values have not been yet fully optimized for the particular case of the LST1, and they can be changed easily in the configuration files. An optimization of the cleaning can lead to a better performance, specially for lower energies, because in those cases Cherenkov showers are small and a lot of information can be lost because of cleaning. Using other information apart from the number of photoelectrons, like the arrival times of the signals to pixels, can help to improve the image parameterization. Low energy showers of γs and hadrons are much more difficult to differentiate, for that reason, an image parameterization which highly

depends on the settings of the cleaning parameters can be problematic when trying to lower the energy threshold of the telescope. In section 4.4 a method for image parameterization not requiring previous cleaning is proposed as an alternative.
The parameterization of the shower image after cleaning, meaning the calculation of the Hillas parameters (see section 2.4.2 and figure 2.9), is performed by a specific function from *ctapipe*. The resulting parameters: intensity (total number of photoelectrons in the shower), width, length, coordinates of ellipse center of gravity, azimuthal angle ϕ, the orientation angle ψ, and the third order moments: *skewness*, which is a measure of the asymmetry of the distribution; and *kurtosis*, which is a measure of whether the distribution is peaked or flat relative to a normal distribution. The calculation of the Hillas parameters with *ctapipe* only requires the image of the shower (i.e. the number of photoelectrons in each pixel) and the information of the camera geometry.
The time parameters, *time gradient* and *intercept* are also calculated as explained in section 3.4.2. These two parameters are specially important for single telescope analysis, because they reflect the direction of development of the shower, which give information of the side of the ellipse where the source position is located in the camera frame.
Other parameters are the *leakage2*, which indicates the percentage of the shower that falls in the two outer pixel rings of the camera. A large leakage indicates that the shower is highly truncated. The *number of islands* is also calculated, which accounts for the number of separated groups of pixels in the image. A typical γ-ray shower will only have one island with elliptical form, but hadronic and heavier nuclei showers tend to produce messy light distributions with several islands of irregular shapes.

The calibration and image parameterization is performed in *cta-lstchain* using the scripts *lstchain_data_r0_to_dl1.py*, *lstchain_mc_r0_to_dl1.py*, depending on if the raw data used is simulated or real data. As is reflected in the script names, this steps reduces the data level from raw R0 data to DL1 (see table 3.2). In general, the image parameters can well describe the shower and therefore pixel wise information is not needed in further steps of the analysis. However, for testing and crosscheck purposes, *cta-lstchain* scripts offer the possibility to store DL1 data with the full images.

4.2.3 Reconstruction of energy and source position

After the calibration and image parameterization, the image parameters should be used to extract information of the primary particle, whereas is a γ-ray or a background event (protons, electrons, heavier nuclei...). In *cta-lstchain* the first step is to reconstruct the energy and arrival direction of the event.
The direction reconstruction is problematic in single telescope mode, because even knowing that the Hillas ellipse should point towards the source position in the camera frame, we do not know in which side of the ellipse is located (this is known as head-tail degeneracy). In stereo mode, this is solved because the source position will be in the

cross point between the line which follows the semi major axis of Hillas ellipses of all cameras, but for single telescope we must rely in other methods. In the case of *cta-lstchain* we make use of a vector known as *disp*. This is the vector going from the center of gravity of the ellipse to the source position.
Originally, the disp quantity could be parameterized in terms of the elongation of the image, meaning the ratio between *width* and *length*. The first parameterization was proposed by the Whipple collaboration [185]:

$$Disp = \xi \cdot \left(1 - \frac{width}{length}\right) \tag{4.3}$$

Where ξ is a factor dependent on the amount of photoelectrons in the shower image (the *intensity* parameter). A more general parameterization was used for MAGIC-I [186]:

$$Disp = A(intensity) + B(intensity) \cdot \frac{width}{length + \rho(intensity) \cdot leakage} \tag{4.4}$$

Where A, B and ρ are the second order polynomial function parameters of $\log(intensity)$.

Instead of using a parameterization, in *cta-lstchain* the disp vector is directly reconstructed using the image parameters. The time parameters will be very important to reconstruct this quantity, because the Cherenkov light from the upper parts of the shower arrive before the light from later developed lower parts, giving the time gradient. The time gradient will therefore give information on which side of the ellipse is pointing to the source position.

The method used in *cta-lstchain* for the reconstruction of these quantities, energy and disp vector, is based on a multidimensional regression technique relying on RFs. RF is a supervised learning algorithm which uses an ensemble of several decision trees (this is known as a *bagging* technique). Decision trees are flow-chart-like structures where each internal node denotes a test on a selected feature. The result is a split in the dataset depending on the value of that feature. The best splitting criterion for regression is typically calculated using the mean squared error (MSE). For each node, the MSE of the two subsets is calculated for each feature and for each possible cut on that feature, until a minimum is reached. This step is repeated at each node until a condition is reached, typically that the MSE of the remaining data sample is below a threshold, or it has too few events to keep splitting. The termination nodes are also known as *leaves*. In the RF, a number of decision trees are trained using a randomly selected subset of the train data (*'bootstrapping'*) and the prediction of all the trees is averaged to reach the final result. Also, the features are always randomly permuted

at each split. These two sources of randomness in the RF prevent the typical problem of over-fitting from too complicated decision trees.

In the basic analysis of *cta-lstchain*, the one carried on in this thesis, a source independent approach is used, where it is not assumed what is the true position of the source. For that reason, the RFs for energy and disp reconstruction are trained using a set of MC simulated γ events triggering the LST1, with diffuse arrival directions distributed in a 5^o radius field of view (FoV). The set of simulated data is divided in two sets, for training and testing. The features used for the splitting of the tree nodes are $log_{10}(intensity)$, *width, length, x, y, psi, phi, width/length, skewness, kurtosis, r, time gradient, intercept, leakage* and *number of islands*, where x, y and r are the coordinates and module of the vector from center of gravity of the Hillas ellipse to the center of the camera. These features can be ordered based on their importance for the regression, as shown in figure 4.1. It can be seen for example that for direction reconstruction there are two features, time gradient and angle ψ, which are the best candidates to make a good splitting, while for γ-hadron separation all features supply with a similar amount of information. The trained RFs are used later to reconstruct the desired values (energy and disp) in the test data. For energy reconstruction, the reconstructed quantity is actually $\log_{10}(E)$. In the case of the disp vector, the two components of the vector (*disp_dx, disp_dy*) are reconstructed using the same RF. For the implementation of the RFs, the Python package *scikit-learn* [187] is used, a machine learning package which offers a large amount of tools for predictive data analysis. The class *RandomForestRegressor* allow an easy training of the RFs, which can be saved to be used later to fit any set of test data with the same format as the training set. This class has a set of parameters that can be modified by the user to optimize the performance of the predictions. In *cta-lstchain* these parameters can be modified by a configuration file. The most relevant parameters and their default values in *cta-lstchain* are shown in table 4.1.

4.2.4 γ-hadron separation

The majority of Cherenkov showers produced in the atmosphere are actually not produced by γ-rays, but by cosmic hadrons (mainly protons). These events trigger the telescopes $\sim 10^4$ times more than γ-rays, therefore it is necessary a very efficient background rejection method to discard hadronic showers and analyze only γ events. The methods for γ-hadron separation in general rely on the morphological differences between the showers produced by the two kinds of particles, including temporal information. This job is much more efficient in stereo mode, because γ-ray showers which trigger different telescopes will produce images in the cameras with a Hillas ellipse pointing towards the source position, while hadronic showers will produce much less correlated images. In mono mode we can only rely on the morphological features of the images, knowing that hadronic shower images will be much more extended, without a clear definite shape and with a higher number of islands than γ showers.
The task of γ-hadron separation is done in *cta-lstchain* with a RF classifier, which follows similar principles as the RF regressor explained in the previous section. Instead

Parameter name	Value for regression	Value for classification	Description
n_estimators	150	100	Number of trees in the forest
criterion	mse	gini	Function to measure the quality of a split
max_depth	50	100	Maximum depth of the tree
min_samples_split	2	2	Minimum number of samples required to split an internal node
min_samples_leaf	2	2	Minimum number of samples to be at a leaf node
max_features	all	all	Number of features to consider when looking for the best split
random_state	42	42	Control the randomness of the bootstrapping and the sampling of features
n_jobs	4	4	Number of jobs to run in parallel

Table 4.1: Some parameters that can be configured for *RandomForestRegressor* and *RandomForestClassifier* classes, with the default values used in *cta-lstchain*.

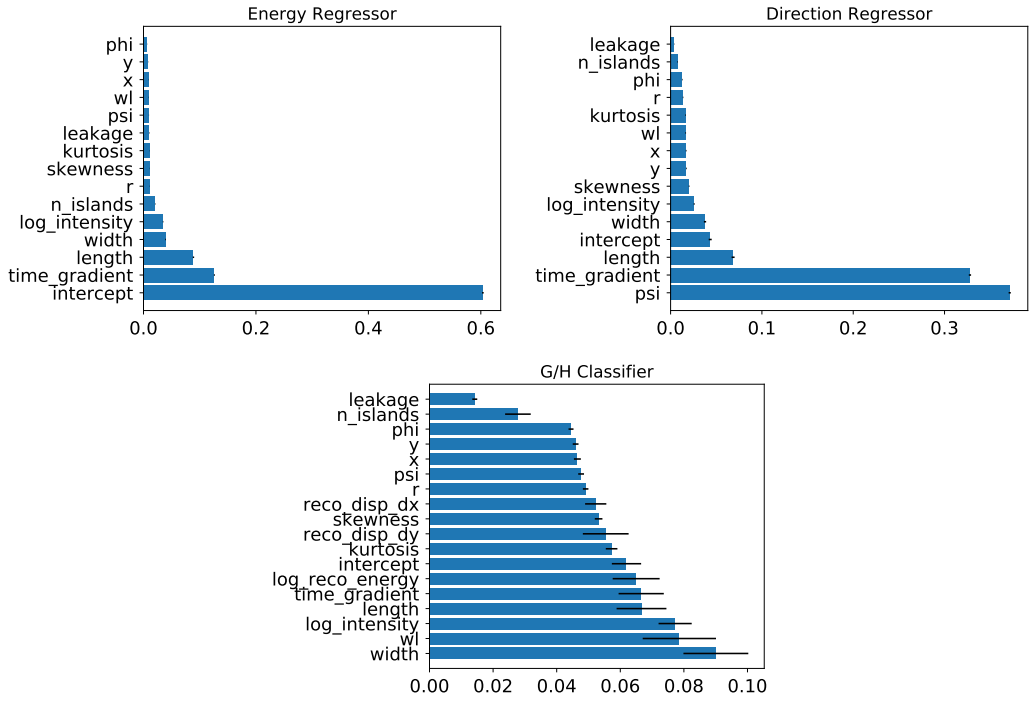

Figure 4.1: Feature importances for the trained RF used in the analysis of MC simulated LST1 data. The higher the importance value, the more relevant is the feature for the regression/classification.

of calculating the value of a variable, it decides between a given number of classes to which each events belongs. In this case, its only a two classes problem: γs or hadrons. The best splitting criterion for classification is calculated in terms of the so-called gini index. The gini index o gini impurity is a measure of how often a randomly chosen event would be incorrectly classified if it was classified randomly following the distribution of classes. The formula for the Gini index I_G calculated for a set of events with J classes is:

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2 \tag{4.5}$$

Where p_i is the probability of an event belonging to the class i to be correctly classified. The Gini index go from 0 to 1, where 0 means that all the events of a subset have been classified to one class, and 1 that the events of the subset are randomly distributed along all the possible classes. The splitting decision will be made minimizing the value of the Gini index.
The training set for the γ-hadron classifier will consist on two sets of MC simulated events, one of diffuse γs and one of protons, with arrival directions coming from a 10^o radius FoV. Usually, the set of hadronic events used for training is taken from real background events recorded with the telescope when pointing to a direction in the sky without any γ-ray source, because the simulation of proton showers has a lot of uncertainties (see section 3.4.1). However, as by the time this thesis was written there were not enough real data from LST1 to build a big enough background data set, all the calculations have been made using MC simulations.
The features used for the splitting of the classifiers are the same from the ones used for the regressors of the previous section, but adding the reconstructed energies and disp vector. To do so, the original γ events set is splitted in a training set for the regressor, and a test set, to which the energy and disp vector are reconstructed. The regressors are also used to reconstruct the energy and disp vector of the set of proton events. Afterwards, this new set of γ and proton events, with reconstructed energy and disp is divided again in a training and test set to build the RF classifier. The classifier is implemented using the *RandomForestClassifier* class from *scikit-learn* which input parameters are listed in table 4.1

4.3 Sensitivity of the LST1

The sensitivity of a telescope is defined as the minimum flux of γ-rays over background that should be detected for a statistically significant detection. Using the simulated MC data and the analysis techniques explained in section 4.2 the differential sensitivity of LST1 in mono mode can be calculated.
The sensitivity is computed for the detection of a point source after 50 hours of observations. To do so, we are using the Li&Ma method, extensively described in

[188], but here the basic concept is summarized.
The significance of detection of a source must be calculated, as a way to evaluate the statistical reliability of an observational result. A typical observation in γ-ray astronomy will consist on pointing to a region where it is suppose to exists a source, recording N_{on} counts in a time t_{on}. Then, to evaluate the background, a region without any source is observed, recording N_{off} counts in a time t_{off}. If the ratio between the observation time of the on and off regions is $\alpha = t_{on}/t_{off}$ then the number of background photons in the on region is estimated by $\hat{N}_B = \alpha N_{off}$ and the probable number of photons contributed by the source is $N_S = N_{on} - \alpha N_{off}$. The significance can be estimated in terms of the likelihood ratio method, where the we test the *null hypothesis*, being the hypothesis where no source exist at all and all the excess counts detected in the on region are due to fluctuations in the background. In this case, N_{on} will follow a Poisson distribution with variance equal to that of the background $< N_B >$. The likelihood ratio can be written as:

$$\lambda = \frac{L(X|E_0, \hat{T}_c)}{L(X|\hat{E}, \hat{T})} \tag{4.6}$$

Where X are the observed data and ($\hat{\mathrm{E}}$,$\hat{\mathrm{T}}$) are the maximum likelihood estimation of the unknown parameters. In the null hypotheses, $E = E_0$ and $T = \hat{T}_c$ are the parameters for the conditional maximum likelihood estimation. In our case, the unknown parameters will be N_S, N_B where in the null hypothesis $N_S = 0$.
The maximum likelihood ratio has the form:

$$\lambda = \left[\frac{\alpha}{1+\alpha}\left(\frac{N_{on}+N_{off}}{N_{on}}\right)\right]^{N_{on}} \left[\frac{1}{1+\alpha}\left(\frac{N_{on}+N_{off}}{N_{off}}\right)\right]^{N_{off}} \tag{4.7}$$

If the null hypothesis is true, and $N_{on}, N_{off} \gtrsim 10$, by the theorem exposed in [188], the quantity $-2ln\lambda$ will asymptotically follow a χ^2 distribution with one degree of freedom. Therefore $\sqrt{(-2ln\lambda)}$ will be equivalent to the absolute value of a standard normal variable, so we can take it as the significance:

$$S = \sqrt{(-2ln\lambda)} \tag{4.8}$$

Typically, a detection is claimed when the significance S is equal to or higher than 5 (5σ detection). To calculate the sensitivity of LST1 we need to know the number of excess counts from a hypothetical source N_S that will lead to $S = 5$ for $\alpha = 1/5$.
To perform this calculation, we need to estimate the number of background (proton) events ($N_B = \alpha N_{off}$) that will remain after analyzing the data and doing the γ-hadron separation. The steps followed to do so are described in the next sections.

4.3.1 Reweigh ting

The MC simulations of Extensive Air Showers (EAS) used for this work were simulated as if the particles (γs or protons) followed a power law spectrum :

$$\frac{dN}{dE} = KE^a \tag{4.9}$$

with a spectral index $a = -2$. However, we want to give the sensitivity with respect to the true spectrum of protons, and typically, the spectrum of the Crab nebula is used for the γ-rays. We need to transform the distribution of simulated events from number of events per energy to a rate of events (events per unit time) per energy which follow the desired spectral shape. We will calculate a spectral weight $w(E)$ for each event, which will depend on its true energy.
Suppose that N MC events have been generated in the energy range (E_1, E_2), following the power law from 4.9, with isotropically distributed directions in a solid angle Ω and impact parameters uniformly distributed in a circular area A, orthogonal to the incident direction of the particles. The quantity K will be:

$$K = \frac{N(a+1)}{E_2^{a+1} - E_1^{a+1}} \tag{4.10}$$

We want to change the shape of this spectrum, and get a new differential spectrum of the shape:

$$\frac{dF}{dE(E)} = \frac{dF}{dE(E_0)} \cdot f(E/E_0) \tag{4.11}$$

Where $\frac{dF}{dE(E_0)}$ is a normalization factor referring to an arbitrary energy E_0, which should be between E_1 and E_2, with units $s^{-1}sr^{-1}m^{-2}TeV^{-1}$ and f is a function that satisfies $f(E = E_0) = 1$. In our case, f will simply be the new power law. To correct the dN/dE simulated spectrum to the desired power law, we should multiply it by $(E/E_0)^{-a}$, so it will become flat, and then by $f(E/E_0)$ to get the correct form. Weighting by these two factors, the corrected number of MC events N' will be the integral:

$$N' = \int_{E_1}^{E_2} \left(\frac{E}{E_0}\right)^{-a} f(E/E_0)dE = \int_{E_1}^{E_2} KE_0^a f(E/E_0)dE \tag{4.12}$$

We need to transform the number of events to a rate (in Hz), in order to calculate the sensitivity for a certain observation time. The total rate calculated in the energy range E_1, E_2 will be:

$$R = \int_{E_2}^{E_1} \frac{dF}{dE} dE d\Omega dA \tag{4.13}$$

Therefore, the final weight $w(E)$ for which the spectrum of each event should me multiplied is:

$$w(E) = \left(\frac{E}{E_0}\right)^{-a} \cdot f(E/E_0) \cdot \frac{N'}{R} \tag{4.14}$$

Where $a = -2$ is the spectral index of the simulated events, E_0 is taken as 1 TeV and $f(E/E_0)$ would depend on the spectral shape to be reproduced. For γ events we take the spectrum of the Crab nebula measure by HEGRA [189]:

$$\left(\frac{dF}{dE}\right)_{Crab} = 2.83 \cdot 10^{-14} \left(\frac{E}{1TeV}\right)^{-2.62} GeV^{-1}cm^{-2}s^{-1} \tag{4.15}$$

And for protons, we use the results from the BESS spectrometer [190]:

$$\left(\frac{dF}{dE}\right)_{Crab} = 9.6 \cdot 10^{-9} \left(\frac{E}{1TeV}\right)^{-2.7} GeV^{-1}cm^{-2}s^{-1} \tag{4.16}$$

The values of the differential sensitivity will be given for a certain number of energy bins, therefore, we must calculate the number of weighted events in each energy bin and multiply them by the observation time. This will give us the quantities N_S and N_B. To calculate the sensitivity we only need N_B, which will be the number of weighted proton events divided by the factor $\alpha = 1/5$.

4.3.2 Cut optimization

To obtain the best possible sensitivity, instead of using all the weighted proton events, we can use the two parameters *gammaness* and θ^2 to perform cuts that will reject the majority of the background.
Gammaness is a number between 0 and 1 assigned by the RF classifier to make the decision on the γ-hadron separation. Events close to 1 will be more γ-like, and events close to 0 will be more proton-like. θ^2 is the angle between the reconstructed direction of the event and the true assumed direction of the source. Events with a high θ^2 are most likely to be proton events, while γs will have a θ^2 close to 0. For protons,
To optimize the cuts in these parameters, we define several bins in *gammaness* and θ^2, calculate the number of weighted proton events left after the cuts in each bin and then use this quantity as N_B for the sensitivity calculation. For each energy bin, we will select the combination of cuts which provides the best sensitivity. We require that after the cuts, at least 10 events of γs and protons remain in the energy bin.

4.3.3 Expected LST1 Performance

The analysis methods described before were used to compute the expected performance of the LST1 in mono mode for the observation of a point source, using a source

	γ (PS) offset = 0.4º	γ diffuse	electron	proton
Energy Range	5 GeV - 50 TeV	5 GeV - 50 TeV	5 GeV - 5 TeV	10 GeV - 100 TeV
Viewcone	0 º	10º	10º	15 º
Core Range	1000 m	1000	1000 m	2500
Input Events	South: $3 \cdot 10^7$	South: $5 \cdot 10^8$	South: $6 \cdot 10^8$	South: $5 \cdot 10^9$
	North: $3 \cdot 10^7$	North: $5 \cdot 10^8$	North: $6 \cdot 10^8$	North: $5 \cdot 10^9$
Triggered Events	South: $1.08 \cdot 10^6$	South: $1.22 \cdot 10^6$	South: $1.18 \cdot 10^6$	South: $8.25 \cdot 10^5$
	South: $9.60 \cdot 10^5$	South: $1.11 \cdot 10^6$	South: $1.04 \cdot 10^6$	South: $8.10 \cdot 10^5$

Table 4.2: Summary of the MC production dedicated to the LST1

independent analysis (meaning no assumption on the source position is made). In this section, the results on energy and angular resolution, γ-hadron separation and sensitivity are discussed.

Data

The data used for the analysis is a set of MC simulations produced specifically for the LST1 in the northern CTA site. The primaries produced were γs as a point source with an offset of 0.4º from the center of the camera, diffuse γs, protons and electrons, but for the results given in this thesis, electrons were not included. The injection of particles were splitted in North and South, but only the South production has been used here. The zenith angle assumed was 20º, the spectral index of the particle spectra was -2. A summary of the main features of the production is given in table 4.2.

After calibrating and parameterizing the data, using three sets of cleaning parameters to study the effect on cleaning in the analysis, the diffuse γ events were used to train the RF regressors. Then, a subset of diffuse γs and the hadron events to which the energy and direction was reconstructed, were used to train the RF classifier for γ-hadron separation. The point source γs, together with a subset of hadron events as background, have been used to produce the performance results. A filter in intensity $>$ 300 phe, leakage < 0.2 and gammaness > 0.5 was applied to the reconstructed events.

Energy resolution

The energy resolution give information on the error committed while reconstructing the energy of an event. It is defined as the 68th percentile of the relative error $\Delta E/E = (E_{reco} - E_{true})/E_{true}$. If the relative error follows a normal distribution, the 68th percentile is equivalent to one σ. The results on energy resolution are shown in figure 4.2. The features used for the energy regression ordered by their importance can be sen in figure 4.1.

Angular resolution

The angular resolution is calculated in a similar way as the energy resolution, but in this case the relative error shown in figure 4.3 refers to the angular difference between

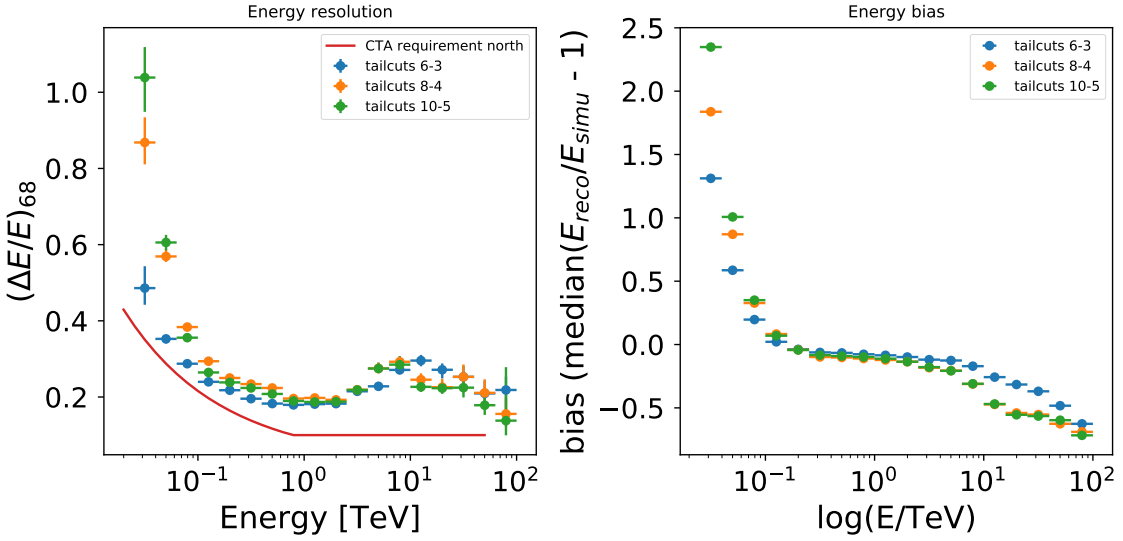

Figure 4.2: Energy resolution plots for the LST1 analysis applied to a MC production of point-like γ events, for three different sets of cleaning parameters (low level-high level threshold in number of photoelectrons). As a reference, the energy resolution requirement for CTA-north is also shown. The energy resolution plot of the *left* has been bias corrected.

the true direction of the source and the reconstructed direction. The features used for the disp vector regression ordered by their importance can be sen in figure 4.1.

γ-hadron separation

The performance in γ-hadron separation can be studied in terms of the receiver operating characteristic (ROC) curve of the RF classifier. The ROC curve illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. It is produced plotting the true positive rate(the rate of γs correctly classified) versus the false positive rare (the rate of protons incorrectly classified as γs) at several thresholds (gammaness). A value of 1 in true positive rate, while 0 in the false positive rate would mean a perfect classification. The closest of the ROC curve to a diagonal, the more similar to an uniformly random distribution is the classification. For this result, a subset of protons were used to test the RF classifier together with the point-like γs. Result is shown in figure 4.4. The features used for the γ/hadron classification ordered by their importance can be seen in figure 4.1.

Effective Area

The effective area calculated for the detection of showers from a point source with the LST1 is shown in figure 4.5.

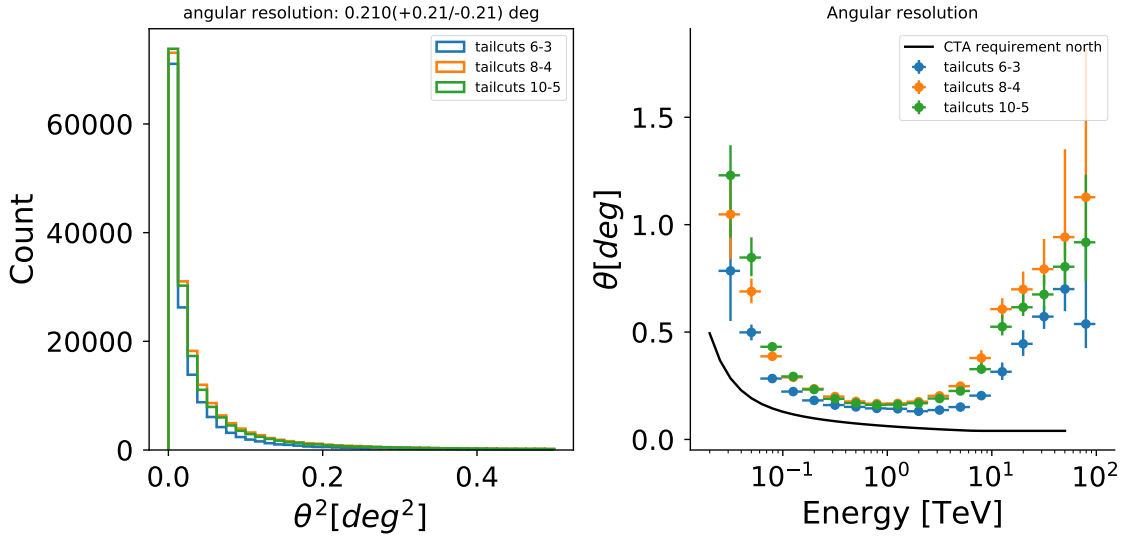

Figure 4.3: *Left:* θ^2 plot for the MC simulated γ point-like events, using three sets of cleaning parameters. *Right*: Angular resolution as a function of reconstructed energy.

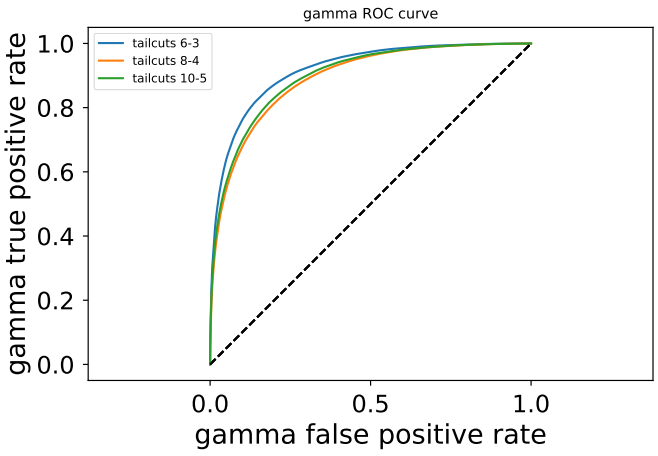

Figure 4.4: ROC curve of the RF classifier applied to point-like gamma and diffuse proton events, for three different sets of cleaning parameters.

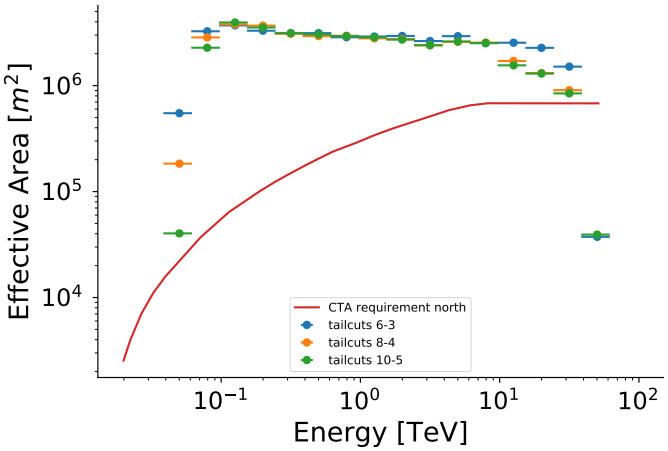

Figure 4.5: Effective area per energy bin for the MC simulated γ point-like events, using three sets of cleaning parameters.

Sensitivity

For the calculation of the sensitivity, a subset of proton events was used as background (off) events. The point-like γ events were used together with the protons to calculate the cuts in gammaness and θ^2 which provided the best sensitivity in each of the 20 energy bins taken. To do so, the sensitivity was calculated for 10 cuts in gammaness and in θ^2 for each energy. The combination of cuts giving the lower value in sensitivity, keeping a minimum of 10 γ and proton events in the bin (both before and after re-weighting), was selected. The resulting sensitivity curve is shown in picture 4.6.

4.4 Expectation-Maximization method for Hillas Parameters calculation without cleaning

As explained in section 4.2.2, cleaning methods often require the adjust of some parameters, which have to be done empirically, trying to find a balance between a good enough background suppression without loosing too much information from the shower. The cleaning parameters affect particularly the analysis of low energy showers. In the tailcuts method, too strong cleaning thresholds tend to eliminate too many pixels of the already small low energy showers, making the task of γ-hadron separation much more difficult. Also, losing the ellipse shape of γ showers lead to wrong calculation of Hillas parameters and consequently of the disp vector.
For all these reasons, with the aim to research for new analysis methods which could improve the performance of LST, A method for image parameterization which do not require a previous cleaning has been developed. It is based in the Expectation-Maximization (EM) algorithm, where recursively the light content in each pixel is assigned to be part of the shower or the background. In the next sections, the basic concepts of the EM method and how it can be applied to images of showers will be

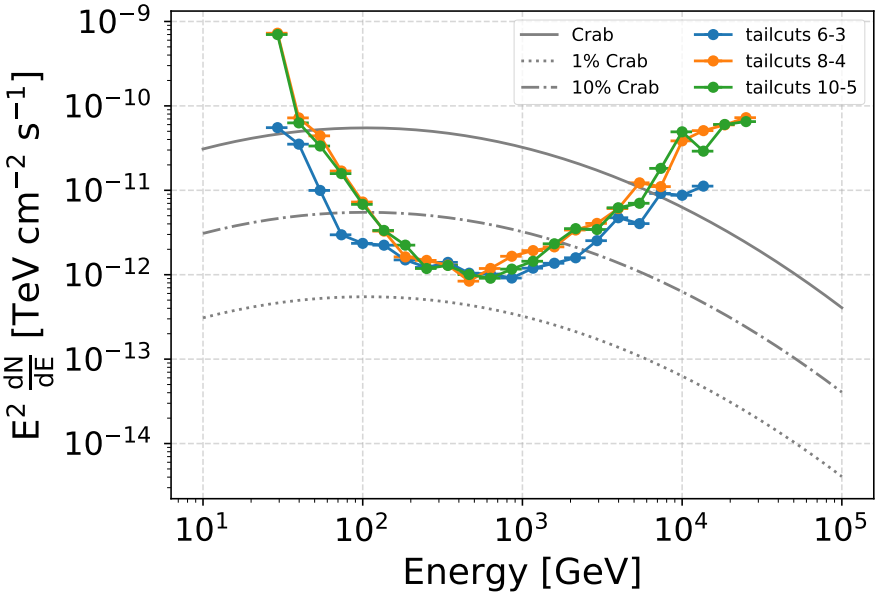

Figure 4.6: Sensitivity plot for 50h of observation of a point-like γ-ray source with a Crab-like spectrum, obtained by selecting the cuts in gammaness and $\theta 2$ which provided the best differential sensitivity per energy bin. As a reference, the spectra of Crab, 10% of Crab and 1 % of Crab emission are shown.

summarized. Then some results comparing the EM with the classic cleaning method will be shown.

4.4.1 The Expectation-Maximization algorithm

The EM algorithm [191] is an iterative method for calculating maximum likelihood estimates of model parameters, where the model depends on unobserved data or latent variables. These latent variables would be those that cannot be observed in the data set, but still can influence other random variables. The algorithm will iterate until finding the model parameters and hidden variables that converge to a maximum likelihood estimation.
Given a statistical model which has generated a set of X observed data points, which depends on a set of latent variables Z and unknown parameters θ, the basic iteration of the EM consist on two steps:

- **Expectation**: Creates an expectation function $Q(\theta|\theta^{(t)})$, which is the evaluation of the log-likelihood with the current estimate of the model parameters $\theta^{(t)}$, meaning it calculates the latent variables Z.

$$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta^{(t)}} \tag{4.17}$$

- **Maximization**: Maximizes the log-likelihood function found in the previous

step to find a new set of estimated parameters.

$$\theta^{(t+1)} = argmaxQ(\theta|\theta^{(t)}) \tag{4.18}$$

A common problem that has been successfully solved by the EM algorithm is the mixture models problem, where there is a set of data which has been produced by several density distribution functions, but is impossible to know which distribution has produced each data point. This distinction between distributions would be the latent variable. This is the particular case we face with shower images. We have a distribution of light in the pixels of the camera, an we know that some photoelectrons come from Cherenkov light, and others belong to background light from the night sky. We want to know the fraction of light in each pixel which belongs to each of the distributions.
We assume that the Cherenkov light will follow a bi-dimensional Gaussian distribution, which will therefore produce the typical elliptical shape in the image, and the background is simply a two dimensional constant distribution. The parameters of the model will be the usual Gaussian parameters (mean and standard deviation in two dimensions) $\{x_0, y_0, \sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}$ and the latent variables will be the fraction of photoelectrons from the Cherenkov shower n_{Ch}/n and from the background n_{bkg}/n in each pixel with n total photoelectrons.
The number of photoelectrons belonging to each distribution will be:

$$N_i = \sum_{pixels} P(pixel/i) \cdot N \tag{4.19}$$

Where N_i is the number of photoelectrons produced by the distribution i (Cherenkov or background), N is the total number of photoelectrons in the image and $P(pixel/i)$ is the probability of a photoelectron in the pixel to have been produced by the distribution i.
Using the Bayes theorem:

$$P(pixel/i) = \frac{P_i(pixel) \cdot P_i}{P(pixel)} \tag{4.20}$$

Where $P_i(pixel)$ is the probability of a photoelectron produced by the distribution i to fall in the pixel, P_i is the probability of a photoelectron to be produced by the distribution i and $P(pixel)$ is the probability of a photoelectron to fall in the pixel. This last probability will actually be:

$$P(pixel) = \sum_i P_i \cdot P_i(pixel) \tag{4.21}$$

Where probabilities $P_i = (P_{Ch}, P_{Bkg})$ are equal to the fraction of photoelectrons in the image belonging to the shower(background) with respect to the total number of

photoelectrons. The probabilities $P_i(pixel)$ can be written as:

$$\begin{aligned} P_{Ch}(pixel) &= BiGaus(x_0, y_0, \sigma_{xx}, \sigma_{yy}, \sigma_{xy}) \cdot A_{pixel} \\ P_{bkg}(pixel) &= A_{pixel}/A_{total} \end{aligned} \tag{4.22}$$

Where A_{pixel} is the area of the pixel and A_{total} the sum of the areas of all pixels. The loop in the EM to solve this problem will go as follows:

1. An initial assumption of the Gaussian parameters is made. Because the bi-dimensional Gaussian corresponds to the Hillas ellipse, the means of the distribution will coincide with the center of gravity of the ellipse, which will be close to the pixel with higher charge. To avoid committing a mistake choosing the initial value due to an outlier pixel, we take the initial means of the Gaussian (x_0, y_0) as the coordinates of the baricenter of the three more luminous pixels. The initial values of the standard deviations are set to arbitrary high values $\sigma_{xx} = 20000$, $\sigma_{yy} = 20000$, $\sigma_{xy} = 0$. Also we assume an initial estimation of the fraction of the light belonging to the shower and background as 50% of the total number of photoelectrons for each.
2. Expectation: Using the previous estimation of Gaussian parameters and fractions of photoelectrons, equation 4.20 is solved and the distributions of the shower and the background are obtained. For each pixel the number of photoelectrons belonging to the shower and to the background is calculated.
3. Maximization: Using the distributions from previous steps, the parameters of the bi-dimensional Gaussian corresponding to the Cherenkov light distribution are calculated, and also the fraction of the total photoelectrons produced by each distribution (which is simply the sum of all shower and background pixels content respectively). The new Gaussian parameters will be:

$$\begin{aligned} mean_x &= \frac{1}{N}\sum_{pixels} n_{Ch} x_{pix} \\ mean_y &= \frac{1}{N}\sum_{pixels} n_{Ch} y_{pix} \\ \sigma_{xx} &= \frac{1}{N}\sum_{pixels} n_{Ch} x_{pix}^2 - mean_x^2 \\ \sigma_{yy} &= \frac{1}{N}\sum_{pixels} n_{Ch} y_{pix}^2 - mean_y^2 \\ \sigma_{xy} &= \frac{1}{N}\sum_{pixels} n_{Ch} x_{pix}^2 y_{pix}^2 - mean_x mean_y \end{aligned} \tag{4.23}$$

4. Steps 2 and 3 are repeated until convergence.

The EM method was used to calculate the Hillas parameters of a small subset of the MC production described in section 4.3.3 ($\sim$ 20k events of diffuse γs and protons for training the RFs, and $\sim$ 200k events of point-like gammas for testing the results). These parameters were used to do the energy and direction reconstruction and the γ-hadron classification. The results, compared to the ones obtained using the tailcuts cleaning and parameterization method are shown in the next section.

4.4.2 Comparison of EM without cleaning and tailcuts cleaning

In this section, results on energy resolution, angular resolution and γ-hadron separation using the EM method for Hillas parameters calculation without previous cleaning are presented, compared to the results obtained applying the classic method of tailcuts cleaning for three sets of cleaning parameters.
The time parameters are not included in the features for RF training because the EM code developed for this work is not compatible with the function for the calculation of time parameters in *ctapipe*.

From figure 4.7 it can be appreciated that the results for the EM method are compatible with the tailcuts cleaning method, reassuring that this is a valid method of Hillas parameterization which allows to go through the step of cleaning which clearly can affect the performance results, as shown in section 4.3.3. The principal current caveat of the method is that it is much more time consuming than the simple tailcuts methods. Taking into account that no time optimization work has been done in the code, great improvements can be expected from the method, to become a potential useful tool for the analysis of low energy events.

4.5 Results on real data

4.6 Summary and conclusion

Along this chapter, the chain for the single telescope data analysis of LST1 has been described, and successfully used to derive the performance of the , and to analyze the first real data taken with the instrument.
About the performance estimated using MC simulations, it has been shown that the energy resolution reaches an error as low as 20% in the range from $\sim$ 100 GeV to a few TeV and the angular resolution in the same energy range is $\sim 0.2^{o}$. From the three sets for tailcuts parameters, the one giving best results has been the 6-3 tailcuts. The best sensitivity of for 50 hours of observation is reached in the range between 100 GeV and 1 TeV, going below the 10% of the Crab flux.
The EM method for Hillas parameterization without cleaning has been presented as a potential alternative, with similar performance to the classic tailcuts cleaning (without time parameters), to avoid the necessity to adjust the tailcuts parameters which clearly affect the performance results. This method is useful for low energy events, with very few photoelectrons, where tailcuts cleaning can suppose the loss of information from the shower.

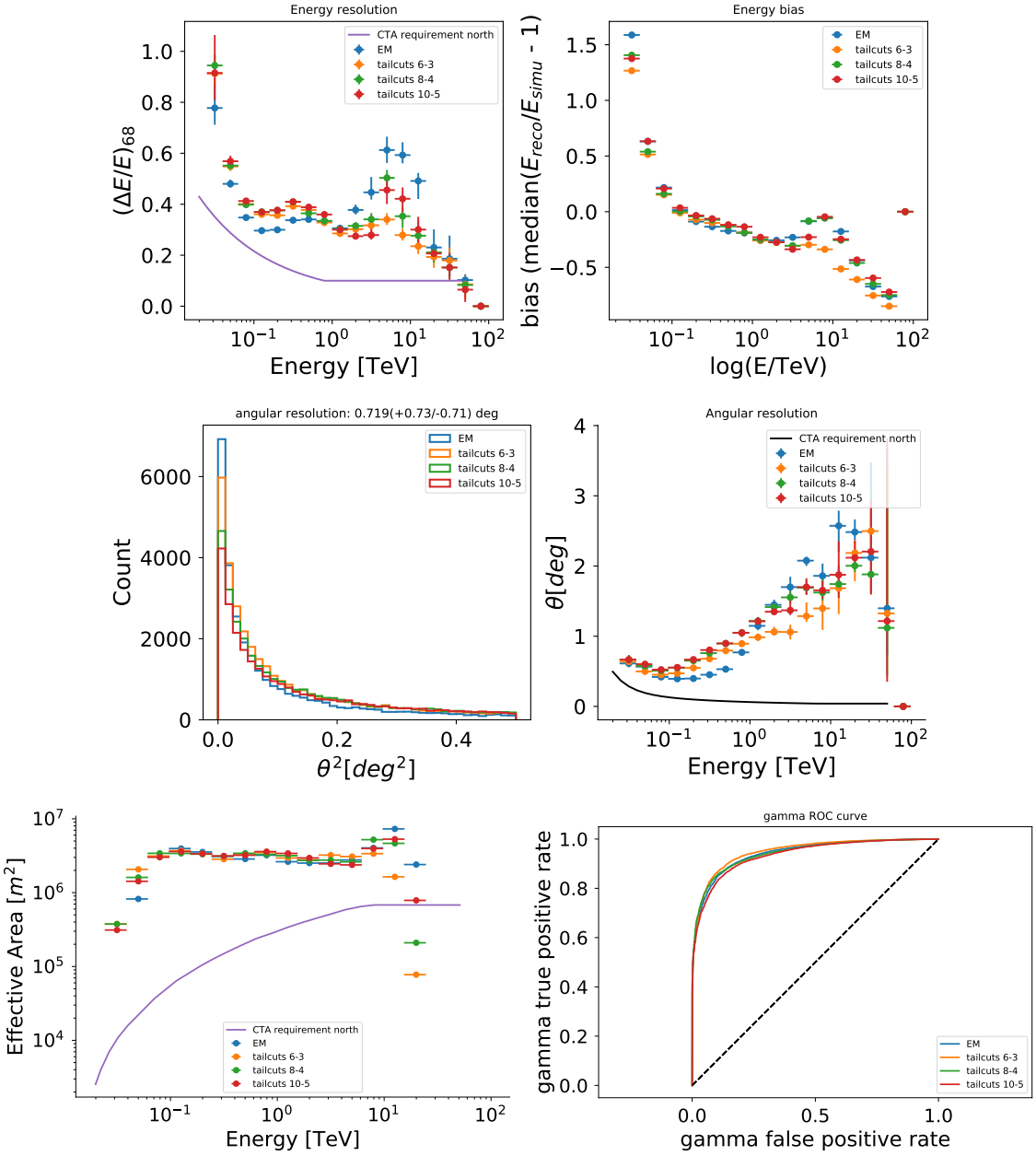

Figure 4.7: Energy resolution (*top*), angular resolution (*middle*), effective area (*bottom-left*) and ROC curve (textitbottom-right) plots for a small subset of events analyzed using the EM method for Hillas parameterization without cleaning, compared to three different subsets of tailcuts cleaning parameters.