PROCEEDINGS OF SCIENCE

ctapipe – Prototype Open Event Reconstruction Pipeline for the Cherenkov Telescope Array

Maximilian Linhoff[a,*] **Lukas Beiske**[a] **Noah Biederbeck**[a] **Stefan Fröse**[a] **Karl Kosack**[b] **and Lukas Nickel**[a] **for the CTA Consortium and Observatory**

[a] *Astroparticle Physics, Department of Physics, TU Dortmund University, Otto-Hahn-Str. 4a, 44227, Dortmund, Germany*

[b] *Université Paris-Saclay, Université Paris Cité, CEA, CNRS, AIM F-91191, Gif-sur-Yvette, France*

E-mail: maximilian.linhoff@tu-dortmund.de

The Cherenkov Telescope Array Observatory (CTAO) is the next-generation ground-based gamma-ray observatory currently under construction. It will improve over the current generation of imaging atmospheric Cherenkov telescopes (IACTs) by a factor of five to ten in sensitivity and it will be able to observe the whole sky from a combination of two sites: a northern site in La Palma, Spain, and a southern one in Paranal, Chile.

CTAO will also be the first open ground-based gamma-ray observatory. Accordingly, the CTAO data processing pipeline is developed as open-source software and `ctapipe` will be a core package therein. The event reconstruction pipeline accepts raw data of the telescopes and processes it to produce suitable input for the higher-level science tools. Its primary tasks include reconstructing the physical properties of each recorded air shower and providing the corresponding instrument response functions.

`ctapipe` is a python framework providing algorithms and command-line tools to facilitate raw data calibration, image extraction, image parametrization and event reconstruction. Its current main focus is the analysis of simulated data but it has also been successfully applied for the analysis of data obtained with the CTA prototype telescopes, and first science results have now been obtained by the LST-1 collaboration using `ctapipe`. A plugin system also allows the processing of non-CTA data.

Recent updates, including event reconstruction using machine learning and a new plugin system as well as the roadmap towards a 1.0 release will be presented.

38th International Cosmic Ray Conference (ICRC2023)
26 July – 3 August, 2023
Nagoya, Japan

*Speaker

PoS(ICRC2023)703

1. Introduction

The Cherenkov Telescope Array Observatory (CTAO)[1] is the next generation very-high-energy gamma-ray observatory, currently under construction. It will be sensitive to gamma-ray energies between ~ 20 GeV and 300 TeV and provide full-sky coverage by operating two sites: one on the Southern hemisphere at the Paranal Observatory in Chile and one on the Northern hemisphere at the Roque de los Muchachos Observatory on the Canary Island of La Palma, Spain.

After its initial construction phase, the *alpha* configuration, the Southern array will comprise 14 Medium-Sized Telescopes (MSTs) with 12 m mirror diameter and 37 Small-Sized Telescopes (SSTs) with 4 m mirror diameter. The Northern site will comprise four Large-Sized Telescopes (LSTs) with 23 m mirror diameter and nine MSTs [8]. Additional funding for at least two LSTs for the Southern array has been secured[2].

CTAO will detect gamma rays by measuring the Cherenkov light emitted in extensive air showers (EAS) using very fast and sensitive cameras based on photo-multiplier tubes (PMTs) or silicon photo-multipliers (SiPMs). From the recorded Cherenkov light, the low-level analysis pipeline has to reconstruct the physical properties of the primary particle: its energy, direction and particle type. The latter is necessary since EAS are also induced by charged cosmic rays, which form a large background for imaging atmospheric cherenkov telescopes (IACTs) such as CTAO.

`ctapipe` is a Python framework implementing the necessary steps to perform these tasks, its development and features will be described in the next section.

2. ctapipe

`ctapipe` is a python package providing library functions and command-line tools to perform the analysis tasks listed in the previous section. It is developed as open-source software and the project – which was started in 2015 – is developed on Github. It is still under heavy development; the current release at the time of writing is 0.19.3 [12]. In total, 61 contributors have made this project possible. Releases are published to PyPI[3], `conda`-packages are provided using `conda-forge`[4] and the documentation is hosted on readthedocs[5].

`ctapipe` is build upon the scientific python ecosystem: it depends on `astropy` [20] for coordinate transformations, high-precision timestamps, physical quantities and table operations, while `numpy` [9] and `scipy` [21] are used for numerical algorithms and statistics. The `pytables`[6] library provides IO for the Hierarchical Data Format, Version 5 (HDF5)[7], the file format used for most intermediate `ctapipe` data products. Finally, ctapipe makes use of the `numba` [13] jit-compiler for most of its performance critical code.

[1] www.cta-observatory.org
[2] See www.cta-observatory.org/project/industry/\#1675157418077-1c4af6db-4be5
[3] pypi.org/project/ctapipe
[4] anaconda.org/conda-forge/ctapipe
[5] ctapipe.readthedocs.org
[6] www.pytables.org
[7] www.hdfgroup.org/solutions/hdf5

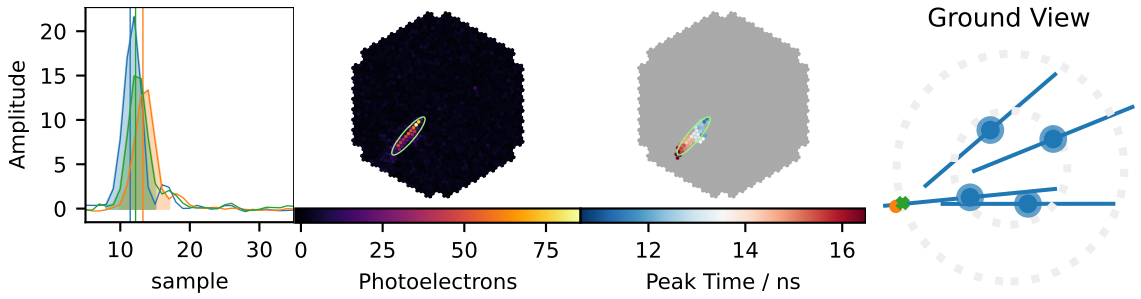


Figure 1: Steps of the analysis up to shower geometry reconstruction: 1. The Cherenkov pulses in each pixel are found and integrated (left) to obtain the number of photons (second from left) and peak times (third from left). 2. The resulting image is cleaned, the pixels not selected are shown in gray in the peak time plot. 3. The images are parametrized, including the Hillas parameters which are visualized using an ellipse. 4. The physical properties of the primary are reconstructed. The plot on the right shows the impact point of the primary on the ground, the green cross is the reconstructed position and the orange point the true value known from the simulations.

The goal of `ctapipe` is to provide the necessary tools and methods to process the pre-calibrated raw data (called DL0) of CTAO to *science-ready data* (called DL3) usable as input to the high-level science analysis tools, which will be based on Gammapy [6].

At DL0, the main unit of data is the *subarray event*, that corresponds to a single air shower recorded by the telescopes assigned to the current observation. A subarray event is composed of a varying number of *telescope events*, data of each individual telescope that was triggered as part of the subarray event. The DL0 telescope data mainly consists of the time-series of the photo-multipliers in the Cherenkov camera, additional metadata, and monitoring data, like the pointing positions of the telescopes. At DL3, only the reconstructed properties of the primary particle and the event timestamp remain for each subarray event and no telescope-wise information is required. In addition to these event lists, the instrument response functions (IRFs), which give the relationship between a true gamma-ray signal and the observed, reconstructed properties of the events are needed for the high-level analysis and are thus also part of DL3 data.

In the "classical" IACT analysis scheme, which is currently implemented in `ctapipe`, the analysis is performed in steps resulting in intermediate data levels:

1. The time-series in each pixel are reduced to two numbers each: the estimated number of Cherenkov photons and their mean arrival time, referred to as *DL1 images*.

2. The resulting images are "cleaned", meaning that only the pixels likely to contain a significant Cherenkov signal are selected for further processing.

3. The cleaned images are described using parameters, including for example the Hillas parameters [10], resulting in a further reduction of data volume. This data level is referred to as *DL1 parameters*.

4. From the DL1 parameters of each telescope, the physical properties of the primary particle are estimated. For the direction, geometrical approaches using the stereoscopic nature of multi-telescope observations can be used. For the energy and particle type, machine learning

methods are employed. In the case of monoscopic events, machine learning is also used for the estimation of the direction.

The following sections describe each of these steps and their implementation in `ctapipe`.

2.1 Image Extraction

The first step in the `ctapipe` analysis is to reduce the pixel-wise time-series information to the number of photons and their mean arrival time. The extracted values can be calibrated for pixel inhomogeneities, both in amplitude and time. `ctapipe` supports different algorithms for extracting these quantities from single-pixel waveforms, from simple peak-finding algorithms to more complex ones that combine the waveforms of multiple pixels or that fit the expected time evolution of the shower and use that to define the integration window for each pixel.

2.2 Image Parametrization

After removal of noise pixels, the cleaned image is parametrized in order to make it exploitable by subsequent algorithms, in particular shower geometry reconstructors and/or machine-learning models that perform the shower property reconstruction. Among the most important parameters are the classical Hillas parameters [10], which describe the orientation and extension of the shower image in the camera, which is needed for the following reconstruction steps. Additionally, `ctapipe` implements general descriptive statistics of the images, morphological features like the number of isolated pixel groups and parameters describing the containment of the shower's image in the camera.

2.3 Reconstruction of Primary Particle Properties

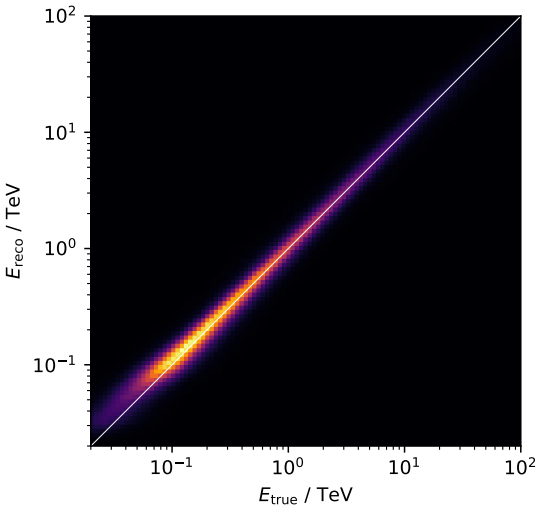
The previous steps are performed independently for each telescope event. Now, the goal is to combine the telescope events into a single prediction for the main physical properties of the primary particle that induced the air shower. Three main properties need to be estimated:

Energy the primary energy, assuming that the shower was induced by a gamma-ray,

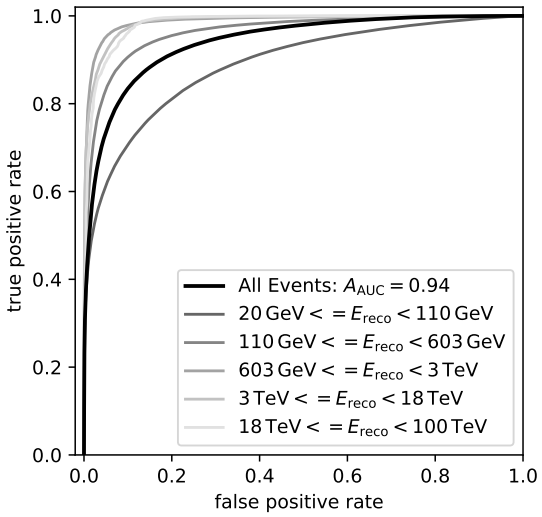
Particle Type the particle type (gamma, electron, proton, ...) that induced the air shower,

Direction the direction on the sky where the primary particle came from.

Tools to train and apply machine learning models provided by scikit-learn [19] to solve these tasks were first introduced in `ctapipe` 0.18.0. Through the configuration system, it is possible to select any of the models implemented in scikit-learn and define their hyperparameters. Since most, if not all, classical machine learning algorithms cannot cope with variable-length input data, the approach chosen for `ctapipe` is to train models for each telescope type and apply them to a single telescope event, augmented with the available subarray-wide information. To form the final, single prediction for the subarray event, the telescope predictions are averaged. The training and application tools allow the application of quality criteria to select which events should be included in the training. It is also possible to combine features in the input files to form new ones on the fly.



(a) Energy migration for simulated diffuse gamma rays up to 2° away from the center of the field of view.



(b) ROC curve for all events and for events in different ranges of the predicted gamma-ray energy. Note that for the background this is not the predicted primary energy but the energy predicted assuming the shower was induced by a gamma ray.

Figure 2: Performance visualizations for the machine learning event reconstruction steps for the CTA-North alpha configuration processed with ctapipe v0.19.2.

Energy Estimation The gamma-ray energy estimation is a one-dimensional regression task, that can be performed using any of the regressors implemented in scikit-learn. Most commonly, decision-tree-based ensemble methods like random forests or different kinds of boosted decision trees are employed. Due to the large value range of multiple orders of magnitude, the models are trained to predict the logarithm of the energy. Models are trained per telescope type (LST, MST, SST) and single-telescope predictions are combined into a common prediction for the subarray event using a simple or weighted average. Multiple weights can be chosen, for example weighting bright images stronger than dim images. Figure 2a shows the energy migration matrix obtained for the CTA-North alpha configuration.

Particle Type Classification Currently, `ctapipe` implements binary classification for a given particle type as signal vs. background, resulting in a score that is a measure of how likely the given event belonged to the signal class. Usually, this means using gamma ray events for the signal class and proton events for the background class. It is possible to include the predicted gamma-ray energy from the previous step in the features for the particle type classification. Figure 2b shows receiver operating characteristic (ROC) curves for different ranges of estimated gamma-ray energy for the CTA-North alpha configuration. Again, one model is trained for each telescope type and telescope events are combined using a (weighted) average.

Geometry Estimation The geometry of an air shower is described by the direction and hypothetical impact point on the ground of the primary particle. In the case of stereoscopic observations, these can be estimated using geometrical optimization approaches from the orientation of the images

in each telescope as described by the Hillas parameters. This is implemented in `ctapipe` and can be used also as input for the machine learning reconstruction of energy and particle type. Especially for the energy estimation, the distance of the telescope to the impact point is crucial.

The direction reconstruction is a much harder problem in case of a single telescope observing on its own. In this case, machine learning is also applied to estimate the direction. In general, this would be a two-dimensional regression, however, relying on the Hillas parameters allows a simplification to a one-dimensional regression and binary classification. Using the so-called disp-method [14], the distance from the center of gravity of the shower along the main shower axis is predicted, this is the absolute value of `disp`. This leaves two possible solutions for the direction, chosing the correct one is the goal of the following binary classification, which can be interpreted as the sign of `disp`.

2.4 Analysis Workflow

The steps from DL0 to the input data for the training of the machine learning models, which comprise the DL1 image parameters and DL2 parameters obtained from the geometrical reconstruction approaches, are performed using the `ctapipe-process` tool. It is run in parallel on many runs of the DL0 simulation output, which are then merged using `ctapipe-merge` into larger datasets for training and validation. This is followed by training the energy models using `ctapipe-train-energy-regressor` on a first set of gamma ray simulations. These models are applied to another set of gamma rays and proton simulations using `ctapipe-apply-models`, which are then used as input to `ctapipe-train-particle-classifier`. Finally, all trained models can be applied to validation datasets, again using `ctapipe-apply-models`. Using the DIRAC transformation system and workflow definitions provided by CTADIRAC [2], these steps can be applied in large scale on the CTA GRID.

2.5 Data Formats

`ctapipe` offers a plugin system for event-wise input data that allows users to implement readers for their data formats without having to modify `ctapipe` itself. `ctapipe` itself comes with two implementations, one for the data format used for CTA simulations, which are produced using `sim_telarray` [3] and one for its own data format using HDF5. The data are split by data level and telescope-wise information is stored in tables for each telescope. Metadata, like units and column descriptions are attached using HDF5 attributes. `ctapipe` offers functionality to conveniently load and join these different tables into `astropy.table.Table` instances.

3. First Scientific Results Obtained with ctapipe-based Analyses

The Large-Sized Telescope 1 (LST-1) has been inaugurated in 2018 and is since in its commissioning phase, taking first scientific observations. Its data analysis software, `lstchain` [16], is based on `ctapipe` and was used to publish the first scientific results obtained with a `ctapipe`-based analysis in [1]. A publication on the telescope and analysis performance using Crab Nebula observations is accepted [5] and is also presented at this conference [17]. Showing also the ability to analyze data from the current generation of telescopes, the performance of joint analysis of MAGIC and LST-1 data using ctapipe is presented in [7].

4. Conclusions and Outlook

With the latest releases, `ctapipe` is able to produce fully reconstructed DL2 event lists using geometrical approaches and scikit-learn-based machine learning models. The analysis can be configured using a flexible configuration system, provenance information is recorded automatically. A few critical steps remain until the full pipeline to science-ready data (DL3) is ready:

- Optimizing event selection criteria, mainly the energy dependent decision threshold in the gamma-hadron classification to optimize the sensitivity.
- Afterwards, the instrument response functions need to be computed. A python library for this task is under development [15], but is not yet directly integrated with `ctapipe` to produce IRFs from `ctapipe` data products.
- To allow multiple science cases in the same data release and reduce systematic uncertainties, CTAO plans to categorize events into discrete types based on their expected reconstruction quality and expected background contamination. This categorization is also yet to be implemented in `ctapipe`.
- Work on more advanced reconstruction algorithms is underway, this includes ImPACT [18] and a plugin system for machine learning models based e. g. on deep neural networks [4, 11].
- While the LST-1 has shown that it is possible to use `ctapipe` successfully for the analysis of actual observations, `ctapipe` itself was mainly focused on the analysis of the CTA simulations. Much of the complexities of calibration and monitoring data still need to be integrated into `ctapipe`.

Acknowledgements

This work was conducted in the context of the CTA Consortium and CTA Observatory. We gratefully acknowledge financial support from the agencies and organizations listed here: https://www.cta-observatory.org/consortium_acknowledgments.

References

[1] S. Abe et al. "Multiwavelength study of the galactic PeVatron candidate LHAASO J2108+5157." In: *Astronomy & Astrophysics* 673.A75 (2023). DOI: 10.1051/0004-6361/202245086.

[2] L. Arrabito et al. "The Cherenkov Telescope Array production system prototype for large-scale data processing and simulations." In: *Proceedings, 25th International Conference on Computing in High Energy and Nuclear Physics*. Vol. 251. 02029. 2021. DOI: 10.1051/epjconf/202125102029.

[3] K. Bernlöhr. "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray." In: *Astroparticle Physics* 30.3 (Oct. 2008), pp. 149–158. DOI: 10.1016/j.astropartphys.2008.07.009.

[4] A. Brill et al. *CTLearn: Deep learning for imaging atmospheric Cherenkov telescopes event reconstruction*. Version 0.7.0. 2023. DOI: 10.5281/zenodo.7908252. URL: https://github.com/ctlearn-project/ctlearn.

[5] CTA-LST Project, S. Abe, et al. *Observations of the Crab Nebula and Pulsar with the Large-Sized Telescope Prototype of the Cherenkov Telescope Array*. 2023. arXiv: 2306.12960 [astro-ph.HE].

[6] C. Deil et al. "Gammapy - A prototype for the CTA science tools." In: *Proceedings, 35th International Cosmic Ray Conference*. Vol. 301. 766. Busan, South Korea, 2017. DOI: 10.22323/1.301.0766.

[7] F. Di Pierro et al. for the CTA-LST Project and the MAGIC collaboration. "Performance of joint gamma-ray observations with MAGIC and LST-1 telescopes." In: *Proceedings, 38th International Cosmic Ray Conference*. Vol. 444. 636. Nagoya, Japan, 2023.

[8] O. Gueta for the CTA Consortium and the CTA Observatory. "The Cherenkov Telescope Array: layout, design and performance." In: *Proceedings, 37th International Cosmic Ray Conference*. Vol. 395. 885. 2021. DOI: 10.22323/1.395.0885.

[9] C. R. Harris et al. "Array programming with NumPy." In: *Nature* 585 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[10] A. M. Hillas. "Cerenkov light images of EAS produced by primary gamma rays and by nuclei." In: *Proceedings, 19th International Cosmic Ray Conference*. Vol. 3. 1985.

[11] M. Jacquemont et al. *GammaLearn*. Version v0.10.1. 2023. DOI: 10.5281/zenodo.7991254. URL: https://gitlab.in2p3.fr/gammalearn/gammalearn.

[12] K. Kosack et al. *ctapipe – Low-level data processing pipeline software for the Cherenkov Telescope Array*. Version v0.19.3. 2023. DOI: 10.5281/zenodo.8063139. URL: https://github.com/cta-observatory/ctapipe.

[13] S. K. Lam, A. Pitrou, and S. Seibert. "Numba: A llvm-based python jit compiler." In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. 2015, pp. 1–6.

[14] R. W. Lessard et al. "A new analysis method for reconstructing the arrival direction of TeV gamma rays using a single imaging atmospheric Cherenkov telescope." In: *Astroparticle Physics* 15.1 (Mar. 2001), pp. 1–18. DOI: 10.1016/s0927-6505(00)00133-x.

[15] M. Linhoff et al. *pyirf*. Version v0.8.1. 2023. DOI: 10.5281/zenodo.7741289. URL: https://github.com/cta-observatory/pyirf.

[16] R. Lopez-Coto et al. *lstchain*. Version v0.10.0. 2023. DOI: 10.5281/zenodo.8046664. URL: https://github.com/cta-observatory/cta-lstchain.

[17] D. Morcuende et al. for the CTA-LST Project. "Performance of the Large-Sized Telescope prototype of the Cherenkov Telescope Array." In: *Proceedings, 38th International Cosmic Ray Conference*. Vol. 444. 594. Nagoya, Japan, 2023.

[18] R. Parsons and J. Hinton. "A Monte Carlo template based analysis for air-Cherenkov arrays." In: *Astroparticle Physics* 56 (2014), pp. 26–34.

[19] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[20] The Astropy Collaboration, A. M. Price-Whelan, et al. "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package." In: *The Astrophysical Journal* 935.2 (Aug. 2022), p. 167. DOI: 10.3847/1538-4357/ac7c74.

[21] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

PoS(ICRC2023)703