Machine Learning in LST: advanced trigger and sensitivity

Tjark Miener (tjark.miener@unige.ch)

13.12.2023 - Swiss CTA Observatory Day 2023-24

CTLearn community

- **CTLearn** is a high-level Python package for using **Deep Learning** for **IACT event reconstruction** maintained by Nieto D. (Madrid) and Miener T. (UniGe)
- Perez A. (PhD student at Madrid), Burmistrov L. (UniGe) and Abellan Beteta C., Bezshyiko Ia., Hijano Mendizabal G., Serra N. (Uni Zurich) joined the team for the **AI Trigger system** project. Expertise on **porting** neural networks to **FPGA**s.
- Lacave B. (PhD student at UniGe with Heller M.) started to work on the **CTLearn** analysis pipeline for the **SST1M** and **LST-1** prototypes.

DOI 10.5281/zenodo.3345947

https://github.com/ctlearn-project/ctlearn

https://ctlearn.readthedocs.io

Outline

- Quick introduction to IACT low-level analysis with Deep Learning
- Ongoing performance studies of LSTs equipped with high-resolution advanced SiPM cameras using CNNs
- Developments on the AI Trigger system of the adv. SiPM camera
- CTLearn at CSCS

IACT low-level analysis

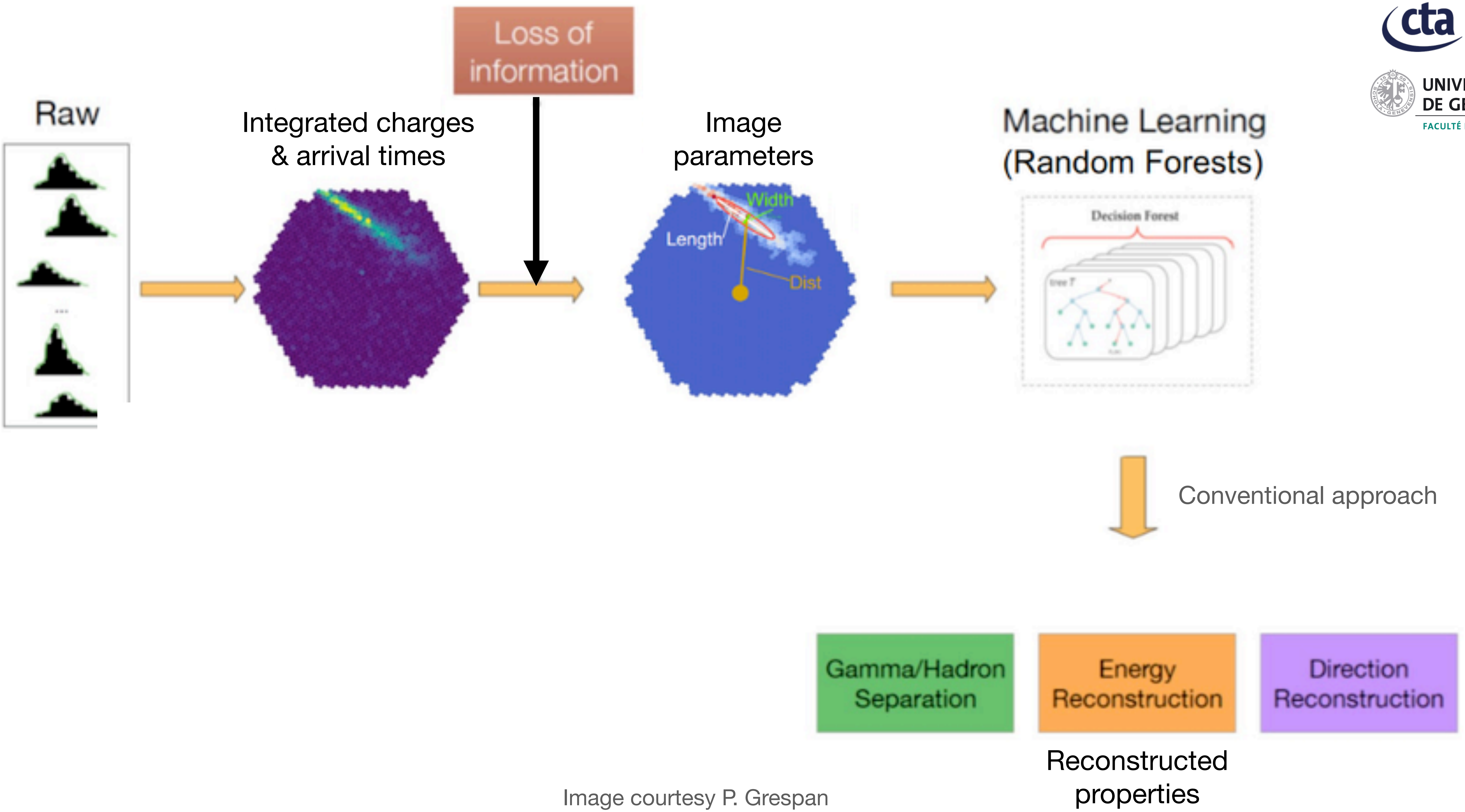


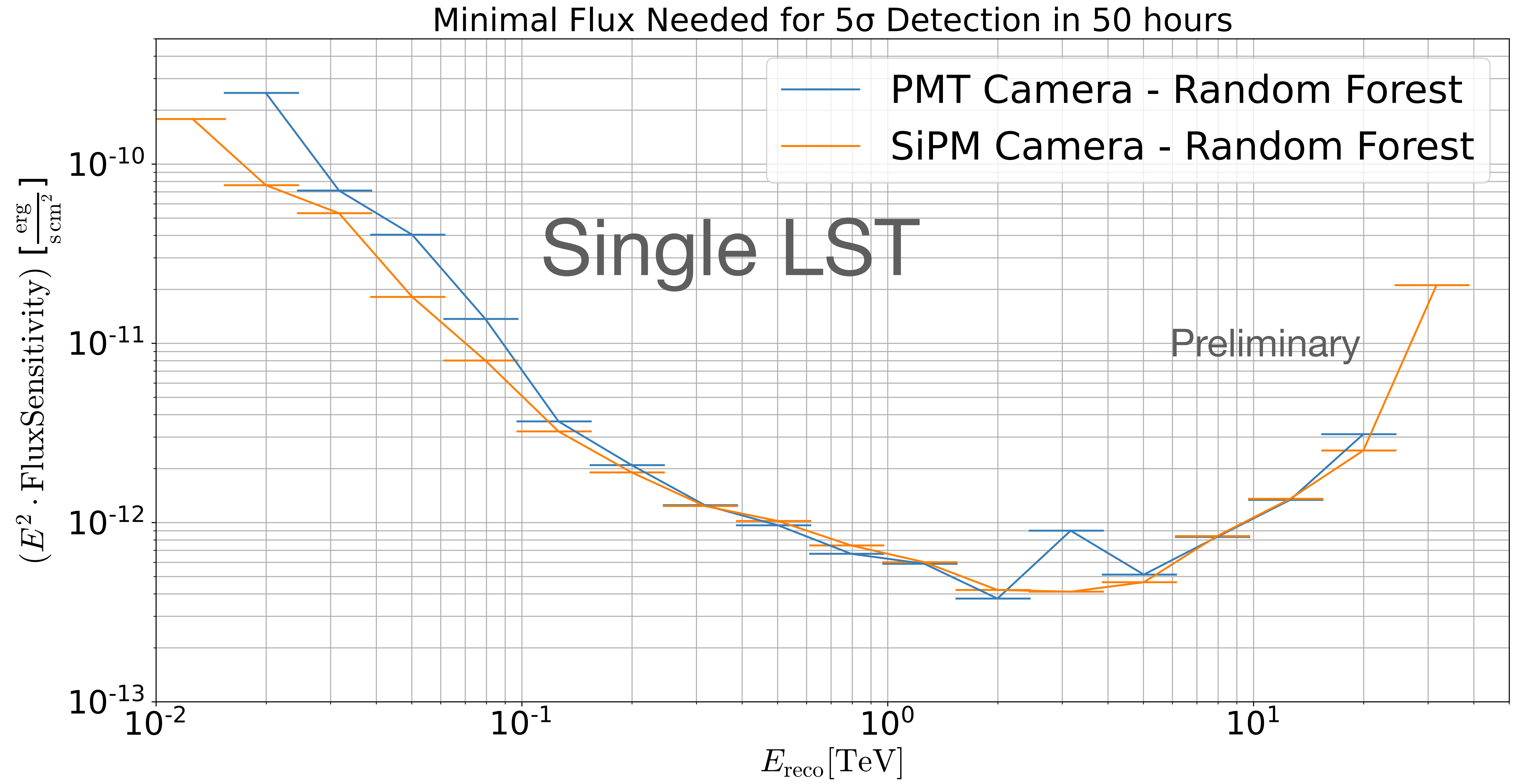
Image courtesy P. Grespan

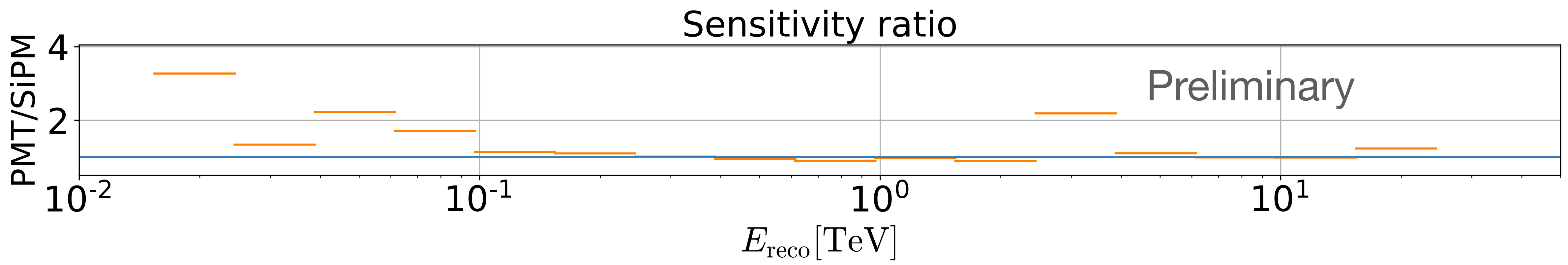
Random Forest: current PMT camera vs high-resolution SiPM camera

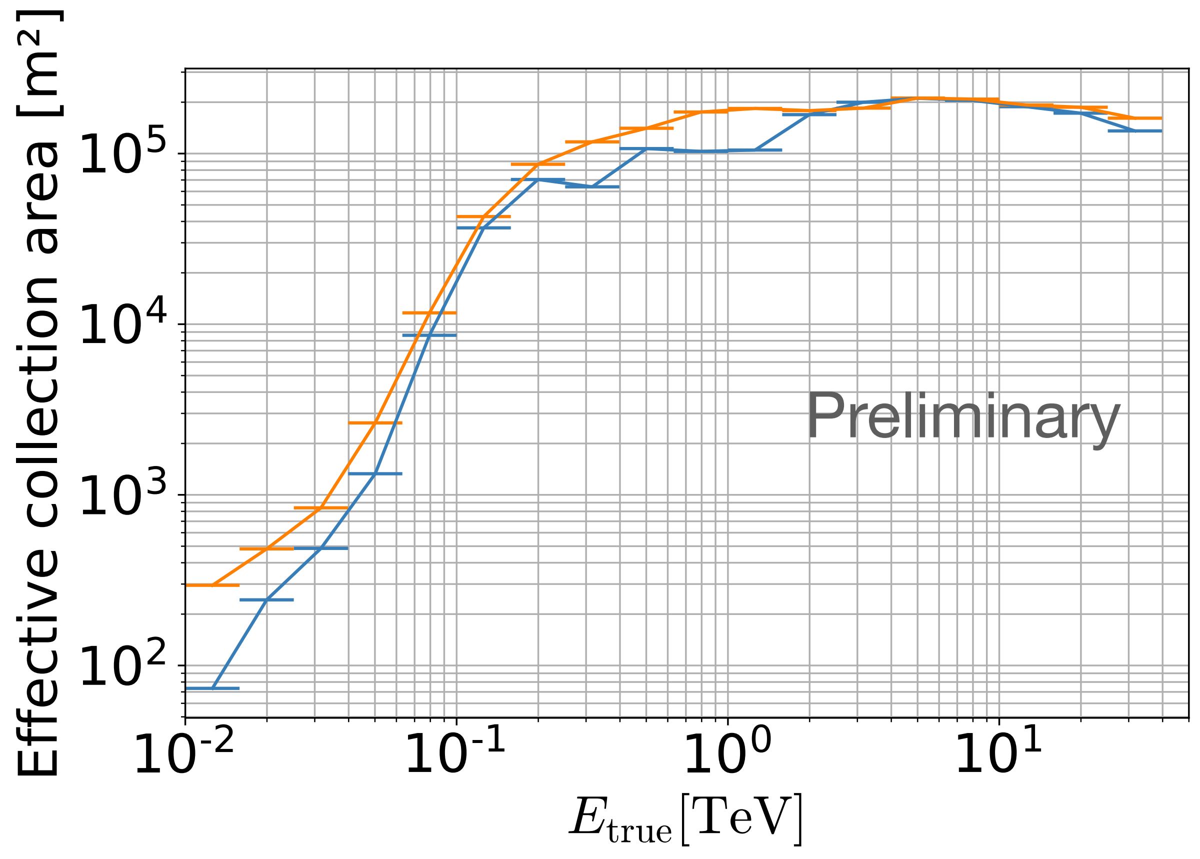
CTLearn

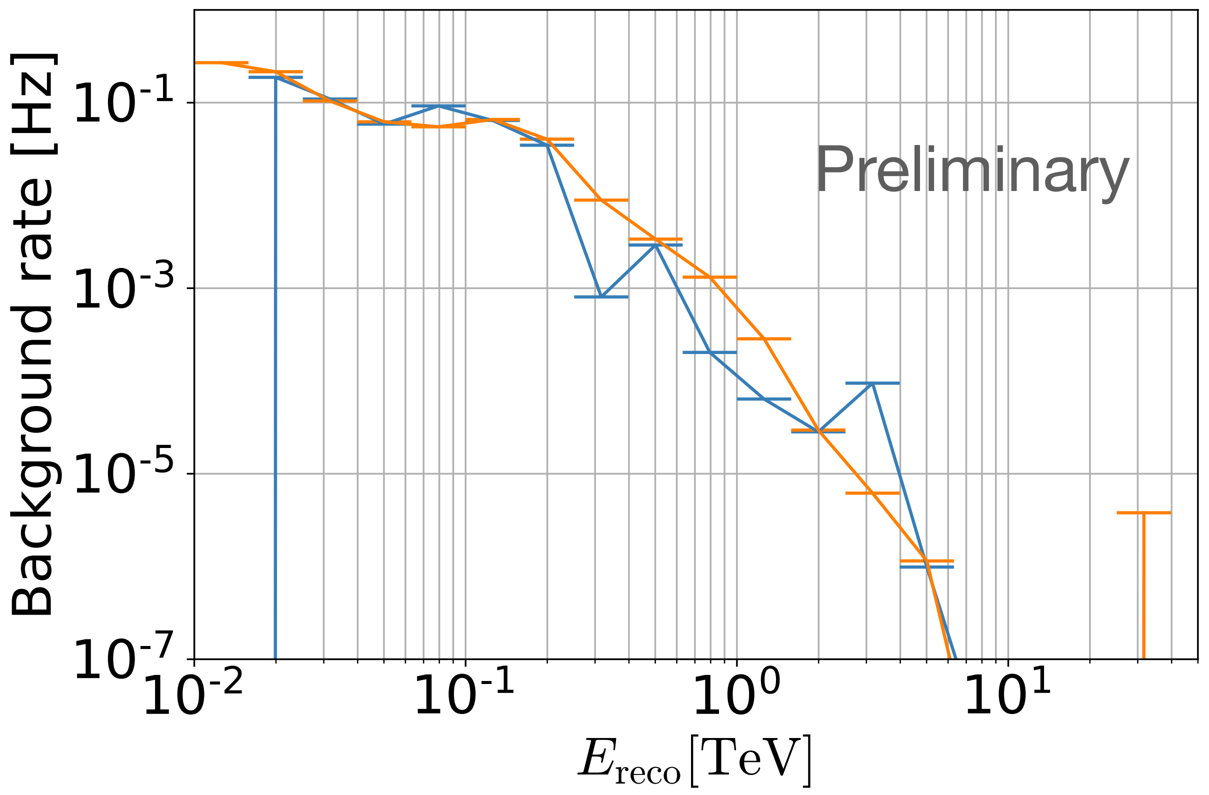
cta cherenkov telescope array

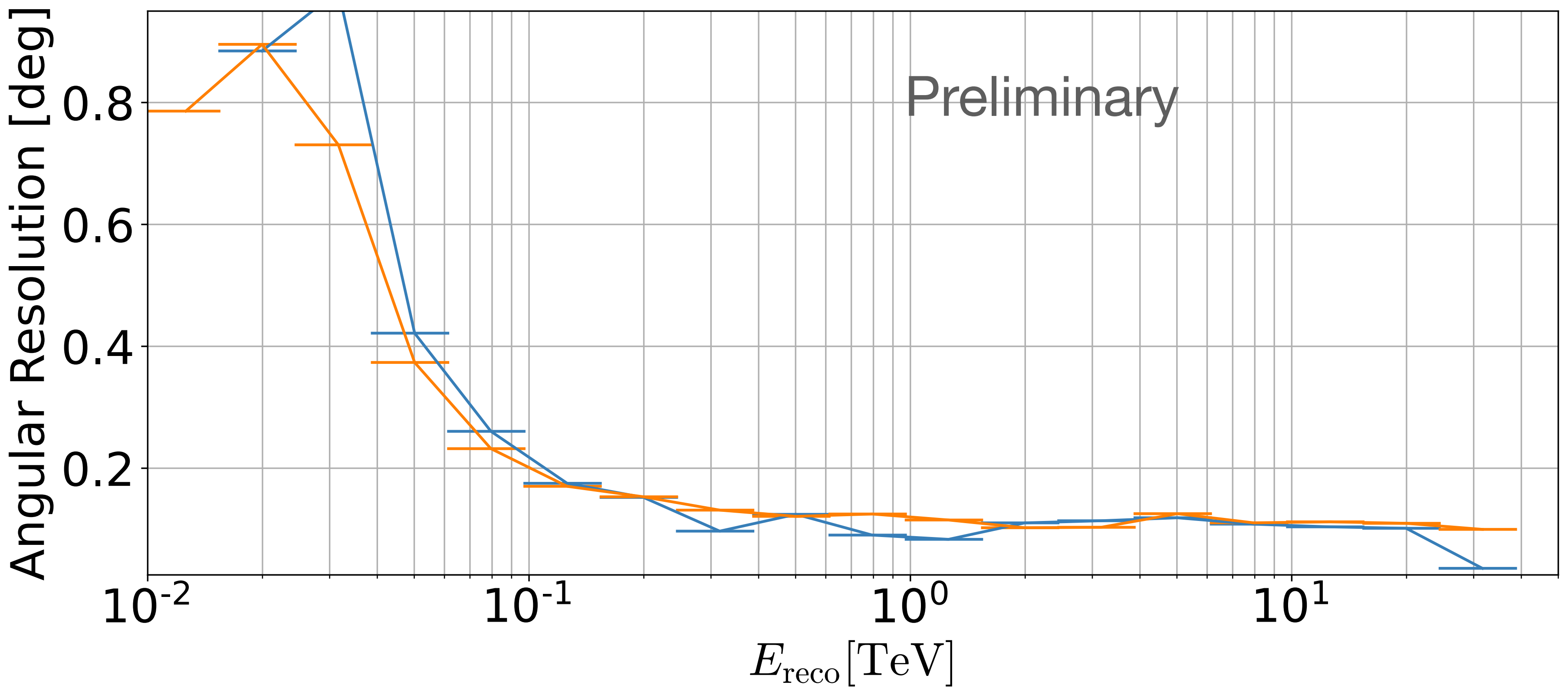
UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

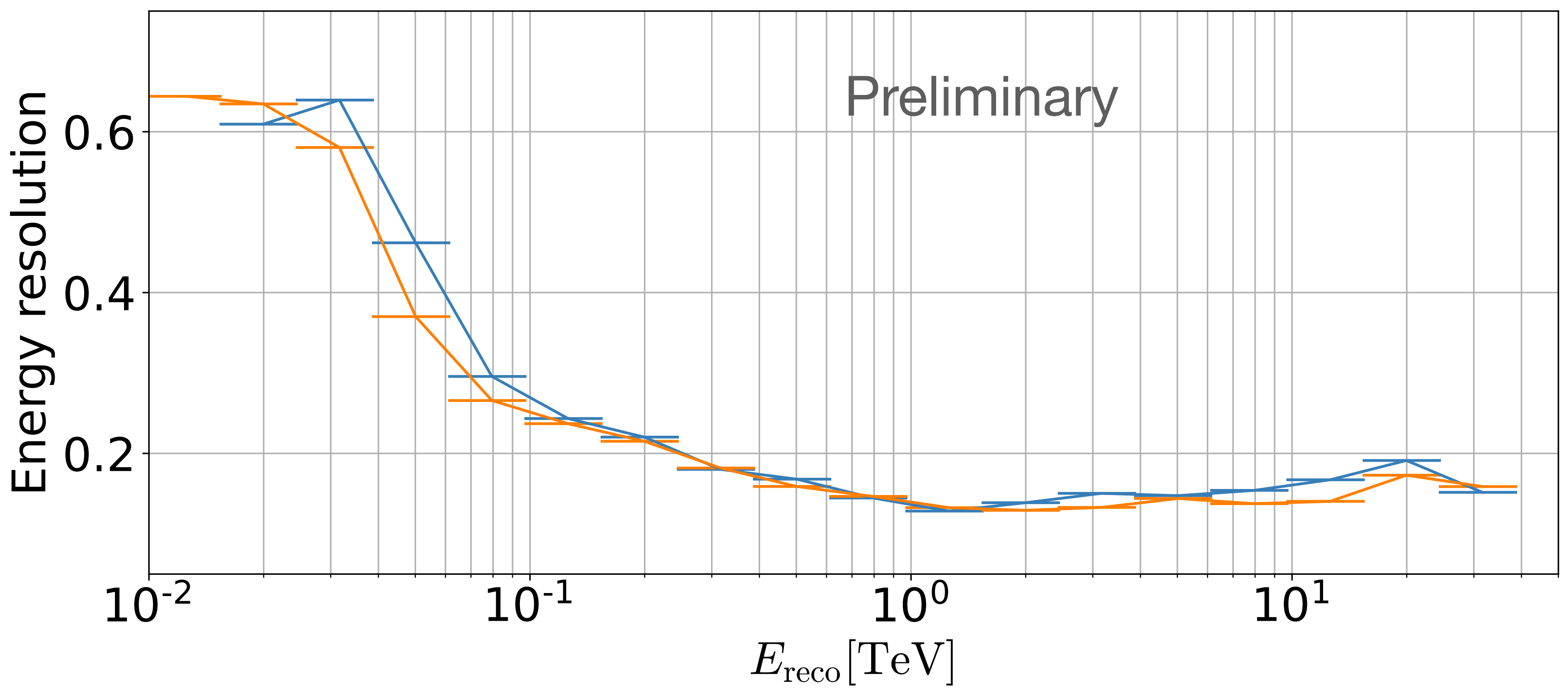












T. Miener et al.

IACT low-level analysis with Deep Learning

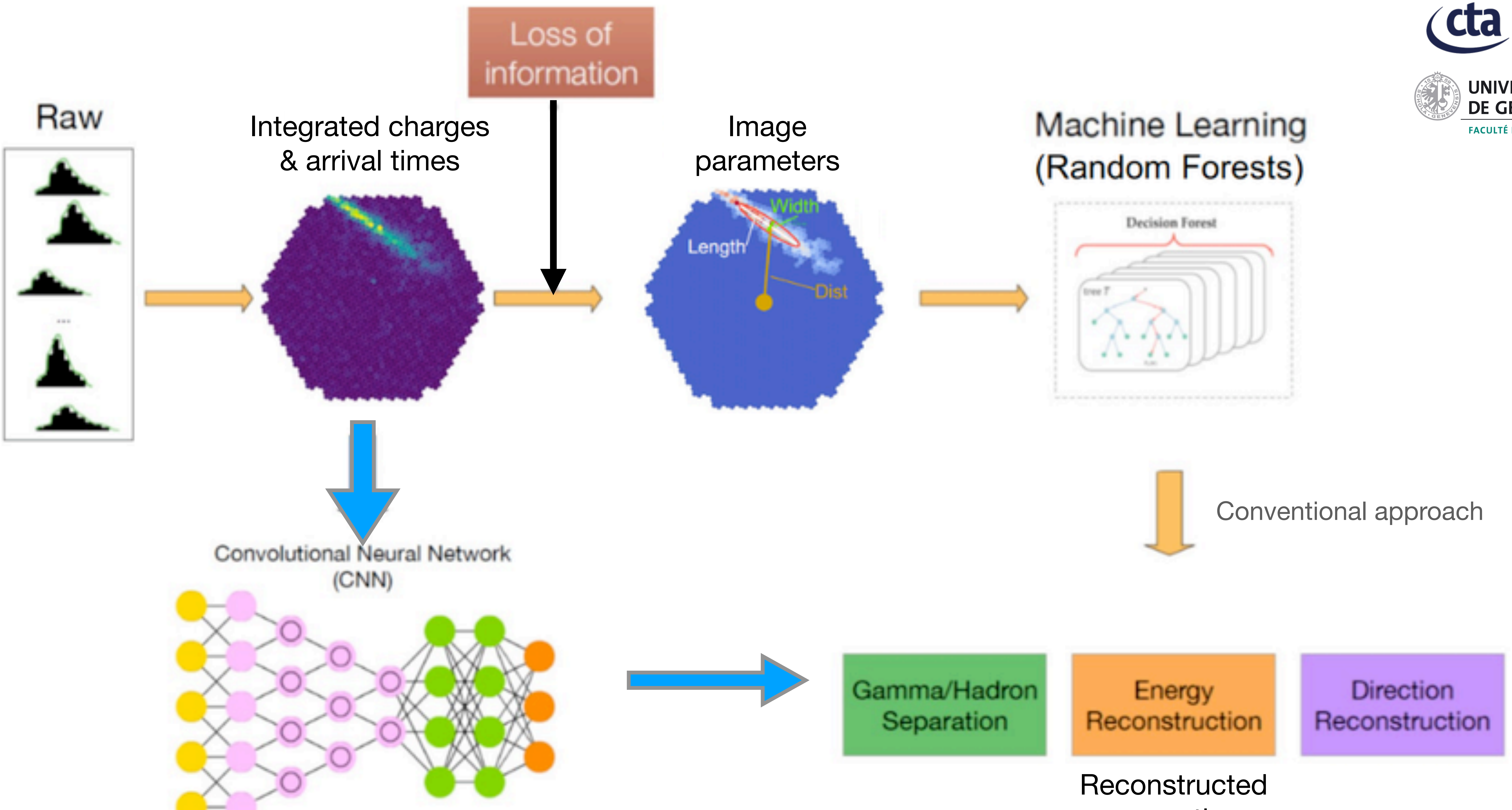
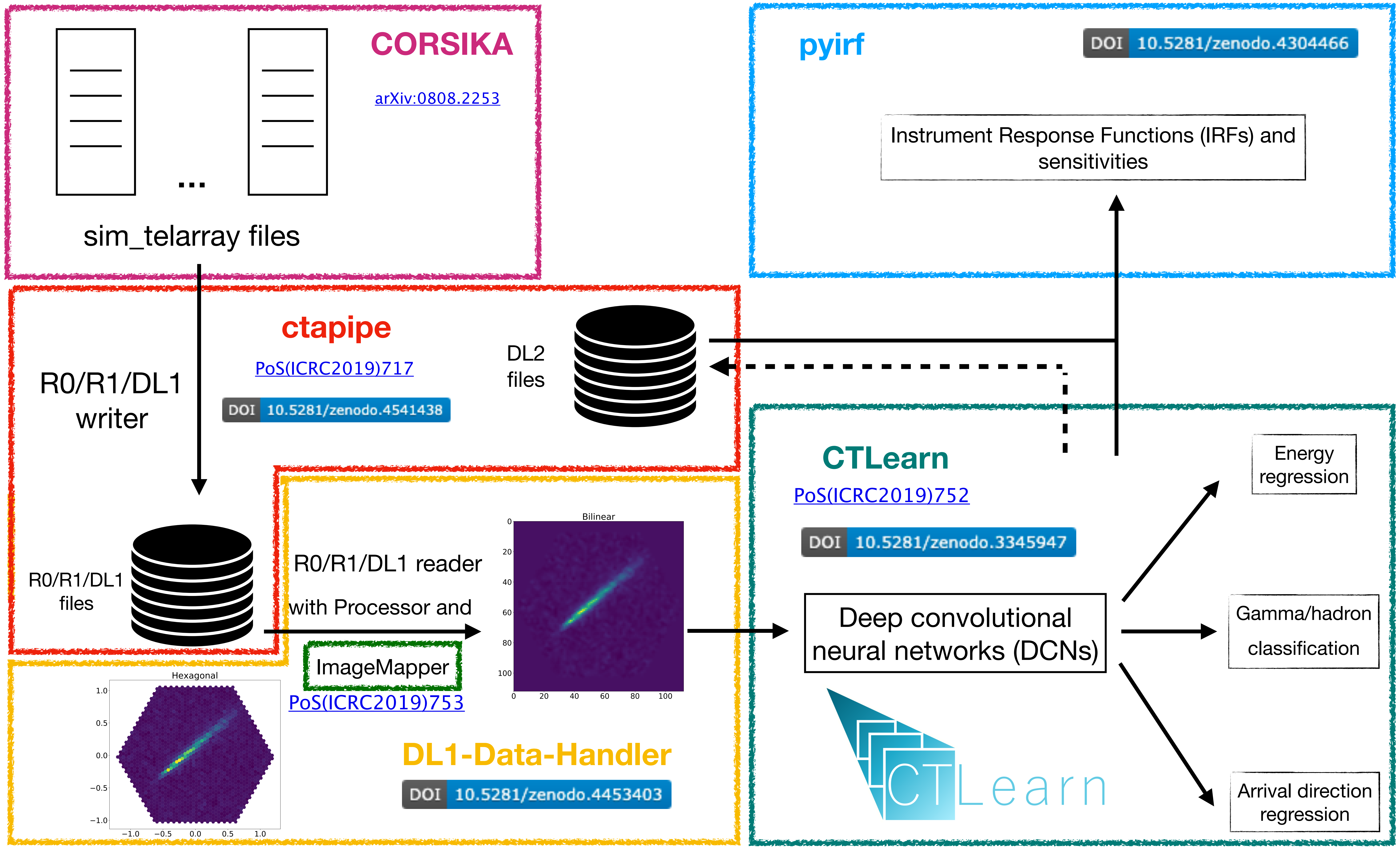
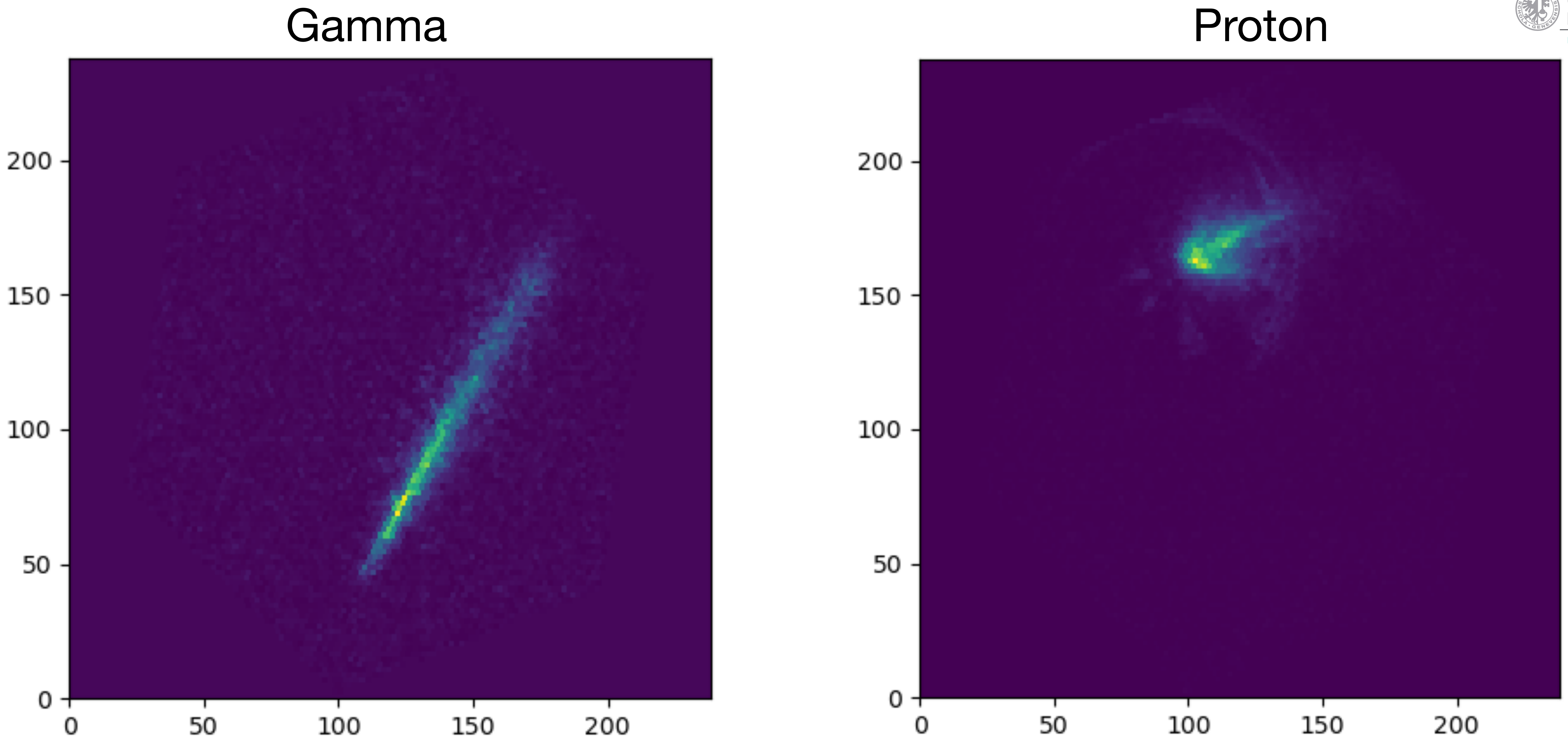


Image courtesy P. Grespan

CTA analysis workflow with CTLearn

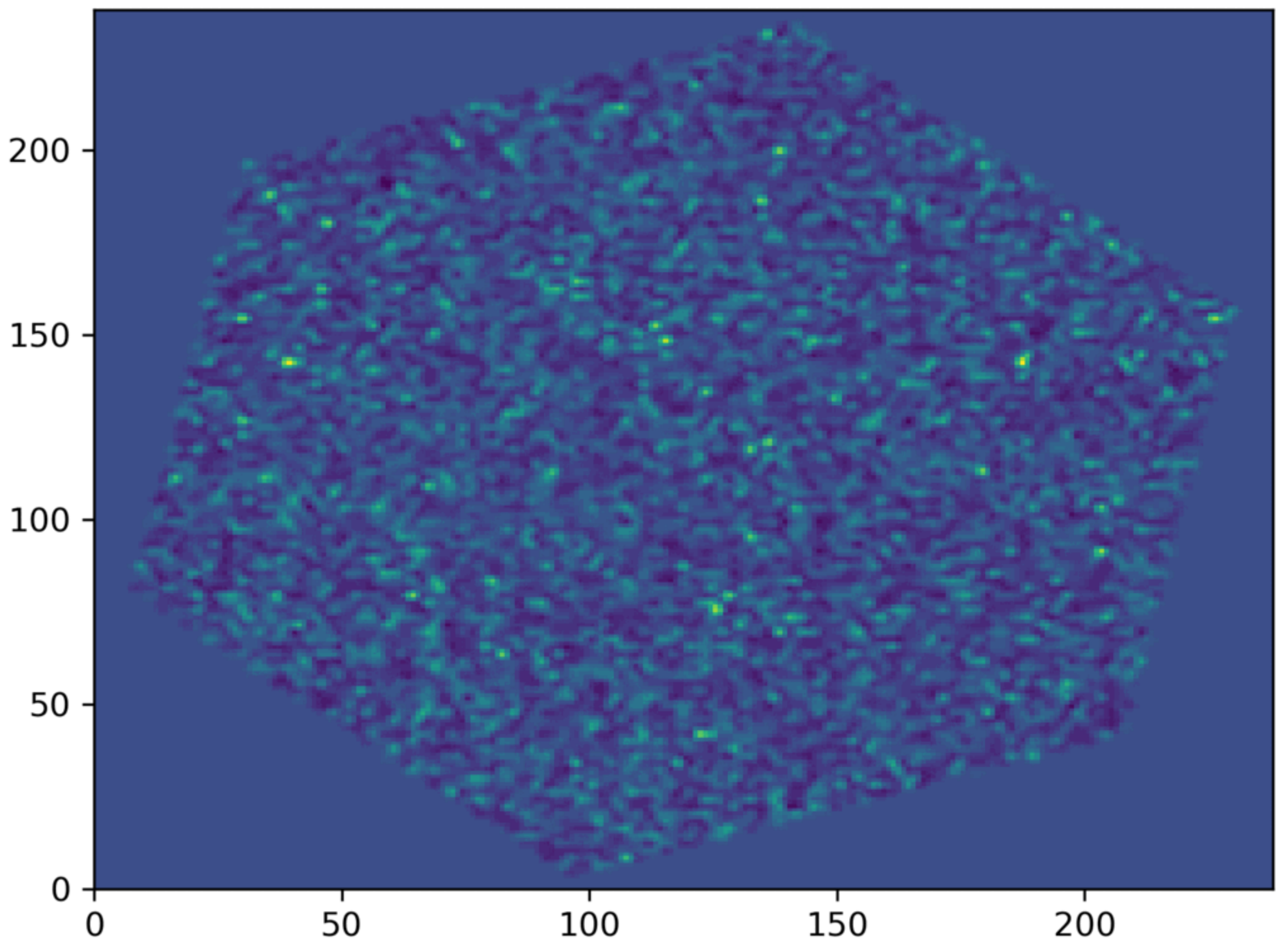


Shower images captured by the adv. LSTSiPM camera through the ImageMapper

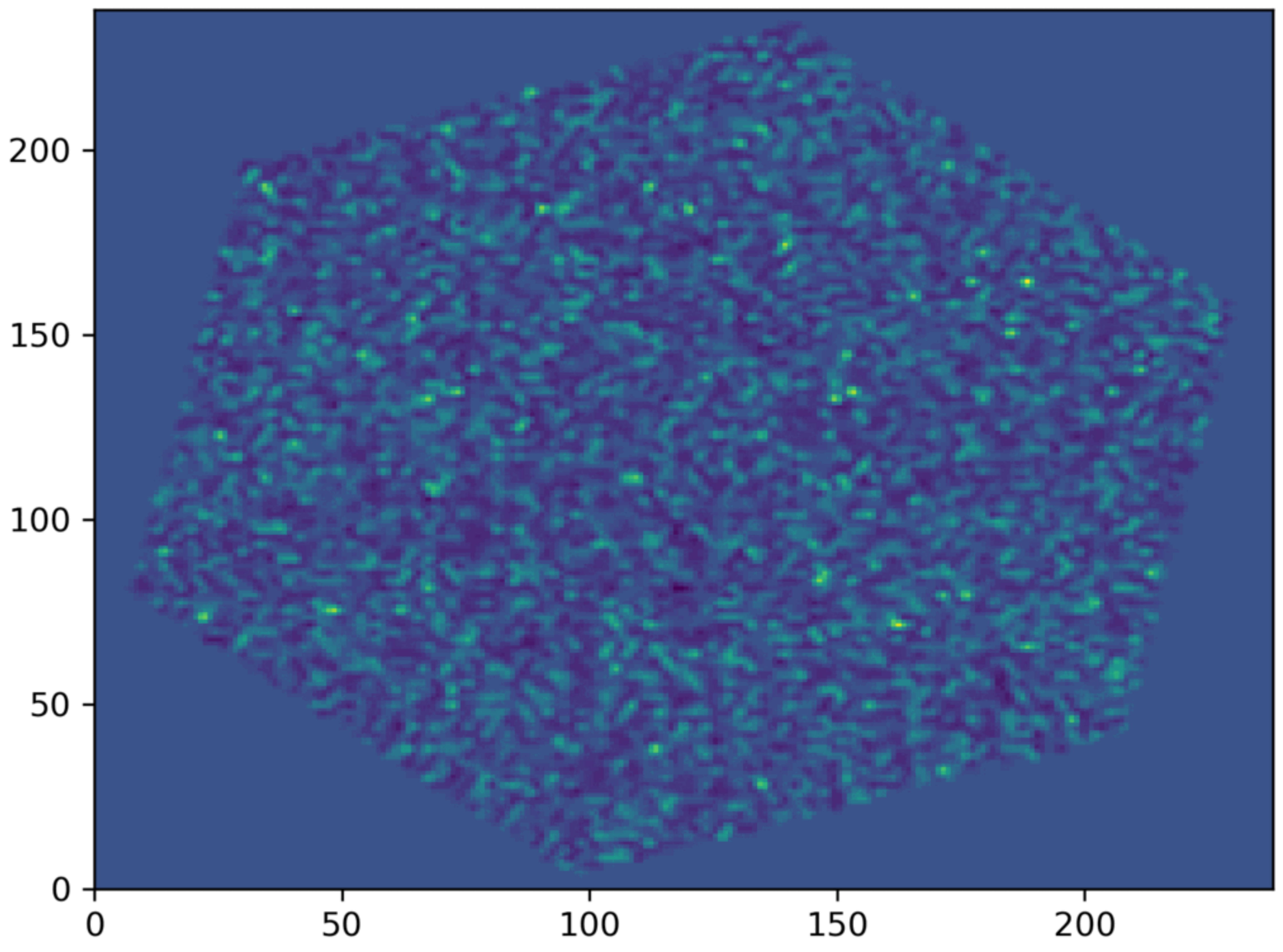


Calibrated waveforms captured by the adv. LSTSiPM camera through the ImageMapper

Gamma



Proton

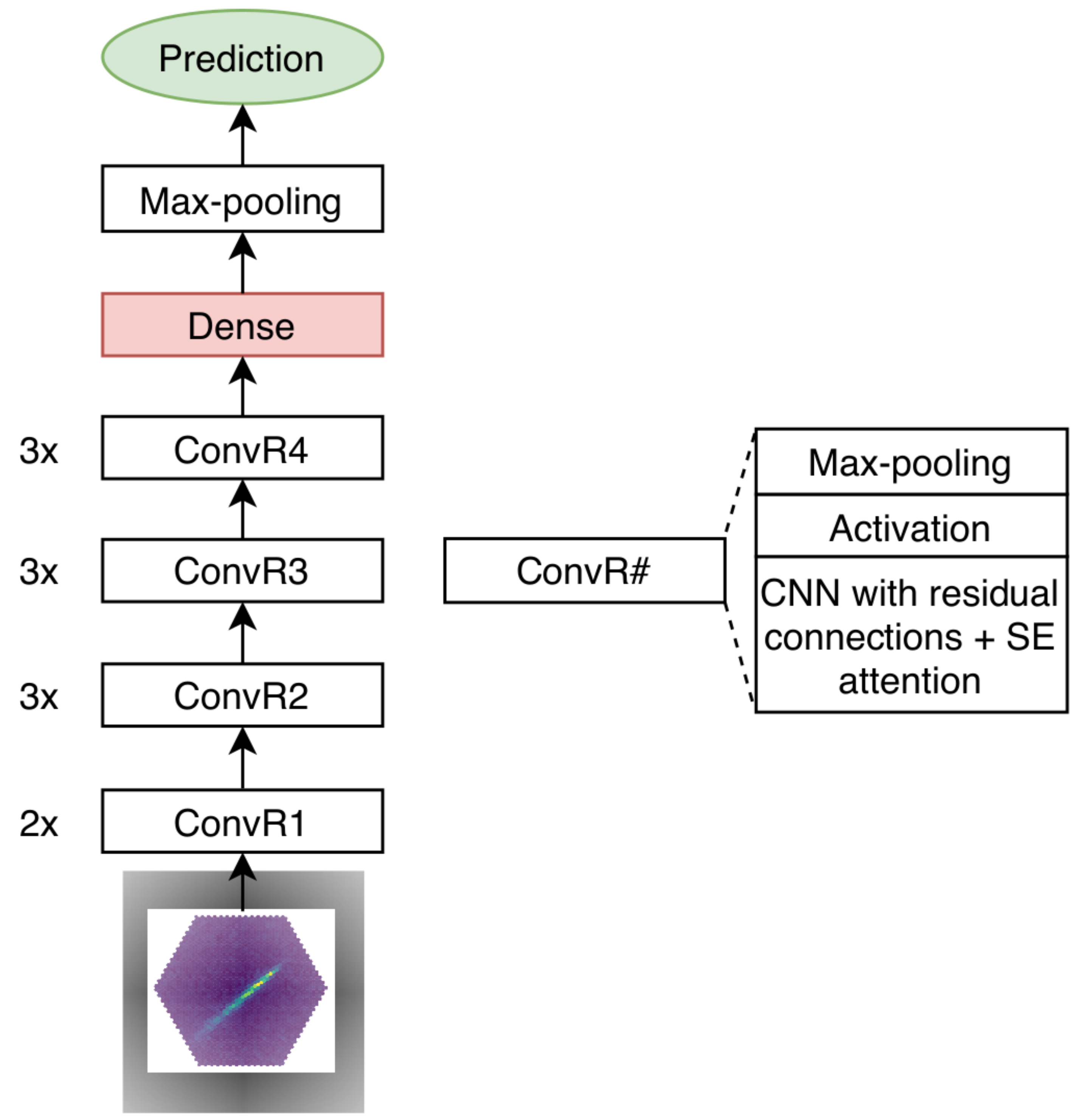


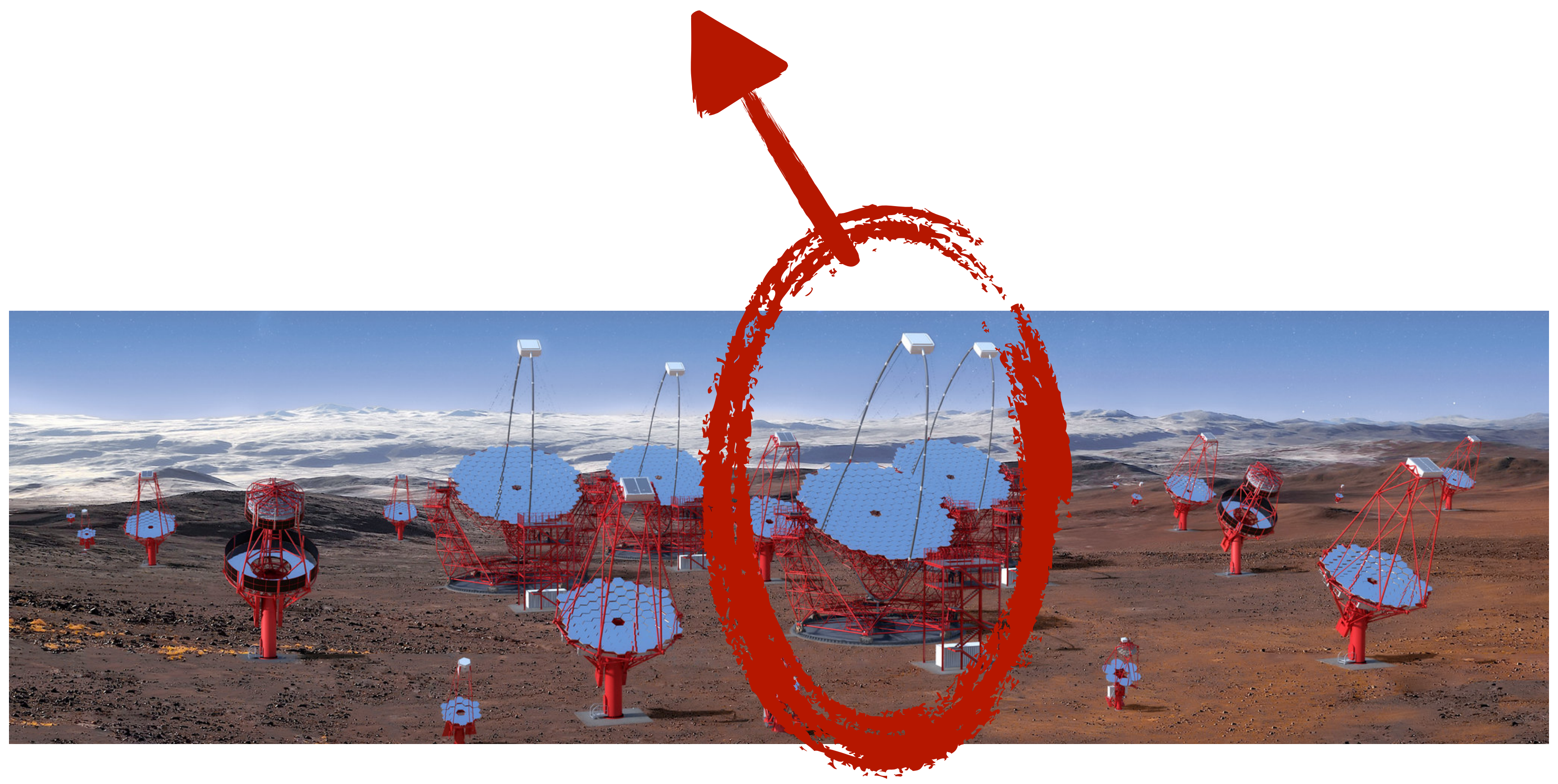
T. Miener et al.

Thin-ResNet (default monoscopic model in CTLearn)

34 layers in total

TRN model

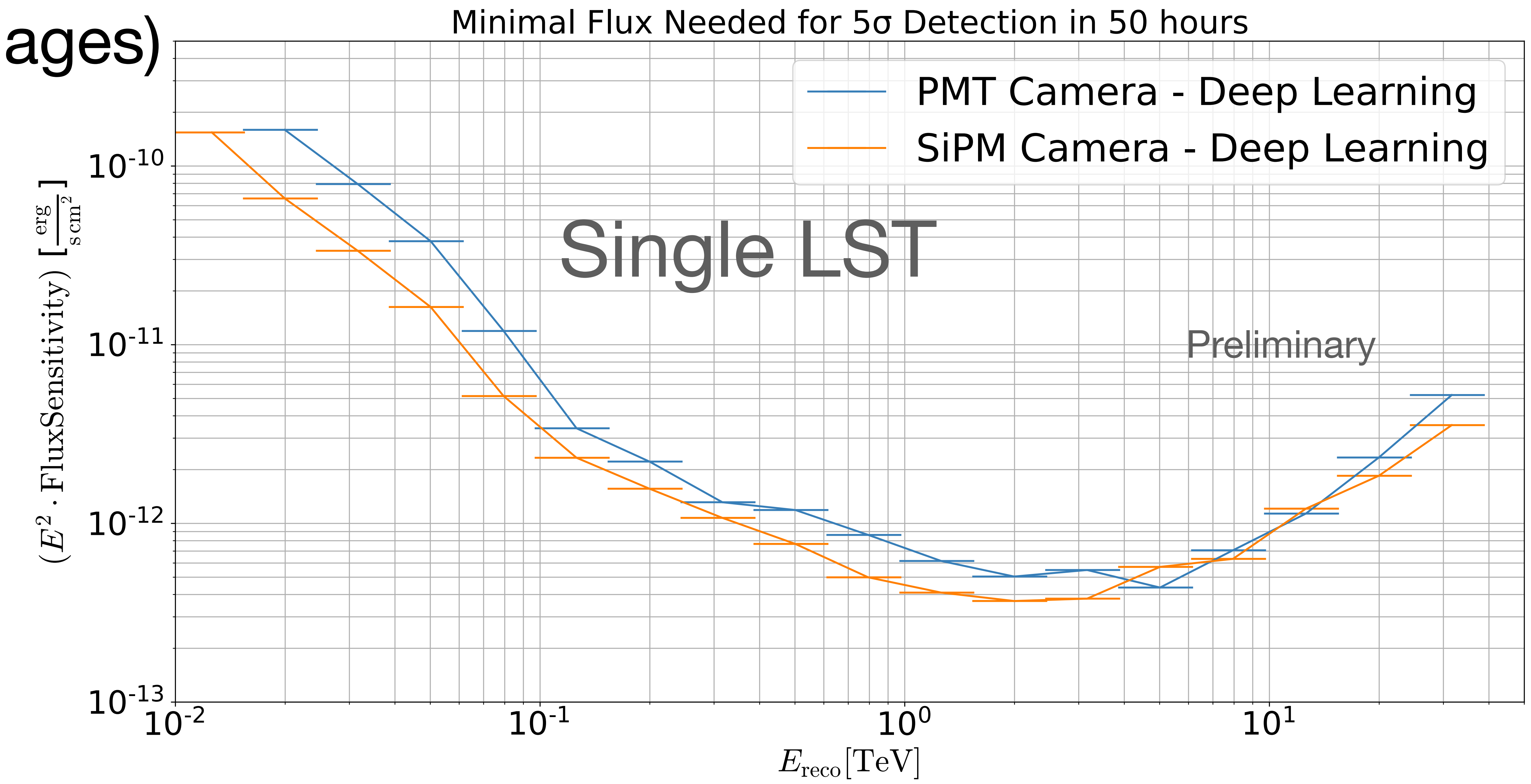


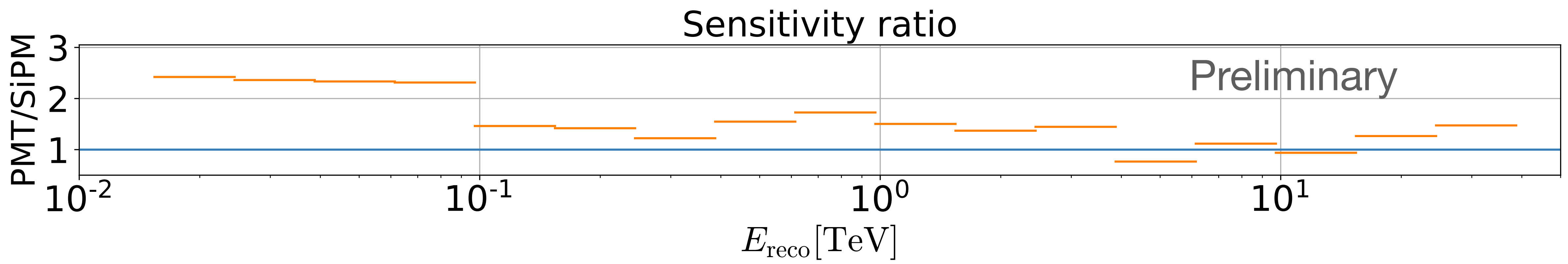


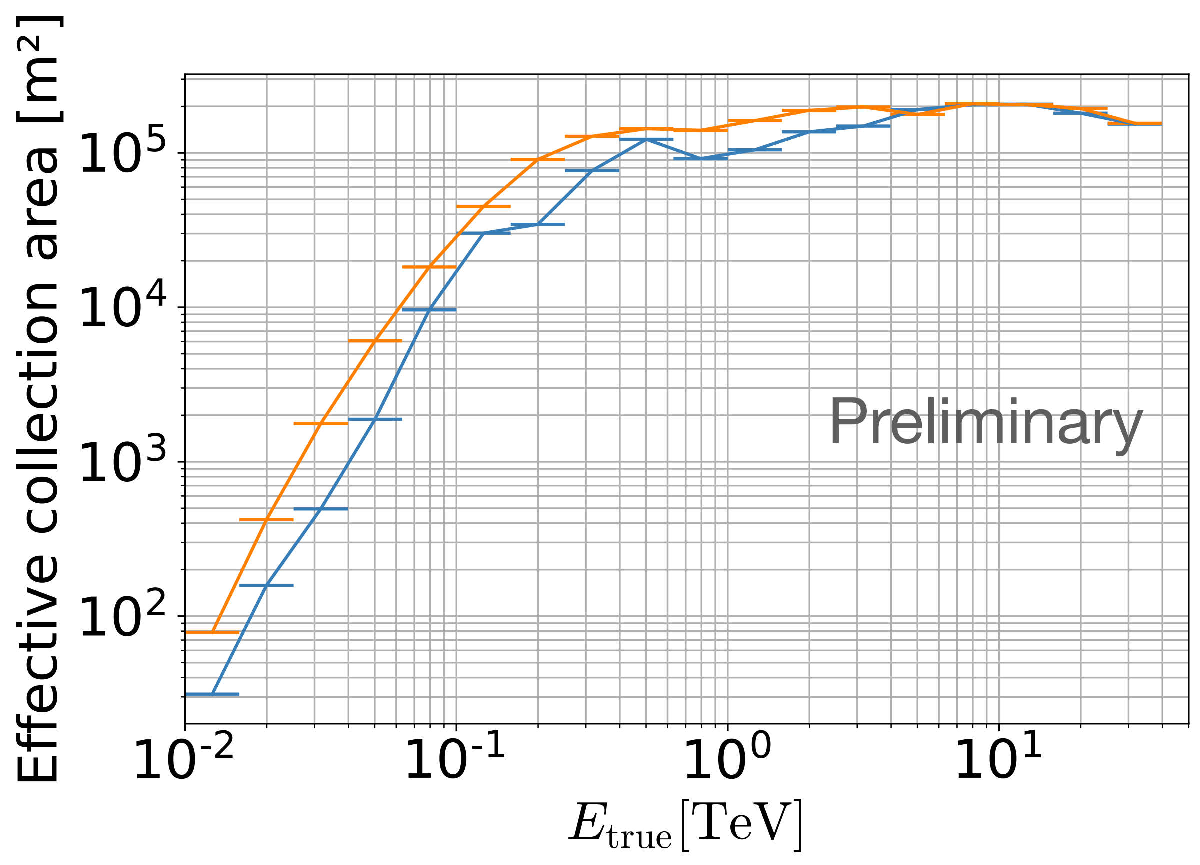
T. Miener et al.

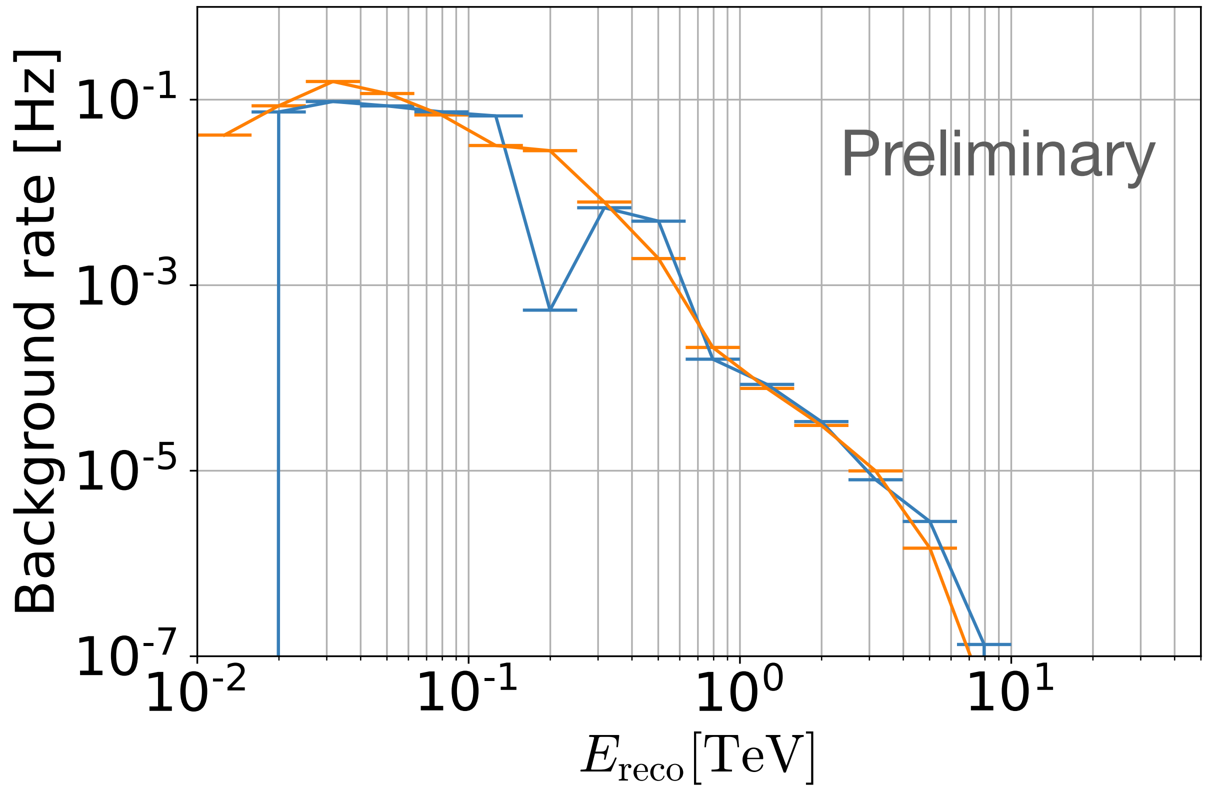
Deep Learning: current PMT camera vs high-resolution SiPM camera

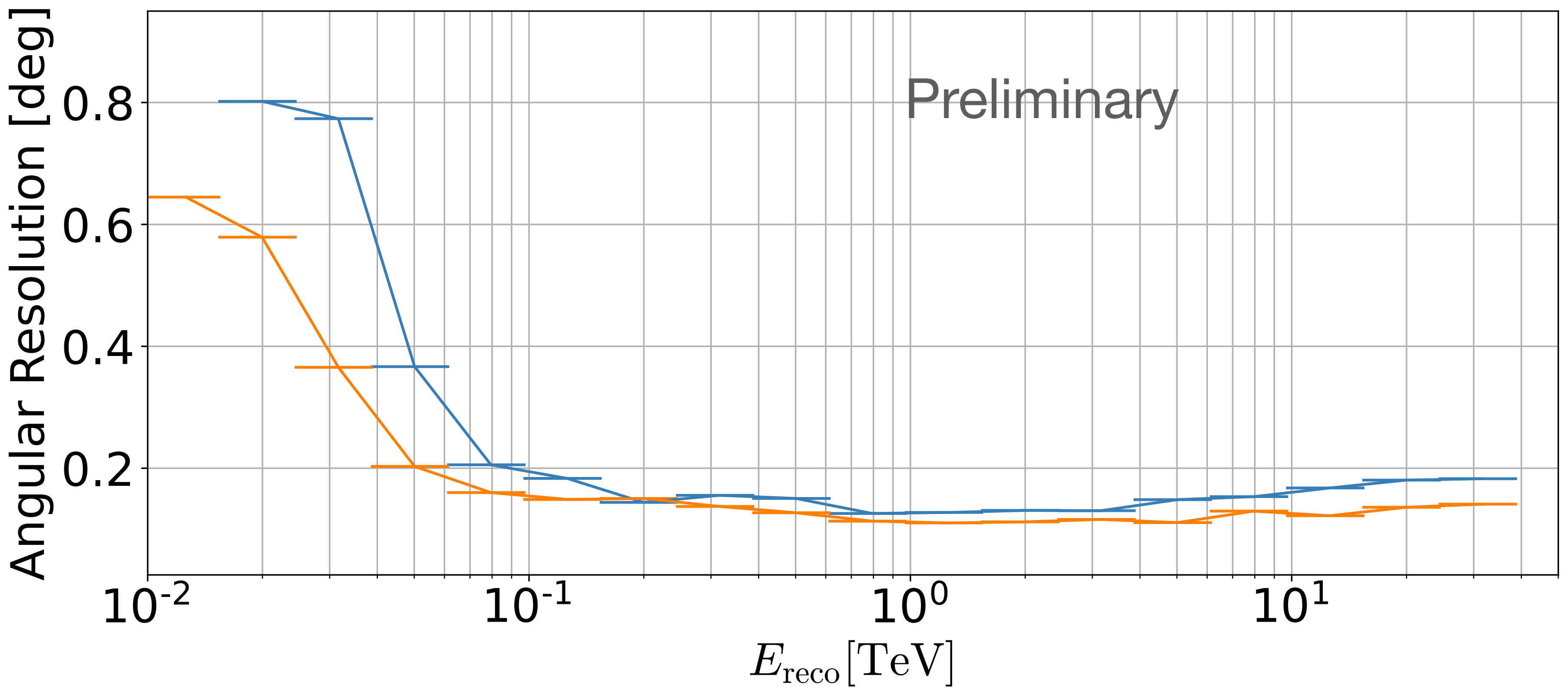
(cleaned images)

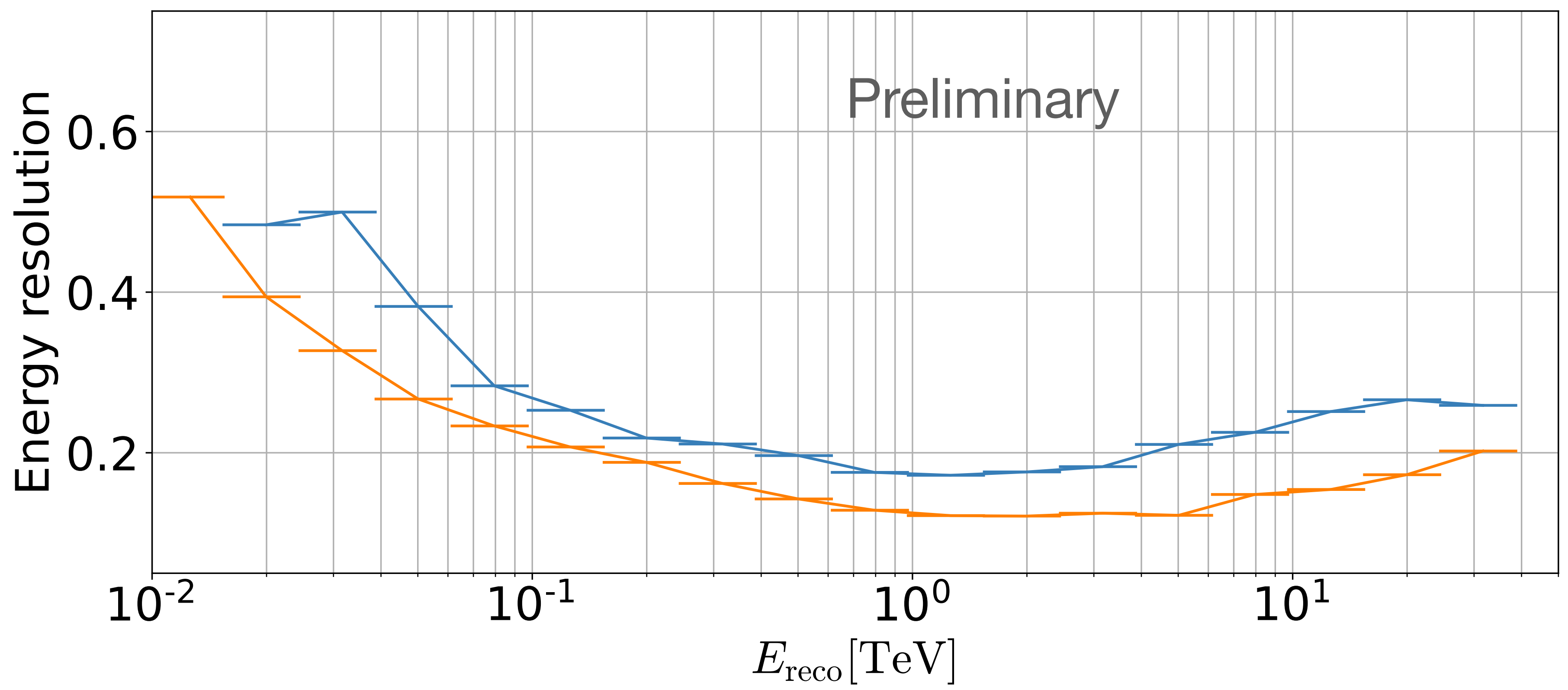






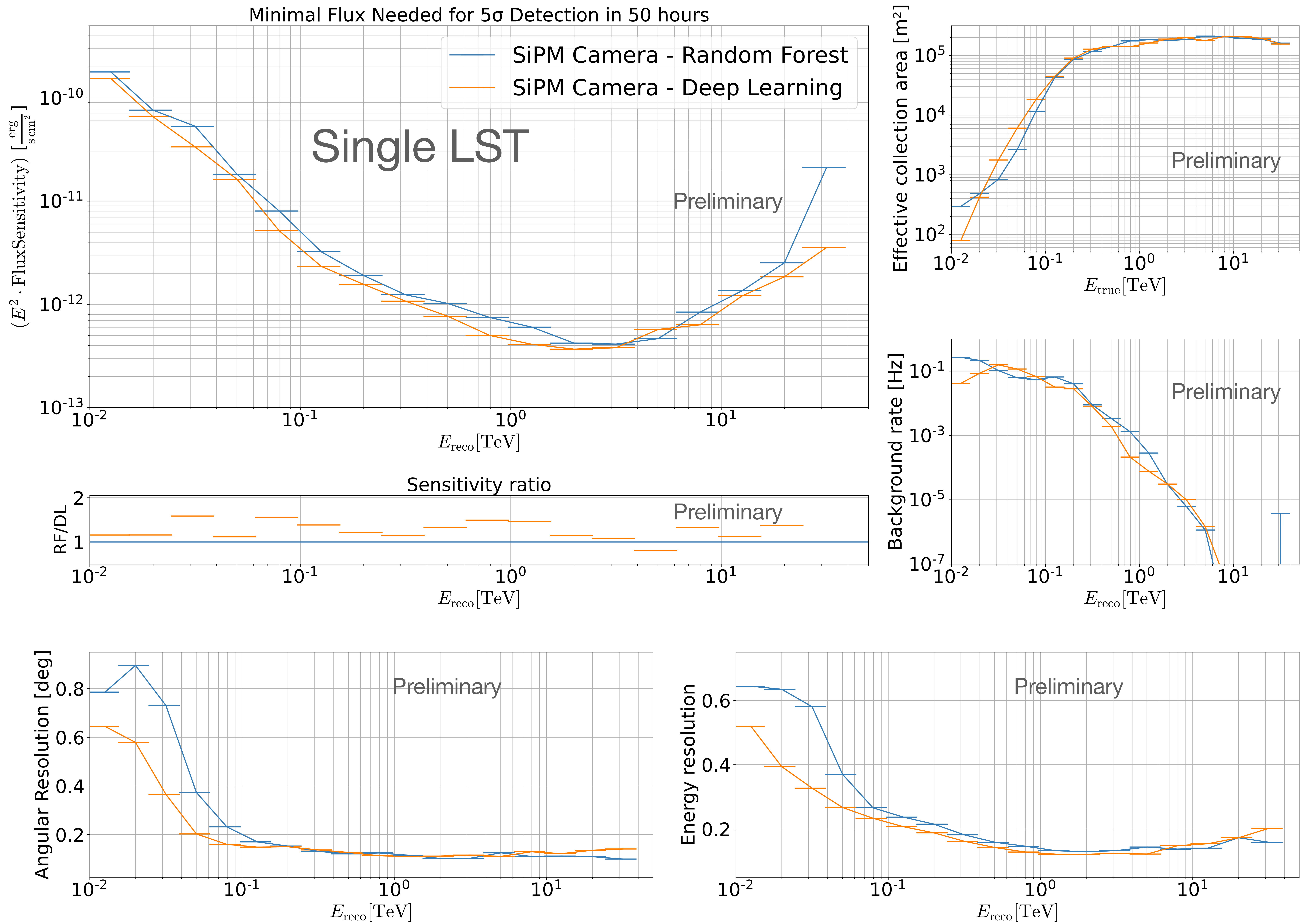






T. Miener et al.

High-resolution SiPM camera: Random Forest vs Deep Learning



T. Miener et al.

Trigger architecture of the adv. LSTSiPM camera

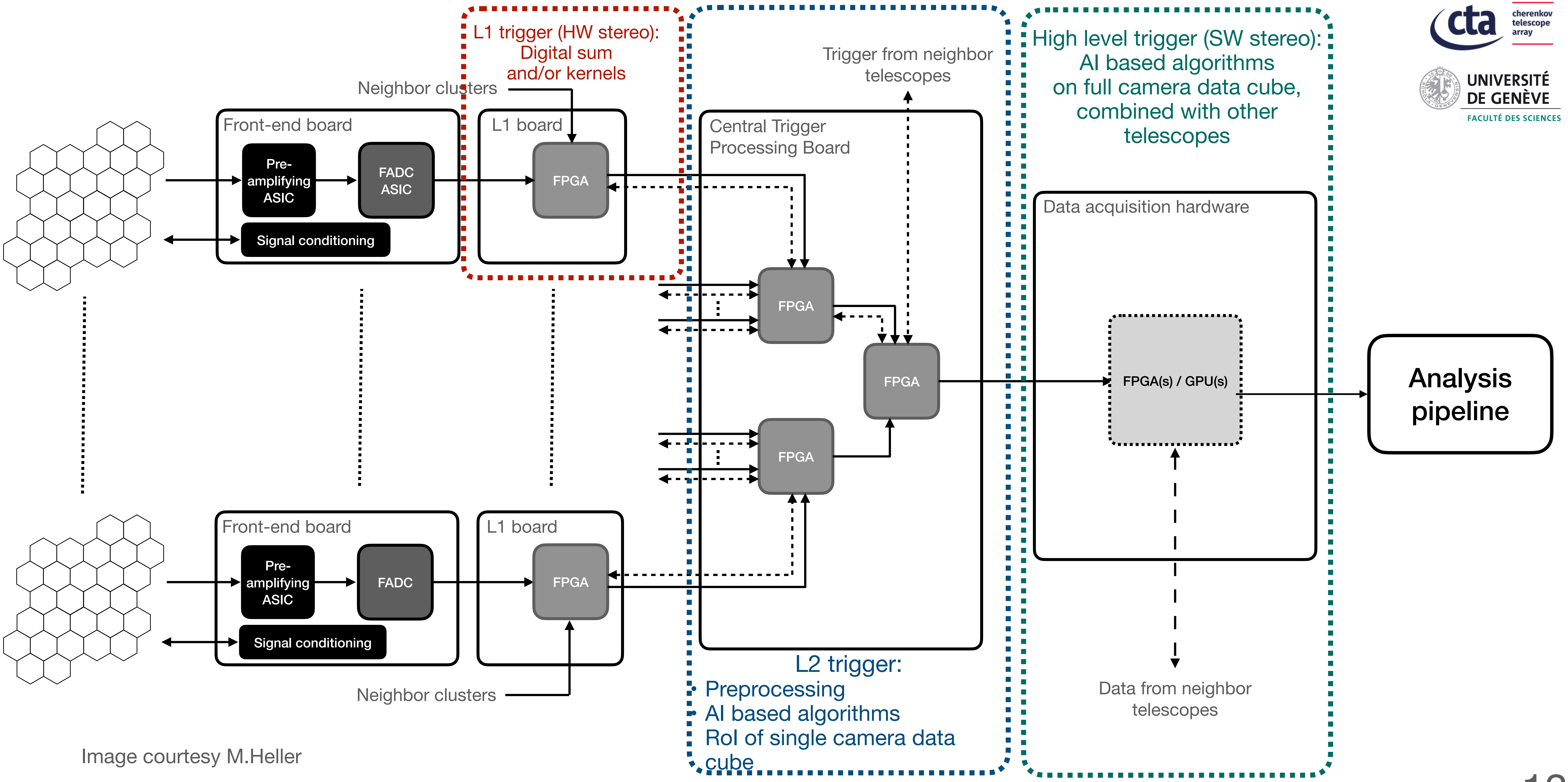
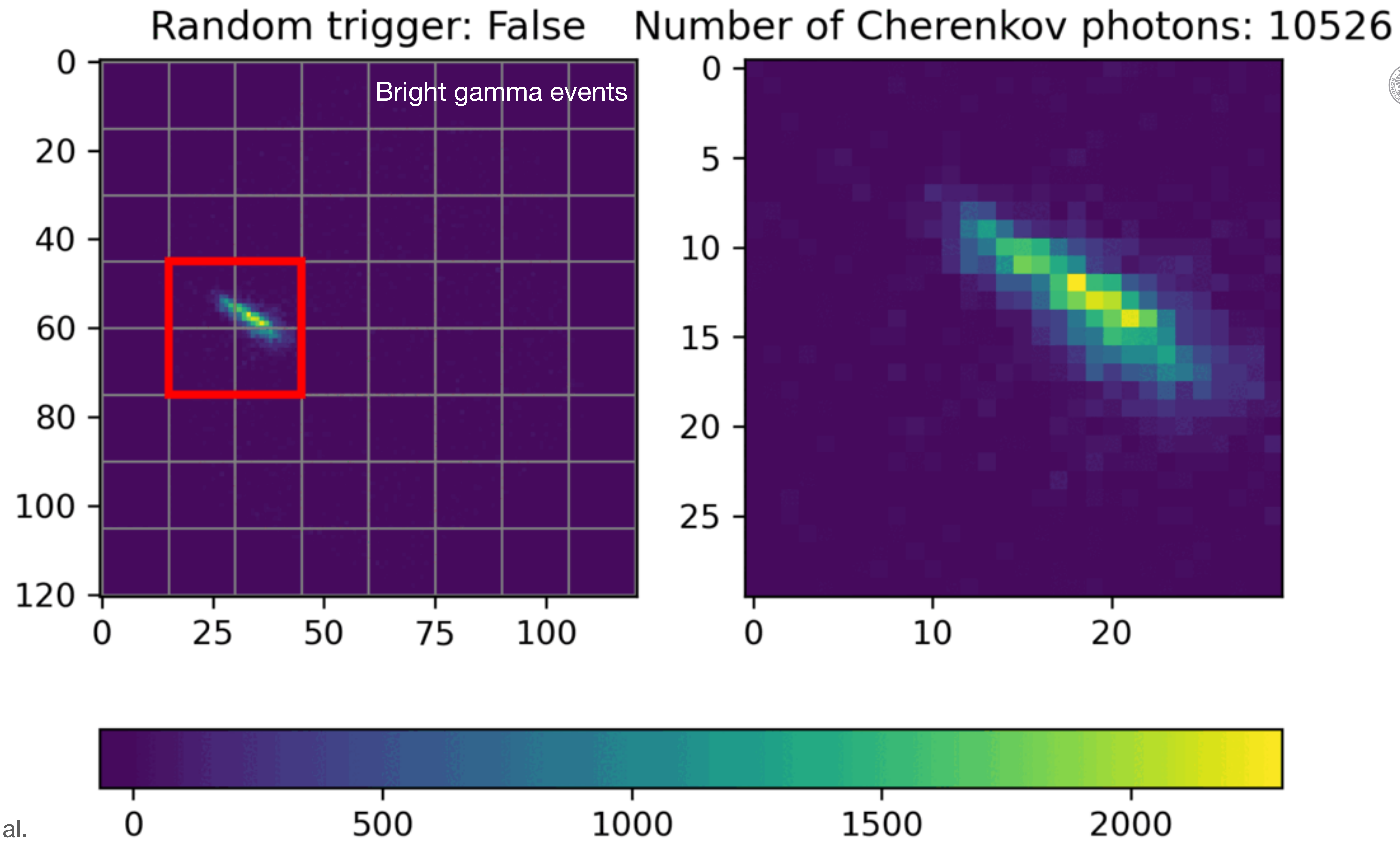


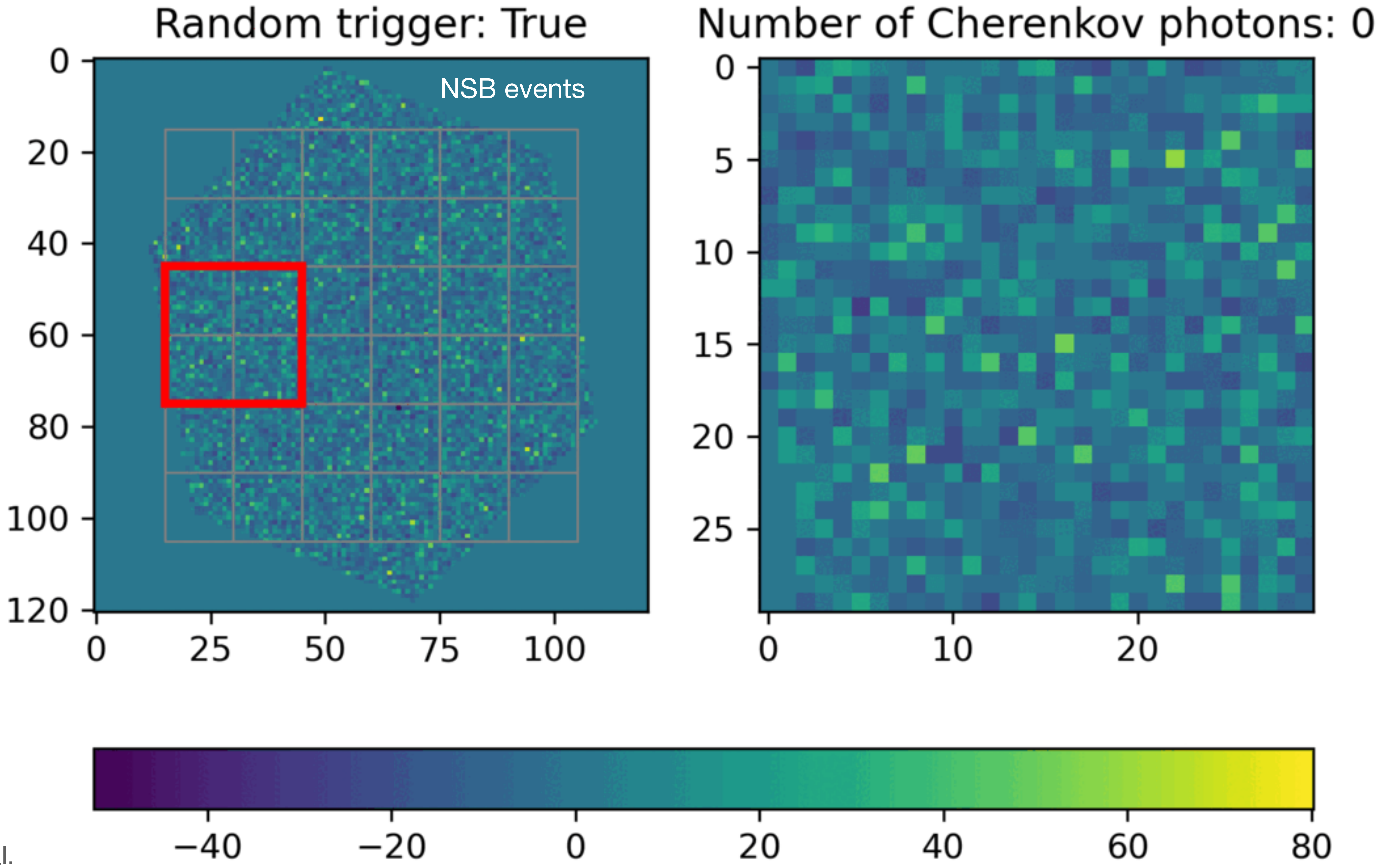
Image courtesy M.Heller

AI Trigger system of the adv. LSTSiPM camera



T. Miener et al.

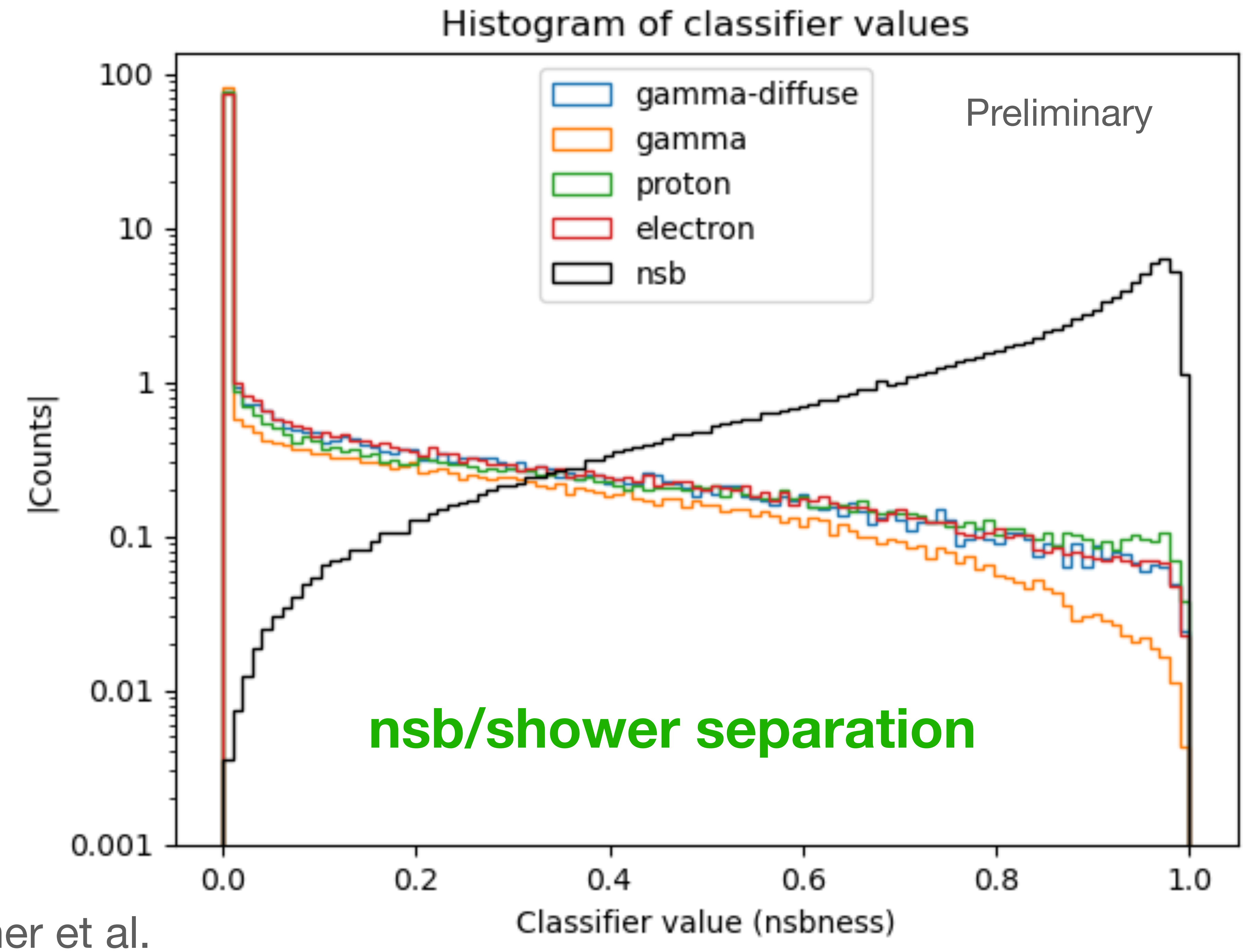
AI Trigger system of the adv. LSTSiPM camera

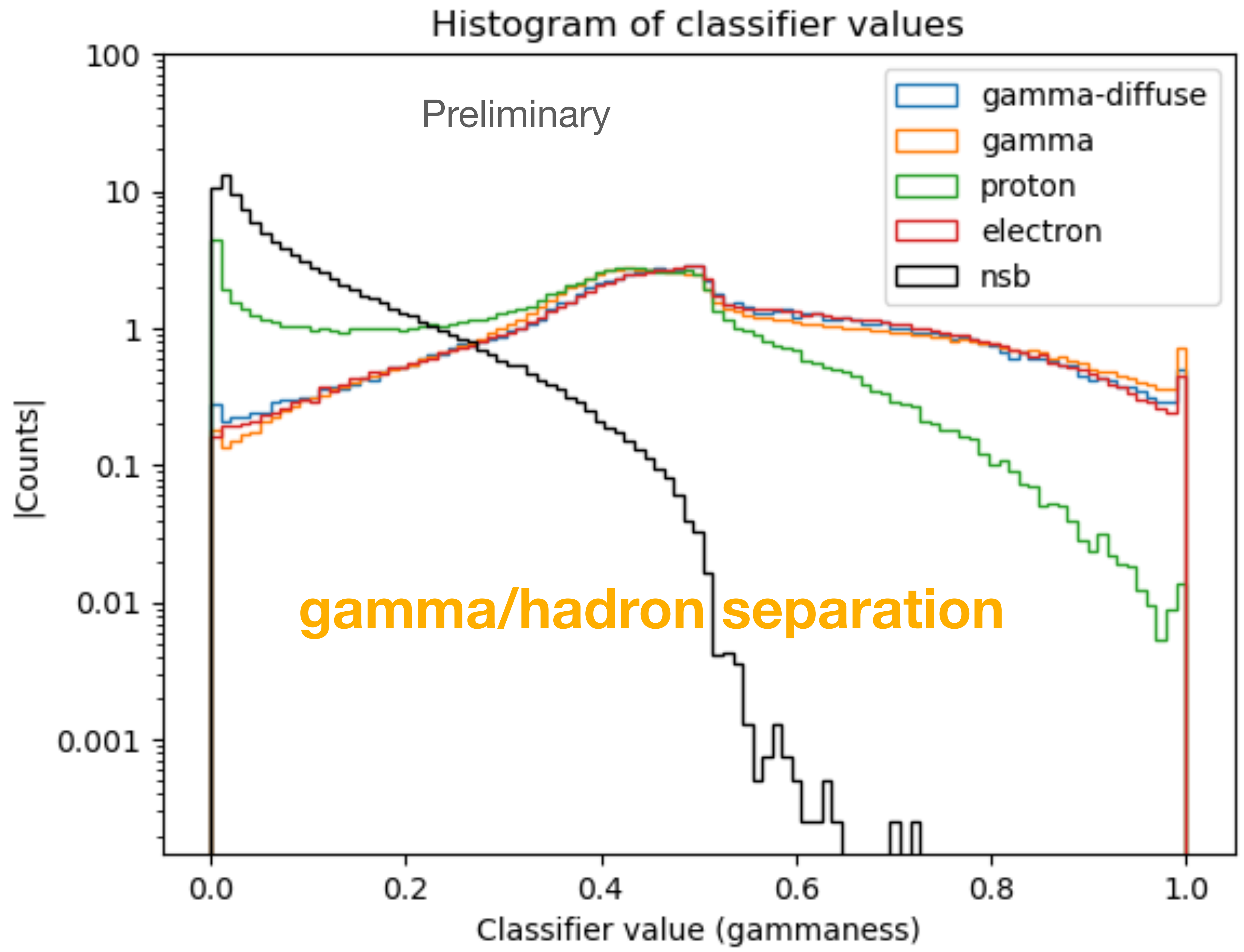


AI Trigger system of the adv. LSTSiPM camera

- **Categorial classification task** is performed including a ‘new class’ for **NSB patches**. Later on it is possible to add muons at training phase, which allows to do **muon tagging**.

- A **non-complex** plain CNN-based model is used that potential can run on the **FPGA** in the **CTP board**.





T. Miener et al.

CTLearn at CSCS

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

- All crucial MC simulation datasets including a trigger-less dataset are available at CSCS.
- CTLearn software and dependencies successfully installed and deployed on the CSCS cluster.
- Different projects for the various applications: cta03 for ML-related studies with LST and cta04 for the AI Trigger system of the adv. SiPM camera.
- New project regarding the processing on real observational data from the LST-1 prototype using CTLearn is submitted. Common interests with the DPPS for processing DVR-DL0 data offline.

Conclusion & outlook

- **CNN**-based methods show **superior** results for the full-event reconstruction with the high-resolution advanced SiPM cameras. **Performance studies in progress**.
- Preliminary studies with **waveforms** (shower developments) is promising and open the doors for various new applications such as the **AI Trigger system**.
- On going effort on **porting** those algorithms on **FPGA**s to run them **real-time** inside the cameras.
- Application on **real observational data** from the SST1M (stereo) and LST-1 (mono) prototypes is the next step. Our focus will lay on working with **cleaned** images and waveforms (or DVR-DL0 data) which is typically more **robust** when applying those algorithms to real observational data.
- Utilizing graph neural networks (GNNs) for **stereoscopic** reconstruction for CTA subarrays in the future.

CTLearn

cta cherenkov telescope array

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

Merci pour votre attention!

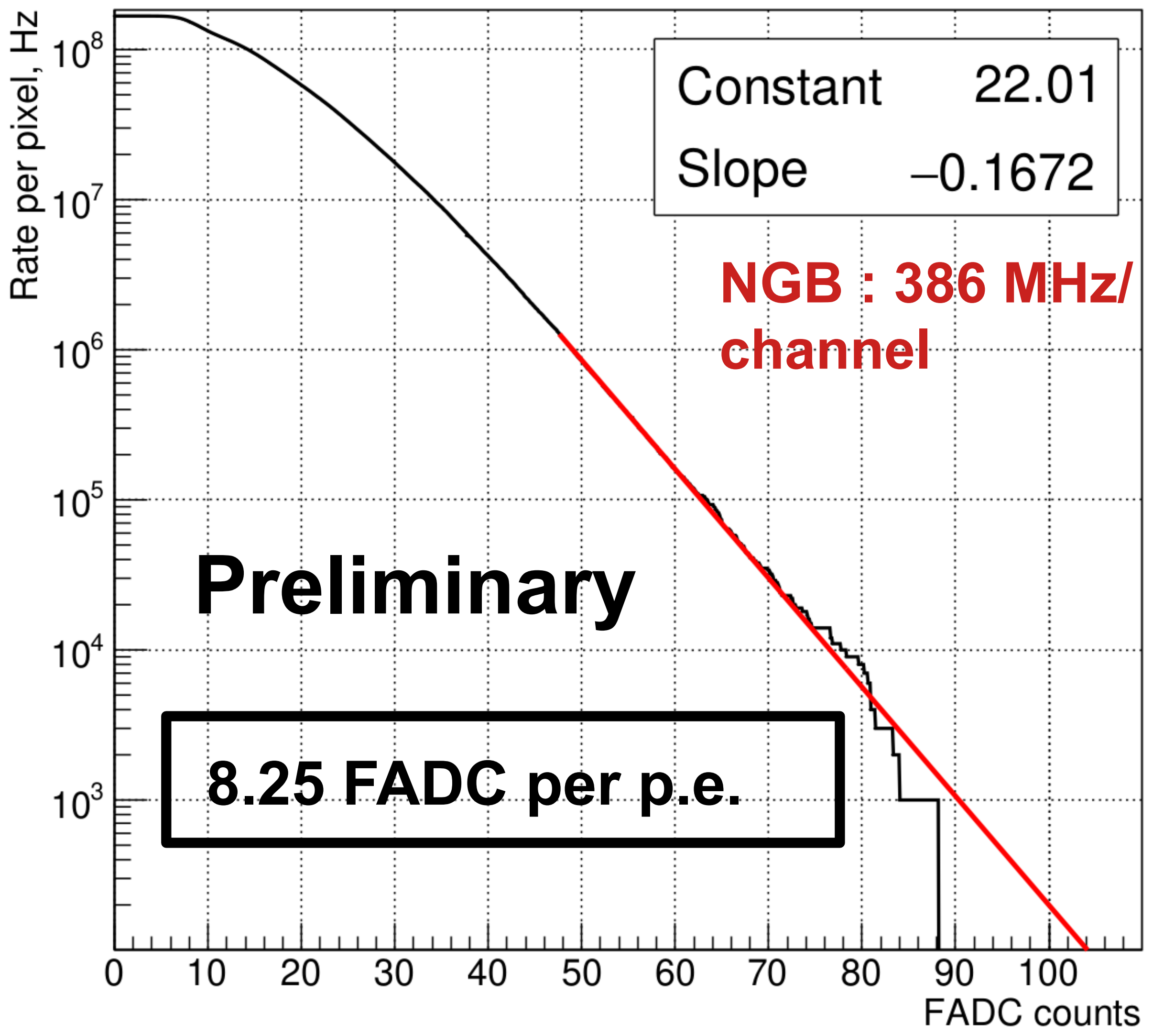
Backup

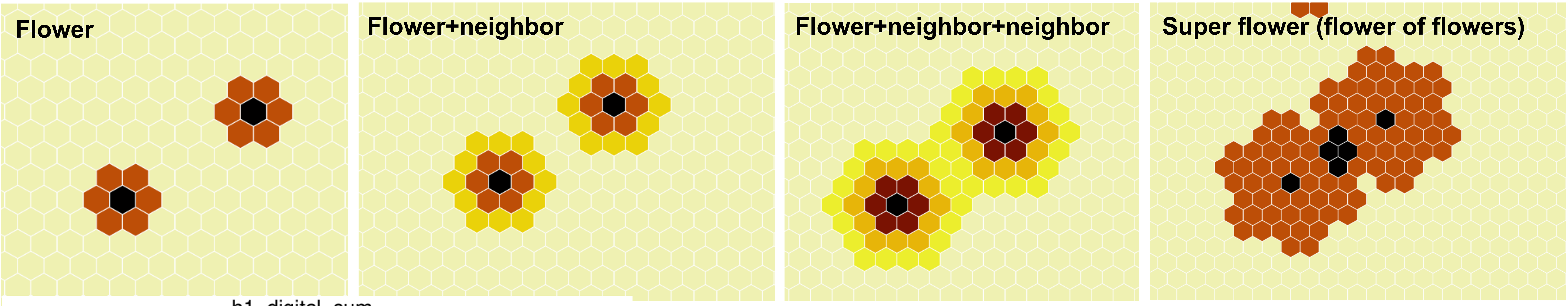
L1 Digital Sum Trigger

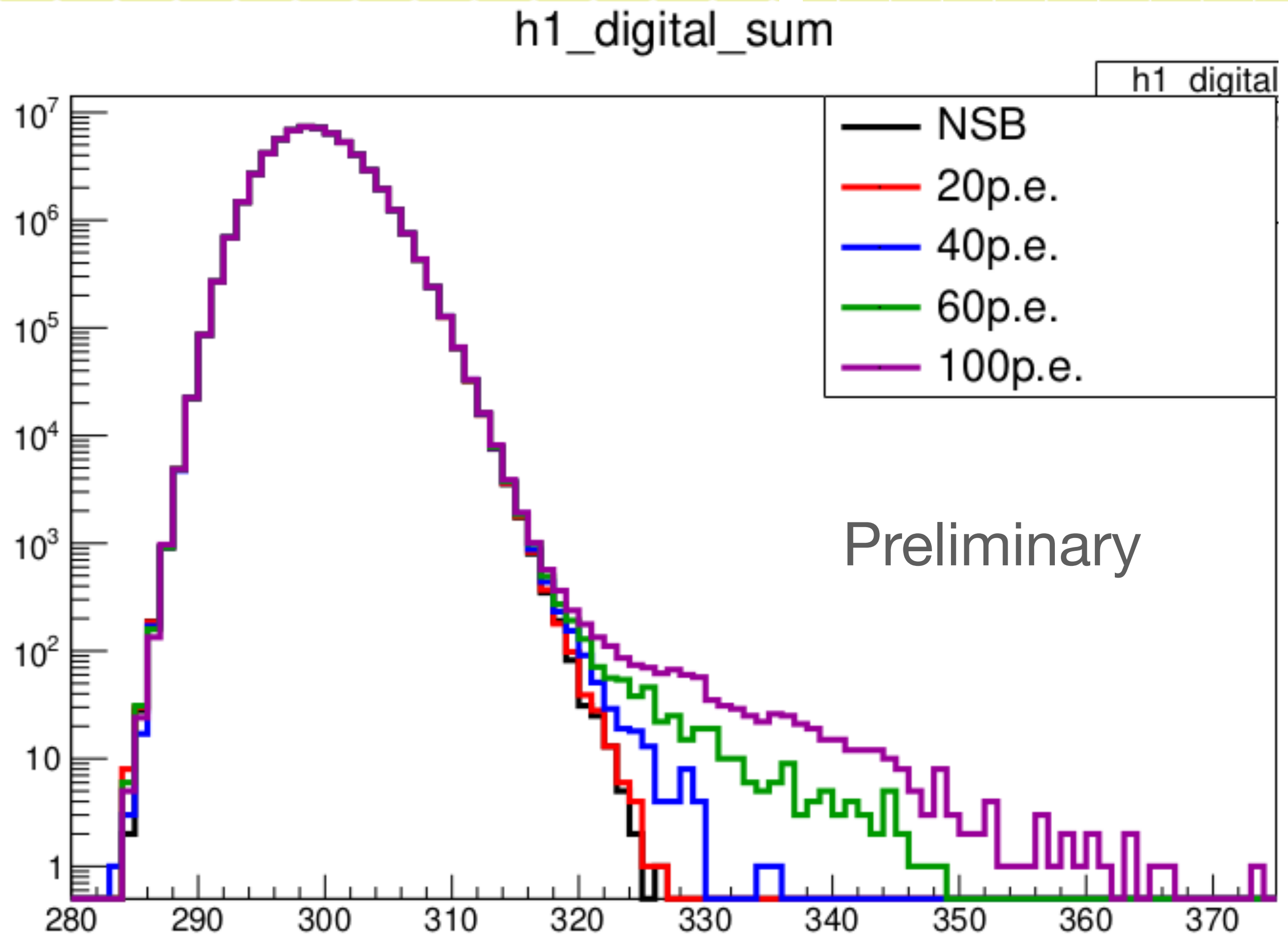
by Leonid Burmistrov (UniGe)

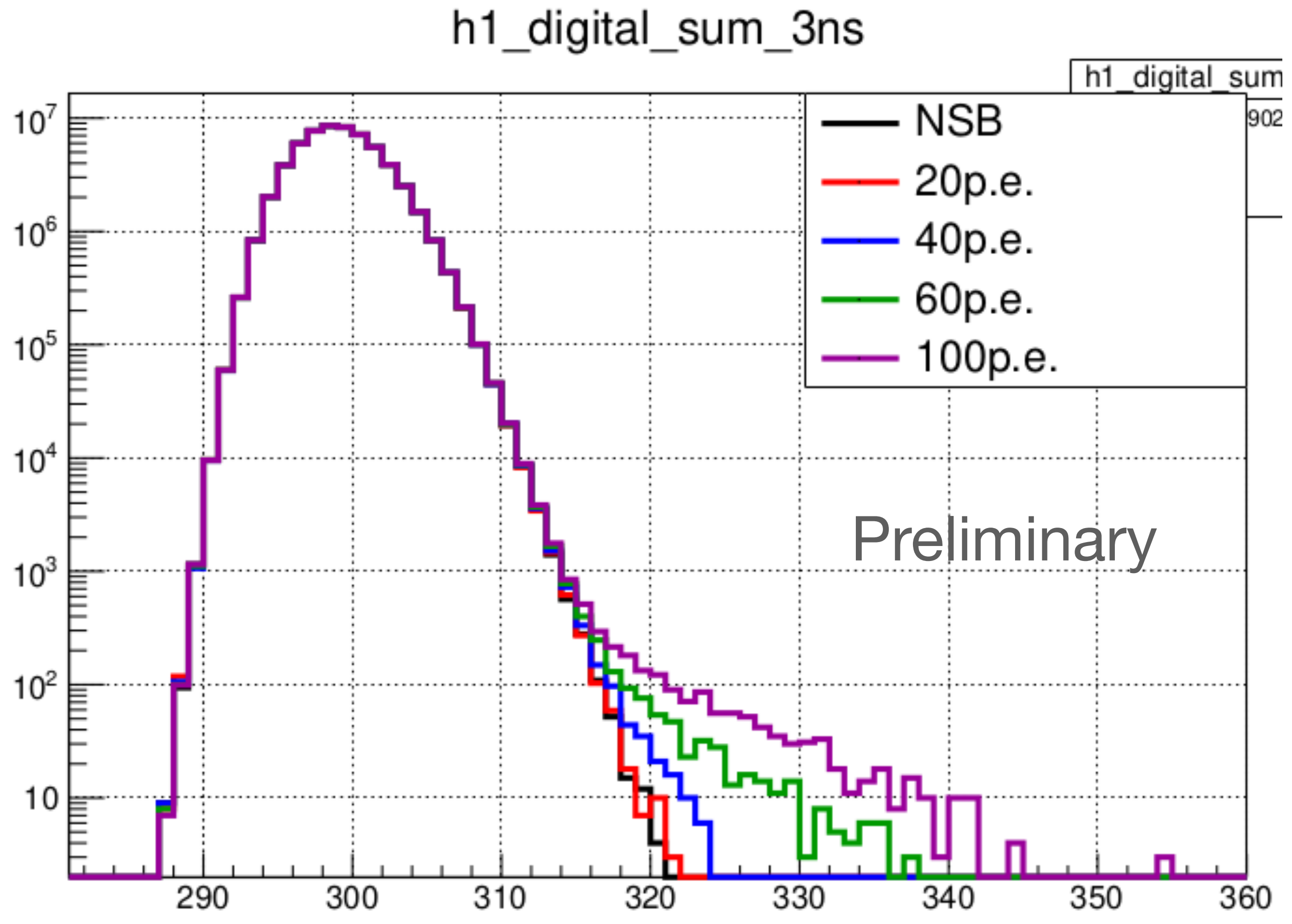
Motivation:

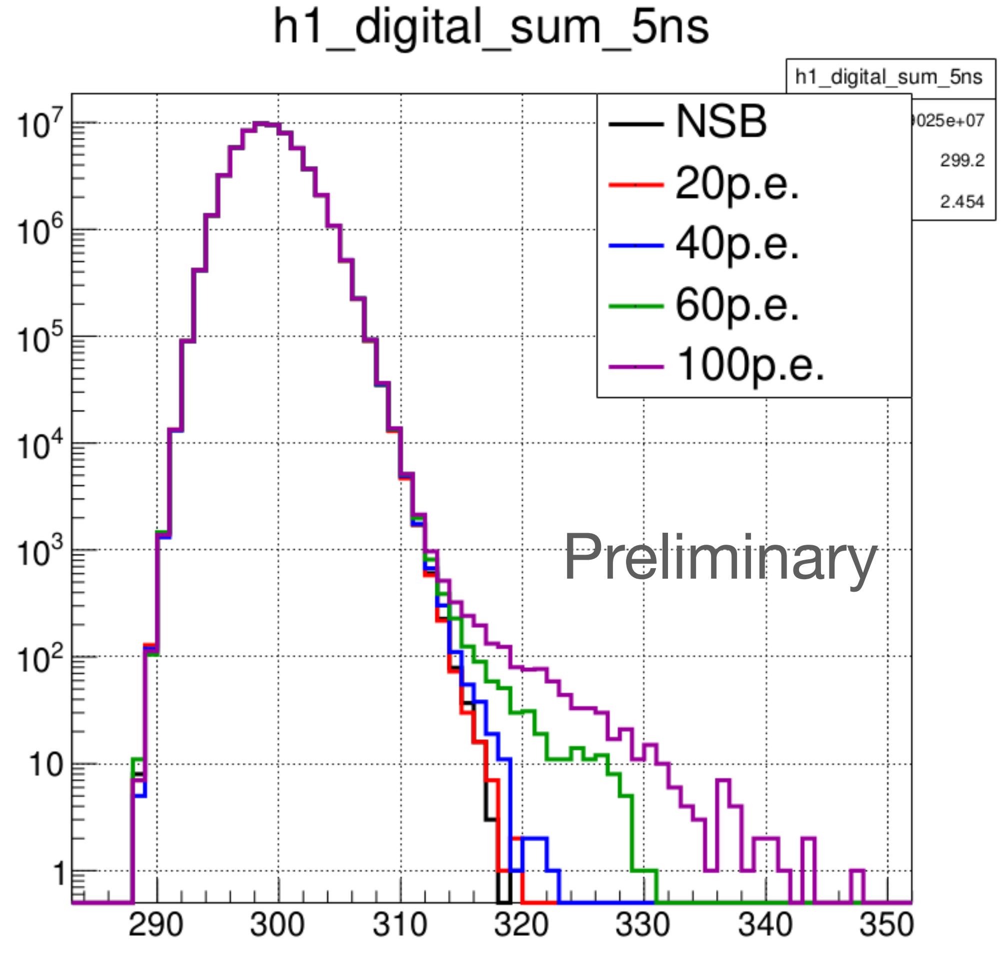
- Have better behaviour than threshold approach
- Build different clusters
- Integrate / differentiate over time samples
- Use different digital filters
- Use "coincidence" between clusters











Waveform-CNN-processing available in CTLearn

- https://github.com/cta-observatory/dl1-data-handler/pull/119 enables raw (R0) and calibrated (R1) waveform reading and preserved all previous DL1DH features (event-wise reading using generators, quality cuts, reading stereoscopic array-level data,…)

- https://github.com/cta-observatory/dl1-data-handler/pull/124 enables pixel-wise pedestal subtraction for raw (R0) waveform.

- https://github.com/ctlearn-project/ctlearn/pull/173 enables CNN-based models to process waveforms.

- PRs are all merged. New releases soon!

AI-based trigger system available soon

- https://github.com/cta-observatory/dl1-data-handler/pull/126 enables data reading of the AI-based trigger system with the ability to crop trigger patches.

- https://github.com/ctlearn-project/ctlearn/pull/180 enables the reconstruction of the true Cherenkov photons in a given trigger patch as a regression task. Not fully tested yet. Proof-of-concept needed.

- PRs are not 100% finalised. Some commits will follow.

Datasets - config settings (recommended by lstchain for PMT & Matthieu for SiPM)

PMT

```
    "CameraCalibrator": {
        "image_extractor_type": "LocalPeakWindowSum"
    },
    "ImageProcessor": {
        "image_cleaner_type": "TailcutsImageCleaner",
        "TailcutsImageCleaner": {
            "picture_threshold_pe": [
                ["type", "*", 8.0]
            ],
            "boundary_threshold_pe": [
                ["type", "*", 4]
            ],
            "min_picture_neighbors": [["type", "*", 2]]
        },
        "ImageQualityQuery": {
            "quality_criteria": [
                ["enough_pixels", "np.count_nonzero(image) > 2"],
                ["enough_charge", "image.sum() > 50"]
            ]
        }
    },
```

SiPM

```
    },
    "CameraCalibrator": {
        "image_extractor_type": "NeighborPeakWindowSum"
    },
    "ImageProcessor": {
        "image_cleaner_type": "TailcutsImageCleaner",
        "TailcutsImageCleaner": {
            "picture_threshold_pe": [
                ["type", "*", 5.0]
            ],
            "boundary_threshold_pe": [
                ["type", "*", 3]
            ],
            "min_picture_neighbors": [["type", "*", 2]]
        },
        "ImageQualityQuery": {
            "quality_criteria": [
                ["enough_pixels", "np.count_nonzero(image) > 2"],
                ["enough_charge", "image.sum() > 50"]
            ]
        }
    },
```

RF ctapipe config settings (inspired by lstchain standard config)

```
# ============================================================================
# ctapipe-ml-train-particle-classifier config file.
# version: VERSION
#
# Configuration for training machine-learning models
# ============================================================================
TrainParticleClassifier:
  CrossValidator:
    n_cross_validations: 5

  ParticleClassifier:
    model_cls: ExtraTreesClassifier
    model_config:
      max_depth: 30
      min_samples_leaf: 10
      n_jobs: -1
      n_estimators: 100
      bootstrap: true
      criterion: "gini"
      max_features: 1.0
      max_leaf_nodes: null
      min_impurity_decrease: 0.0
      min_samples_split: 10
      min_weight_fraction_leaf: 0.0
      oob_score: false
      random_state: 42
      verbose: 0
      warm_start: false
      class_weight: null
```

```
    features:
      - log_hillas_intensity
      - hillas_fov_lon
      - hillas_fov_lat
      - hillas_length
      - hillas_width
      - hillas_skewness
      - hillas_kurtosis
      - area
      - timing_slope
      - leakage_intensity_width_2

    #stereo_combiner_cls: "StereoMeanCombiner"
    #StereoMeanCombiner:
    #  log_target: true

    QualityQuery:
      quality_criteria:
        #- ["HillasValid", "HillasReconstructor_is_valid"]
        - ["enough intensity", "hillas_intensity > 25"]
        - ["Positive width", "hillas_width > 0"]
        - ["enough pixels", "morphology_n_pixels > 3"]
        - ["not clipped", "leakage_intensity_width_1 < 0.2"]

    FeatureGenerator:
      features:
        - ["area", "hillas_width * hillas_length"]
        - ["log_hillas_intensity", "np.log(hillas_intensity)"]
```

DL1-DH I/O performance during training - CNN-based full-event reconstruction on calibrated waveforms with CTLearn

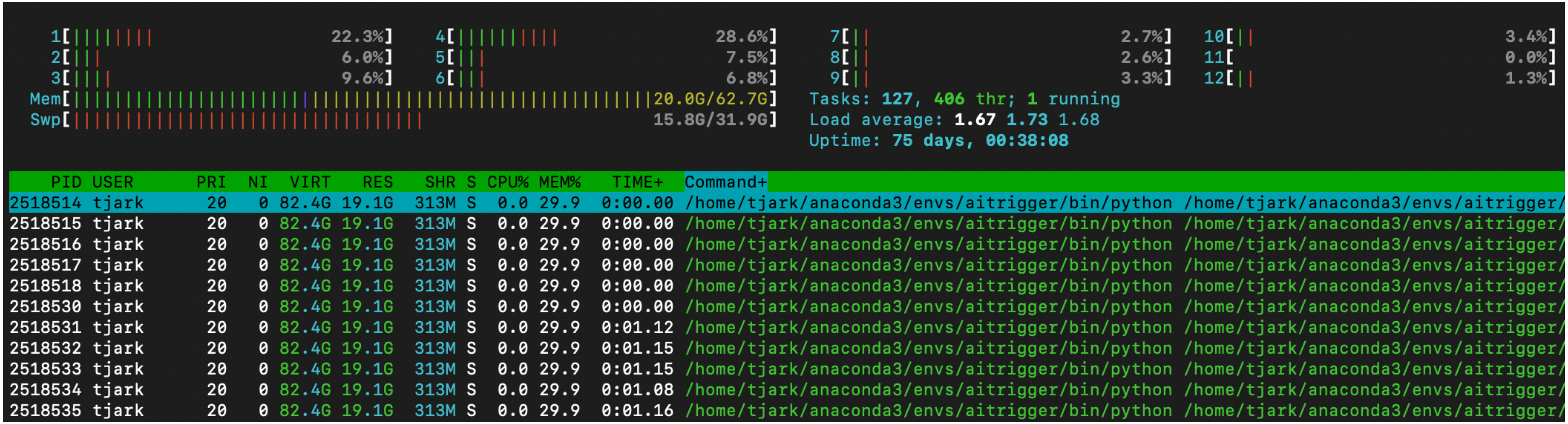
CTLearn

cta cherenkov telescope array

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

- Training with merged files is more efficient. We can now train with the full production (80 % reserved for the training set for gamma-diffuse and proton) without running into RAM issues.

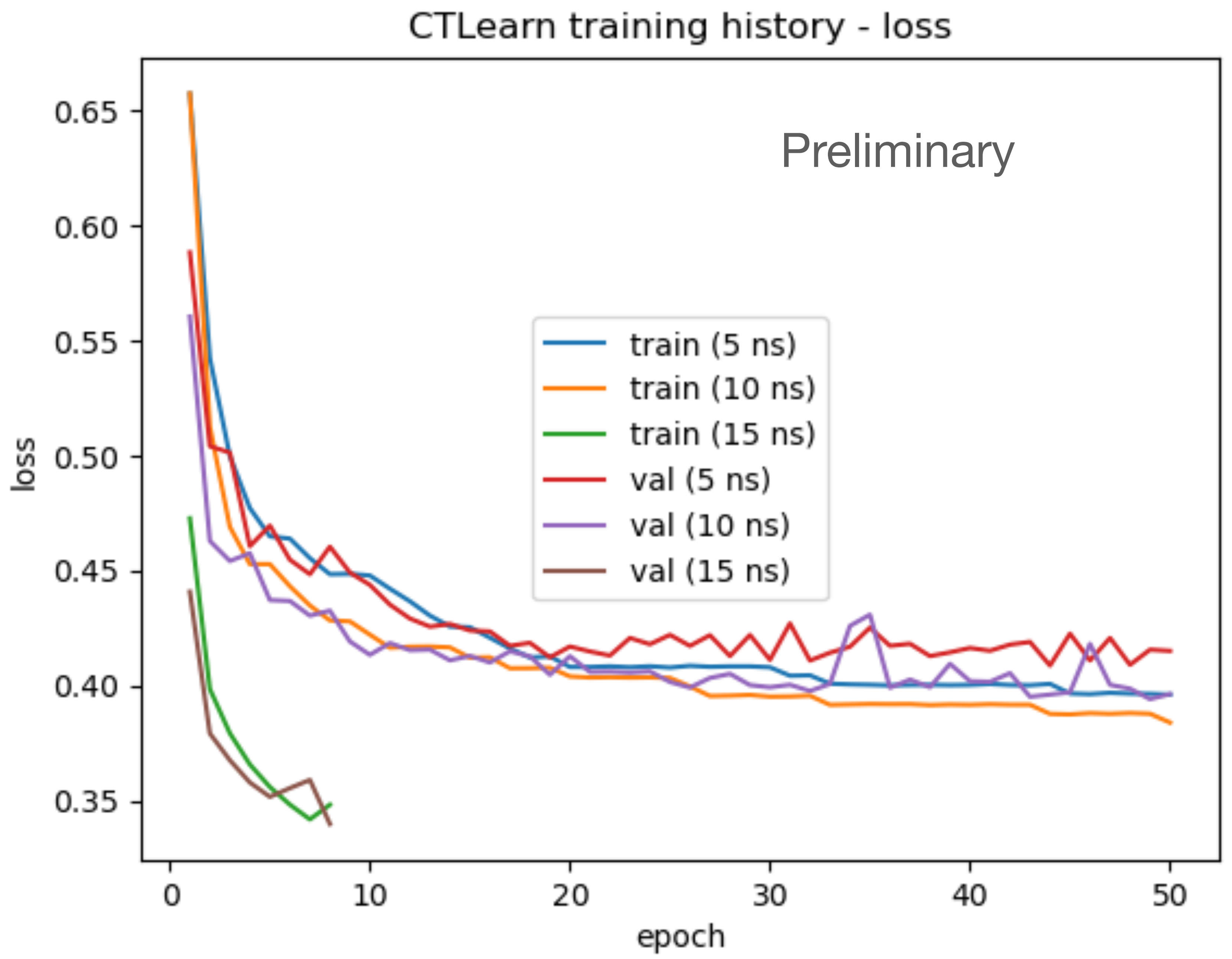
- RAM usage for the training with a ~4TB dataset is less than 10GB per GPU.

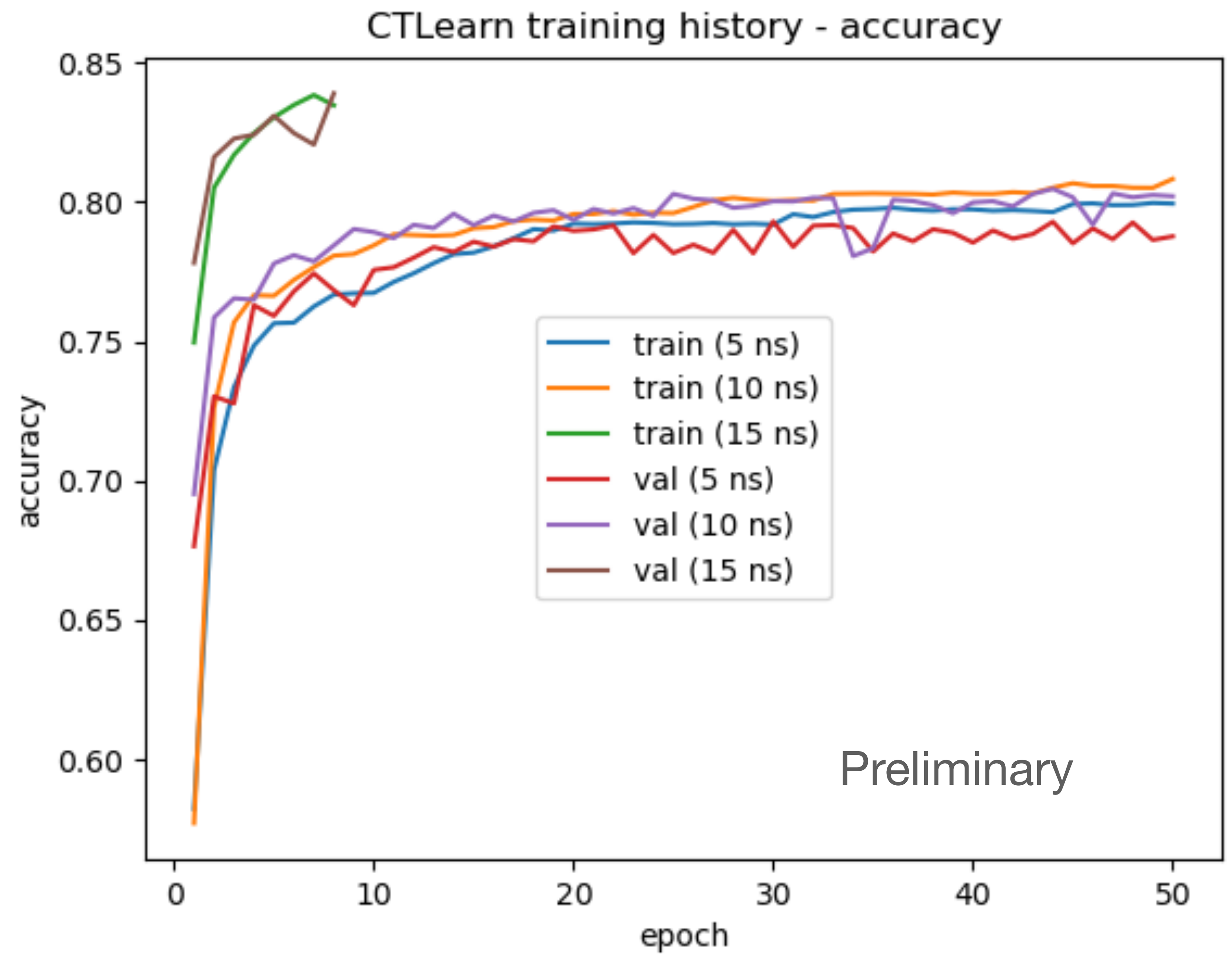


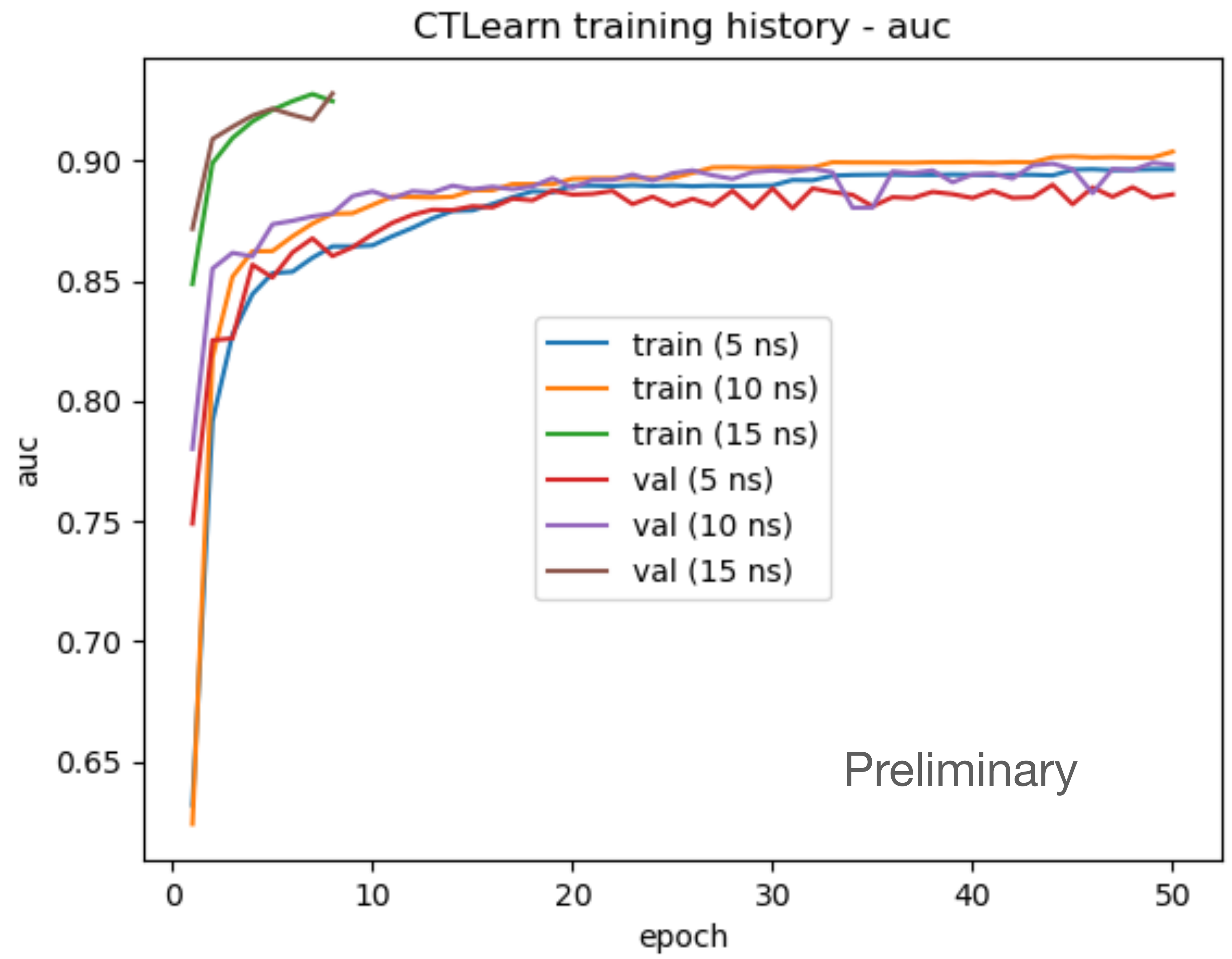
Training process - CNN-based full-event reconstruction on calibrated waveforms with CTLearn

- **Gamma/hadron separation** is possible for CNN-based models on calibrated waveforms!
- Calibrated waveform (R1) training was performed with loose quality cuts (hillas_intensity > 50 & leakage_intensity_width_2 < 0.2) extracting 5, 10, and 15 snapshots (nano seconds) around the shower maximum
- 5 and 10 snapshots were trained with a fraction of the training stats because I was using the run-wised files (less efficient). Training for 15 snapshots (15ns) was done with the whole training stats using the merged files.

5 ns (low stats) vs 10 ns (low stats) vs 15 ns (full stats)



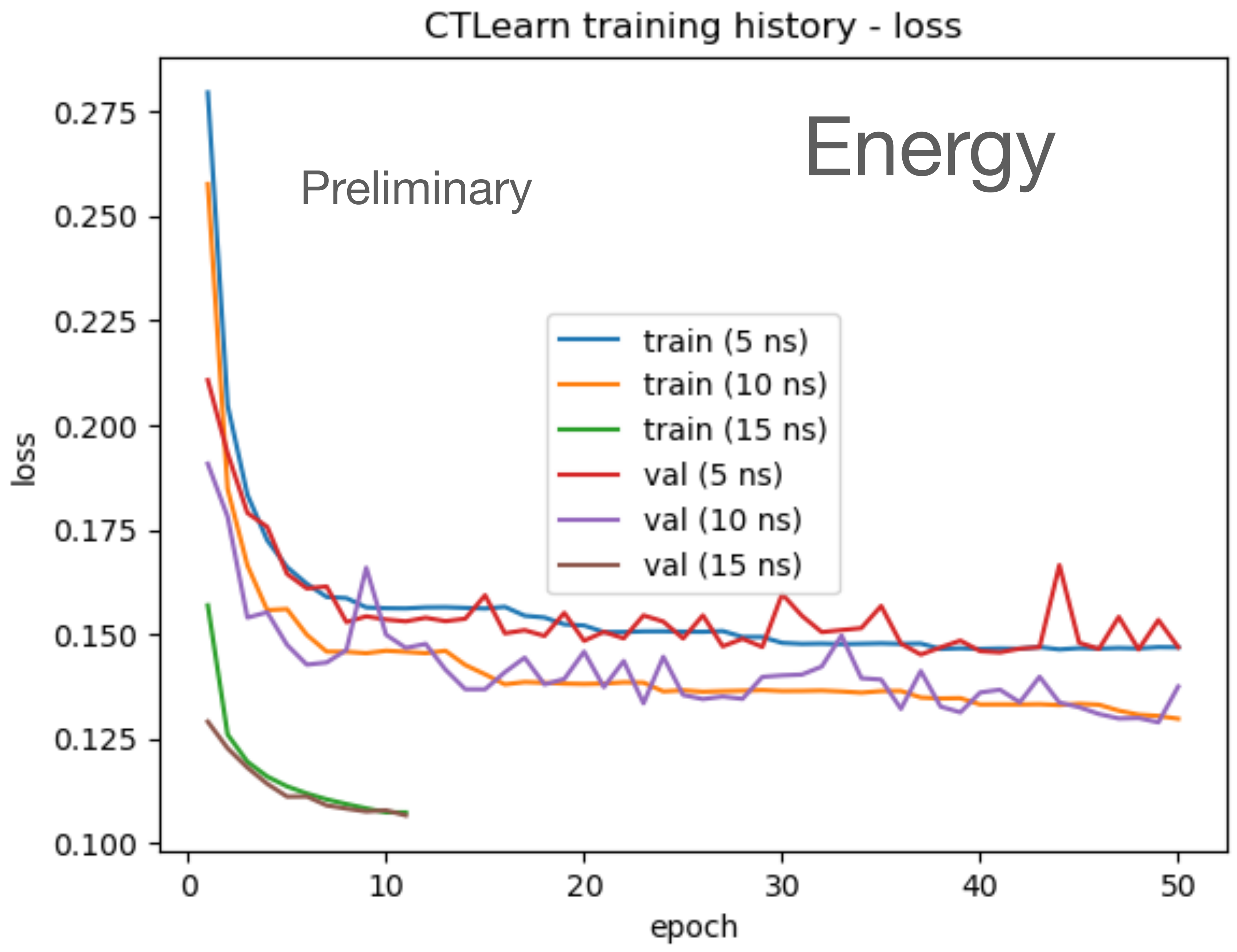


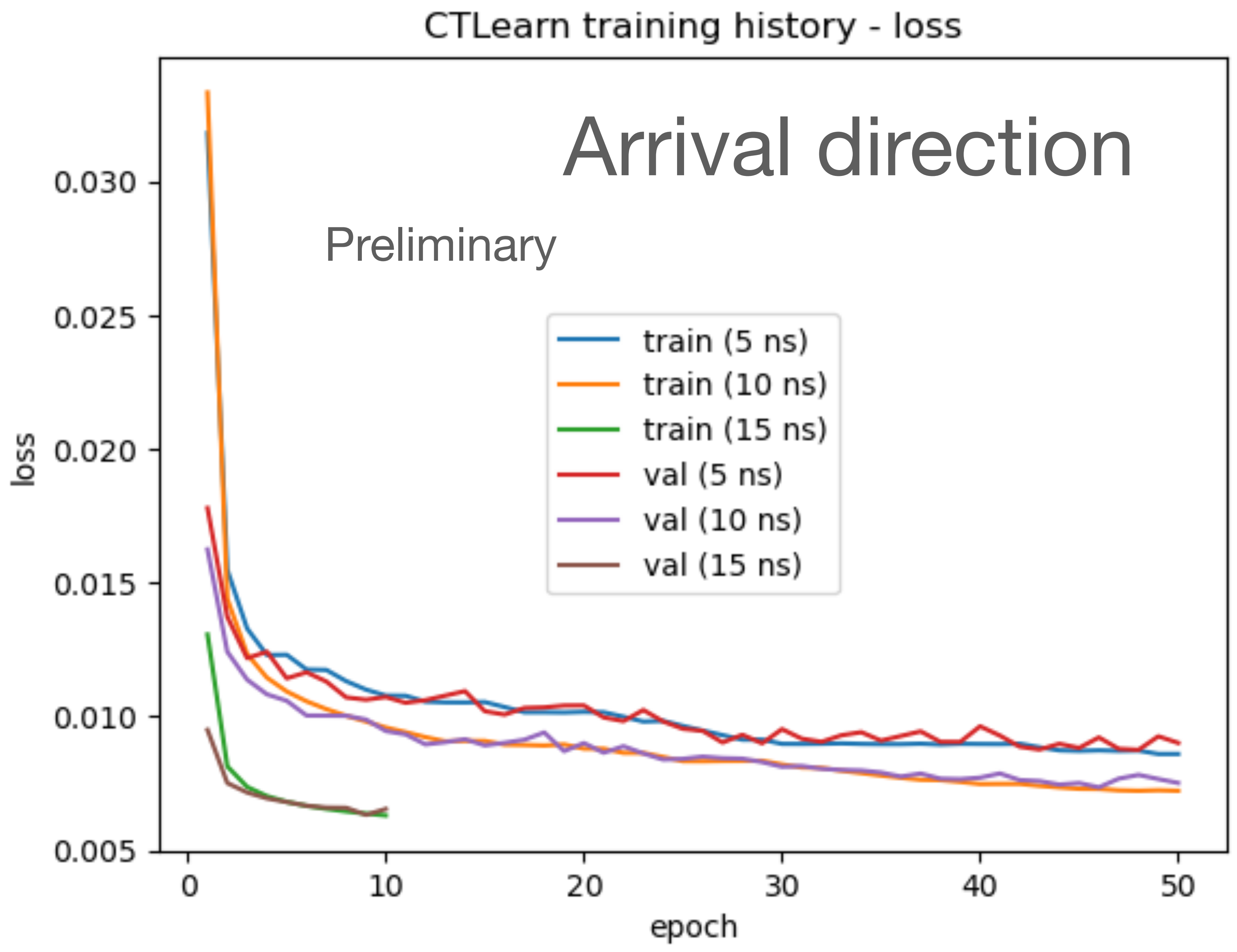


Training process - CNN-based full-event reconstruction on calibrated waveforms with CTLearn

- **Energy and arrival direction regression** is possible for CNN-based models on calibrated waveforms!
- Calibrated waveform (R1) training was performed with loose quality cuts (hillas_intensity > 50 & leakage_intensity_width_2 < 0.2) extracting 5, 10, and 15 snapshots (nano seconds) around the shower maximum
- 5 and 10 snapshots were trained with a fraction of the training stats because I was using the run-wised files (less efficient). Training for 15 snapshots (15ns) was done with the whole training stats using the merged files.

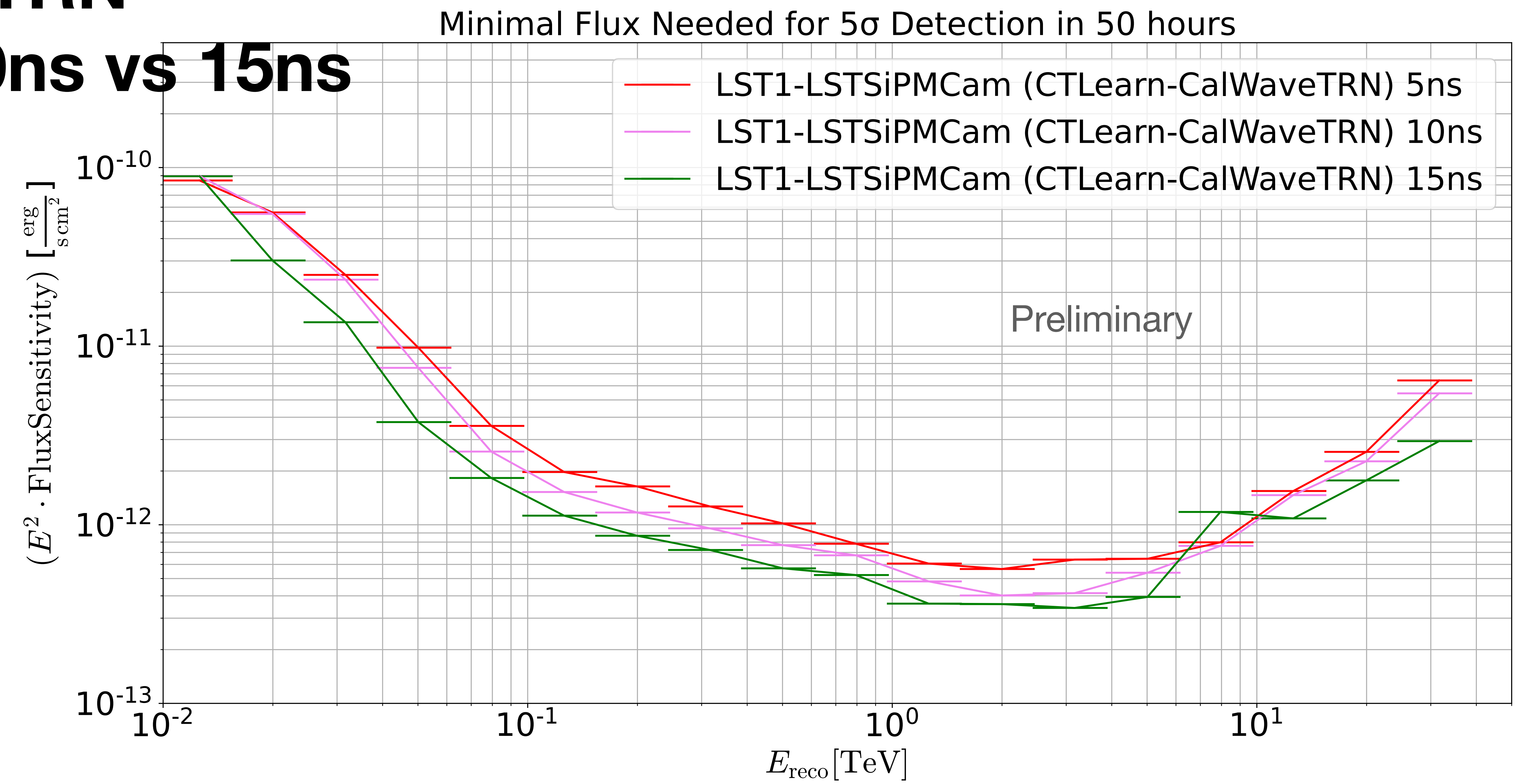
5 ns (low stats) vs 10 ns (low stats) vs 15 ns (full stats)

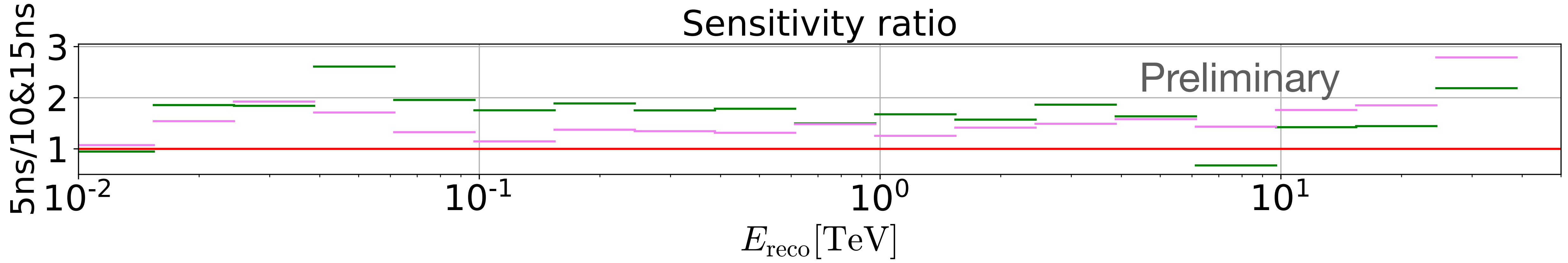


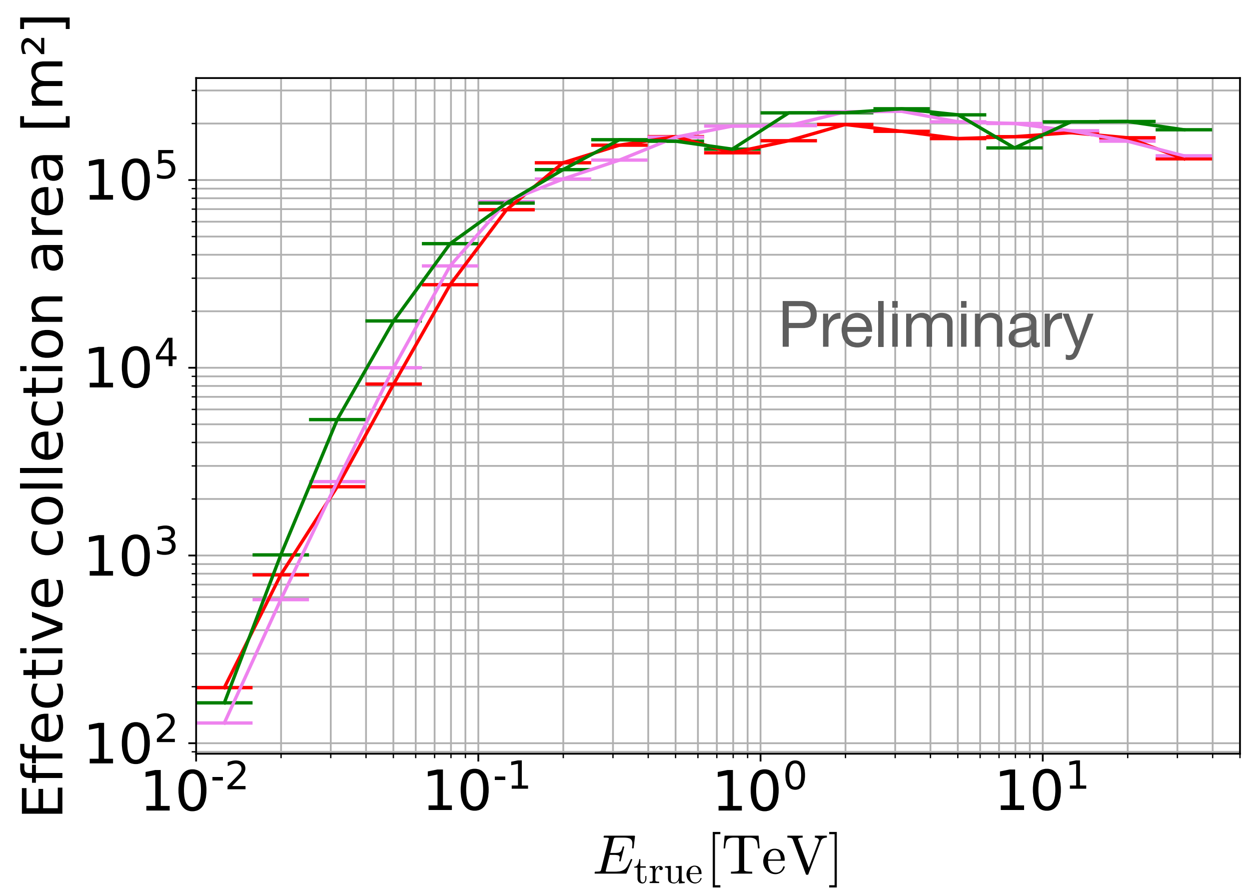


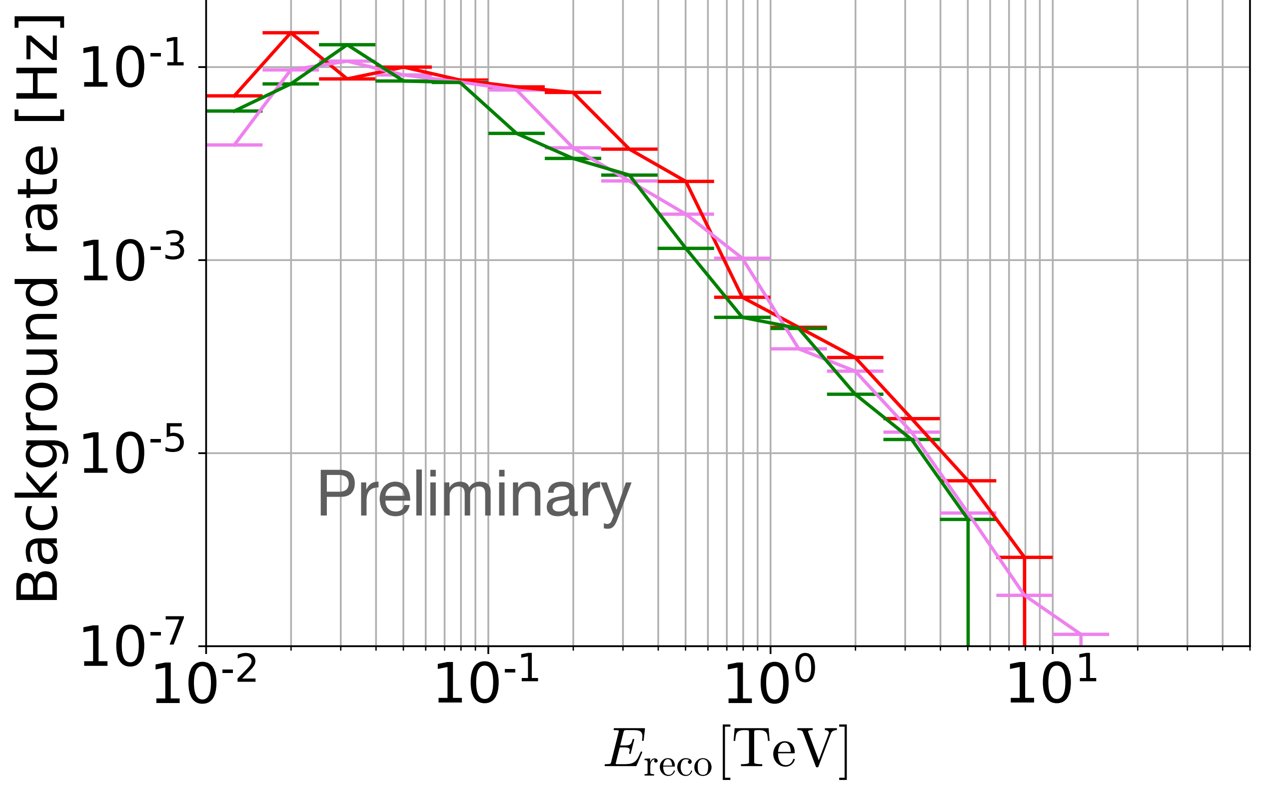
T. Miener et al.

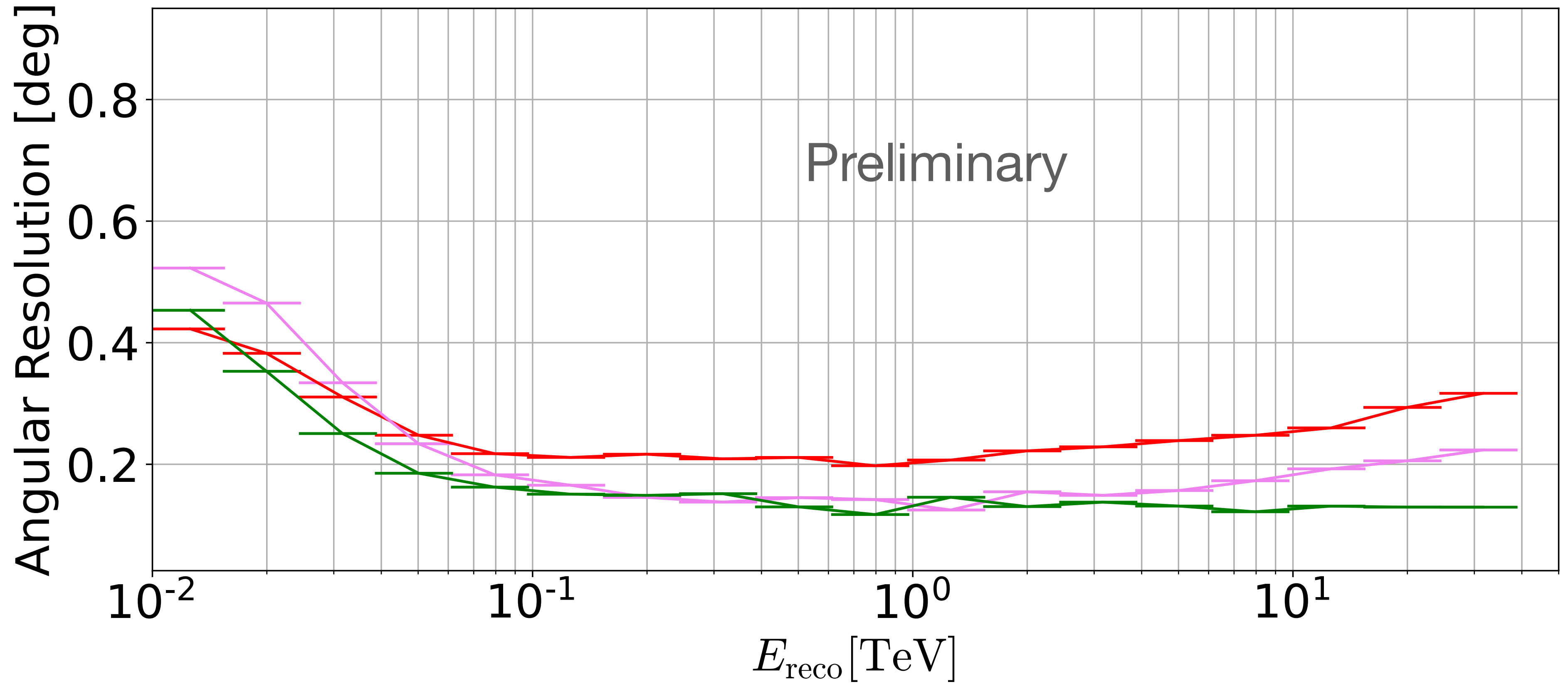
CalWaveTRN
5ns vs 10ns vs 15ns

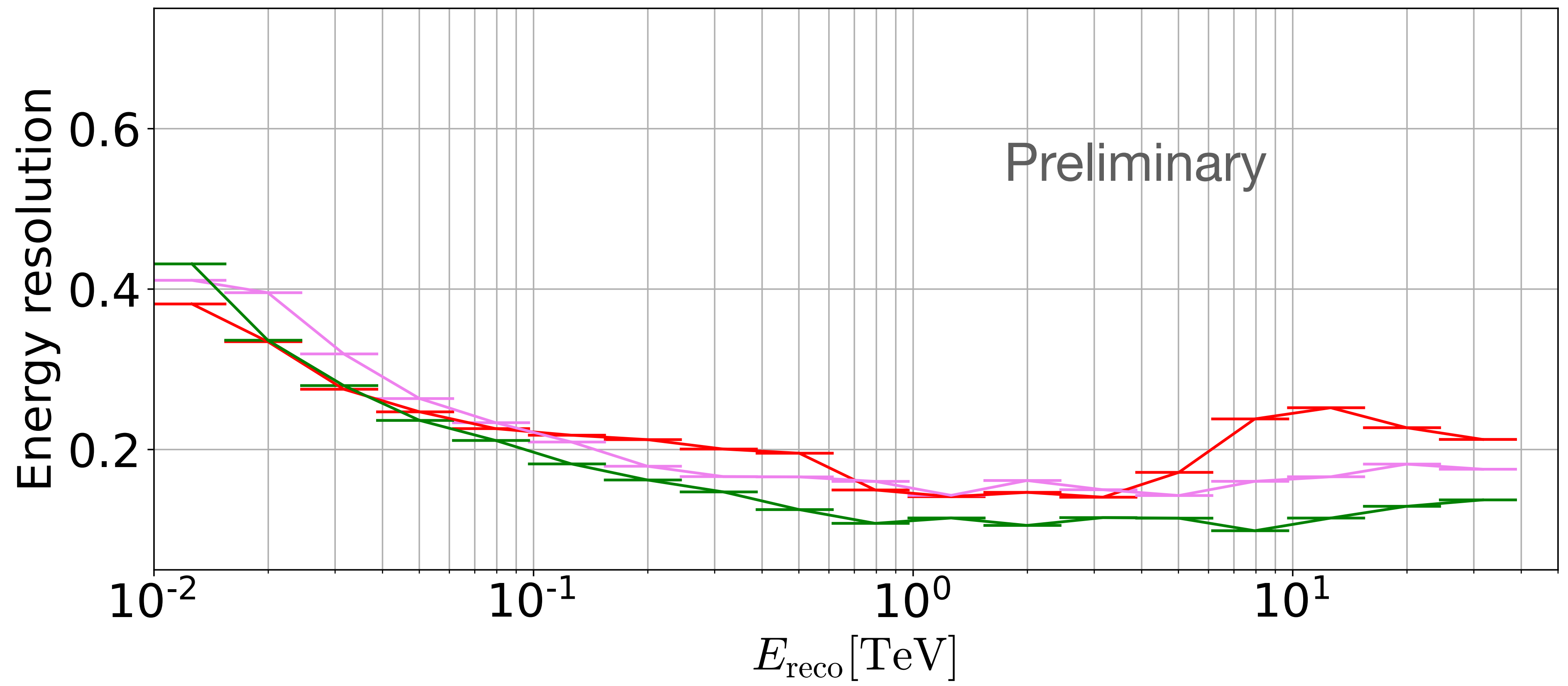






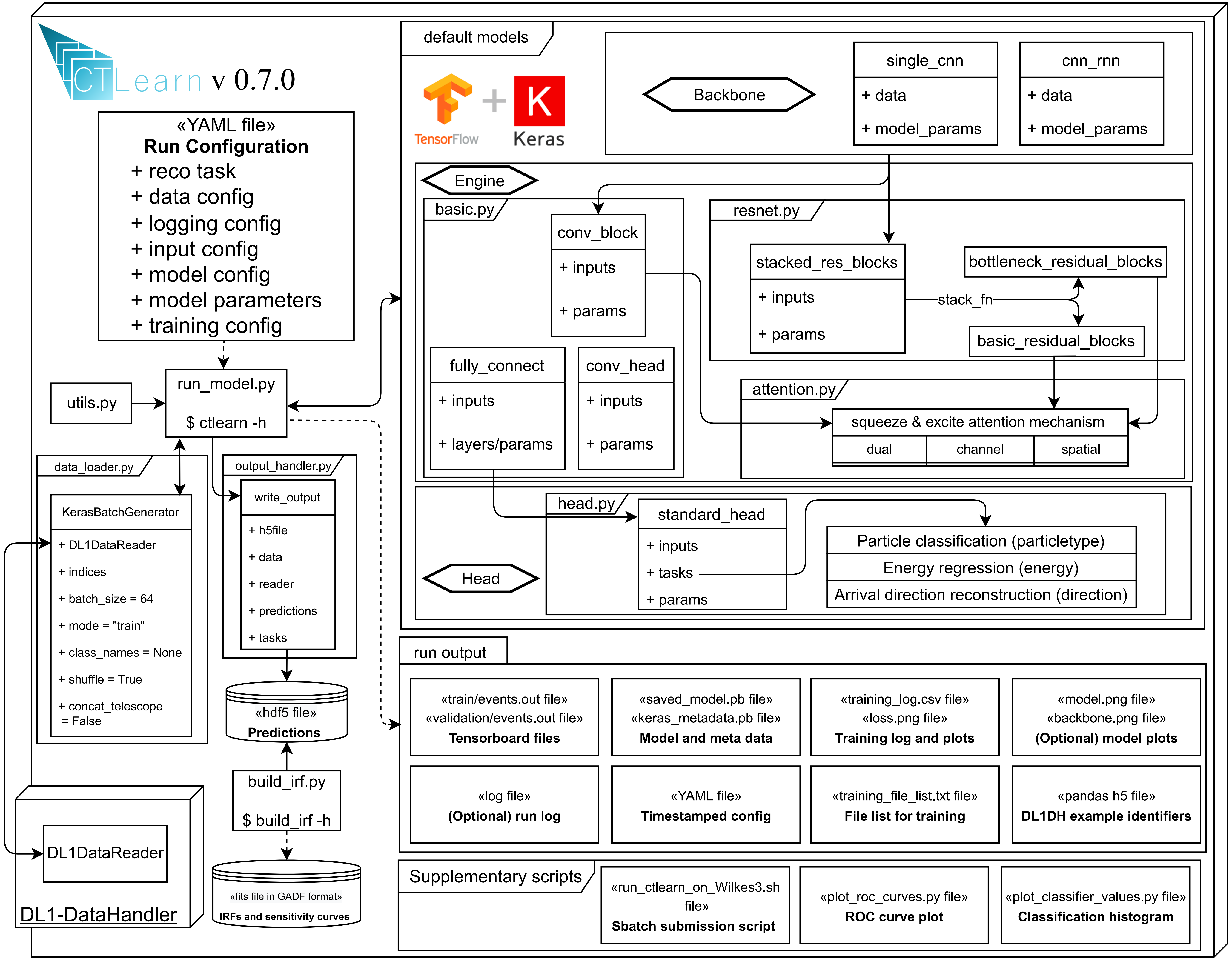




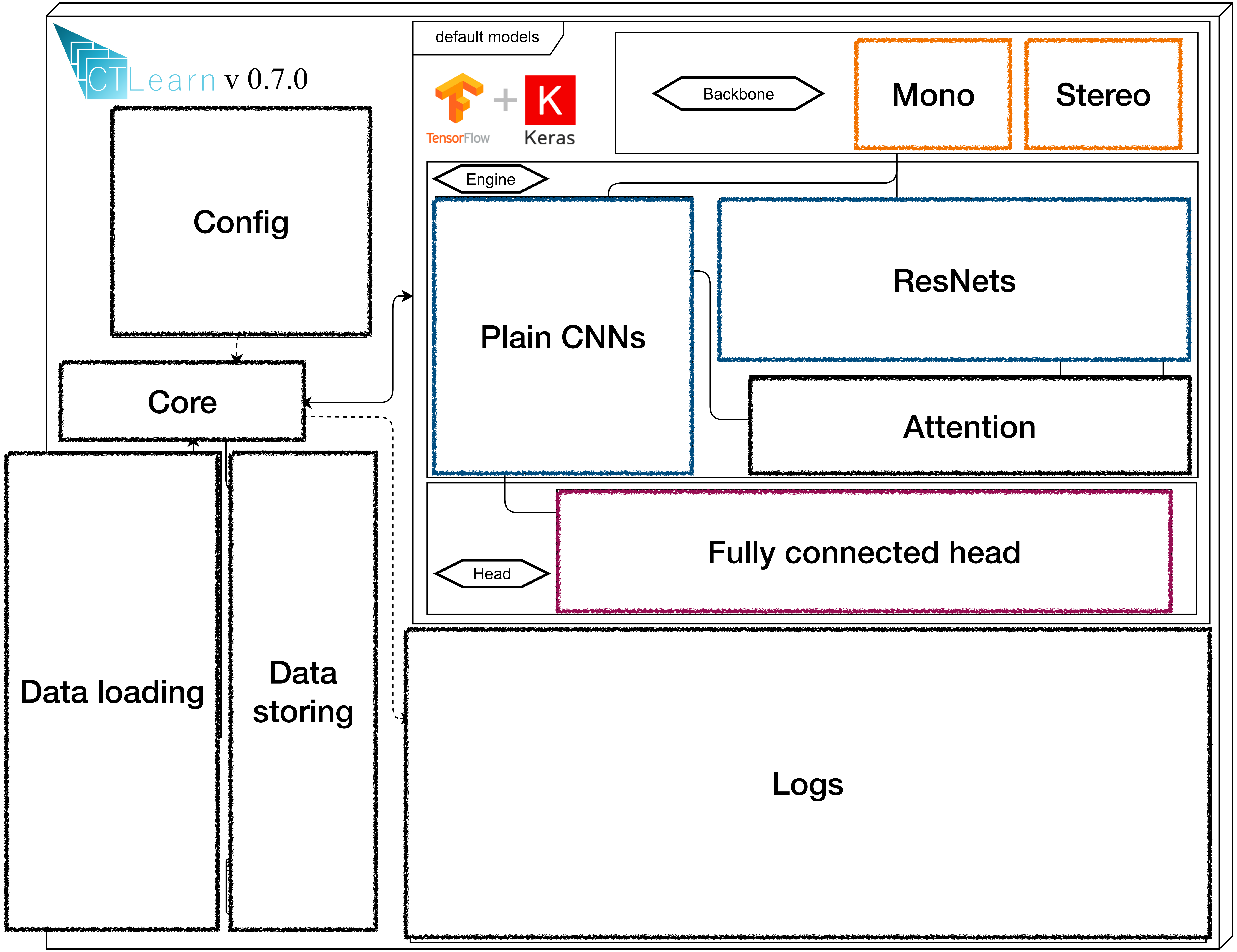


T. Miener et al.

CTLearn workflow

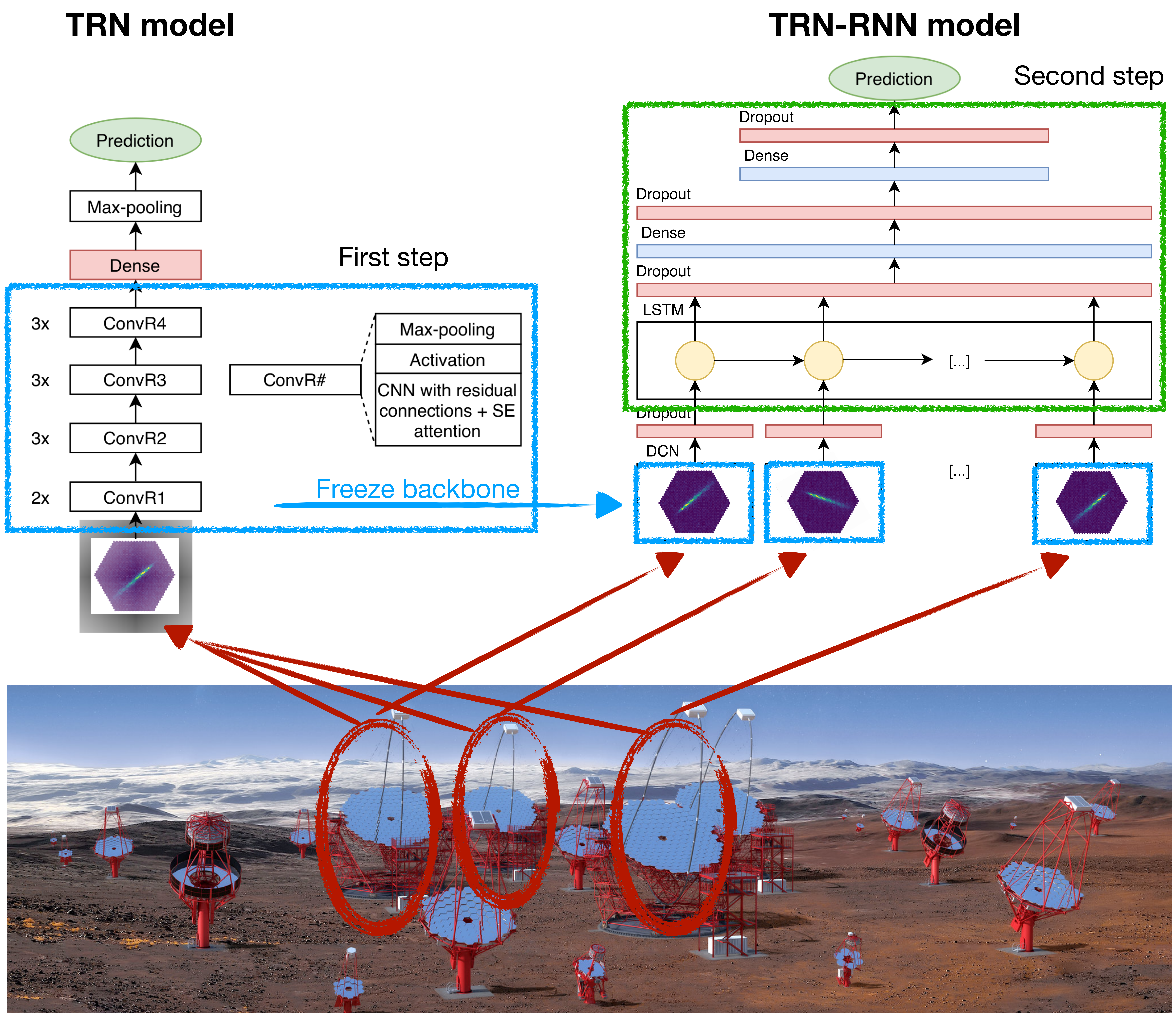


CTLearn workflow



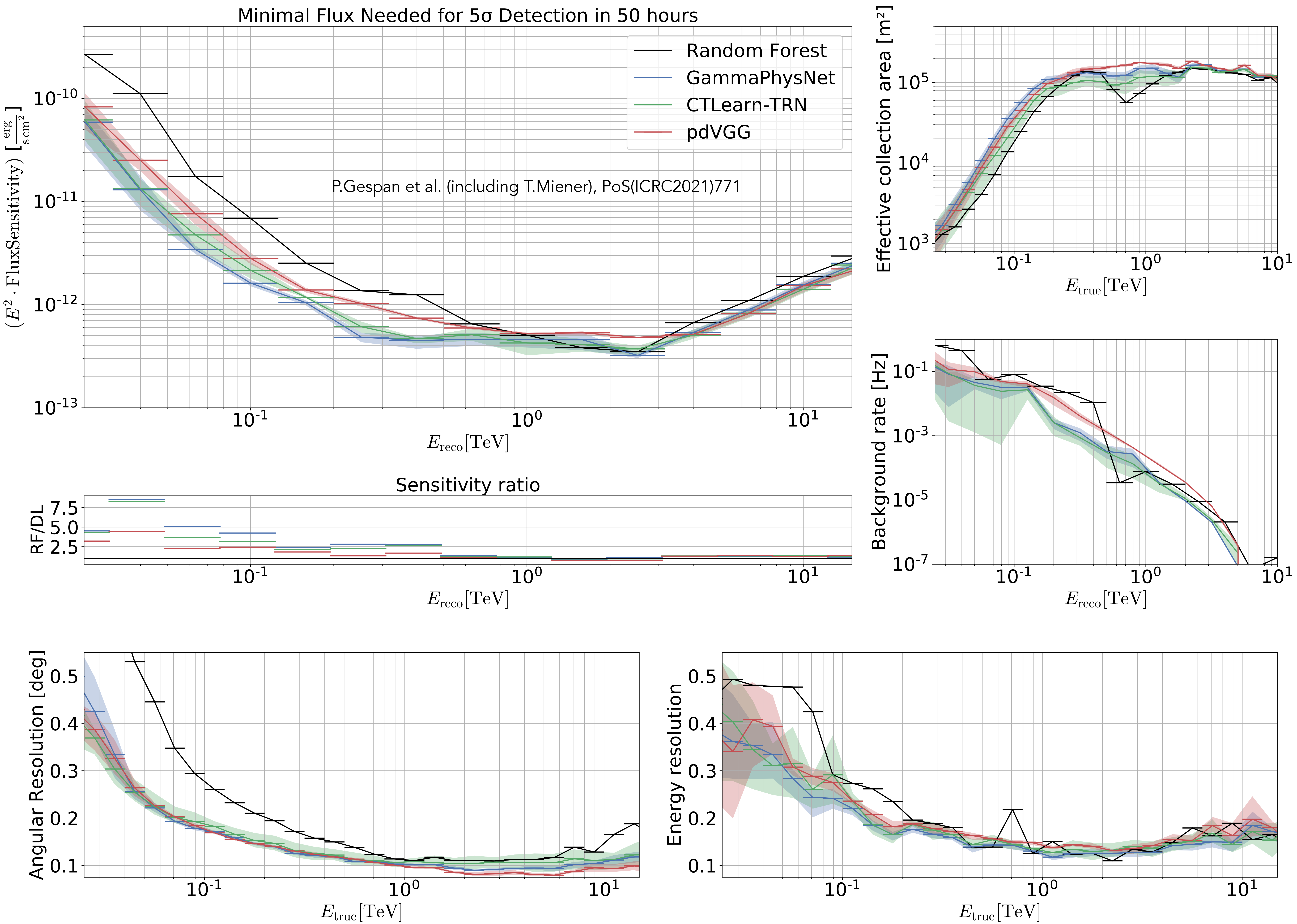
T. Miener et al.

Default CTLearn models (current best-performing)



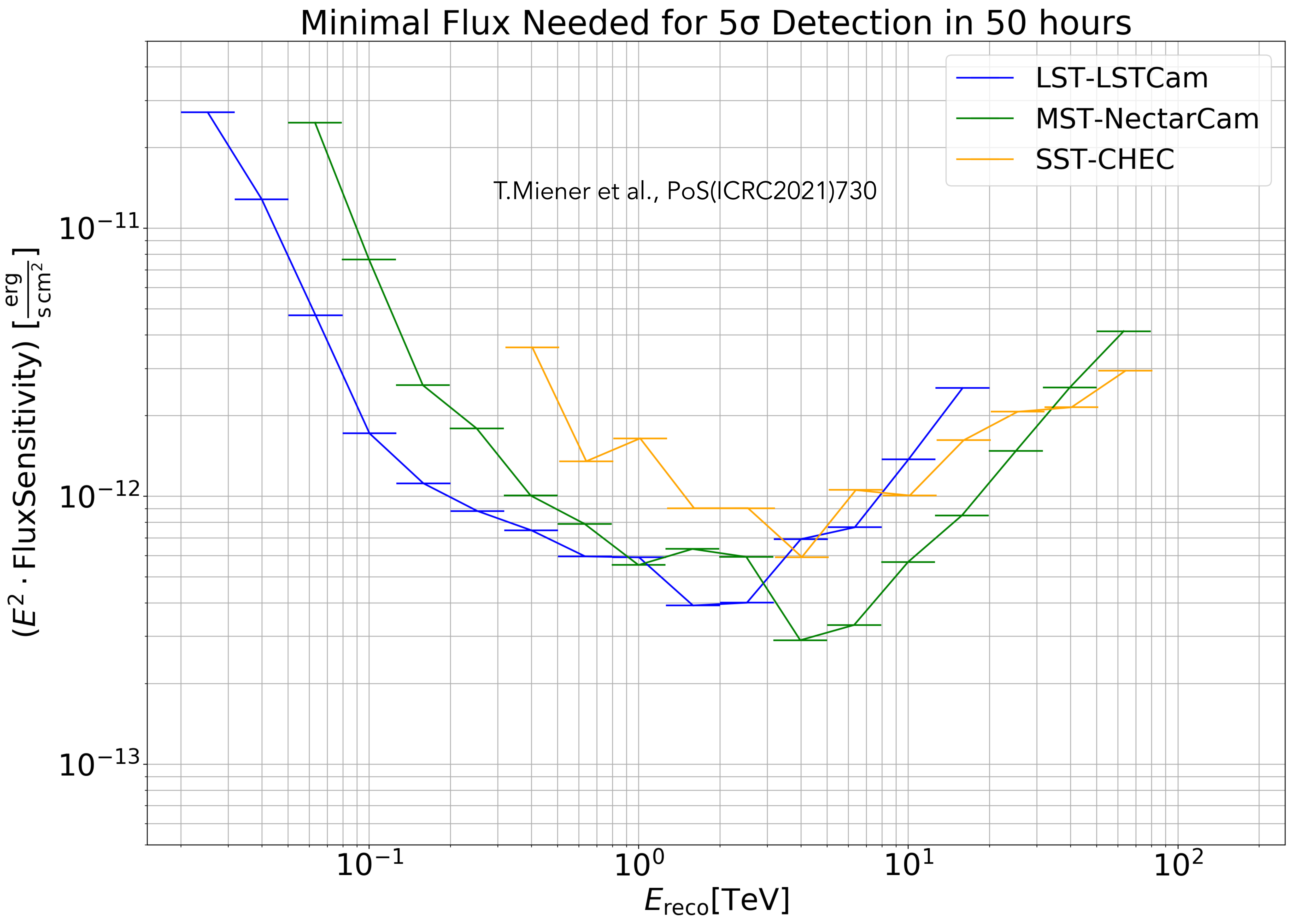
T. Miener et al.

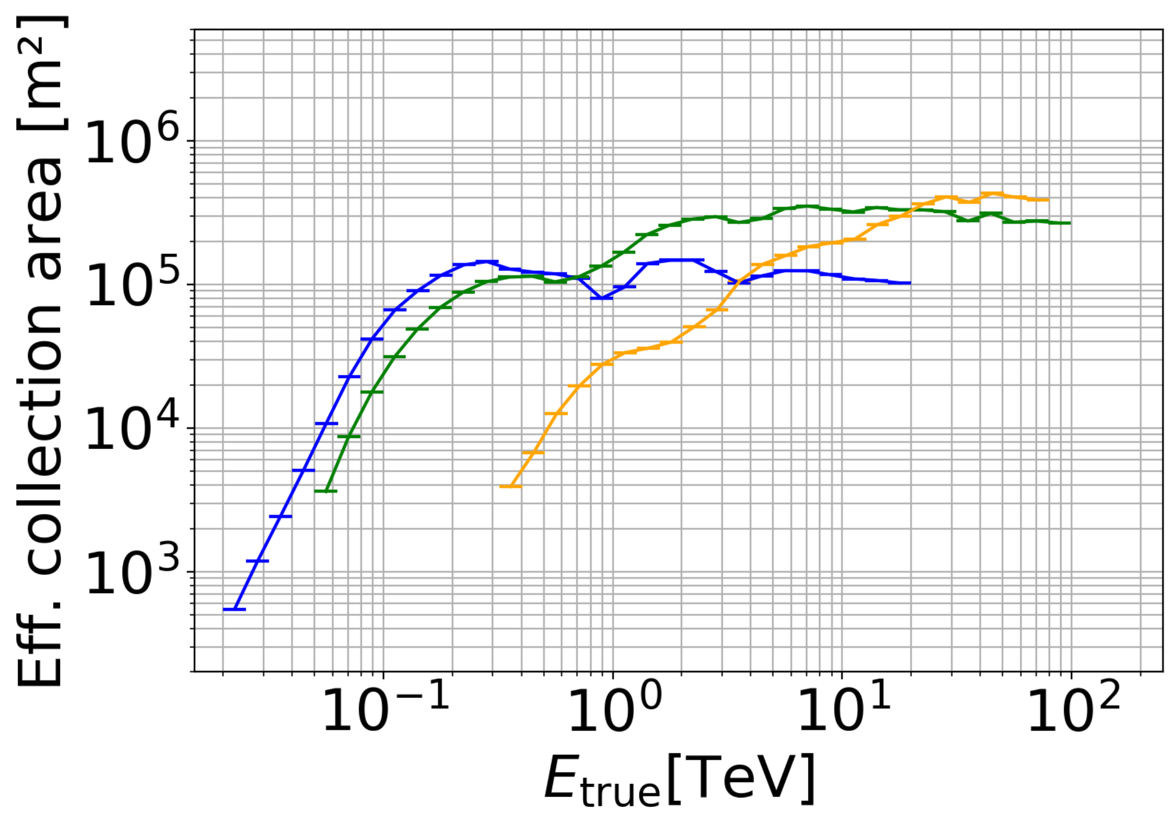
DL-driven event reconstruction applied to simulated data from a single LST of CTA

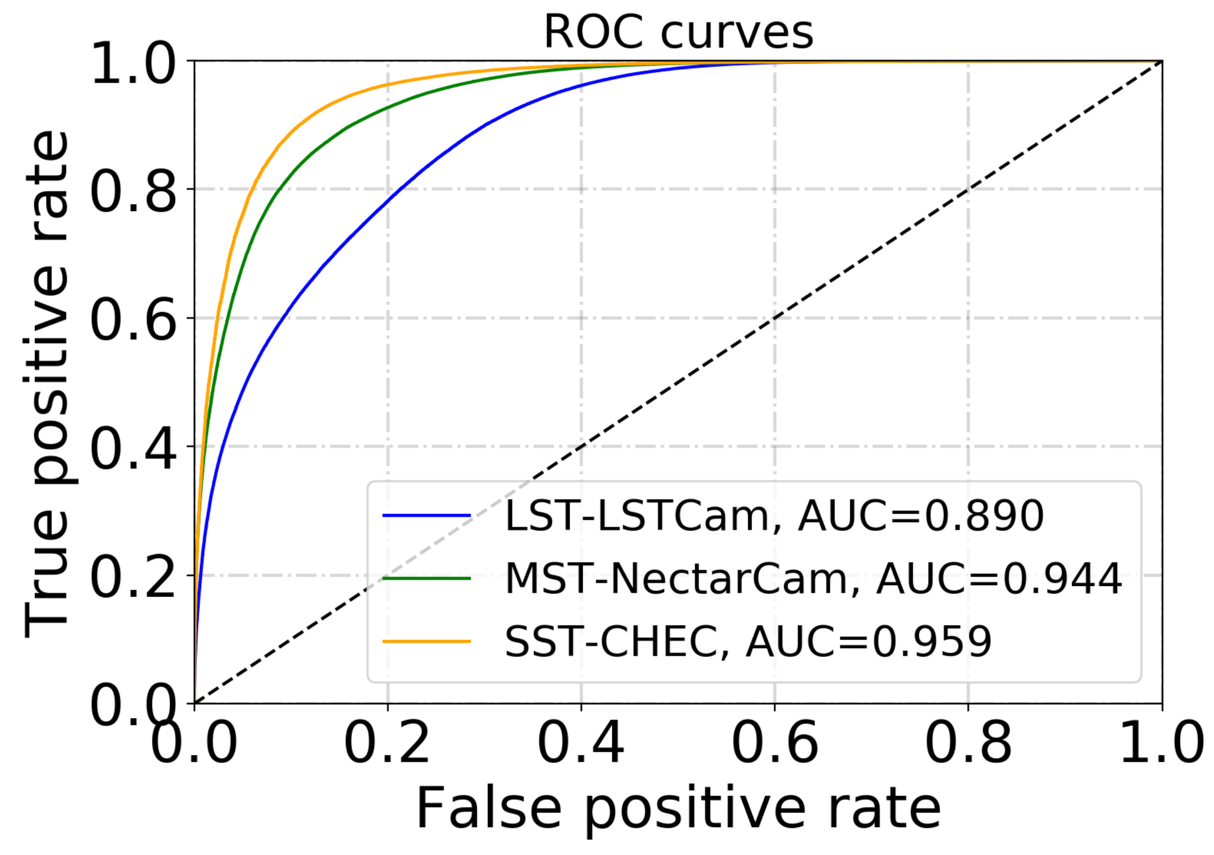


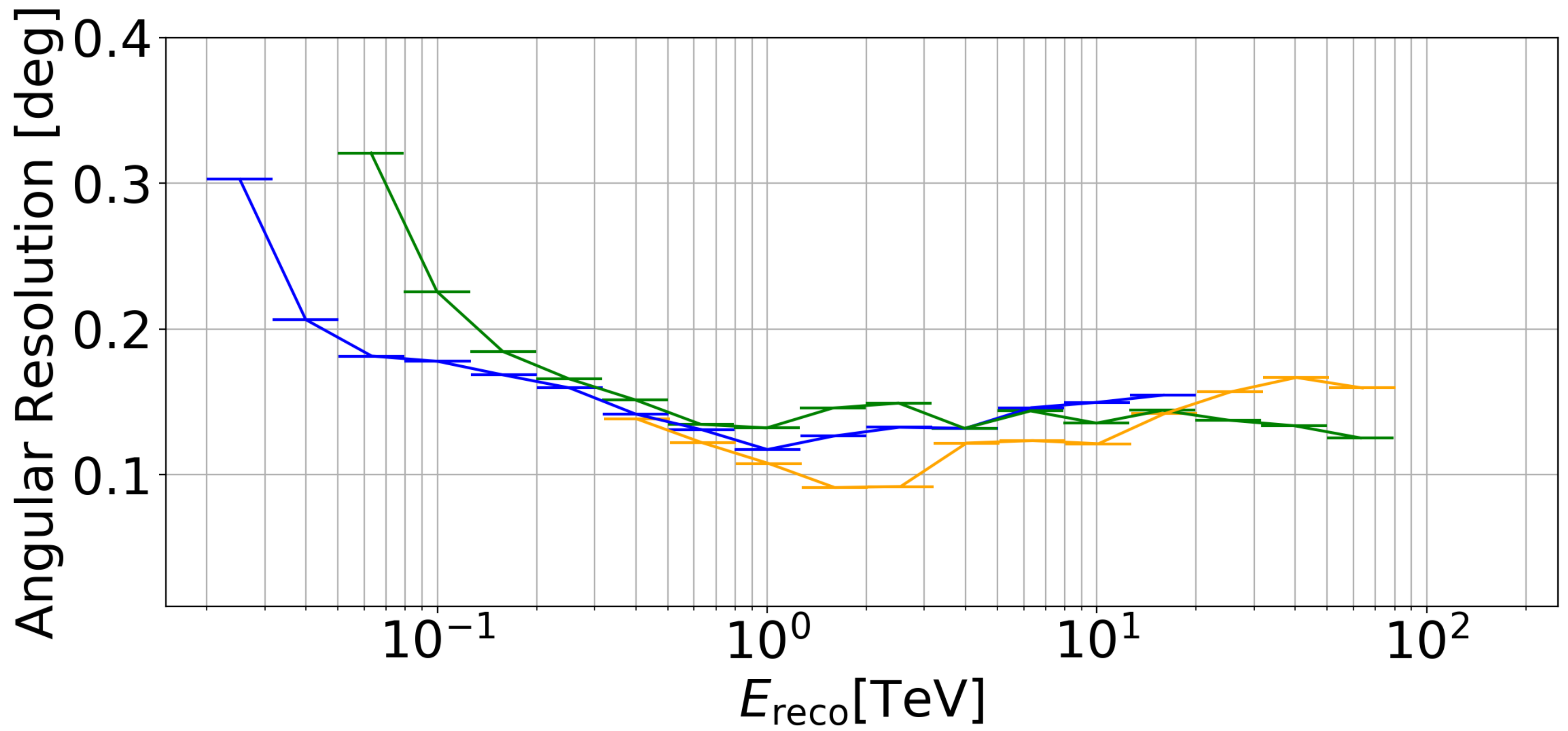
T. Miener et al.

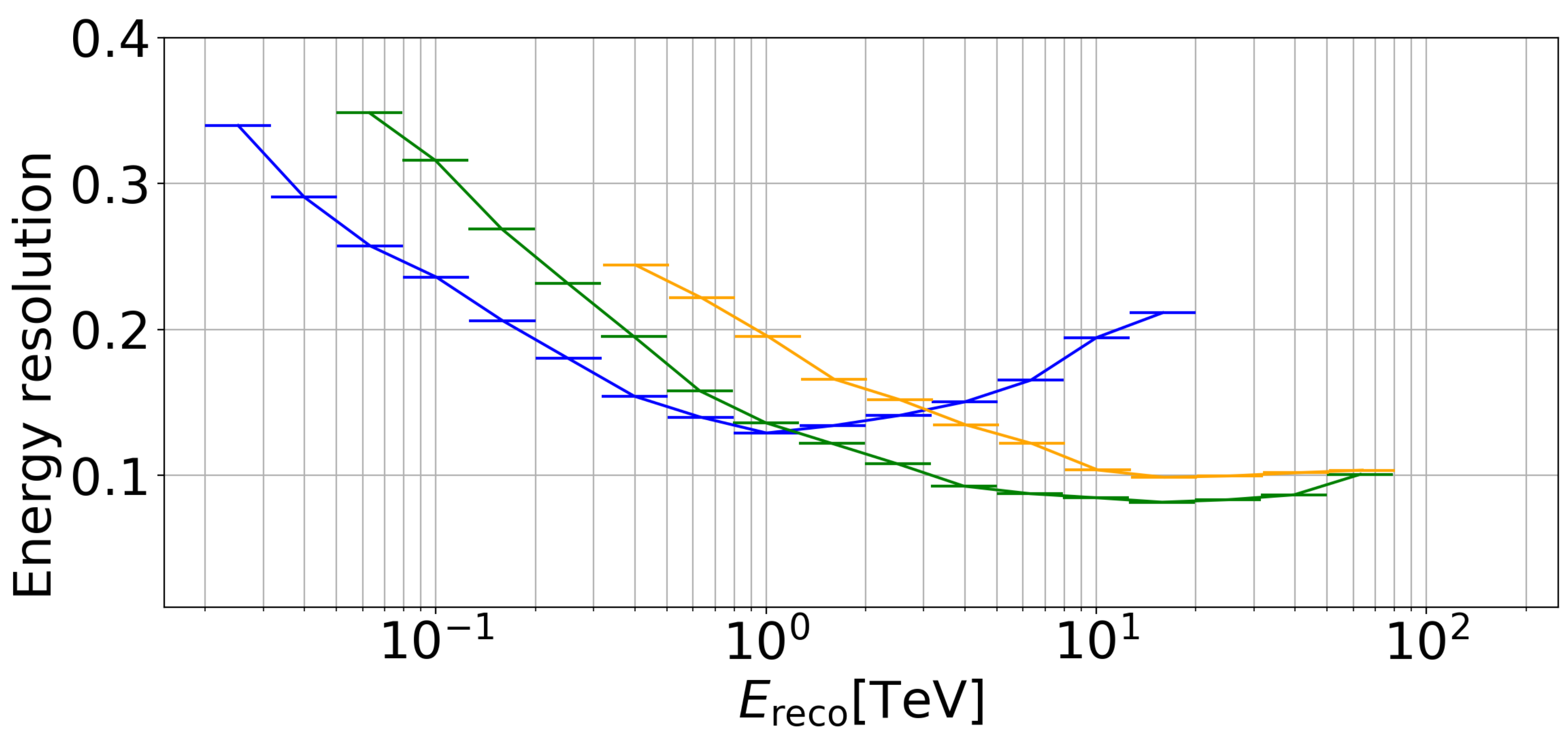
First DCN-based full-event reconstruction on all CTA telescope types (single-telescope)





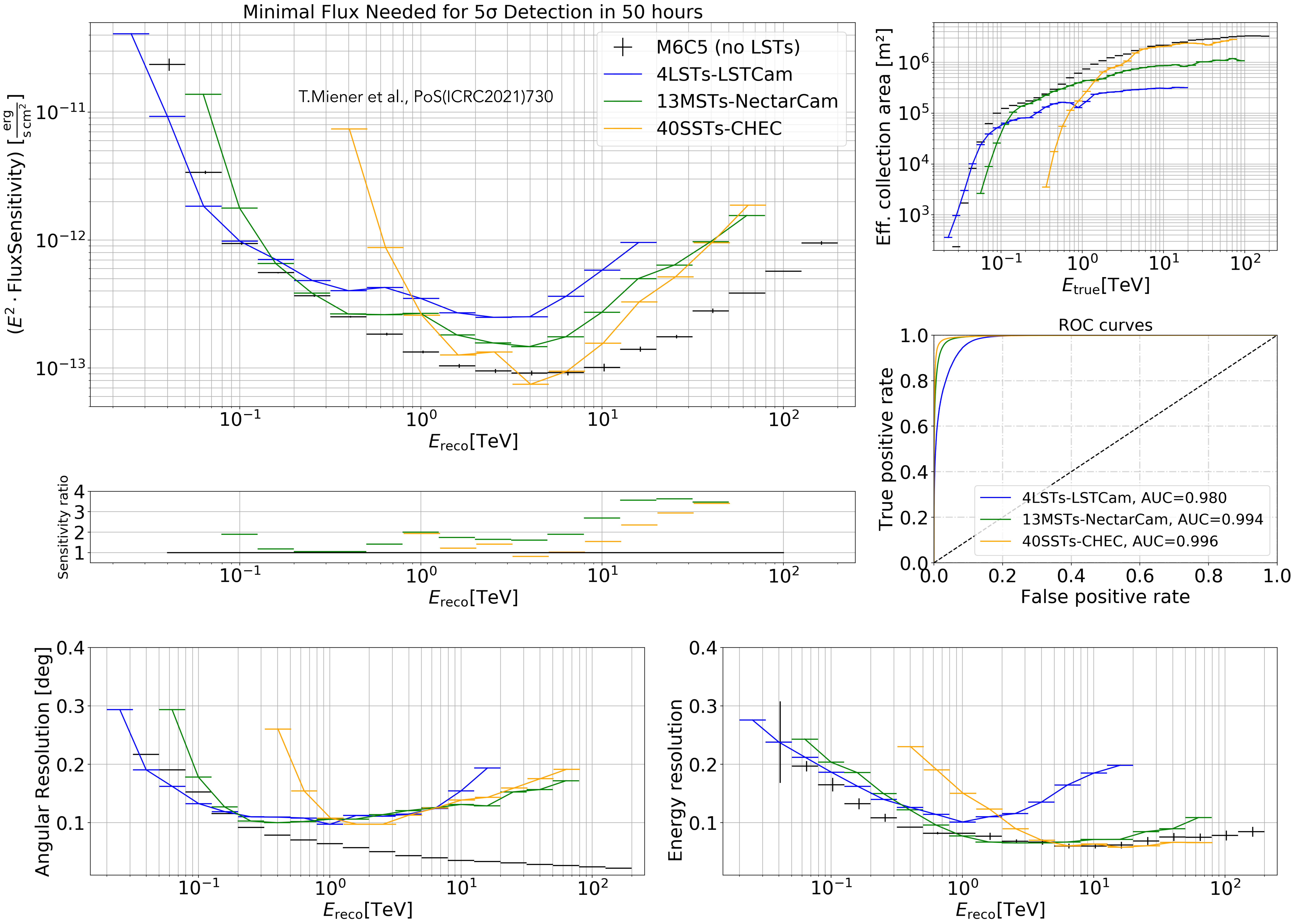






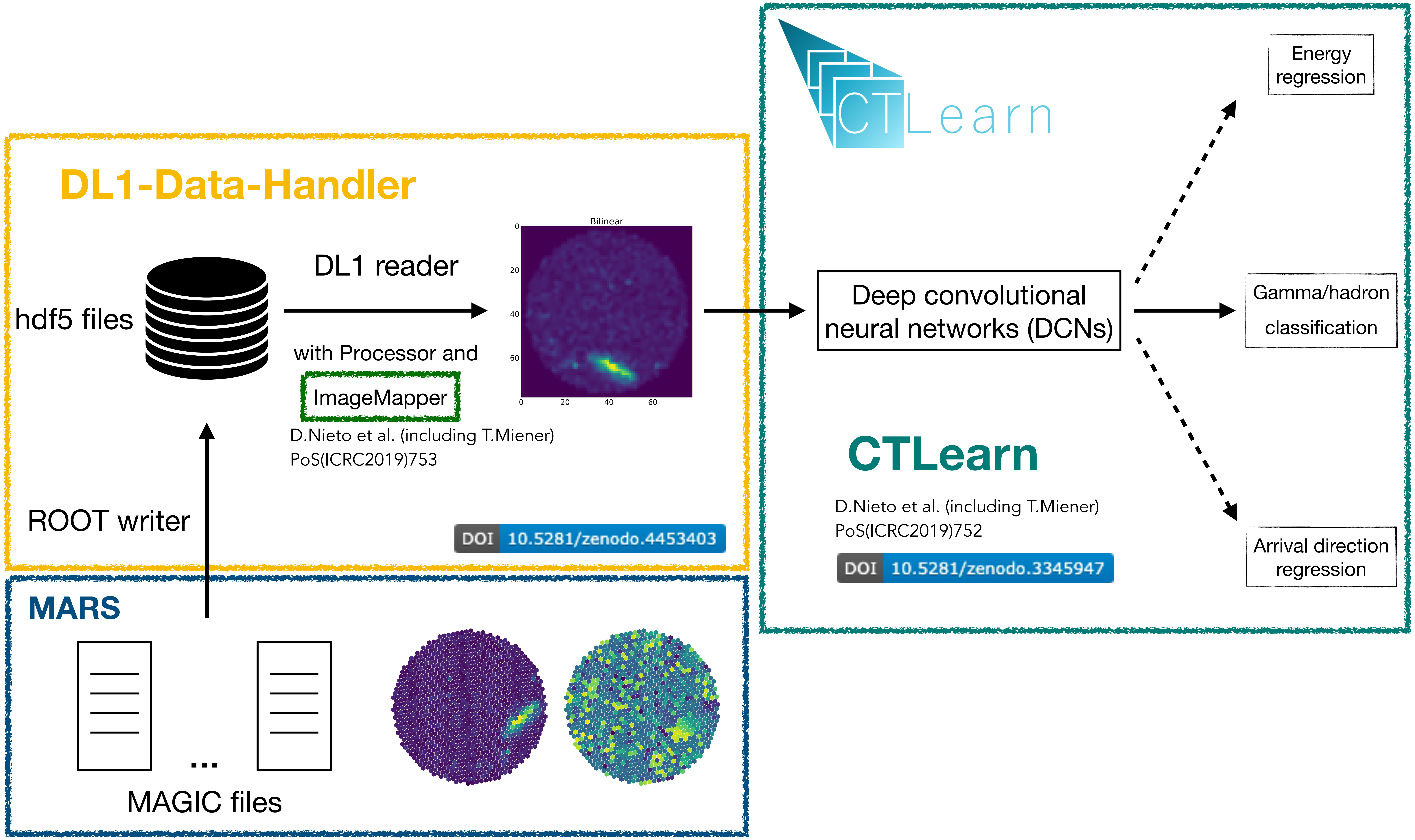
T. Miener et al.

First DCN-based full-event reconstruction on all CTA telescope types (multi-telescopes)



T. Miener et al.

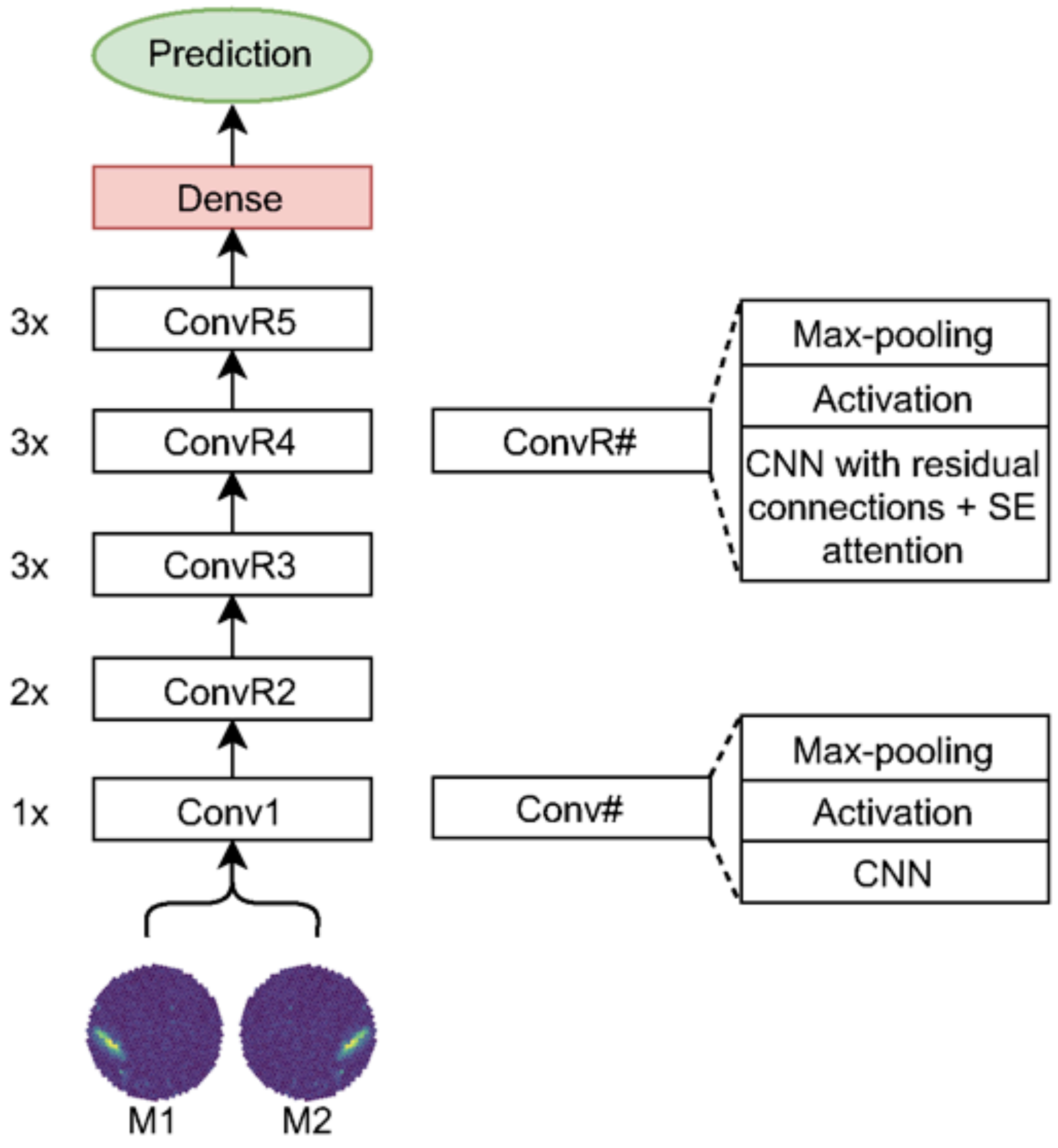
DL application on real data with the MAGIC telescopes

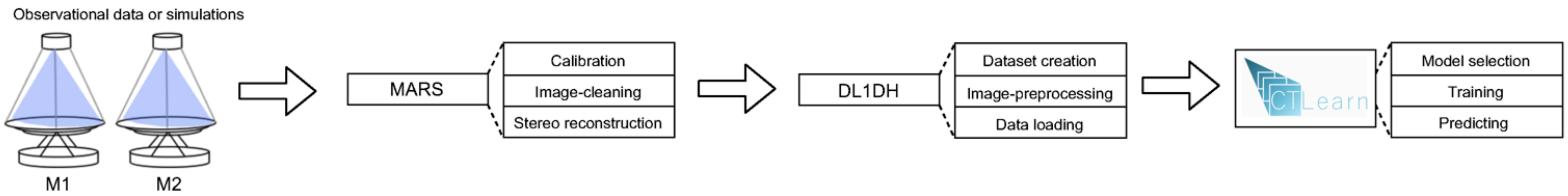


T. Miener et al.

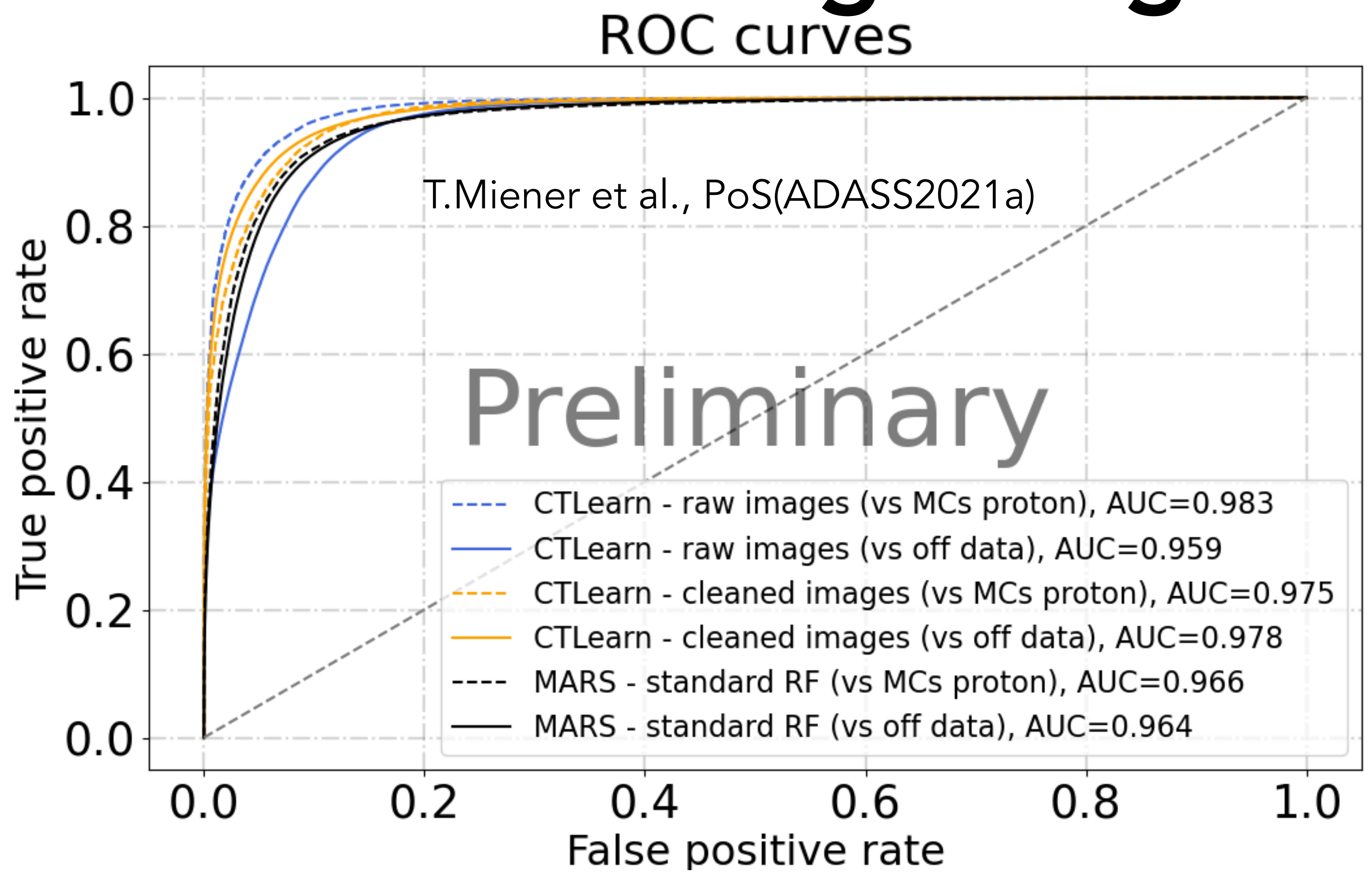
DL application on real data with the MAGIC telescopes

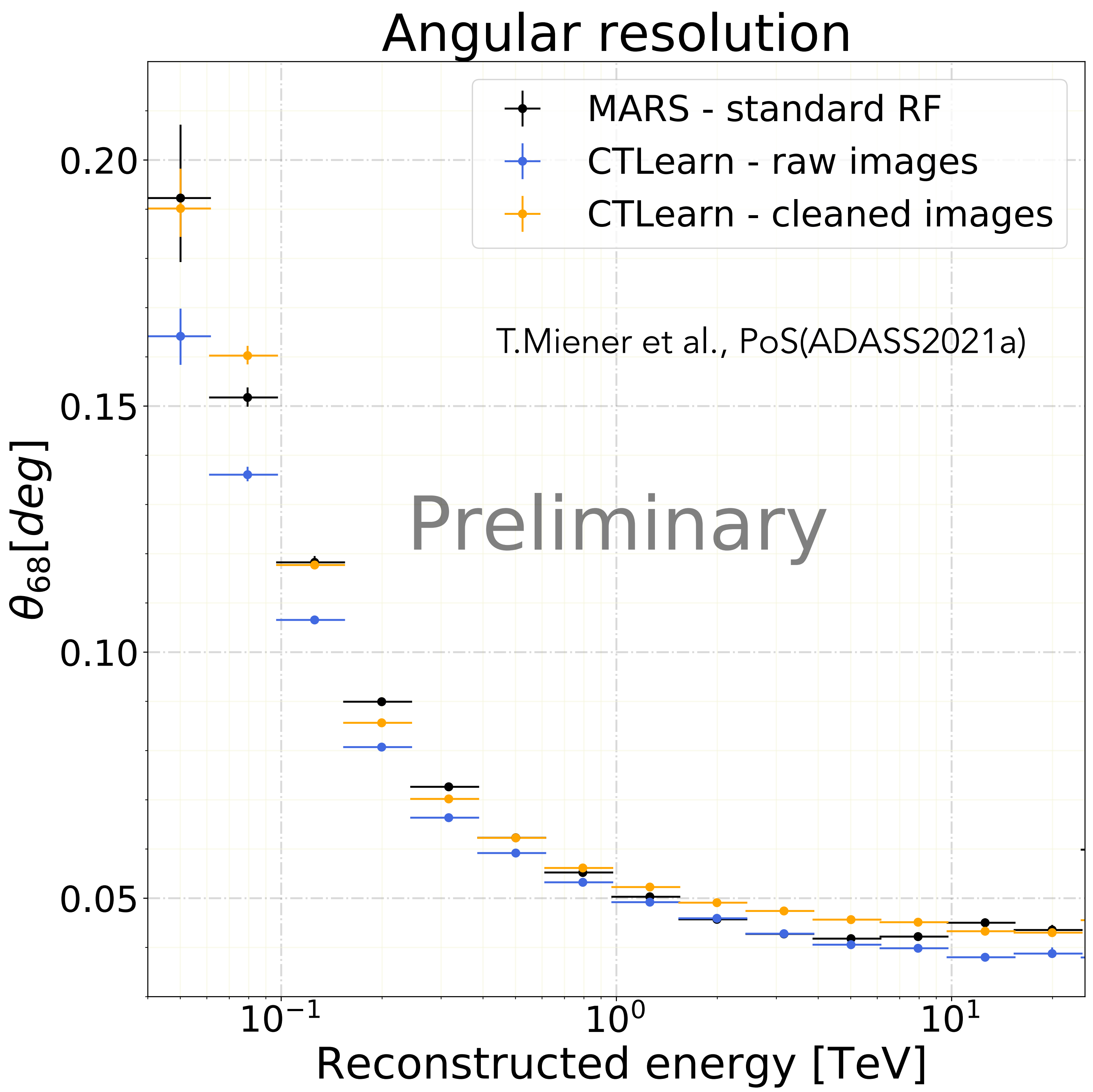
- Establishment of an alternative data processing pipeline for the MAGIC telescopes based on *MARS* and *DL1DH+CTLearn*
- First source detection (Crab Nebula) using DCNs from the *CTLearn* team
- We explored two different analysis schemes, where we trained the same TRN model with raw images and cleaned images
- A Crab Nebula sample of 2.93 h have been analyzed with *CTLearn* and the latest *MARS* software using the standard settings for the analysis focusing of the full range (FR; >250 GeV) - and low energy (LE; >100 GeV) range.

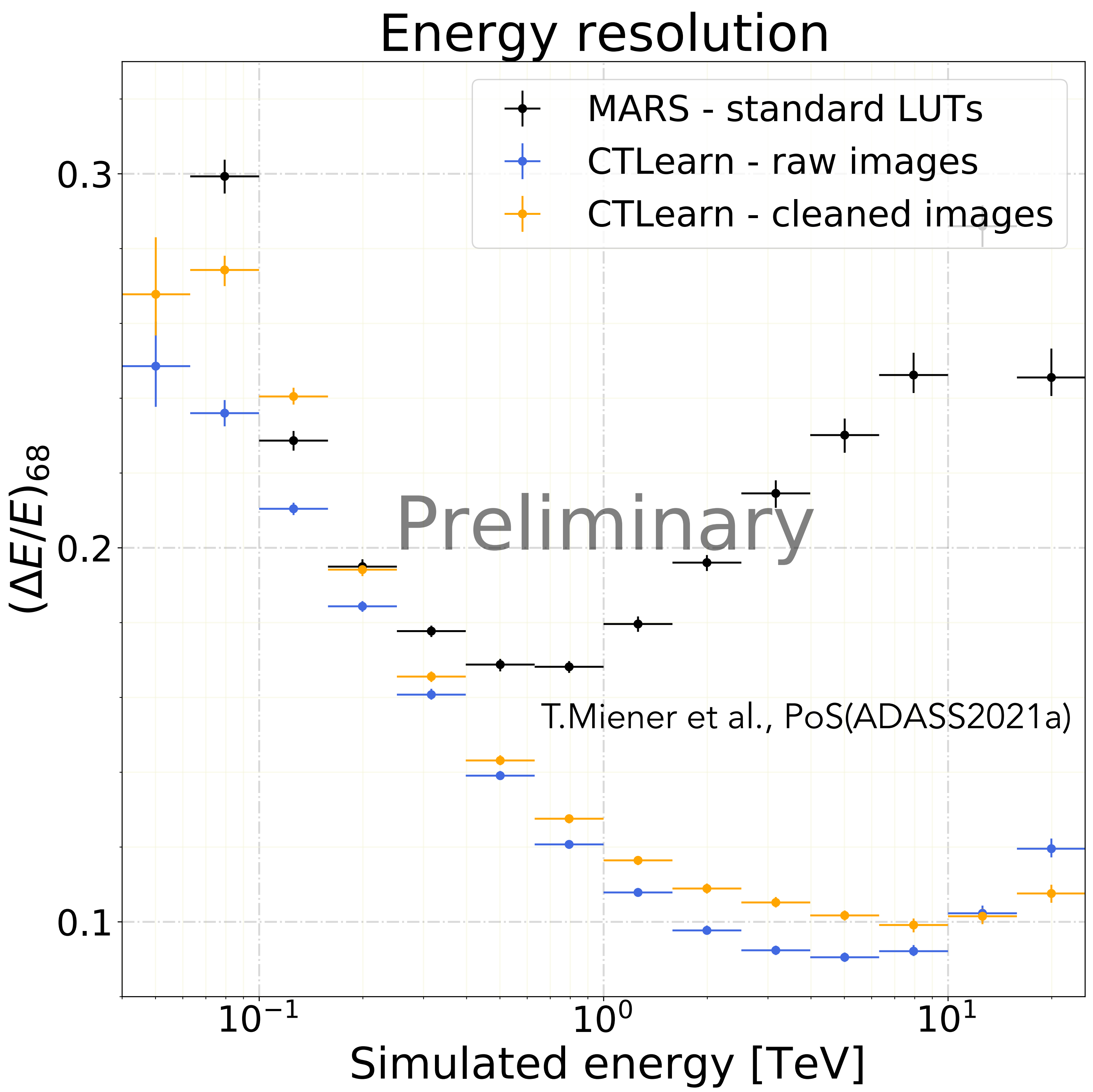




Reconstruction performance using MC gamma simulations

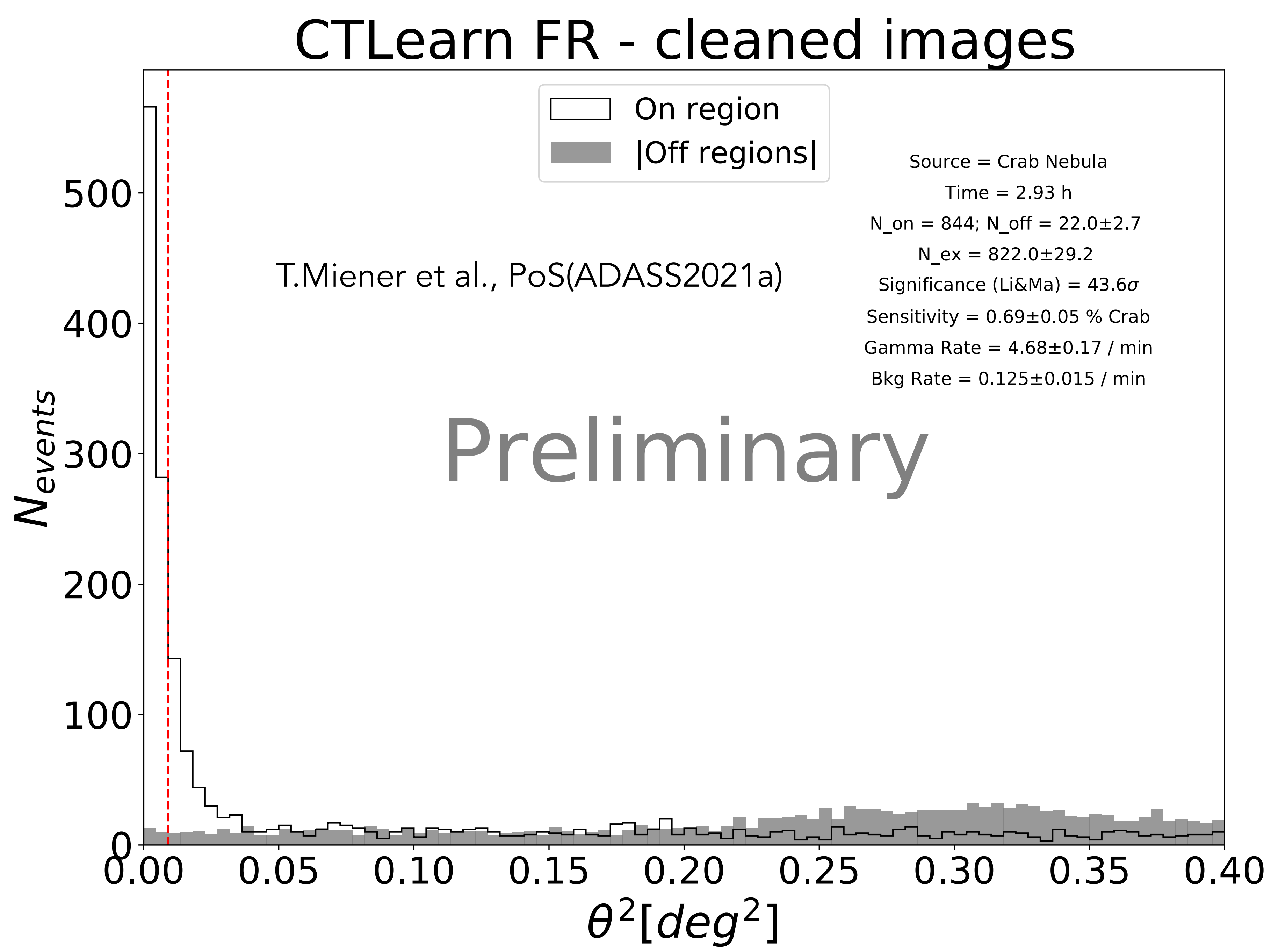


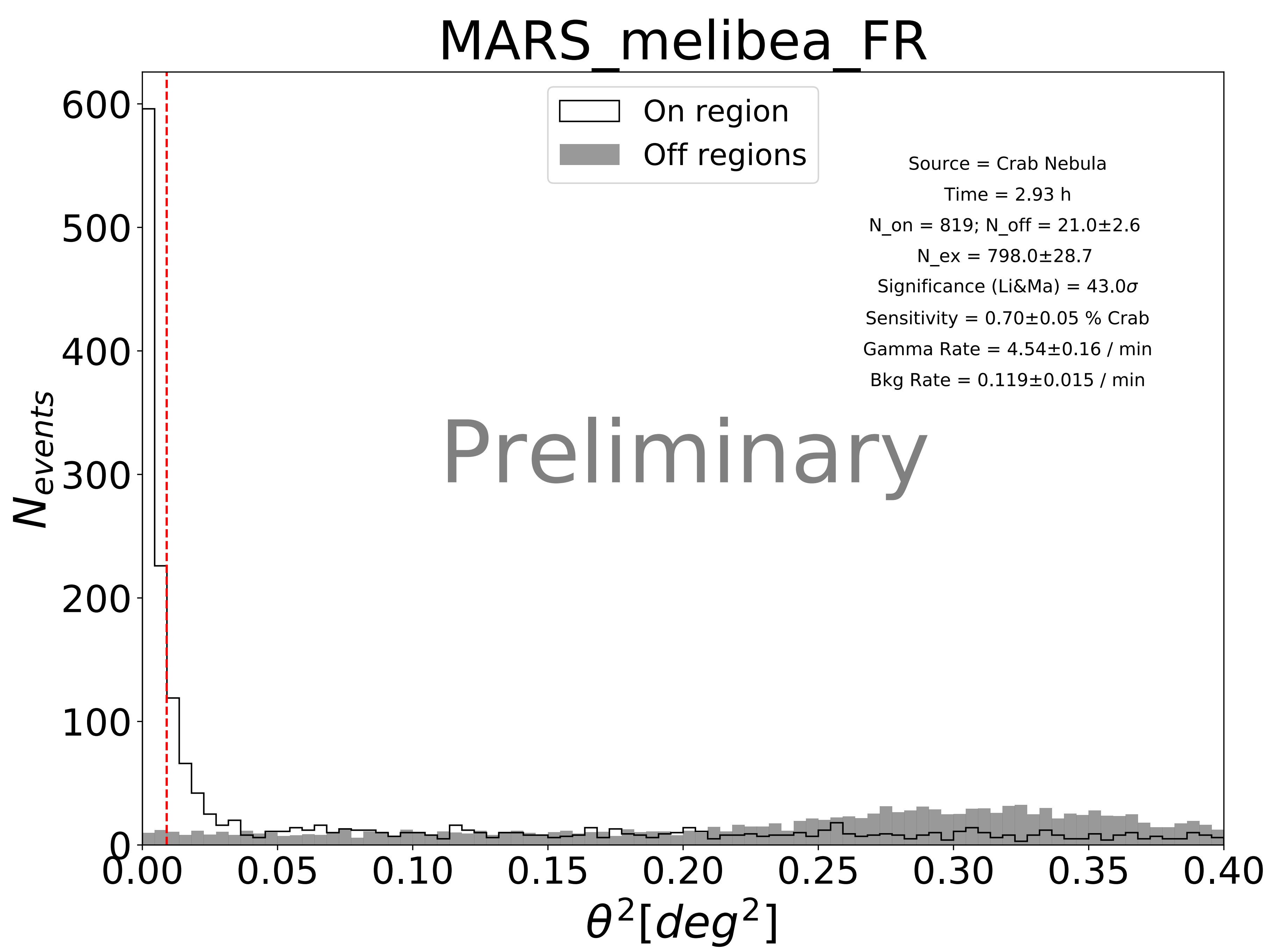




T. Miener et al.

Crab Nebula analysis with *CTLearn*





Analysis	N_{on}	N_{off}	N_{ex}	γ rate [/min]	bkg rate [/min]	Sen. [% Crab]	Sig. (Li&Ma)
MARS – FR	819	21.0 ± 2.6	798.0 ± 28.7	4.54 ± 0.16	0.119 ± 0.015	0.70 ± 0.05	43.0σ
CTLearn – FR (raw)	629	23.3 ± 3.1	605.7 ± 25.3	3.45 ± 0.14	0.133 ± 0.018	0.97 ± 0.08	36.5σ
CTLearn – FR (cleaned)	844	22.0 ± 2.7	822.0 ± 29.2	4.68 ± 0.17	0.125 ± 0.015	0.69 ± 0.05	43.6σ
MARS – LE	3579	679.0 ± 15.0	2900.0 ± 61.7	16.49 ± 0.35	3.861 ± 0.086	1.09 ± 0.03	61.1σ
CTLearn – LE (raw)	2730	673.7 ± 20.0	2056.3 ± 56.0	11.70 ± 0.32	3.832 ± 0.114	1.53 ± 0.05	47.5σ
CTLearn – LE (cleaned)	3536	680.7 ± 15.1	2855.3 ± 61.3	16.24 ± 0.35	3.872 ± 0.086	1.11 ± 0.03	60.4σ

T.Miener et al., PoS(ADASS2021a)