

Deep Learning techniques for reconstruction on ASTRI Mini-Array Monte Carlo data

S. Lombardi,^{a,b,*} F. Visconti,^{a,b} and M. Mastrogiuseppe^a for the ASTRI Project^c

^aINAF, Osservatorio Astronomico di Roma, Via Frascati 33, I-00078 Monte Porzio Catone (Roma), Italy

^bASI, Space Science Data Center, Via del Politecnico s.n.c., I-00133 Roma, Italy

^c<http://www.astri.inaf.it/en/library/>

E-mail: saverio.lombardi@inaf.it, francesco.visconti@inaf.it

The interaction of gamma rays and cosmic rays with the Earth's atmosphere initiate air showers that, in turn, induce the emission of Cherenkov photons detectable by ground-based Imaging Atmospheric Cherenkov Telescopes (IACTs). Any data analysis software for gamma-ray astronomy with IACTs requires an essential component to discriminate the nature of the primary particle, as well as to reconstruct its energy and arrival direction. In this field, the standard reconstruction approach is to use supervised machine learning techniques, mostly based on decision trees or Random Forest, which build models by training on simulated data using image and stereoscopic parameters as input features. This approach can be overcome by deep learning techniques, directly operating on pixelated camera images recorded by the array telescopes as input to models. In this way, all available information per each shower image can potentially be exploited for reconstruction, without relying solely on derived parameters. We evaluated some deep learning techniques on Monte Carlo simulated data of the ASTRI Mini-Array, an array of nine dual-mirror 4-m class IACTs under deployment at the *Observatorio del Teide* (Tenerife, Spain), sensitive to gamma-ray radiation in the 1–200 TeV energy range. In this contribution we present how deep learning algorithms such as convolutional neural networks can be used to reconstruct events acquired by the ASTRI Mini-Array; we will first describe the analysis work flow and introduce the architectures, and then compare the performance obtained with the new reconstruction methods with that of standard method.

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



*Speaker

1. Introduction

Convolutional Neural Networks (CNNs) have made a significant impact in the field of computer science [1], particularly in the realm of computer vision. This breakthrough can be attributed not only to the formal codification of their building blocks during the latter half of the previous century, but also to the advancements in hardware technology. Notably, the emergence of the CUDA [2] programming language by Nvidia in 2007 enabled the utilization of Graphical Processing Units (GPUs) for general-purpose programming. Consequently, the scientific community began to develop algorithms and effectively employ these techniques to address a wide range of interdisciplinary challenges.

In recent years, there has been an exceptional surge of interest within the machine learning community, driven by the remarkable performance of generative models in both image and text domains. This momentum is ongoing, as the field continues to witness the development of increasingly impressive models. Additionally, novel concepts and network components, such as the attention mechanism [3] and transformers [4], originally designed for Natural Language Processing (NLP), have successfully found applications in computer vision problems as well. While initially emerging from industry rather than academia, these innovations have demonstrated their versatility in general-purpose contexts, prompting their rapid adoption by the scientific community across various domains of knowledge.

Different deep learning techniques and approaches are also being investigated in the context of very high-energy (VHE) astrophysics [5] and, in particular, of Imaging Atmospheric Cherenkov telescopes (IACTs) arrays [6] to reconstruct the physical properties of the recorded events [7–9]. The ultimate aim is to overcome the performance of standard, consolidated reconstruction methods implemented in the typical IACT standard analysis chains (see e.g. [10, 11]) by exploiting all available information per each shower image [12].

In this work we present a first preliminary investigation of some deep learning techniques in the context of the ASTRI Mini-Array [13], an array of nine dual-mirror 4-m class IACTs under deployment at the *Observatorio del Teide* (Tenerife, Spain) aimed to observe the gamma-ray sky in the 1–200 TeV energy range. More information on the system and its scientific goals can be found in [14].

2. Analysis work flow

The analysis work flow adopted for the present study can be divided into three stages:

1. *Single-telescope image selection and multi-image input generation*: the main inputs of our analysis work flow are single-telescope cleaned Cherenkov images¹. The input data (FITS) files were taken from the so-called *ASTRI MA Prod2-Teide* production [15], specifically at the Data Level 1b (DL1b) [10]. These files, obtained through the standard ASTRI pipeline (v0.3.1), contain cleaned and parameterized data from all the nine telescopes comprising the ASTRI Mini-Array. We considered diffuse gamma, proton, and electron species with Zenith

¹Details on the two-threshold (L1/L2) two-pass cleaning algorithm implemented in the ASTRI pipeline can be found e.g. in [10]. For the present analysis, a value of 10 pe for L1 and 5 pe for L2 (half of the L1 value) were set.

and Azimuth telescope pointing directions equal to 20° and 0° (i.e. North), respectively. Then, single-telescope cleaned images that passed typical selection criteria² were combined to create multi-image inputs of each Cherenkov event (see Figure 2), for all considered species. These multi-images were then stored and referenced in both a MariaDB database and an on-premise object storage system. The latter serves as a convenient service that can be queried via HTTP requests from any application, and it also functions as a caching system for dvc³ (the data versioning software).

2. *Machine learning pipeline*: an efficient machine learning pipeline based on TensorFlow⁴ libraries was implemented. The pipeline is effectively managed using dvc, which not only facilitates the handling of stages through a simple yaml⁵ file but also enables data versioning. By leveraging the robust mechanisms of git, one can easily switch between different versions of not only the code but also the data. For each considered deep learning method (see Sec. 2.1), look-up-tables (LUTs) for the reconstruction of the Cherenkov events were created from suitable training data samples of multi-image files of diffuse gamma and proton species. The LUTs were applied then to the remaining (testing) data samples of each species. The machine learning pipeline outputs are the fully-reconstructed array-wise Cherenkov events with parameters comprising the probability of an event being a gamma particle (*gammaness*), reconstructed energy, and estimated arrival direction⁶.
3. *Reconstruction and performance analysis*: the fully-reconstructed Cherenkov events (of the testing data samples), including reconstructed parameters from all considered deep learning methods (see Sec. 2.1) and those achieved with the standard method (based on Hillas parameters [16] and Random Forest [17]) implemented in the official ASTRI pipeline, were stored in both the supporting database for subsequent offline analysis and the standard ASTRI DL2b [10] (FITS) files. Then, a reconstruction and a performance analysis were performed to assess, on a comparison basis, the effectiveness of each reconstruction method. The reconstruction analysis is aimed to produce *Quality* (Q) factor and *Receiver Operating Characteristic* (ROC) curves, whereas the performance analysis goal is to assess the overall differential sensitivity of the system, as a function of the off-axis.

2.1 Neural network architectures

For this work, several Convolutional Neural Network (CNN) architectures were trained and evaluated using the ASTRI Mini-Array Monte Carlo simulated data samples mentioned in Sec. 2; the most interesting results are obtained via a conventional CNN implementation and a well-known model such as *ResNet50* [18]. The ResNet50 model was enriched with attention module such as

²Namely, the filter cuts applied to single-telescope images of all telescopes for all considering species were: SIZE > 50 [phe], LEAKAGE < 0.1, NUMISLANDS < 2 [10].

³<https://www.dvc.org/>

⁴<https://www.tensorflow.org/>

⁵<https://yaml.org/>

⁶It is worth mentioning that in this work we considered energy reconstruction and arrival direction estimation for the Cherenkov events (belonging to the testing data samples) exclusively from the standard method, obtained through the standard ASTRI pipeline [10]. The investigation of the energy reconstruction and arrival direction estimation achieved with deep learning techniques will be addressed in future work.

the *Convolutional Block Attention Module* (CBAM) [19] and *Squeeze and Excitation* (SE) [20], which emerged as particularly intriguing models also in very high-energy (VHE) astrophysics [8]; to the best of the authors' knowledge though, only the SE mechanism has been tested on Cherenkov data [9], while the CBAM is a novelty in this field.

The Custom CNN, implemented using keras⁷, is designed for image processing tasks and can be used for regression or classification based on the task parameter. The Custom CNN starts with rescaling input multi-images between 0 and 1. It consists of three convolutional layers with ReLU activation. The first and second Conv2D layers have 64 filters with a kernel size of (3, 3), while the last Conv2D layer has 128 filters of the same size. MaxPooling2D layers are added after each Conv2D layer to reduce spatial dimensions and retain important information. The output is flattened and passed to a Dense layer with ReLU activation. A Dropout layer with a rate of 0.3 helps prevent overfitting. The final Dense layer has one unit for regression or a variable number for classification, with a sigmoid activation function for class probabilities.

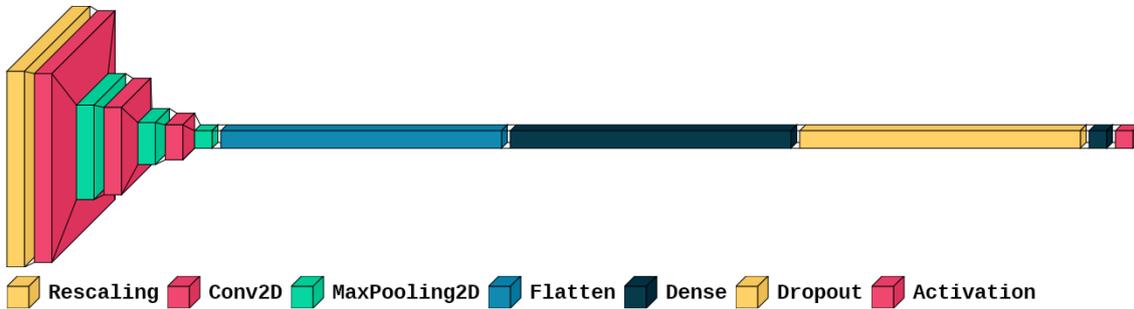


Figure 1: Custom CNN visual representation. It can be seen the squeezing action of the layers reducing the input size up to the flatten layer, which prepare to the fully connected last step.

The CBAM network incorporates the Convolution Block Attention Module (CBAM) to enhance representation. It includes a Channel Attention Module (CAM) that assigns adaptive weights to feature map channels, emphasizing relevant channels. The Spatial Attention Module (SAM) captures spatial dependencies, highlighting important regions. CBAM combines channel attention weights and spatial attention maps through element-wise multiplication, resulting in attention-enhanced feature maps. CBAM is compatible with various CNN architectures and was integrated with ResNet50 in this work.

The SE mechanism enhances CNNs by modeling channel-wise dependencies. It applies global average pooling to capture channel-wise statistics. In the excitation step, a small fully connected network learns channel-wise attention weights based on the squeezed features. Multiplying the attention weights with the original feature maps produces attention-reweighted feature maps. The SE mechanism can be integrated into different CNN architectures, improving their discriminative power: as well as for CBAM, SE has enriched a ResNet50 model in this work.

2.2 Input and output data

The ASTRI Mini-Array is an array of nine Cherenkov telescopes. Each telescope's Cherenkov camera has 37 squared active Photo-Detector Module (PDM) (each of 64 pixels), arranged in a sort

⁷<https://keras.io>

of *cross* layout; the event multi-images (inputs of the machine learning pipeline) were generated setting to black the non existing PDMs in a simplified square layout of 56x56 pixels (rescaling input images between 0 and 1). In order to maximize the information carried by the input for the neural network models, it was decided to use as input the stacking of event images (with a minimum number of images per event equal to two) as seen by single telescopes (see Fig. 2).

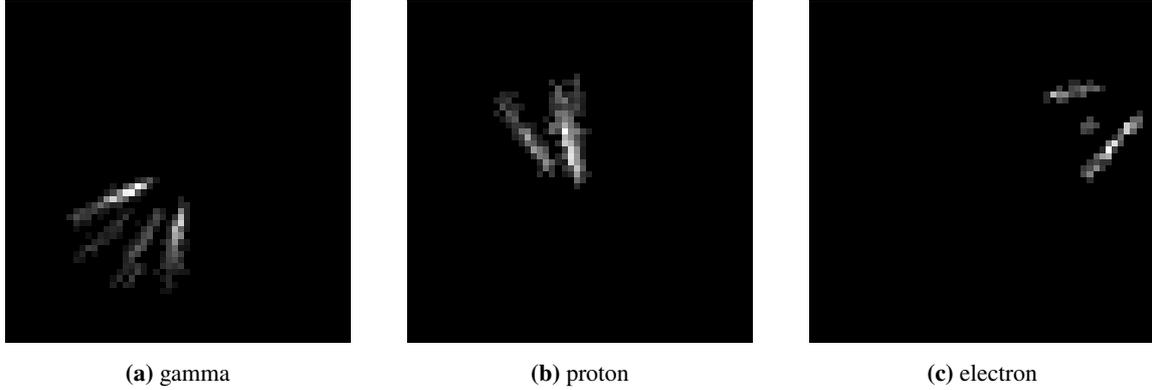


Figure 2: Sample of images of stereoscopic events. Only events with at least two images surviving the single-telescope image cleaning and selection criteria were taken into account in the analysis.

To get the single-telescope images that make up the multi-images, after a calibration step, each single-telescope image was cleaned in order to suppress some of the background noise⁸; then standard quality cuts were applied on Hillas parameters basis, mainly in order to discard images recorded too close to the camera edge ($LEAKAGE < 0.1$) and with small SIZE ($SIZE < 50$ [phe]).

To build the training data samples, a logarithmic binning in SIZE was applied, and then the same number of diffuse gamma and proton events⁹ were randomly picked in each bin: this pruning step is a quite important procedure when doing Random Forest analysis which is expected to have a lighter impact when using deep learning techniques; the aim, however, was to get a direct comparison with the standard analysis: in this way, all the models used were trained and tested on the same number of events, and this is also important for the performance analysis (Sec. 4).

All the models were trained with the same gamma/proton dataset, where $\sim 1.8 \times 10^5$ stereoscopic events (multi-images for deep learning methods), corresponding to $\sim 6 \times 10^5$ single-telescope images, were selected for each population, equalizing their numbers in each SIZE bin. The final machine learning reconstruction models to be compared were dubbed: standard method (STD), Custom CNN (CUSTOM), Convolutional Block Attention Module (CBAM), and Squeeze and Excitation (SE). For each of them, the fully-reconstructed Cherenkov events (belonging to the testing data samples) were computed and used for the reconstruction (Sec. 3) and performance analysis (Sec. 4).

3. Reconstruction analysis

The reconstruction analyses were divided in three (reconstructed) energy bins: $E \in [10^{-0.5}, 10^{0.5}]$ TeV, $E \in [10^{0.5}, 10^{1.5}]$ TeV, and $E \in [10^{1.5}, 10^{2.5}]$ TeV. For a statistical comparison of the results,

⁸Details of the calibration and cleaning algorithms can be found in [10], where the standard ASTRI Mini-Array data processing pipeline is described in details.

⁹It is worth mentioning that the electron population was used only at the inference stage.

the ROC curves with their *Area Under the Curve* (AUC) score were computed (see Fig. 3 and Tab. 1), together with the Q factor, defined as $Q = \frac{\epsilon_\gamma}{\sqrt{\epsilon_h}}$, where ϵ_γ is the fraction of well classified gammas, and ϵ_h is the fraction of protons misclassified as gammas (see Fig. 4).

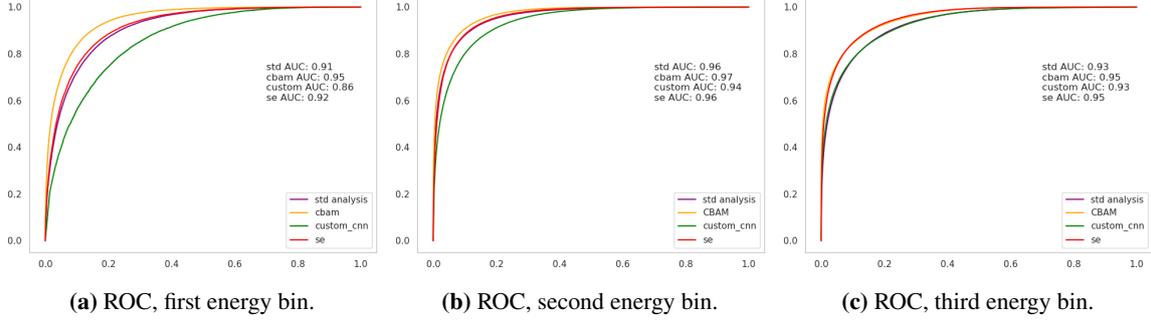


Figure 3: ROC curves for the four considered reconstruction methods, in each energy bin.

Energy bin	STD	CUSTOM	CBAM	SE
$E \in [10^{-0.5}, 10^{0.5}]$	0.91	0.86	0.95	0.92
$E \in [10^{0.5}, 10^{1.5}]$	0.96	0.94	0.97	0.96
$E \in [10^{1.5}, 10^{2.5}]$	0.93	0.93	0.95	0.95

Table 1: AUC score comparison for the four considered methods, in each energy bin.

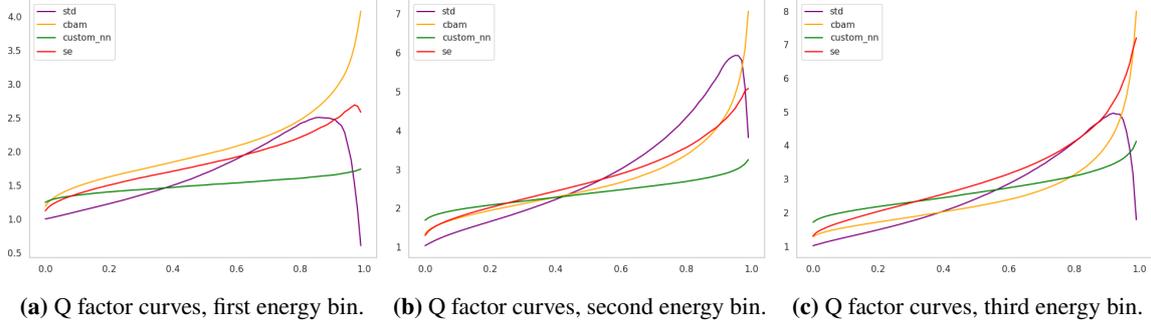


Figure 4: Q factor curves for the four considered reconstruction methods, in each energy bin.

While ROC and AUC results are quite similar, with the exception of the CUSTOM method, which seems to provide lower discriminating power than all other methods, Q factor curves suggest an improvement in discriminating power over STD method for SE and, in particular, CBAM method. Further considerations can be drawn at the end of section 4, where the performance analysis is discussed.

4. Performance analysis

For all considered reconstruction methods, we derived the point-like source differential sensitivity of the ASTRI Mini-Array in 50 hours for five off-axis bins from 0° to 5° in the reconstructed

energy range between $10^{-0.5} \simeq 0.3$ TeV and $10^{2.5} \simeq 300$ TeV (using five logarithmic energy bins per decade). To do that, we used the standard software tools and procedure currently implemented in the ASTRI pipeline (more details can be found in [15]; see also [21]). Since we found a common, relative trend among the different methods in all the different off-axis bins, we considered the averaged off-axis differential sensitivity in the entire range from 0° to 5° for better comparison. Resulting sensitivity curves are shown in Fig. 5. The ratios between the sensitivity obtained by CUSTOM, CBAM, and SE methods and that of STD are also shown. While the CUSTOM and SE methods provide differential sensitivities that are generally worse than those obtained with the standard method below $\simeq 10^{1.7} \simeq 50$ TeV and slightly better above that energy, the CBAM method outperforms the STD one in the whole energy range. In particular, for energies above $\simeq 10^{1.0} \simeq 10$ TeV the CBAM method reaches an averaged gain in performance up to 40%. This is an encouraging result that certainly deserves further investigations.

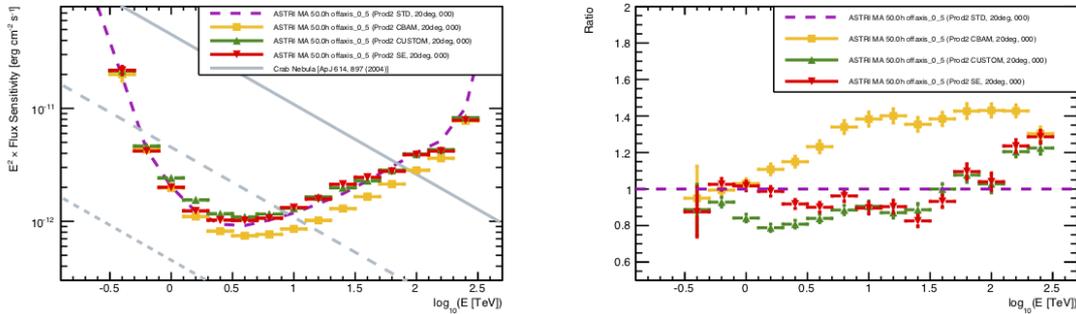


Figure 5: *Left:* ASTRI Mini-Array averaged off-axis (from 0° to 5°) point-like source differential sensitivity (at a zenith angle of 20° and an azimuth angle of 0°) for four different methods: STD (blue dashed line), CBAM (orange points), CUSTOM (green points), SE (red points). The Crab Nebula spectral model (grey lines) is taken from [22]. *Right:* Ratios between the sensitivity obtained by CUSTOM, CBAM, and SE methods and that of STD. The ratios are calculated so that higher values correspond to better performance.

5. Summary and outlook

In this contribution we presented a first preliminary investigation of some deep learning techniques in the context of the ASTRI Mini-Array. For our study, we exclusively made use of Monte Carlo simulated data. Although some interesting results were found, the work is still in progress. First step the authors will take, will be that of modifying the input: no more *stacked* multi-images of the same Cherenkov event as seen from different telescopes will be used, but *tiled* multi-images, following the recent tendency in scientific literature [9, 23, 24], which seems to provide better results, particularly for energy and direction reconstruction, not considered in this work. In parallel, temporal information will be added as another channel to the image signal, in order to enrich event information to be given as input to the networks. Another activity will be to let the machine learning pipeline be more and more efficient, with faster data batches loading; a move to PyTorch¹⁰ is ongoing and will be compared with the actual TensorFlow pipeline, as soon as it

¹⁰<https://pytorch.org/>

is ready. The ultimate aim of our efforts is to provide an alternative, consolidated reconstruction pipeline based on improved deep learning methods for the reduction of both Monte Carlo simulated and real data of the ASTRI Mini-Array.

Acknowledgments

This work was conducted in the context of the ASTRI Project. We gratefully acknowledge support from the people, agencies, and organisations listed here: <http://www.astr.iinaf.it/en/library/>. We acknowledge financial support from the ASI-INAF agreement n. 2022-14-HH.0. This work is (partially) supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU. This paper went through the internal ASTRI review process.

References

- [1] Alzubaidi, L. et al., *J. Big Data* 8, 53 (2021)
- [2] Nickolls, J. *IEEE Hot Chips 19 Symposium (HCS)*, Stanford, CA, USA, 1-12 (2007).
- [3] Vaswani, A. et al., *Advances in neural information processing systems*, 30 (2017).
- [4] Khan, S. et al., *ACM computing surveys (CSUR)*, 54(10s), 1-41 (2022).
- [5] Aharonian, F., Bergström, L. and Dermer, C., *Astrophysics at Very High Energies*, Springer (2013).
- [6] de Naurois, M., & Mazin, D., *Comptes Rendus Physique*, 16(6-7), 610 (2015).
- [7] Shilon, I. et al., *Astroparticle Physics*, 105, 44-53 (2019).
- [8] Miener, T. et al, *Proc. 37th ICRC*, Berlin, Germany, PoS(ICRC2021)730 (2021).
- [9] Jacquemont, M. et al., *Proc. CBMI*, Lille, France, 1-6 (2021).
- [10] Lombardi, S. et al., *Proc. SPIE*, 10707, 107070R (2018).
- [11] Zanin, R. et al., *Proc. 33rd ICRC*, Rio de Janeiro, Brazil, *J.Phys.*44, 5 (2013).
- [12] Visconti, F., *arXiv:2212.10281* (2022).
- [13] Scuderi, S. et al., *Journal of High Energy Astrophysics*, 35, 52–68 (2022).
- [14] Giuliani, A. et al., *Proc. 38th ICRC*, Nagoya, Japan, PoS(ICRC2023)892 (2023).
- [15] Lombardi, S. et al., *Proc. 37th ICRC*, Berlin, Germany, PoS(ICRC2021)884 (2021).
- [16] Hillas, A. M., *Proc. 19th ICRC*, la Jolla, 3, 445 (1985).
- [17] Breiman, L., *Machine Learning*, 45(1), 5-32 (2001).
- [18] He, K. et al, *Proc. of the IEEE conference on computer vision and pattern recognition*, 770-778 (2016).
- [19] Woo, S. et al., *Proc. of the European conference on computer vision (ECCV)*, 3-19 (2018).
- [20] Hu, J. et al., *Proc. of the IEEE conference on computer vision and pattern recognition*, 7132-7141 (2018).
- [21] Pintore, F. et al., *Proc. 38th ICRC*, Nagoya, Japan, PoS(ICRC2023)722 (2023).
- [22] Aharonian, F. et al., *ApJ* 614, 897 (2004).
- [23] Miener, T. et al., *Proc. Gamma2022*, Barcelona, Spain, *arXiv:2211.16009* (2022).
- [24] Songshaptak, D. et al., *Phys. Rev. D* 107, 083026 (2023).