Sensipy: simulate gamma-ray observations of transient astrophysical sources

Jarred G. Green[1,2], Barbara Patricelli[3,4], Antonio Stamerra[4,5], Monica Seglar-Arroyo[6]

[1] Max-Planck-Institut für Physik, Garching, Germany
[2] Technische Universität München, Germany
[3] University of Pisa, Italy
[4] INAF - Osservatorio Astronomico di Roma, Italy
[5] Cherenkov Telescope Array Observatory, Bologna, Italy
[6] Institut de Fisica d'Altes Energies (IFAE), Bellaterra (Barcelona), Spain

Summary

We present `sensipy`, an open-source Python toolkit for simulating observations of transient astrophysical sources, particularly in the high-energy (HE, keV-GeV) and very-high-energy (VHE, GeV-TeV) gamma-ray ranges. The most explosive events in our universe are often short-lived, emitting the bulk of their energy in a relatively narrow time window. Due to often rapidly fading emission profiles, understanding how and when to observe these sources is crucial both to test theoretical predictions and efficiently optimize available telescope time.

The information extracted from the tools included in `sensipy` can be used to help astronomers investigate the detectability of sources considering various theoretical assumptions about their emission processes and mechanisms. This information can further help to justify the feasibility of proposed observations, estimate detection rates (events/year) for various classes of sources, and provide scheduling insight in realtime during gamma-ray and multi-messenger observational campaigns.

Statement of need

The need for a toolkit like `sensipy` became clear while attempting to estimate the detectability of VHE counterparts to gravitational wave (GW) signals from binary neutron star mergers (BNS) with the upcoming Cherenkov Telescope Array Observatory (CTAO). During development, it became apparent that the included tools could be applied not only to VHE counterparts of BNS mergers, but also to other transient sources like gamma-ray bursts (GRBs), active galactic nuclei flares, novae, supernovae, and more.

Between GW, neutrino, optical, and space-based gamma-ray experiments, thousands of low-latency alerts are sent out to the greater community each year (Abac et al. 2025; Kienlin et al. 2020; Abbasi et al. 2023). However, very few of these events actually result in detections in the VHE gamma-ray regime. This is due to many factors, including the rapid decay of fluxes, delay in telescope repointing, uncertainty on the sky localization of the source, and observatory duty cycles. In the face of these challenges, `sensipy` aims to help answer the following questions for gamma-ray astronomers interested in optimizing their follow-up campaigns:

- Given a theoretical emission model, what are the detection possibilities with a given instrument?
- How much observing time is needed to detect a source given a delay in starting observations?
- At what significance level is a source detectable given a certain observation time?
- How long does a source remain detectable after the onset of emission?
- How can intrinsic source properties (e.g. distance, flux) and observing conditions (e.g. latency, telescope pointing) affect detectability?
- How can these results for catalogs of simulated events inform follow-up strategies in realtime?

State of the field

Currently, `gammapy` is the largest player in the field of gamma-ray astrophysics, providing a substantial set of tools designed for the high-level analysis of data from many major future and currently-operating observatories (Donath et al. 2023). We make use of an applicable set of `gammapy` and `astropy` primitives under the hood, such that experienced users do not have to learn new APIs when beginning work with `sensipy`. Given that, the decision to develop `sensipy` as a standalone package is twofold. Firstly, our goal is not to participate directly in the analysis of telescope data, but to provide a set of simulation tools which can help to plan observation campaigns for these observatories. Secondly, as we provide methods which can be integrated directly into telescope control systems, it is important to keep `sensipy` focused, lightweight, and modular.

Software design

The two main inputs to any `sensipy` pipeline are:

- an instrument response function (IRF), which describes how a telescope performs under specific observing conditions
- intrinsic time-dependent emission spectra for a source, which can be provided in either a FITS or CSV format

Given these inputs, `sensipy` builds upon primitives provided by `astropy` and gammapy to provide the main functionalities outlined below (The Astropy and Price-Whelan Collaboration et al. 2022; Donath et al. 2023). In addition, mock datasets are provided with working code examples, and batteries are included for easy access to publicly-available IRFs, e.g. (Cherenkov Telescope Array Observatory and Consortium 2021).

Design philosophy

Most users already come to sensipy with their own theoretical models at hand, so providing clear APIs with simple and speedy onboarding is front-of-mind during development. The documentation is focused on small and self-contained quick-start examples along with every feature, so that users can directly begin with code-blocks relevant to their problem.

In addition, because this package can also be built directly into telescope control software in order to provide realtime insights into observational campaigns, we chose to organize sensipy into a number of smaller modules that can be individually imported, with each module in turn only calling upon the bare set of primitives needed. Each of the modules and their functionalities are briefly described below.

Sensitivity Curve Calculation with sensipy.sensitivity

Sensitivity curves represent the minimum flux needed to detect a source at a given significance (usually 5σ) given an exposure time t_{exp}. Such curves are often used to compare the performances of different instruments, and sensipy can produce them in two flavors: integral and differential sensitivity curves. The sensitivity itself depends heavily on the rapidly-changing spectral shape of an event, which itself may be highly affected by distance due to the extragalactic background light (EBL). All of these factors are automatically taken into account.

Simulating Observations with sensipy.source

This class addresses the fundamental question: if we begin observations with a latency of $t_L = X$ min after an alert, what observation time is required in order to achieve a detection? In addition, the class can also determine the inverse: given an observation time, at what significance can a source be detected? Given that the user has already calculated the sensitivity curve for an event, sensipy can determine if the source is actually detectable, given t_L. When detectable, the exposure time necessary for detection is also calculated.

Working with large catalogs with sensipy.detectability

sensipy can further estimate the overall detectability of entire classes of objects, given a catalog or survey of simulated events under various conditions.

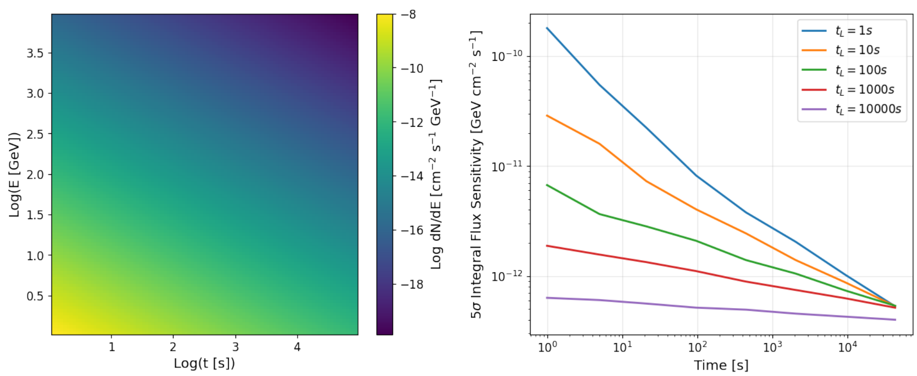

Figure 1: A 2-D representation of the intrinsic time-dependent VHE gamma-ray flux for an example transient event (left) and the corresponding integral flux sensitivity of CTAO for a source with this spectrum (right). The sensitivity curves are calculated for different latencies (t_L) after the event onset.

By performing and collating a large number of observation simulations for various events and latencies t_L, the toolkit can help produce visualizations which describe the optimal observing conditions for such events.

Realtime applications with `sensipy.followup`

Tables of observation times can also be used as lookup tables (LUTs) during telescope observations in order to plan observation campaigns. For example, the following workflow can be implemented within `sensipy`:

1. a catalog of simulated spectra is processed with the above pipeline considering various observation conditions, and a LUT is created
2. a transient alert arrives during normal telescope operation and telescopes begin observing the event position with a latency of t_L
3. the LUT is filtered and interpolated in realtime in order to quickly calculate an informed estimate on the exposure time needed for a detection

Follow-ups of poorly localized events

In addition, the functions included in `sensipy.followup` may be used in tandem with scheduling software like `tilepy` for the realtime follow-up of poorly-localized events, including GRB, GW, and neutrino alerts (Seglar-Arroyo et al. 2024). These scheduling tools create an optimized list of telescope pointings on the sky, while `sensipy` is used simultaneously to optimize the exposure time needed at each new pointing.

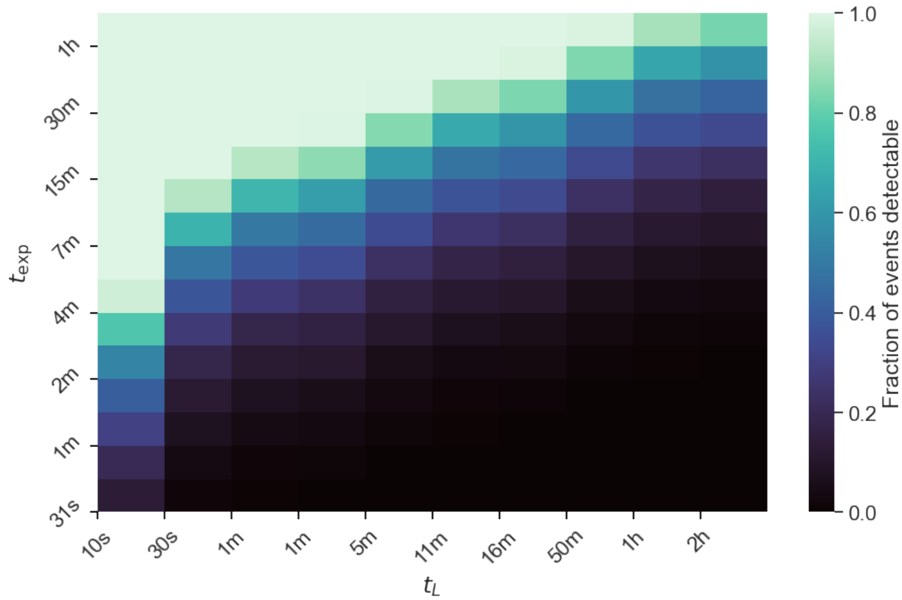

Figure 2: Detectability heatmap produced with `sensipy`. Given a large catalog of transient events, this `sensipy` heatmap shows what fraction are potentially detectable given a specific observation time t_{exp} and latency t_L since the event onset.

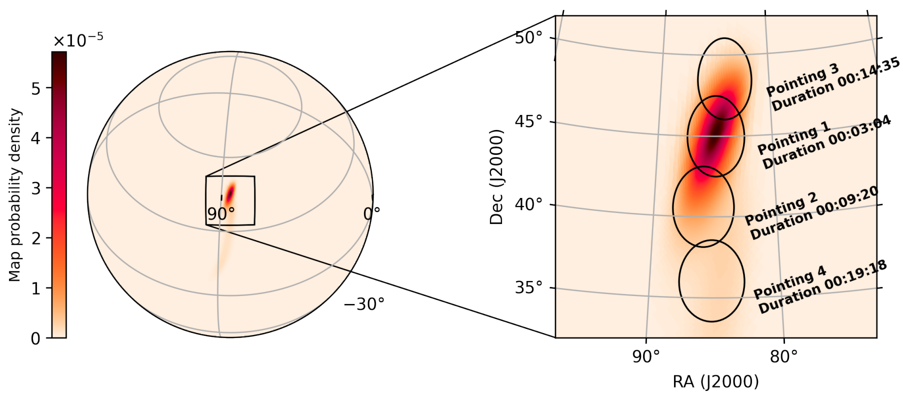

Figure 3: A follow-up coverage map for an example GW event (S250704ab). The ordering of pointings is calculated by `tilepy` and the optimal observing time at each pointing by `sensipy`.

Research impact statement

The `sensipy` package has been under development for the past four years, and in 2025 the community began to grow rapidly as the package reached maturity. The software has been adapted by members of the CTAO Collaboration for a number of applications, including the official evaluation of the prospects of GW follow-up campaigns with the observatory. Numerous talks at major conferences in the field have included contributions created with `sensipy` as part of the main results (Patricelli et al. 2022; Green et al. 2024; Carosi et al. 2025; Seglar-Arroyo et al. 2025). In addition, independent researchers have also made use of the package in order to simulate observational campaigns with other observatories, including the Astrofisica con Specchi a Tecnologia Replicante Italiana (ASTRI) Mini-Array and the Large Array of imaging atmospheric Cherenkov Telescope (LACT) (Macera et al. 2025). Finally, workflows based upon the aforementioned `sensipy.followup` module, together with the `tilepy` package, are being internally tested within the Major Atmospheric Gamma Imaging Cherenkov (MAGIC) Telescopes and Large-Size Telescope (LST) collaborations for GW and GRB alert follow-ups.

AI usage disclosure

AI tools (GitHub Copilot, Grammarly, Cursor Bugbot) were used to proofread documentation and pull requests as well as scaffold tests. Instances where functions or classes were primarily generated with assistance from GitHub Copilot (auto model) are explicitly marked in the Python docstrings. All AI-assisted suggestions were carefully reviewed and approved by the authors of this manuscript. AI tools were not used in the writing of this manuscript in any capacity.

References

Abac, A. G., I. Abouelfettouh, F. Acernese, et al. 2025. "GWTC-4.0: An Introduction to Version 4.0 of the Gravitational-Wave Transient Catalog." *The Astrophysical Journal* 995 (December): L18. https://doi.org/10.3847/2041-8213/ae0c06.

Abbasi, R., M. Ackermann, J. Adams, et al. 2023. "IceCat-1: The IceCube Event Catalog of Alert Tracks." *The Astrophysical Journal Supplement Series* 269 (1): 25. https://doi.org/10.3847/1538-4365/acfa95.

Carosi, Alessandro, Antonio Stamerra, Barbara Patricelli, et al. 2025. "Bridging Gravitational Waves and High-Energy Gamma Rays: Searching for sGRB Afterglows from Compact Binary Coalescences with CTAO." *International Cosmic Ray Conference (ICRC) 2025 (Conference Poster)*. https://indico.cern.ch/event/1258933/contributions/6497376/.

Cherenkov Telescope Array Observatory and Consortium. 2021. *CTAO Instrument Response Functions - Prod5 Version V0.1*. Zenodo. https://doi.org/10.5281/zenodo.5499840.

Donath, Axel, Régis Terrier, Quentin Remy, et al. 2023. "Gammapy: A Python Package for Gamma-Ray Astronomy | Astronomy & Astrophysics (a&a)." *Astronomy & Astrophysics* 678: A157. https://www.aanda.org/component/article?access=doi&doi=10.1051/0004-6361/202346488.

Green, Jarred Gershon, Alessandro Carosi, Lara Nava, et al. 2024. *Chasing Gravitational Waves with the Cherenkov Telescope Array*. https://doi.org/10.22323/1.444.1534.

Kienlin, A. von, C. A. Meegan, W. S. Paciesas, et al. 2020. "The Fourth Fermi-GBM Gamma-Ray Burst Catalog: A Decade of Data." *The Astrophysical Journal* 893 (1): 46. https://doi.org/10.3847/1538-4357/ab7a18.

Macera, Samanta, Biswajit Banerjee, Monica Seglar-Arroyo, and Gor Oganesyan. 2025. "Detection of TeV Emission from Poorly Localized GRBs with Ground Based IACTs, and Future Perspective in the Era of CTAO." *TeV Particle Astrophysics (TeVPA) 2025 (Conference Talk)*, November 4. https://indico.ific.uv.es/event/7986/contributions/28068/.

Patricelli, Barbara, Alessandro Carosi, Lara Nava, et al. 2022. *Searching for Very-High-Energy Electromagnetic Counterparts to Gravitational-Wave Events with the Cherenkov Telescope Array*. https://doi.org/10.22323/1.395.0998.

Seglar-Arroyo, Monica, Halim Ashkar, Mathieu de Bony de Lavergne, and Fabian Schüssler. 2024. "Cross Observatory Coordination with Tilepy: A Novel Tool for Observations of Multimessenger Transient Events." *The Astrophysical Journal Supplement Series* 274 (September): 1. https://doi.org/10.3847/1538-4365/ad5bde.

Seglar-Arroyo, Mónica, Jarred Green, Lara Nava, Antonio Stamerra, Fabian Schüssler, and Barbara Patricelli. 2025. "Bridging Gravitational Waves and High-Energy Gamma Rays: Searching for sGRB Afterglows from Compact Binary Coalescences with CTAO." *TeV Particle Astrophysics (TeVPA) 2025 (Conference Talk)*, November 4. https://indico.ific.uv.es/event/7986/contributions/28071/.

The Astropy and Price-Whelan Collaboration, Adrian M., Pey Lian Lim, Nicholas Earl, et al. 2022. "The Astropy Project: Sustaining and Growing a Community-Oriented Open-Source Project and the Latest Major Release (V5.0) of the Core Package*." *The Astrophysical Journal* 935 (2): 167. https://doi.org/10.3847/1538-4357/ac7c74.