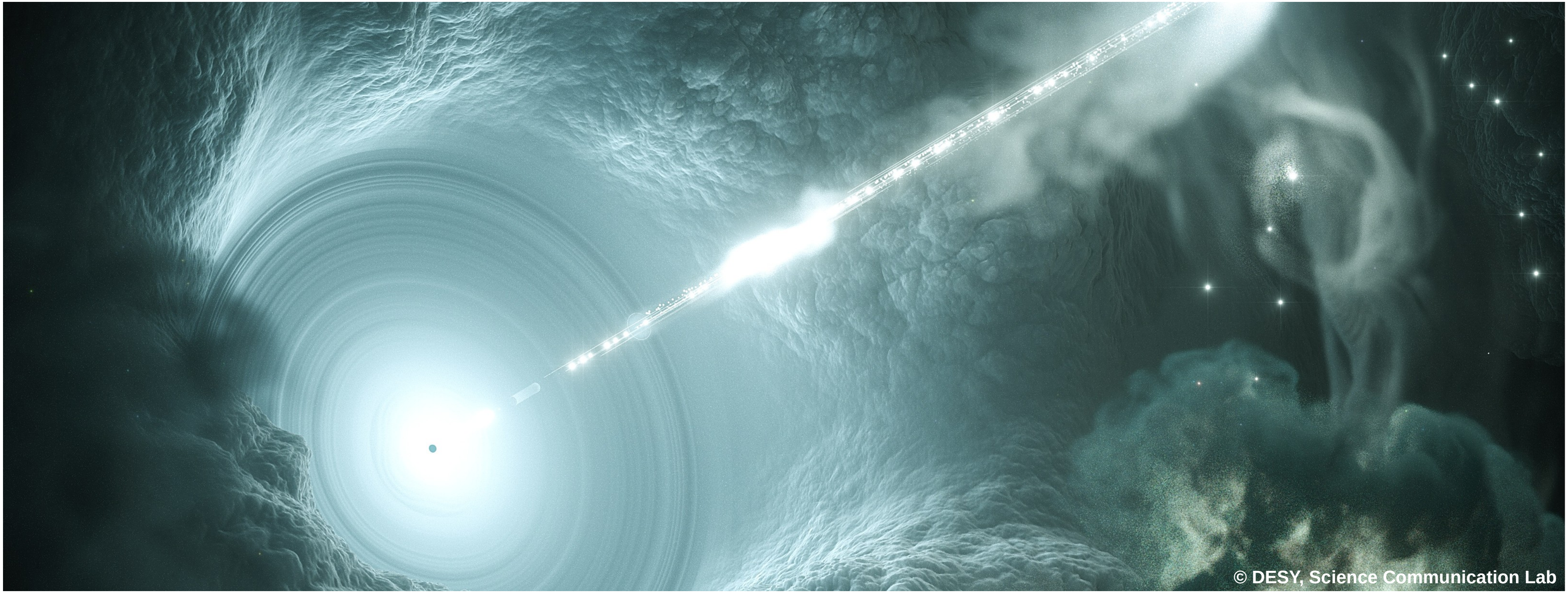
Facilitating multi-messenger modelling using Gammapy

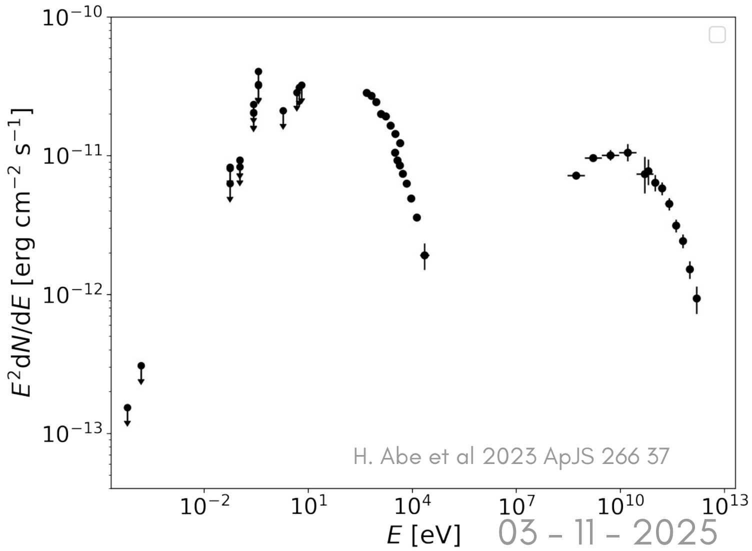
Lea Heckmann, Matteo Cerruti, Bruno Khélifi, Justin Albinet



Current AGN modelling

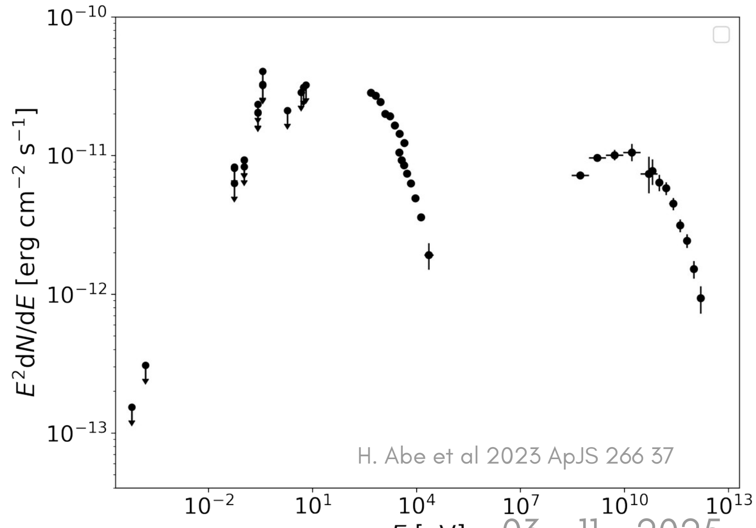
Current AGN modelling

- Combining MWL data on flux point level:



Current AGN modelling

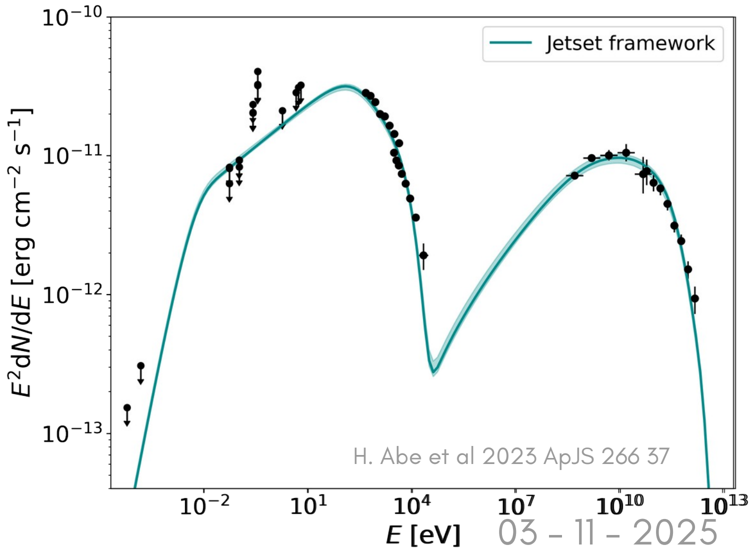
- Combining MWL data on flux point level:
    - We construct flux points for single instruments separately
        - using different frameworks for different instruments
        - assuming simplistic models (e.g. power-law) – single instruments cannot constrain physical models (e.g. SSC)



H. Abe et al 2023 ApJS 266 37

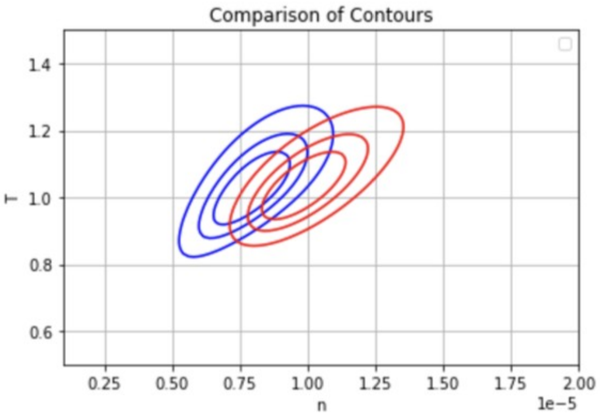
Current AGN modelling

- Combining MWL data on flux point level:
  - We construct flux points for single instruments separately
    - using different frameworks for different instruments
    - assuming simplistic models (e.g. power-law) – single instruments cannot constrain physical models (e.g. SSC)
  - Selection of the physical model based on flux points



Current AGN modelling

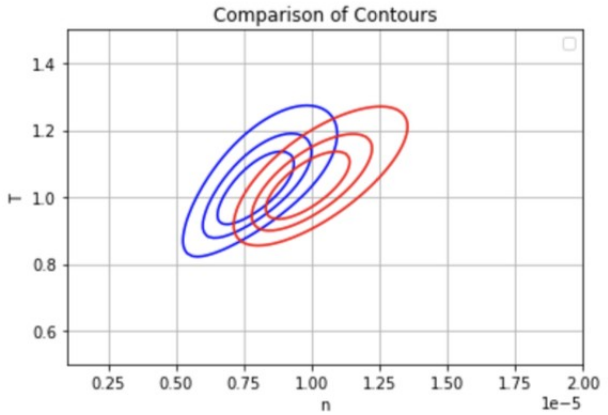
- Combining MWL data on flux point level:
  - We construct flux points for single instruments separately
    - using different frameworks for different instruments
    - assuming simplistic models (e.g. power-law) – single instruments cannot constrain physical models (e.g. SSC)
  - Selection of the physical model based on flux points



- Fitting on flux points adds a bias:
  - Simple test with a blackbody spectrum using X-rays (XSPEC (1999ascl.soft10005A)) shows a bias of 1 sigma

Current AGN modelling

- Combining MWL data on flux point level:
    - We construct flux points for single instruments separately
        - using different frameworks for different instruments
        - assuming simplistic models (e.g. power-law) – single instruments cannot constrain physical models (e.g. SSC)
    - Selection of the physical model based on flux points



- Fitting on flux points adds a bias:
    - Simple test with a blackbody spectrum using X-rays (XSPEC (1999ascl.soft10005A)) shows a bias of 1 sigma

- **Long-term goal: Enable event-based multi-messenger fits in Gammapy**
    - We start with blazars and JetSet as an example here

A new approach

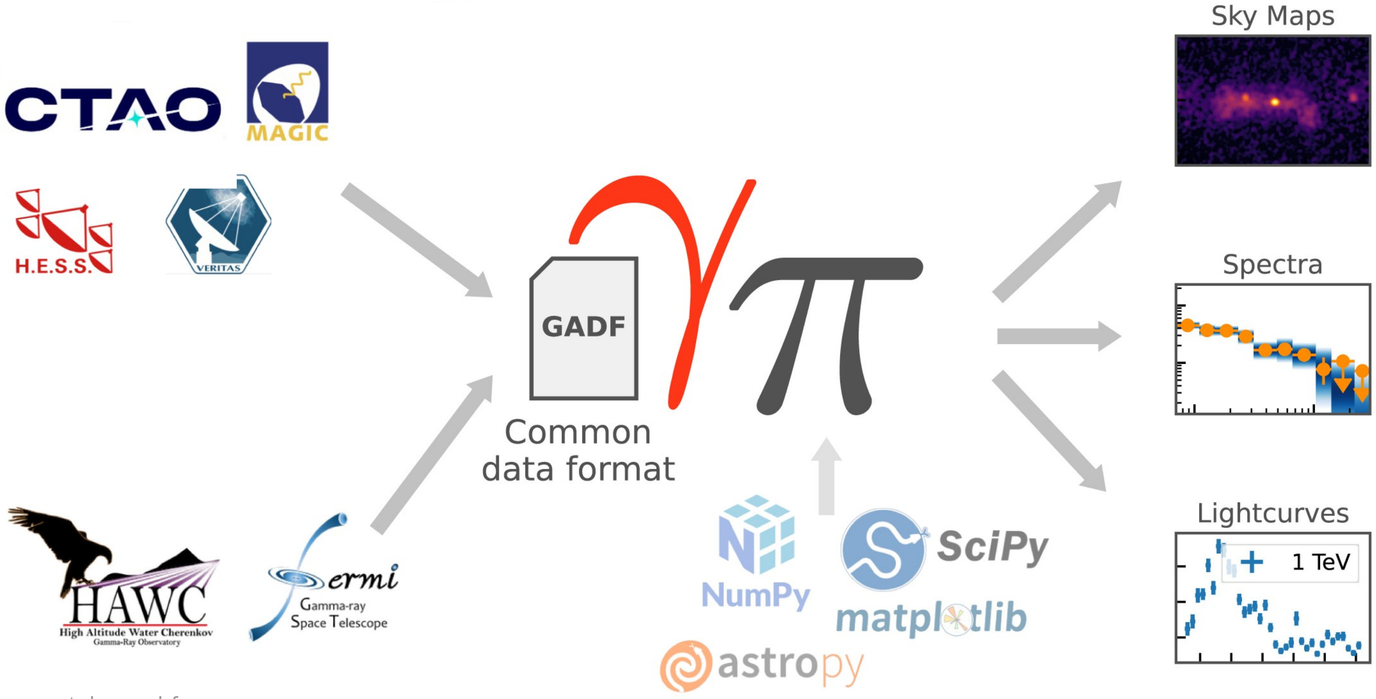
A **Python** package for **gamma-ray** astronomy

- Joined MWL fit on event basis using Gammapy

A new approach

A **Python** package for **gamma-ray** astronomy

- Joined MWL fit on event basis using Gammapy
- Gammapy:
  - Python package for gamma-ray astronomy



Adapted from:
A. Donath et al. 2023, A&A, 678, A157

Gammapy

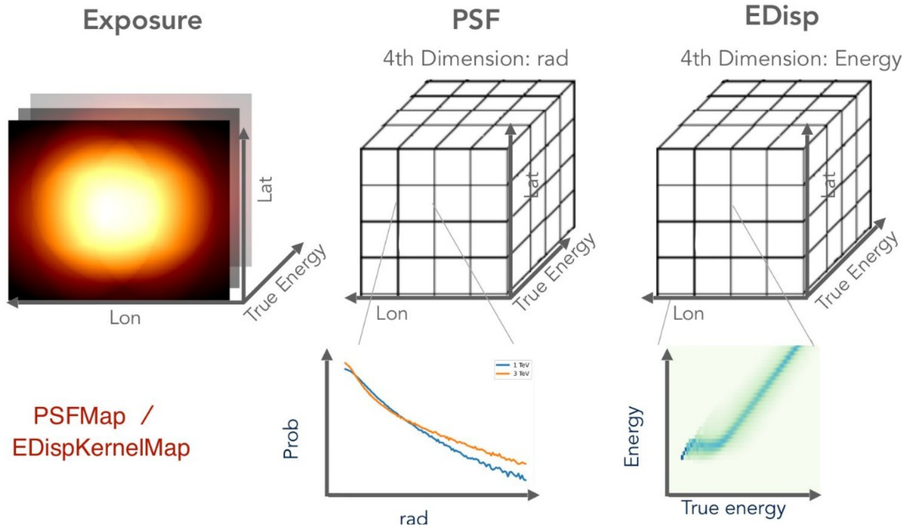
DL3

γ-like events

Time	RA	DEC	Energy	...
...				
...				

+

instruments response functions (IRFs)



Gammapy

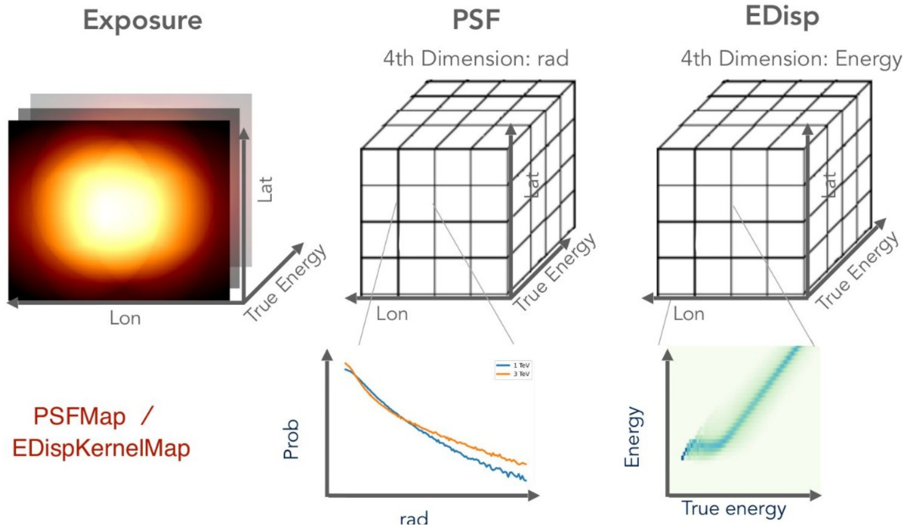
DL3 → DL4

γ-like events

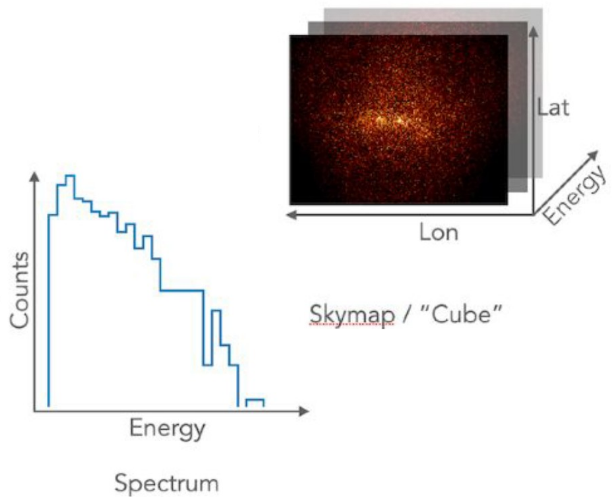
Time	RA	DEC	Energy	...
...				
...				

\+

instruments response functions (IRFs)



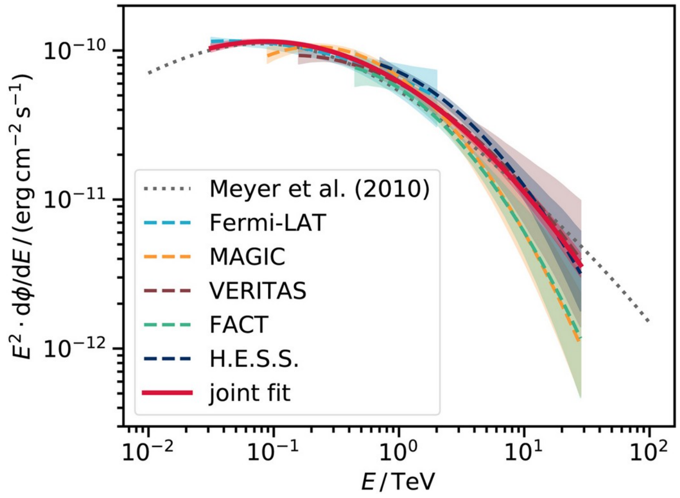
Binned data datasets:



Gammapy & MWL data

- Works nicely for gamma-ray data and joint instrument fits:

C. Nigro et al. 2019, A&A 625, A10



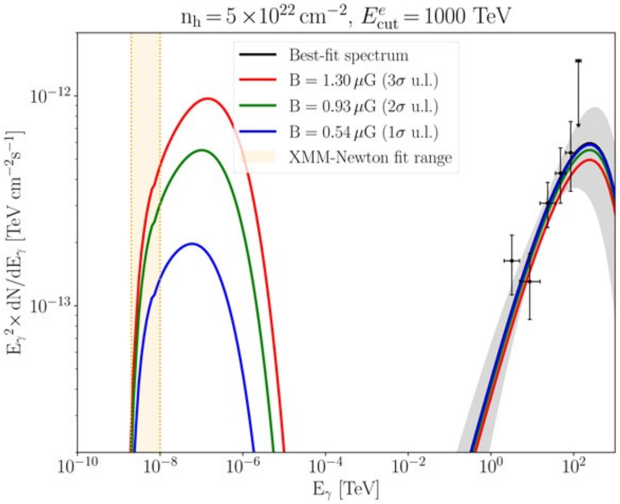
or also see e.g.:
F. Aharonian et al. 2024
A&A 686, A308

Gammapy & MWL data

- Works nicely for gamma-ray data and joint instrument fits:

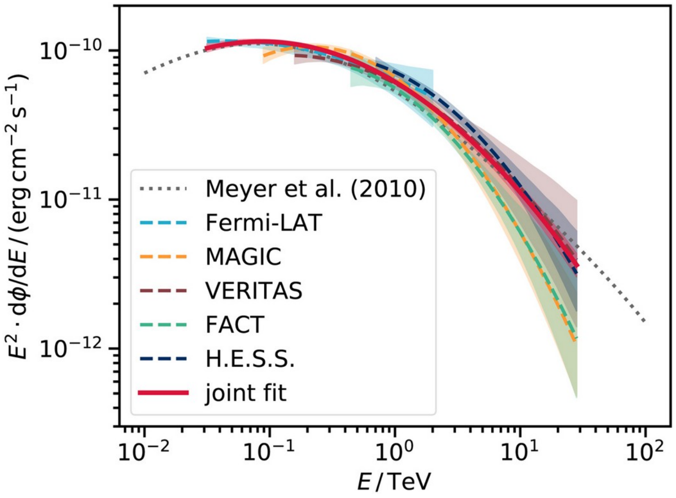
- Extension to X-rays:

  https://github.com/registerrier/gammapy-ogip-spectra



L. Giunti et al. 2022, A&A 667, A130

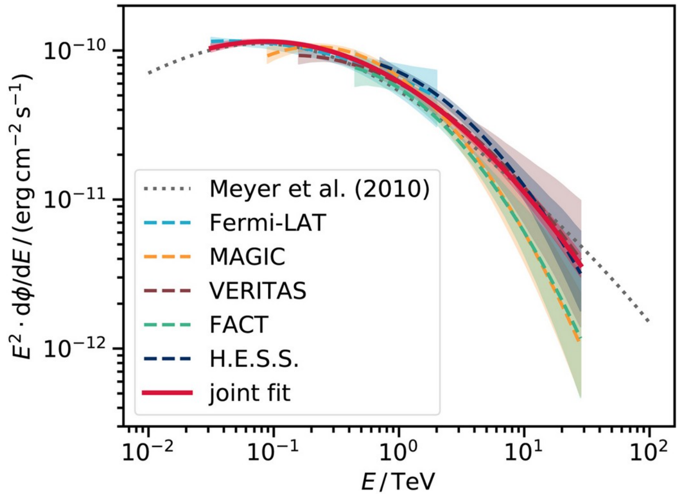
C. Nigro et al. 2019, A&A 625, A10



or also see e.g.:
F. Aharonian et al. 2024
A&A 686, A308

Gammapy & MWL data

C. Nigro et al. 2019, A&A 625, A10

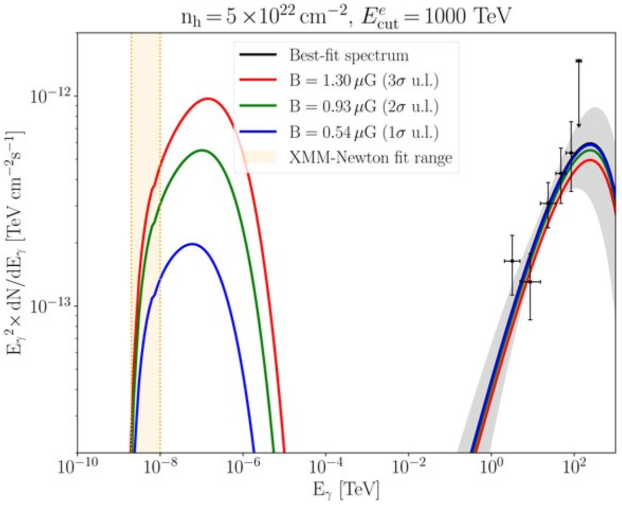


or also see e.g.:
F. Aharonian et al. 2024
A&A 686, A308

- Works nicely for gamma-ray data and joint instrument fits:

- Extension to X-rays:

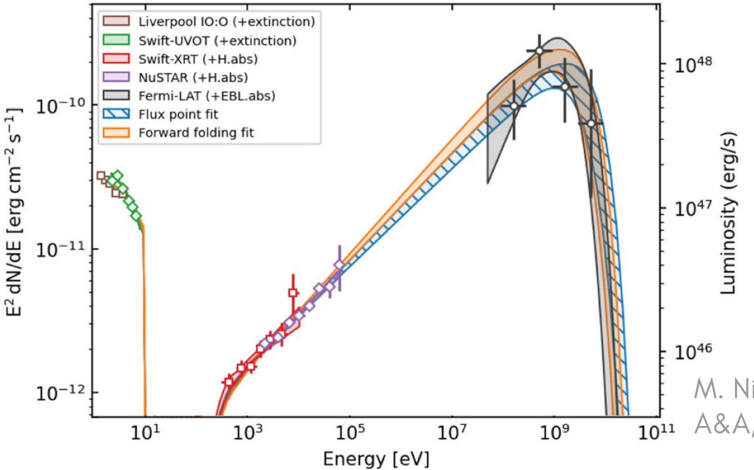
https://github.com/registerrier/gammapy-ogip-spectra



L. Giunti et al. 2022, A&A 667, A130

- Extension to UV/ optical:

https://github.com/mireianievas/gammapy_mwl_workflow



M. Nievas Rosillo et al. 2025,
A&A, 693, A287

New AGN modelling approach with Gammapy

- Building on this gammapy_mwl_workflow and public data

New AGN modelling approach with Gammapy

- Building on this gammapy_mwl_workflow and public data

- IRFs from UV to gamma-rays within Gammapy
  - Real data from Swift: UVOT + XRT, Fermi-LAT (UV, X-rays and gamma-rays)
  - Simulations from CTAO (Very-high-energy gamma-rays)

New AGN modelling approach with Gammapy

- Building on this gammapy_mwl_workflow and public data

- IRFs from UV to gamma-rays within Gammapy
  - Real data from Swift: UVOT + XRT, Fermi-LAT (UV, X-rays and gamma-rays)
  - Simulations from CTAO (Very-high-energy gamma-rays)

- Simulation of MWL data based on these IRFs using Gammapy:
- Physical model:
  - Leptonic Self-Synchrotron (SSC) model with EBL absorption
    - JetSeT (2020ascl.soft09001T) with its Gammapy plugin

+ Systematics for different bands

+ Absorption processes via the sherpa (10.5281/zenodo.593753.) module

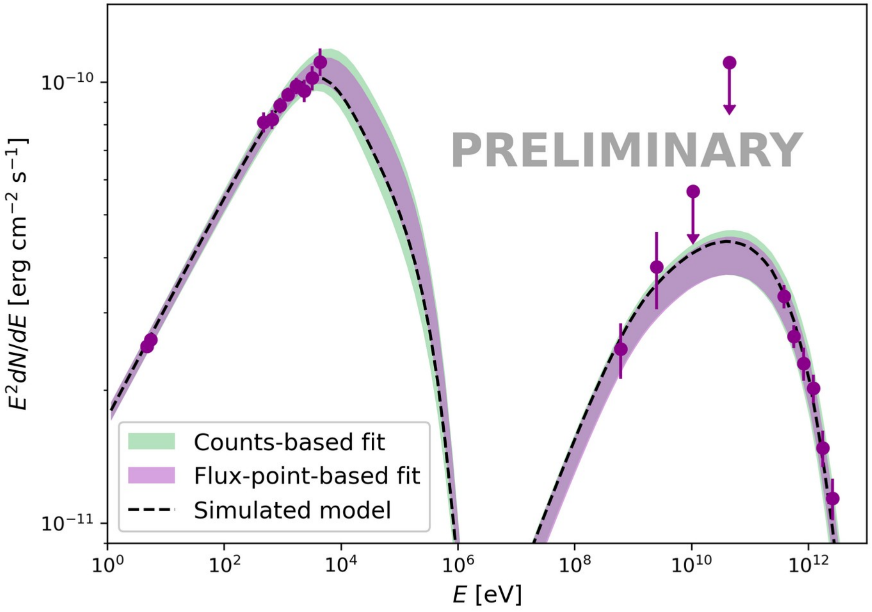
Gammapy & SSC

- We produce 2 MWL Gammapy datasets:
  1) Event data (DL3) set with event lists and IRFs
  2) Flux points data set:
     - Produced for each waveband
     - Assuming simple power laws
     - Mimics common approaches

Gammapy & SSC

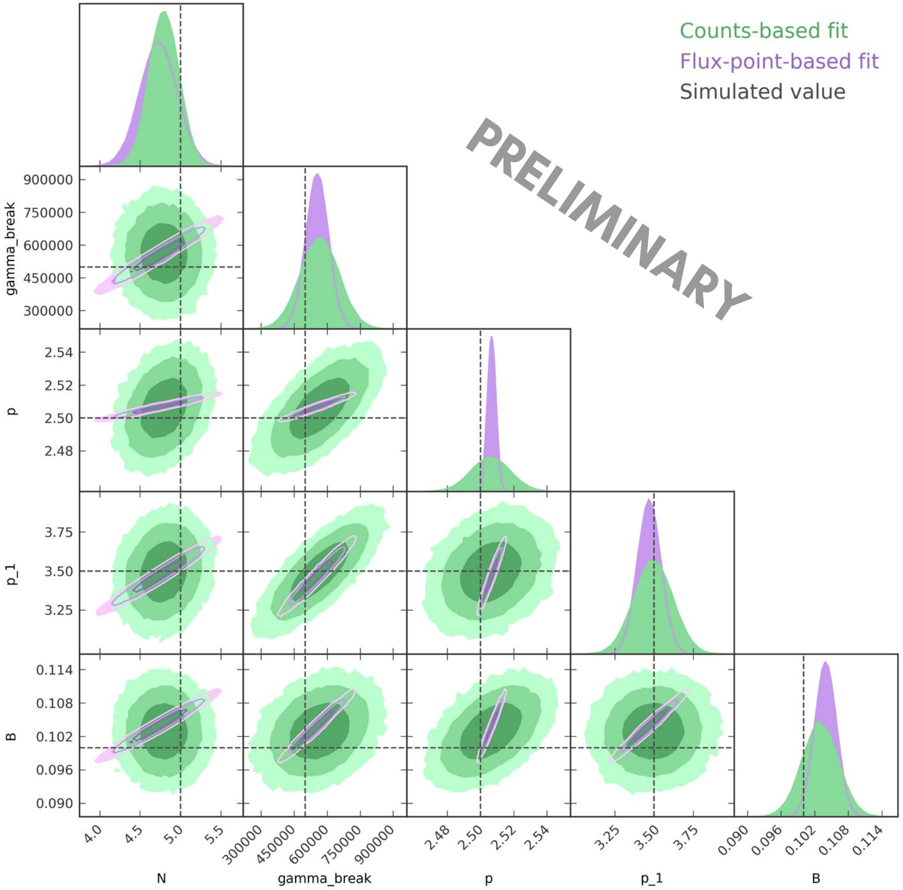
- We produce 2 MWL Gammapy datasets:
    1) Event data (DL3) set with event lists and IRFs
    2) Flux points data set:
        - Produced for each waveband
        - Assuming simple power laws
        - Mimics common approaches

- We fit both datasets with a SSC model using JetSeT



Facilitating multi-messenger modelling using Gammapy

- Works quite well and events fit shows
  - less bias/better error estimation
  - better convergence – up to ~5 free parameters

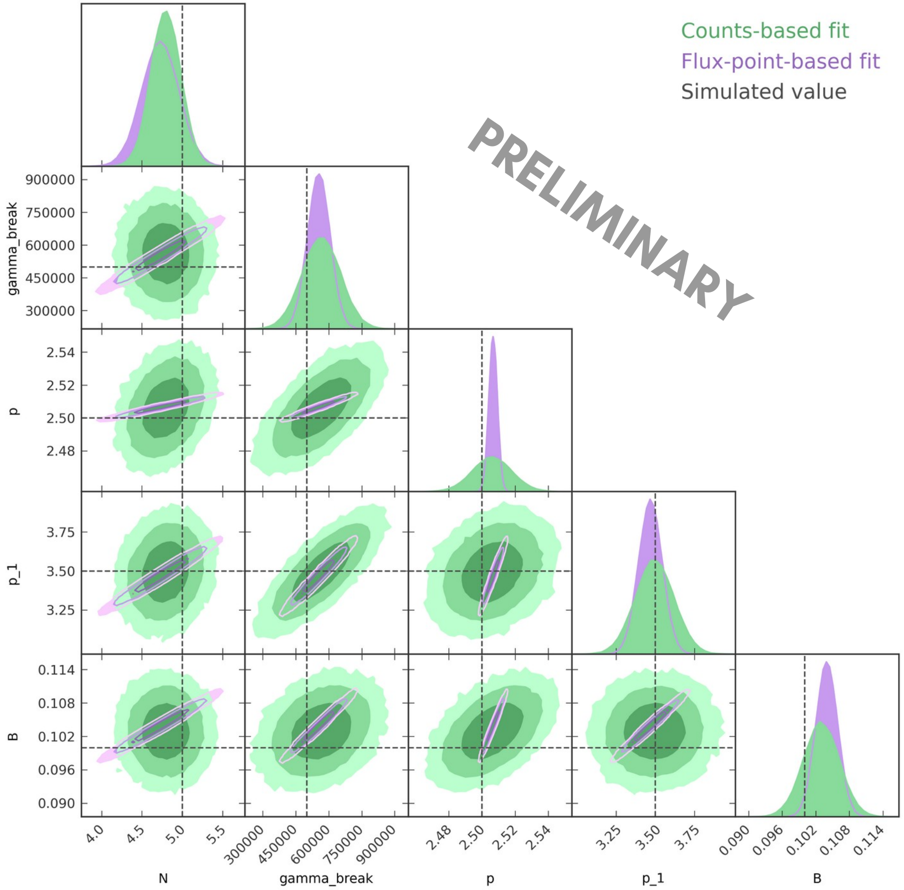


Facilitating multi-messenger modelling using Gammapy

- Works quite well and events fit shows
  - less bias/better error estimation
  - better convergence – up to ~5 free parameters

- But events fit takes longer (5-10x), e.g. 35 vs 10 min for 3, 190 vs. 50 min for 5 free parameters
  - JetSet timing inside Gammapy at one E: 0.02 sec
  - More complex models (e.g. hadronic ones) > 1 min

    → cannot be run on the fly

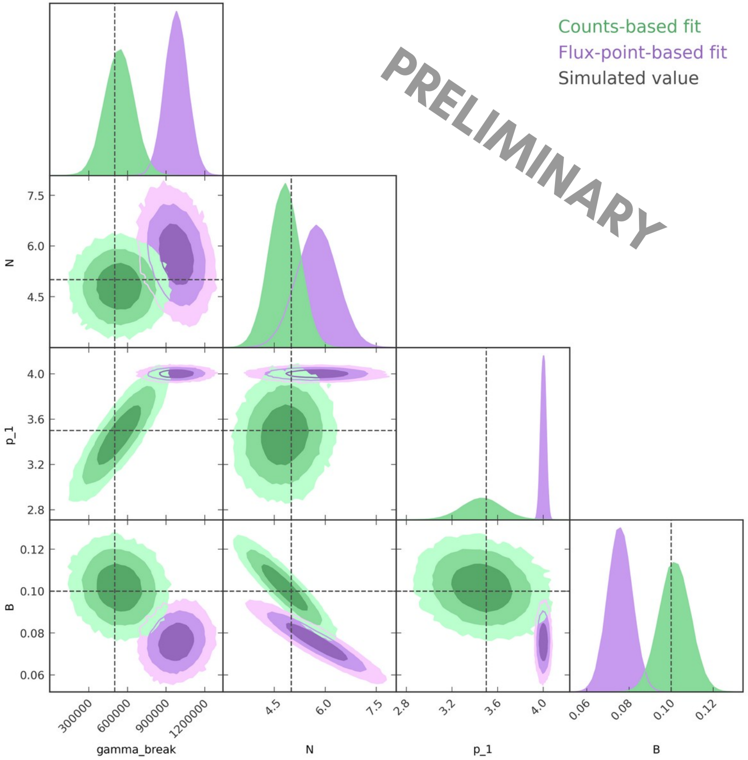


Facilitating multi-messenger modelling using Gammapy

- Work-around methods:
    - Grid of precomputed models
        - 10 grid points per parameters
        - 200 grid points in energy
    - Use grid interpolation during the fit

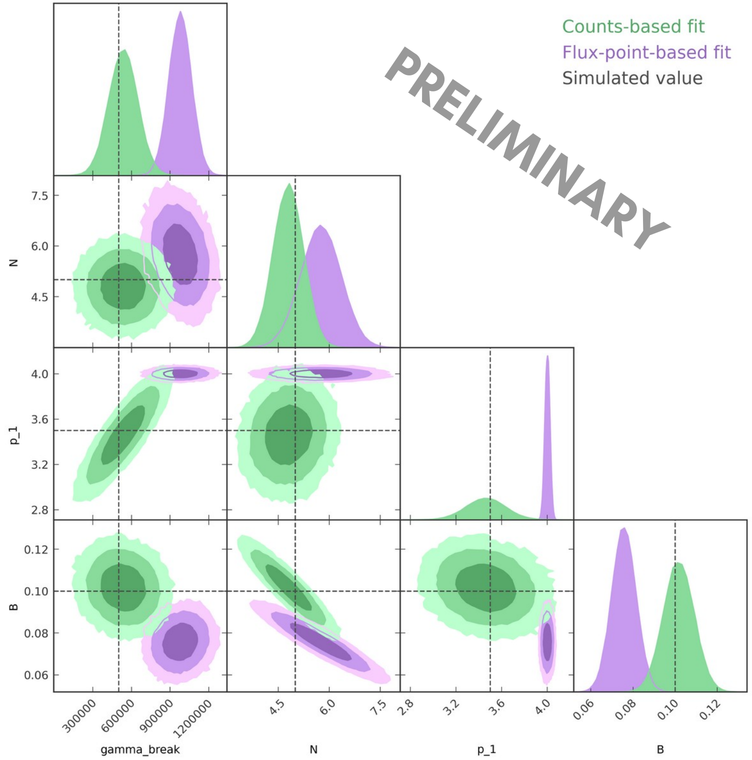
Facilitating multi-messenger modelling using Gammapy

- Work-around methods:
  - Grid of precomputed models
    - 10 grid points per parameters
    - 200 grid points in energy
  - Use grid interpolation during the fit
  - Seems to work well



Facilitating multi-messenger modelling using Gammapy

- Work-around methods:
  - Grid of precomputed models
    - 10 grid points per parameters
    - 200 grid points in energy
  - Use grid interpolation during the fit
  - Seems to work well
  - BUT e.g. overloads memory for a high number of paramters



Lessons learnt so far

- **Gammapy** enables us to **fit physical models directly on event basis**
  - Reducing biases and assumptions
  - Taking into account absorption, systematics or e.g. also ULs naturally
  - But we are still limited by the large amount of parameters

Lessons learnt so far

- **Gammapy** enables us to **fit physical models directly on event basis**
    - Reducing biases and assumptions
    - Taking into account absorption, systematics or e.g. also ULs naturally
    - But we are still limited by the large amount of parameters

- For **fast models (e.g. leptonic ones)**
    - Fitting can be run by Gammapy **on the fly**
    - We also investigate various fitting methods and Bayesian nested sampling, which is limited by the Fermi alanysis → minuit preferred currently

Lessons learnt so far

- **Gammapy** enables us to **fit physical models directly on event basis**
  - Reducing biases and assumptions
  - Taking into account absorption, systematics or e.g. also ULs naturally
  - But we are still limited by the large amount of parameters

- For **fast models (e.g. leptonic ones)**
  - Fitting can be run by Gammapy **on the fly**
  - We also investigate various fitting methods and Bayesian nested sampling, which is limited by the Fermi alanysis → minuit preferred currently

- For **slow models (e.g. hadronic ones)** – **multimessenger fitting**
  - We are currently investigating **grid models + interpolation** methods
  - **Neural networks** and other machine learning methods might be a promising alternative

Thank you
for your attention!