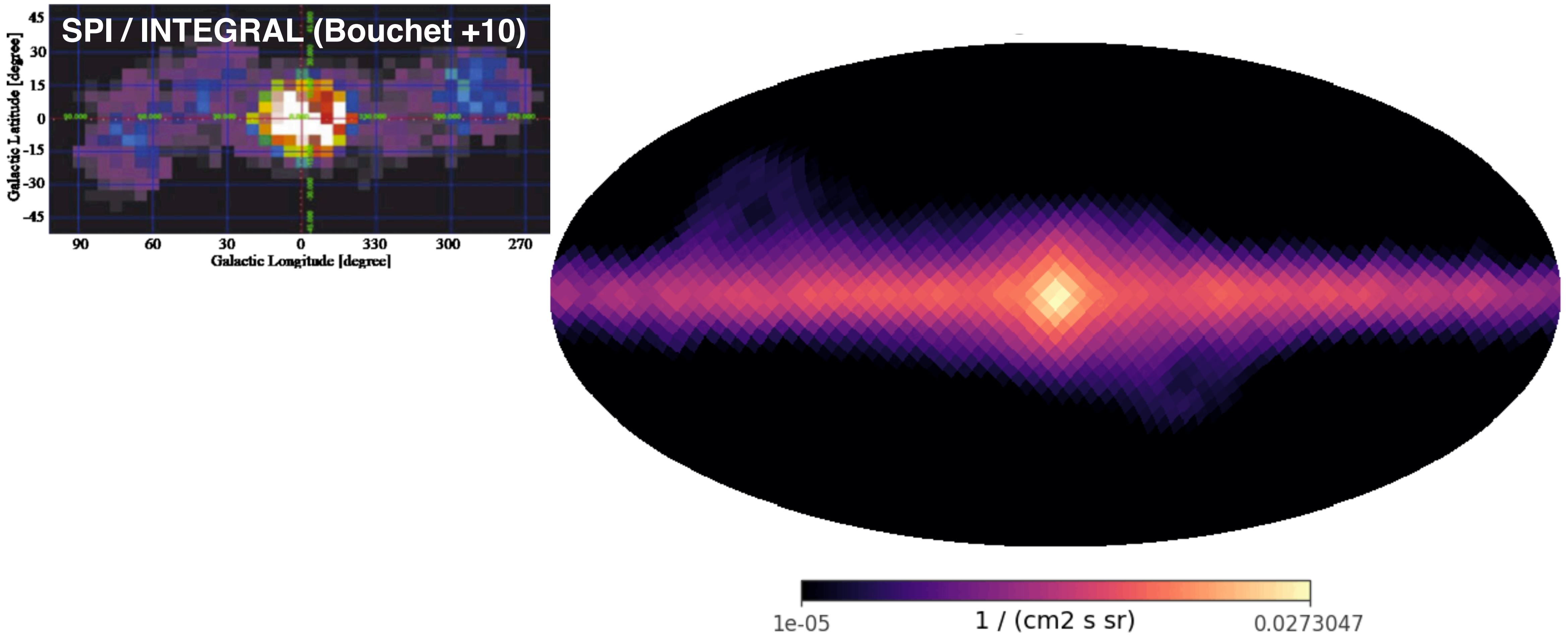
German COSI team meeting
Mainz, 30-31 January 2025

The current status and future prospects of the image analysis

Hiroki Yoneda

Julius-Maximilians-Universität Würzburg

Gamma-ray line observations with ~10x better sensitivity

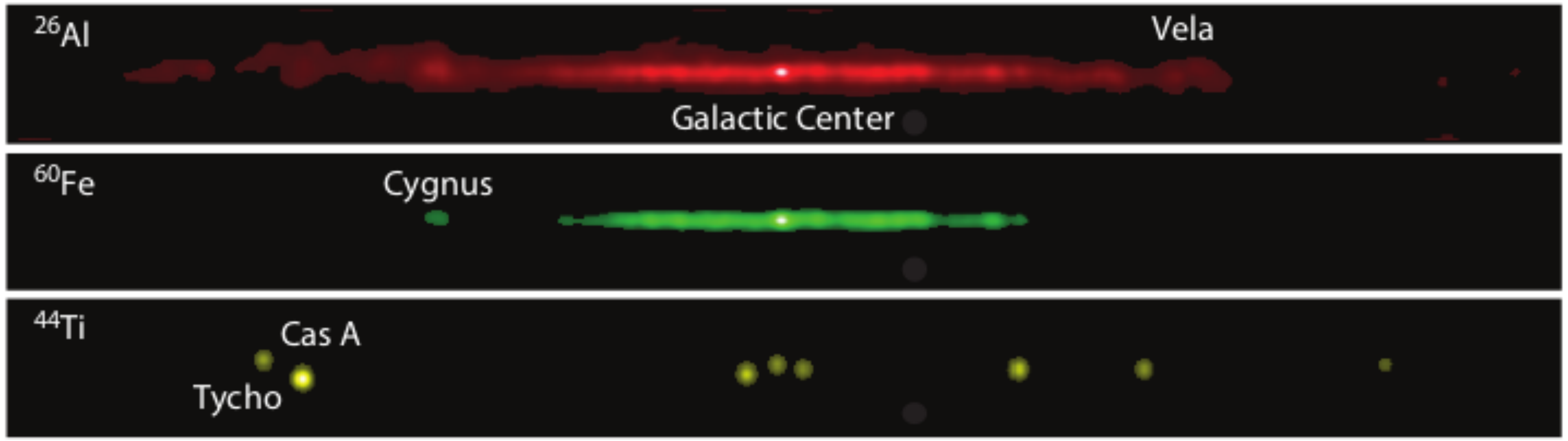


511 keV line (positron annihilation)

- 511 keV image of the bulge and disk
- The disk scale-height measurement
- Line/continuum spectroscopy, e.g., red/blue shift, o-Ps continuum emission

Nuclear gamma-ray lines

- First all-sky image of Fe-60
- Al-26 image with x10 improved sensitivity than COMPTEL
- Search for Ti-44 sources (Cas A, Tycho, SN1897A, etc.)



Fundamental physics: MeV dark matter search (e.g., Yu Watanabe's phD thesis)

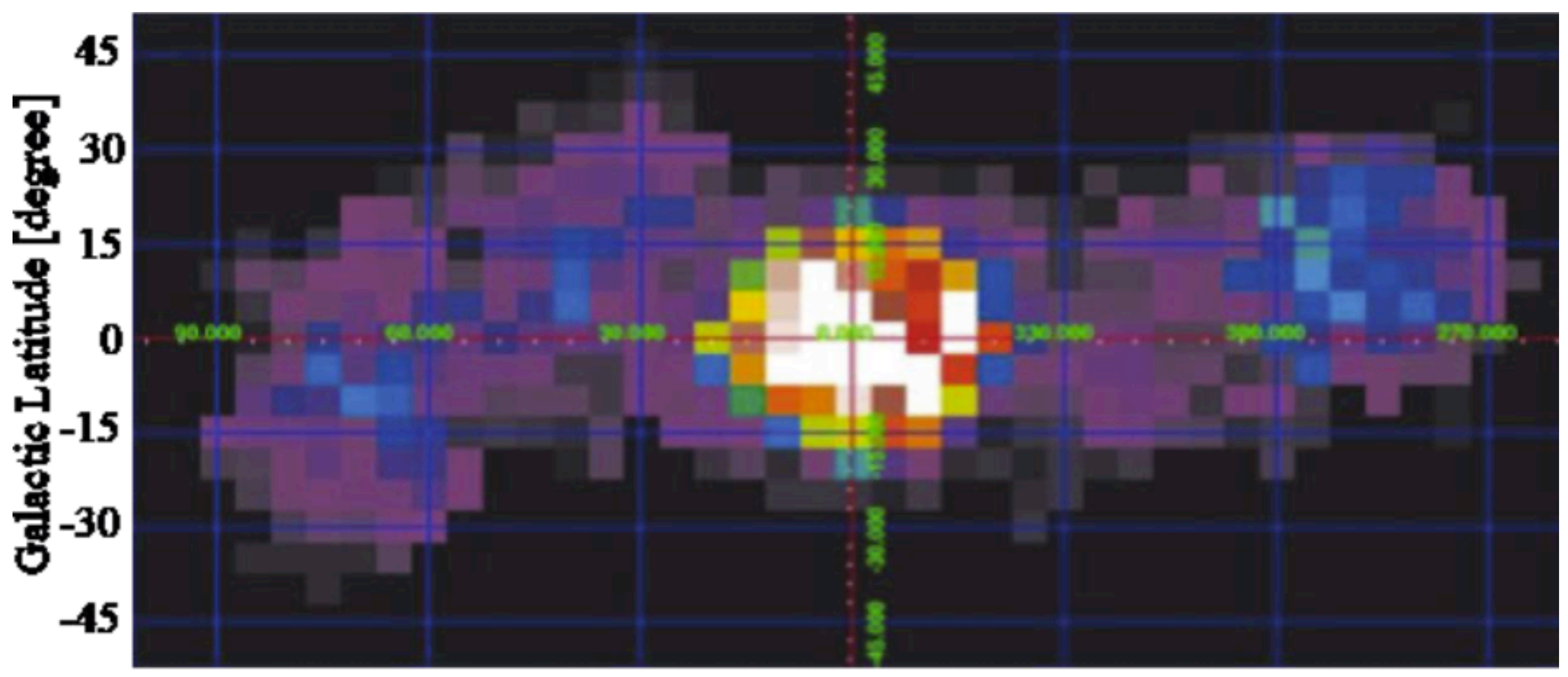
Approaches of image analysis

$$\text{Data} \leftrightarrow \text{Response} \times \text{Model} + \text{Background}$$

Image Deconvolution = Non-parametric approach

- Estimate gamma-ray flux on each pixel

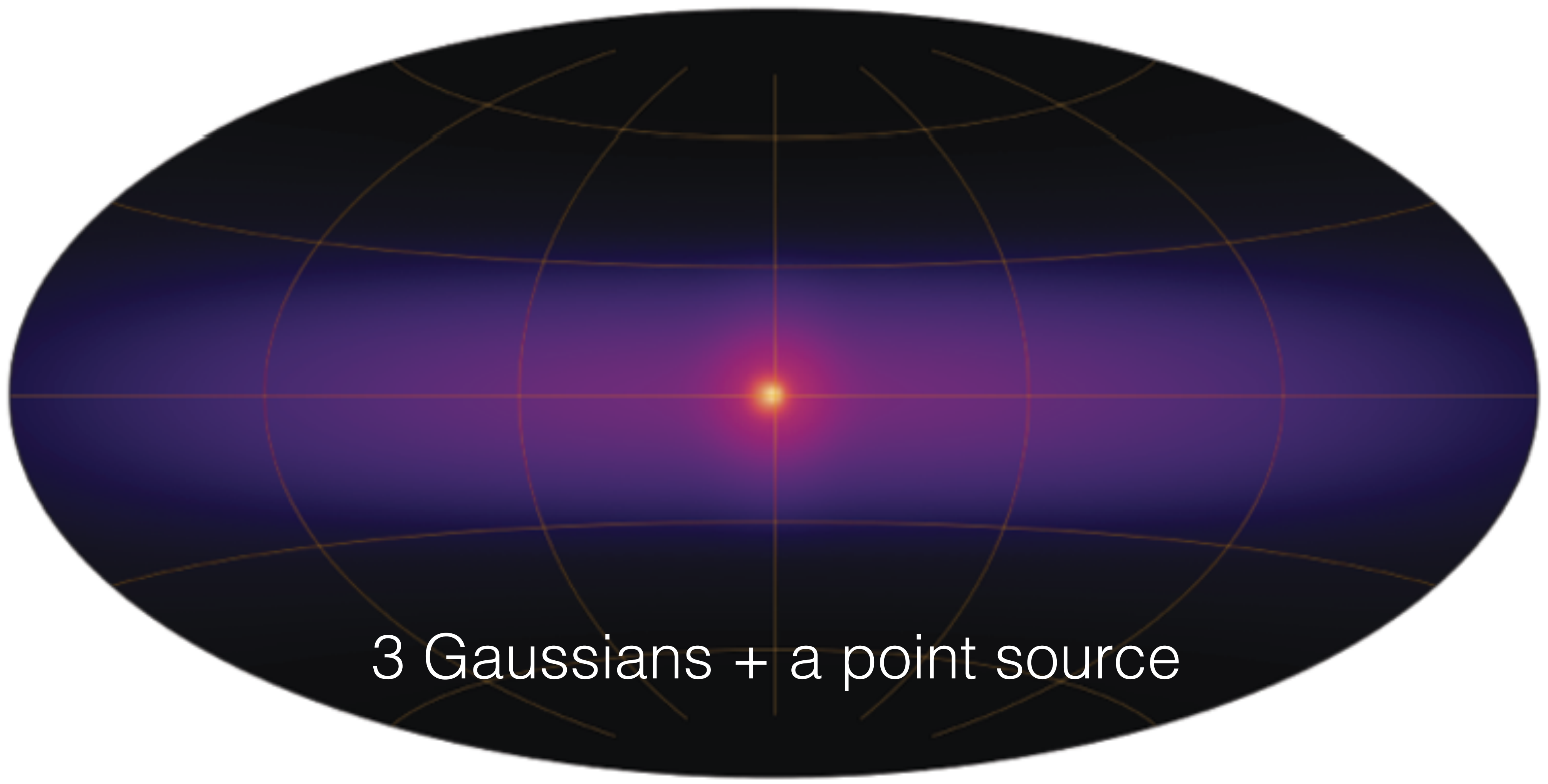
$$\{\lambda_1, \lambda_2, \cdots, \lambda_j, \cdots\}$$



Extended Model Fitting = Parametric approach

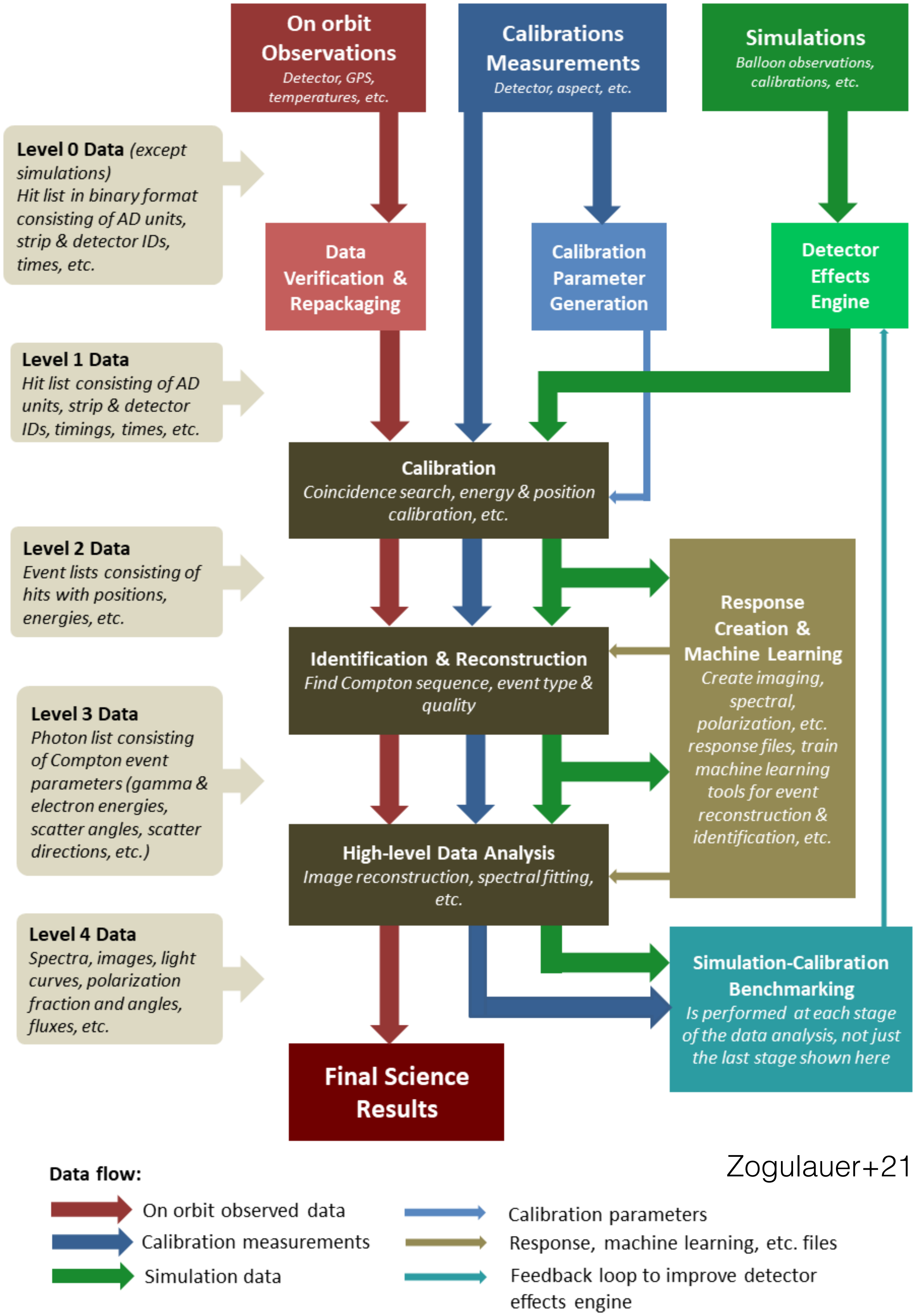
- Fit the data with a spatial model
  - e.g. 2d gaussian disk

$$\frac{N_{511\mathrm{keV}}}{2\pi\sigma_l\sigma_b} \exp\left(-\frac{1}{2}\left[\frac{l^2}{\sigma_l^2} + \frac{b^2}{\sigma_b^2}\right]\right)$$



In either case, we find a parameter set that maximize a defined statistics, e.g., likelihood

COSI tools: Data analysis and simulation



Zogulauer+21

MEGAlib: (Medium Energy Gamma-ray Astronomy library)

- For raw-level data analysis and simulation
  - Detector simulation, calibration
  - Event identification and reconstruction
  - Response generation

COSIpy

- **For high-level data analysis**
- **python-based**
- **Most of users will start with COSIpy**

- **Spectral analysis**
- **Image deconvolution, fitting**
- **Polarization analysis** (to be implemented)
- **Source localization etc.**

Model Fitting (spectral / spatial / polarization fitting)

Parametric approach (with regards to imaging)

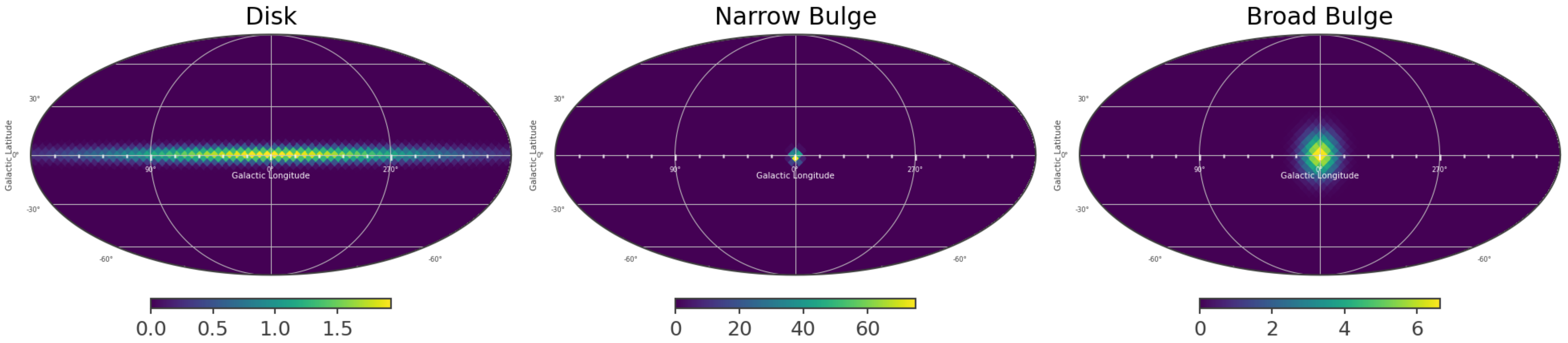
- Fitting the data with a spatial model

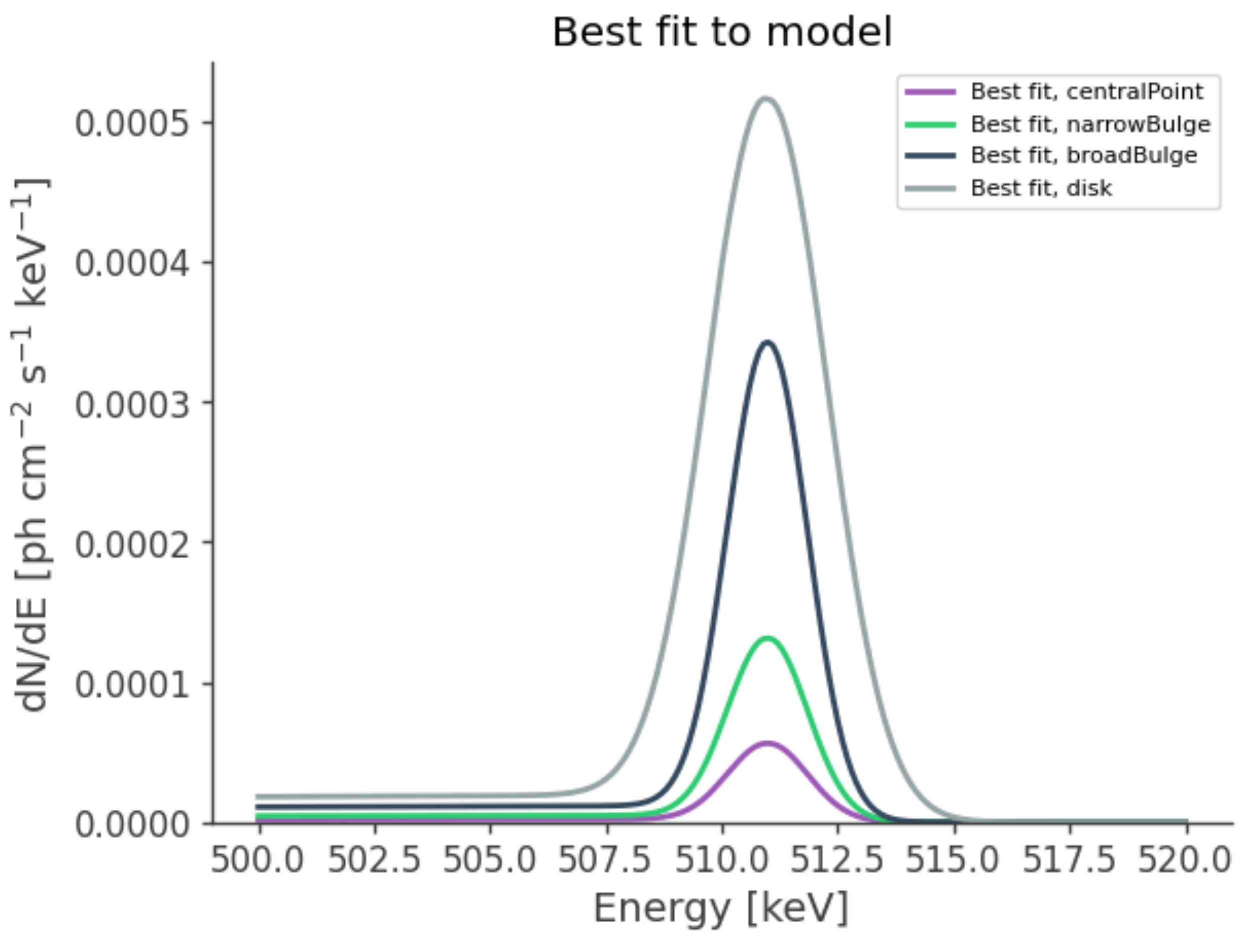
  e.g. 2d Gaussian disk, physical model (galprop model, CO map)

- Less parameters but strong assumptions on the spatial distribution

Fitting with the Multi-Mission Maximum Likelihood framework (3ML)

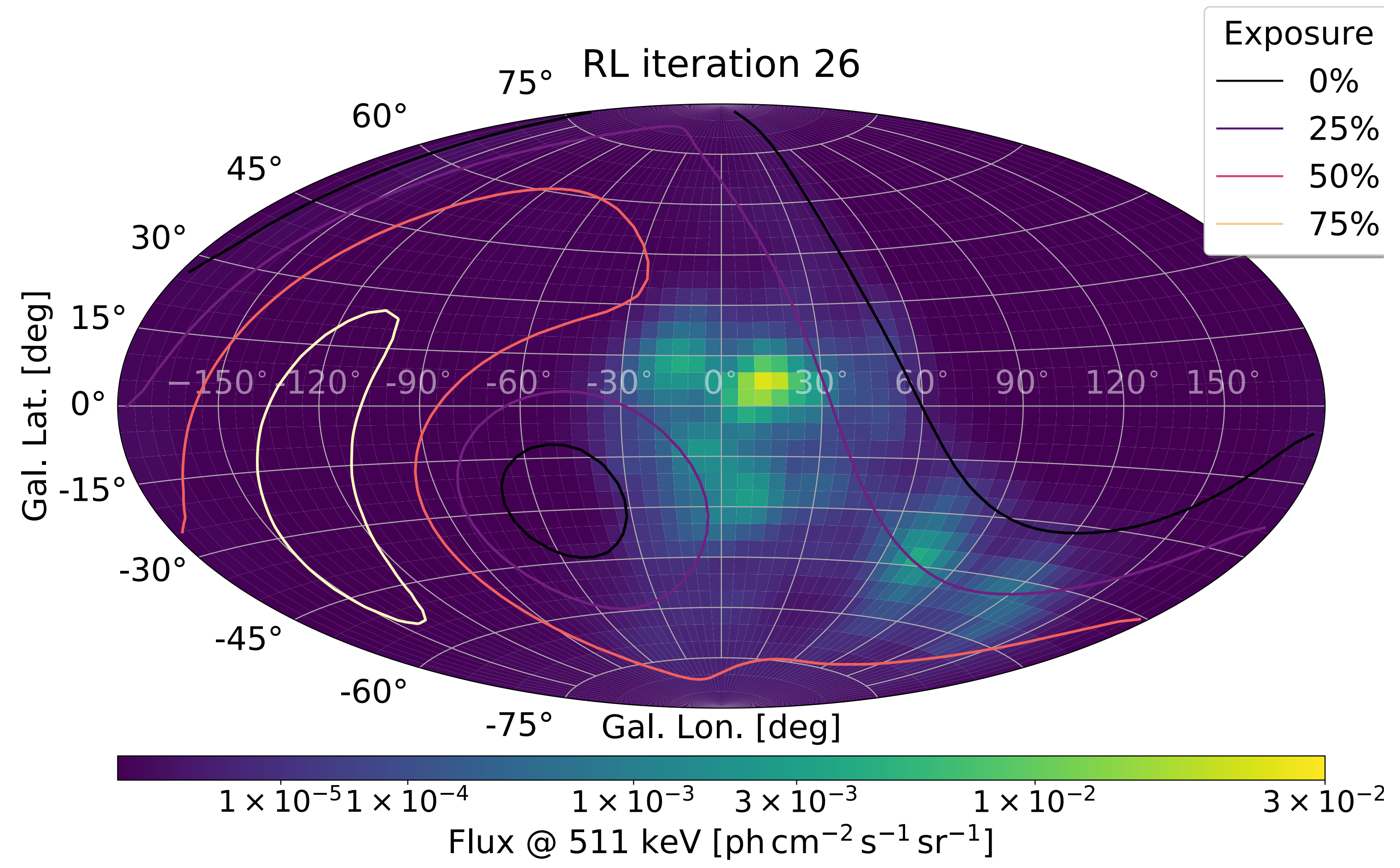
- Multi-wavelength/multi-messenger analysis framework (with X-rays, Fermi, neutrino, etc.)
- We fit multiple point/extended sources simultaneously
- We can use customizable models (astromodels)





https://github.com/cositools/cosipy/blob/main/docs/tutorials/spectral_fits/extended_source_fit/diffuse_511_spectral_fit.ipynb

Image Deconvolution with Richardson-Lucy algorithm



511 keV map from the COSI ballon flight (Siegert+20)

Non-parametric approach

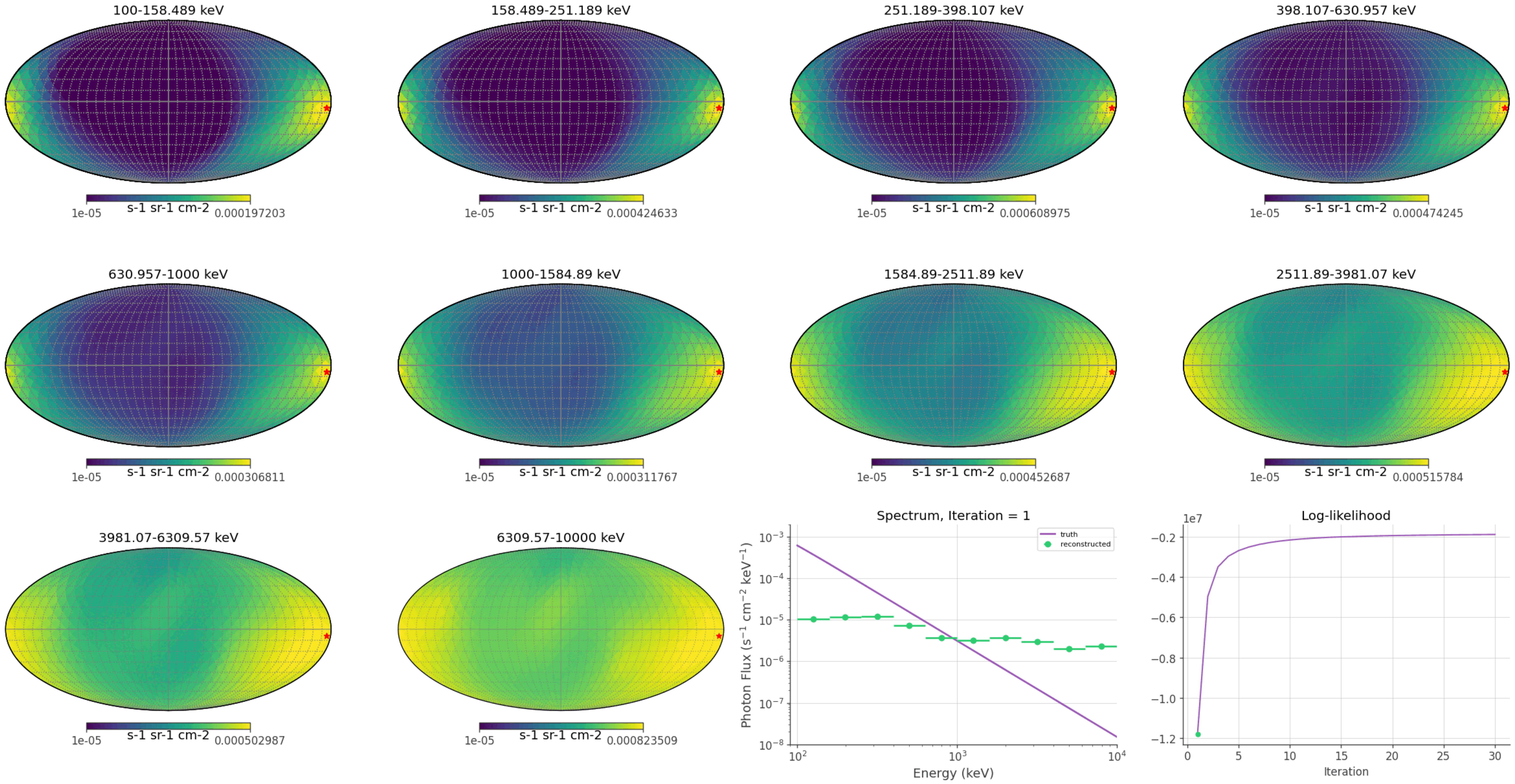
- Flux pixel-by-pixel/energy-bin-by-bin
- Less assumption on the spatial distribution
- More parameters (= the number of pixels)

Richardson-Lucy algorithm

- reconstruct the image iteratively and maximize the likelihood function
- derived from the EM algorithm with Poisson distribution

$$\epsilon_i = \sum_j R_{ij} \lambda_j^{\mathrm{old}} + b_i \qquad \lambda_j^{\mathrm{new}} = \frac{\lambda_j^{\mathrm{old}}}{\sum_i R_{ij}} \sum_i \frac{D_i}{\epsilon_i} R_{ij}$$

Image Deconvolution with 3-month Crab+background data



Spectral and spatial reconstruction with optimizing ~10^4 parameters

511 keV imaging: https://github.com/cositools/cosipy/tree/main/docs/tutorials/image_deconvolution/511keV/GalacticCDS

PR#188

ImageDeconvolution

It performs the image deconvolution

by using the following three classes

Model (←histpy.Histogram)

(base class: ModelBase)

It defines the model to be reconstructed

DataInterface

(base class: ImageDeconvolutionDataInterface)

It defines the data, backgrounds, how to calculate expected counts etc.

DeconvolutionAlgorithm

(base class: DeconvolutionAlgorithmBase)

It defines the image reconstruction algorithms (Richardson-Lucy, MREM etc.)

PR#188

ImageDeconvolution

It performs the image deconvolution

by using the following three classes

Model (←histpy.Histogram)

(base class: ModelBase)

It defines the model to be reconstructed

→ 3D imaging by Hugh Bates

Parallel computation by Anaya

↑

DataInterface

(base class: ImageDeconvolutionDataInterface)

It defines the data, backgrounds, how to calculate expected counts etc.

MAP RL by HY

↑

DeconvolutionAlgorithm

(base class: DeconvolutionAlgorithmBase)

It defines the image reconstruction algorithms (Richardson-Lucy, MREM etc.)

Image deconvolution with Maximum a posteriori (MAP) estimation

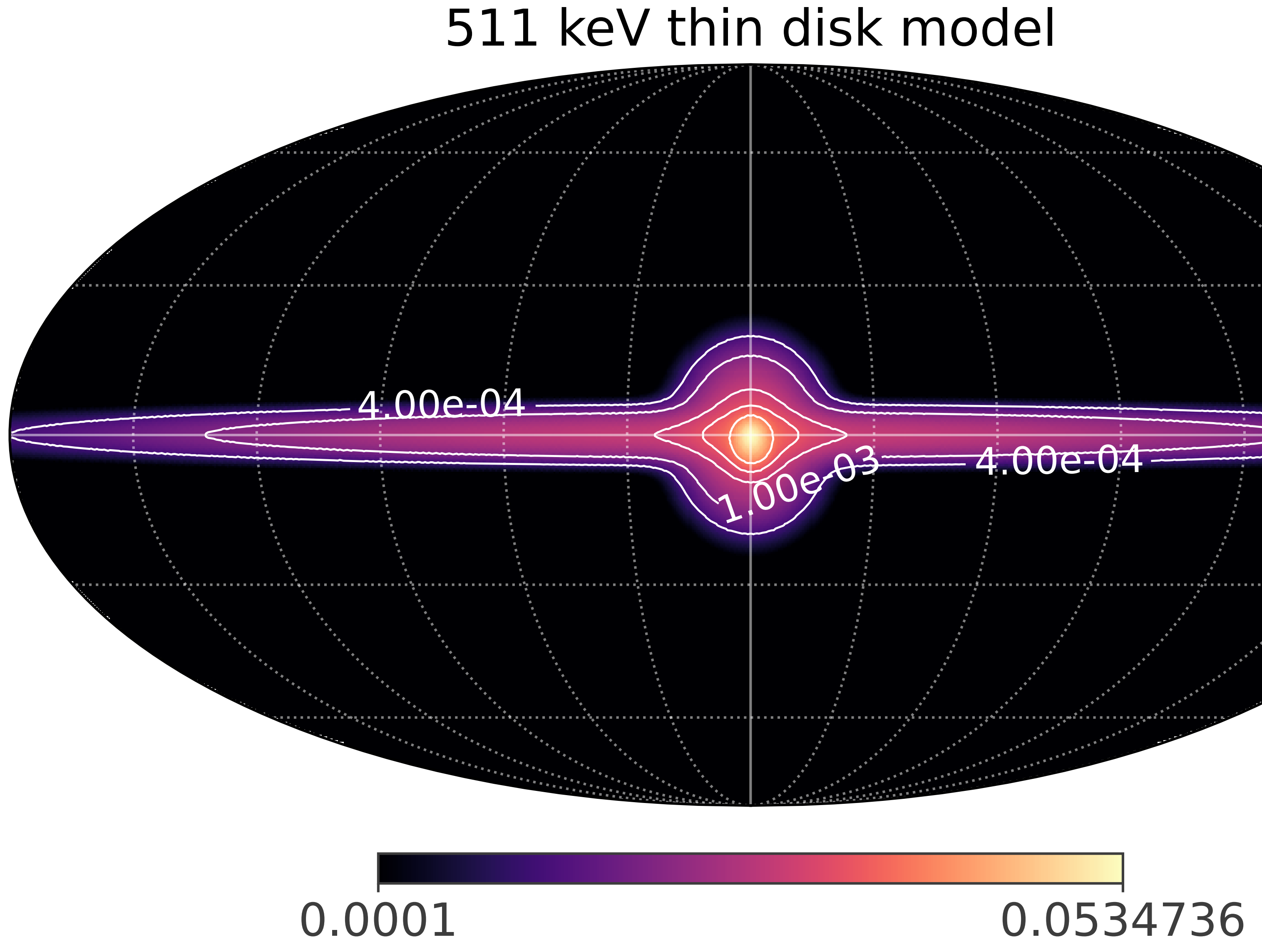
Imaging of the black hole shadow with EHT

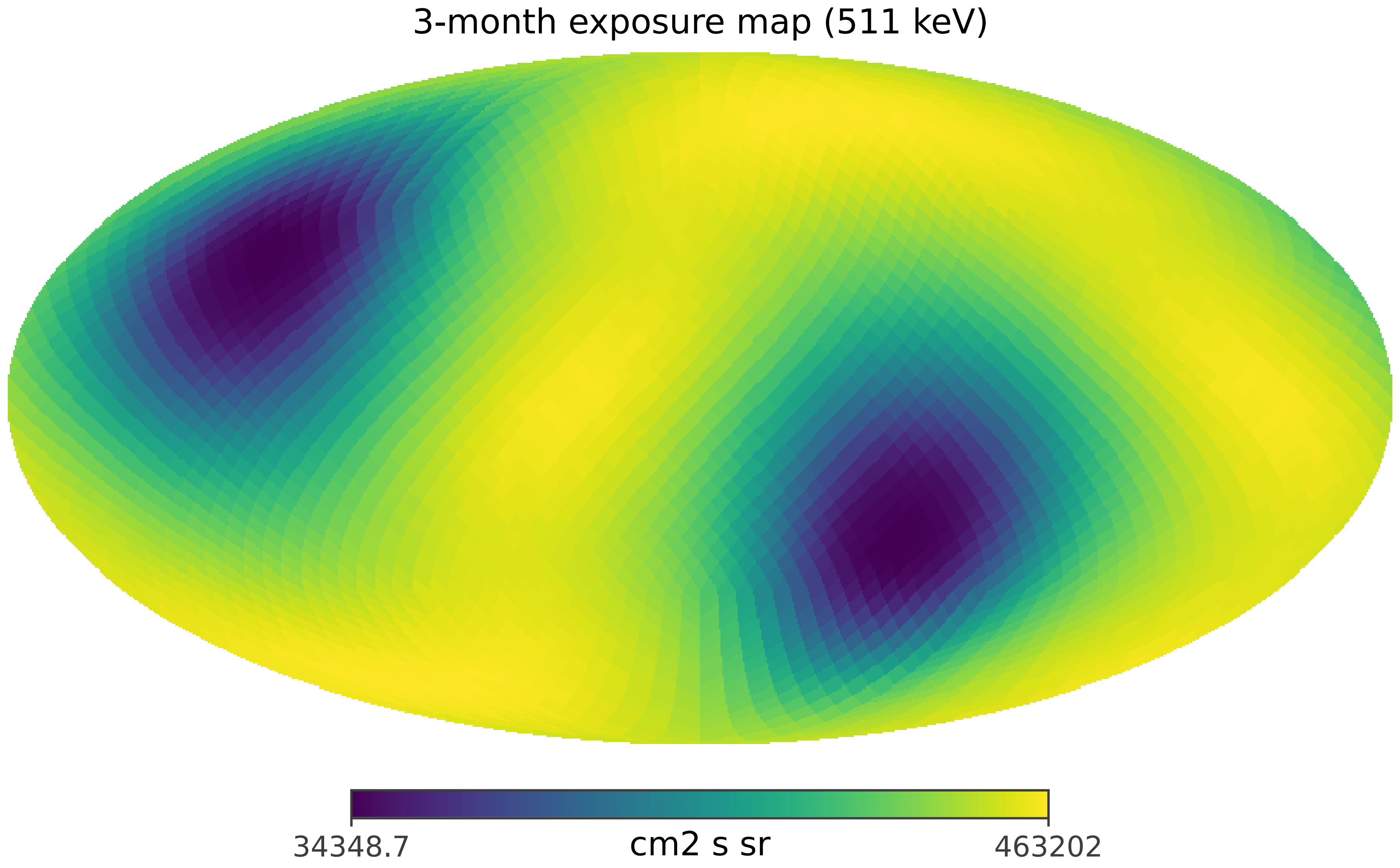
- They minimize **(Chi-square)** + **(Regularization terms)**
- Regularization terms includes
  - Sparseness: L1 norm
  - Smoothness: Total Squared Variation, Total Variation
  - Flatness: Entropy
  - Absolute value: Total flux

Applying the MAP estimation to COSI data analysis

- We minimize **(Log-likelihood)** + **(Regularization terms)**
- The optimization is performed with **Richardson-Lucy algorithm**
- Choose a regularization term applicable to Poisson data, e.g., L1 norm cannot work well

Applying the MAP RL to COSI 3-month simulation data





- Simulating the 3-month COSI observations
- Used the thin disk model dataset
- Used all of the background simulations

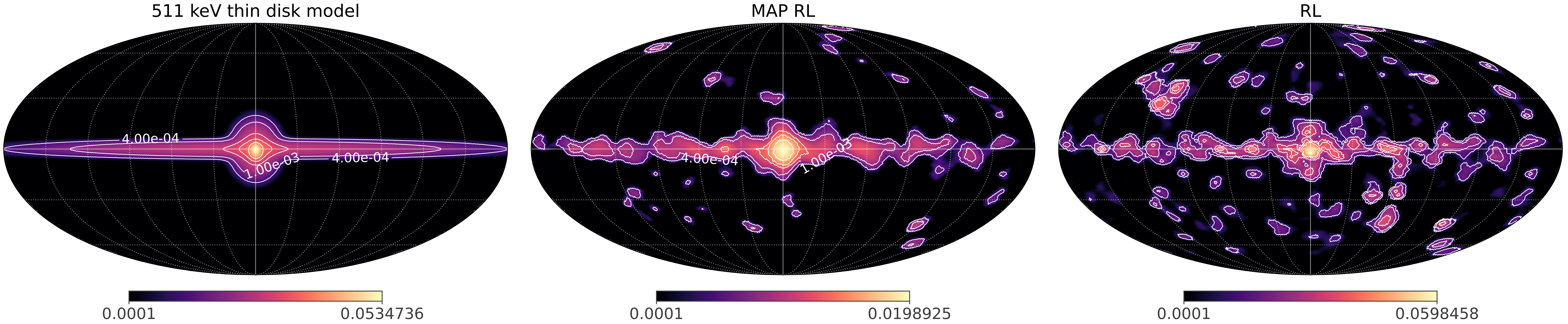
- Using the modified Richardson-Lucy algorithm, we maximize the following posterior probability

$$\sum_i D_i \log \epsilon_i - \sum_i \epsilon_i$$ ←**log-likelihood**

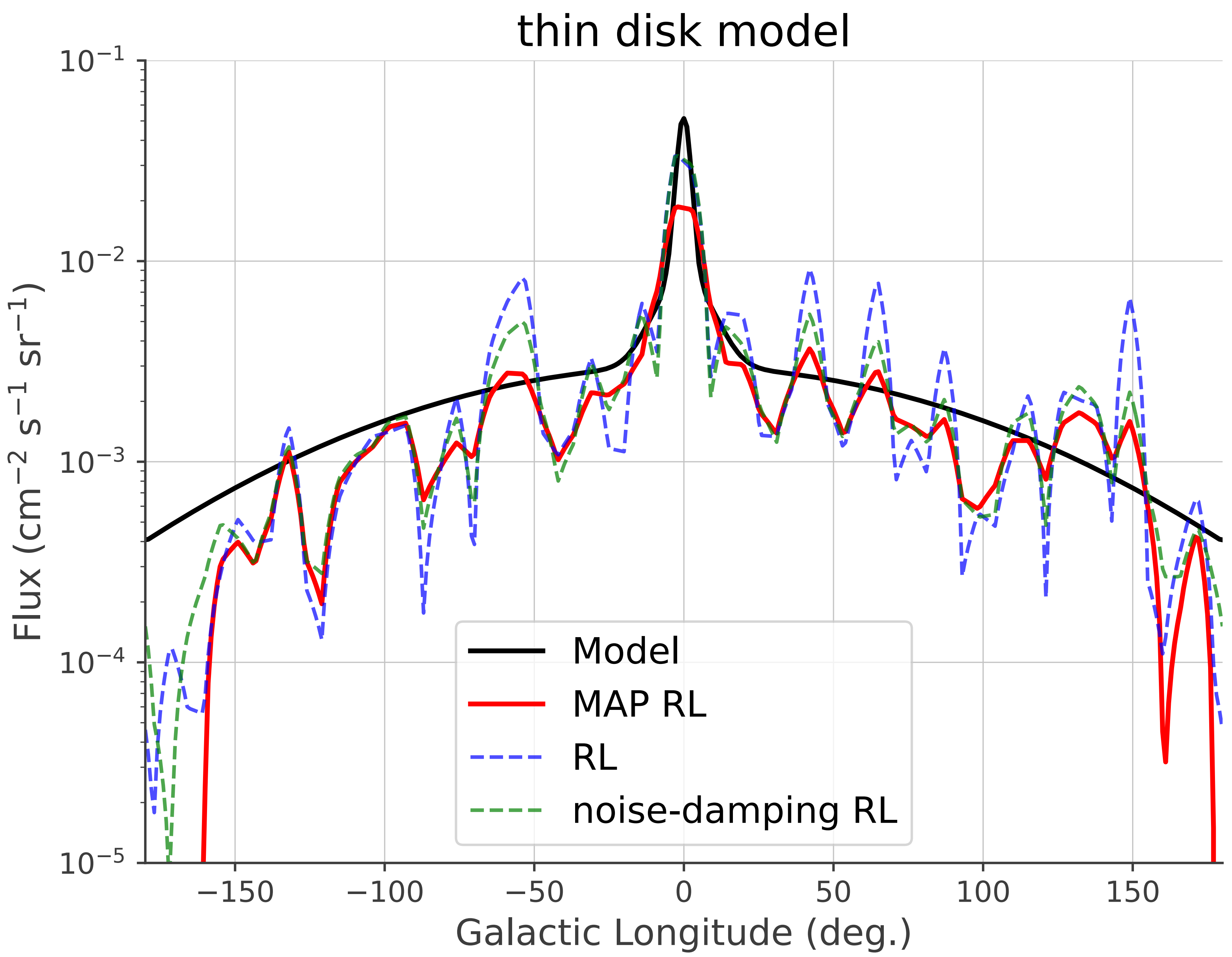
$$-c^{\mathrm{TSV}} \sum_j \sum_{k \in \sigma_j} (\lambda_j - \lambda_k)^2 - c^{\mathrm{SP}} \sum_j \log \lambda_j$$

↑ **smoothness** ↑ **sparseness**

Comparison of MAP image with the conventional RL



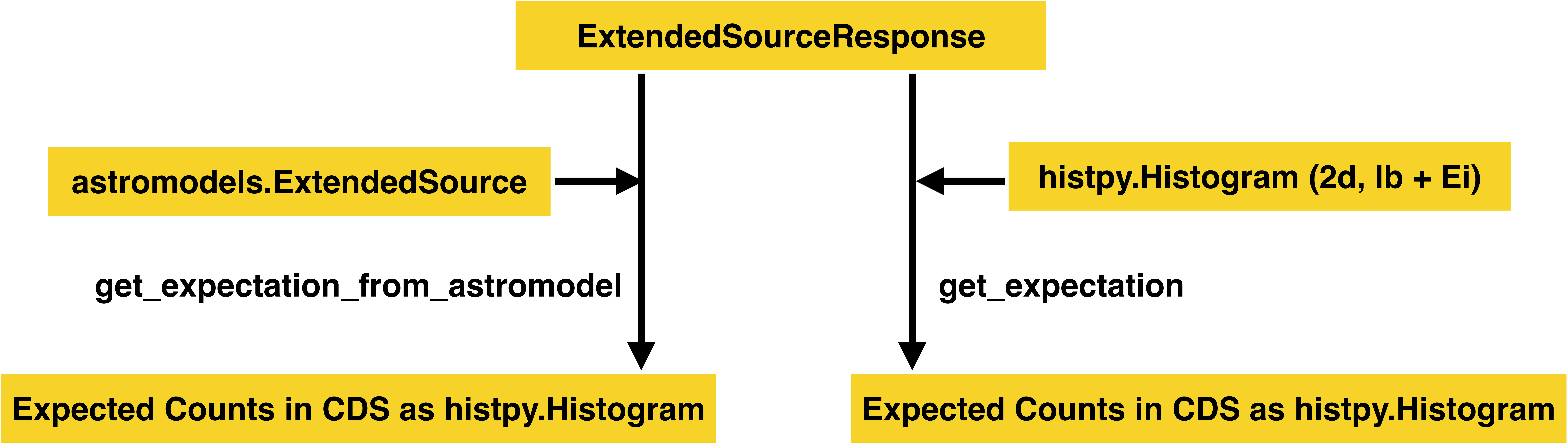
Algorithm	Total Flux (cm^{-2} s^{-1})	
	Thin Disk Model	Thick Disk Model
MAP RL	2.91×10^{-3}	2.72×10^{-3}
RL	4.26×10^{-3}	4.33×10^{-3}
Noise-damping RL	4.11×10^{-3}	4.17×10^{-3}
Model	2.59×10^{-3}	2.64×10^{-3}



Other things completed or in progress

Implemented a class to handle an extended source response (pre-computed response in Gal. coord)

- https://github.com/cositools/cosipy/pull/223
- A functionality to generate an extended source response from a full detector response is under review: https://github.com/cositools/cosipy/pull/284
- The source injector for an extended source can be implemented based on this.
- Krishna is working on it.



Other things completed or in progress

Improving computational performance

- Convolving the extended source response with the image includes large matrix calculation
- 24 GB for the continuum (10 energy bands, 7 deg.), 4 GB for the line (3 deg.)

Testing several approaches

- Choosing an optimal function for the numpy calculation (np.tensordot)
- Testing a GPU calculation (with the CuPY library), resulting in x50 faster speed
- Testing parallel calculation (with openMPI), lead by Anaya Valluvan in UCSD
  - https://github.com/cositools/cosipy/pull/274
- Optimizing the code by Washington University people (Jeremy Buhler, Augustus Thomas)
  - https://github.com/cositools/cosipy/pull/255

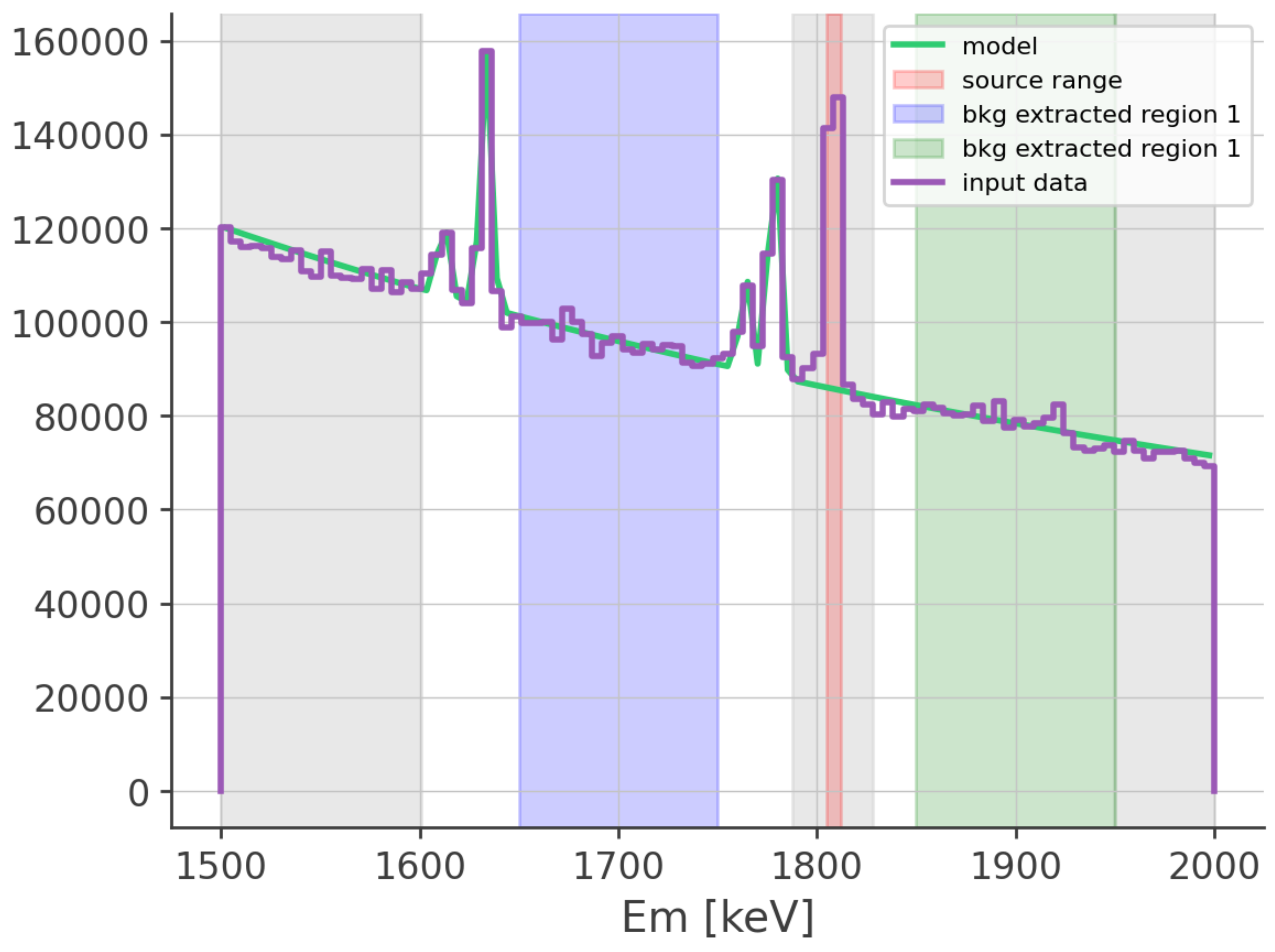
CuPy

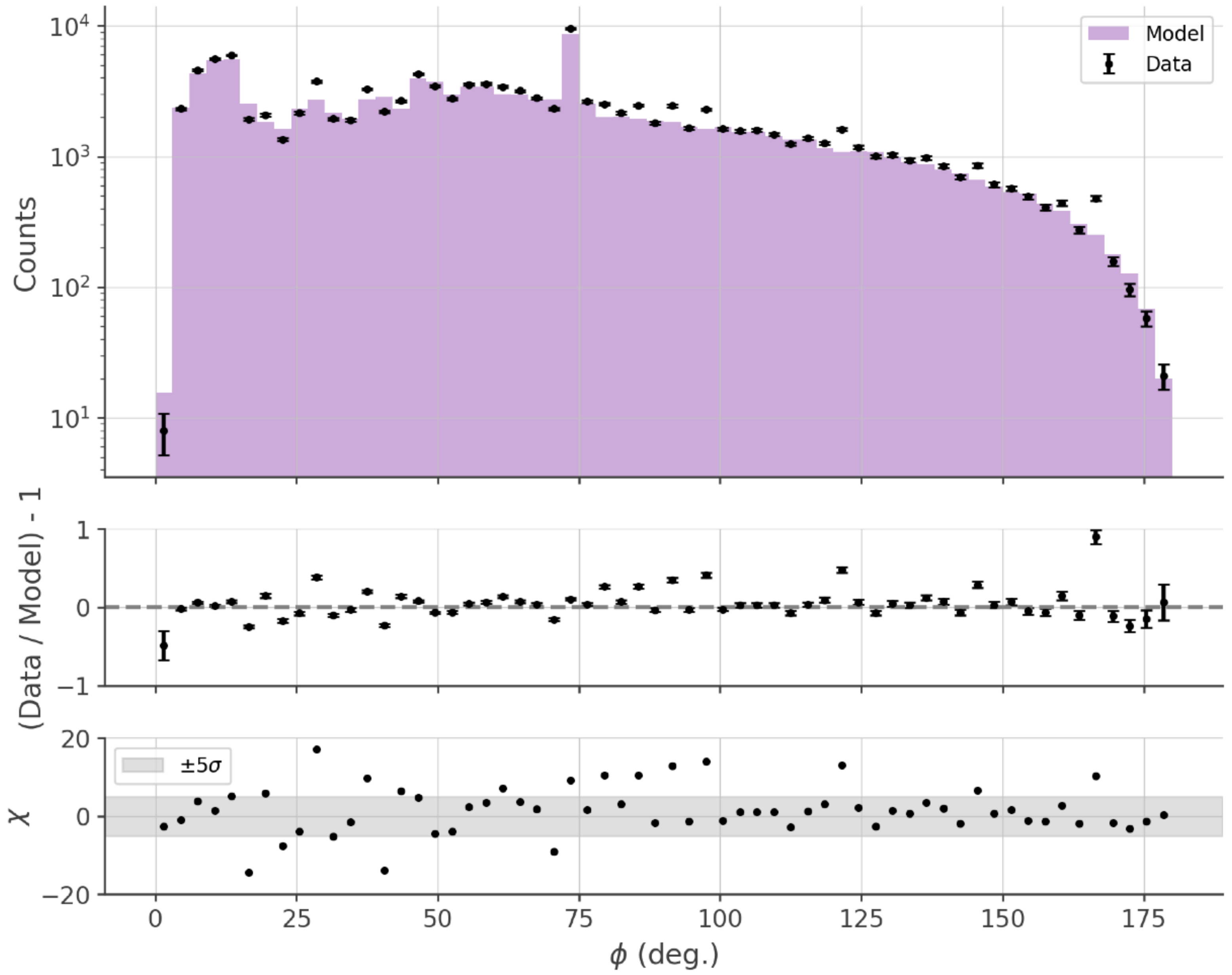
These two things will be directly related with how we describe the response function

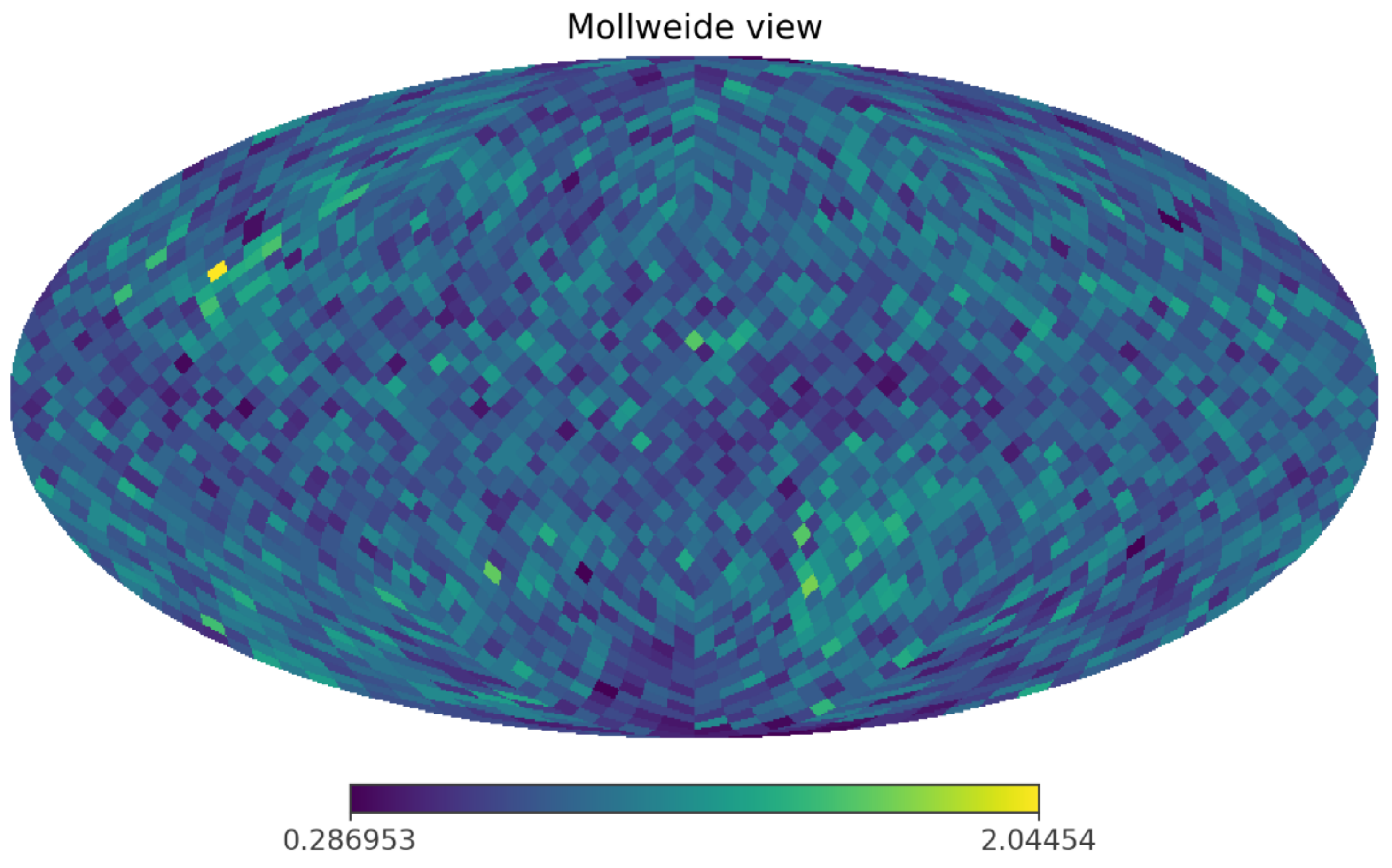
Other things completed or in progress

Implemented a (alpha-version) class to estimate a background model from Savitri's simulation data

- Line background generator by Saurabh, HY
  - https://github.com/cositools/cosipy/pull/252
- Continuum background generator by Chris Karwin
  - https://github.com/cositools/cosipy/pull/235







Future (small and big) potential projects

Challenges

How to handle response and data

- O(100) GB for <1 deg. resol.→ parallel computation, better response descriptions, NN
- Considering the earth occultation, data should be divided into several subsets for the best S/N

How to model background

- 511 keV lines are also produced from background events, e.g., radioactivation by charged particles
- Need to investigate several approaches and find good tracers of each background component

How to make the most plausible image, especially the image deconvolution

- The number of free parameters (= pixel number) is large, causing overfitting and artifacts
- Need to incorporate some informations in the analysis, a shape of image, background scale range

Implementing and comparing more image deconvolution algorithms

- **e.g., Maximum Entropy**

log-likelihood　　　　Entropy

$$\sum_i D_i \log \epsilon_i - \sum_i \epsilon_i + c^{\mathrm{ENT}} \sum_j \lambda_j \left(1 - \log\left(\frac{\lambda_j}{m_j}\right)\right)$$

```
class PriorEntropy(PriorBase):
    def __init__(self, coefficient, model, param): # param needs to be implemented
        super().__init__(coefficient, model, param)
        self.prior_image = param['prior_image']

    def log_prior(self, model):
        return self.coefficient * np.sum(model * (1 - np.log(model / self.prior_image)))

    def grad_log_prior(self, model):
        return -1 * self.coefficient * np.log(model / self.prior_image)
```

- **Multi-resolution EM**
  - need to understand how to calculate the wavelet function on HealPix map
- **Information Field Theory?**

Accelerate the image deconvolution using state-of-art algorithms

Many algorithms have been proposed to accelerate the EM algorithm

- Accelerated ML-EM algorithm (Knoedlseder+99), only implemented currently

$$\lambda_j^{k+1} = \lambda_j^k + \alpha^k \delta\lambda_j^k$$

$$\alpha^k < \max(-\lambda_j^k / \delta\lambda_j^k)$$

SQUAREM algorithm by Du and Varadhan 2020

Algorithm 1: Pseudocode for SQUAREM.

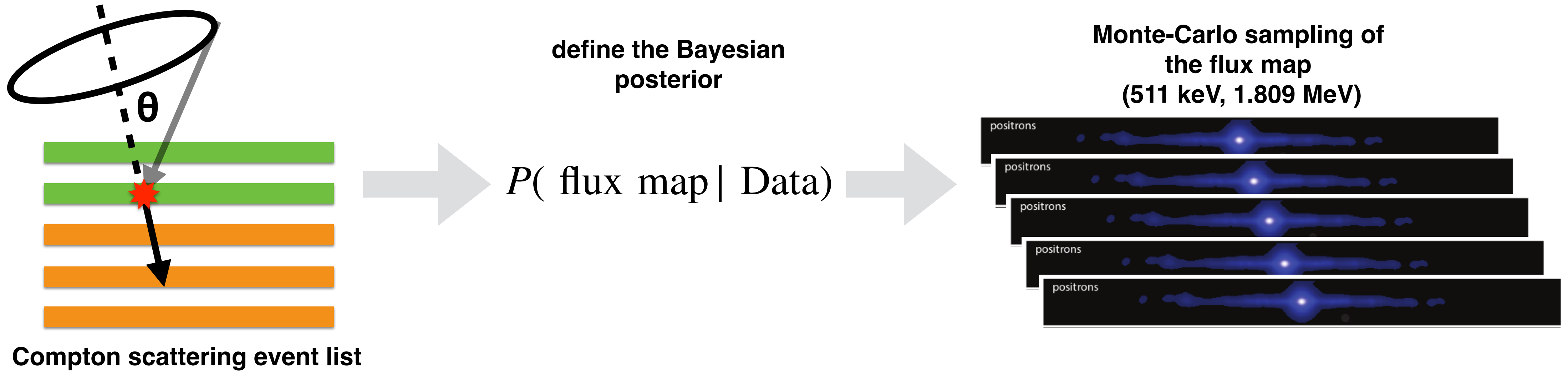
Input: F, L, θ_0, $\eta \geq 0$
while *not converged* **do**
  $\theta_1 = F(\theta_0)$
  $\theta_2 = F(\theta_1)$
  $r = \theta_1 - \theta_0$
  $v = (\theta_2 - \theta_1) - r$
  Compute steplength α
  $\theta_{sq} = \theta_0 - 2\alpha r + \alpha^2 v$
  if $L(\theta_{sq}) > L(\theta_2) - \eta$ **then**
    Set $\theta' = \theta_{sq}$.
  else
    $\theta' = \theta_2$
  end
  $\theta_0 = F(\theta')$, stabilization step (done only if $\theta' = \theta_{sq}$)
end

Ikeda+00,14

$$\rho^{new}(\boldsymbol{u}) \propto \frac{\rho^{(l+1)}(\boldsymbol{u})^2}{\rho'(\boldsymbol{u})}$$

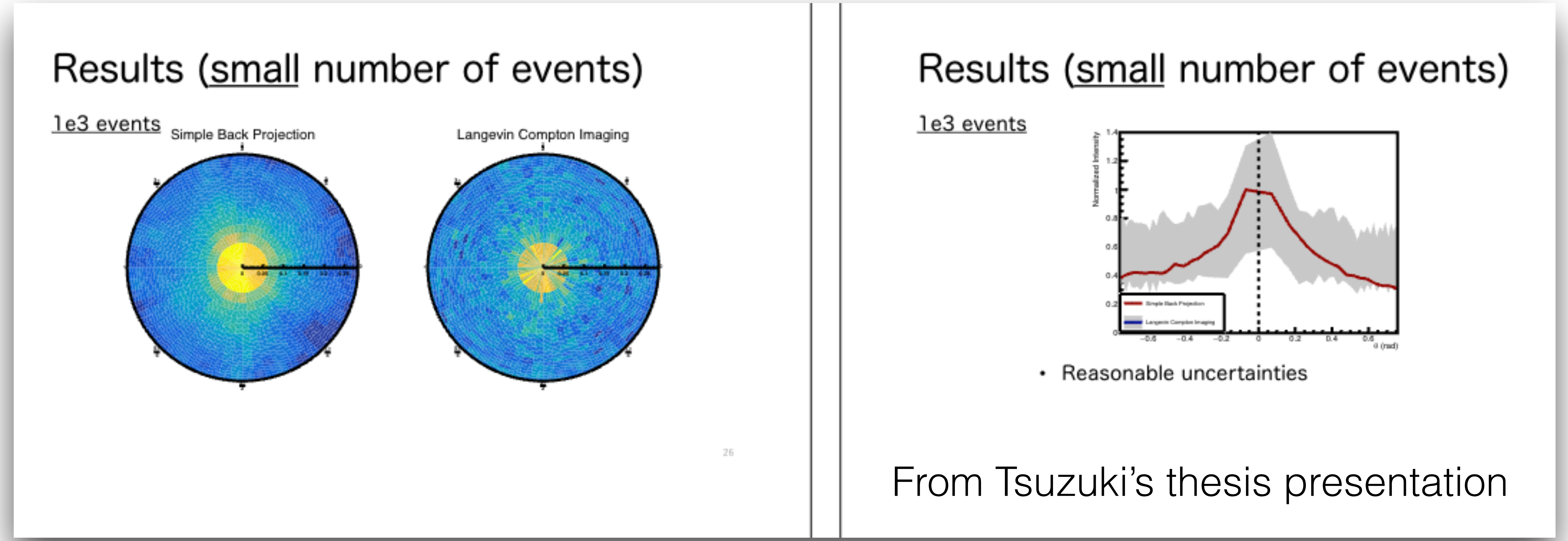
Error estimation in the image deconvolution

Tsuzuki proposed a “workable” Markov chain Monte Carlo method for Compton imaging (Tsuzuki, phD thesis, a paper in prep.)



He found that Langevin MCMC is a good choice for Compton imaging

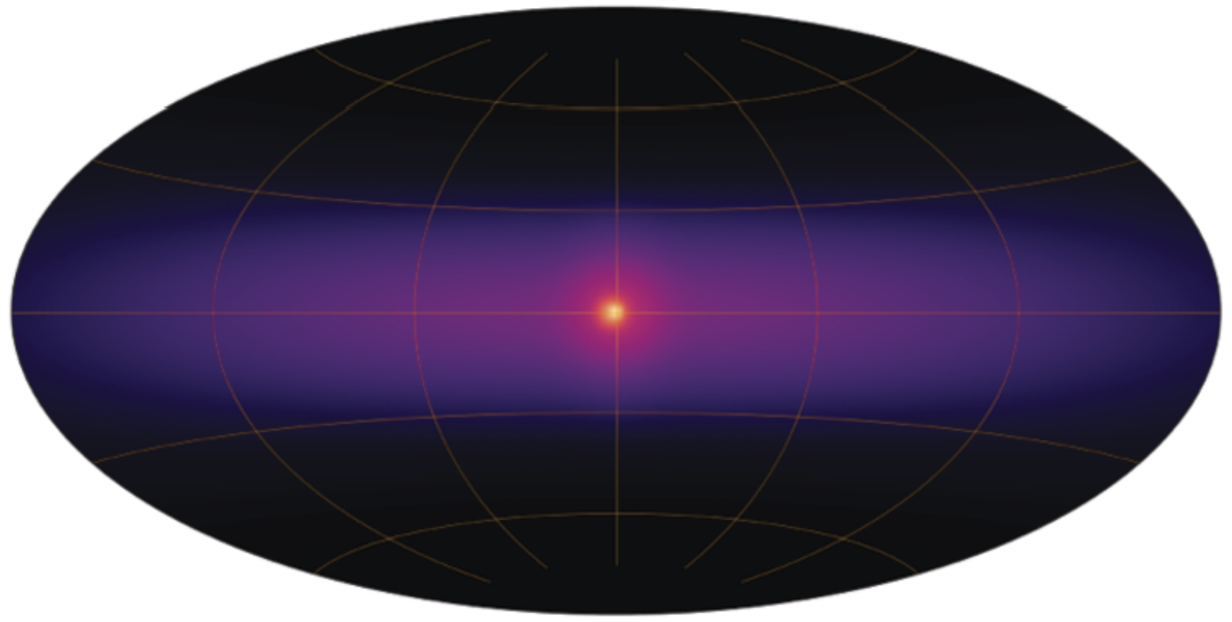
- $$\lambda \leftarrow \lambda + \frac{1}{2}\epsilon^2 \frac{\partial \ln P(\lambda \mid D)}{\partial \lambda} + \epsilon \xi$$
- ξ is a Gaussian noise



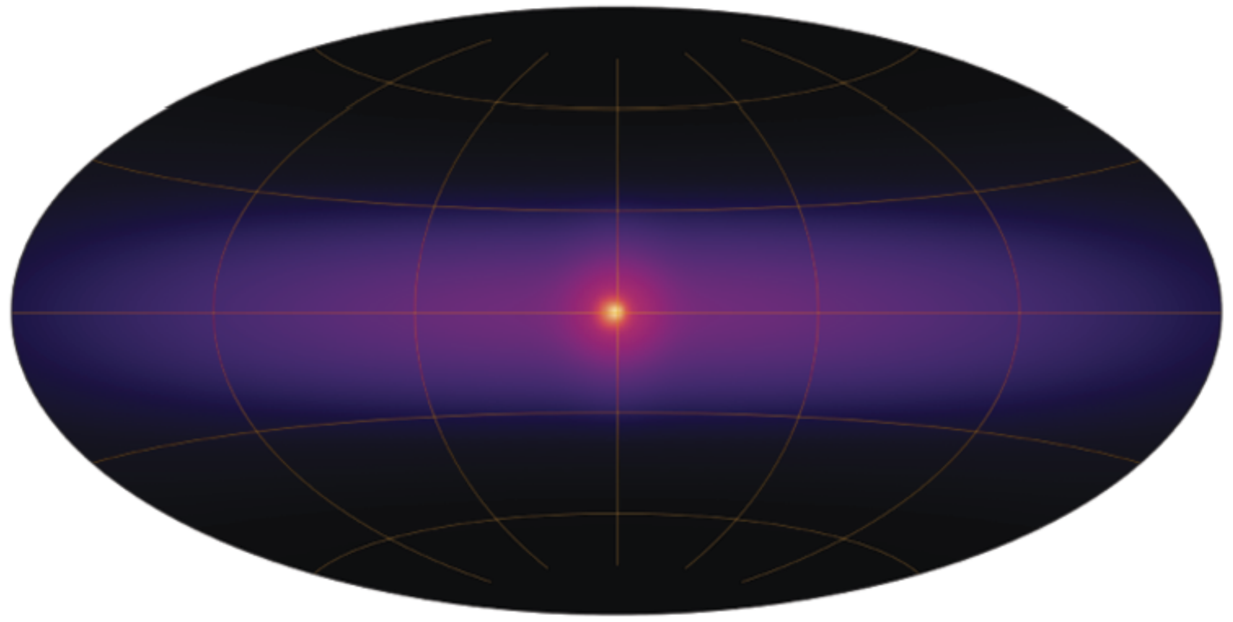
From Tsuzuki’s thesis presentation

Multi-component image reconstruction

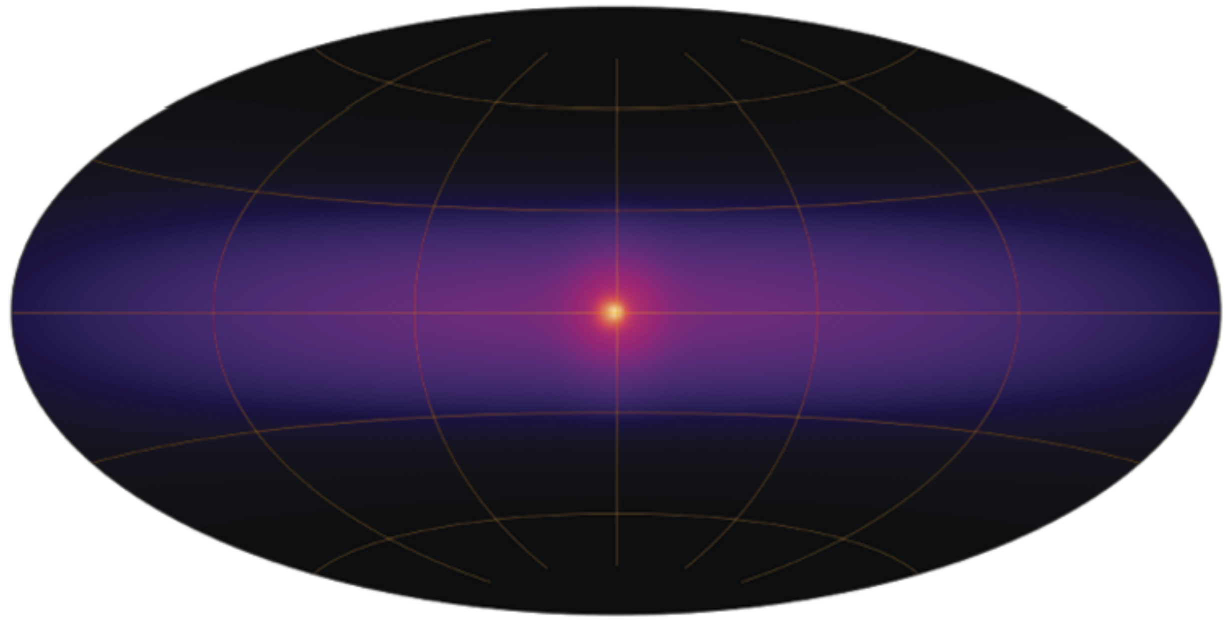
e+ line



e+ continuum



Galactic diffuse



COSI data interface for multi spatial component model

Perform image deconvolution

- Positronium fraction map over the sky
- Ionization, temperature at each annihilation site
- Discuss positron sources (if assuming the propagation)

Response handing (most important and challenging)

Currently (in DC2/3), we prepare response files for each (limited) science case, i.e., 511 keV, Al-26, Ti-44, continuum

- Having multiple detector response files is a hassle (continuum, line, imaging, polarization)
- Increasing the resolution of the detector response is not sustainable. Already too big.
- The current interpolation of the response is not very good
- etc. (from Israel's slides)

Parametrizing the response using relative coordinates by Israel

- https://drive.google.com/file/d/1_tGLfbYSf9bRpbCwJK3nQSfSDo_DJc-h/view?usp=sharing

An idea about the response matrix compression by HY

- https://drive.google.com/file/d/11p9XQC3lGU0L2iUO7Ym_kcZ3NqjWIpc-/view?usp=sharing

Neural network response by Pascal, Andreas

- The talk yesterday: https://indico.him.uni-mainz.de/event/227/contributions/1732/attachments/1037/1634/nn_response_workshop_pascal_janowski.pdf

Also, need to think about the computational performance (especially for the image analysis)

Good Time Intervals / Good event selection

Need to exclude time intervals, e.g.,

- when the background rate is high, like during SAA
- when a target source is not in the FoV
  - it is also important for the image deconvolution to maximize the sensitivity
- when a nearby source becomes very bright, e.g., blazar flare

Need to check if we should use all of the CDS

- cutting events with small/large scattering angles improves systematic uncertainties?

Background estimation

Both are an alpha-version, and there are lots of room to be improved!

- **Line background generator by Saurabh, HY**
  - Expecting the background event distribution from adjacent energy bins
  - It may not work for 511 keV because the lower adjacent energy bin includes positronium continuum emission
  - Should be better interpolations, e.g, on-/off-pointing data, neural network
- **Continuum background generator by Chris Karwin**
  - Masking the Compton-cone region of a target in CDS and filling the mask data space by interpolation
  - For multiple sources, this approach may not work well

Some ideas

- **Can estimate the background by separating a time-constant component (bkg) from a time-variable one (astronomical sources) in the local coordinate?**
- **Can include some detector information, e.g, saturated count rate of BGO, event rate from BTO?**
  - BTO paper: http://arxiv.org/abs/2501.16434
- **Can directly fit the data using Savitri's background simulation? Or can estimate some of background components by comparing data with the simulation?**