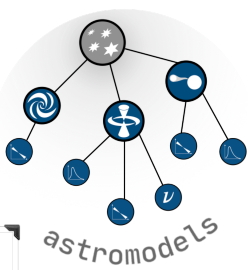


Advancements in Multi-Wavelength Data Analysis: The ThreeML Framework for High-Energy Astronomy

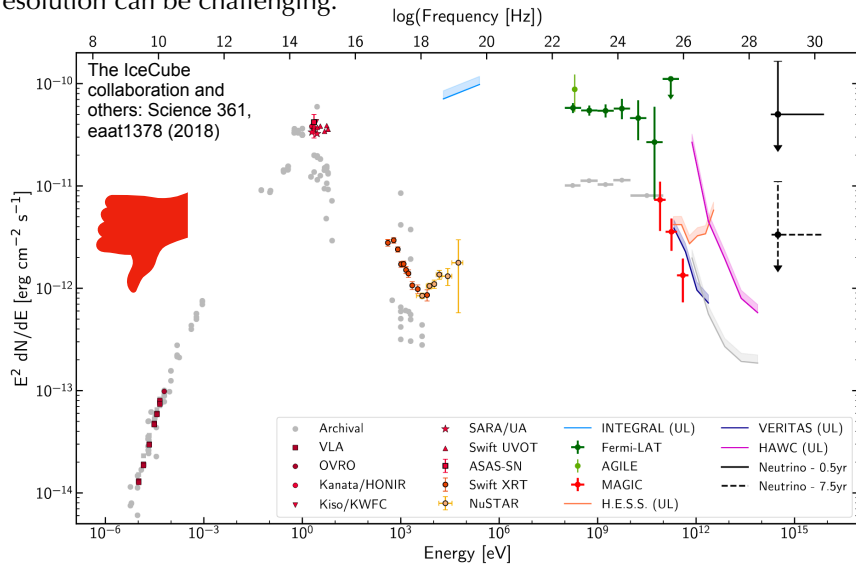
Nicola Omodei, Niccoló Di Lalla (Stanford University) and the ThreeML team



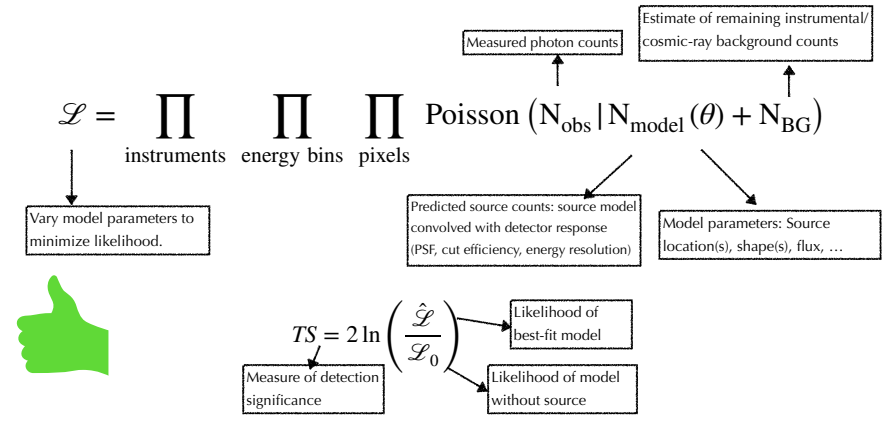
The Multi Mission Maximum Likelihood framework (*ThreeML*) is a Python-based software package designed for multi-wavelength data analysis in high-energy astronomy. Integrating X-ray and gamma-ray data from various instruments, along with measurements at lower wavelengths, is essential for unlocking the full potential of observational data. However, the lack of standardization and unique challenges posed by each instrument often complicate the process of combining data from multiple sources. *ThreeML* addresses these challenges with its flexible, plugin-based structure, allowing for the seamless inclusion of data from diverse observatories in their native formats. Leveraging *astromodels*, a versatile modeling framework, *ThreeML* enables separate handling of source modeling and data access from likelihood optimization, facilitating a flexible combination of both aspects. Moreover, in addition to frequentist maximum likelihood analysis, *ThreeML* supports Bayesian analysis through posterior distribution sampling.

Multi-Messenger and Multi-Wavelength Astronomy

- **Standard (not optimal) approach:** Unfold data, fit to high-level data (e.g. differential flux measurements, SED)
 - Unfolding makes (implicit) assumptions on spectrum, source size/shape etc. (not always consistent).
 - Weak sources: deal with upper limits, non-gaussian uncertainties etc.
 - Combining information from instruments with different angular/energy resolution can be challenging.

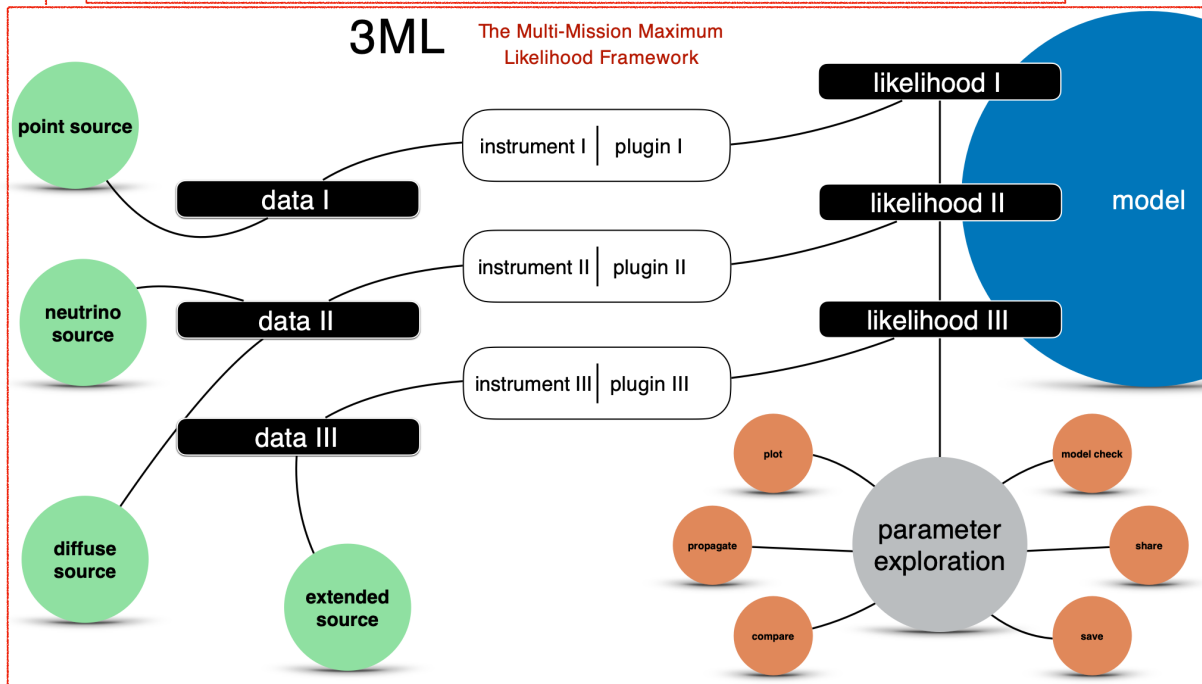


- **Better approach:** Forward-fold model, fit to low-level data (e.g. photon counts)
 - Model assumptions are made explicit.
 - **Plug-ins** used to deal with different formats for data/instrument response.



Modeling with astromodels

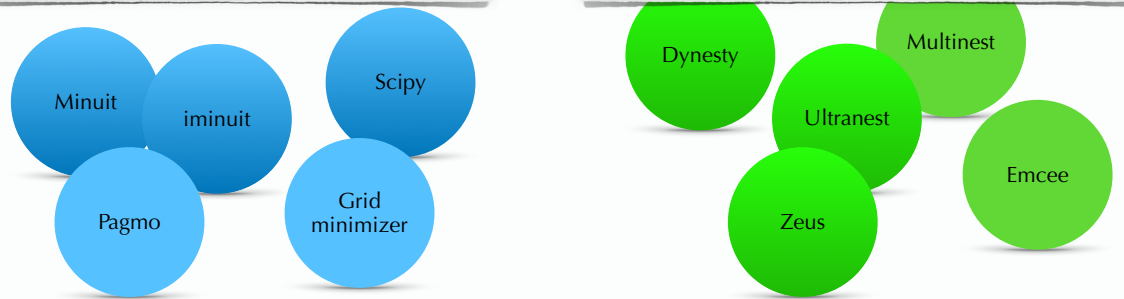
- A **model** contains one or more **sources**.
 - Sources have a spatial shape (point source/extended), one or more spectral components, with optional polarization.
 - Energy-dependent morphology also supported.
- **Phenomenological models:** Powerlaw, log-parabola, ...
- **Physical models:** Blackbody, photometric absorption, DM spectra, Synchrotron spectrum via *naima*, EBL absorption via *ebtable*, APEC models via *pyatomdb*
- Access to **all xspec models** (if *xspec/xspecmodelonly* installed).
- **Flexible:** Users can define their own functions at run time, or:
 - Read in tabulated spectra/fits maps.
 - Combine (add, multiply etc) existing functions (e.g. absorption terms).
- Parameters may be **linked** via arbitrary functions.



Minimizers and Samplers

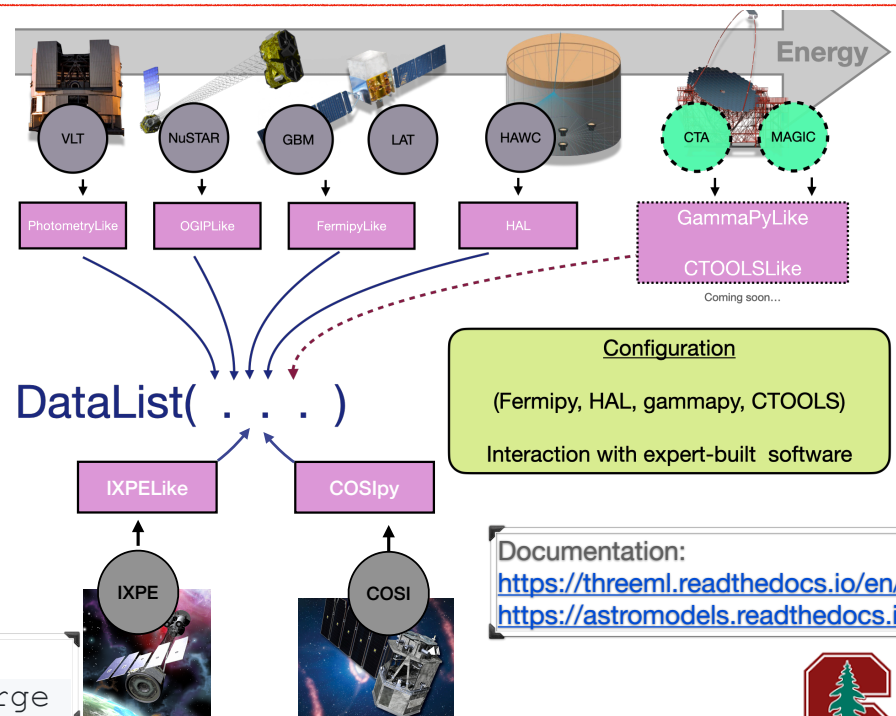
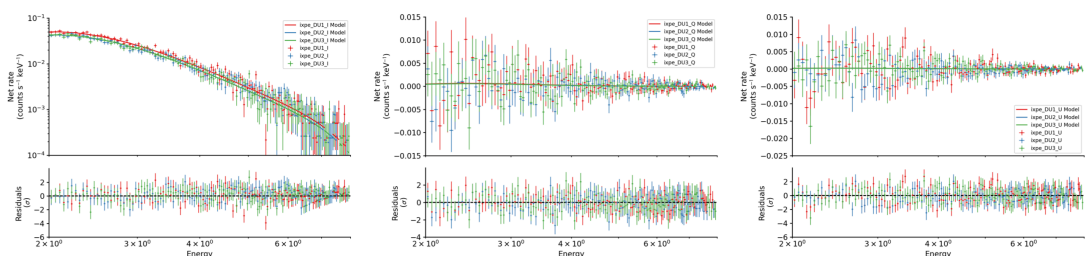
```
mle_analysis = JointLikelihood(model, datalist)
local_minimizer = LocalMinimization("minuit")
local_minimizer.setup(ftol=1e-4)
results = mle_analysis.fit()
```

```
bayes = BayesianAnalysis(model, datalist)
bayes.set_sampler("emcee")
bayes.sampler.setup(n_observations=100)
results = bayes.sample()
```



The Plugins

- Plugins are able to:
 - Read in data/instrument response files in "native" format.
 - Convolve model with detector response, calculate model predictions.
 - Calculate the likelihood.
 - Produce (pseudo-) random simulated datasets given a model to assess the goodness of fit.
 - Optional convenience functions for visualization etc.
- Plugins can be wrappers for external software (e.g. *fermipy*) or standalone (e.g. *HAL*).
- Joint likelihood fits between multiple plugins (same or different type)



```
pip install astromodels threeml
conda install astromodels threeml -c threeml conda-forge
```

