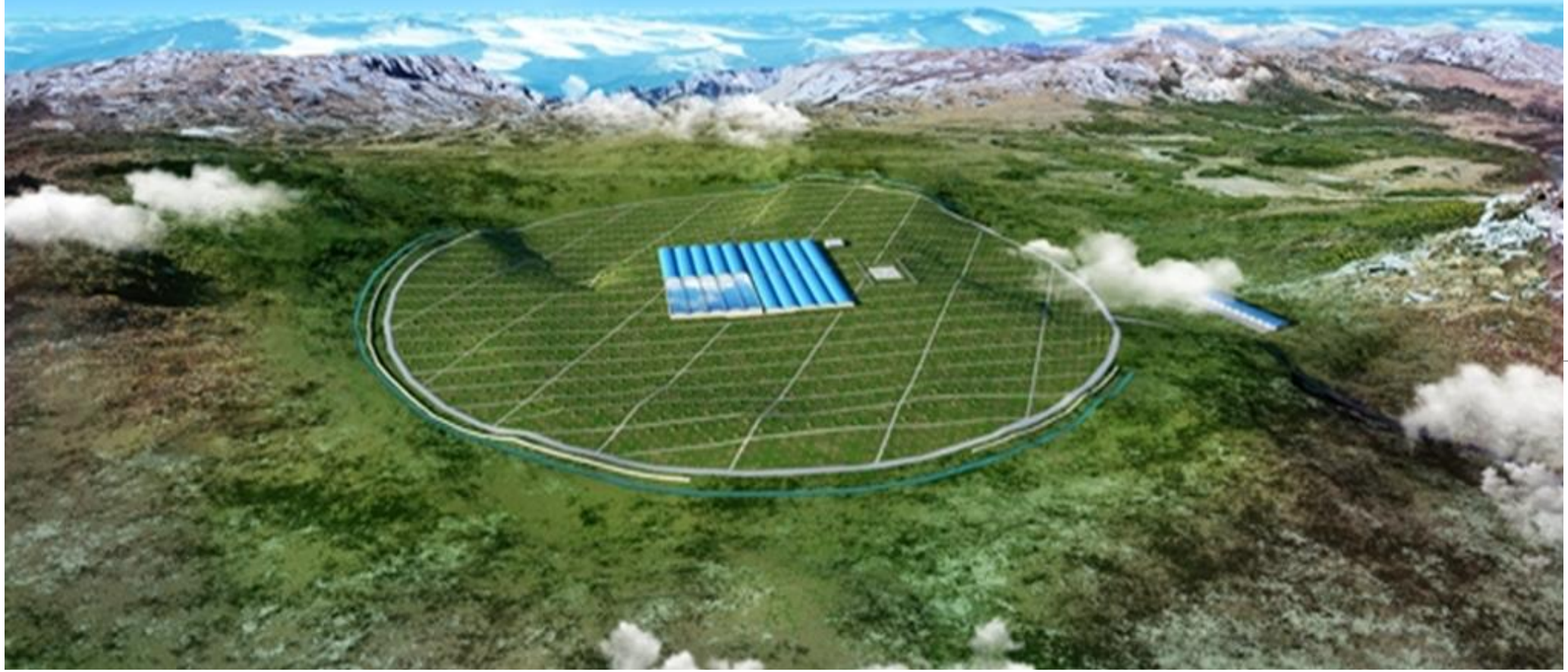
A 3D Likelihood Analysis Tool for LHAASO-KM2A data

Duan Kaikai and Huang Xiaoyuan

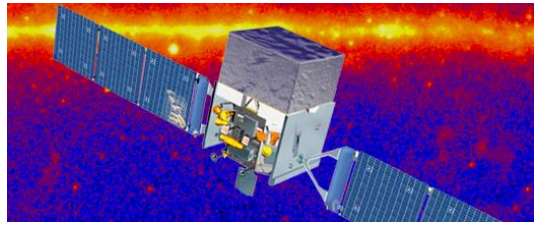
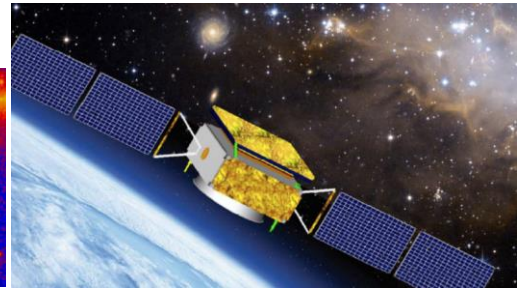
2021/10/14 @ Shanghai

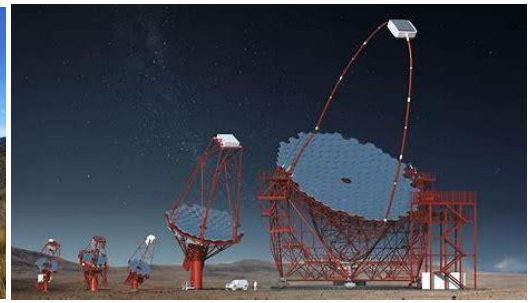
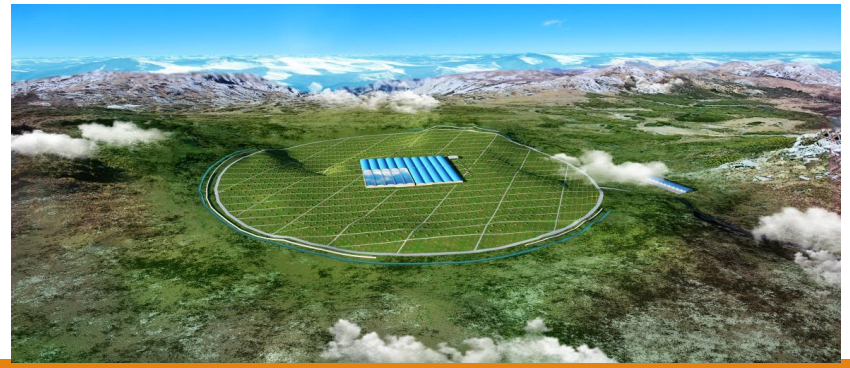


What's 3D likelihood

Spectrum + Spatial distribution Fit

- Space-base: Fermi-LAT, DAMPE
- Ground-base: HWAC, HESS, CTA

The source model is considered as:

$$S(E,\hat{p},t)=\sum_i s_i(E,t)\delta(\hat{p}-\hat{p}_i)+S_G(E,\hat{p})+S_{\rm eg}(E,\hat{p})+\sum_l S_l(E,\hat{p},t),$$

Point Sources | Galactic & EG Diffuse Sources | Other Sources

This model is folded with the Instrument Response Functions (IRFs) to obtain the predicted counts in the measured quantity space (*E',p',t'*):

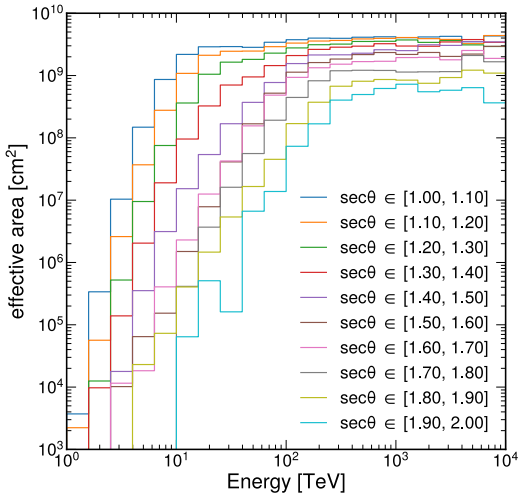
$$M(E',\hat{p}',t)=\int_{\rm SR} dEd\hat{p}R(E',\hat{p}',t;E,\hat{p})S(E,\hat{p},t)$$

where

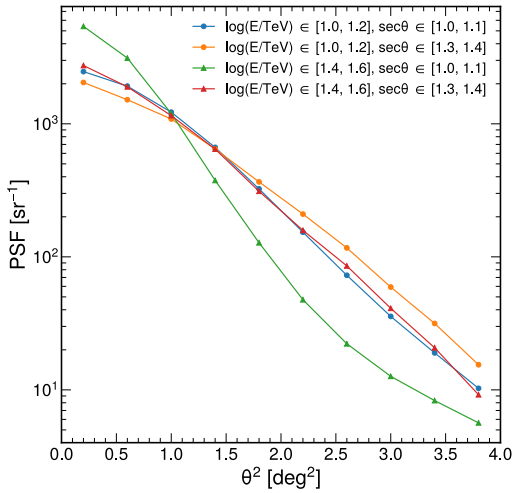
$$R(E',\hat{p}';E,\hat{p},t)=A(E,\hat{p},\vec{L}(t))D(E';E,\hat{p},\vec{L}(t))P(\hat{p}';E,\hat{p},\vec{L}(t))$$

KM2A IRFs from Simulation

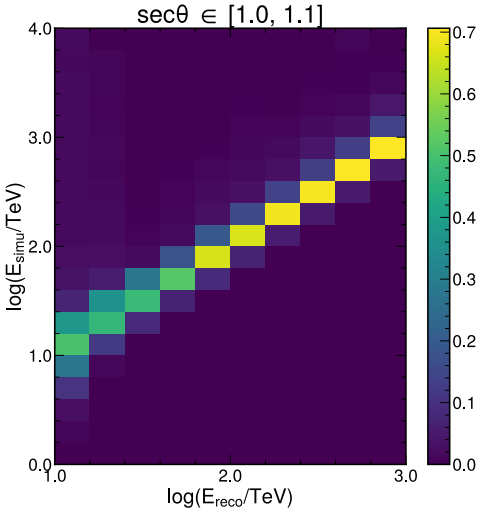
Instrument Response Functions (IRFs) including the effective area, point-spread function and energy dispersion represent the performance of the detections like sensitivity, angular and energy resolution.



Effective Area

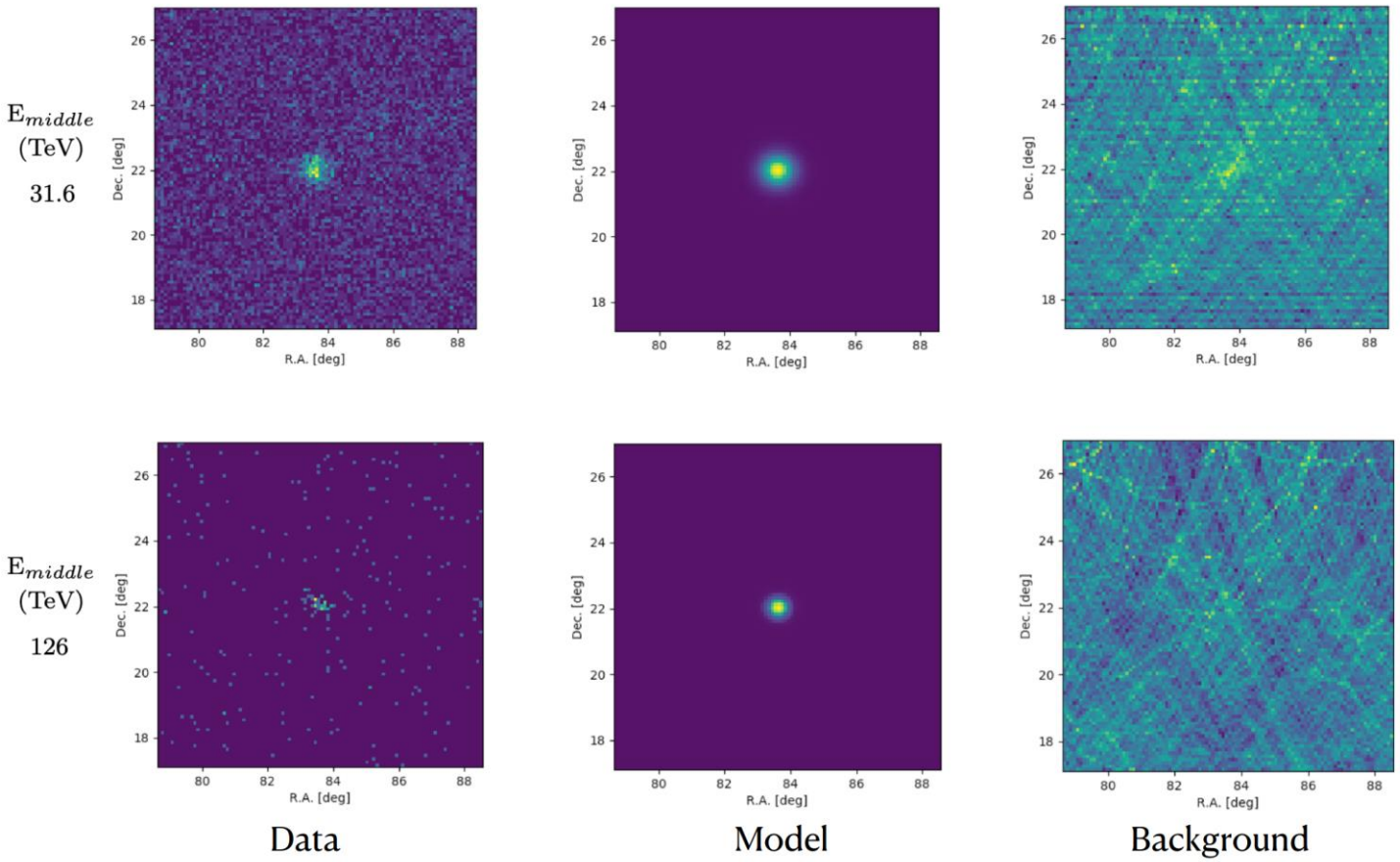


Point Spread Function

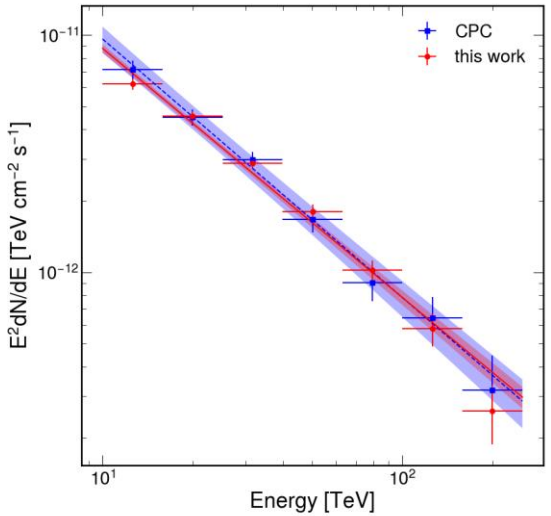


Energy Dispersion

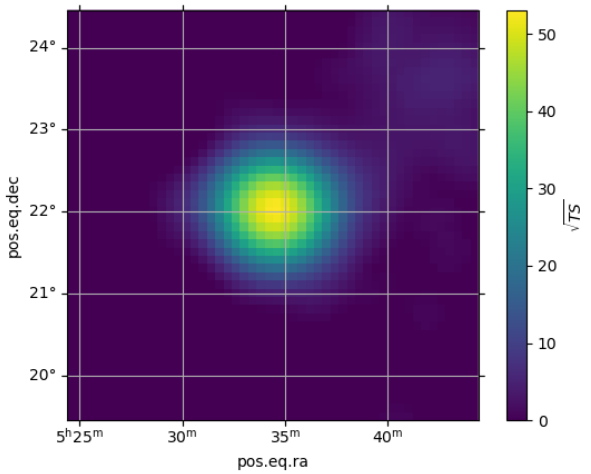
KM2A Data and Model Prediction



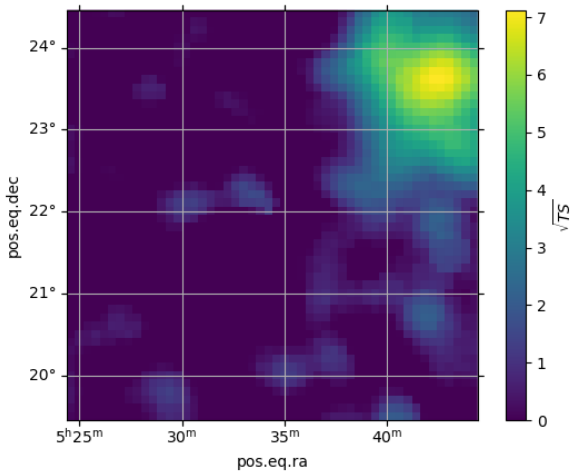
Analysis for Point Source



Spectral Energy Distribution (SED)



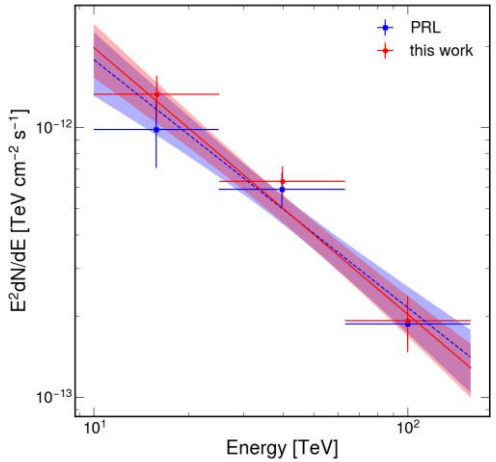
Test Statistic (TS) map



residual TS map

The analysis with this software could give consistent results with those using traditional method.

Analysis for Extended Source



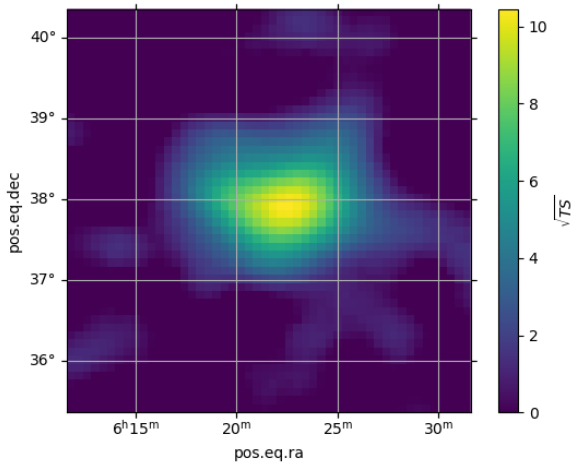


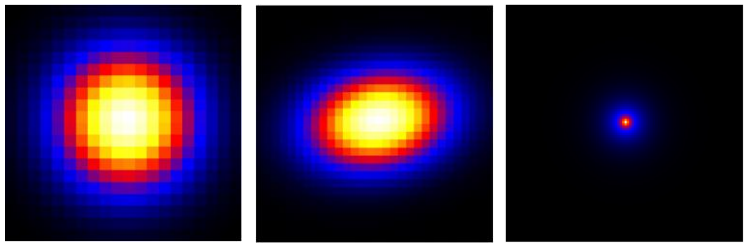
TABLE I. Results of the morphological analyses of LHAASO J0621+3755.

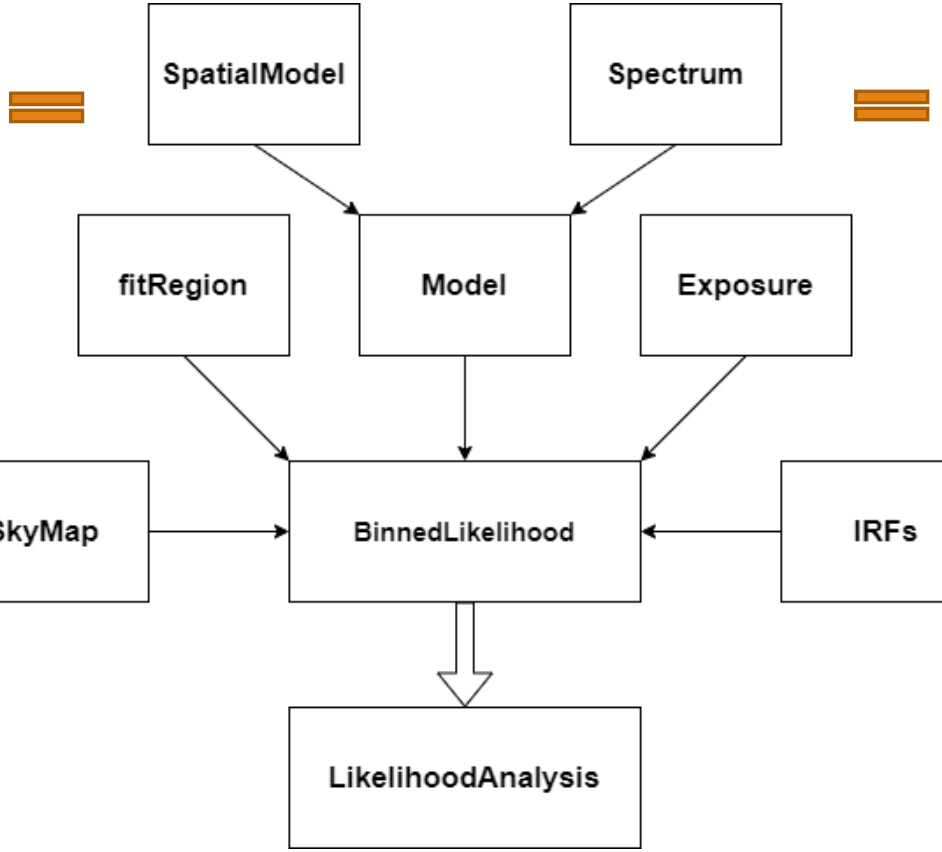
Template	Extension[a] (°)	RA (°)	Dec (°)	TS	N_p^b
Point source	-	95.56 ± 0.10	37.85 ± 0.07	63.0	3
2D Gaussian	0.40 ± 0.07	95.47 ± 0.11	37.92 ± 0.09	79.5	4
Uniform disk	0.70 ± 0.10	95.44 ± 0.11	37.94 ± 0.09	80.2	4
Diffusion	0.91 ± 0.20	95.48 ± 0.10	37.90 ± 0.09	78.1	4

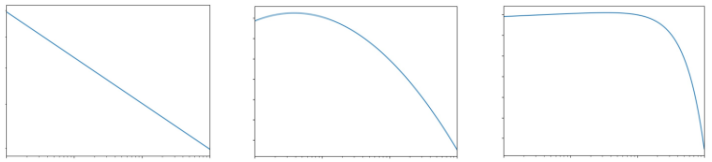
[a]Radius for the uniform disk; σ for the Gaussian model; θ_d for the diffusion model. [b]N_p is the number of parameters in the model.

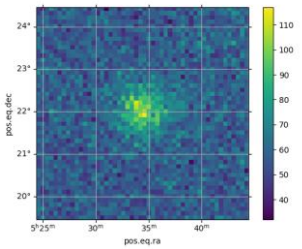
Template	Extension	RA	DEC	Prefactor	Index	TS
Point source	-	95.62+/-0.09	37.92+/-0.06	1.83+/-0.28	2.89+/-0.18	95.1
2D Gaussian	0.36+/-0.06	95.52+/-0.09	37.89+/-0.07	3.0+/-0.4	2.99+/-0.15	133.8
Uniform disk	0.65+/-0.12	95.51+/-0.08	37.92+/-0.07	2.9+/-0.4	3.01+/-0.15	130.9
Diffusion	0.82+/-0.17	95.54+/-0.08	37.88+/-0.06	3.1+/-0.4	2.99+/-0.14	125.0

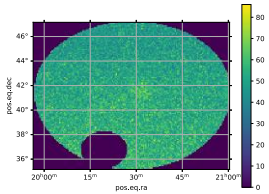
Implementation

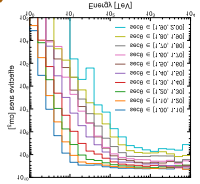


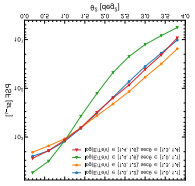


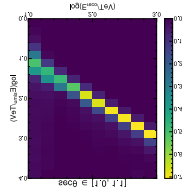


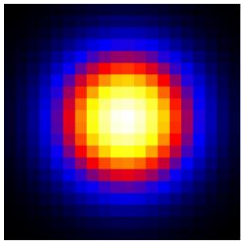
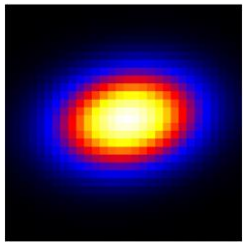
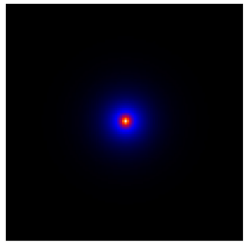
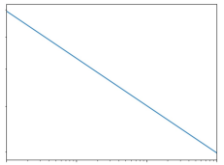
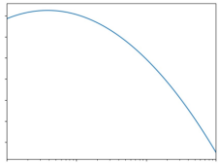
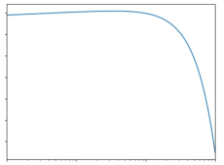










For Users - Input

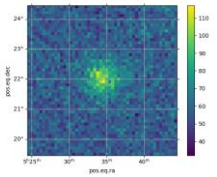
SpatialModel:
- Point Source
  - SkyDirFunction: RA, DEC
- Extended Source
  - RadialDisk: RA, DEC, Radius
  - RadialGaussian: RA, DEC, Sigma
  - Ring2D: RA, DEC, R_in, R_out
  - Ellipse2D: RA, DEC, A, B, Theta
  - Gaussian2D: RA, DEC, X_stddev, Y_stddev, Theta
  - PulsarHalo: RA, DEC, ThetaD
  - SpatialMap
  - MapCubeFunction

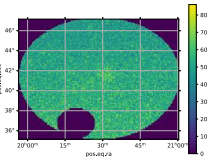
Spectrum:
- PowerLaw: Prefactor, Index, Scale
- BrokenPowerLaw: Prefactor, Index1, Index2, BreakValue
- LogParabola: norm, alpha, beta, Eb
- PLExpCutOff: Prefactor, Index, Scale, Cutoff
- ……

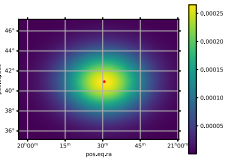
```
crab:
  name: crab
  spatialModel:
    DEC:
      free: 0
      max: 24
      min: 20
      name: DEC
      scale: 1
      value: 22.02
    RA:
      free: 0
      max: 84.0
      min: 83.0
      name: RA
      scale: 1
      value: 83.63
    type: SkyDirFunction
  spectrum:
    Index:
      error: 0.036741536956782284
      free: true
      max: 10
      min: 0
      name: Index
      scale: -1
      value: 3.0526707697089317
    Prefactor:
      error: 0.02858086637415047
      free: true
      max: 100
      min: 0
      name: Prefactor
      scale: 1e-14
      value: 1.059534645016219
    Scale:
      free: 0
      max: 100
      min: 0
      name: Scale
      scale: 1
      value: 20
```

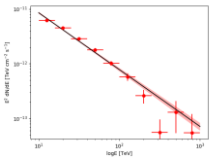
```
crab:
  name: crab
  type: PointSource
  spatialModel:
    type: SkyDirFunction
    RA: …
    DEC: …
  spectrum:
    type: LogParabola
    norm: …
    alpha: …
    beta: …
    Eb: …

newsrc:
  name: newsrc
  type: PointSource
  spatialModel:
    type: SkyDirFunction
    RA: …
    DEC: …
  spectrum:
    type: PowerLaw
    Prefactor: …
    Index: …
    Scale: …
```

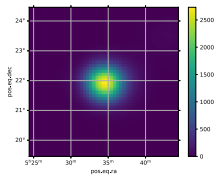
For Users - Output











Scripts:

- DataSelect.py name ra dec dx dy
  - plotCountsmap.py noncube.fits/nbkgcube.fits
  - plotfitRegion.py fitRegion.yaml
  - plotSpatialMap.py model_input.yaml srcName
  - plotSpectrum.py model_input.yaml srcName
- BinnedAnalysis.py srcName logEmin logEmax
  - plotSED.py srcName logEmin logEmax Ebins
  - plotTSmap.py TSmap.fits

```yaml
inRegion:
  CygnusCocoon:
    xref:
      value: 307.17
      unit: degree
    yref:
      value: 41.17
      unit: degree
    frame: fk5
    rad:
      value: 6
      unit: degree

outRegion:
  src:
    xref:
      value: 304.85
      unit: degree
    yref:
      value: 36.80
      unit: degree
    frame: fk5
    rad:
      value: 1.5
      unit: degree
```

```yaml
LikelihoodValue: -708945.8533701402
Nbkg_LowerError: '-0.003'
Nbkg_UpperError: '0.003'
Nbkg_error: '0.003'
Nbkg_value: '1.006'
crab:
  DEC_value: '22.02'
  Flux_error: '1.31e-14'
  Flux_value: '4.19e-13'
  Index_LowerError: '-0.04'
  Index_UpperError: '0.04'
  Index_error: '0.04'
  Index_scale: '-1.00e+00'
  Index_value: '3.04'
  Npred: '4928.40'
  Prefactor_LowerError: '-0.03'
  Prefactor_UpperError: '0.03'
  Prefactor_error: '0.03'
  Prefactor_scale: '1.00e-14'
  Prefactor_value: '1.04'
  RA_value: '83.63'
  Scale_value: '20.00'
  TsValue: 2830.3610210744664
```

For Developers

Data: on/off cube data

Time: lt_mjd or lt_lst

IRFs: aeff, psfcube, edispcube

More SpatialModel and Spectrum

```python
class PowerLaw(Spectrum):
    def GetSpecFunc(self, e):
        n = self.GetParValue('Prefactor')
        gamma = self.GetParValue('Index')
        e0 = self.GetParValue('Scale')

        return n * (e / e0)**gamma


class BrokenPowerLaw(Spectrum):
    def GetSpecFunc(self, e):
        n = self.GetParValue('Prefactor')
        gamma1 = self.GetParValue('Index1')
        gamma2 = self.GetParValue('Index2')
        eb = self.GetParValue('BreakValue')

        if type(e) in (int, float):
            if e < eb:
                dnde = n * (e / eb)**gamma1
            else:
                dnde = n * (e / eb)**gamma2
        else:
            dnde = np.zeros_like(e)
            dnde[e <= eb] = n * (e[e <= eb] / eb)**gamma1
            dnde[e >= eb] = n * (e[e >= eb] / eb)**gamma2
        return dnde
```

```python
class RadialGaussian(SpatialModel):
    def GetSpatialMap(self, ccube=None):
        self.RA = self.GetParValue('RA')
        self.DEC = self.GetParValue('DEC')
        self.Sigma = self.GetParValue('Sigma')

        if ccube:
            skycrd = self.GetSkycrd()
            sep = skycrd.separation(ccube.skycrd).to(u.rad).value
            data = 1/(2*np.pi*np.deg2rad(self.Sigma)**2) * np.exp(-sep**2 / np.deg2rad(self.Sigma)**2
            / 2).reshape(ccube.nxpix, ccube.nypix)
            wcs = ccube.wcs
        else:
            deltax = deltay = 0.1
            nxpix = nypix = 10 * self.Sigma / deltax
            sigma = self.Sigma / deltax
            mod = models.Gaussian2D(1., nxpix/2, nypix/2, sigma, sigma, 0)

            x, y = np.mgrid[0:nxpix, 0:nypix]
            data = mod(x, y)
            data = data / (data * np.deg2rad(deltax) * np.deg2rad(deltay)).sum()
            wcs = maptools.make_wcs(nxpix, nypix, deltax, deltay, 'C', self.RA, self.DEC, 'CAR')
        return SkyMap.NewSpatialMap(data, wcs)

    def GetSkycrd(self):
        return SkyCoord(ra=self.RA*u.degree, dec=self.DEC*u.degree, frame='fk5')
```

Any suggestions or contributions are welcome!

Summary and Plans

We provide a 3D likelihood analysis tool for KM2A data analysis.

With the half-array data and IRFs, the results are consistent with traditional method.

We plan to apply this tool for 3-quarters and full-array of KM2A and WCDA data.

Summed likelihood for KM2A data and joint Fit for KM2A and WCDA data.

Append Healpix projection for large region analysis.

Thanks for your attention