Friedrich-Alexander-Universität
Erlangen-Nürnberg

Using Penalizing Likelihood Method and Nuisance Parameters in Gammapy

Katrin Streil
Erlangen, 28.10.2022

Friedrich-Alexander-Universität
Erlangen-Nürnberg

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

Using Penalizing Likelihood Method and Nuisance Parameters in Gammapy

Parameters which are not of intrinsic interest (in contrast to model parameters), but describe any kind of systematic uncertainty.

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Using **Penalizing Likelihood Method** and Nuisance Parameters in Gammapy

Maximum Likelihood method with an additional penalizing term describing the behavior of the Nuisance Parameters (to avoid over-fitting)

Applications:

- In general: describing any kind of systematic uncertainty
- Examples:
  - Uncertainty on the effective area (in progress)
  - Uncertainty on the energy reconstruction
  - Uncertainty on the 3D background modeling ….

Motivation: Fit Statistics

- **Cash** Poisson data with background **model**

$$C = 2 \times (\mu_{\rm sig} + \mu_{\rm bkg} - n \times log(\mu_{\rm sig} + \mu_{\rm bkg}))$$

- N: number of counts, Poisson random variable
- mu_sig/ mu_bkg: expected number of counts from the source / bkg
- MapDataset, SpectrumDataset
- **Not** including background estimation uncertainties

- **Wstat** Poisson data with background **measurement**

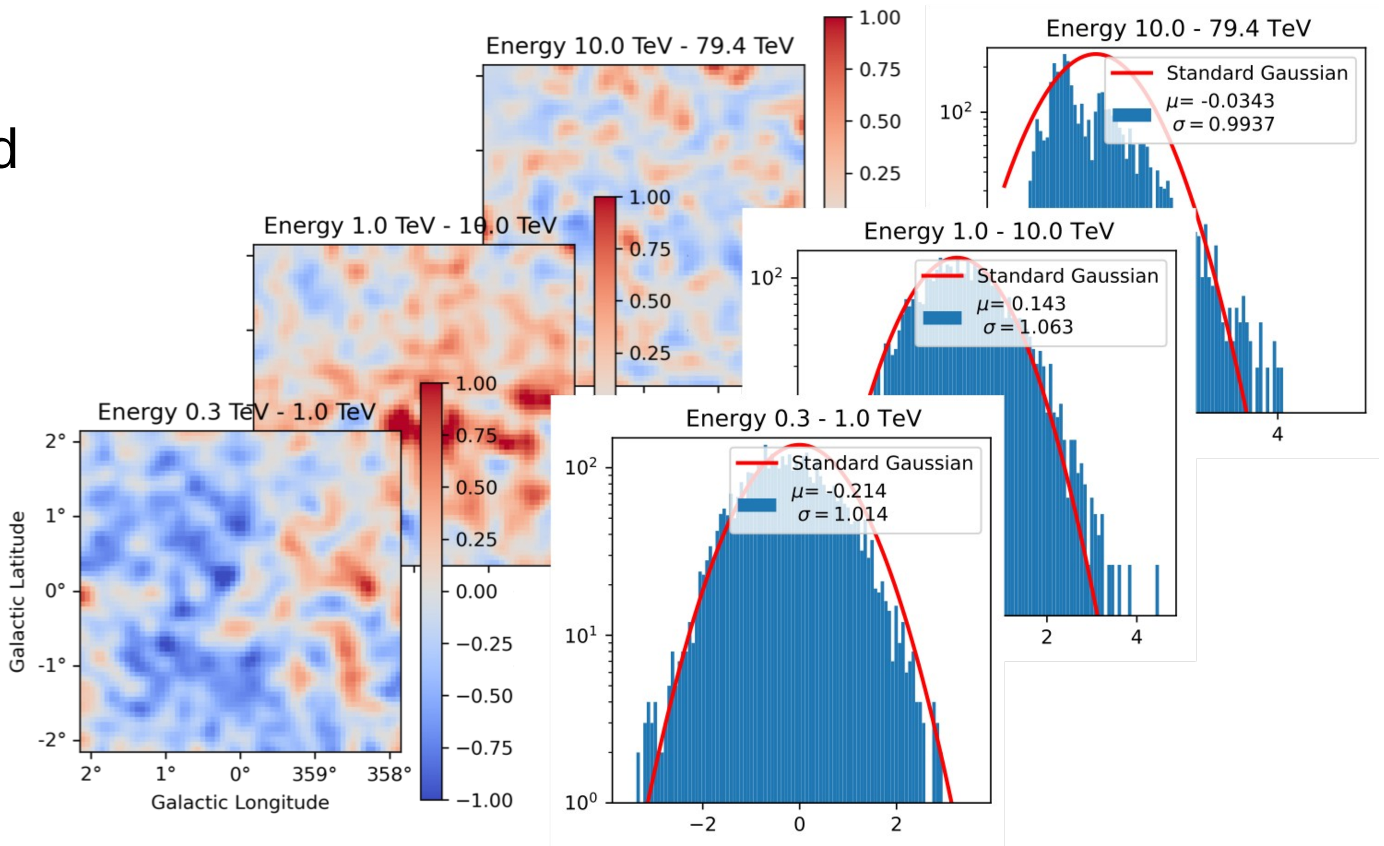
$$W = 2\big(\mu_{\rm sig} + (1 + 1/\alpha)\mu_{\rm bkg} - n_{\rm on} \log(\mu_{\rm sig} + \mu_{\rm bkg}) - n_{\rm off} \log(\mu_{\rm bkg}/\alpha)\big)$$

- n_off/ n_on: number of counts in OFF region (bkg only) / ON region (bkg + signal)
- Alpha: ratio of ON and OFF acceptances
- MapDatasetOnOff, SpectrumDatasetOnOff
- Including background estimation uncertainties

Motivation: Fit Statistics

- Example: 3D analysis of the Galactic Center
- Residual maps show mismodelled background
- Significance distribution deviating from the Normal distribution

→ Nuisance parameters to describe the systematics due to background mismodelling



Penalizing Likelihood Method

Nuisance Parameters

- Set of Parameters to describe the 'background perturbations'
- Change the background prediction (bin-wise weights)
- Larger Parameters space → widening of the Lscan → Larger model parameter uncertainties
- N bins → N additional parameters
- Penalizing term to avoid over-fitting

$$\boxed{\bar{\mathrm{bg}}_k = \mathrm{bg}_k \cdot (1 + \vec{\mu}_k)}$$

$$\mathcal{L}_k = \frac{\mu_k^{n_k}}{n_k!} \exp(-\mu_k)$$

$$-2 \ln \mathcal{L}_k = -2 \left[n_k \ln(\mu_k) - \mu_k \right]$$

$$\bar{\mathcal{L}}_k = \frac{\mu_k^{n_k}}{n_k!} \exp(-\mu_k) \cdot \exp\left[-\frac{1}{2} \Delta N_k \sum_{l=0}^{N} (K^{-1})_{kl} \Delta N_l \right]$$

$$-2 \ln \bar{\mathcal{L}}_k = -2 \left[n_k \ln(\mu_k) - \mu_k \right] + \left[\Delta N_k \sum_{l=0}^{N} (K^{-1})_{kl} \Delta N_l \right]$$

Penalizing Likelihood Method

Penalty Term

- Gaussian Penalty term
- Correlation matrix K
  - Describing the systematic
  - Matrix-product of the spatial and spectral correlation matrices
  - Gaussian: parameterized by amplitude and length

$$\boxed{\bar{\mathrm{bg}}_k = \mathrm{bg}_k \cdot (1 + \vec{\mu}_k)}$$

$$\mathcal{L}_k = \frac{\mu_k^{n_k}}{n_k!} \exp(-\mu_k)$$

$$-2 \ln \mathcal{L}_k = -2 \left[n_k \ln(\mu_k) - \mu_k \right]$$

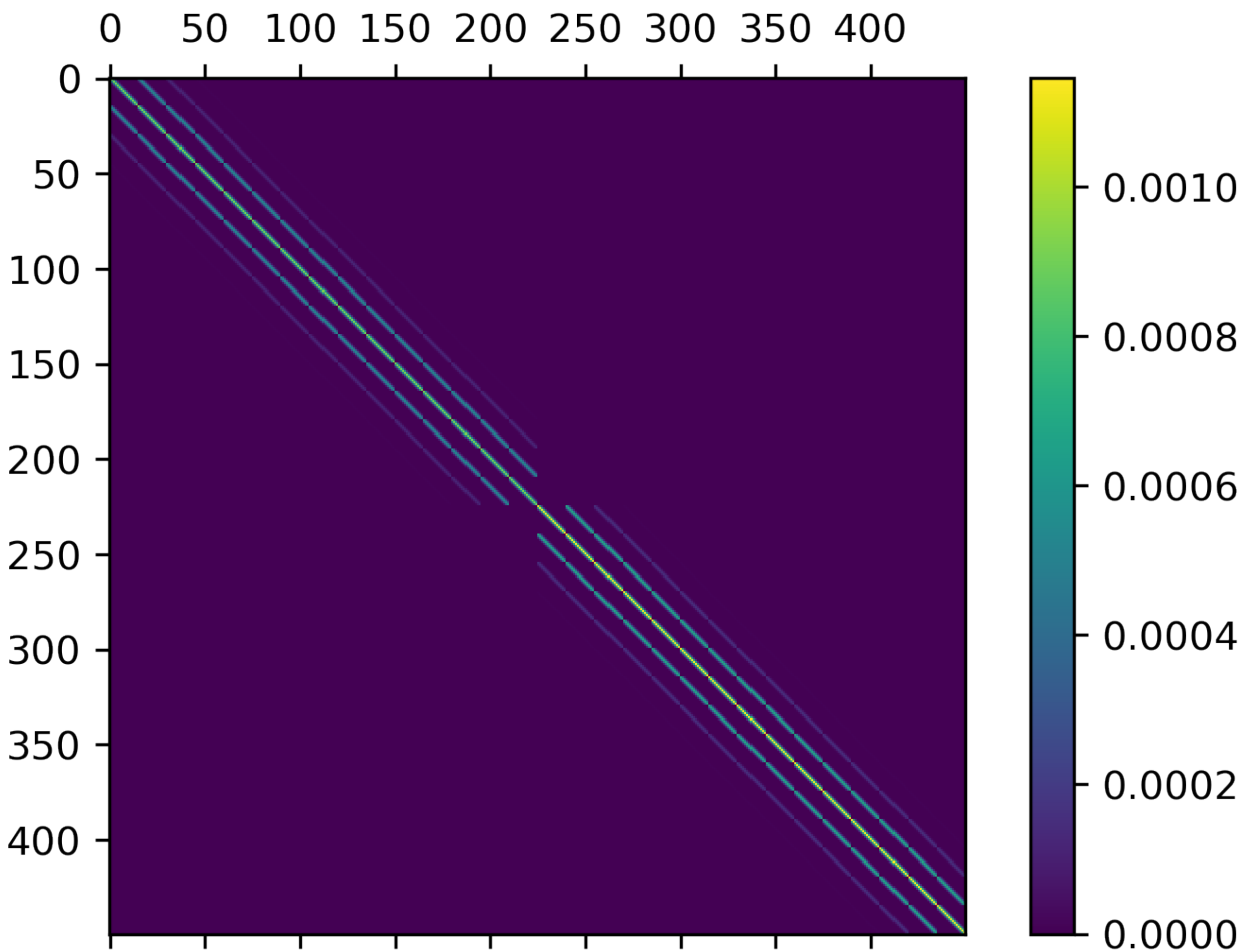
$$\bar{\mathcal{L}}_k = \frac{\mu_k^{n_k}}{n_k!} \exp(-\mu_k) \cdot \exp\left[-\frac{1}{2} \Delta N_k \sum_{l=0}^{N} (K^{-1})_{kl} \Delta N_l \right]$$

$$-2 \ln \bar{\mathcal{L}}_k = -2 \left[n_k \ln(\mu_k) - \mu_k \right] + \left[\Delta N_k \sum_{l=0}^{N} (K^{-1})_{kl} \Delta N_l \right]$$

Penalizing Likelihood Method

Penalty Term

- Gaussian Penalty term
- Correlation matrix K
  - Describing the systematic
  - Matrix-product of the spatial and spectral correlation matrices
  - Gaussian: parameterized by amplitude and length
  - (two analysis methods to determine the values)



Penalizing Likelihood Method

Implementation

```
class MapDatasetNuisance(MapDataset):
    """Map dataset for likelihood fitting with nuisance parameters to describe the systematics.
```

inv_corr_matrix : `~numpy.array`

Inverted correlation matrix to describe the correlations between the nuisance parameters.
Needs to have same dimension as nuisance parameters.

N_parameters : `~ gammapy.modeling.Parameters`

Nuisance Parameters to be fitted.

nuisance_mask : `~ gammapy.map`

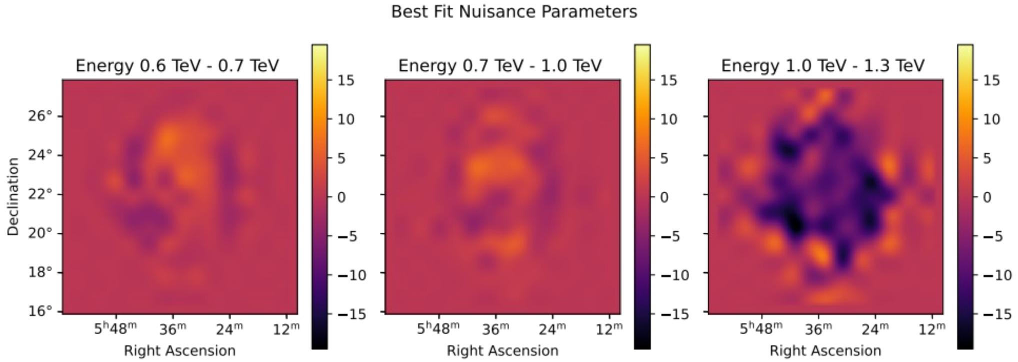
Masking the regions where the nuisance parameters are applied to the background.

→ Reduction of the amount of parameters (computation time):
Not one parameter for each bin, but averaging the systematic over multiple neighbor bins (depending on the angular size of the systematics)

Penalizing Likelihood Method

Implementation – Nuisance Parameters

```
def N_map(self):
    """Map of the Nuisance parameters

    Helper function to evaulate the nuisance parameters in the correct geometry.

    Returns
    -------
    npred_background : `Map`
        Predicted counts from the nuisance parameters.
    """
    if self.nuisance_mask is not None:
        from scipy.interpolate import interp2d
        N_map = Map.from_geom(self.geoms['geom_down'])
        N_map.data[np.where(self.nuisance_mask== True)] = self.N_parameters.value
        fac = int(self.geoms['geom'].data_shape[1] / self.geoms['geom_down'].data_shape[1])
        N_map_up = Map.from_geom(self.geoms['geom'])

        x = self.geoms['geom_down'].to_image().get_idx()[0][0] * fac + fac/2
        y = self.geoms['geom_down'].to_image().get_idx()[0][0] * fac + fac/2
        x_new =  self.geoms['geom'].to_image().get_idx()[0][0]
        y_new =  self.geoms['geom'].to_image().get_idx()[0][0]

        for e, z_e in enumerate(N_map.data):
            f = interp2d(x = x, y= y, z =z_e, kind='cubic',fill_value = None, bounds_error = False )
            N_map_up.data[e,:,:] = f(x_new, y_new).reshape(
                np.shape(N_map_up.data[0,:,:]))

    return N_map_up
```



Penalizing Likelihood Method

Implementation – Nuisance Parameters

```python
def npred_background(self):
    """Predicted background counts

    The predicted background counts depend on the parameters
    of the `FoVBackgroundModel` defined in the dataset and on the values of the nuisance parameters.

    Returns
    -------
    npred_background : `Map`
        Predicted counts from the background.
    """
    background = self.background
    if self.background_model and background:
        if self._background_parameters_changed:
            values = self.background_model.evaluate_geom(geom=self.background.geom)
            if self._background_cached is None:
                self._background_cached = background * values
            else:
                self._background_cached.quantity = (
                    background.quantity * values.value
                )
        if self.N_parameters is not None:
            self._background_cached.data = self._background_cached.data* (1+ self.N_map().data)
        return self._background_cached
    else:
        return background

    return background
```

Penalizing Likelihood Method

Implementation - Likelihood

$$-2\ln\bar{\mathcal{L}}_k = -2\left[n_k\ln(\mu_k) - \mu_k\right] + \left[\Delta N_k\sum_{l=0}^{N}(K^{-1})_{kl}\Delta N_l\right]$$

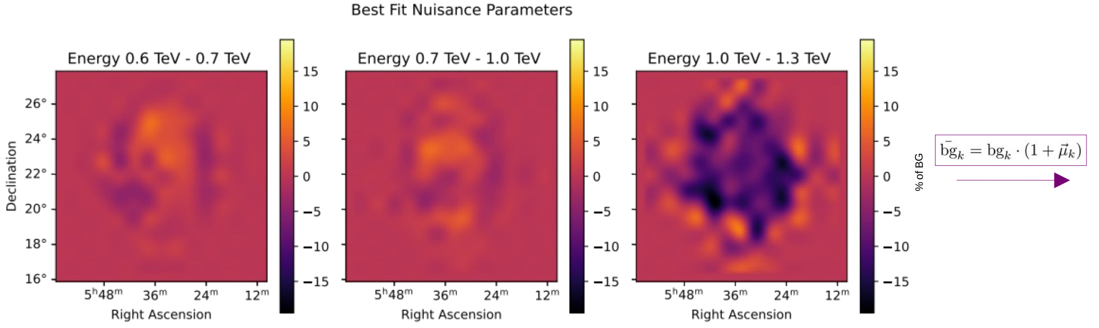
```python
def stat_array(self):
    """Likelihood per bin given the current model parameters + Gaussian of N parameters"""
    G =   np.matmul(self.inv_corr_matrix  ,self.N_parameters.values.ravel() )*self.N_parameters.values.ravel()
    stat =  cash(n_on=self.counts.data, mu_on=self.npred().data)
    return  G+stat

def stat_sum(self):
    """Total likelihood given the current model parameters + Gaussian of N parameters."""
    counts, npred = self.counts.data.astype(float), self.npred().data
    G =  np.matmul(np.matmul(self.inv_corr_matrix   ,self.N_parameters.value.ravel() ),self.N_parameters.value.ravel() )
    if self.mask is not None:
        return cash_sum_cython(counts[self.mask.data], npred[self.mask.data]) + G
    else:
        return cash_sum_cython(counts.ravel(), npred.ravel()) + G
```

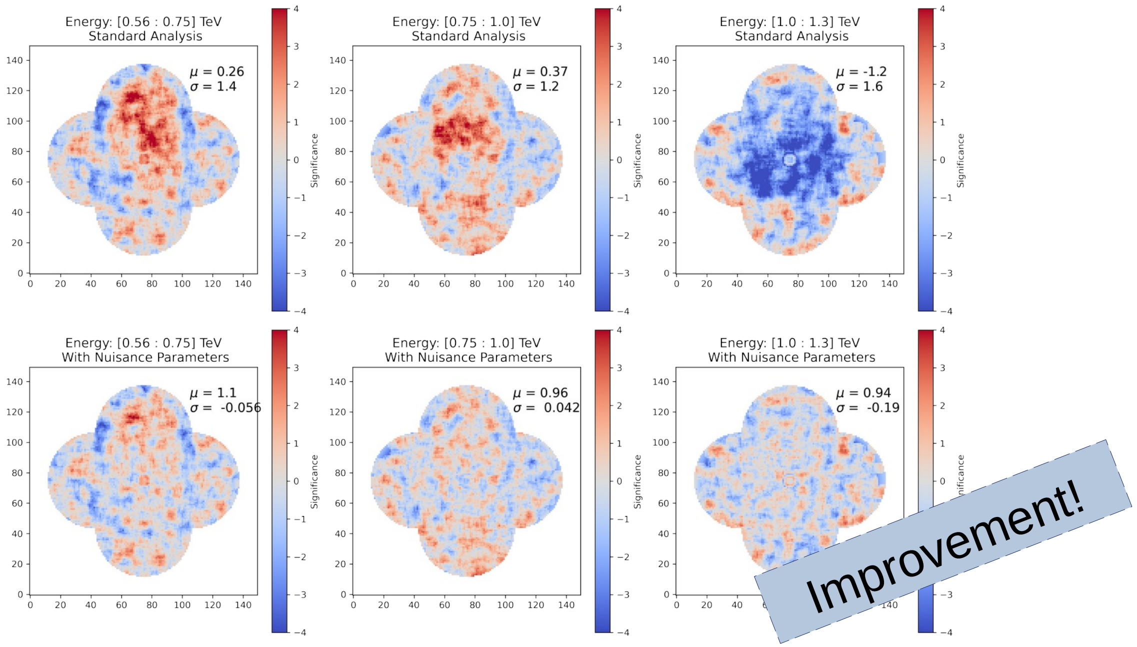
Additional parameters are used in methods like: from_hdulist, slice_by_idx, etc.

Fitting

Results and Comparison to Standard Method

- Obtain best fit nuisance and model parameters
- Improved Significance map and distribution
- Model parameter errors including statistical and systematic errors

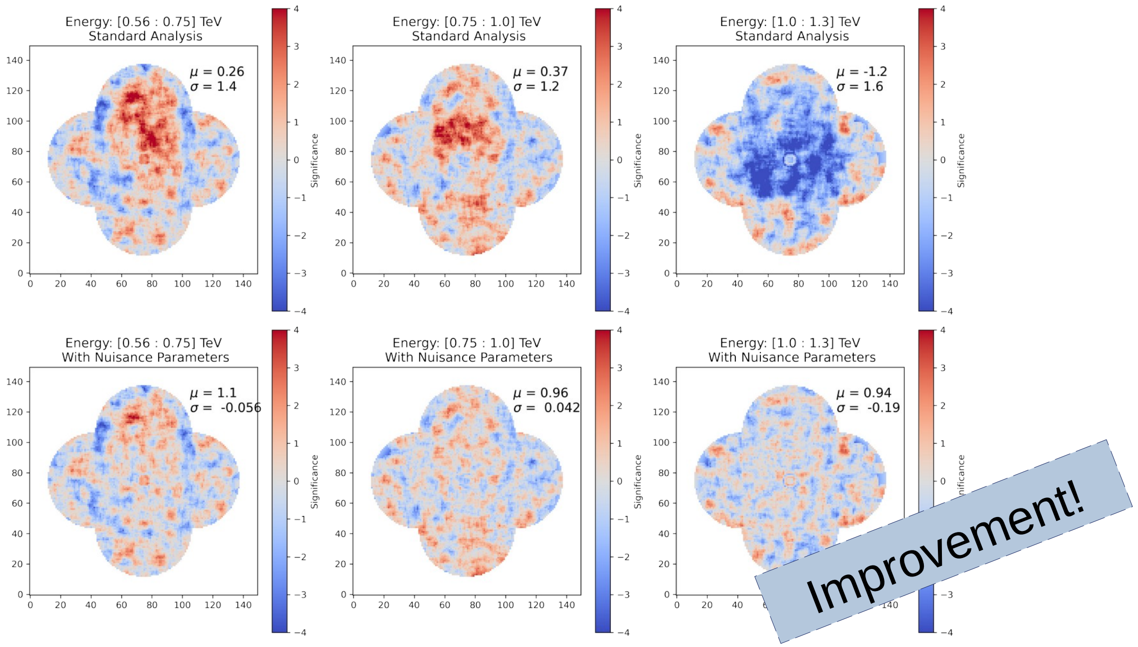




	Amplitude [1e-11 cm-2 s-1 TeV-1]	Spectral Index	Bg Normalisation
Standard Analysis	$3.729 \pm_{stat} 0.025$	$2.557 \pm_{stat} 0.0143$	$0.997 \pm_{stat} 0.002$
With Nuisance Parameters	$3.739 \pm_{stat} 0.025 \pm_{sys} 0.001$	$2.555 \pm_{stat} 0.0143 \pm_{sys} 0.0003$	$1.007 \pm_{stat} 0.002 \pm_{sys} 0.0001$

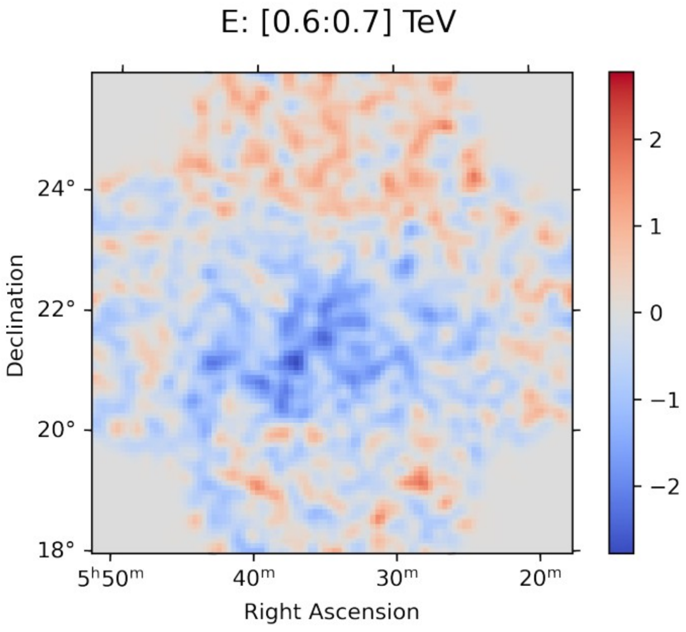
Summary

- Binned Maximum Likelihood method in 3D with Gaussian penalizing Prior term
- Nuisance parameters to describe systematic uncertainties of unknown distribution
- Advantages:
  - Improved background description
  - Automatic computation of the systematic error on the model parameters
- Outlook:
  - Testing and quantifying improvement and robustness (Asimov datasets)
  - Application (for now mainly GC)
  - Including other sources of systematics (effective area, energy reconstruction)

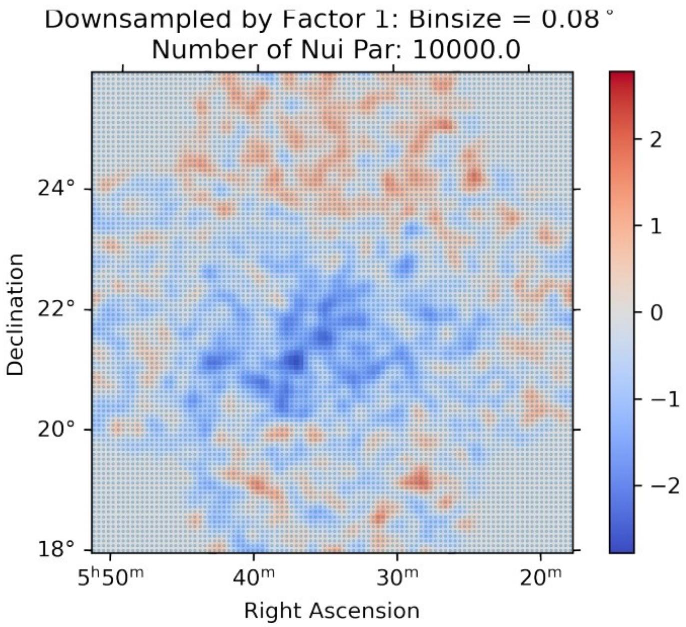


Backup

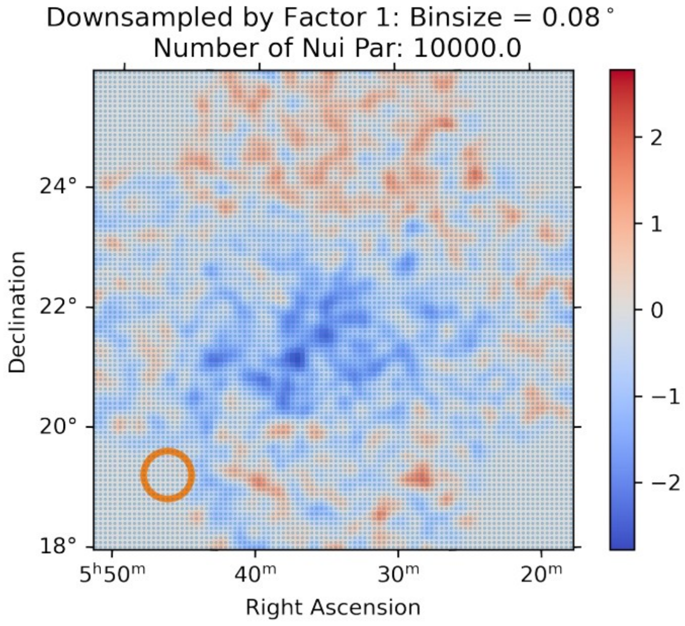
Method of Down-sampling



Smoothed spatial residual of the standard analysis



Adding the spatial nuisance parameters in each bin

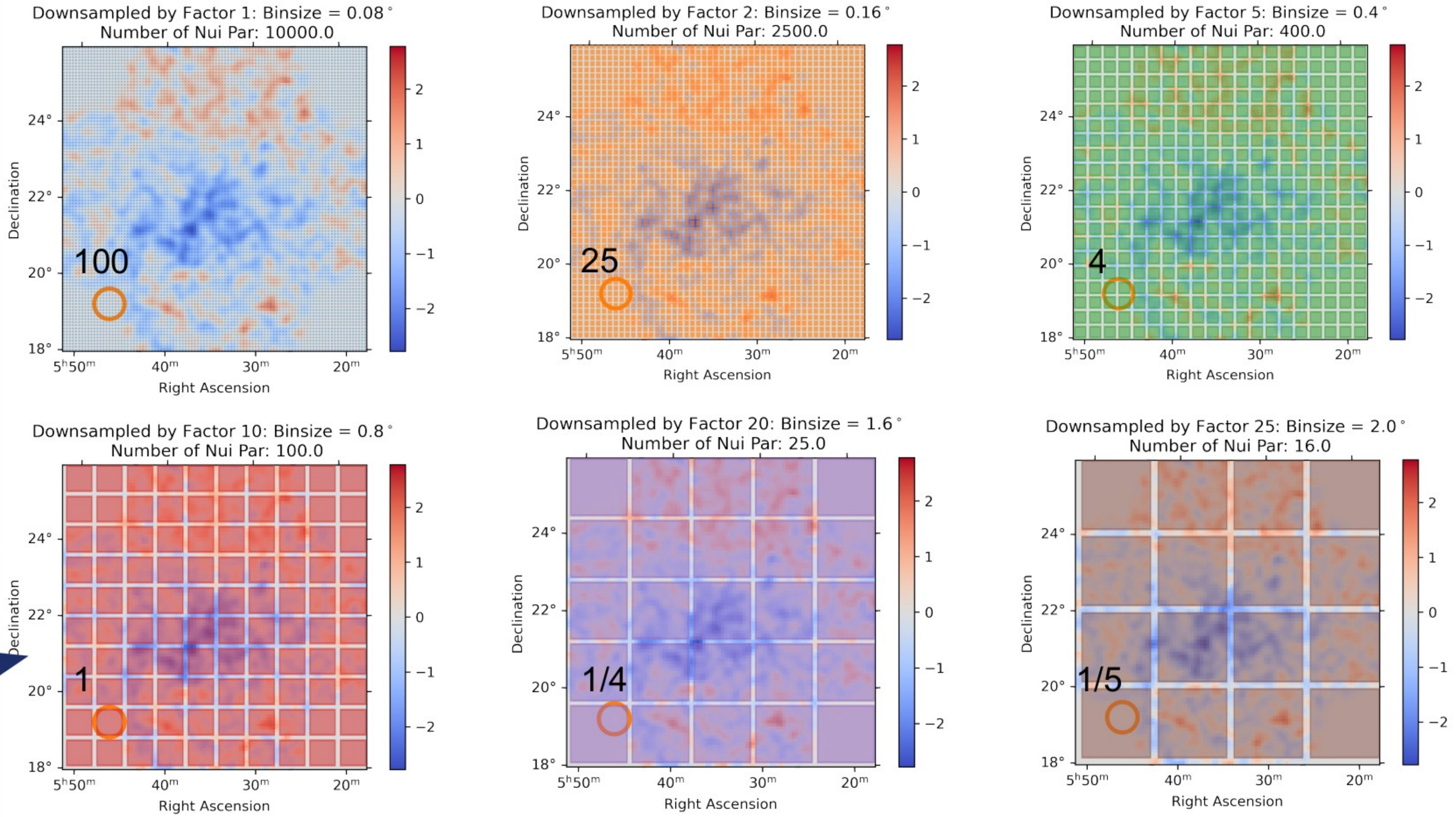


Visualisation of the Gaussian spatial correlation length of 0.8 deg = 100 strongly correlated nuisance parameters

Method of Down-sampling

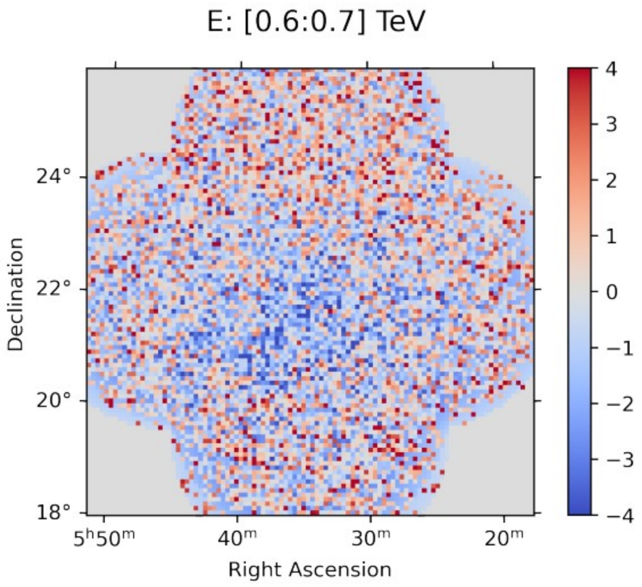
Idea:
Use one nuisance parameter as average over multiple spatial neighbor bins!
Use different binning for
the nuisance parameters
than for counts cube

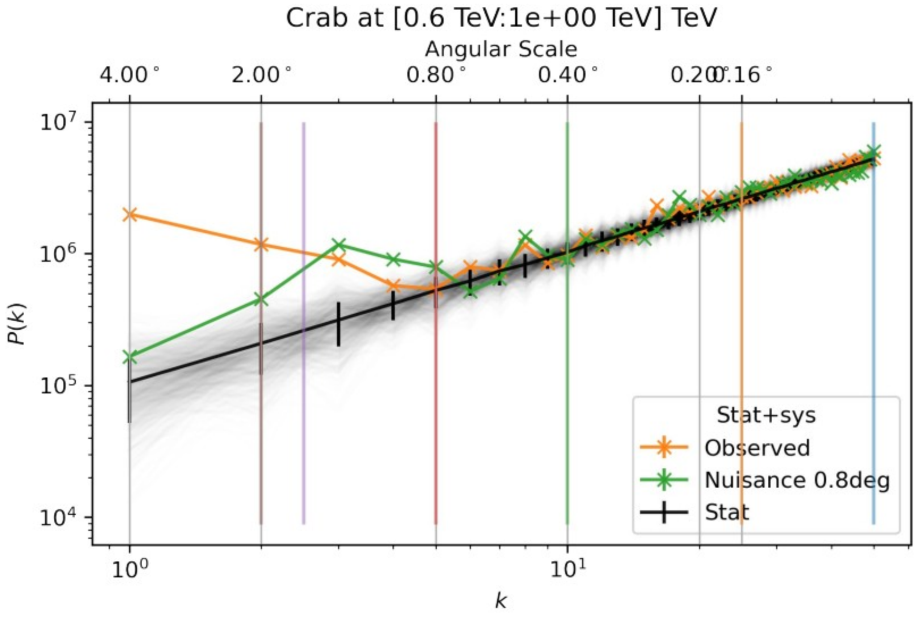
Choose the bin size such
that one bin is correlation length



Fourier Trafo to Obtain Correlation Length

- What is the optimal amplitude σ and correlation length?
- **Spatial residual of the standard analysis:**
  - Angular spectrum from 2D Fourier transformation
  - Compare with angular spectrum of datasets containing only statistical fluctuations





OFF-data Analysis to Obtain Amplitude