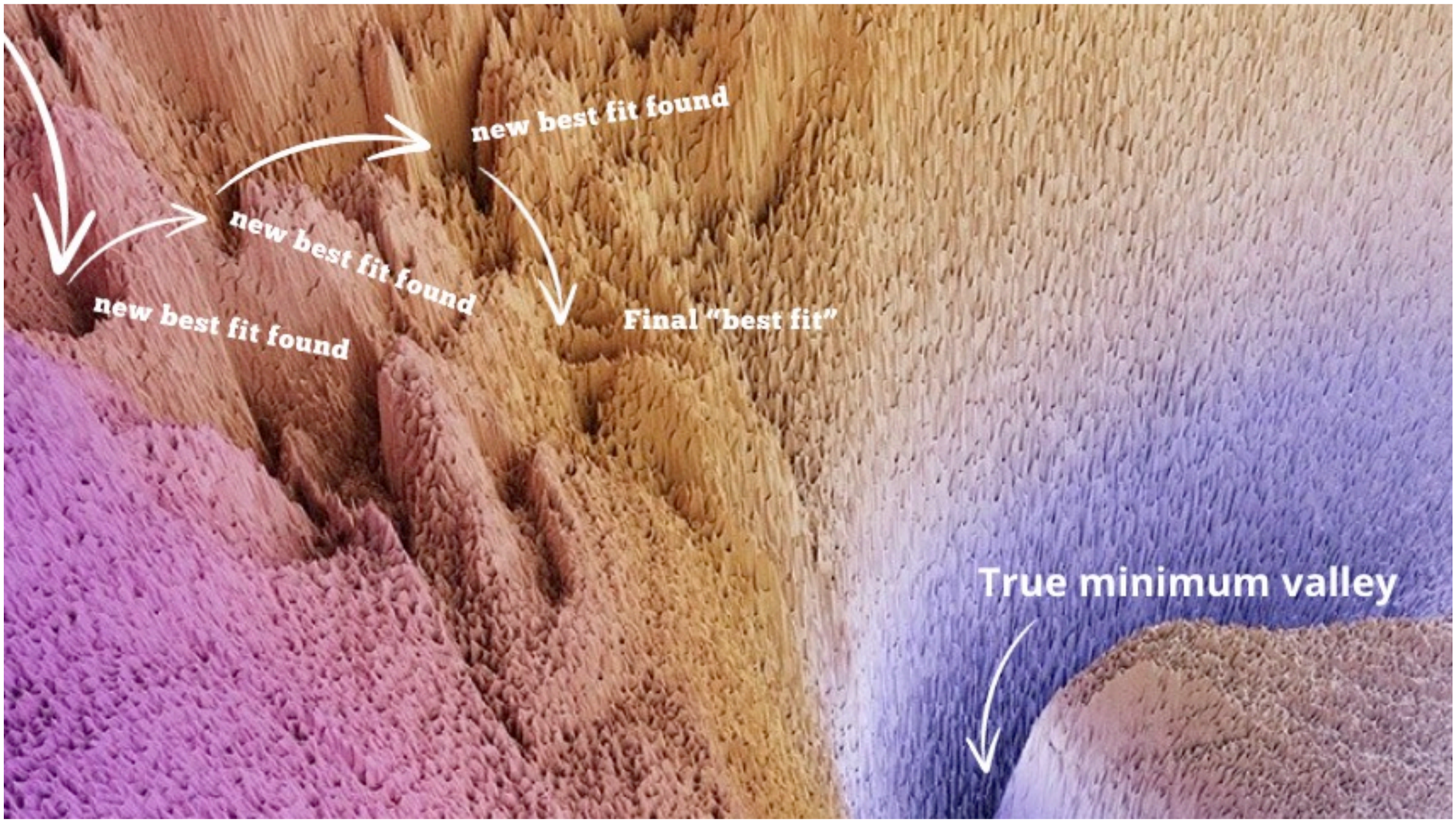
Likelihood minimization & model fitting

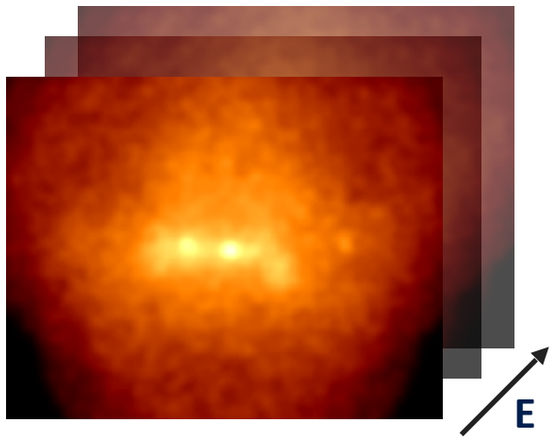
Illustration of the loss landscape for a Deep Neural Network on ImageNet



https://losslandscape.com/gallery/

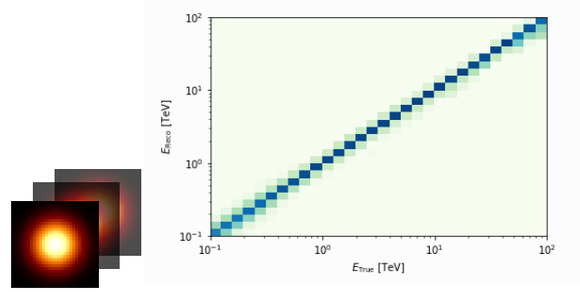
Datasets modeling and fitting

A binned analysis



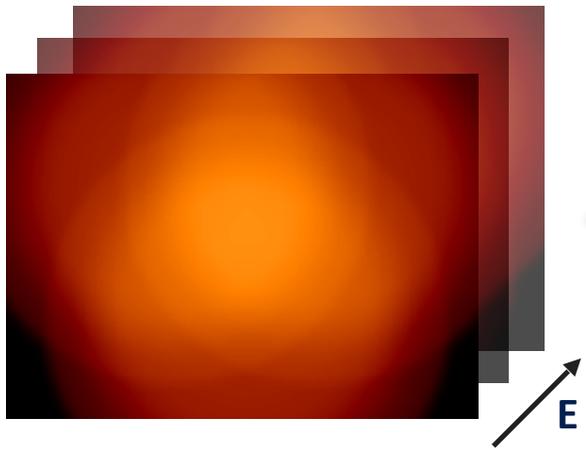
Reduced data

I/O



Reduced IRFs

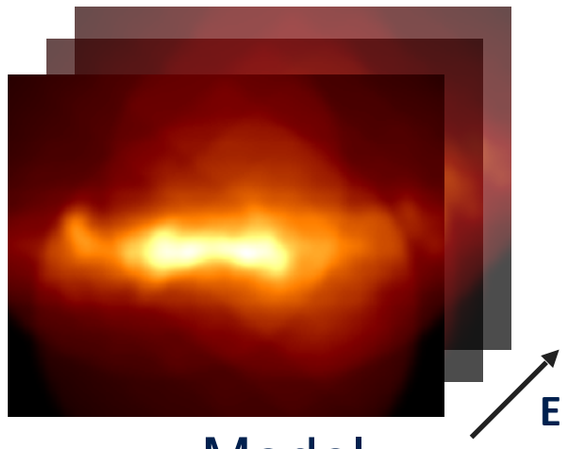
```
fit = Fit(optimize_opts={"print_level": 1})
result = fit.run(datasets=[dataset])
```



Bkg model

E

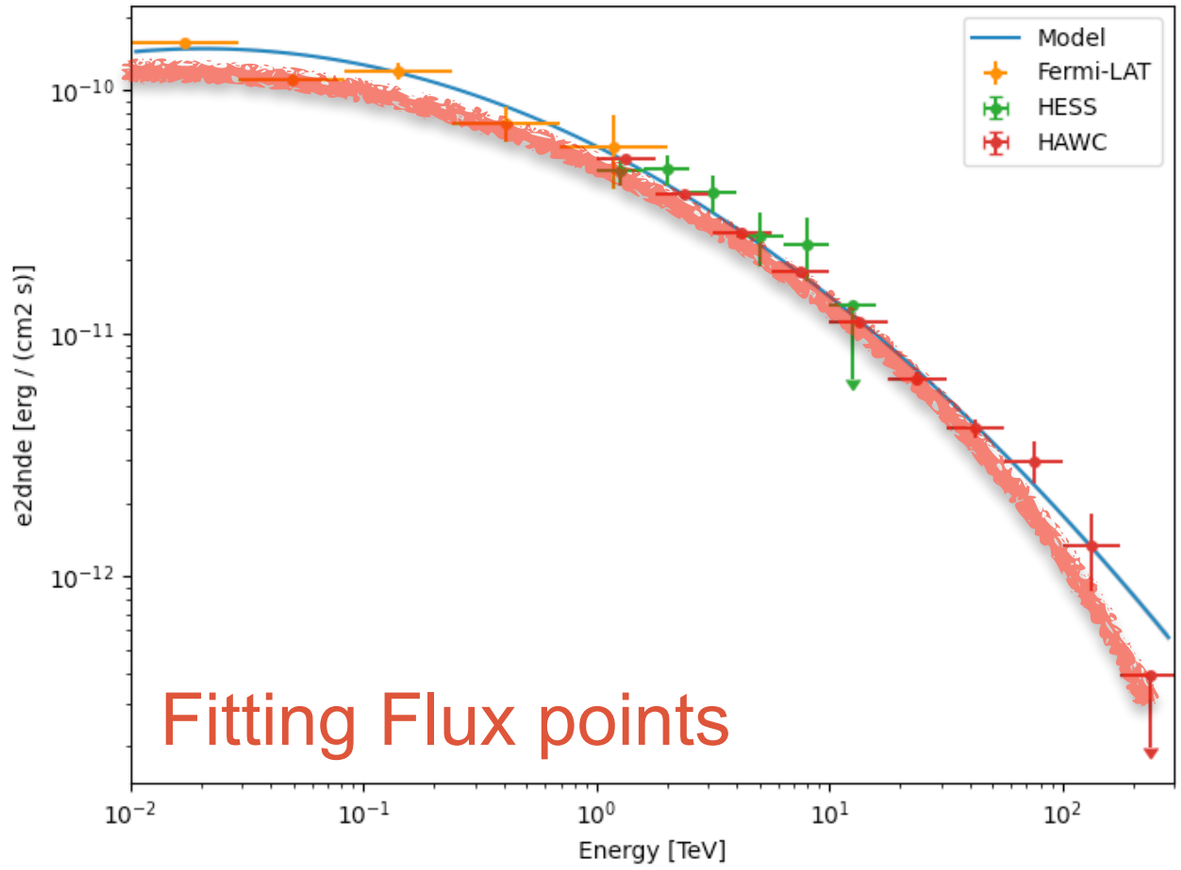
$L(x \mid \theta)$

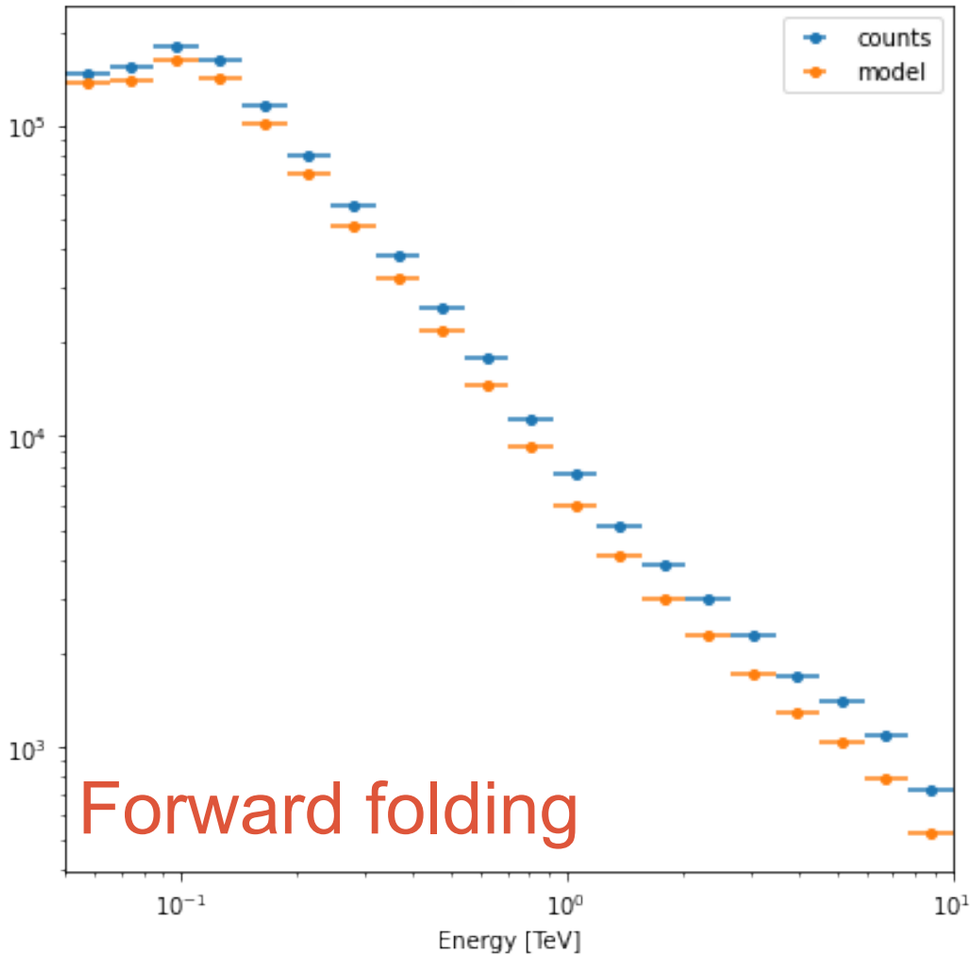


Model

see: [Dataset fitting tutorial]

What is forward folding ?





- Data are transformed to physical information
- Flux point modeling : a chi2 fit on flux points
  - Loss of statistical information
  - No handling of correlation between points

- Data are not transformed: N_{obs}
- The physical model is
  - Flux -> N_{pred} counts
- Proper statistical treatment
  - In particular for low counts

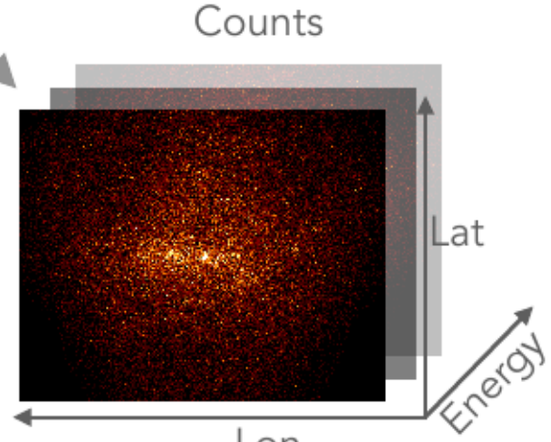
Poisson Log-Likelihood

List of gamma-like events…

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
5407363825684	123890826.66805482	84.97964	23.89347	10.352011
5407363825695	123890826.69749284	84.54751	21.004095	4.0246882
5407363825831	123890827.23673964	85.39696	19.41868	2.2048872

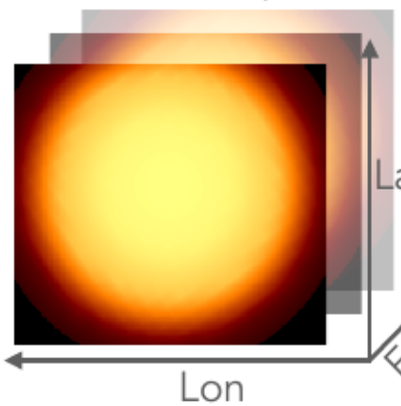
…binned into…

$N_{Obs} =$



Bkg Template

$N_{Bkg} =$



$\cdot\, n_0 \cdot \left(\frac{E}{E_0}\right)^{-\Gamma}$

"Cash statistics": summed over all "bins"

$$\mathscr{C} = 2 \sum_i N^i_{Pred} - N^i_{Obs} \cdot \log N^i_{Pred}$$

i: spectral channels or 3D voxels

$$N_{Pred} = N_{Bkg} + \sum_{Src} N_{Pred,Src}$$

- Predicted counts are **computed per model component** ("source / object") and summed
- A **"global" background model** template with "correction parameters" is added

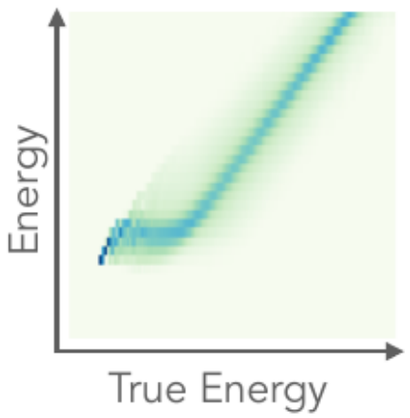
How is N_{pred} obtained ?

An analytical source model or template is **"forward folded" through the instrument response function (IRF)** to predict the measured number of counts…

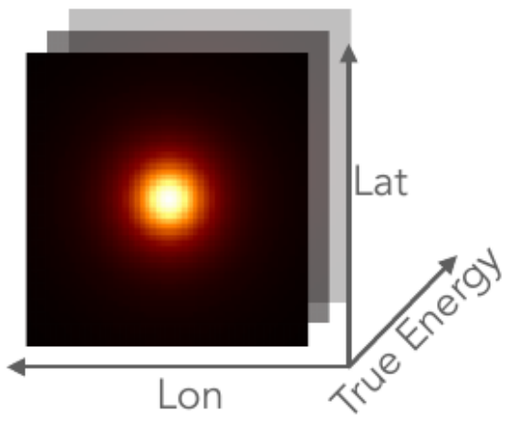
$$N_{Pred,Src} = \mathrm{EDISP}_{Src}(\mathrm{PSF}_{Src}(\mathscr{E}_{Src} \cdot f_{Src}))$$

$$f_{Src} = f_{Spectral}(E) \cdot f_{Spatial}(E, l, b) \cdot f_{Temporal}(t)$$

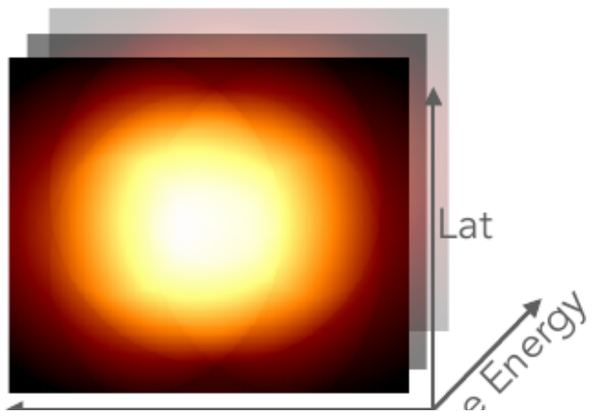
Energy Dispersion Matrix



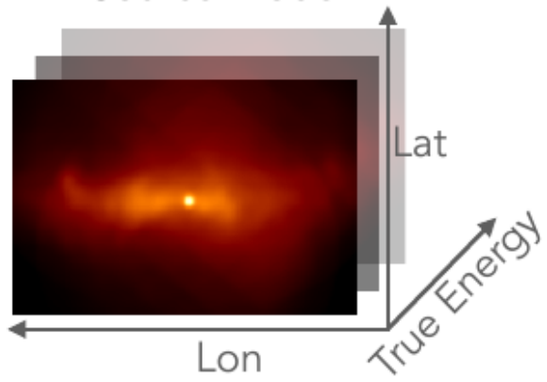
PSF Kernel



Exposure
(eff. area x lifetime)



Source Model



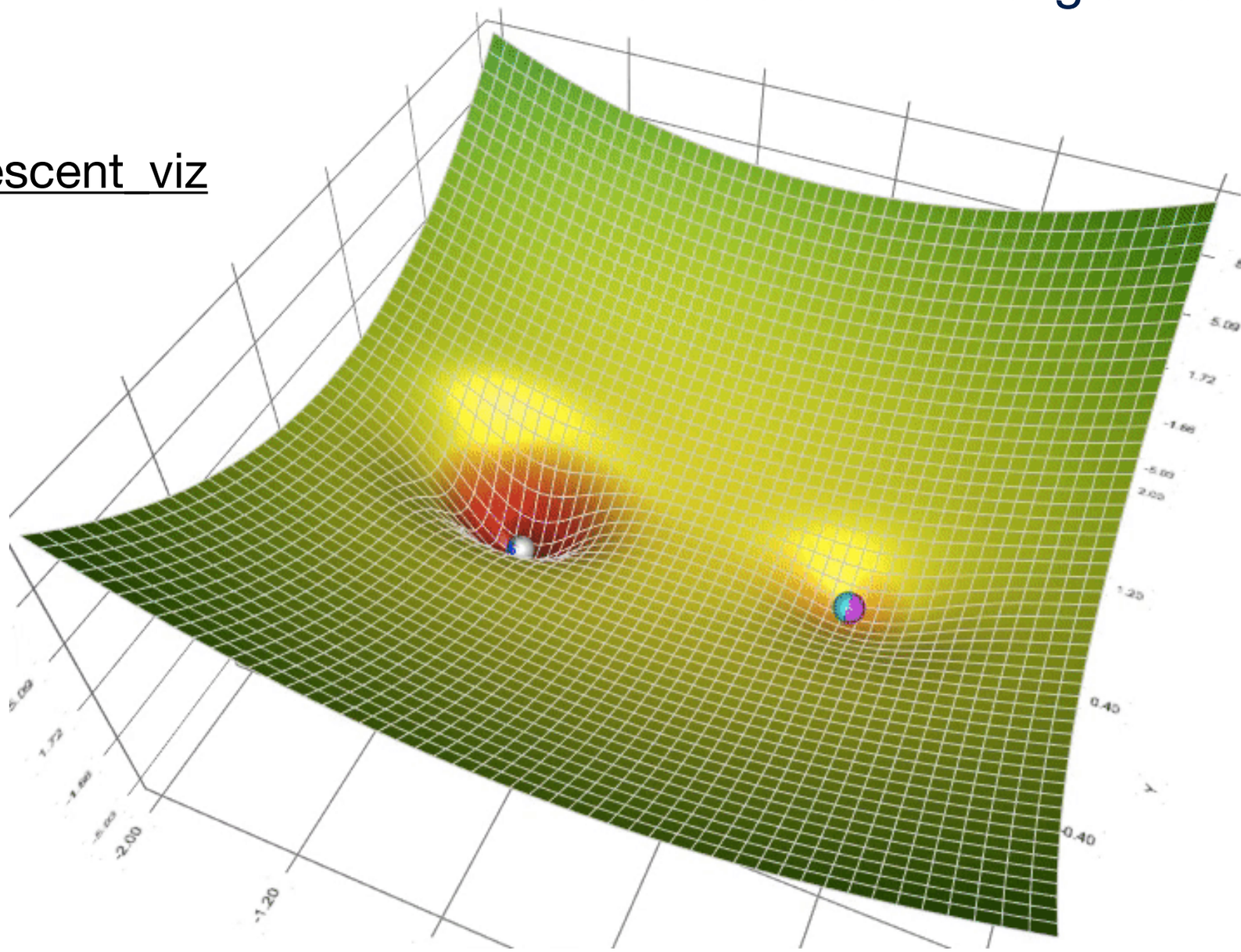
Note on the loss function

- Model parameter estimation is performed through maximum likelihood technique:
  - Cash statistics is used for counts data with a known background
    - The 3D analysis with a model background in the IRF
  - Wstat statistics is used for counts data with a measured background
    - Typically the 1D analysis where the bkg is estimated from the OFF regions
    - Or a 3D analysis with ON/OFF estimation

What is a fit ?

- Need a loss function + minimizer :
  - Gradient Descent (e.g. Scipy minimize, iMinuit, sherpa fit, etc)
  - Markov chain Monte Carlo
  - Nested Sampling methods

Never take a best-fit for granted



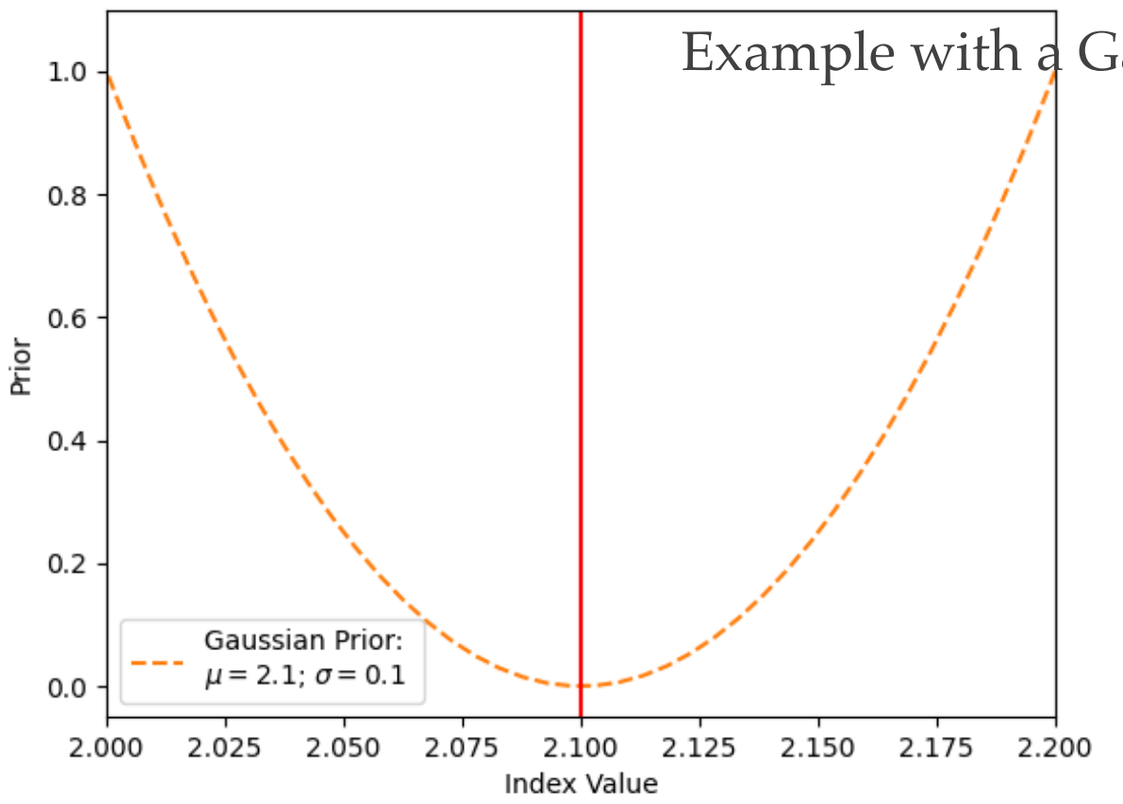
github.com/lilipads/gradient_descent_viz

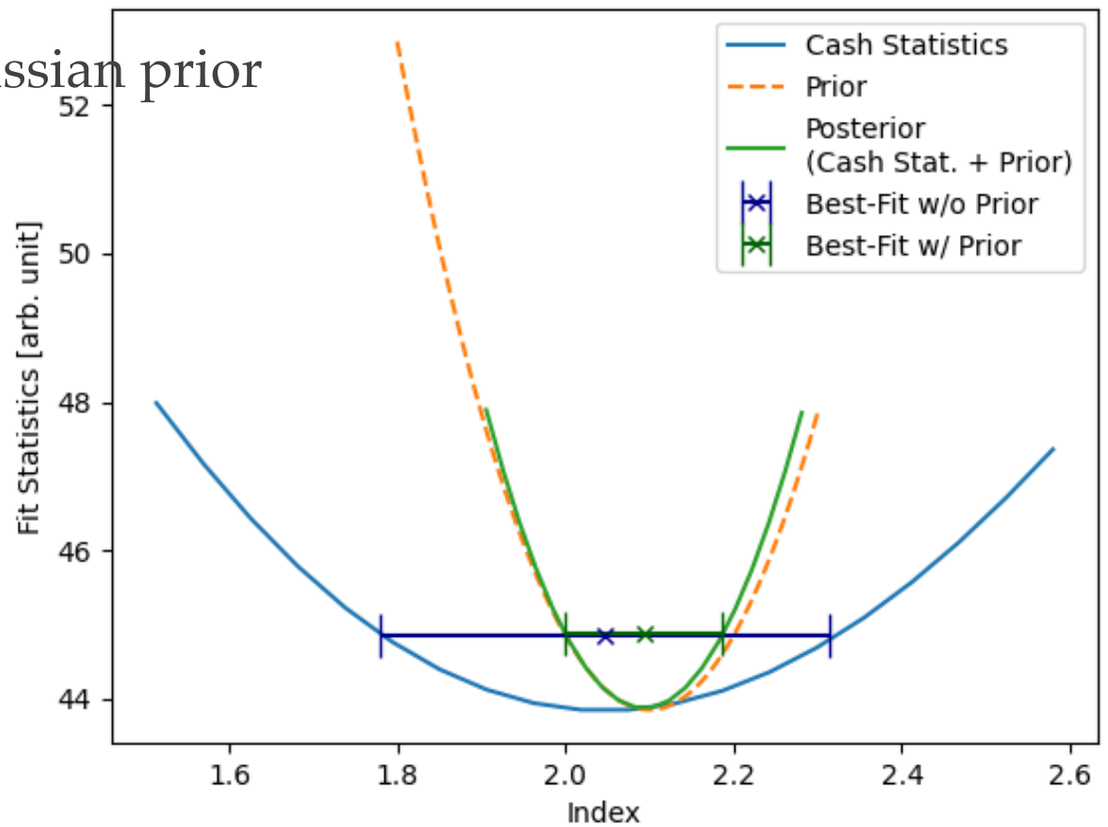
Adding priors on Parameters

- Prior: A probability density function of the model parameters

- Includes information about the parameters
- Added to the fit statistic to get the Posterior
- Possible to add Custom priors

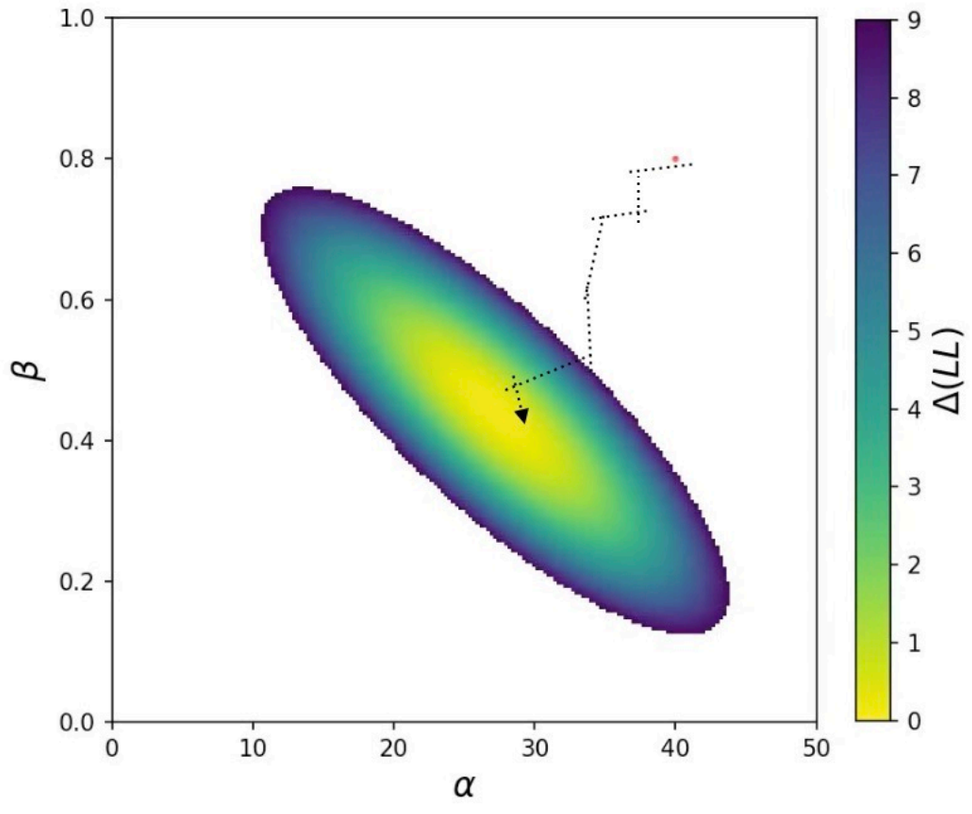
Example with a Gaussian prior





Traditional fitting method

- **Gradient descent based method**
  - **Levenberg Marquardt**
  - **Migrad in Minuit for example**
  - **Gradient is estimated numerically at each step**



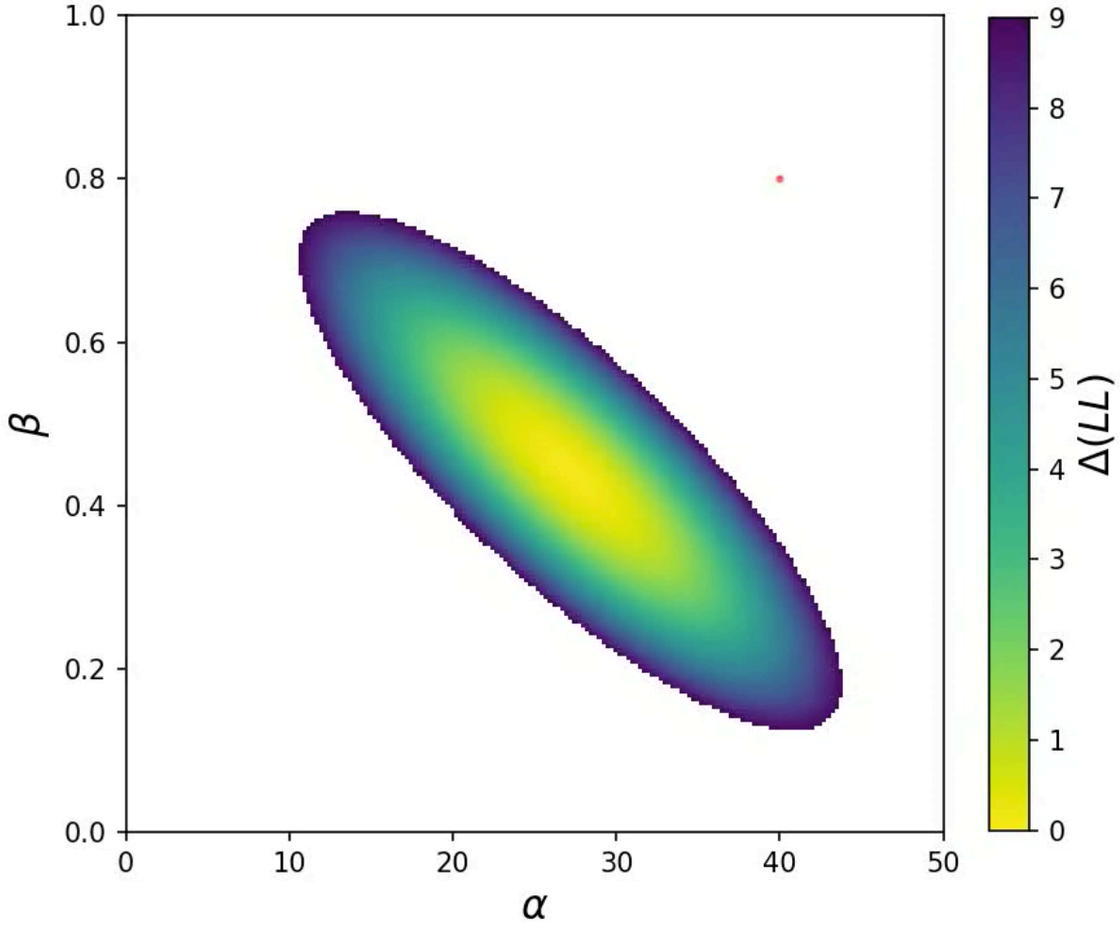
Once at best-fit stops
No information about local likelihood
Sometimes fit fails :

Migrad

FCN = -2.676e+07	Nfcn = 378
EDM = 0.00356 (Goal: 2e-06)	time = 2.3 sec
INVALID Minimum	No Parameters at limit
ABOVE EDM threshold (goal x 10)	Below call limit
Covariance / Hesse ok	APPROXIMATE / NOT pos. def. / FORCED

Markov Chain Monte Carlo (MCMC)

- **What are they:**
  - **Monte Carlo: samples are used to approximate the probability distribution**
  - **Markov Chain: semi-random walk in potential**
  - **Walkers explore the local likelihood**

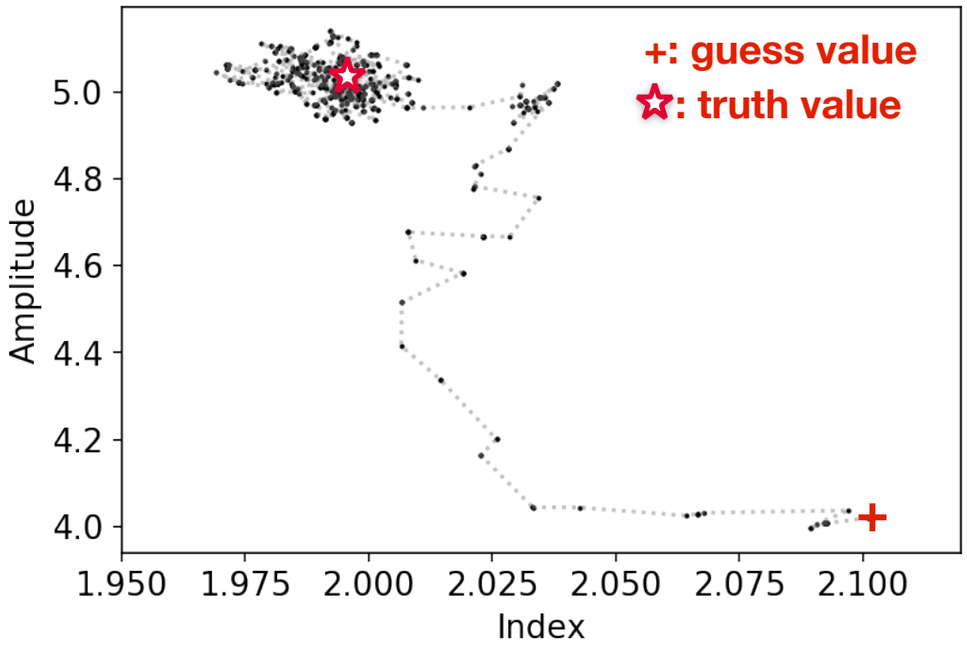


Random walk directed by potential (likelihood) spend most of their time in interesting region

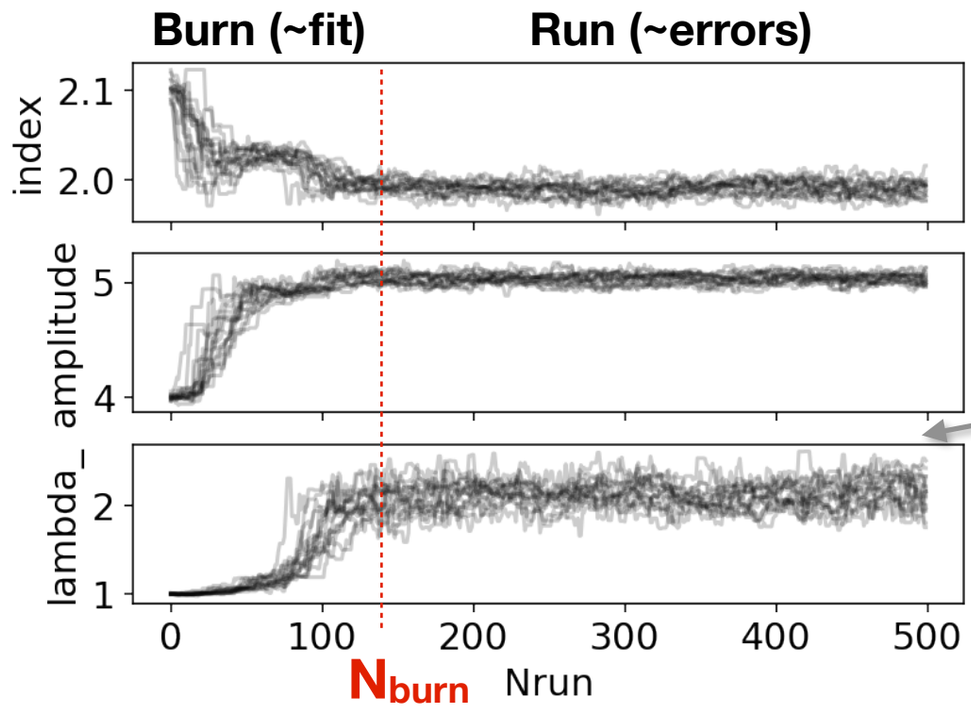
Technicaly not a fit (no convergence) it’s a phase space parameter exploration

MCMC issues : inside out approach

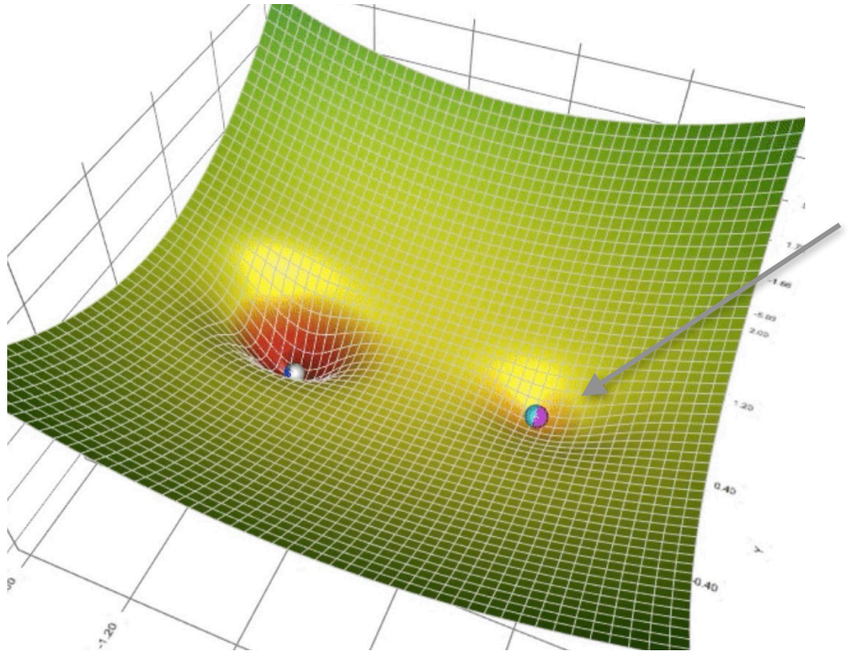
1 walker evolving for 500 steps



10 walkers evolving for 500 steps



each element of the chain are samples of the target posterior distribution.



But what if all your walkers end up here ?

MCMC limitations:
- When do you stop a chain ?
- Has it converged ?
- How to choose init point ?

What is a Bayesian analysis (user point of view)

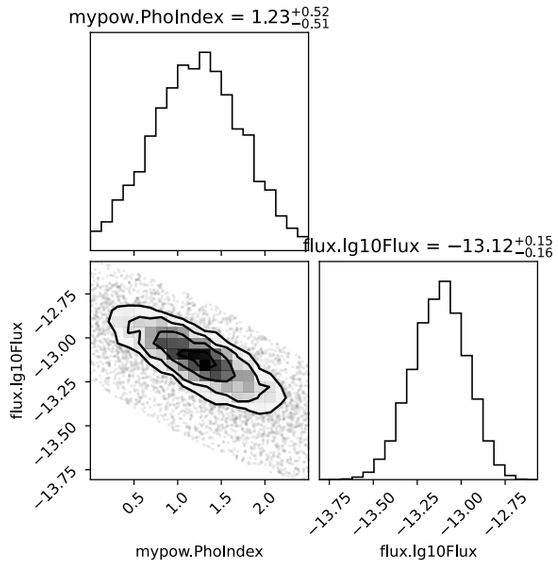
Priors

- Prior information on parameters
- E.g. : Norm > 0
- P1 < Param < P2
- Norm_bkg Gaussian(1, sigma)

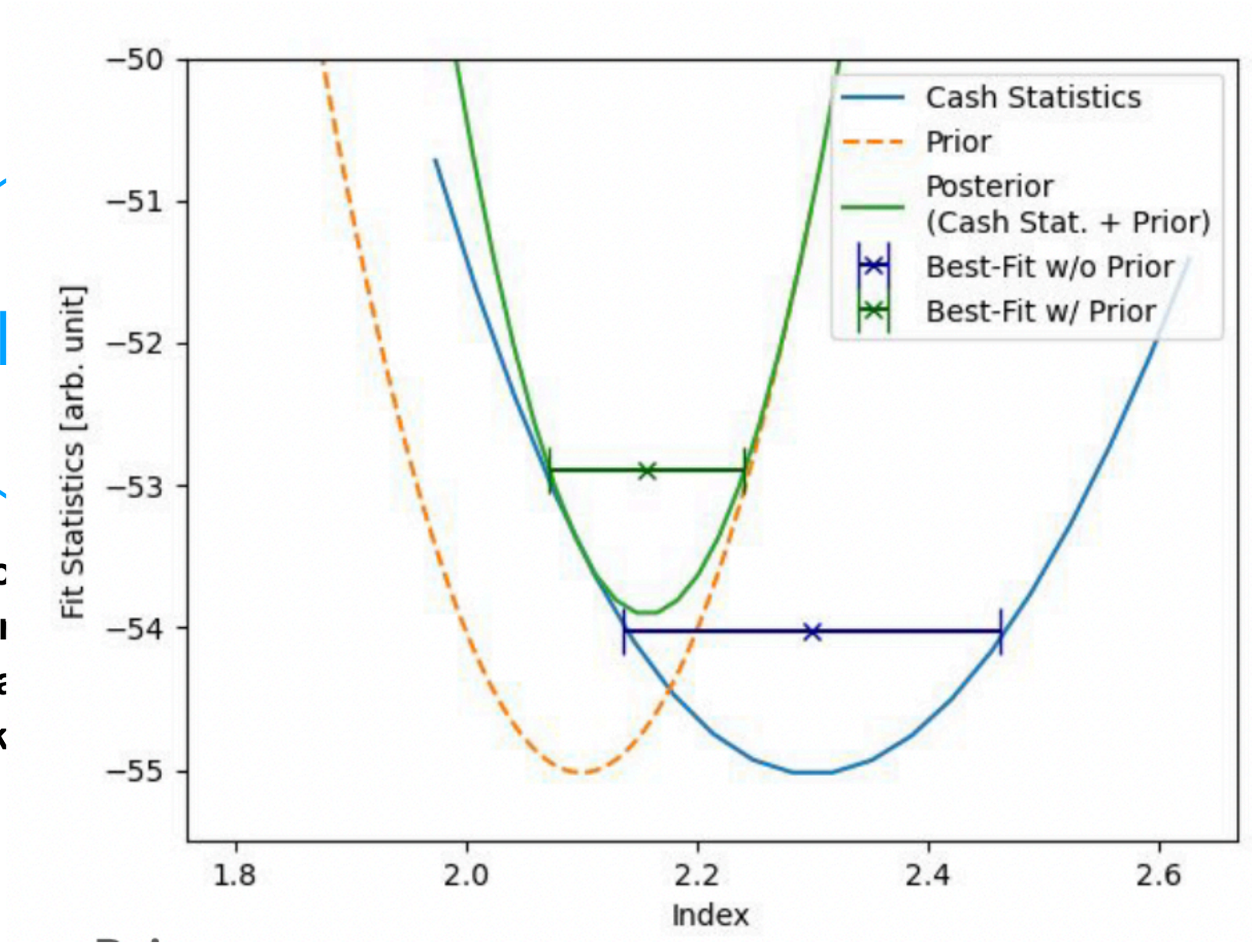
Data

- Modified Likelihood
- Likelihood = data_term + priors
-

Posterio rs



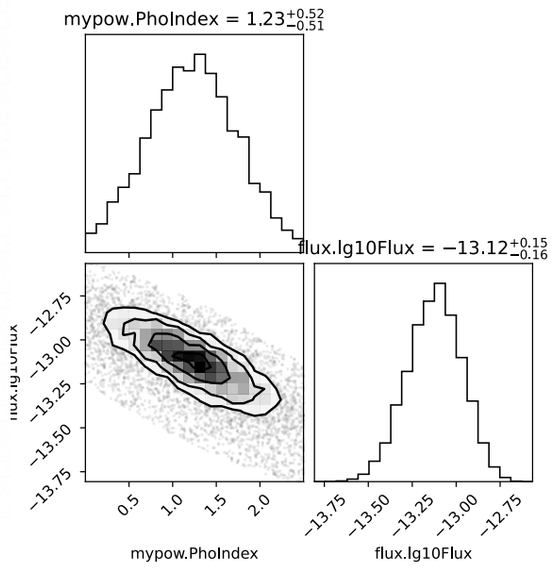
What is a Bayesian analysis (user point of view)



I

Posterio
rs

- **Prior infc**
- **E.g. : Nor**
- **P1 < Para**
- **Norm_bk**



Maximum a posteriori (MAP)

Nested Sampling methods

- **Nested Sampling (Skilling, 2004) is a Monte Carlo algorithm for estimating an integral over a model parameter space θ**
  - **Integral :**
    - **$\int$ Like(Data|θ) * Prior(θ) dθ = Z = Bayesian evidence**
  - **Z can be used to compare models even if <u>not</u> nested (morphology : mwl template vs disk)**
  - **This integral is also what will provide the normalized posterior distributions**
- **Main idea is integration is done by switching frame from many θ to a volume variable**

Unlike MCMC methods, which attempt to estimate the posterior $\mathcal{P}(\mathbf{\Theta})$ directly, Nested Sampling instead focuses on estimating the evidence

$$\mathcal{Z} \equiv \int_{\Omega_\Theta} \mathcal{P}(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta} = \int_{\Omega_\Theta} \mathcal{L}(\mathbf{\Theta}) \pi(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta} \quad (6)$$

As this integral is over the entire multi-dimensional domain of $\mathbf{\Theta}$, it is traditionally very challenging to estimate.

Nested Sampling approaches this problem by refactoring this integral as one taken over prior volume X of the enclosed parameter space

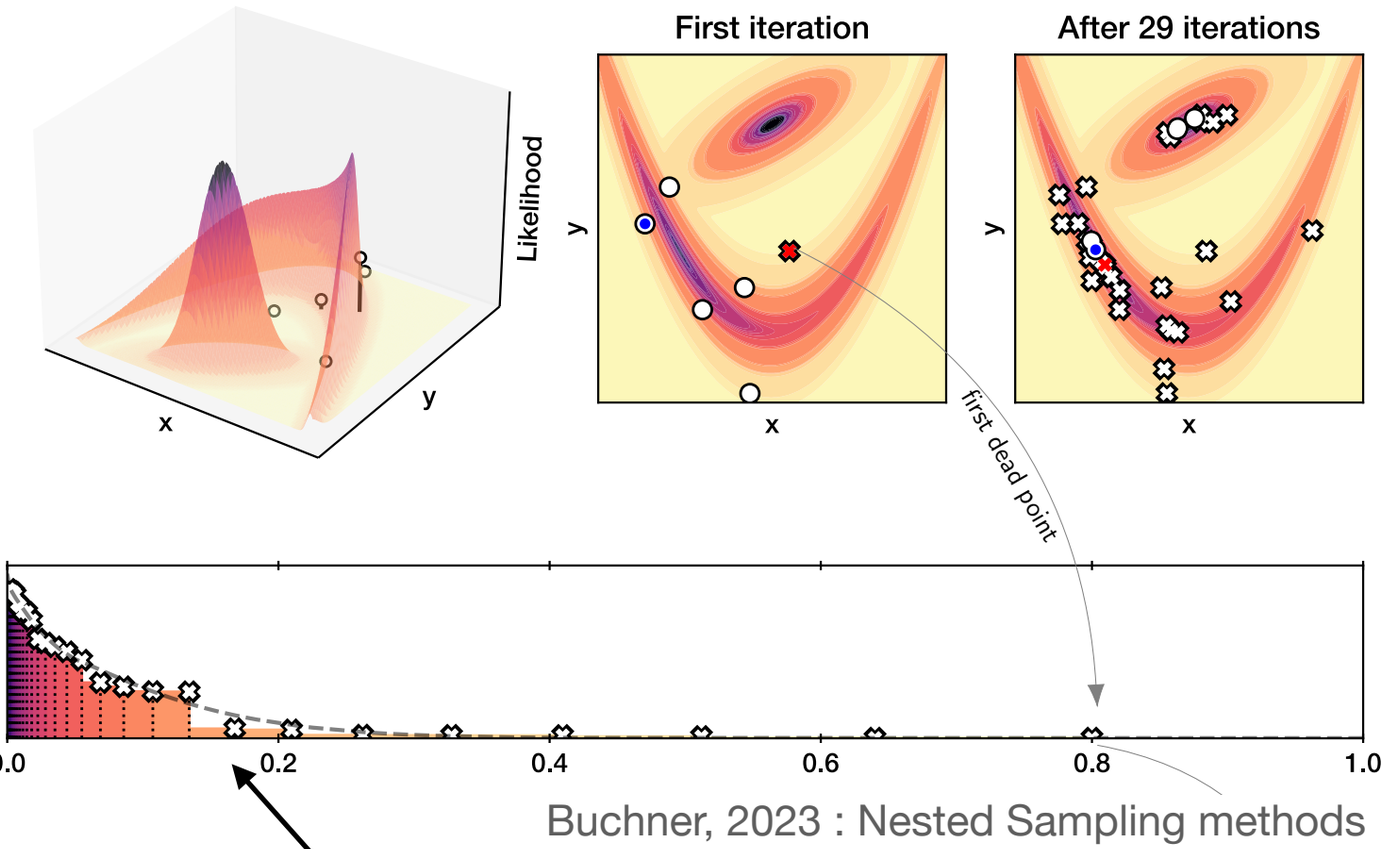
$$\mathcal{Z} = \int_{\Omega_\Theta} \mathcal{L}(\mathbf{\Theta}) \pi(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta} = \int_0^1 \mathcal{L}(X) \mathrm{d}X \quad (7)$$

Speagle, 2020

- **Analogy to spherical coordinates:**

$$\int \mathcal{P}(x, y, z) \mathrm{d}x \mathrm{d}y \mathrm{d}z = \int \mathcal{P}(V(r)) \mathrm{d}V(r) = \int \mathcal{P}(r) 4\pi r^2 \mathrm{d}r$$

Nested Sampling methods



Buchner, 2023 : Nested Sampling methods

Define priors on params

Transform param space to volume unity cube X

Draw random uniform N_{live} (~400-1000)

Iterate until stop criteria:

- **Remove worst LogLike point**
- **Draw new point with a better LL**
  - **Likelihood-restricted prior sampling**
  - **(The tricky part)**
- **Increment Z= Z + L*ΔX**
- **Stop criteria : ΔZ/Z<tolerance**

Volume X

$$Z = \sum L \Delta X$$

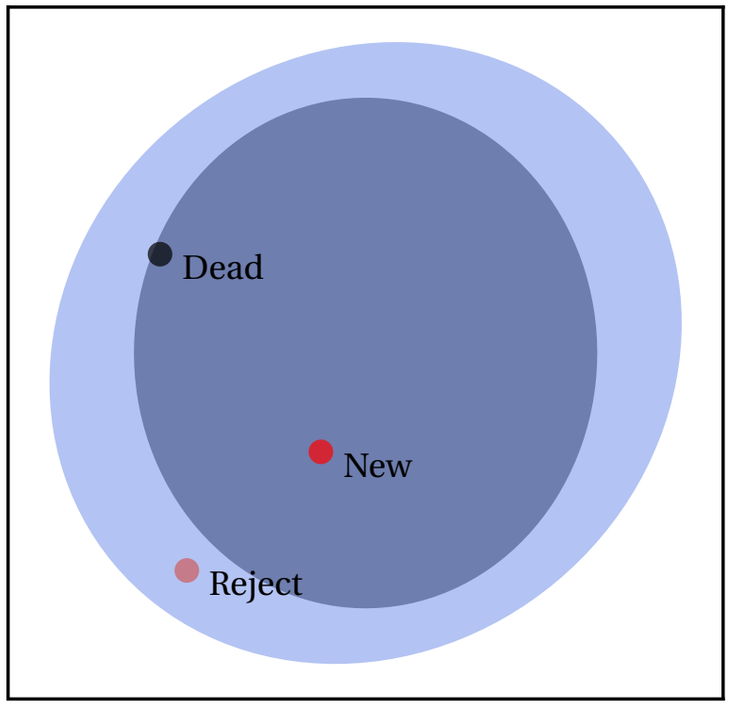
Integration via trapezoid rule

Z = Bayesian evidence = $\int$ Like(Data|θ) * Prior(θ) dθ

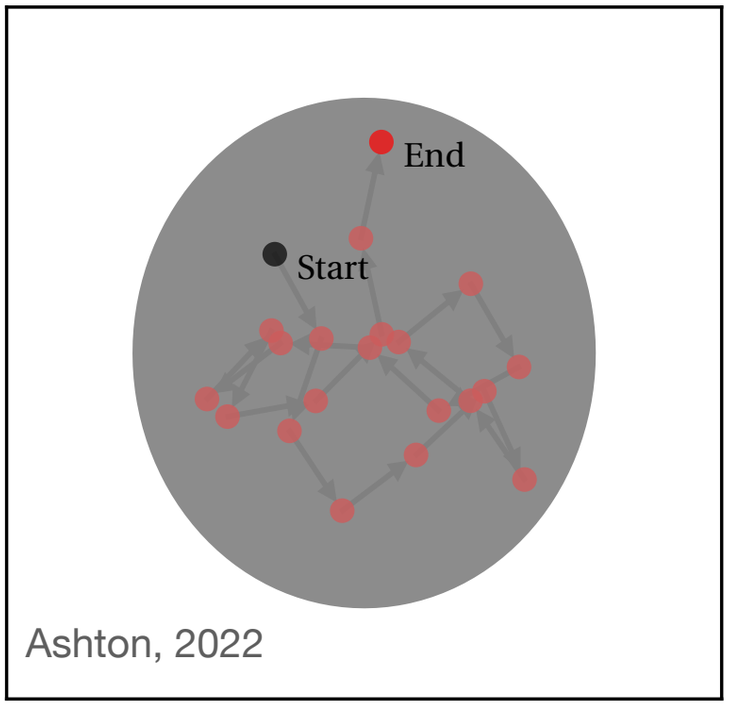
Sampling from a restricted prior ($Like_{new} > Like_{removed}$)

- **The tricky part : how to sample points with a better LogLike**
  - **This means drawing point inside the iso LogLike contours that we don't kno**

- **Take advantage from the fact that if NLive is large enough. Surviving points already provide trace the landscape**
- **So drawing an encapsulating ellipsoid and try to sample from this ellipsoid. Reject if needed**
- **Or if landscape too complex or high dimension use a step sampler**

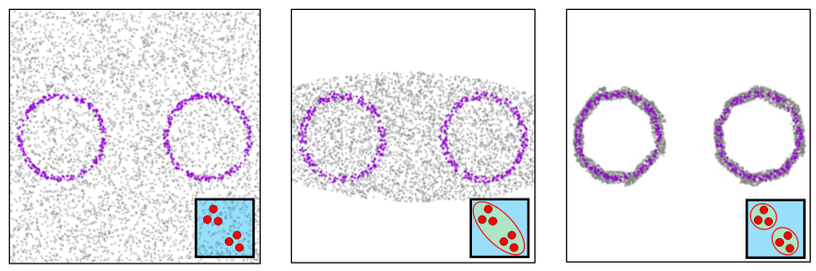


Ellipsoid sampling

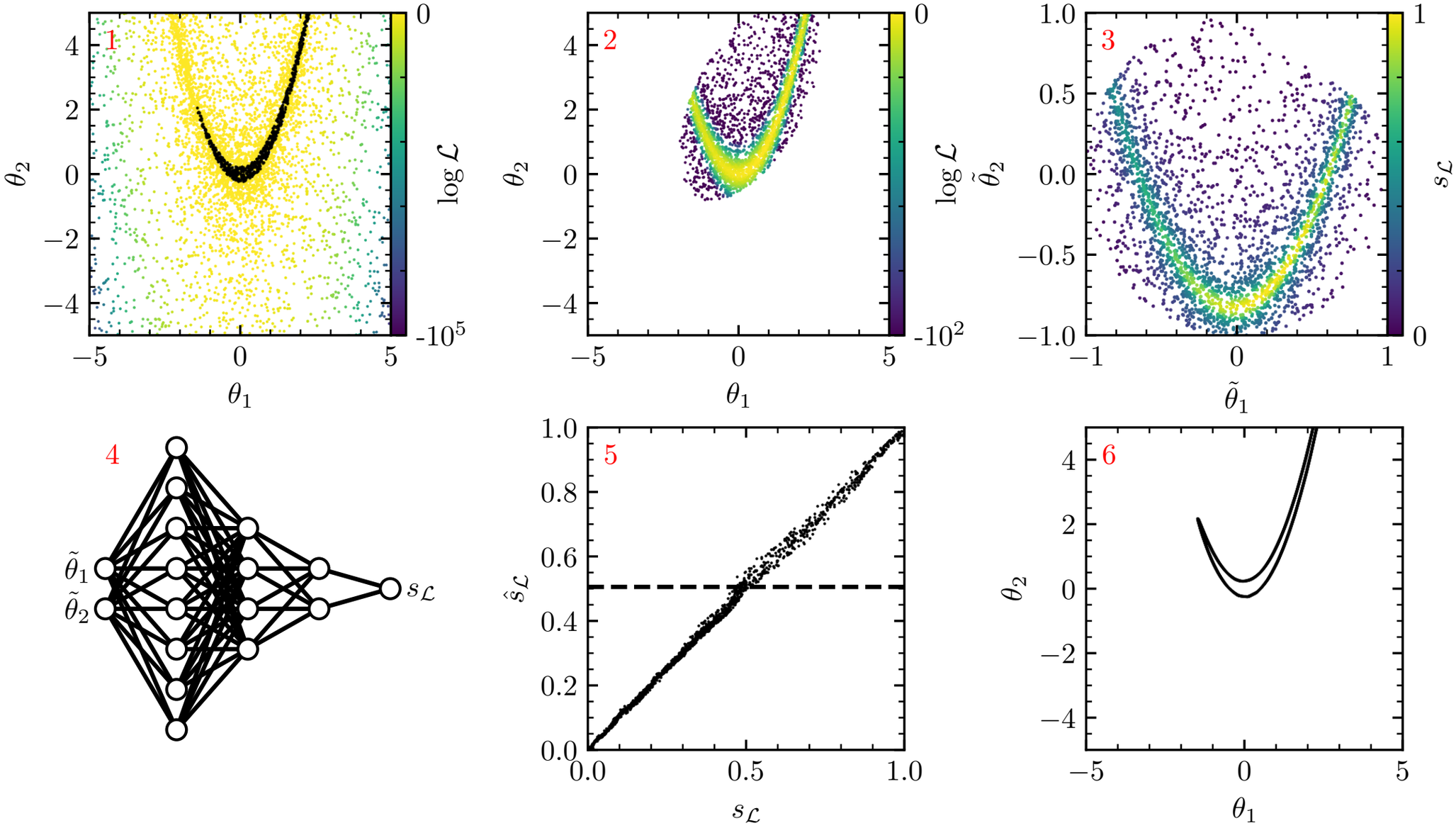


Random walk

Bounding distributions



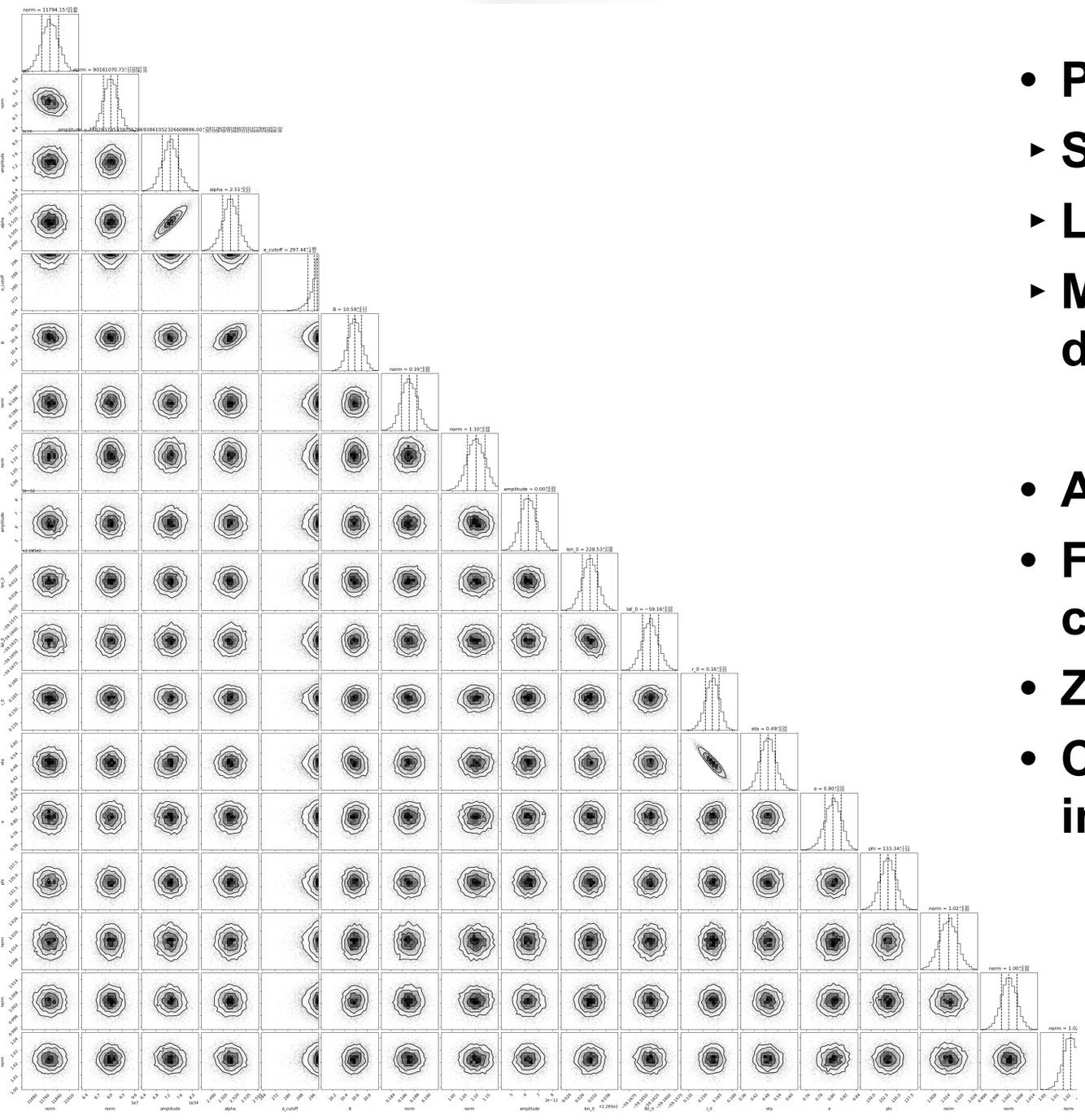
Points proposals via likelihood estimators with ML



Lange J., 2023

Figure 3. Diagram depicting how new proposal volumes during the exploration phase are constructed. For this example, we chose the two-dimensional Rosenbrock likelihood. The steps are as follows. (1) The set of N_{live} points with the highest likelihood, the so-called live set, is identified. (2) One or multiple non-overlapping bounding ellipsoids are drawn around the live set. (3) The coordinates $\boldsymbol{\Theta}$ of points in the ellipsoid are transformed into the ellipsoid coordinates $\tilde{\boldsymbol{\Theta}}$ using a Cholesky decomposition. Similarly, likelihood values are converted into likelihood scores $0 \leqslant s_\mathcal{L} \leqslant 1$. (4) The transformed coordinates and likelihood scores are used to train a neural network. (5) A cut $\hat{s}_{\mathcal{L},\text{min}}$ in the predicted likelihood score is determined that corresponds to the likelihood score of the live set. (6) The new proposal volume is defined as that part of the bounding ellipsoid where the predicted likelihood score is above $\hat{s}_{\mathcal{L},\text{min}}$.

Outputs



- **Posterior distributions**
  - **Stop criteria : ΔZ/Z<tolerance**
  - **Large advantage vs MCMC**
  - **MCMC has no good stopping criteria and depends from init position**

- **Another advantage :**
- **Full integration providing Z (MCMC do not compute Z)**
- **Z is the Bayesian evidence :**
- **Can compare Non nested models & included handling of free dof**

Table 1. Results of the fit of the LAT data between 1 GeV and 1 TeV using different spatial models. The second column reports the TS values obtained for each spatial model, while column 3 indicates the number of degrees of freedom k adjusted in the model. The delta Akaike criterion is reported in the fourth column. See Sect. 2.2 for more details.

Spatial model	TS	k	ΔAIC
X-ray template	28.6	2	-24.6
Radio template	28.8	2	-24.8
Infrared template	29.7	2	-25.7
Best point source (PS)	32.1	4	-24.1

deltaZ

Tips & tricks for fitting

- **Choose a reasonable starting point**
- **! Always plot your N_{pred} counts map to investigate issues !**
- **Set some boundaries (min, max)**
  - **Goal is to avoid unphysical values:**
    - **Negative fluxes, positions outside box, too large size**
    - **But be careful for upper-limit then**
- **Start with a simpler model and add complexity if needed:**
  - **Start with frozen spatial positions**
  - **PL first then ExpCutOff PL, source extension, etc**
  - **Mask regions that are too complex**
- **Freeze some parameters that cannot be constrained**

- **If more than 5 free params, try Nested sampling**
- **Drop MCMC, Nested sampling is more reliable**

Pray for the minimizer god