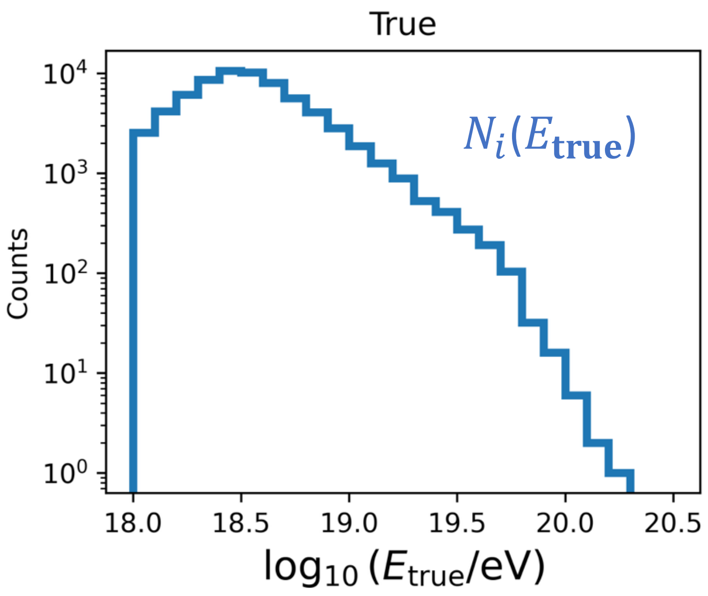
Forward-folding

Kozo Fujisue
Academia Sinica Institute of Physics

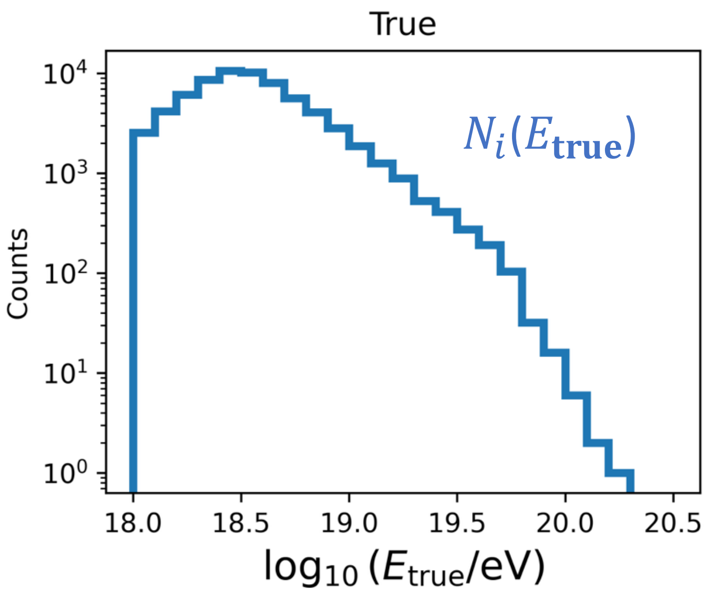
True values & Observed values



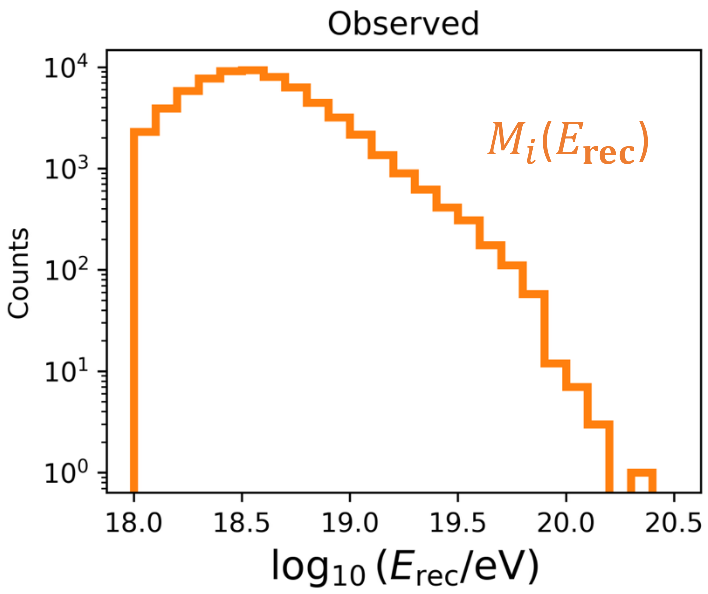
What we want to know
$N_i(E_{\mathbf{true}})$

(Through this lecture and hands-on, we treat binned-data, but essentially same for unbinned-data)

True values & Observed values



Detector response

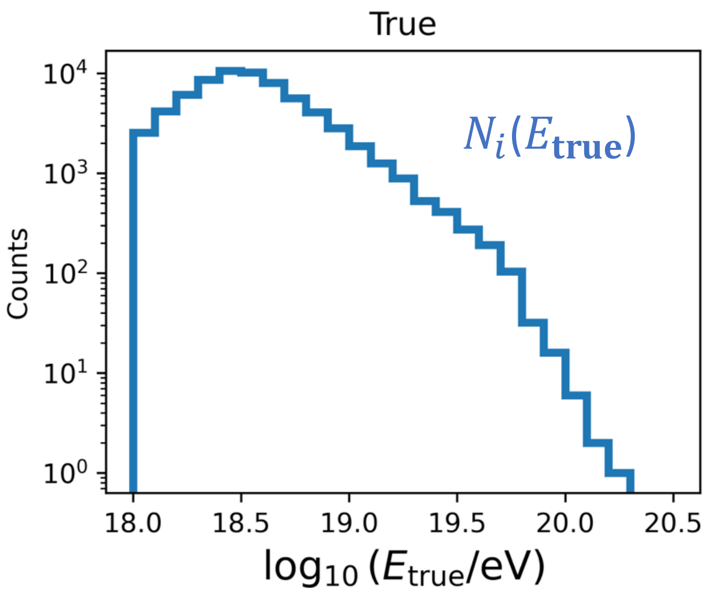


What we want to know
$N_i(E_{\mathbf{true}})$

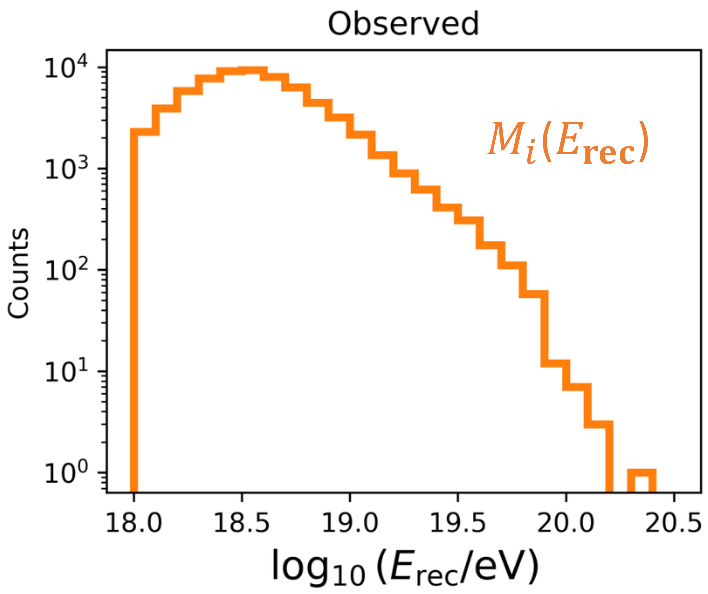
What we observe
$M_i(E_{\mathbf{rec}})$

(Through this lecture and hands-on, we treat binned-data, but essentially same for unbinned-data)

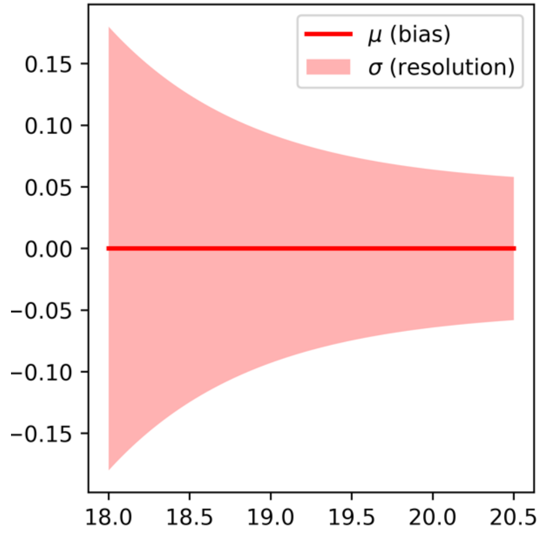
True values & Observed values



Detector response

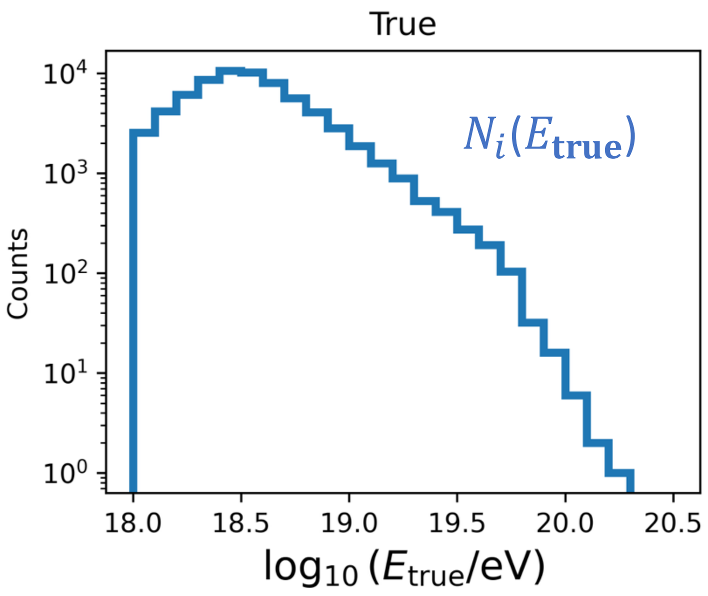


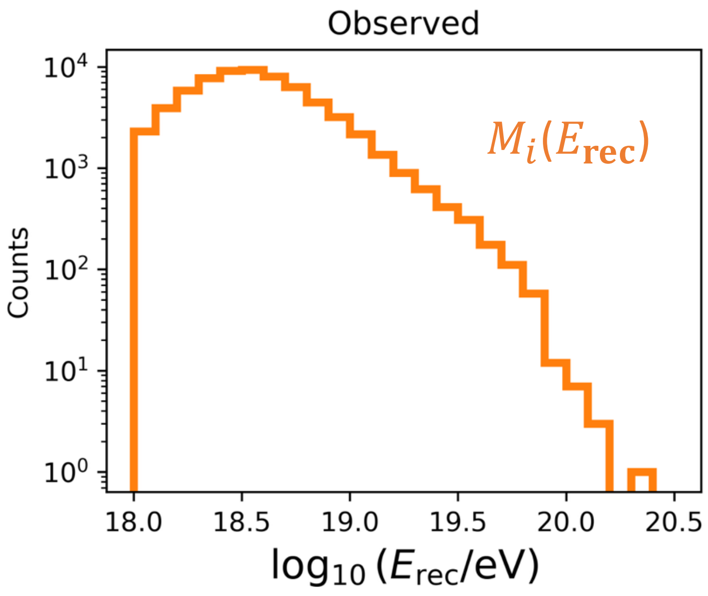
What we want to know
$N_i(E_{\mathbf{true}})$



What we observe
$M_i(E_{\mathbf{rec}})$

True values & Observed values





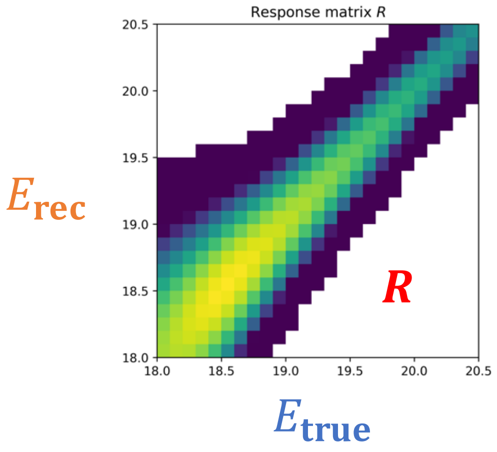
Detector response

$\boldsymbol{R_{ij}}(E_{\mathrm{true}} \to E_{\mathrm{rec}})$

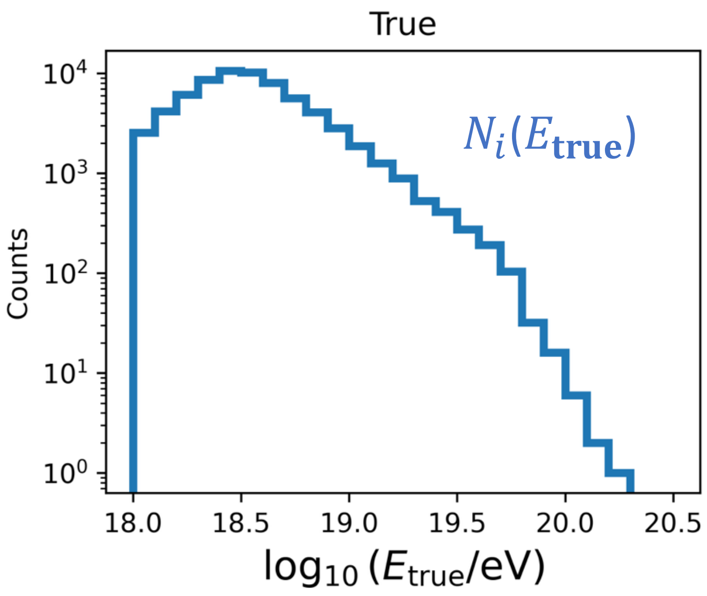
What we want to know
$N_i(E_{\mathbf{true}})$

What we observe
$M_i(E_{\mathbf{rec}})$

$$N_i(E_{\mathbf{true}}) \times \boldsymbol{R_{ij}}(E_{\mathrm{true}} \to E_{\mathrm{rec}}) = M_i(E_{\mathbf{rec}})$$

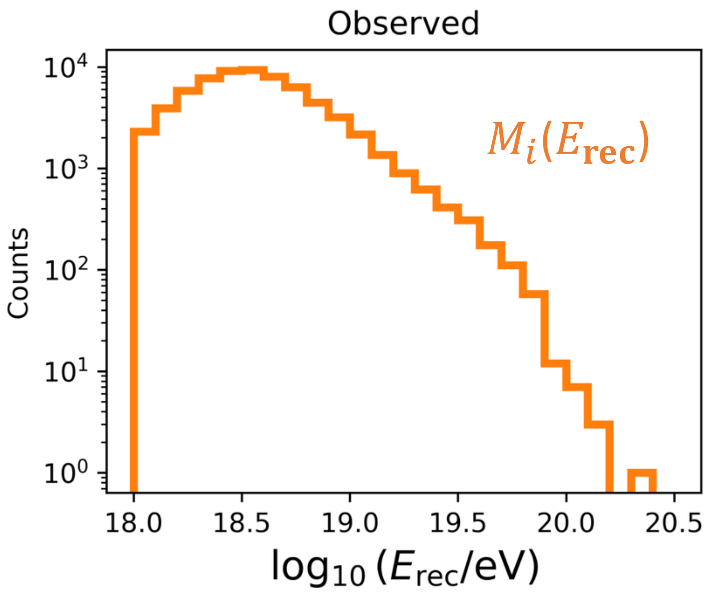


Unfolding: solving an inverse problem



Unfolding

$\boldsymbol{R^{-1}}_{ij}(E_{\mathrm{true}} \leftarrow E_{\mathrm{rec}})$

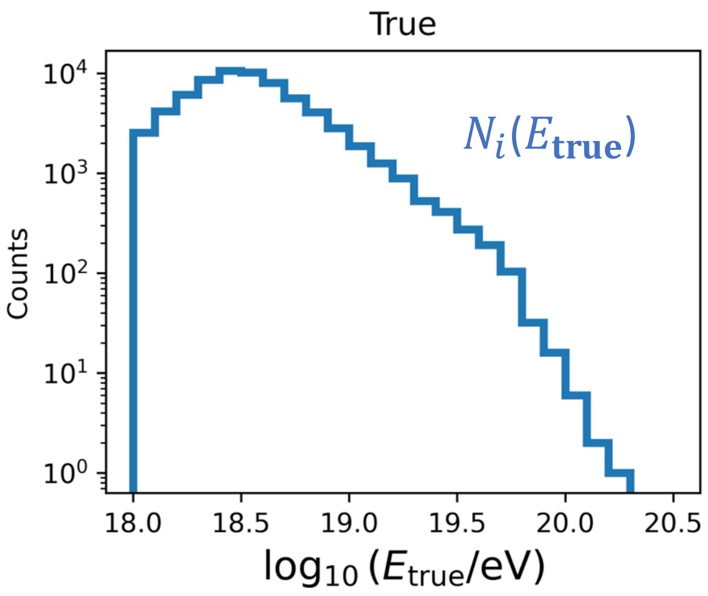


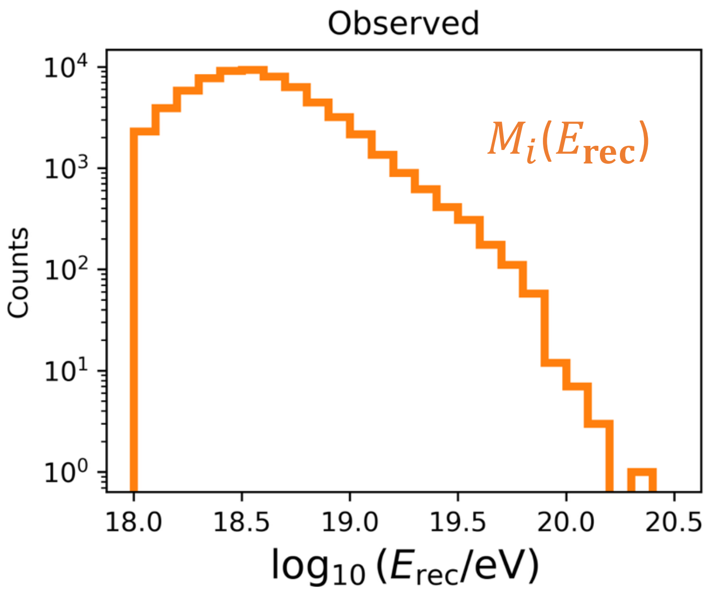
What we want to know

What we observe

$$N_i(E_{\mathbf{true}}) = M_i(E_{\mathbf{rec}}) \times \boldsymbol{R^{-1}}_{ij}(E_{\mathrm{true}} \rightarrow E_{\mathrm{rec}})$$

Unfolding: solving an inverse problem





Unfolding

$\boldsymbol{R^{-1}}_{ij}(E_{\mathrm{true}} \leftarrow E_{\mathrm{rec}})$

What we want to know

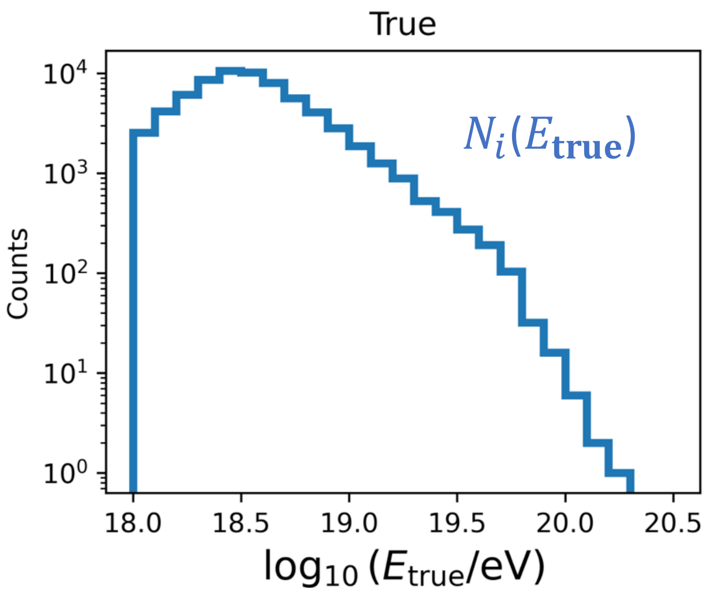
What we observe

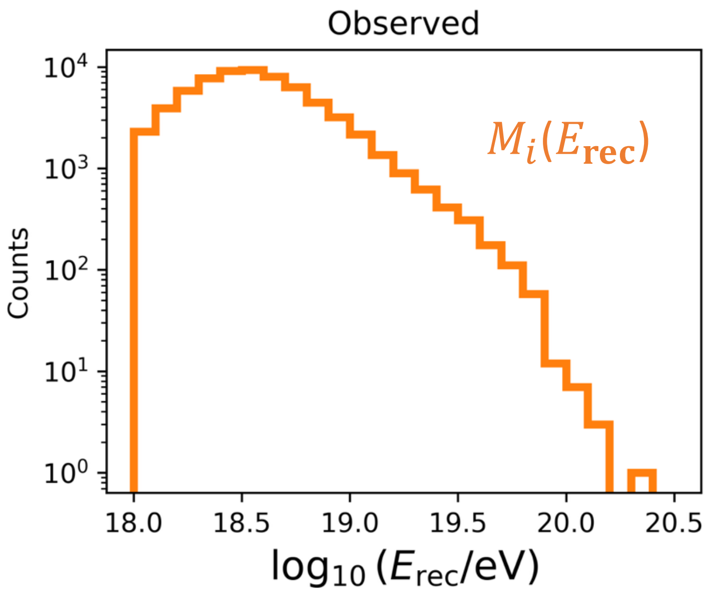
$$N_i(E_{\mathbf{true}}) = M_i(E_{\mathbf{rec}}) \times \boldsymbol{R^{-1}}_{ij}(E_{\mathrm{true}} \rightarrow E_{\mathrm{rec}})$$

Unfolding: solving an <u>inverse problem</u> (逆問題)

It is not easy

Unfolding: solving an inverse problem





Unfolding

$\boldsymbol{R^{-1}}_{\boldsymbol{ij}}(E_{\mathrm{true}} \leftarrow E_{\mathrm{rec}})$

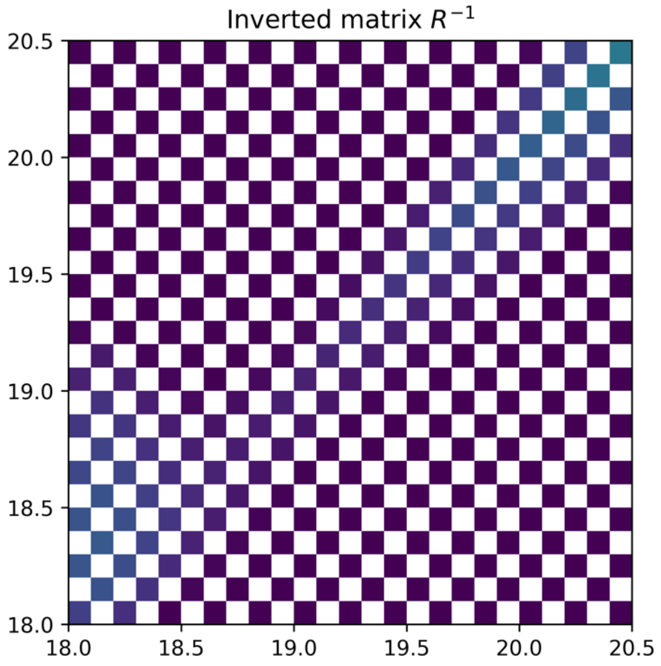
What we want to know

What we observe

<u>Difficulties of Unfolding (1): amplification of noises</u>

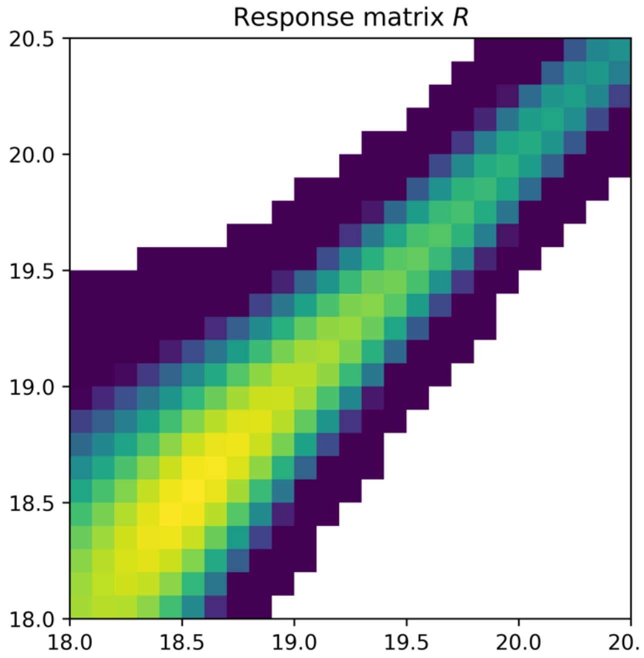
- The off-diagonal components amplifies noises (introduces another uncertainties).
  - More problematic for low-statistics measurements (like us).
- To stabilize the unfolded result, regularization is required, and it is subjectively chosen:
  - Tikhonov / SVD / Bayesian /Machine learning, …

Unfolding: solving an inverse problem



R^{-1}

inverting
(no regularization)

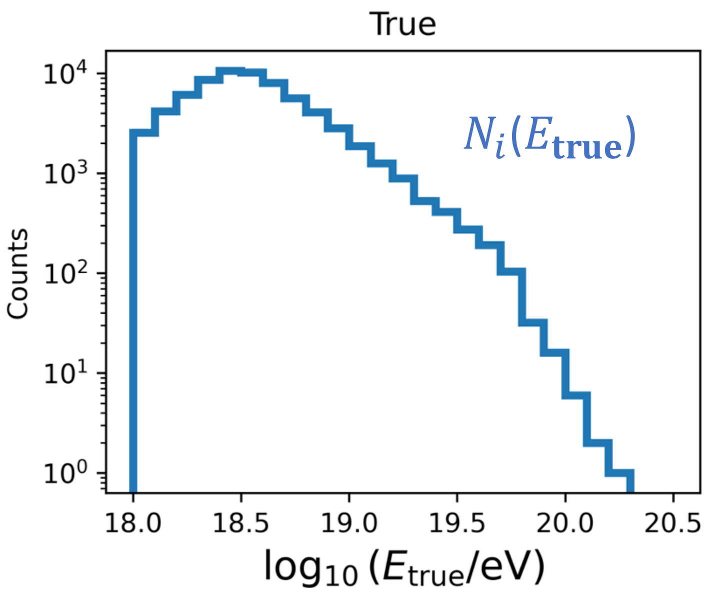


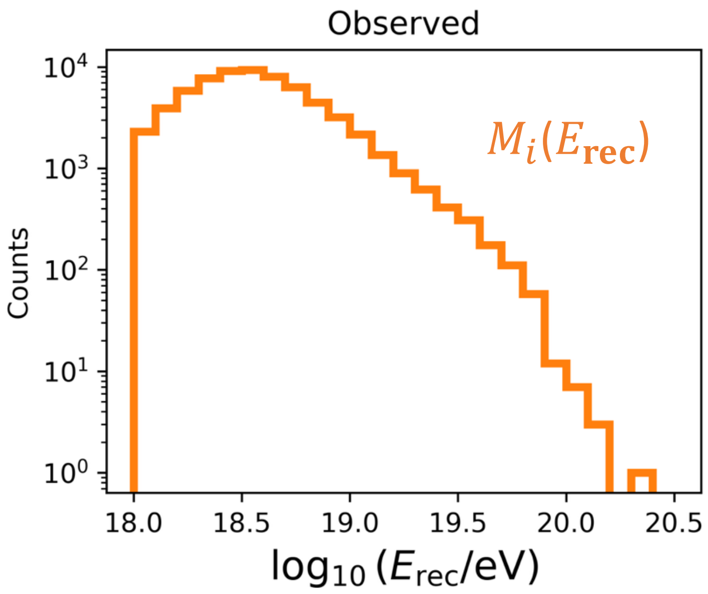
R

<u>Difficulties of Unfolding (1): amplification of noises</u>

- The off-diagonal components amplifies noises (introduces another uncertainties).
  - More problematic for low-statistics measurements (like us).
- To stabilize the unfolded result, regularization is required, and it is subjectively chosen:
  - Tikhonov / SVD / Bayesian /Machine learning, …

Unfolding: solving an inverse problem





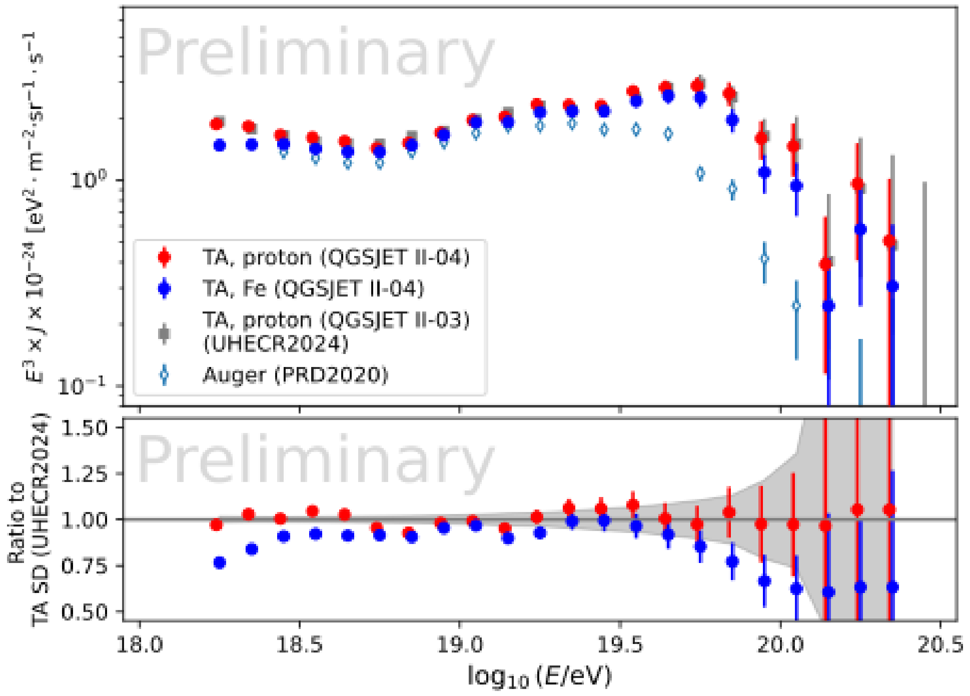
Unfolding

$\boldsymbol{R^{-1}}_{\boldsymbol{ij}}(E_{\rm true} \leftarrow E_{\rm rec})$

<u>Difficulties of Unfolding (2): difficult to estimate systematic uncertainties</u>

- Typical method:
  1) Evaluating various systematic uncertainties **separately**
     - Unfolding with a parameter shifted by +/- 1σ uncertainty
  2) Summing them quadratically / linearly (assuming correlations)
     - Considering correlations is usually difficult, requiring approximations
- Often evaluating extreme scenarios and show conservative systematic uncertainties

Unfolding: solving an inverse problem



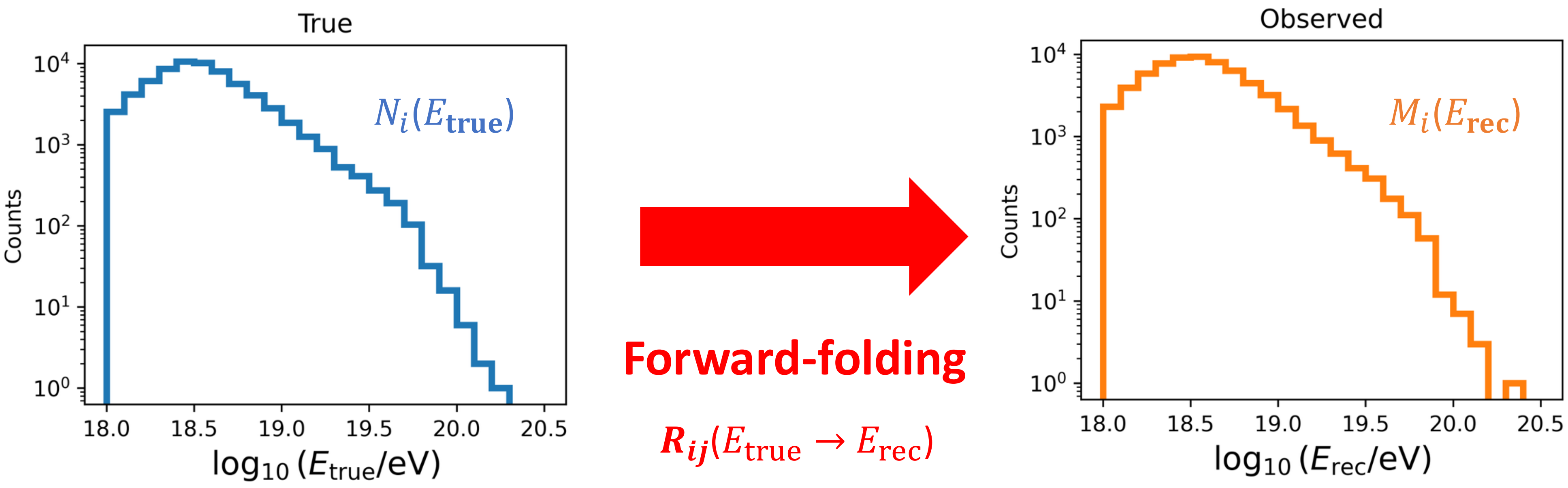
(e.g.) K. Fujisue for TA collab. ICRC2025

- Unfolding assuming different mass compositions
  - Pure proton
  - Pure iron

Difficulties of Unfolding (2): difficult to estimate systematic uncertainties

- Typical method:
  1) Evaluating various systematic uncertainties **separately**
     - Unfolding with a parameter shifted by +/- 1σ uncertainty
  2) Summing them quadratically / linearly (assuming correlations)
     - Considering correlations is usually difficult, requiring approximations
- Often evaluating extreme scenarios and show conservative systematic uncertainties

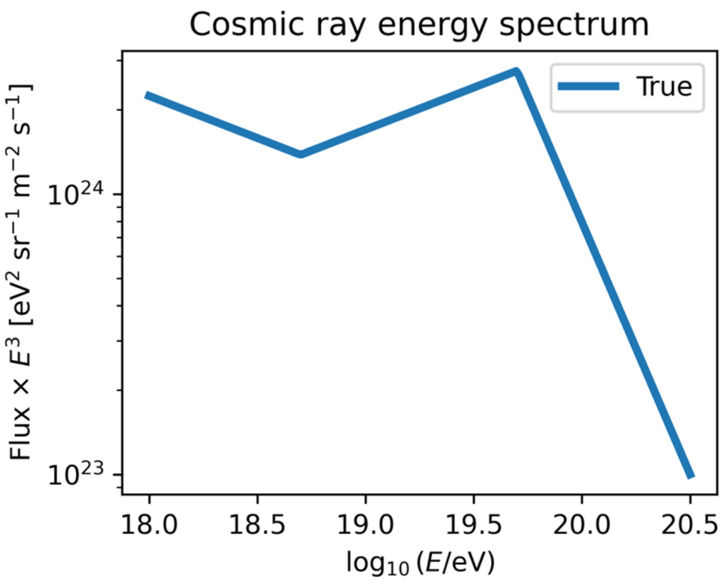
Forward-folding: free from the inverse problem



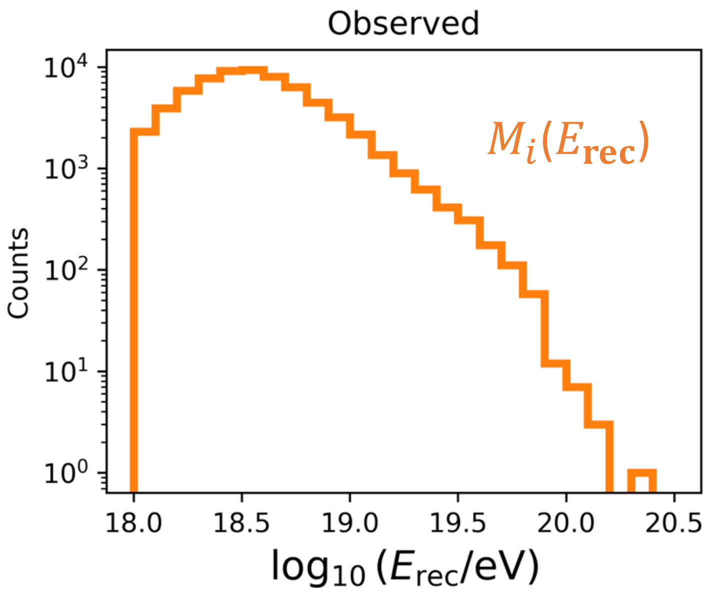
Forward-folding:

(e.g.) Parameterize each $N_i(E_{\mathbf{true}})$ and find the best set of $N_i(E_{\mathbf{true}})$ that reproduces the observed distribution M_i (= $N_i \times \boldsymbol{R_{ij}}$) best

Forward-folding: free from the inverse problem



Forward-folding



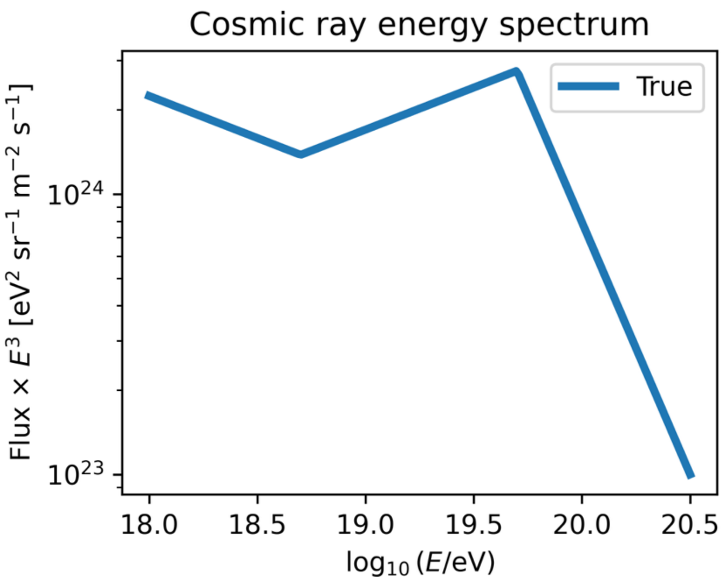
$\boldsymbol{R}$

(effective exposure) × (response) × dE

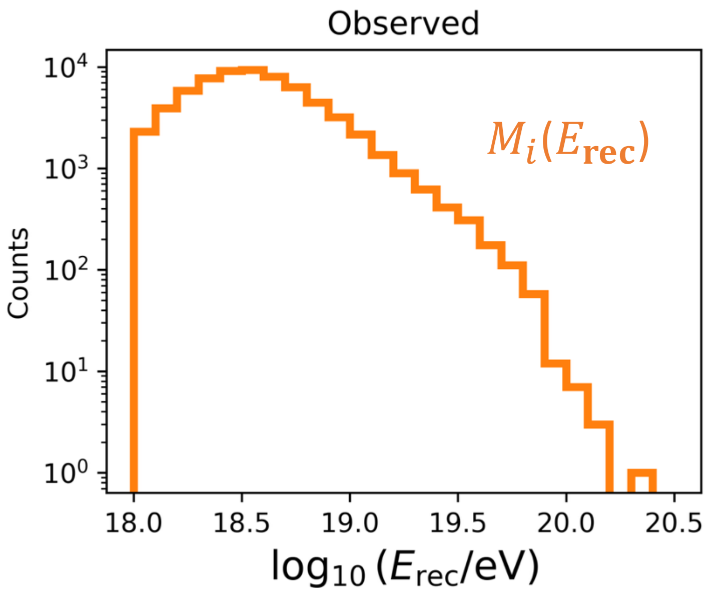
Forward-folding:

(e.g.) Parameterize energy spectrum and find the best set of the parameters that reproduces the observed distribution M_i best

Forward-folding: free from the inverse problem



Forward-folding



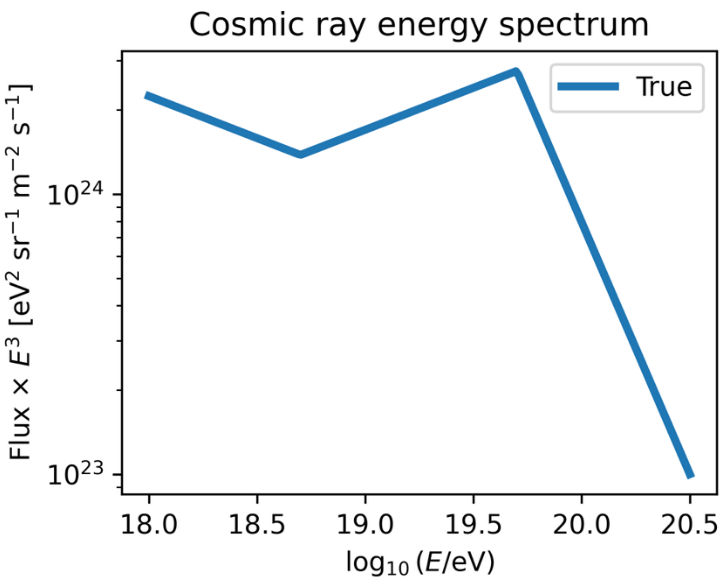
$\boldsymbol{R}$

(effective exposure) × (response) × dE

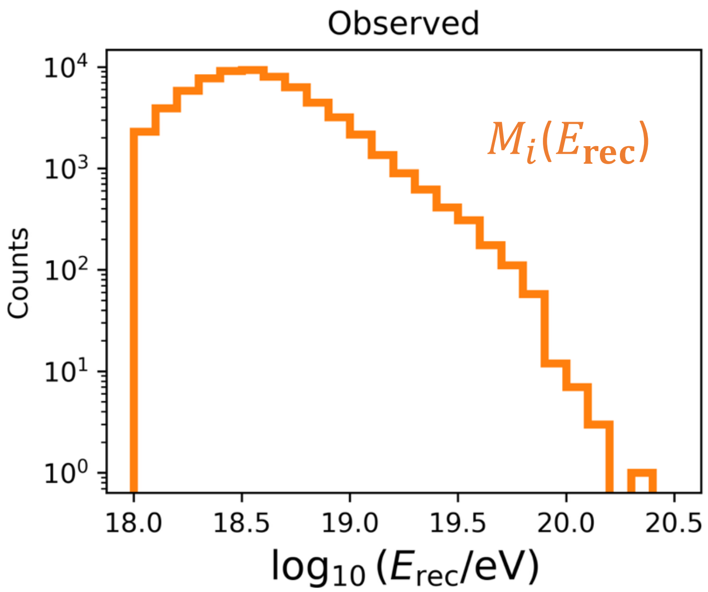
Difficulties of **Forward-folding** :

- Computationally expensive
  - Requires likelihood evaluation for many set of parameters

Forward-folding: free from the inverse problem



Forward-folding



$\boldsymbol{R}$

(effective exposure) × (response) × dE

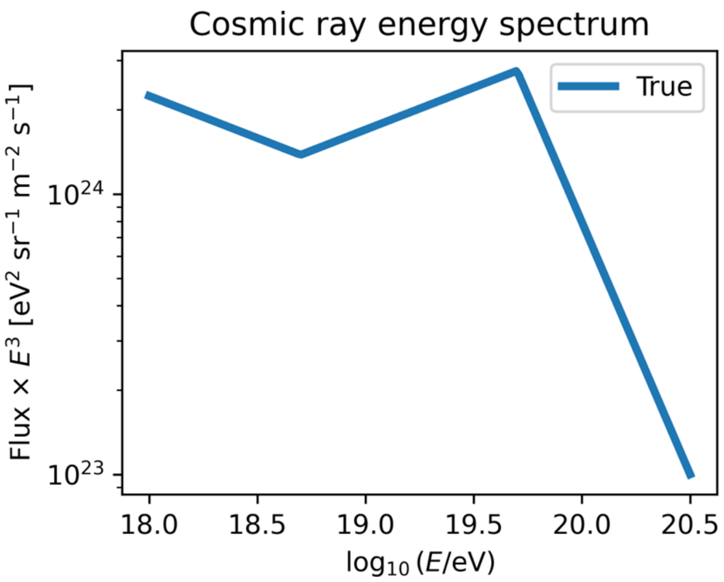
Difficulties of **Forward-folding** :

- Computationally expensive
  - Requires likelihood evaluation for many set of parameters

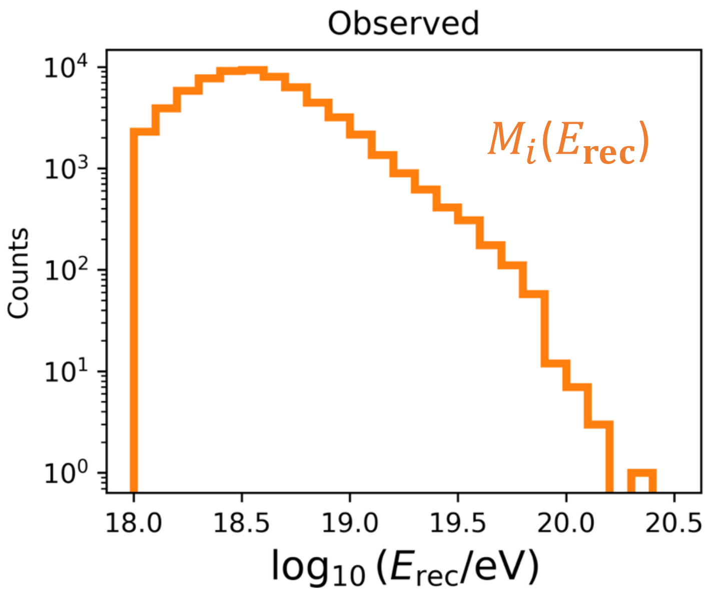
Grid search: $\propto \mathcal{O}(\boldsymbol{N}^{\boldsymbol{d}})$

- N: number of grids per parameter
- d: number of parameters

Forward-folding: free from the inverse problem



Forward-folding



$\boldsymbol{R}$

(effective exposure) × (response) × dE

<u>Difficulties of **Forward-folding**</u> :

- Computationally expensive
  - Requires likelihood evaluation for many set of parameters

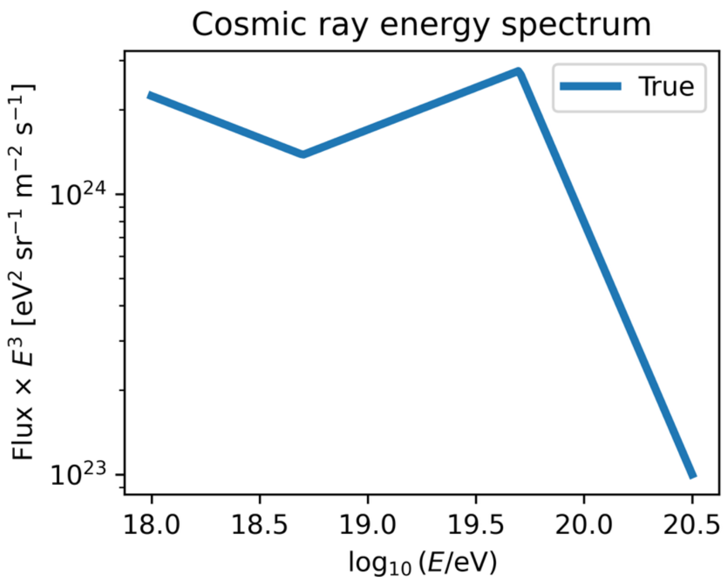
Grid search: $\propto \mathcal{O}(\boldsymbol{N}^{\boldsymbol{d}})$

- N: number of grids per parameter
- d: number of parameters

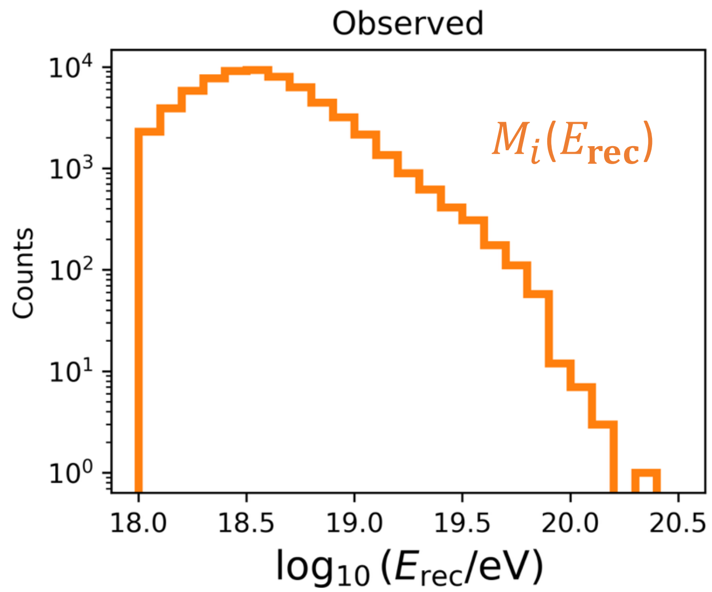
MCMC (Markov Chain Monte Carlo) : $\propto \mathcal{O}(\boldsymbol{M} \cdot \boldsymbol{d})$

- M: number of sampling (nearly)

Forward-folding: free from the inverse problem



Forward-folding



$\boldsymbol{R}$

(effective exposure) × (response) × dE

Difficulties of **Forward-folding** :

- Computationally expensive
  - Requires likelihood evaluation for many set of parameters

It is not a very big deal with MCMC (even for large $\boldsymbol{d}$)

Grid search: $\propto \mathcal{O}(\boldsymbol{N}^{\boldsymbol{d}})$

- N: number of grids per parameter
- d: number of parameters

MCMC (Markov Chain Monte Carlo) : $\propto \mathcal{O}(\boldsymbol{M} \cdot \boldsymbol{d})$

- M: number of sampling (nearly)

MCMC (Markov Chain Monte Carlo)

MCMC:
numerically approximate high-dimensional integrals

$$\int d\theta\, f(\theta) p(\theta) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} f(\theta_n)$$

(e.g.) expectation value of $f(\theta)$
θ: set of parameters
N: number of samples

Hamiltonian Monte Carlo (HMC):
one of the MCMC algorithms (we will use this)

- Demo: https://chi-feng.github.io/mcmc-demo/app.html

Bayesian inference

$$p(\theta|x) \propto p(x|\theta)\ p(\theta)$$

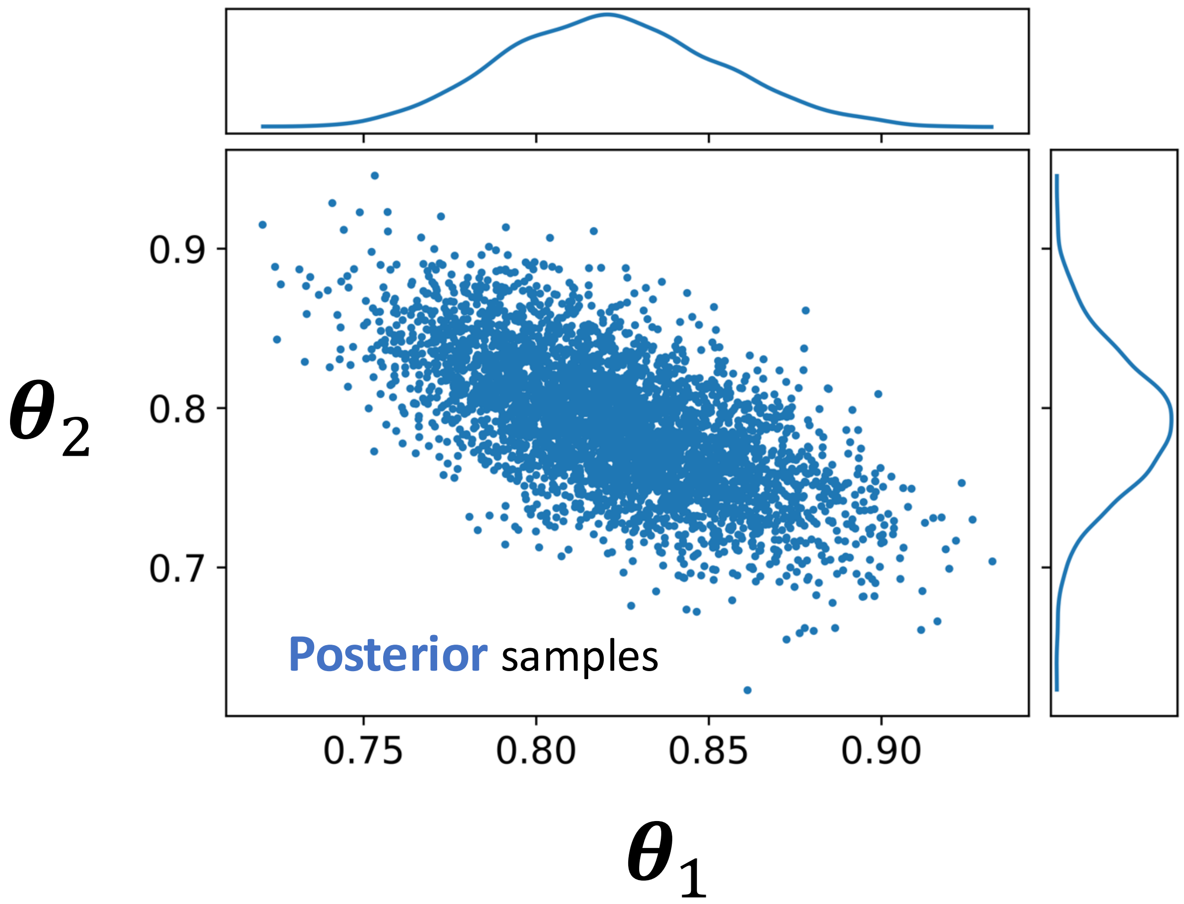
Posterior Likelihood Prior

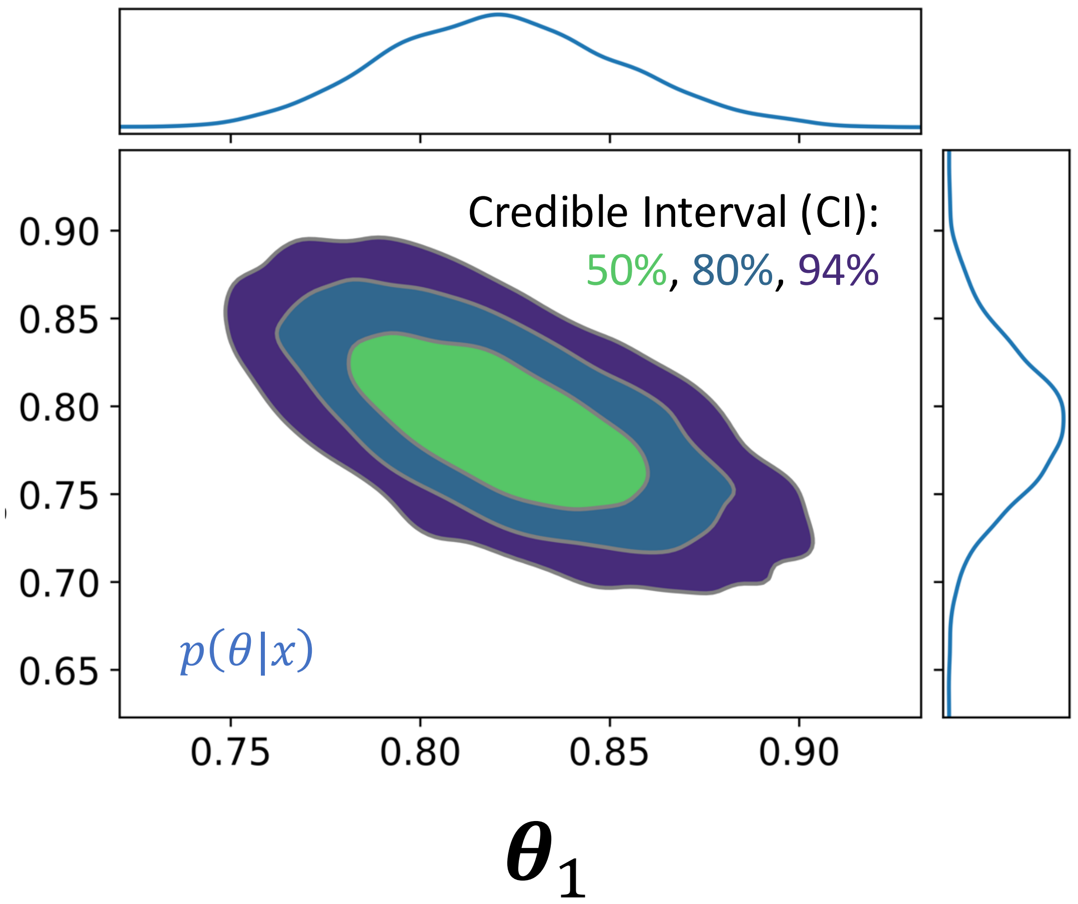
$\boldsymbol{\theta}$: model parameters
$\boldsymbol{x}$: observed data

MCMC efficiently samples from the **posterior** distribution of parameters

Bayesian inference

(e.g.) Two parameters

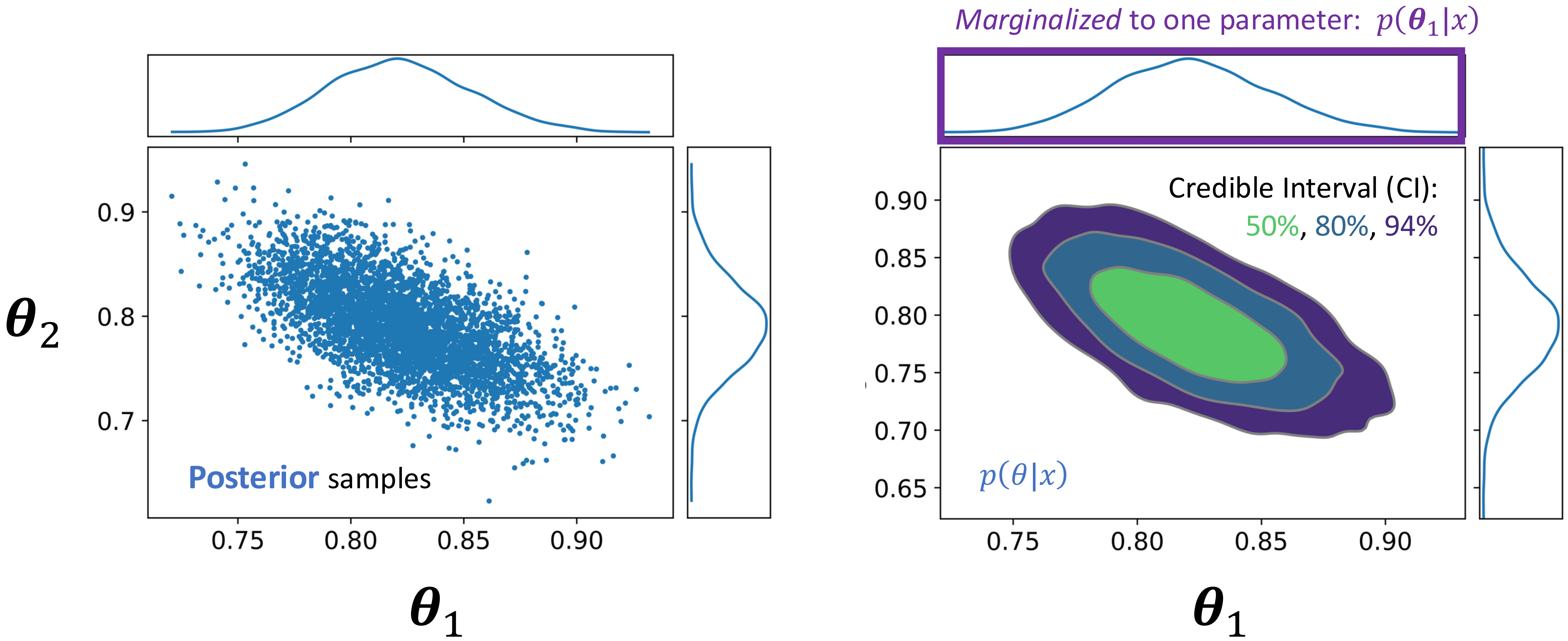




- **Posterior** samples approximate the posterior distribution (requires sufficiently large samples).
- Correlations of parameters are naturally captured.
  - Works efficiently even for a large number of parameters !

Bayesian inference

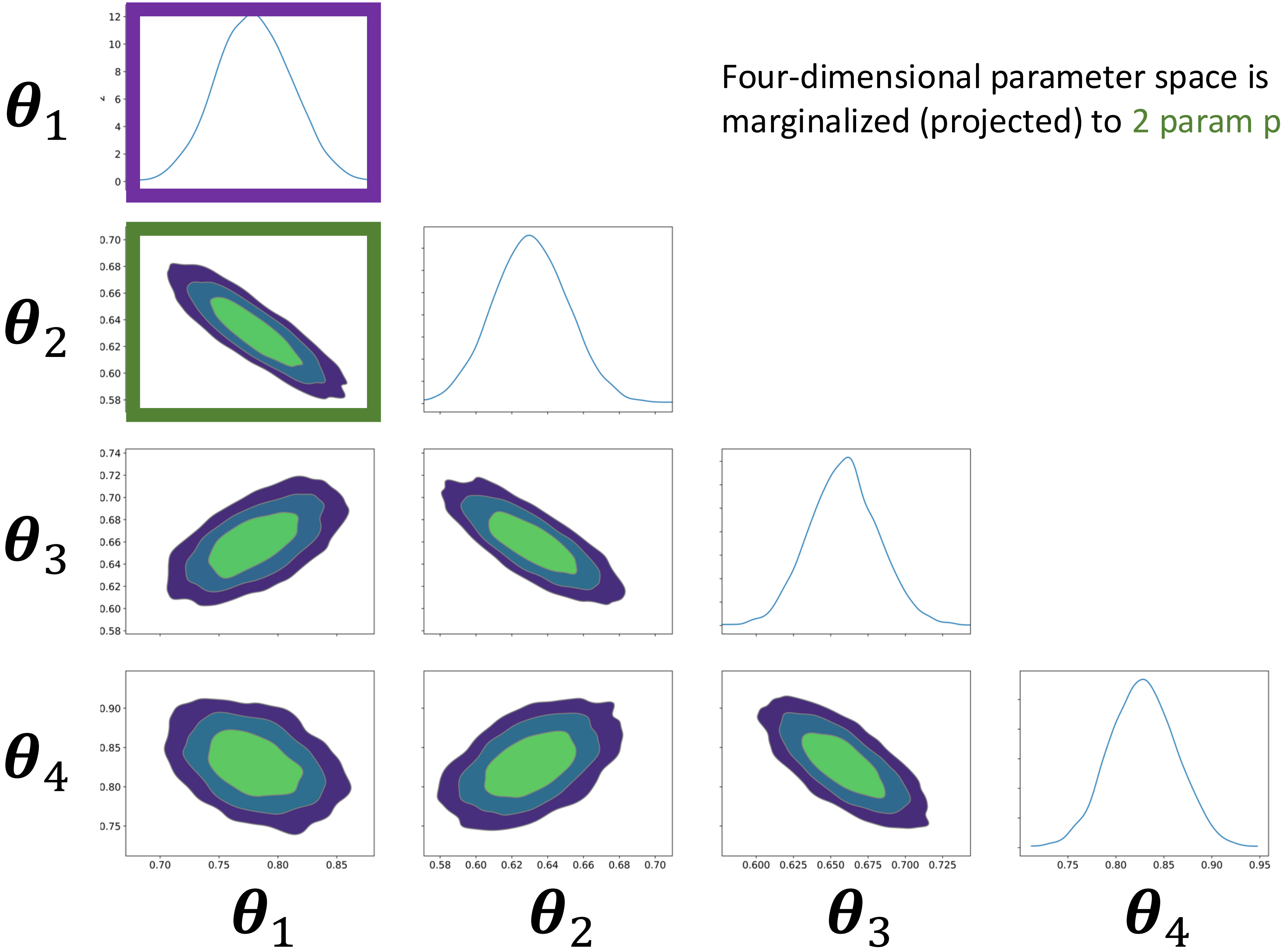
(e.g.) Two parameters



- **Posterior** samples approximate the posterior distribution (requires sufficiently large samples).
- Correlations of parameters are naturally captured.
    - Works efficiently even for a large number of parameters !

Bayesian inference

(e.g.) Four parameters



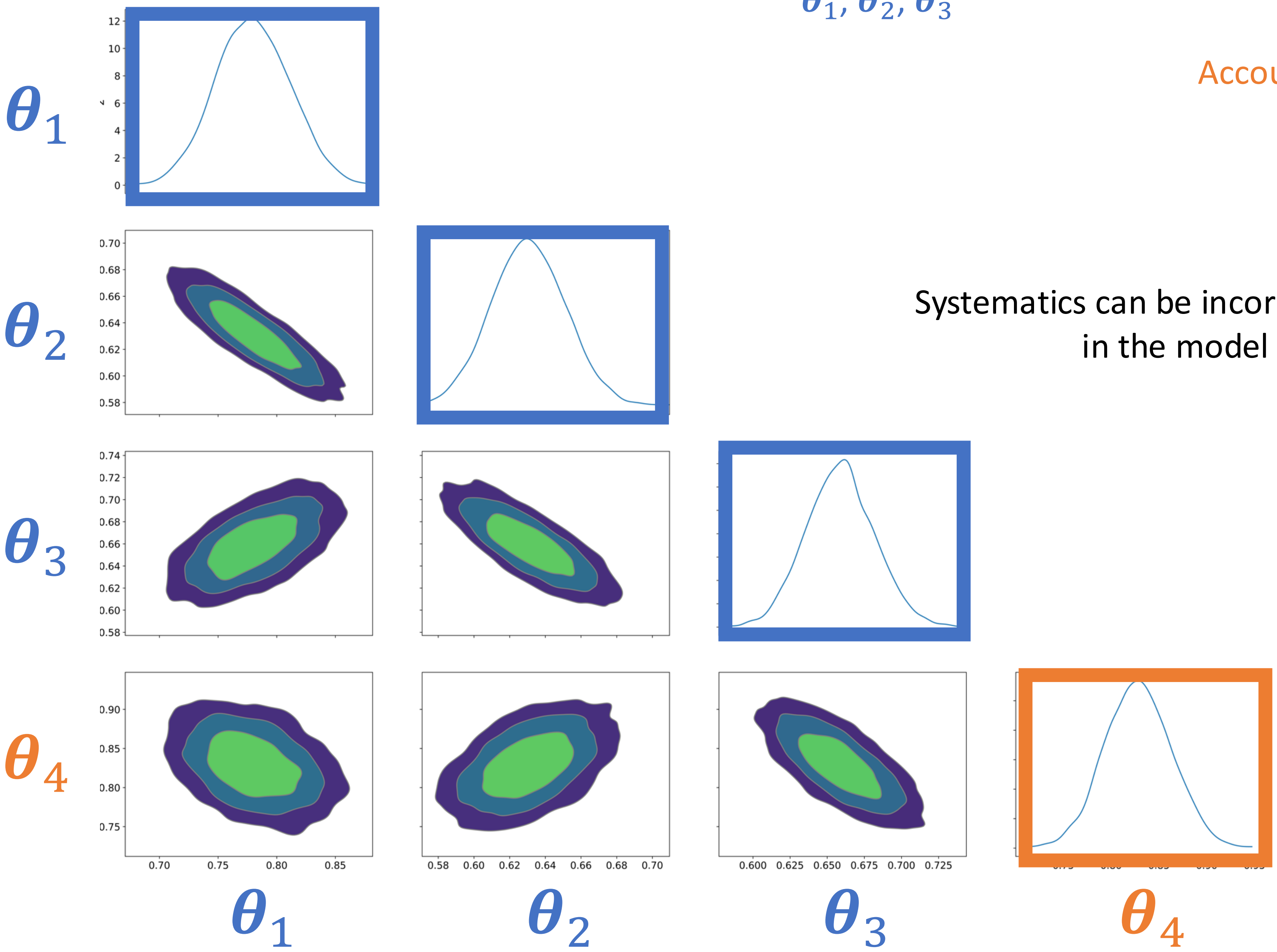
Four-dimensional parameter space is marginalized (projected) to 2 param pairs / 1 param.

Bayesian inference

(e.g.) Three parameters of interest + One nuisance parameter

$\theta_1, \theta_2, \theta_3$ θ_4

Accounts for systematics



Systematics can be incorporated in the model as parameters

Bayesian inference

$$p(\theta|x) \propto p(x|\theta)\ p(\theta)$$

Posterior Likelihood Prior

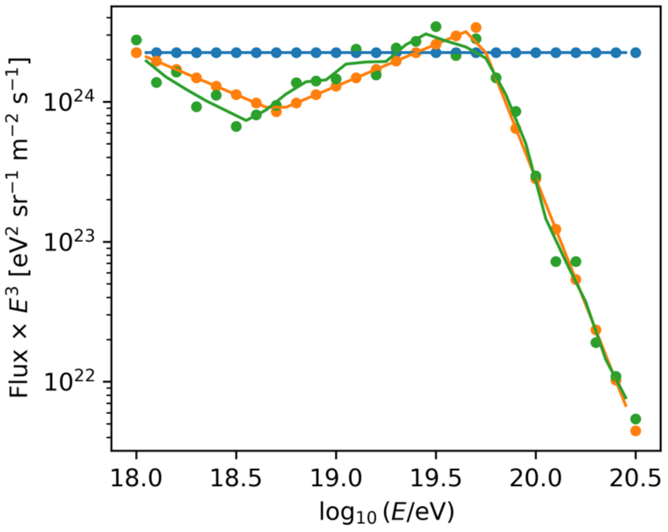
$\boldsymbol{\theta}$: model parameters
$\boldsymbol{x}$: observed data

- In Bayesian inference, prior is also important
  - If our data is not sensitive to the parameters at all (i.e. $p(x|\theta)$ = const.), posterior follows priors: $p(\theta|x) \propto p(\theta)$
  - Bayesian inference is like updates of the prior (our knowledge before the observation) using our observed data.
  - If the prior is poorly chosen, the resulting posterior may be unreliable.

Forward-folding framework: this hands-on (01)

Model: cosmic ray flux (at each energies)

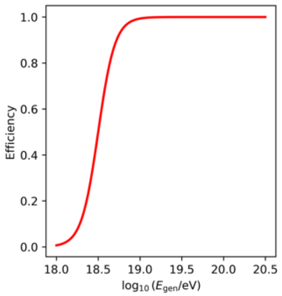
parameterized with $\boldsymbol{\theta}$



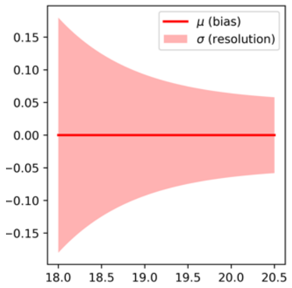
and prior $p(\theta)$

Forward model (Response)

$\boldsymbol{R(\theta) \rightarrow x}$



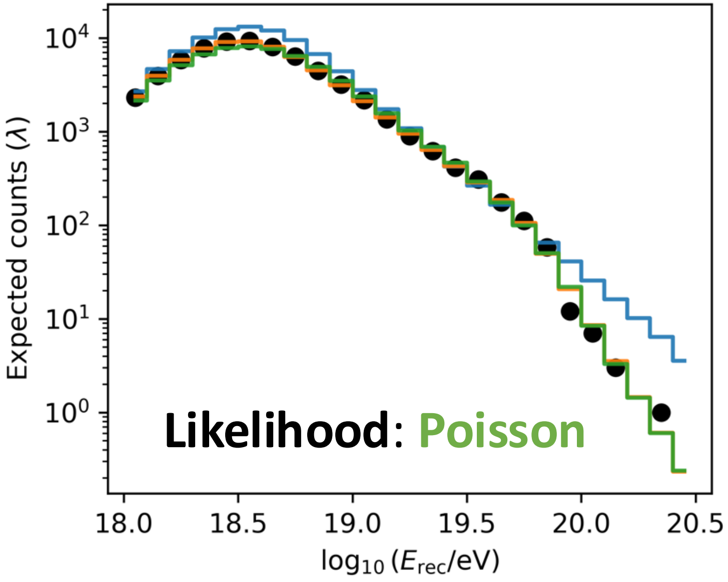
Effective exposure



Energy response

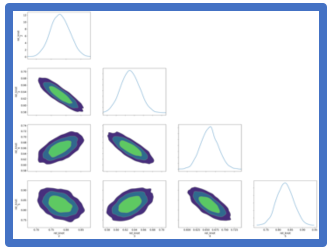
Observed data $\boldsymbol{x}$:

reconstructed energy event counts



MCMC sampling

Posterior $p(\theta|x)$



(only showing four params here)

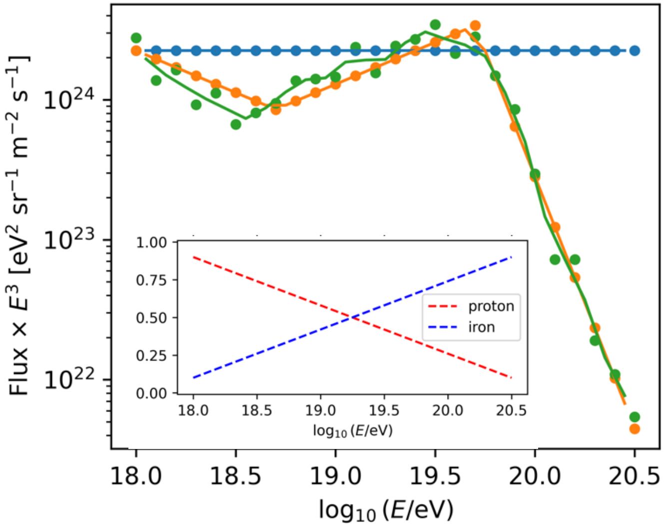
Python + Stan

(MCMC sampling)

Forward-folding framework: this hands-on (02)

Model: cosmic ray flux (at each energies) + mass fractions (p, Fe)

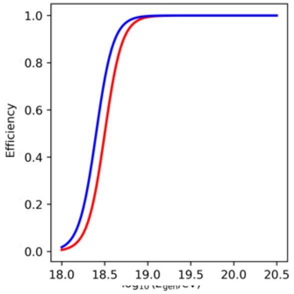
parameterized with $\boldsymbol{\theta}$



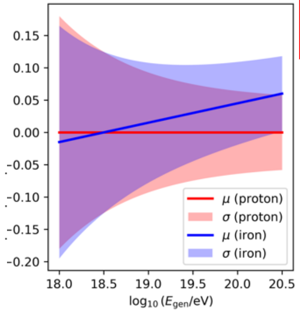
and prior $p(\theta)$

Forward model (Response)

$\boldsymbol{R(\theta)} \rightarrow \boldsymbol{x}$



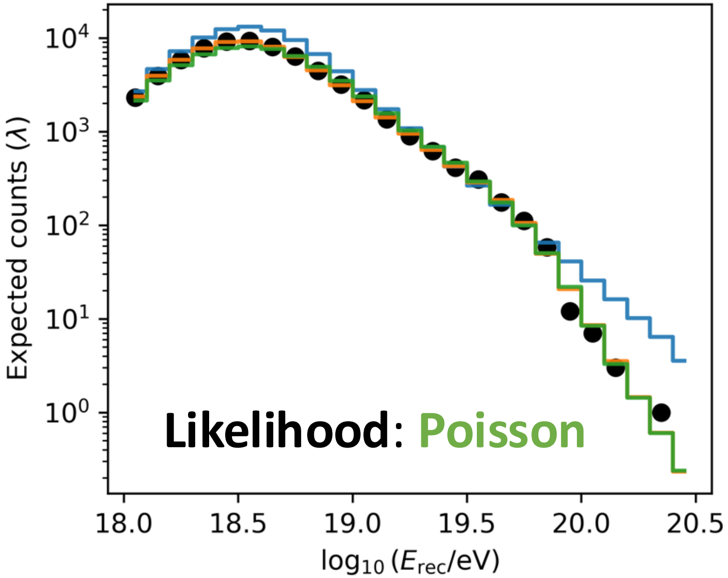
Effective exposure



Energy response

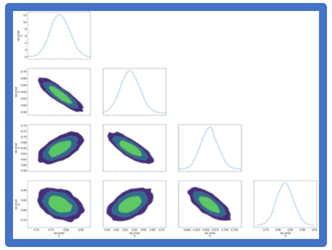
Observed data $\boldsymbol{x}$:

reconstructed energy event counts



MCMC sampling

Posterior $p(\theta|x)$

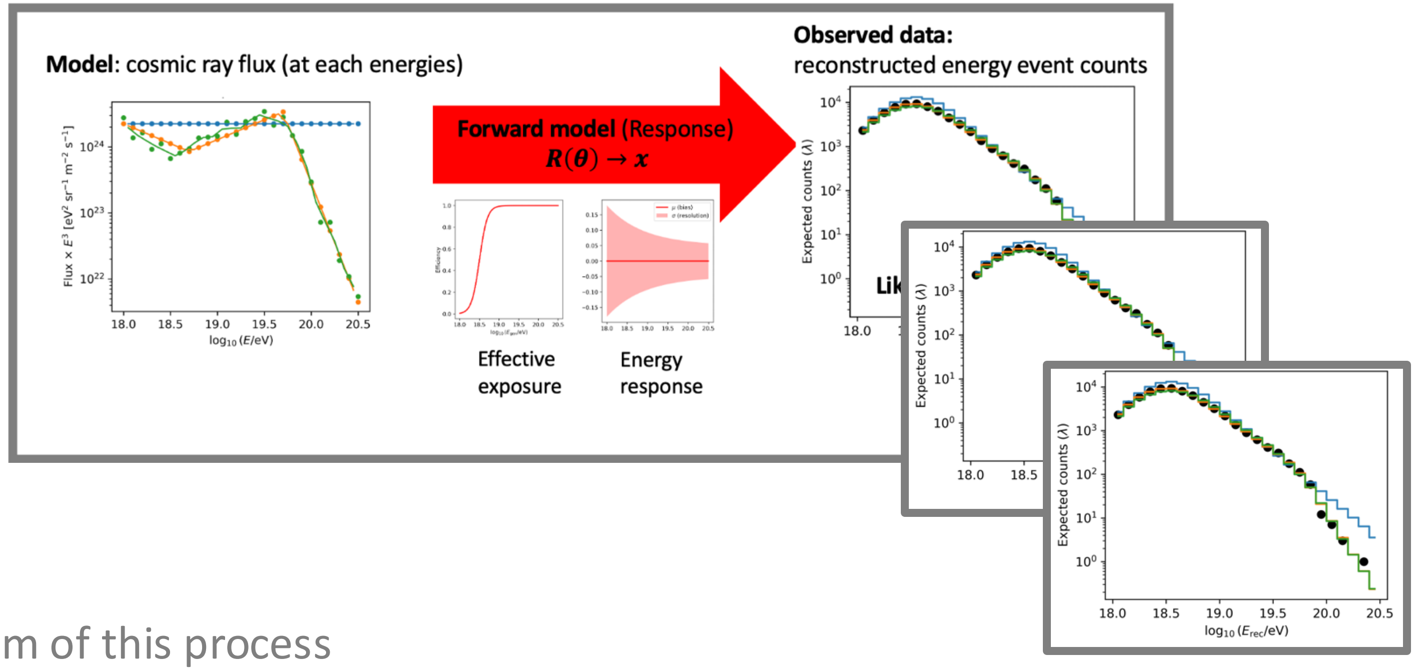


(only showing four params here)

Python + Stan

(MCMC sampling)

(Not covered in this hands-on) Forward-folding framework extension



Imagine solving the inverse problem of this process
considering correlations of all parameters

- Multiple observables
- Multiple detectors (TA, TALE, TA×4, …)

Summary

- **Forward-folding** is free from solving the inverse problem.
  - Model can be (relatively) easily extended.

- **MCMC** is a powerful tool to sample posteriors for large number of parameters.
  - Many processes (including systematics) can be incorporated.

Preparation for the hands-on

- Python environment setup
  - python3 -m venv ~/venv-stan
  - source ~/venv-stan/bin/activate
    - Which python
  - pip install --upgrade pip
  - pip install numpy scipy matplotlib cmdstanpy arviz ipykernel
  - python -m cmdstanpy.install_cmdstan --verbose
  - python -m ipykernel install --user --name=venv-stan --display-name "Python3.12 (venv-stan)”

- When you launch a notebook, choose "Python3.12 (venv-stan)” (upper right)