Simulation Based Inference for X-ray astrophysics

Simon Dupourqué (IRAP, Toulouse)
Lumières Workshop (2026)

IFU
irap
astrophysique & planétologie

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

1. A short recap on Bayesian inference
2. Introduction to Simulation-Based Inference (SBI) and its building blocks

} X-ray spectroscopy as an example

3. Turbulence and surface brightness fluctuations in galaxy clusters

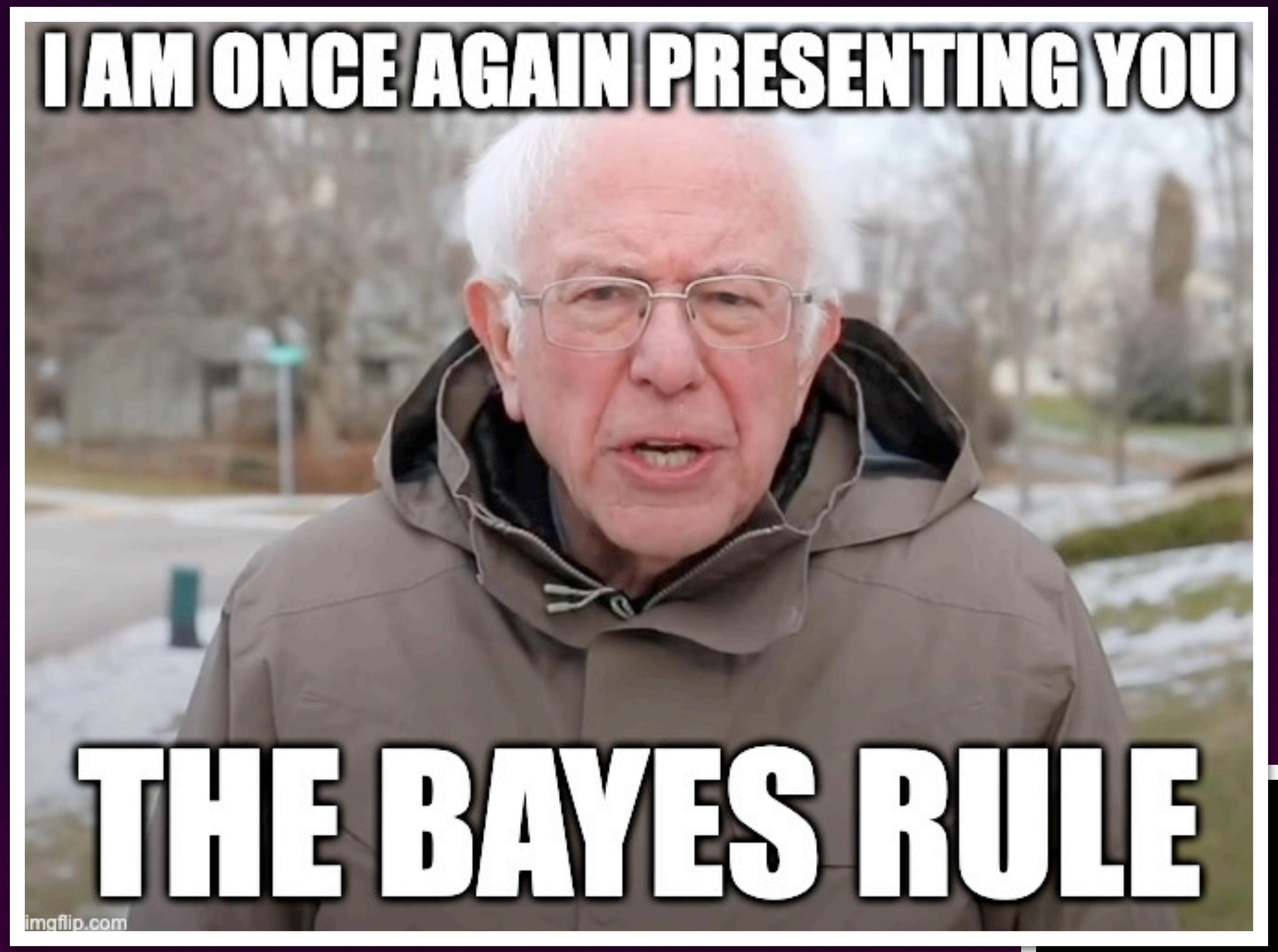
} Application to X-COP and CHEXMATE cluster samples

Bayesian inference

θ : parameters

X: observation(s)

Bayesian inference

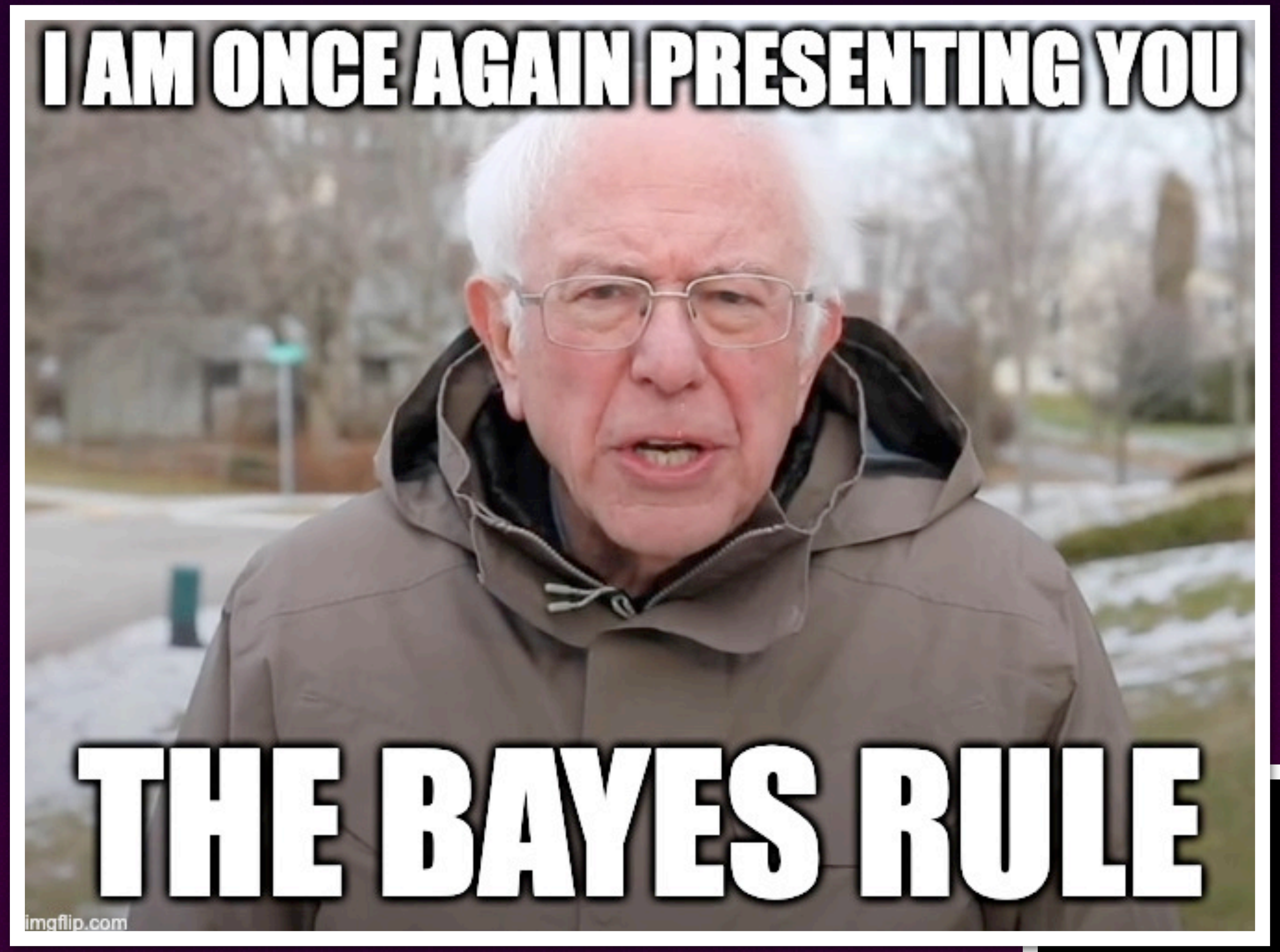


θ : parameters

X: observation(s)

$$P(\theta \mid X) = \frac{P(X \mid \theta)}{P(X)} P(\theta)$$

Bayesian inference



θ : parameters

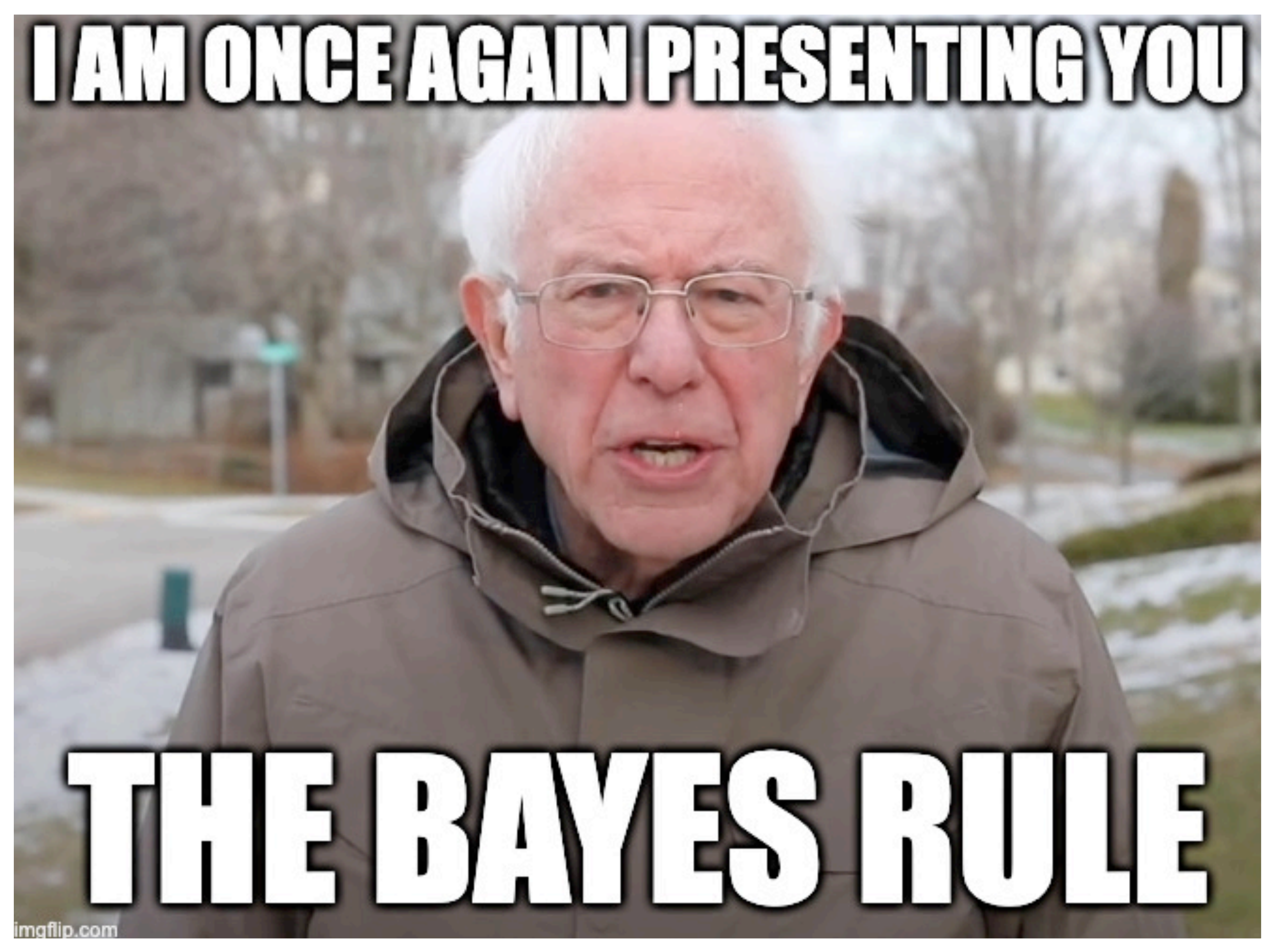
X: observation(s)

$$P(\theta | X) = \frac{P(X | \theta)}{P(X)} P(\theta)$$

Prior

A *priori* probability of the parameters

Bayesian inference



θ : parameters

X: observation(s)

$$P(\theta | X) = \frac{P(X | \theta)}{P(X)} P(\theta)$$

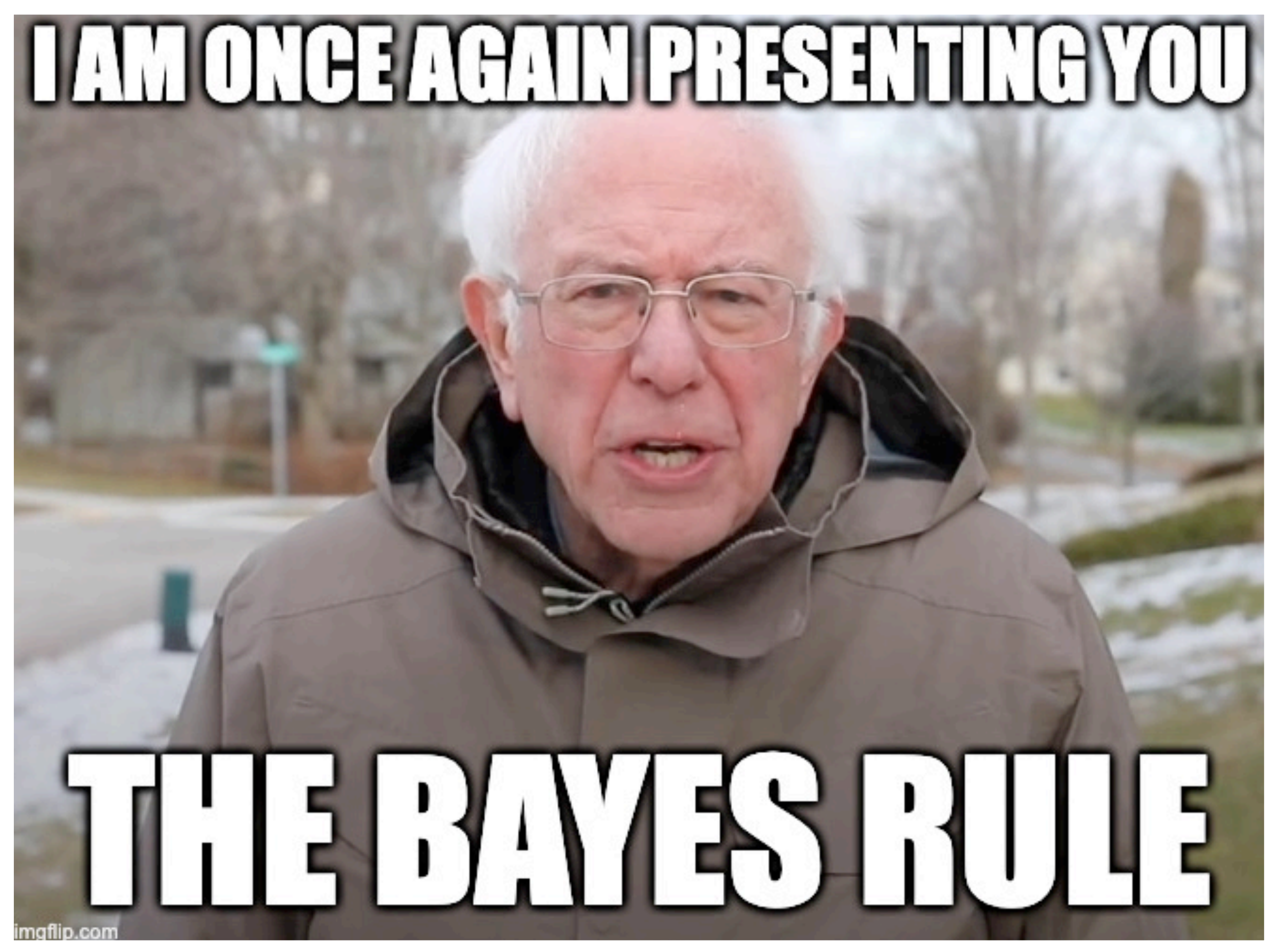
Likelihood

Probability of the observation(s) given the parameters

Prior

A *priori* probability of the parameters

Bayesian inference



θ : parameters

X: observation(s)

$$P(\theta | X) = \frac{P(X | \theta) P(\theta)}{P(X)}$$

Likelihood

Probability of the observation(s) given the parameters

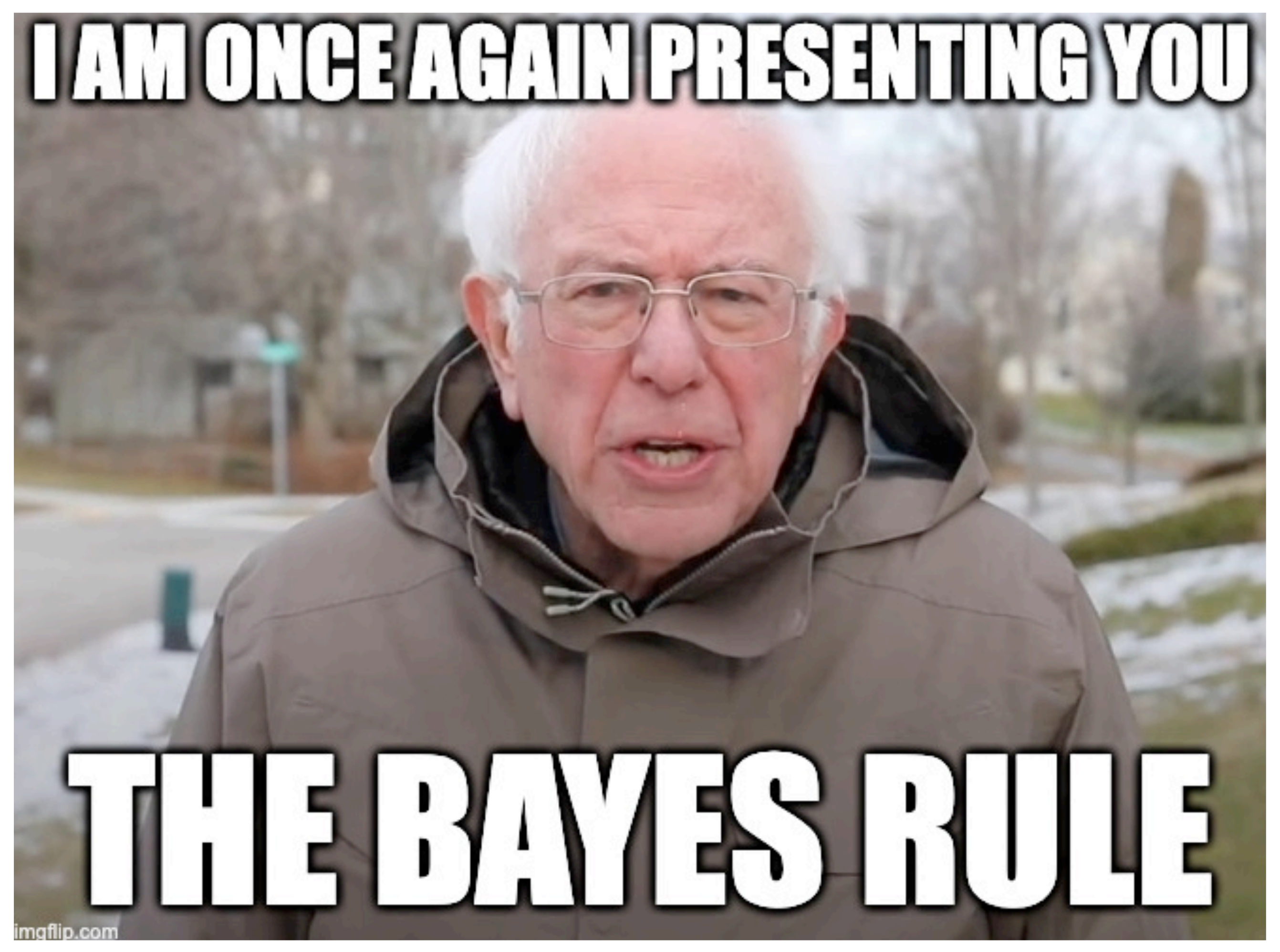
Posterior

A *posteriori* of the parameters given the observation(s)

Prior

A *priori* probability of the parameters

Bayesian inference



θ : parameters

X: observation(s)

$$P(\theta | X) = \frac{P(X | \theta) P(\theta)}{P(X)}$$

Likelihood

Probability of the observation(s) given the parameters

Posterior

A *posteriori* of the parameters given the observation(s)

Prior

A *priori* probability of the parameters

Reason why Bayesian inference is hard to perform

Illustration with X-ray spectroscopy

$$\mathcal{S}(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Illustration with X-ray spectroscopy

$$S(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Observed data

Spectra measured by the instrument

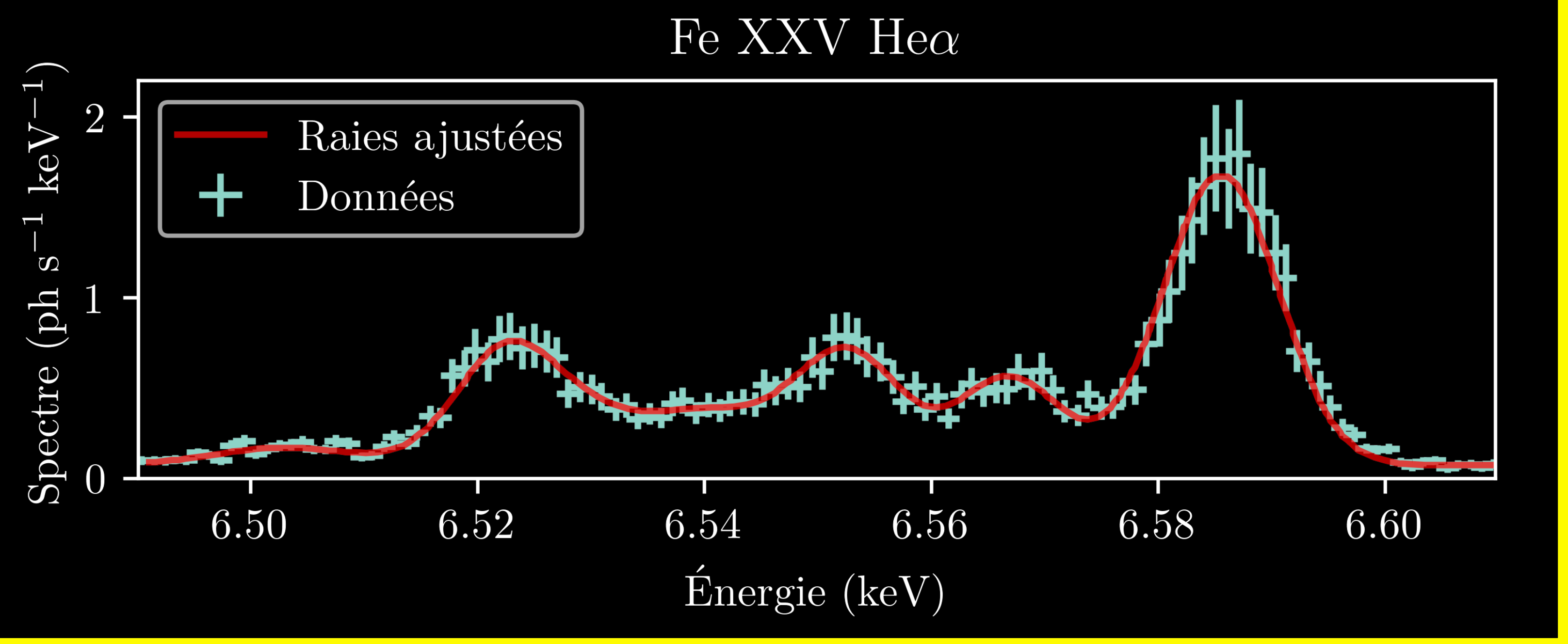
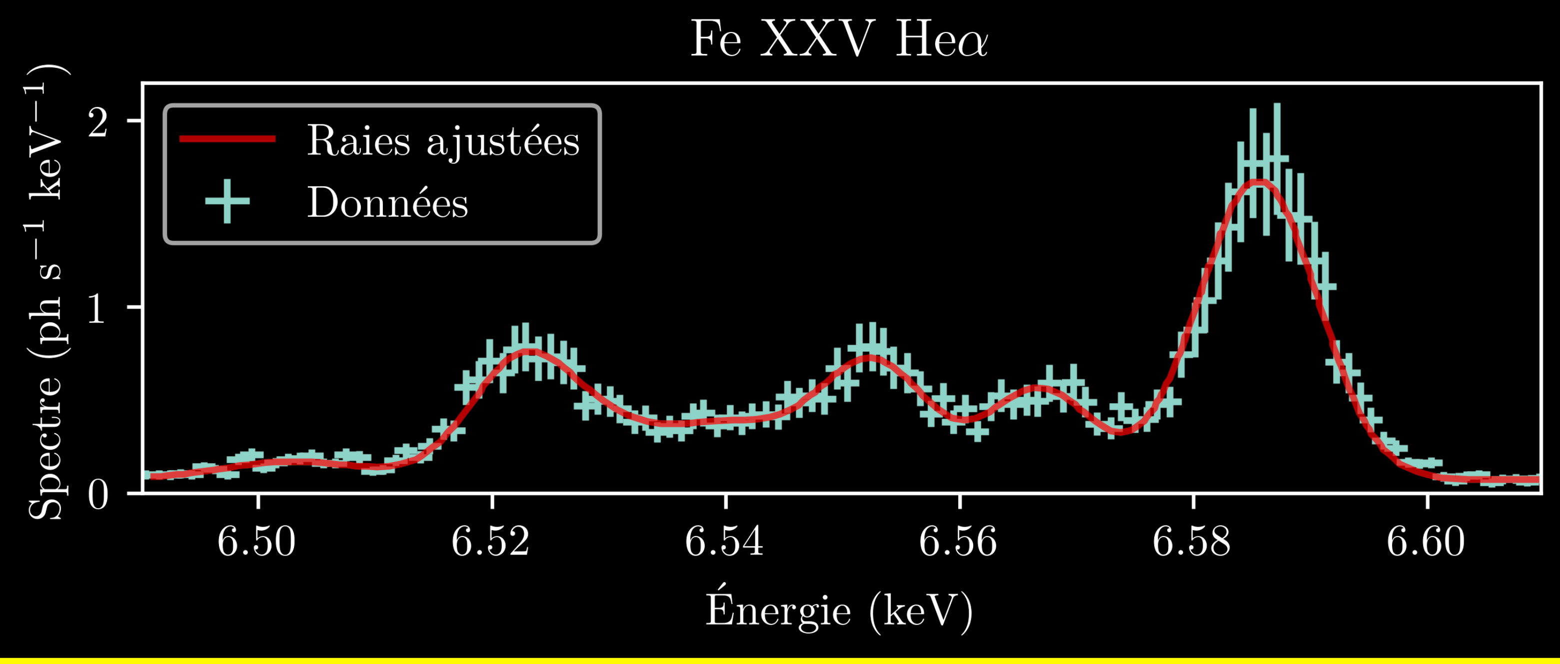


Illustration with X-ray spectroscopy

$$\mathcal{S}(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

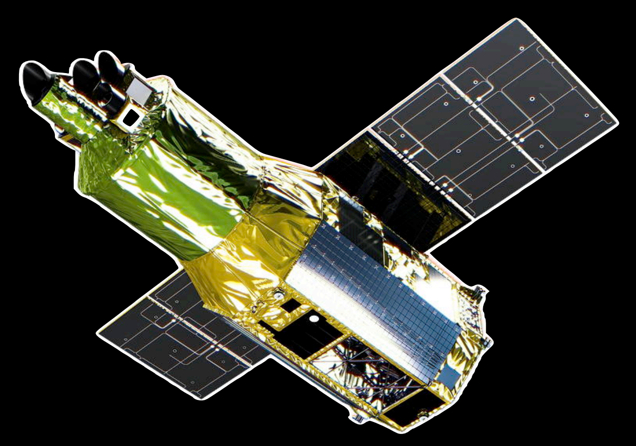
Observed data

Spectra measured by the instrument



Model

Spectral model and instrumental convolution



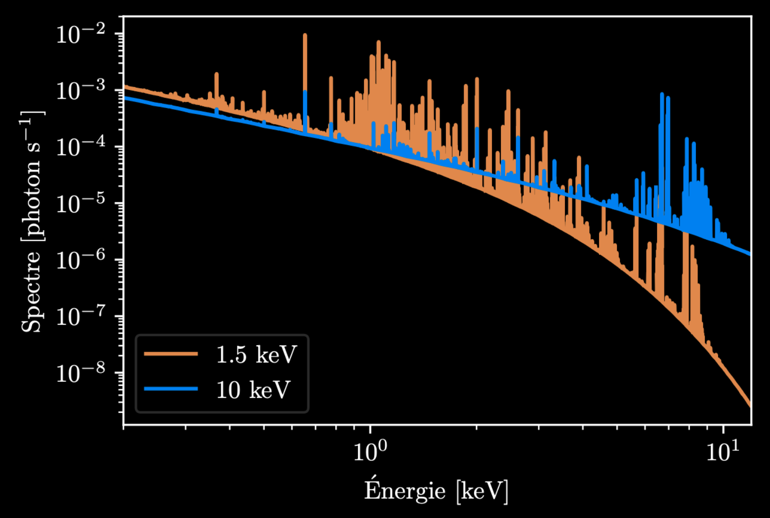


Illustration with X-ray spectroscopy

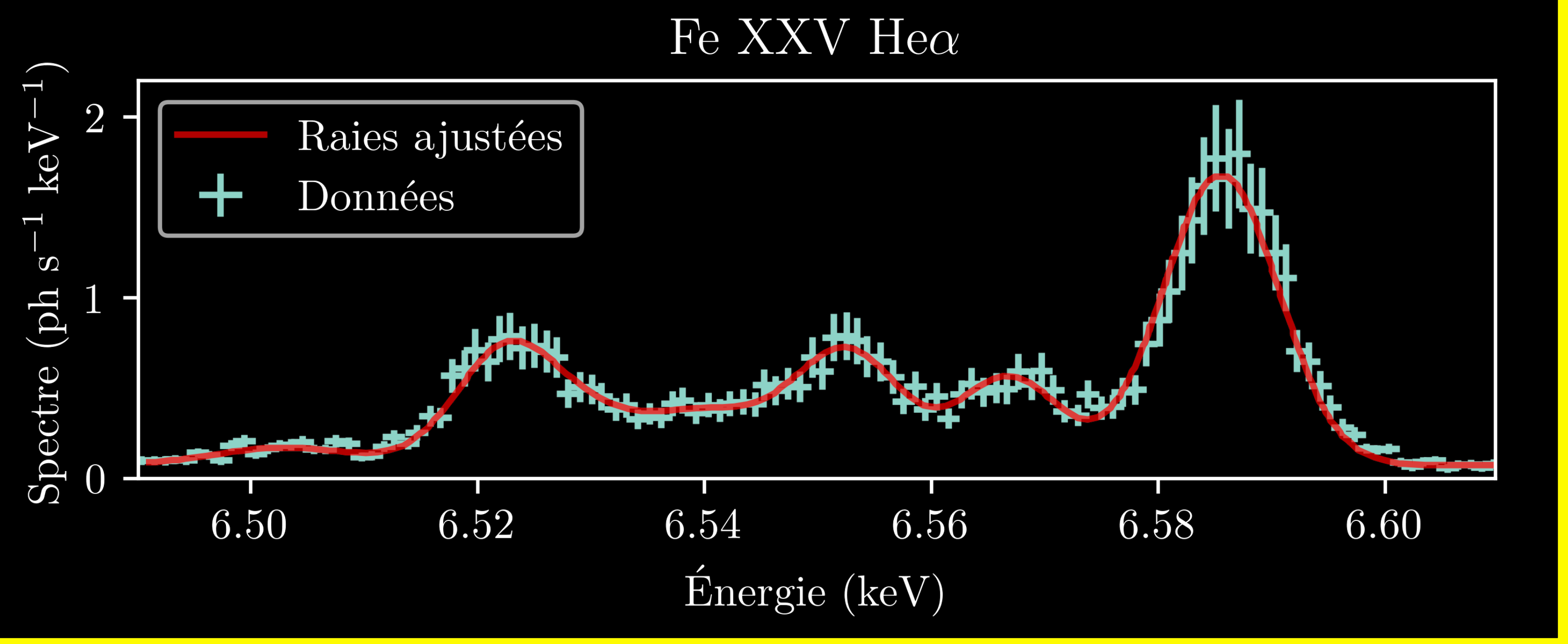
$$S(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Parameters

Temperature, Abundance, Redshift…

Observed data

Spectra measured by the instrument



Model

Spectral model and instrumental convolution

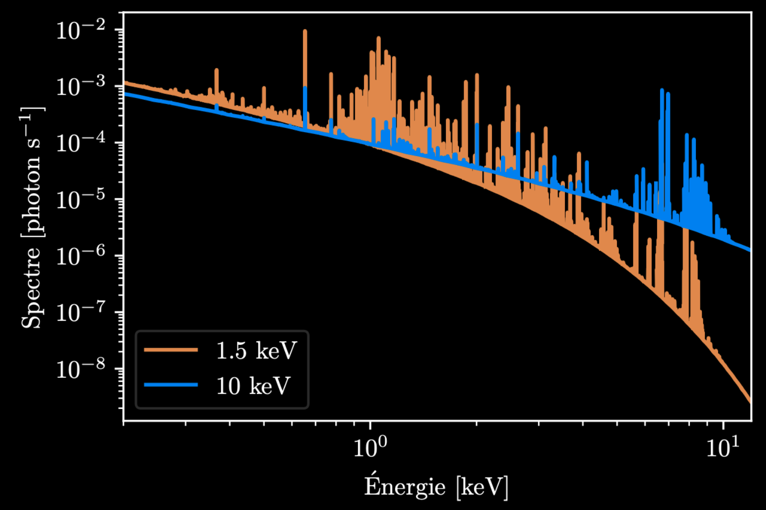


Illustration with X-ray spectroscopy

$$S(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Parameters

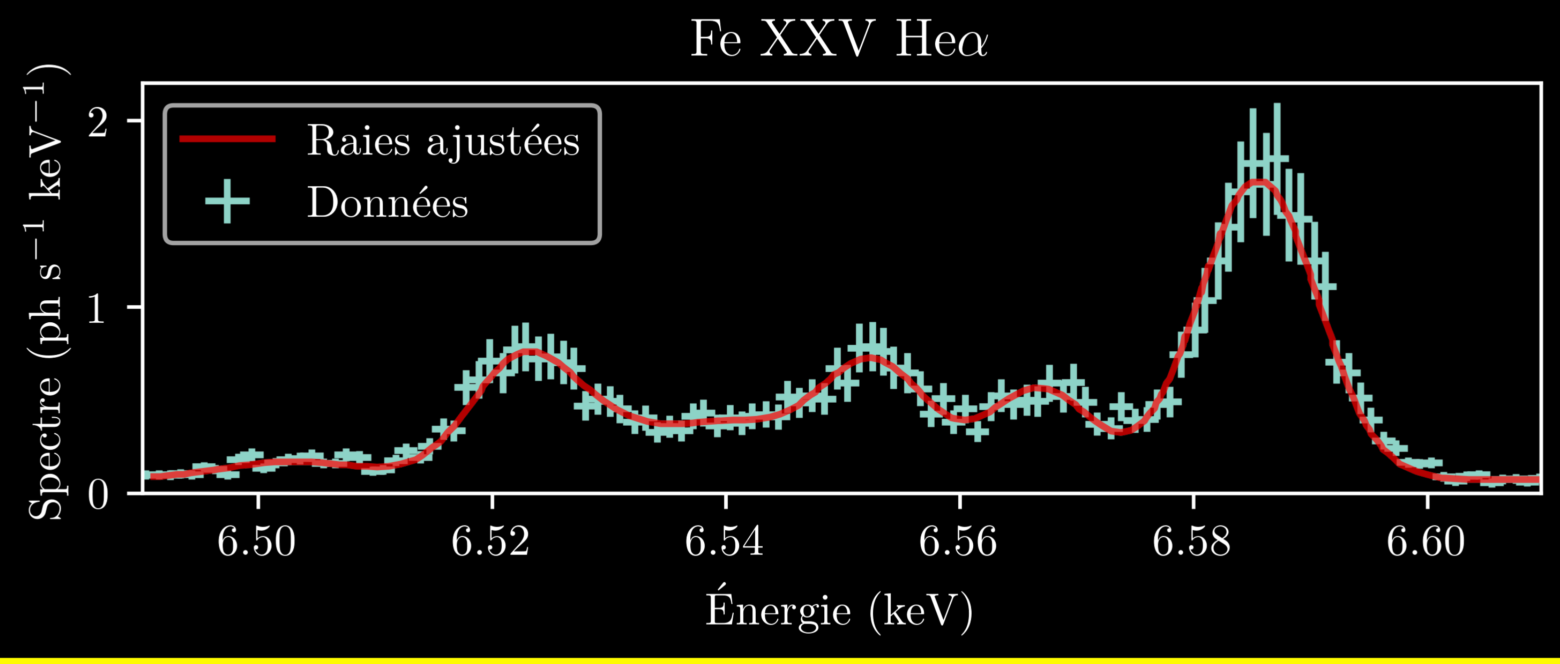
Temperature, Abundance, Redshift…

Expected photons in each channel

$$\lambda \equiv S(E,\theta)$$

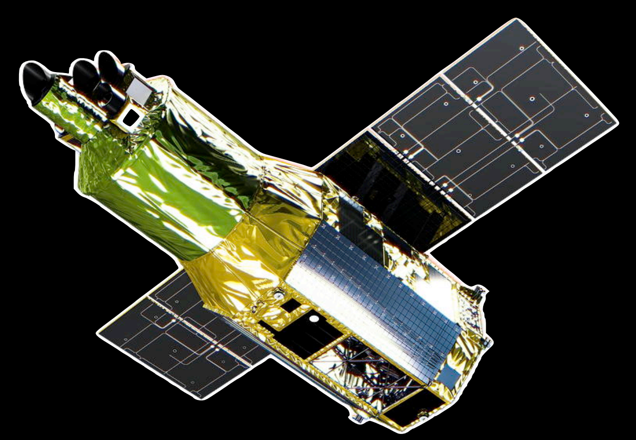
Observed data

Spectra measured by the instrument



Model

Spectral model and instrumental convolution



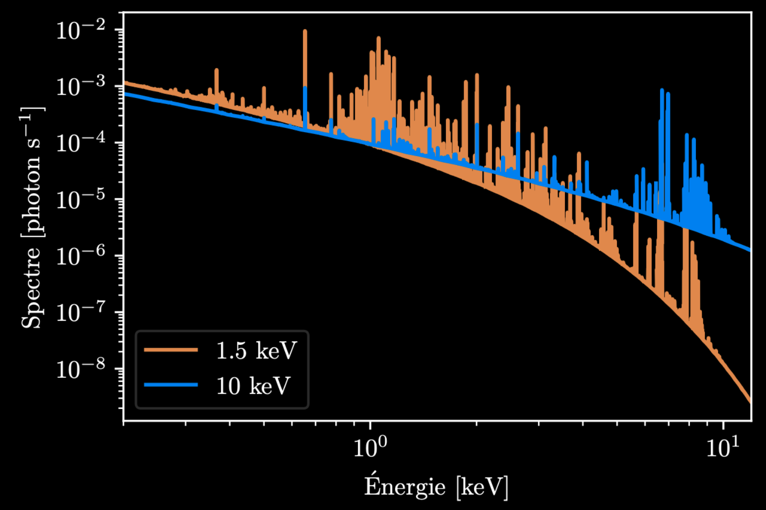


Illustration with X-ray spectroscopy

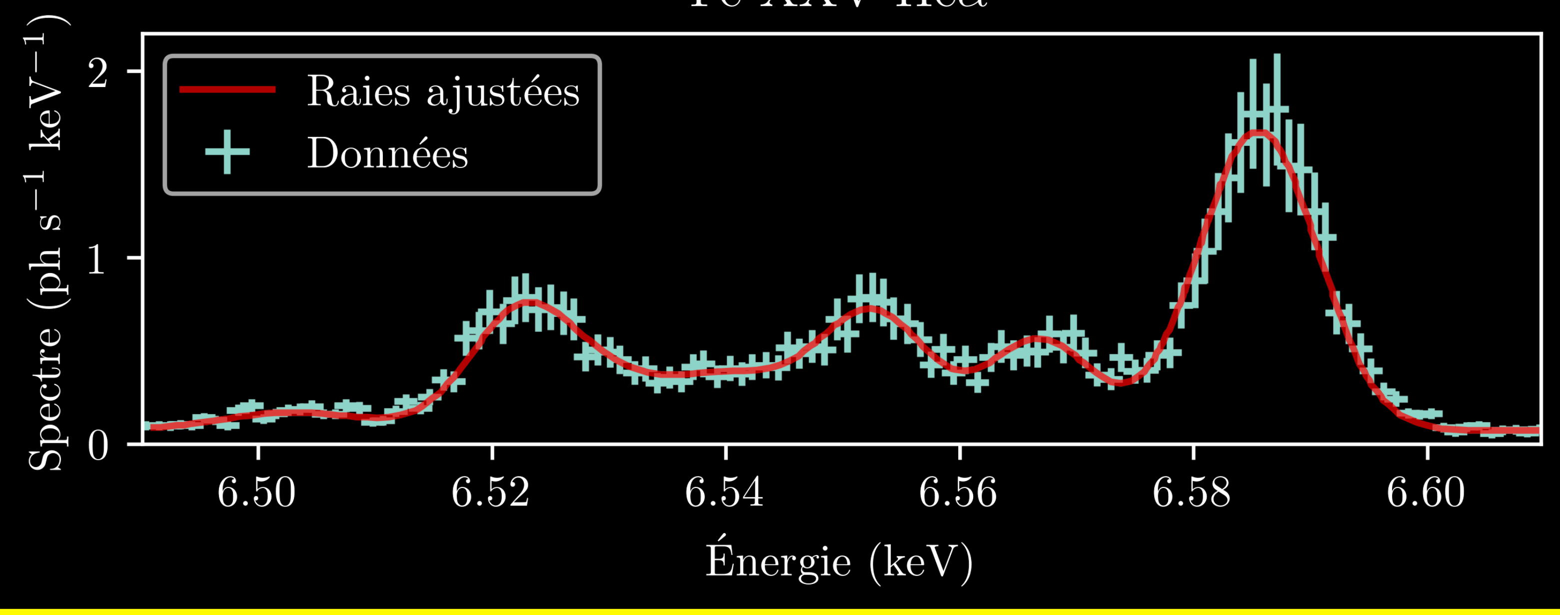
$$\mathcal{S}(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Parameters

Temperature, Abundance, Redshift…

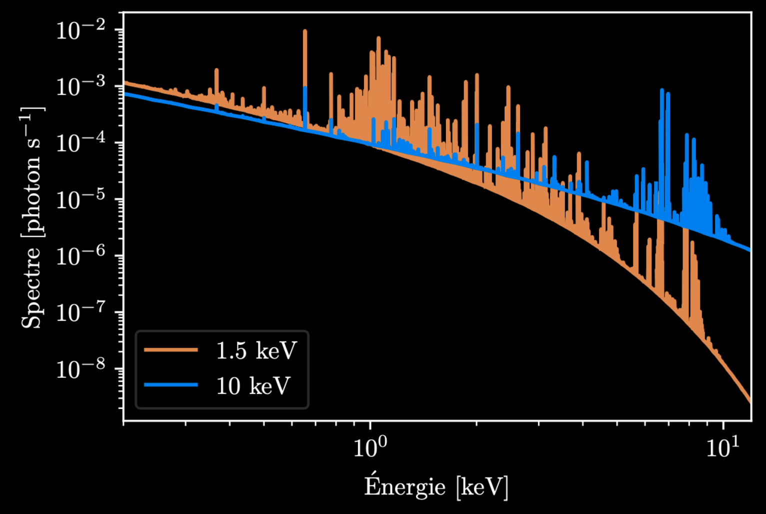
Observed data

Spectra measured by the instrument



Model

Spectral model and instrumental convolution



Expected photons in each channel

$$\lambda \equiv \mathcal{S}(E,\theta)$$

Counting process : Poisson likelihood

$$P(X = k \,|\, \theta) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Turn the likelihood into a posterior distribution

Traditional Bayesian inference

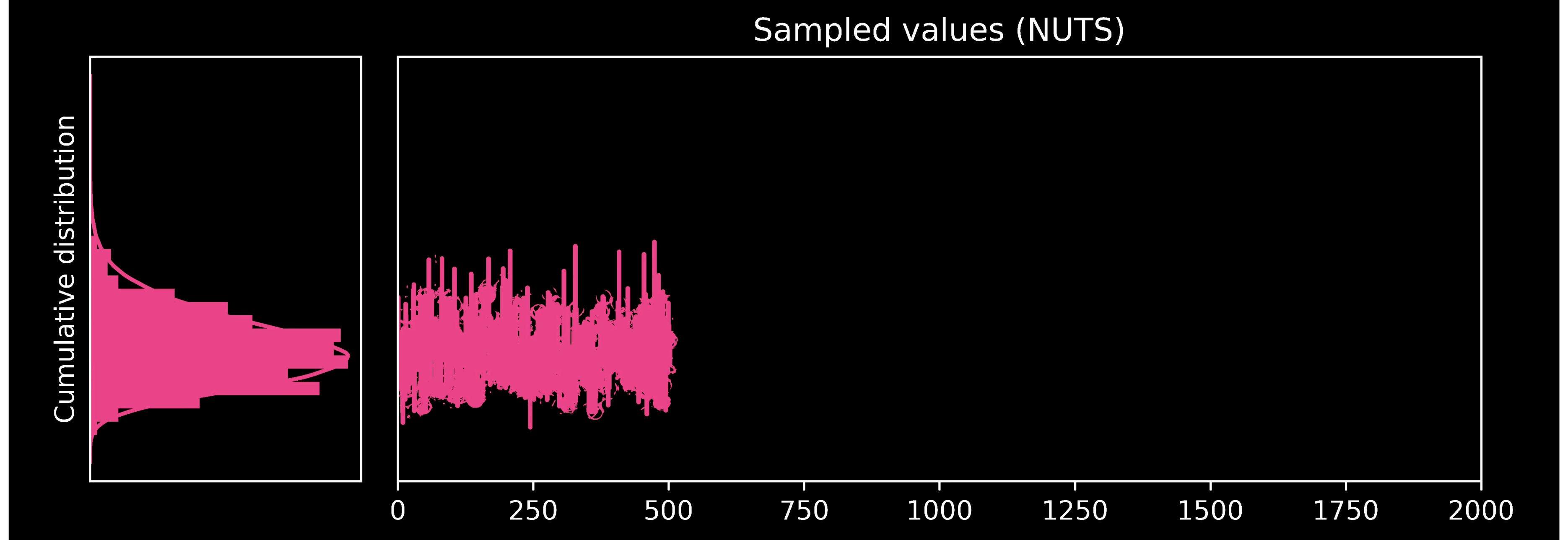
Evaluate

$$P(X|\theta)$$

Traditional Bayesian inference

Sampling $\{\theta\}_i \sim P(\theta | X)$

Monte Carlo Markov Chain (HMC, NUTS, AIES), Nested Sampling



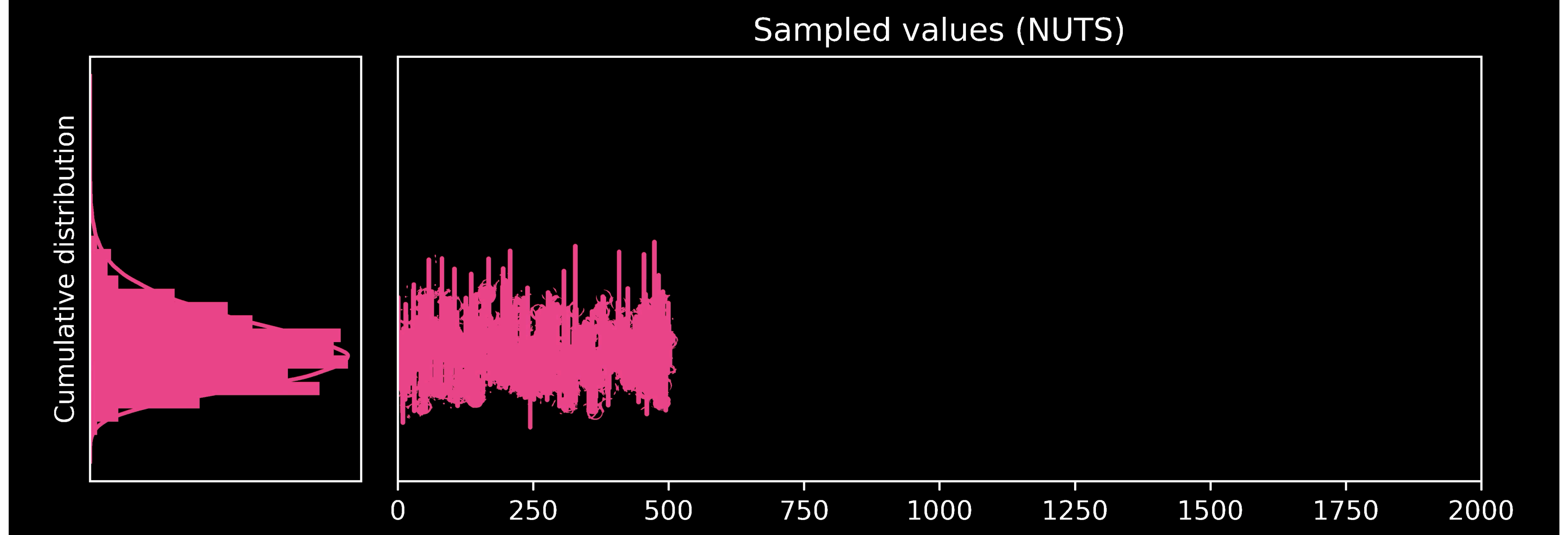
Evaluate

$$P(X | \theta)$$

Traditional Bayesian inference

Sampling $\{\theta\}_i \sim P(\theta|X)$

Monte Carlo Markov Chain (HMC, NUTS, AIES), Nested Sampling

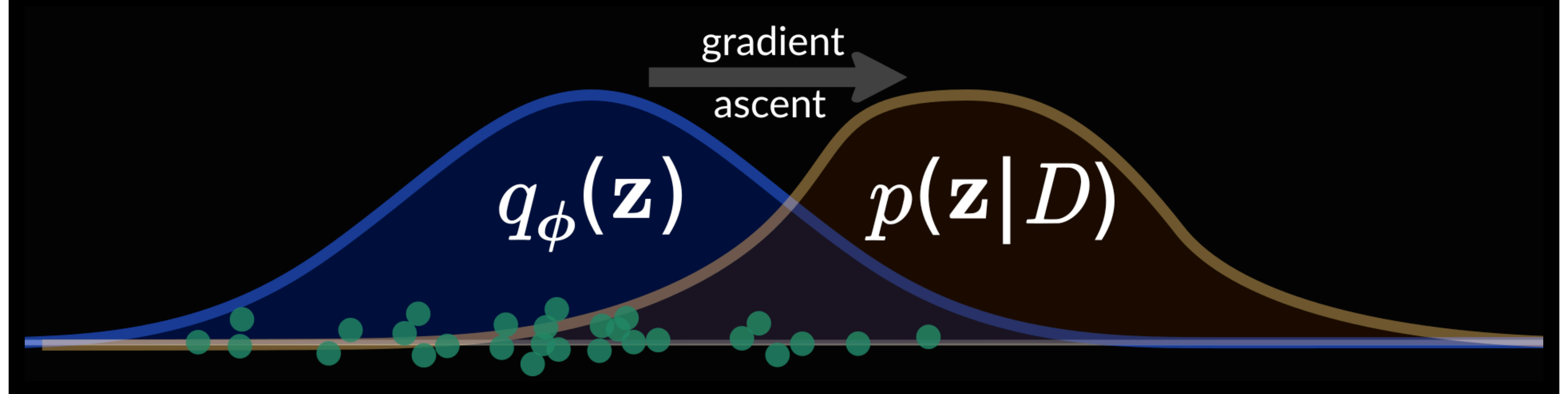


Evaluate

$$P(X|\theta)$$

Variational $q(\theta) \simeq P(\theta|X)$

Minimize Evidence Lower Bound for a parametric and analytical approximation of the posterior distribution



Simulation-based inference (SBI)

Simulation-based inference (SBI)

- Use simulations of the observable to train a neural density estimator to either learn the posterior distribution $P(\theta | X)$, the likelihood $P(X | \theta)$ or Bayes ratios.

Simulation-based inference (SBI)

- Use simulations of the observable to train a neural density estimator to either learn the posterior distribution $P(\theta|X)$, the likelihood $P(X|\theta)$ or Bayes ratios.
- Draw many θ parameters, simulate $X(\theta)$ accordingly, train a network to learn the mapping, *and voilà*.

Simulation-based inference (SBI)

- Use simulations of the observable to train a neural density estimator to either learn the posterior distribution $P(\theta|X)$, the likelihood $P(X|\theta)$ or Bayes ratios.
- Draw many θ parameters, simulate $X(\theta)$ accordingly, train a network to learn the mapping, *and voilà*.
- Works with intractable likelihood functions and transformed representations of the observable

Illustration with X-ray spectroscopy

$$\mathcal{S}(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Illustration with X-ray spectroscopy

$$\mathcal{S}(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

Expected photons in each channel

$$\lambda \equiv S(E,\theta)$$

Illustration with X-ray spectroscopy

$$S(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

↓

Expected photons
in each channel

$$\lambda \equiv S(E,\theta)$$

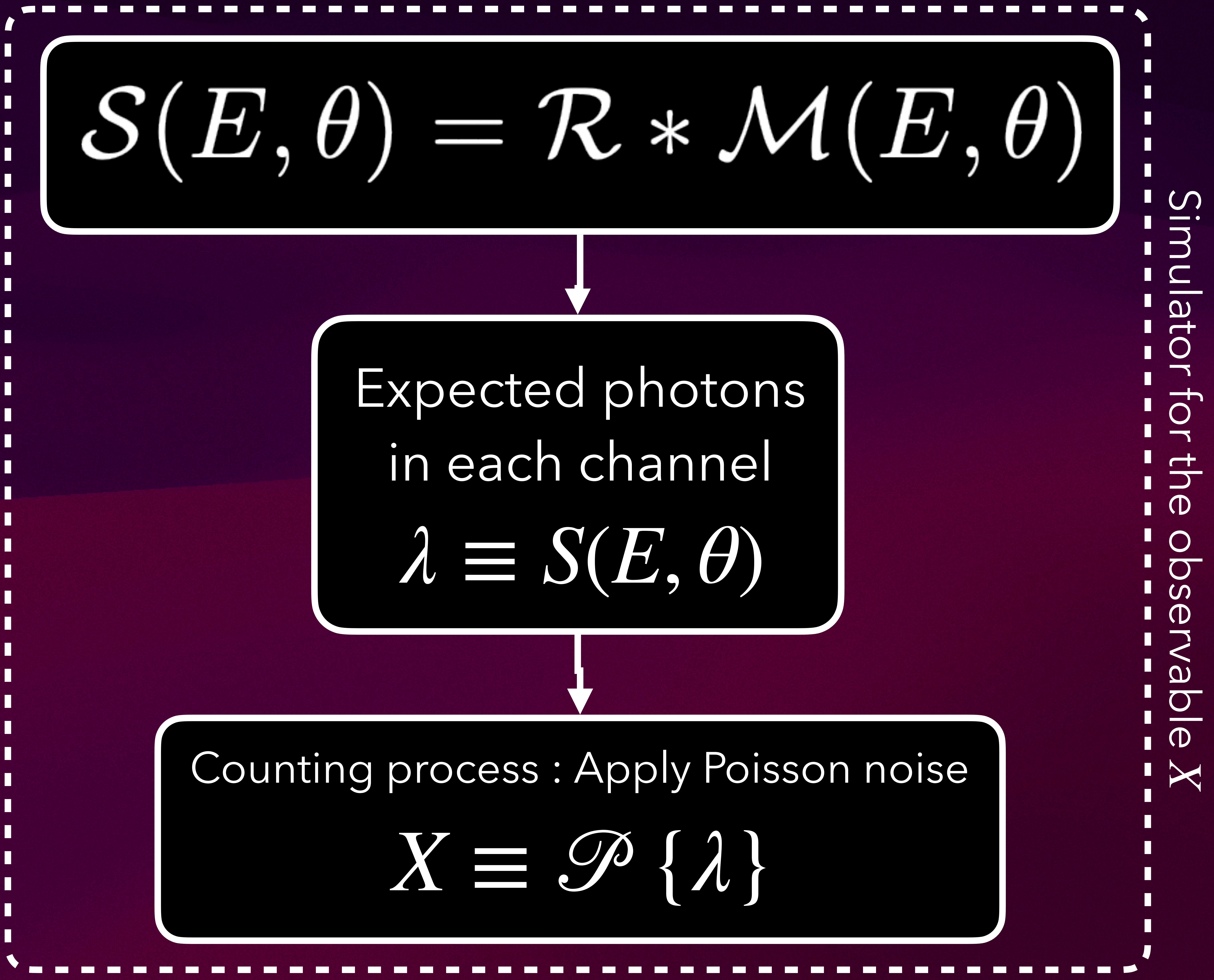
↓

Counting process : Apply Poisson noise

$$X \equiv \mathscr{P}\{\lambda\}$$

Simulator for the observable X

Illustration with X-ray spectroscopy



- Repeat this for many values of parameters θ and generate pairs of $\{\theta, X\}_i$

Illustration with X-ray spectroscopy

$$S(E,\theta) = \mathcal{R} * \mathcal{M}(E,\theta)$$

↓

Expected photons in each channel

$$\lambda \equiv S(E,\theta)$$

↓

Counting process : Apply Poisson noise

$$X \equiv \mathscr{P}\left\{\lambda\right\}$$

Simulator for the observable X

- Repeat this for many values of parameters θ and generate pairs of $\{\theta, X\}_i$
- Train a neural network to learn the distribution of parameters and observables

Normalizing flows

Normalizing flows

- Parametric transformations that are **fast to compute** and **easy to invert**

Normalizing flows

- Parametric transformations that are **fast to compute** and **easy to invert**
- **Universal approximators** for well behaved probability distributions

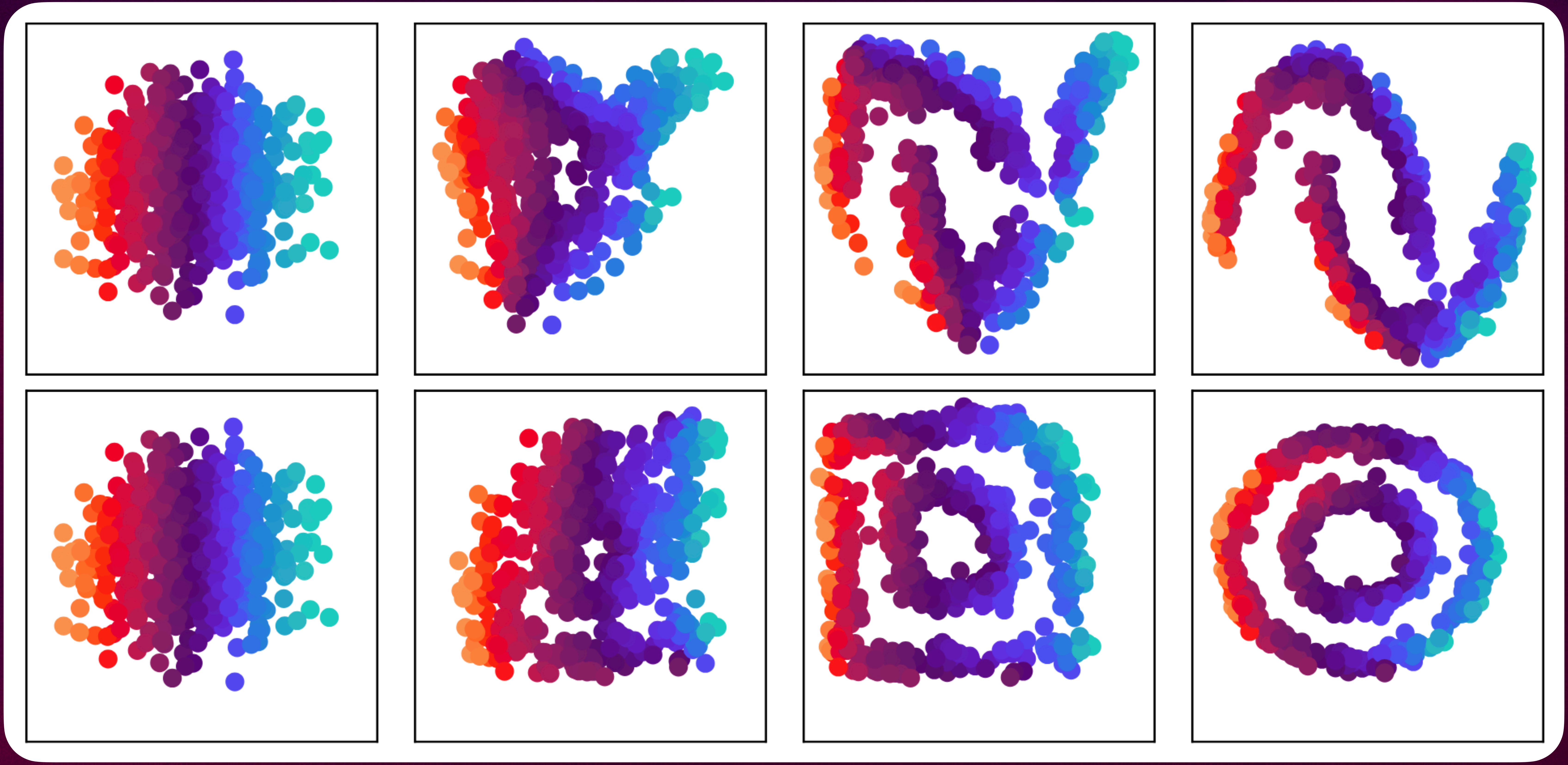
Normalizing flows

- Parametric transformations that are **fast to compute** and **easy to invert**
- **Universal approximators** for well behaved probability distributions
- Learn any distribution as the **transformation** of a **Gaussian** latent variable

Normalizing flows

- Parametric transformations that are **fast to compute** and **easy to invert**
- **Universal approximators** for well behaved probability distributions
- Learn any distribution as the **transformation** of a **Gaussian** latent variable
- Works by stacking reversible blocks of e.g. **Masked Auto-Encoders**

Latent distribution



Learned distribution

$Z_1 \xrightarrow{F_1(Z)} Z_2 \xrightarrow{F_2(Z)} Z_3 \xrightarrow{F_3(Z)} Z_4$

Building the transform blocks

Single block

$$\mathbf{Z} = F(\mathbf{U})$$

$$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u)\,|\det J_F(u)|^{-1}$$

With

$$Z = (z_1, \ldots z_i)$$

$$U = (u_1, \ldots u_i)$$

Building the transform blocks

Single block

$$\mathbf{Z} = F(\mathbf{U})$$

$$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u) \, | \det J_F(u) |^{-1}$$

With

$$Z = (z_1, \ldots z_i)$$

$$U = (u_1, \ldots u_i)$$

The determinant of the Jacobian is the bottleneck

→ Make it **triangular**

Building the transform blocks

Single block

$$\mathbf{Z} = F(\mathbf{U})$$

$$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u)\,|\det J_F(u)|^{-1}$$

With

$Z = (z_1, \ldots z_i)$

$U = (u_1, \ldots u_i)$

The determinant of the Jacobian is the bottleneck
→ Make it **triangular**

Each z_i is a function of the previous entries only
$z_i = f(u_i, u_{i-1}, \ldots, u_0)$

Building the transform blocks

Single block

$$\mathbf{Z} = F(\mathbf{U})$$

$$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u)\,|\det J_F(u)|^{-1}$$

With

$Z = (z_1, \dots z_i)$

$U = (u_1, \dots u_i)$

The determinant of the Jacobian is the bottleneck
→ Make it **triangular**

Each z_i is a function of the previous entries only
$z_i = f(u_i, u_{i-1}, \dots, u_0)$

In general : $z_i = \theta_1 \times u_i + \theta_2$
where
$(\theta_1, \theta_2) = \mathbf{\Theta}(\mathbf{u_{i-1}}, \dots, \mathbf{u_0})$

Building the transform blocks

Single block

$$\mathbf{Z} = F(\mathbf{U})$$

$$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u) \,|\, \det J_F(u) \,|^{-1}$$

With

$Z = (z_1, \ldots z_i)$

$U = (u_1, \ldots u_i)$

The determinant of the Jacobian is the bottleneck
→ Make it **triangular**

Each z_i is a function of the previous entries only
$z_i = f(u_i, u_{i-1}, \ldots, u_0)$

In general : $z_i = \theta_1 \times u_i + \theta_2$
where
$(\theta_1, \theta_2) = \mathbf{\Theta}(\mathbf{u_{i-1}}, \ldots, \mathbf{u_0})$

$\Theta(u_{i-1}, \ldots, u_0)$ is the neural network we train

Building the transform blocks

Single block

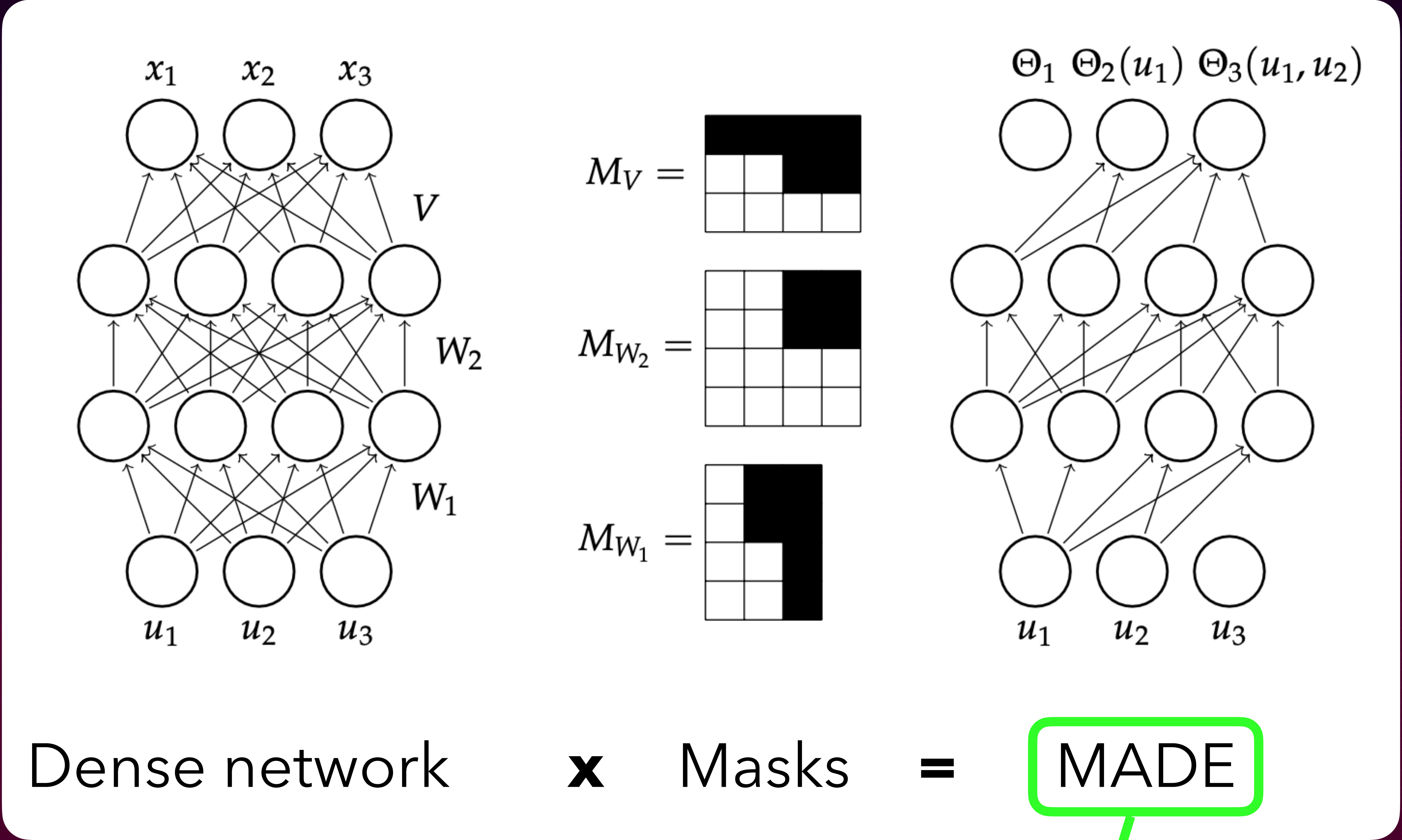
$\mathbf{Z} = F(\mathbf{U})$

$p_{\mathbf{Z}}(z) = p_{\mathbf{U}}(u)\,|\det J_F(u)|^{-1}$

With

$Z = (z_1, \ldots z_i)$

$U = (u_1, \ldots u_i)$



The determinant of the Jacobian is the bottleneck → Make it **triangular**

Each z_i is a function of the previous entries only $z_i = f(u_i, u_{i-1}, \ldots, u_0)$

In general : $z_i = \theta_1 \times u_i + \theta_2$ where $(\theta_1, \theta_2) = \mathbf{\Theta}(\mathbf{u_{i-1}}, \ldots, \mathbf{u_0})$

Masked Autoencoder for Density Estimation

$\Theta(u_{i-1}, \ldots, u_0)$ is the neural network we train

Two flavors of SBI

Single round for amortized inference

Multiple round for fast convergence

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

Multiple round for fast convergence

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

Multiple round for fast convergence

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Multiple round for fast convergence

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Few simulations for the training set (~ 5k)

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Few simulations for the training set (~ 5k)

↓

Training of the normalizing flow

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Few simulations for the training set (~ 5k)

↓

Training of the normalizing flow

↓

Posterior parameters for a single fine-tuned observation

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Few simulations for the training set (~ 5k)

↓

Training of the normalizing flow

↓

Posterior parameters for a single fine-tuned observation

Nb of rounds

Two flavors of SBI

Single round for amortized inference

Many simulations for the training set (~ 100k)

↓

Training of the normalizing flow

↓

Posterior parameters for multiple observations using the same network

Fast inference for multiple observations

Multiple round for fast convergence

Few simulations for the training set (~ 5k)

↓

Training of the normalizing flow

↓

Posterior parameters for a single fine-tuned observation

Nb of rounds

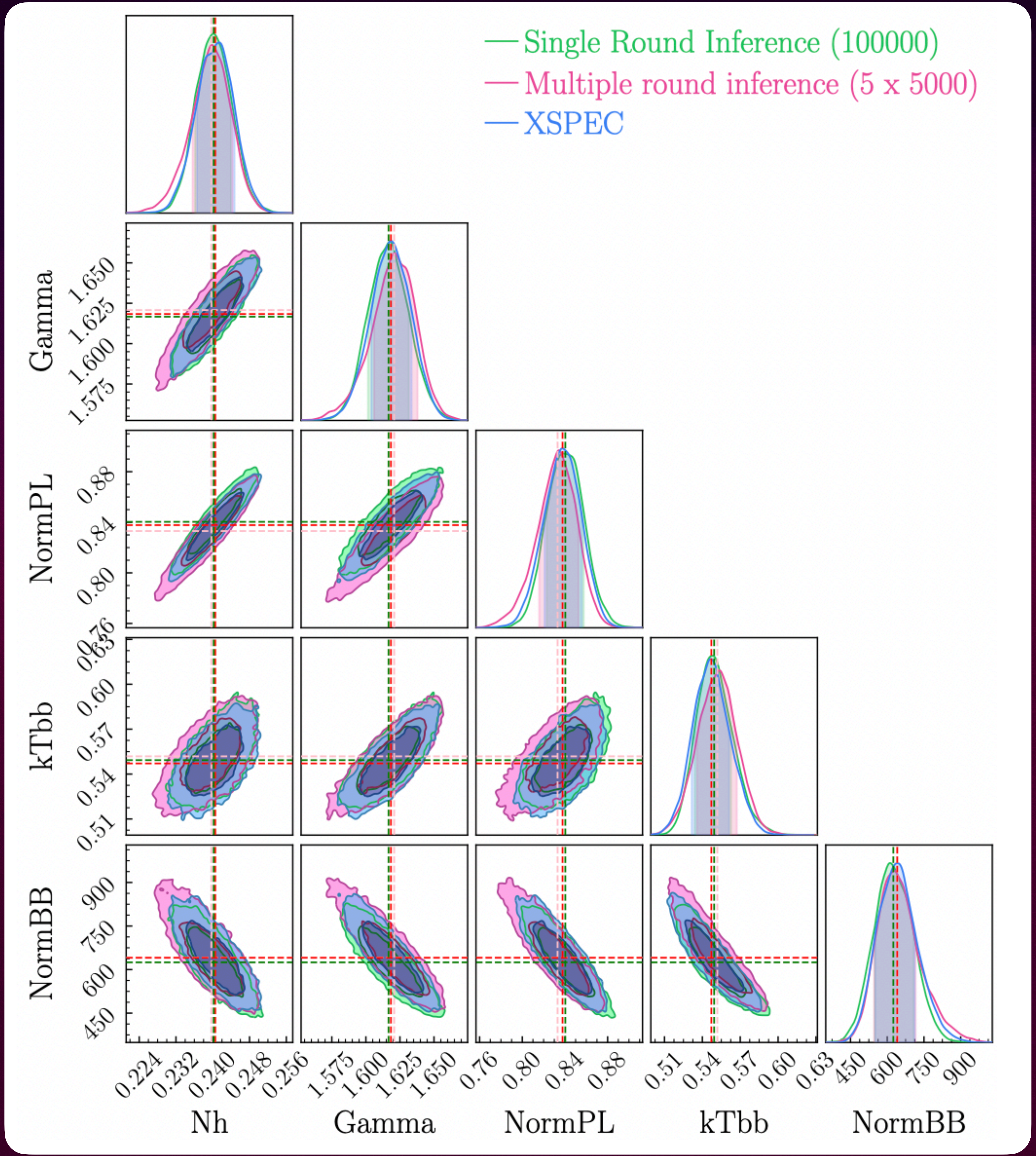
Fast inference for single observation

Comparison with Bayesian Inference

Direct comparison between SBI and traditional Bayesian Inference for a XMM-Newton source

Green and **Red** : two flavors of SBI
Blue : reference (MCMC)

SBI performs **similarly** as **MCMC** in X-ray spectroscopy while being much faster



What SBI is good at ?

What SBI is good at ?

- **Speeding up slow inference :** If a likelihood is available, it is straightforward to implement a simulator and use it. Example : any fit with XRISM data and a model with +5 parameters

What SBI is good at ?

- **Speeding up slow inference :** If a likelihood is available, it is straightforward to implement a simulator and use it. Example : any fit with XRISM data and a model with +5 parameters
- **Bulk inference :** You have numerous similar observables that you would want to fit at once. Example : **time-resolved spectroscopy**

What SBI is good at ?

- **Speeding up slow inference :** If a likelihood is available, it is straightforward to implement a simulator and use it. Example : any fit with XRISM data and a model with +5 parameters
- **Bulk inference :** You have numerous similar observables that you would want to fit at once. Example : **time-resolved spectroscopy**
- **Automatic marginalization :** You have nuisance parameters or extra noise but analytical marginalization in unfeasible. Example : **calibration uncertainties**

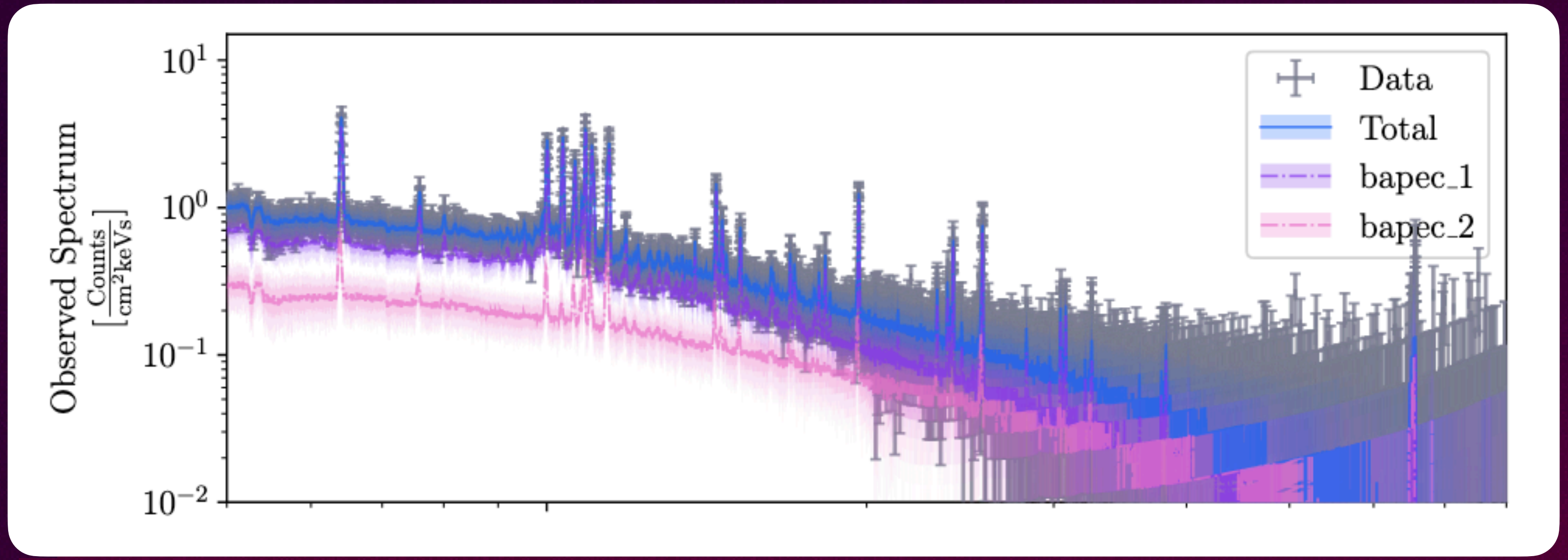
What SBI is good at ?

- **Speeding up slow inference :** If a likelihood is available, it is straightforward to implement a simulator and use it. Example : any fit with XRISM data and a model with +5 parameters
- **Bulk inference :** You have numerous similar observables that you would want to fit at once. Example : **time-resolved spectroscopy**
- **Automatic marginalization :** You have nuisance parameters or extra noise but analytical marginalization in unfeasible. Example : **calibration uncertainties**
- **Likelihood free inference:** The maths are too hard and you can't derive a satisfactory likelihood for your observable Example : **compressed representation**

Most important thing for SBI users

Look for meaningful representation of your observables (Feature Engineering)

X-IFU ~ 24k dimensions mapping a 10 parameter space

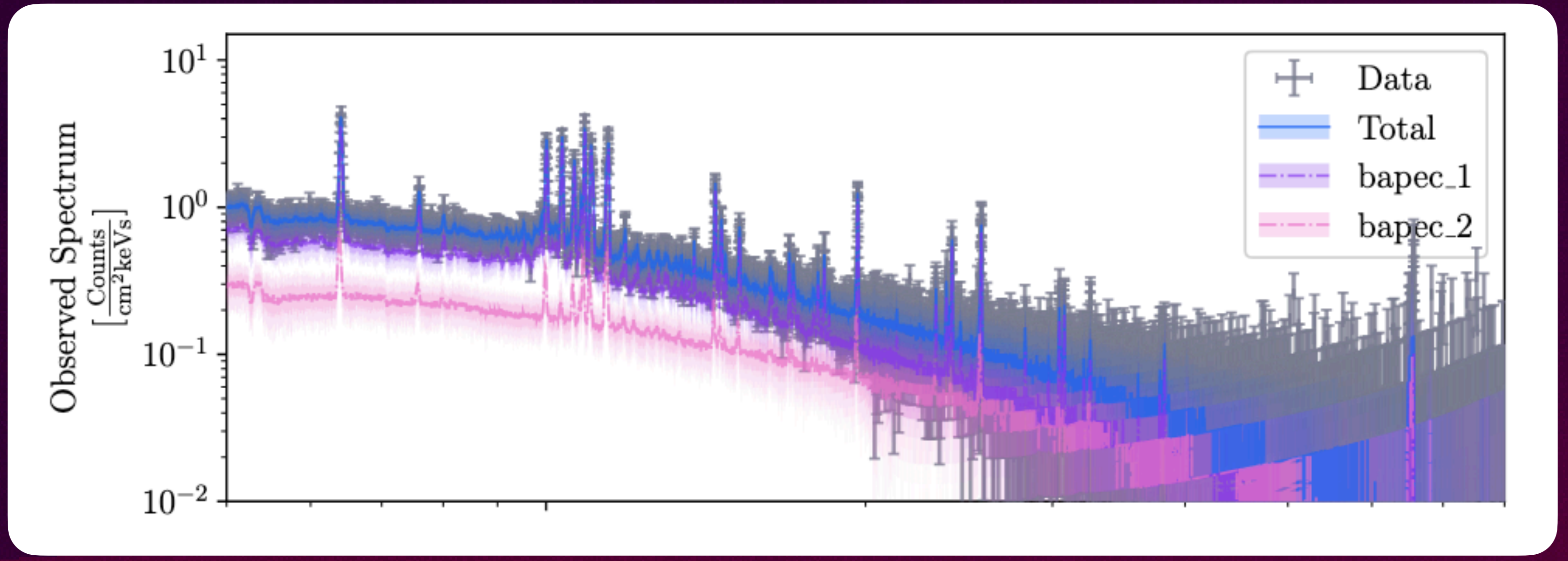


Reduce high dimension observables to small and weakly covariant statistics

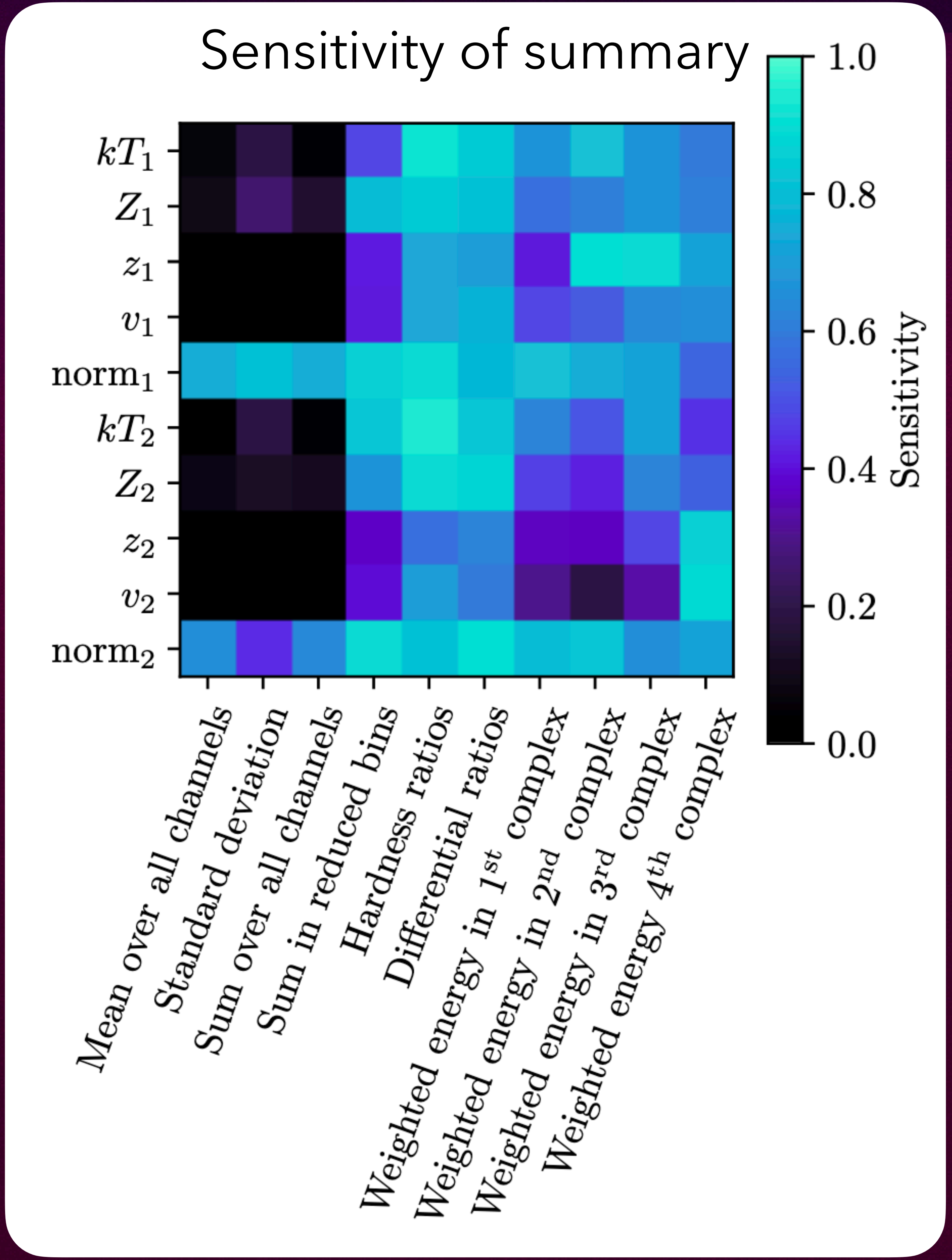
Most important thing for SBI users

Look for meaningful representation of your observables (Feature Engineering)

X-IFU ~ 24k dimensions mapping a 10 parameter space



Physically motivated & handcrafted statistics

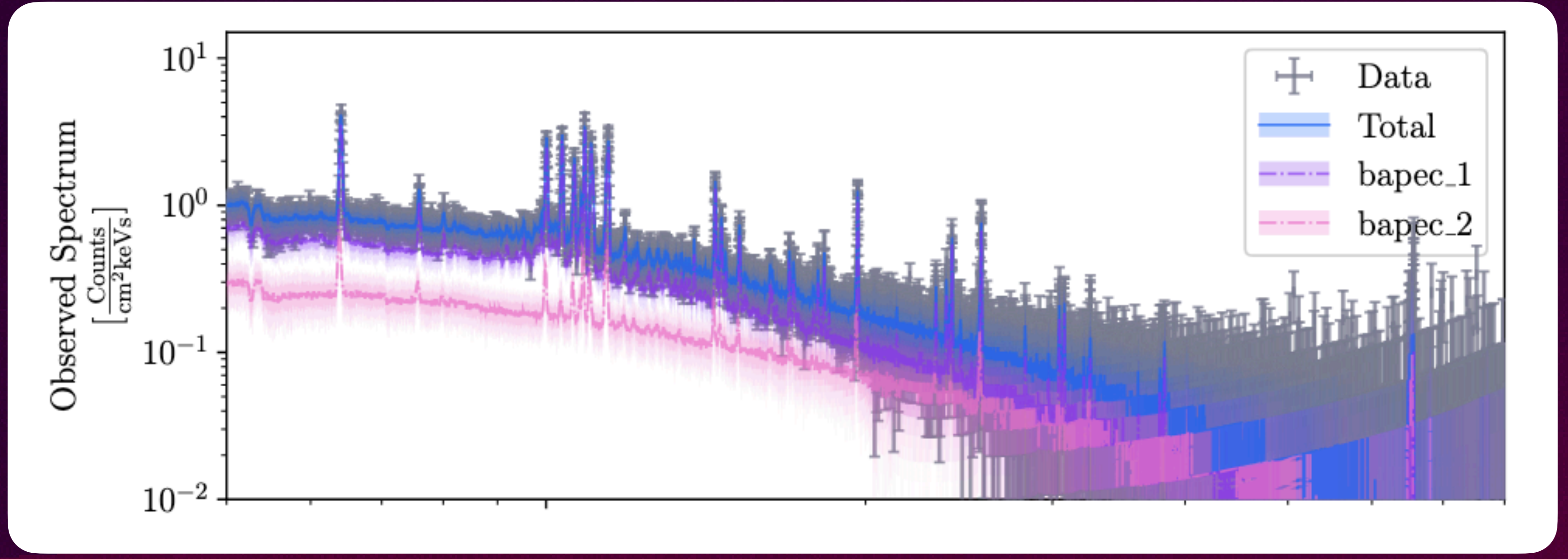


Reduce high dimension observables to small and weakly covariant statistics

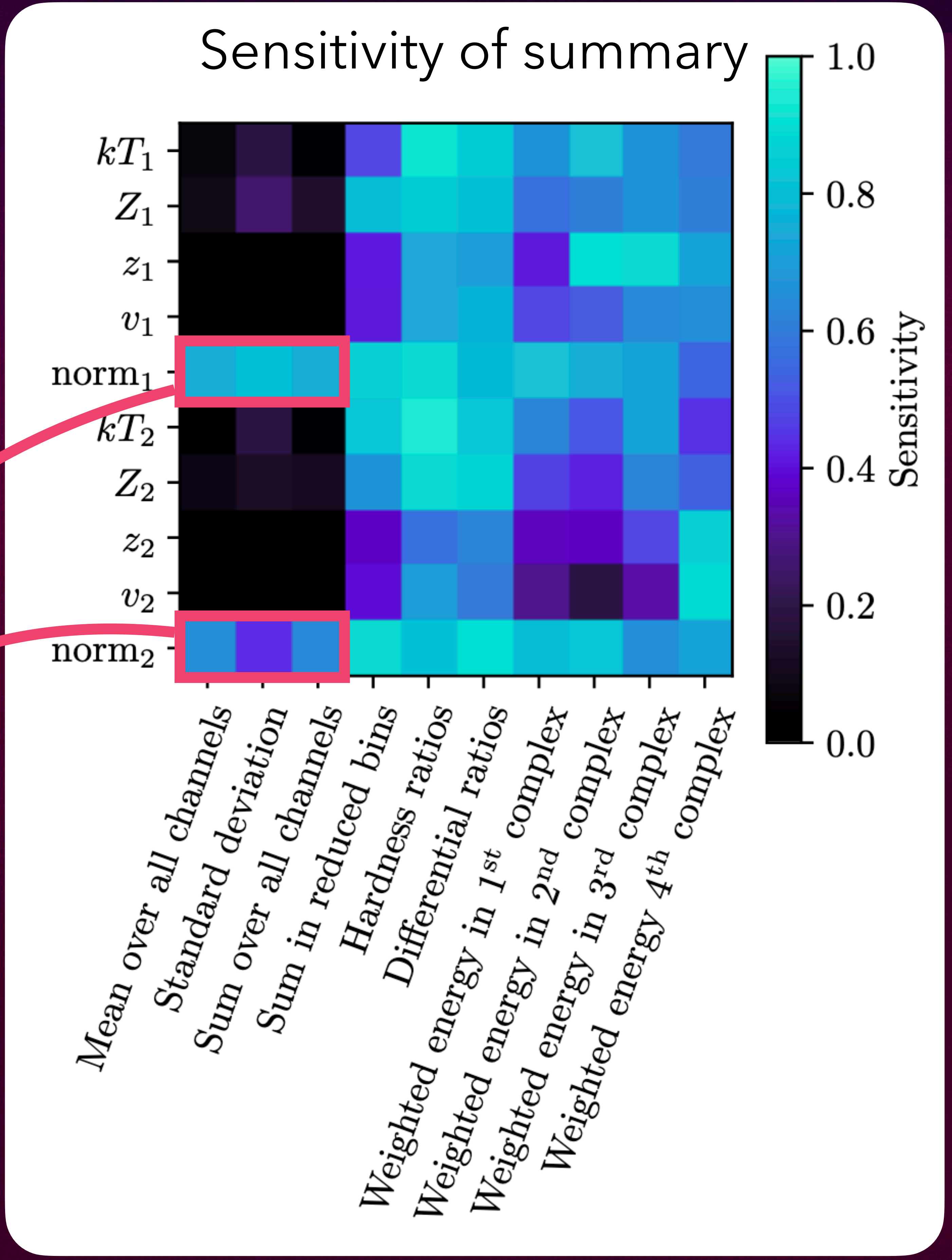
Most important thing for SBI users

Look for meaningful representation of your observables (Feature Engineering)

X-IFU ~ 24k dimensions mapping a 10 parameter space



Physically motivated & handcrafted statistics



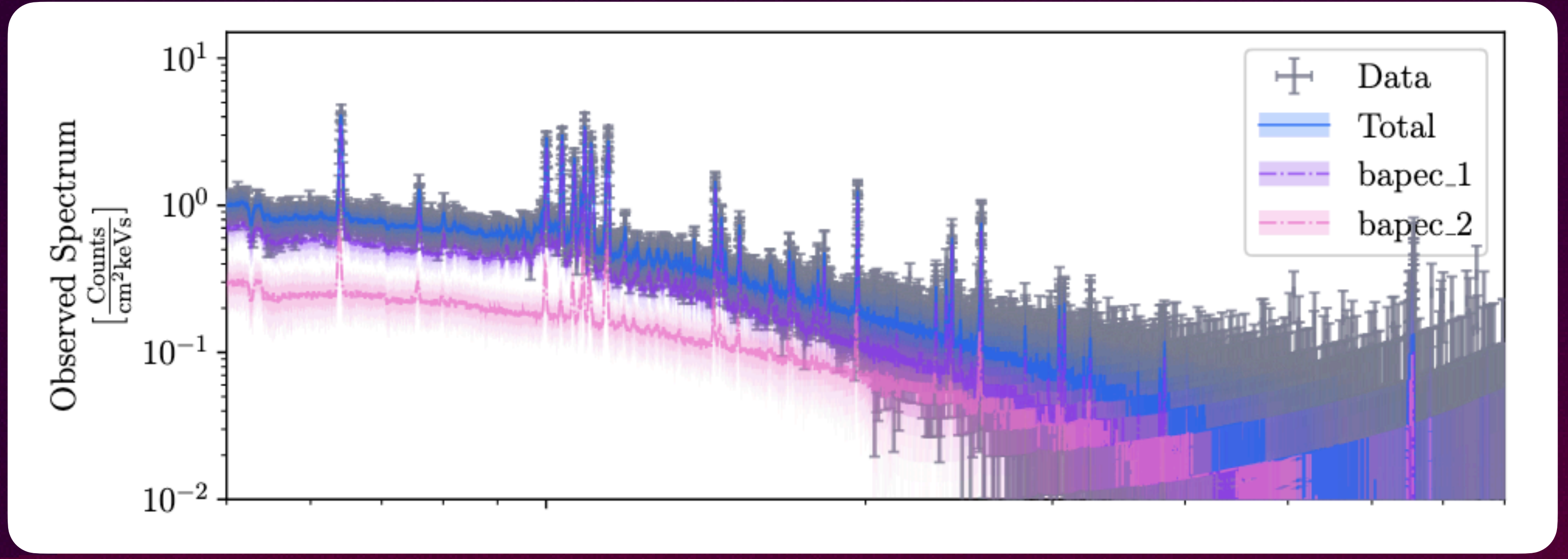
Reduce high dimension observables to small and weakly covariant statistics

Global summaries correlate with the total photon information

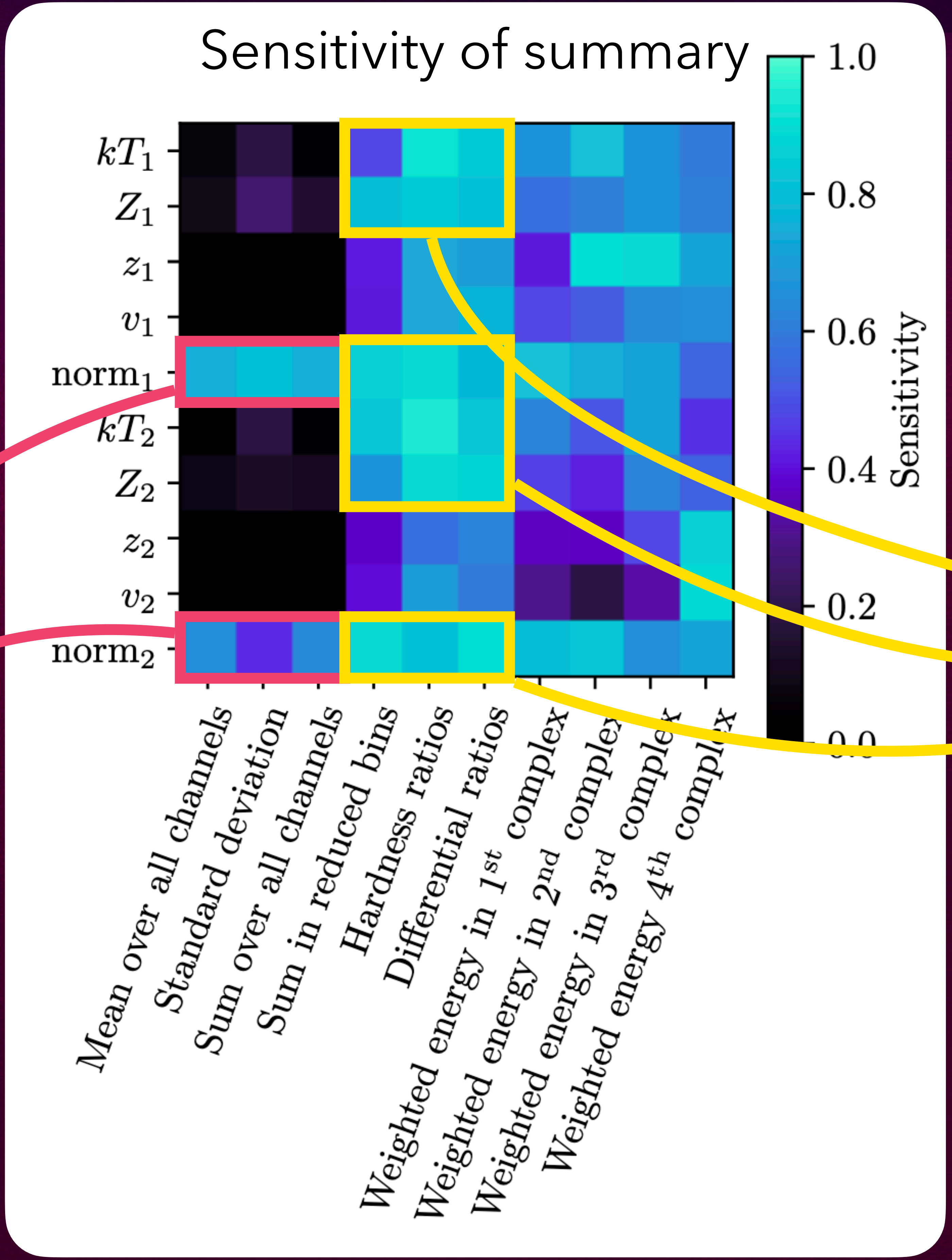
Most important thing for SBI users

Look for meaningful representation of your observables (Feature Engineering)

X-IFU ~ 24k dimensions mapping a 10 parameter space



Physically motivated & handcrafted statistics



Reduce high dimension observables to small and weakly covariant statistics

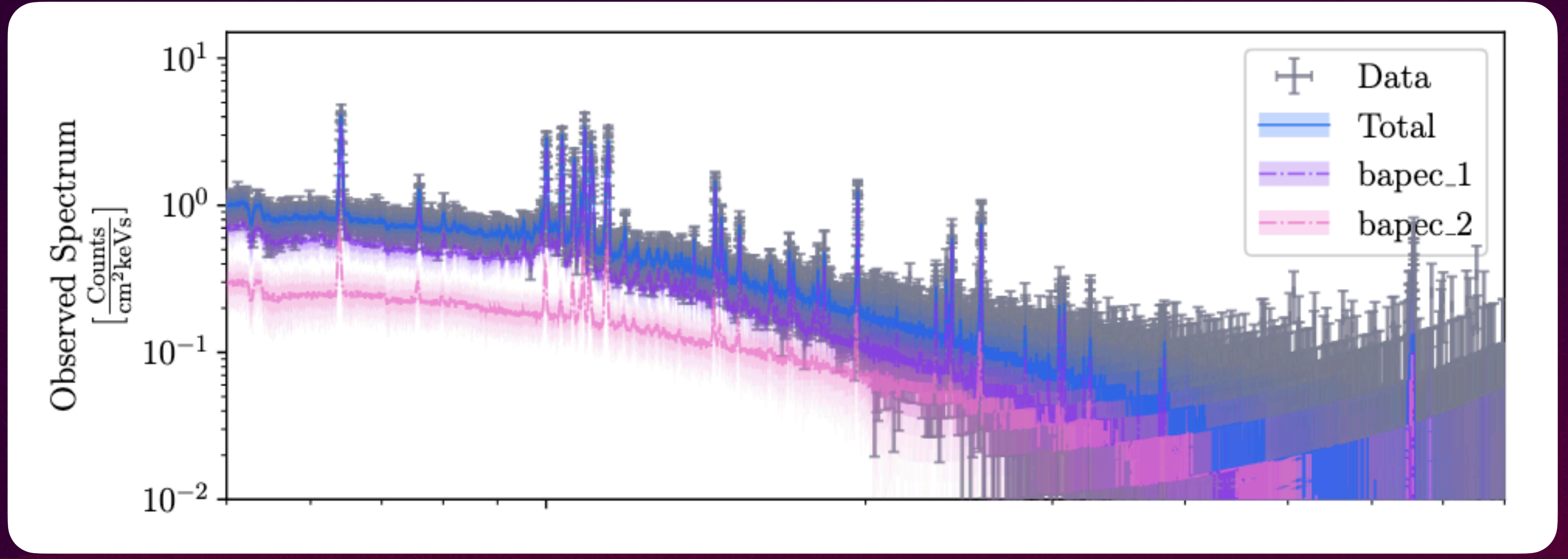
Global summaries correlate with the total photon information

Shape summaries correlate *mostly* with global shape parameters

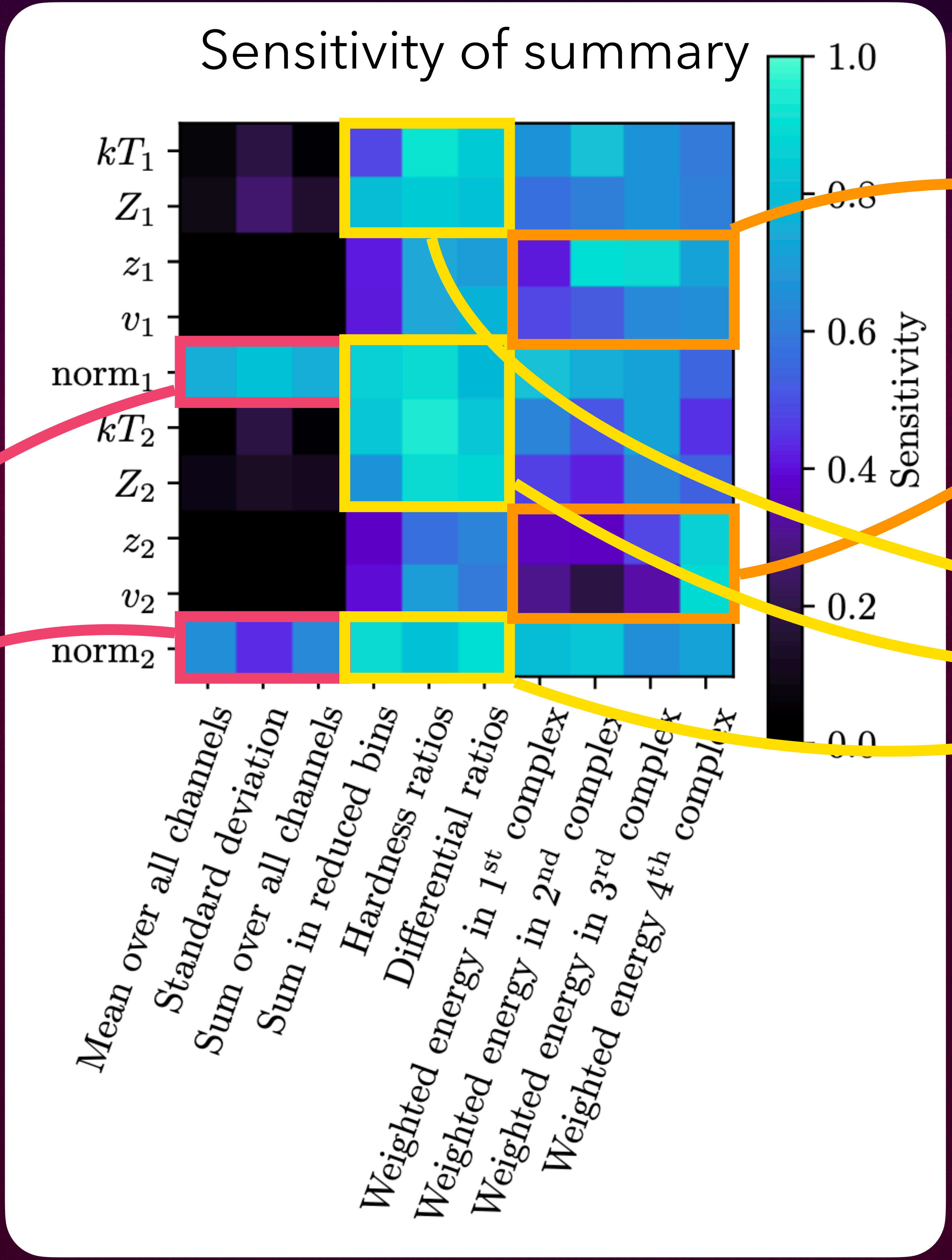
Most important thing for SBI users

Look for meaningful representation of your observables (Feature Engineering)

X-IFU ~ 24k dimensions mapping a 10 parameter space



Physically motivated & handcrafted statistics



Line parameters correlate with the motion and composition of the plasma

Shape summaries correlate *mostly* with global shape parameters

Reduce high dimension observables to small and weakly covariant statistics

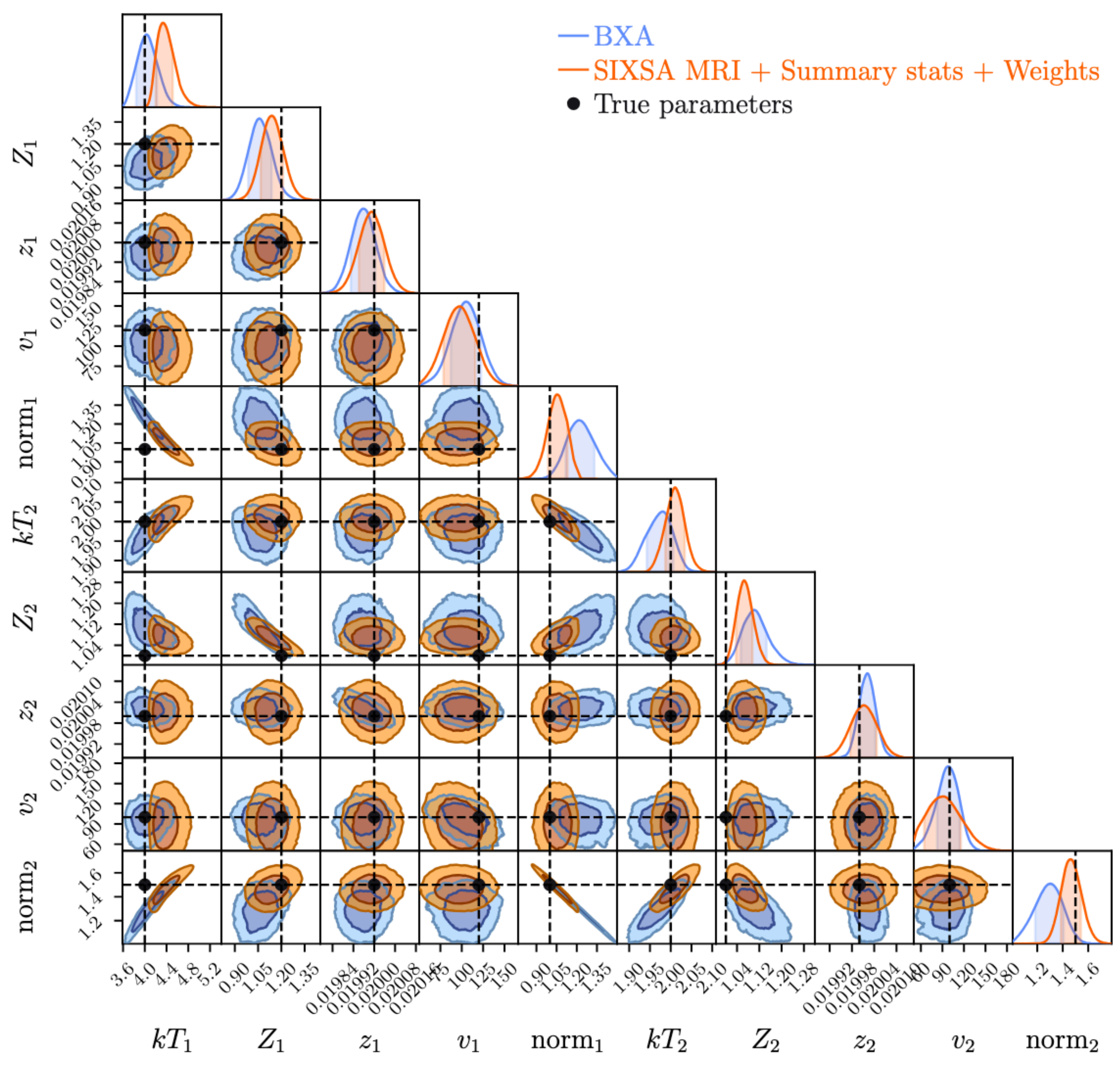
Global summaries correlate with the total photon information

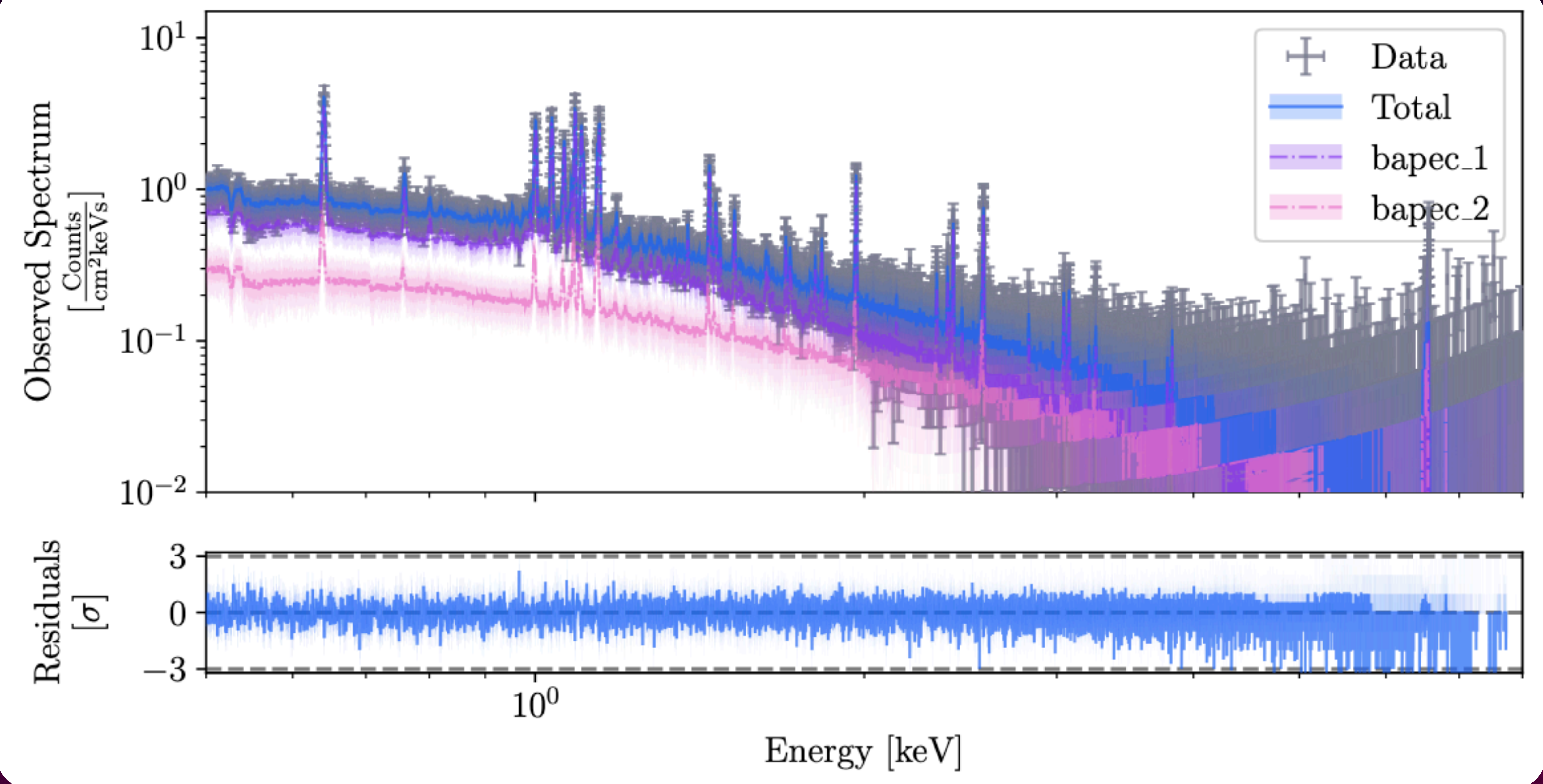
(a) O VIII

(b) Fe XXIV + Ne X

(c) Fe XXIV + Mg XII

(d) Fe XXV

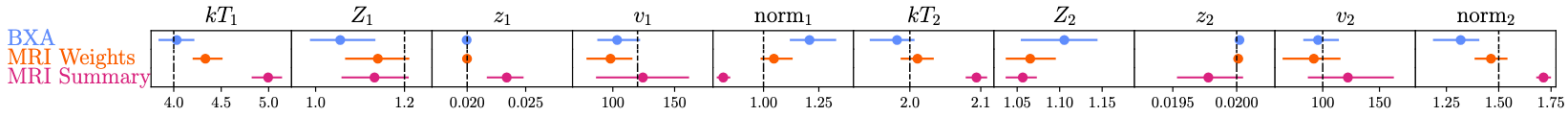




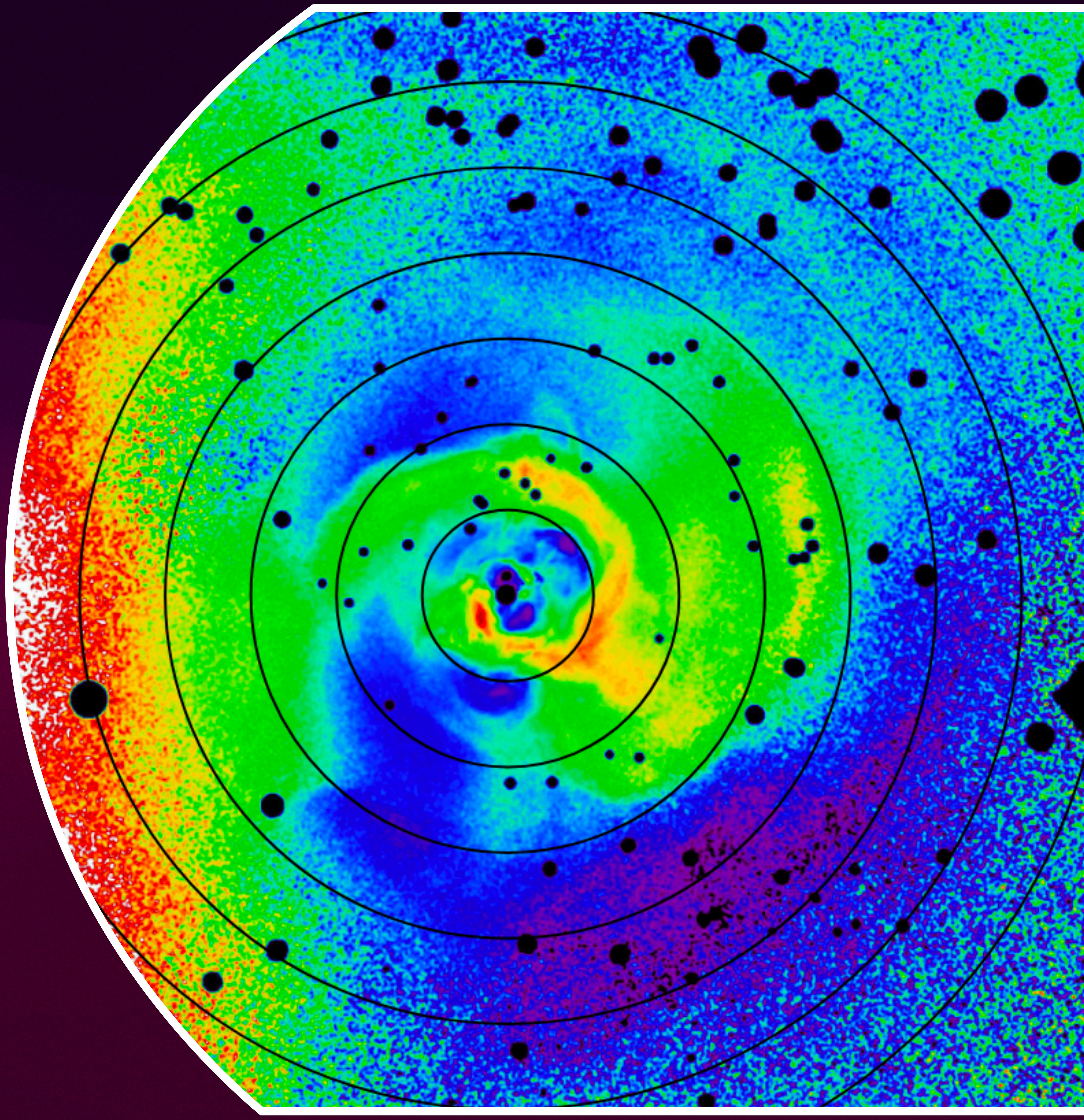
There is room for improvement

- Improve the compression
- Use the likelihood information

→ Check Didier's talk!



SBI and the dynamic assembly of galaxy clusters



Adapted from Zhuravleva & al. 2015

Galaxy clusters in a nutshell

Galaxy clusters in a nutshell

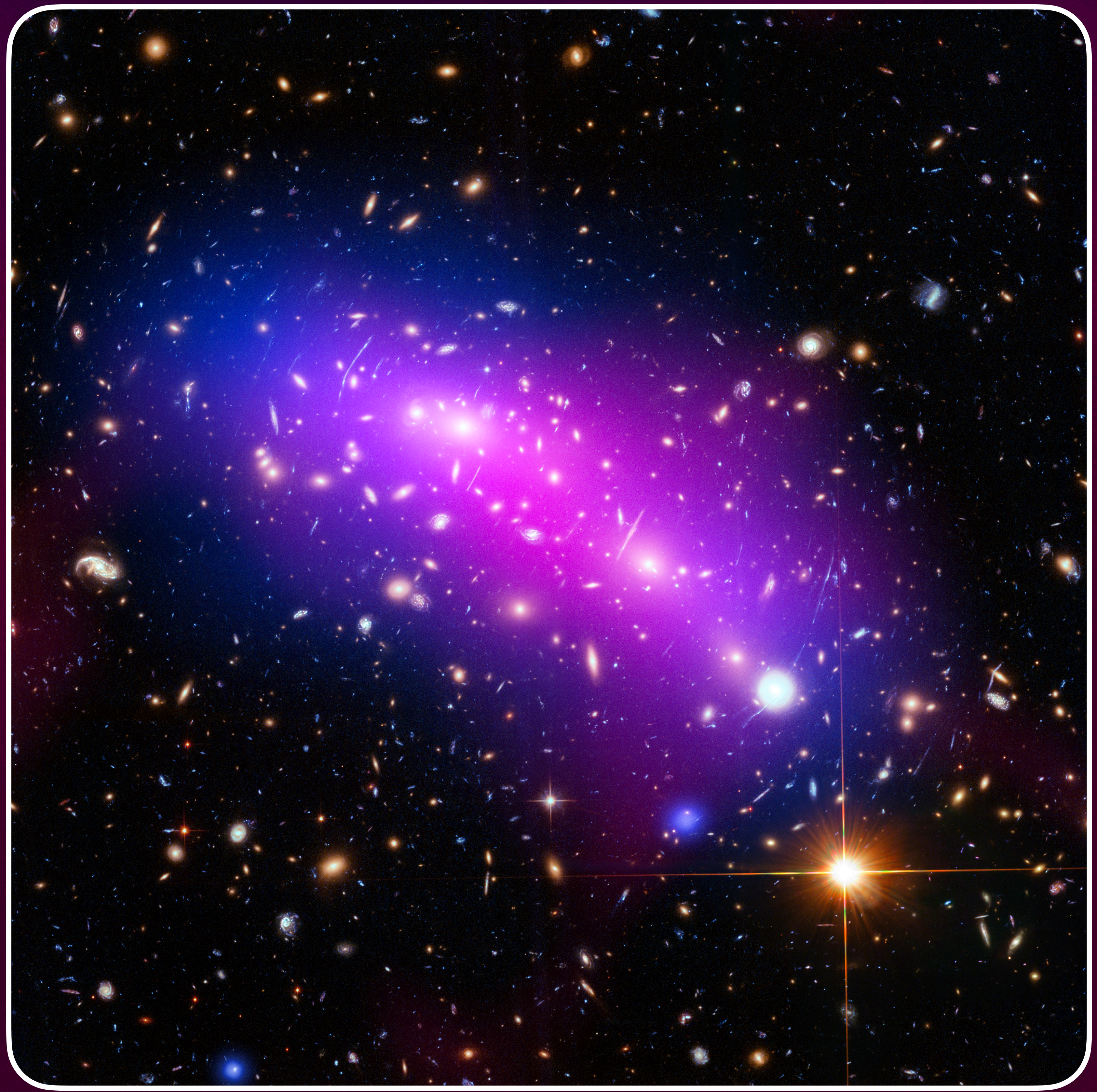
- **Largest gravitationally bound structures** in the Universe 📈

Galaxies only

Galaxy clusters in a nutshell

- **Largest gravitationally bound structures** in the Universe 📈
- **Galaxies** (1%), significant amount of **baryonic gas** (10%) and mostly **dark matter** (89%)

Galaxies only

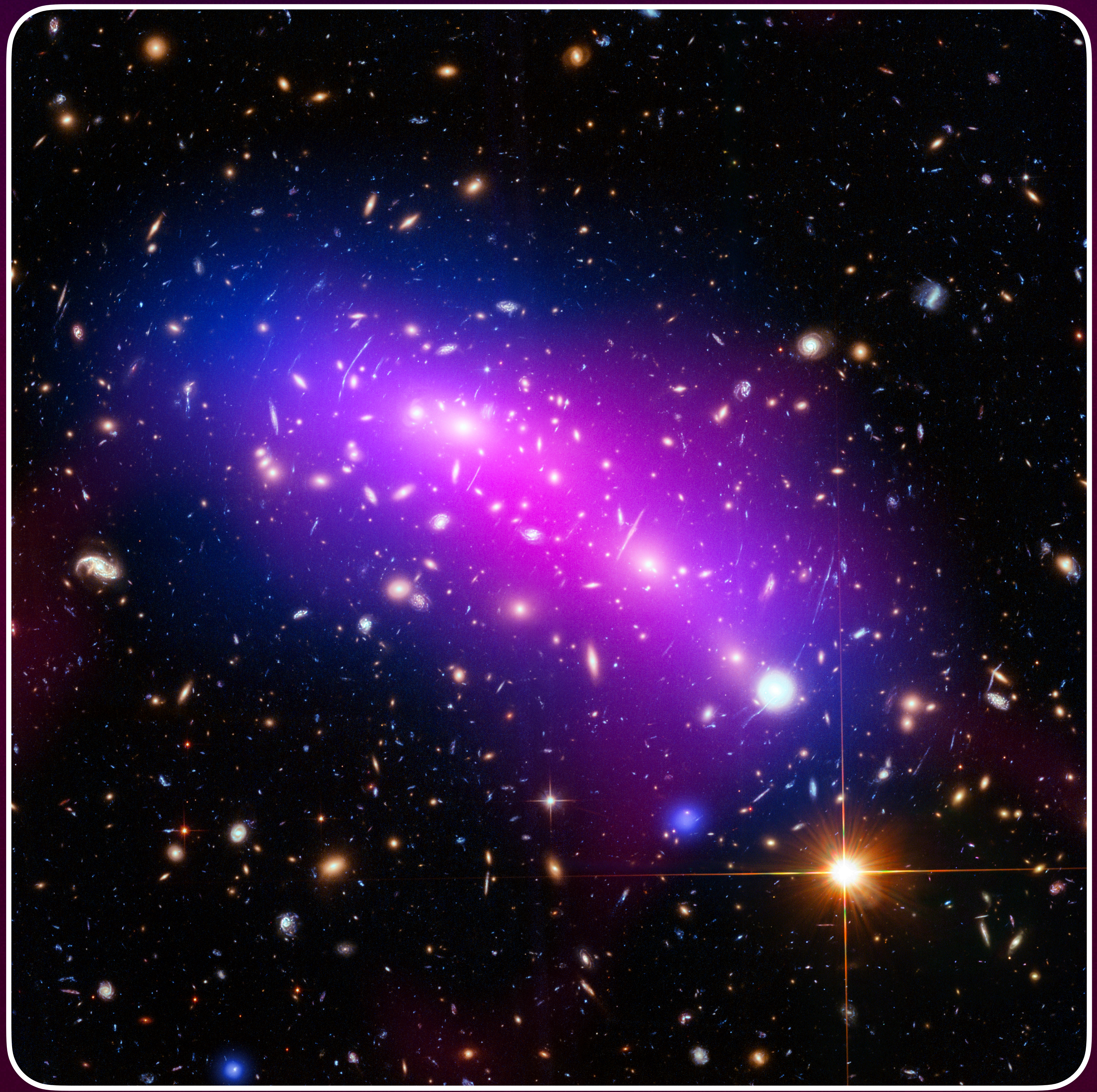


Galaxies + **gas** + **dark matter**

Galaxy clusters in a nutshell

- **Largest gravitationally bound structures** in the Universe 📈
- **Galaxies** (1%), significant amount of **baryonic gas** (10%) and mostly **dark matter** (89%)
- The **baryonic gas** deviates from hydrostatic equilibrium, probably due to **turbulent motion**
- Better understanding this motion is key to use galaxy clusters as **cosmological probes**

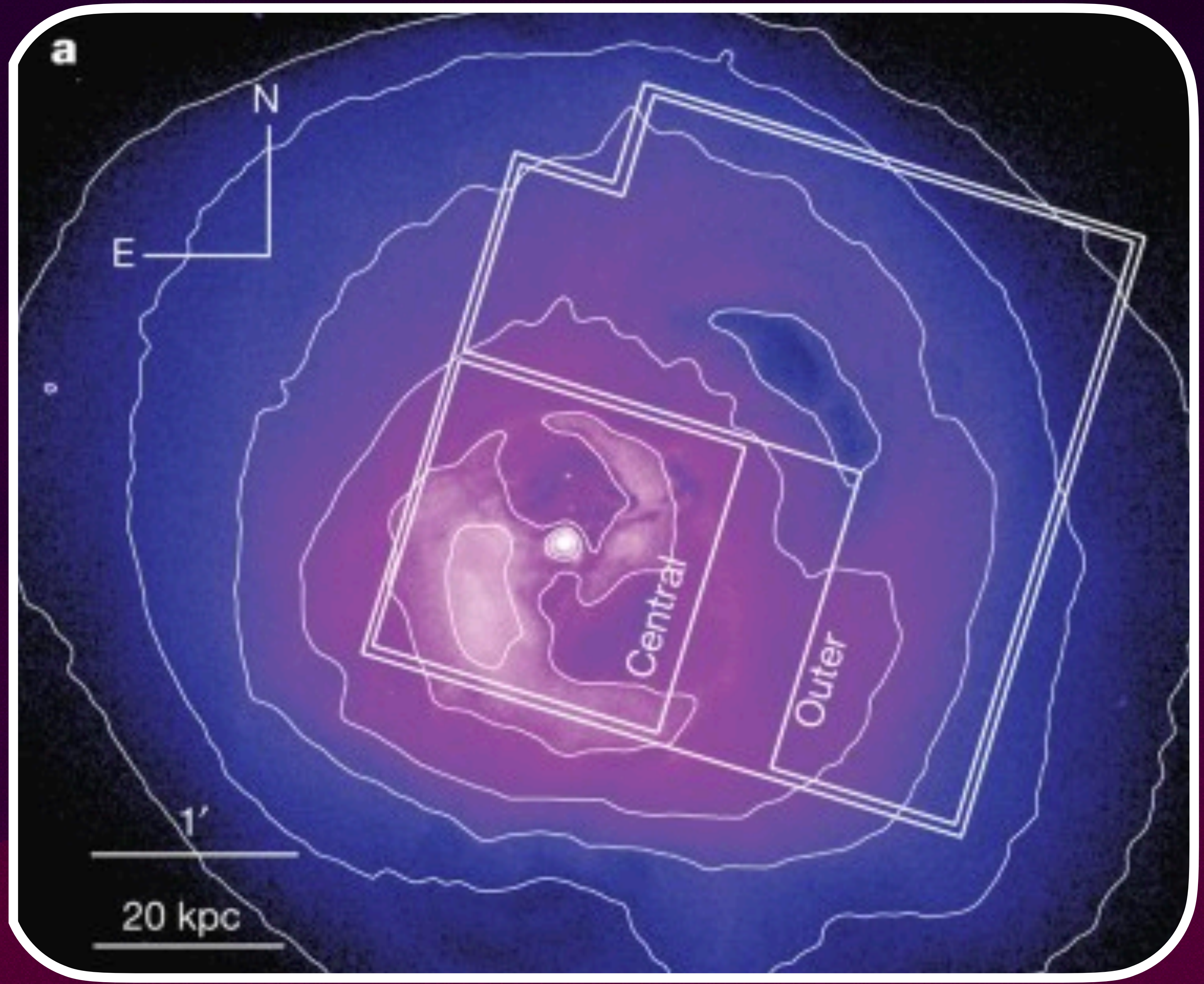
Galaxies only



Galaxies + **gas** + **dark matter**

Direct view

Adapted from Hitomi Collaboration (2016)

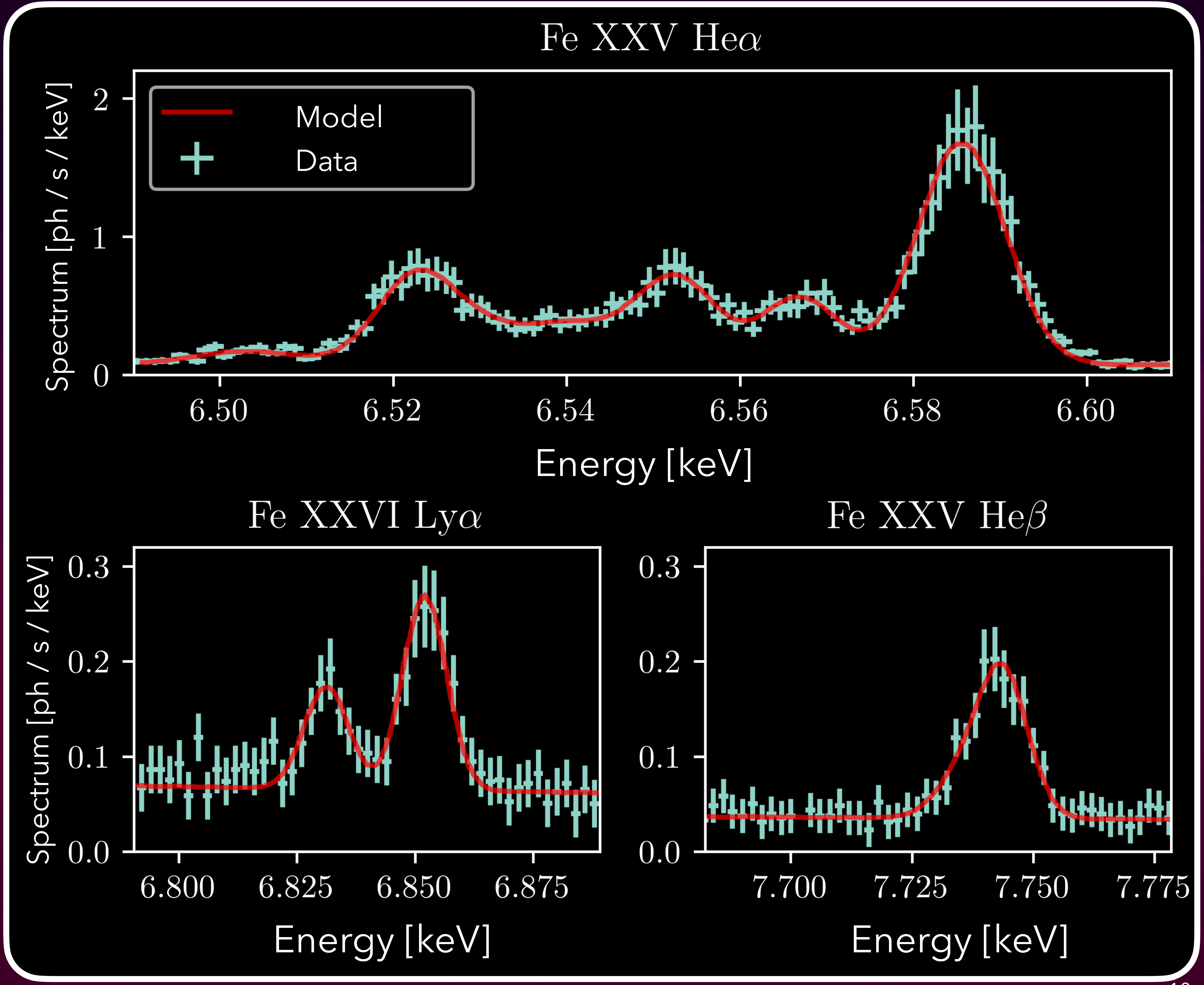


Gas motion has a direct effect on emission lines

Centroid shift ⇔ Bulk motion

Broadening ⇔ Integrated motion

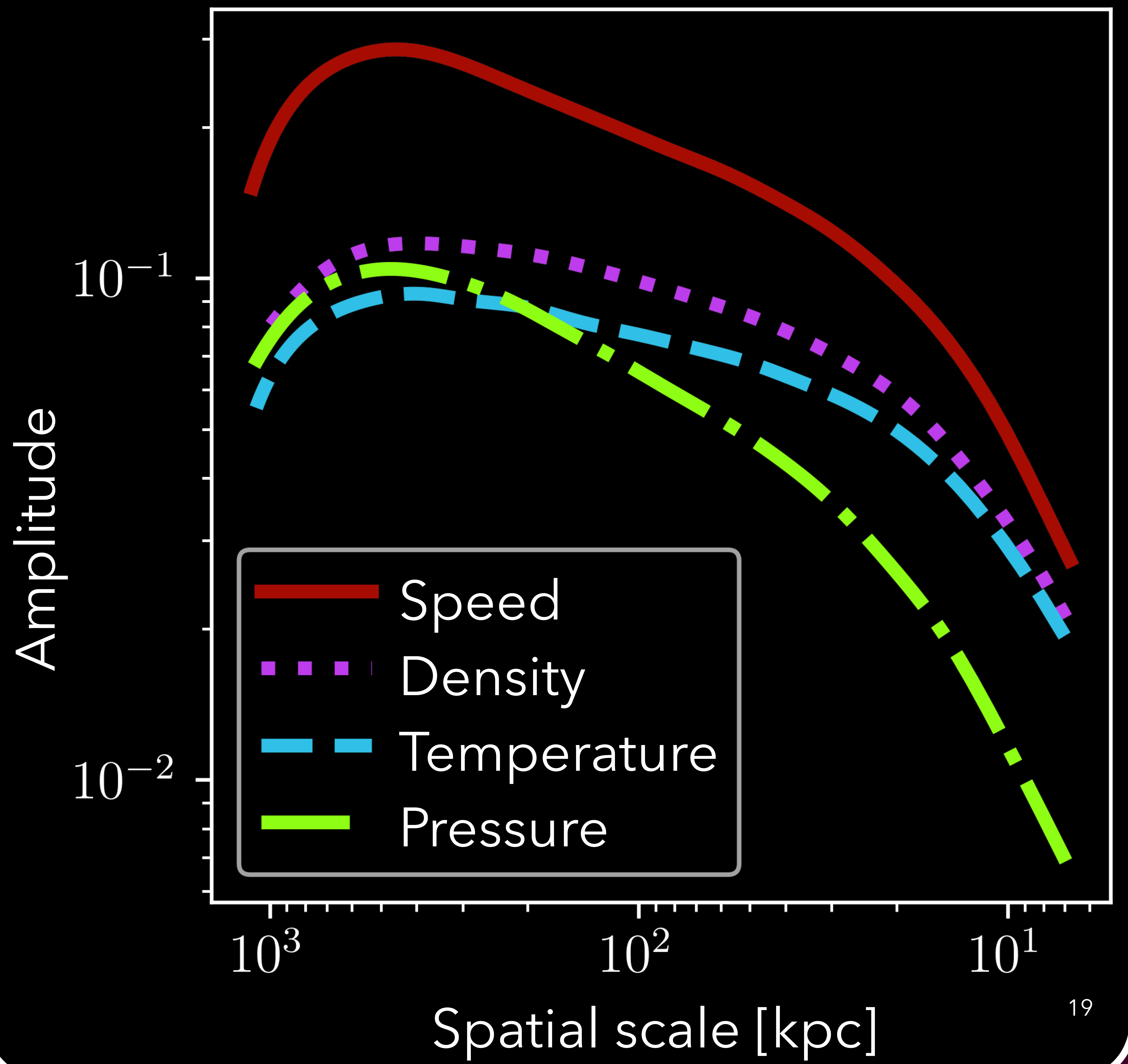
XRISM results in Dominique's talk



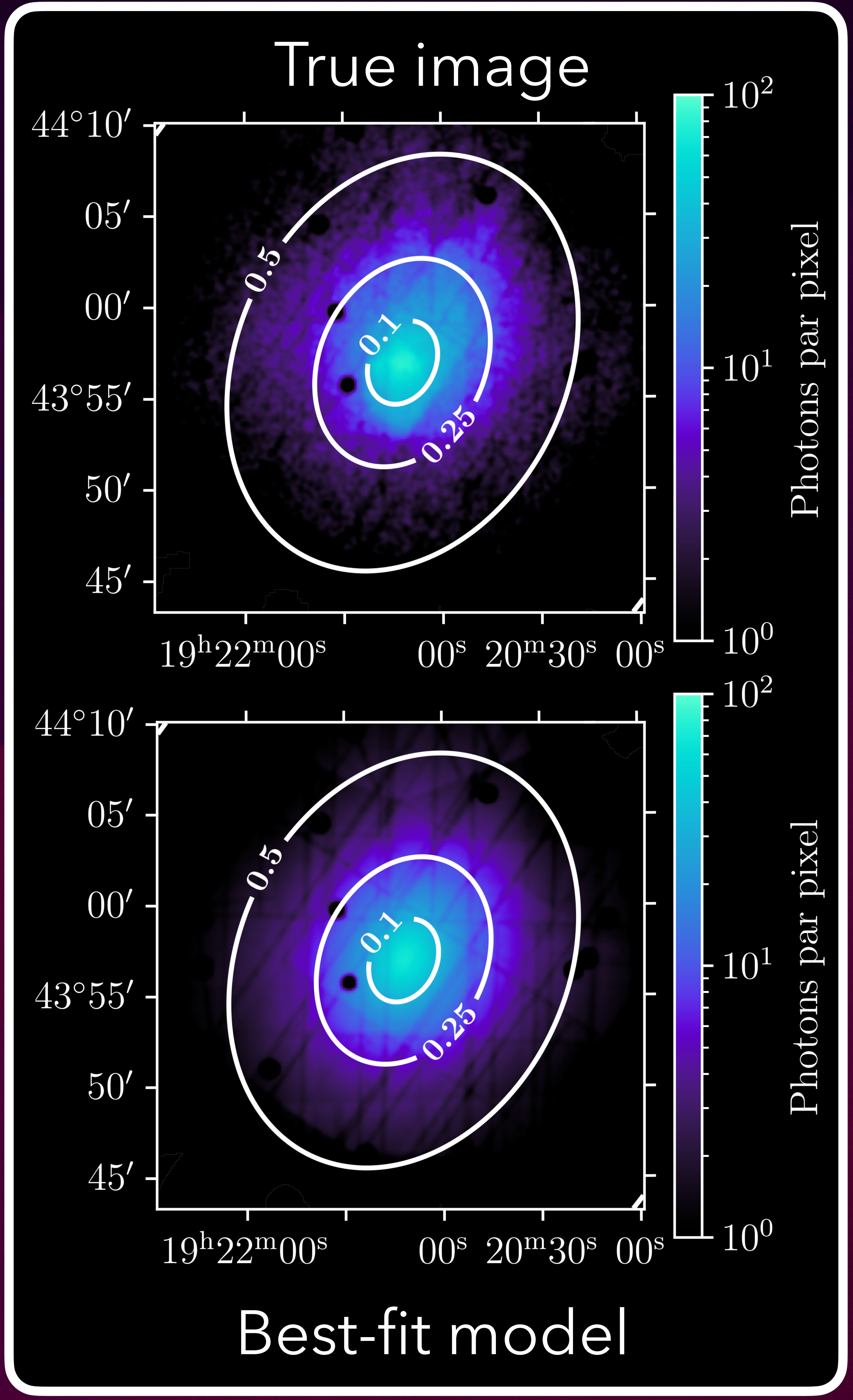
Indirect view

1. Gas motions induce thermodynamical fluctuations
2. Thermodynamical fluctuations translate in observable fluctuations (i.e. X-ray or SZ)
3. Correlations between the fluctuations and the gas motions are quantified with numerical simulations

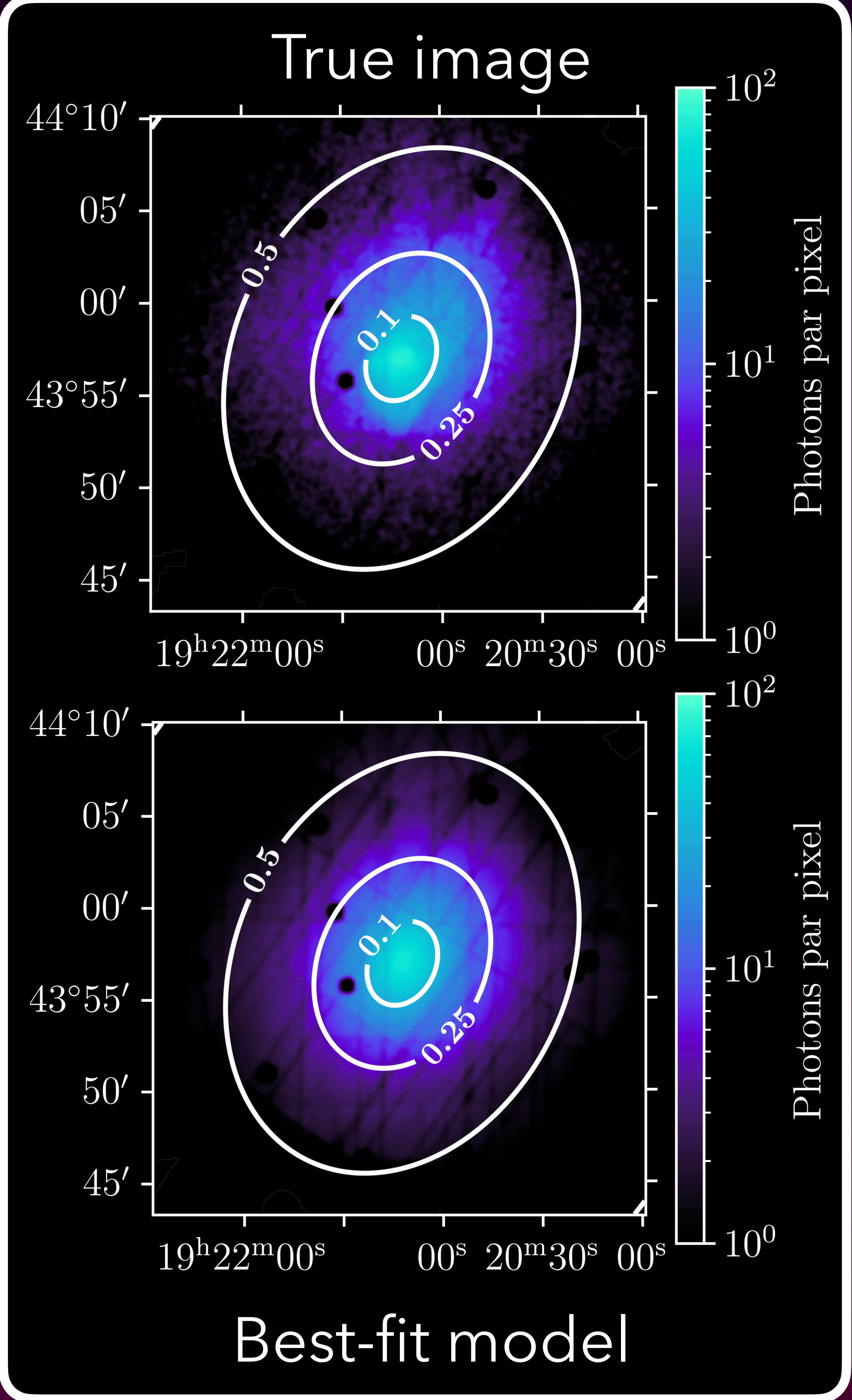
Adapted from Gaspari +2014

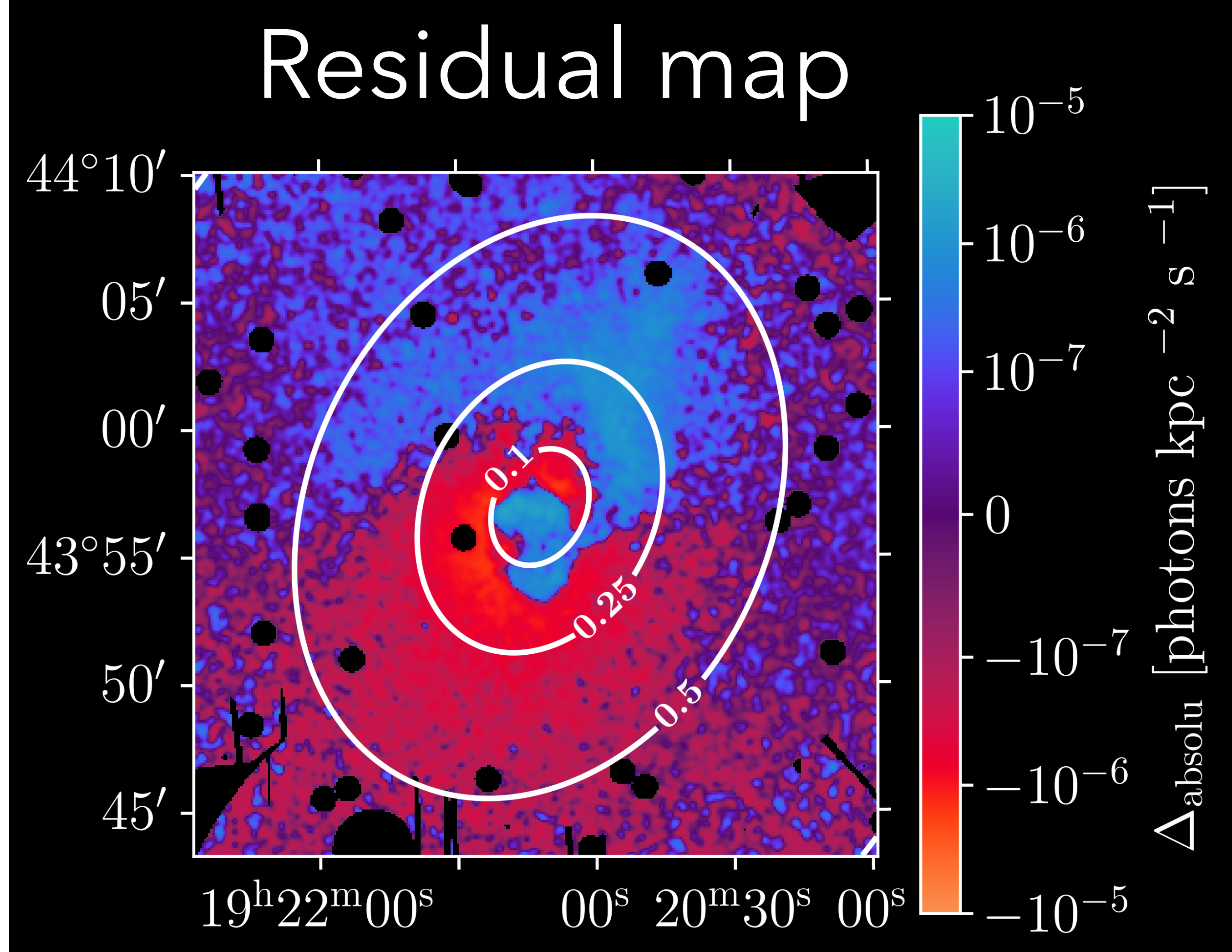


Probing the turbulent motion with fluctuations



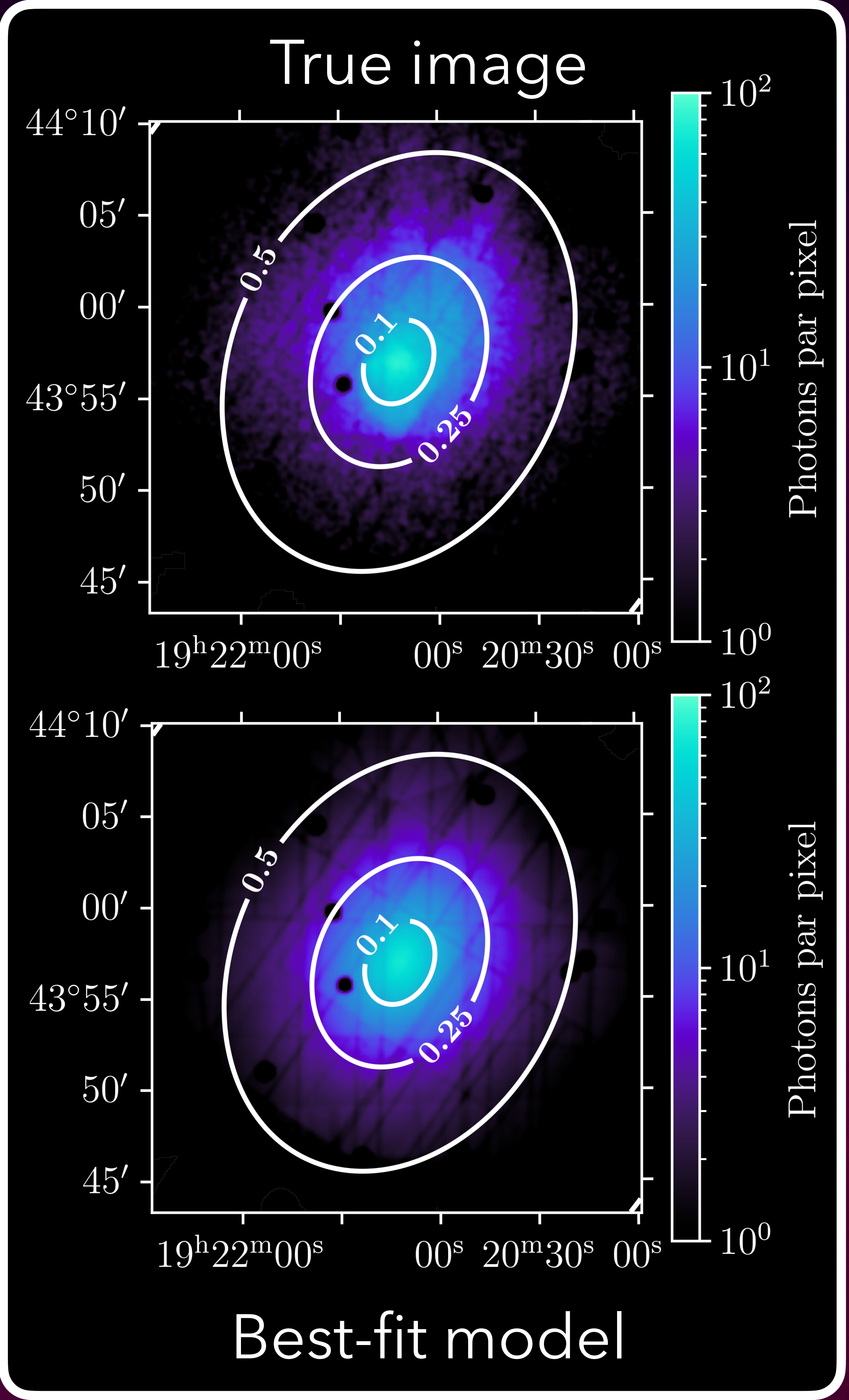
Probing the turbulent motion with fluctuations



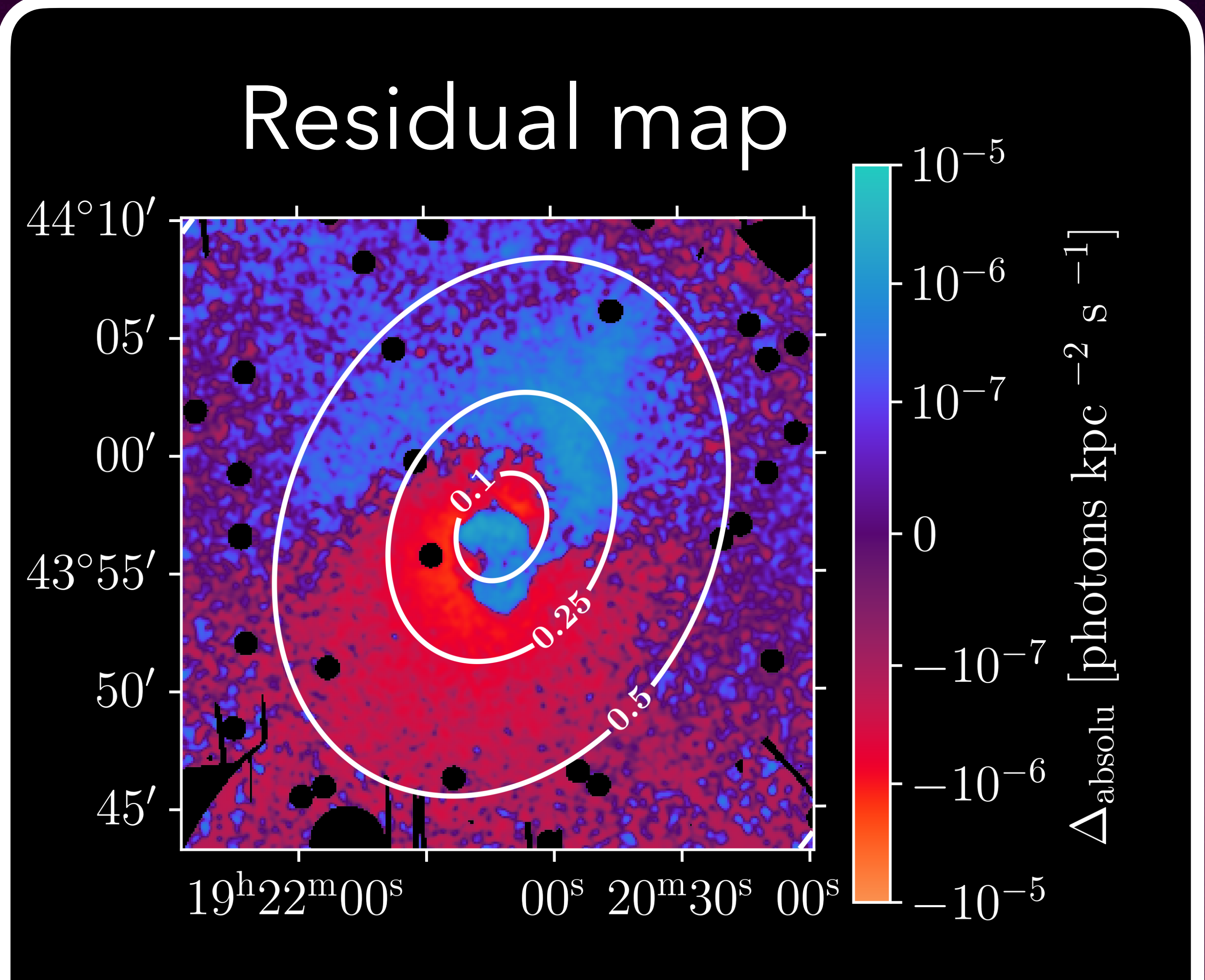


The structures in this map are linked to turbulence

Probing the turbulent motion with fluctuations

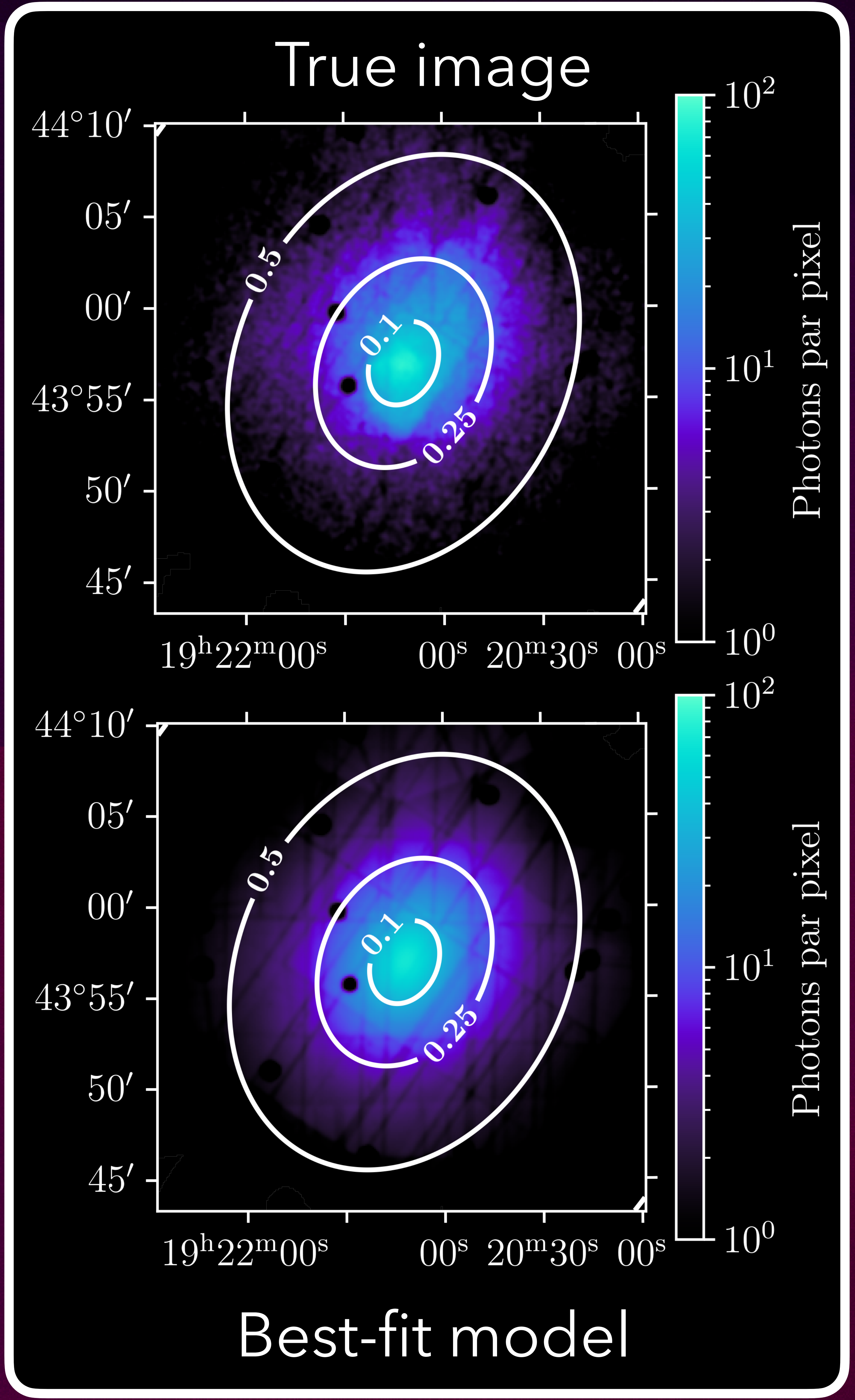


Assume that fluctuations are a GRF with Kolmogorov-like spectrum

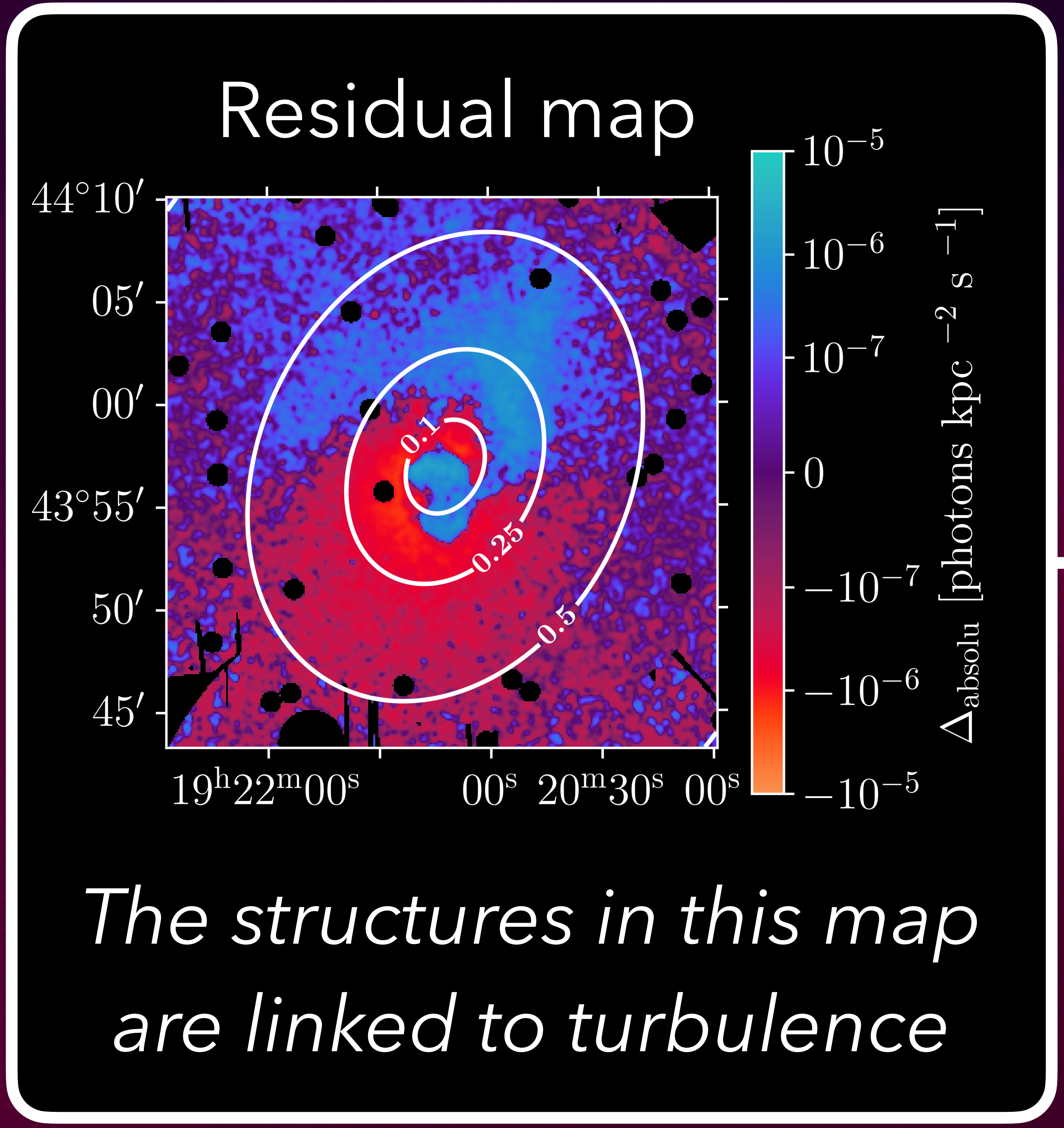


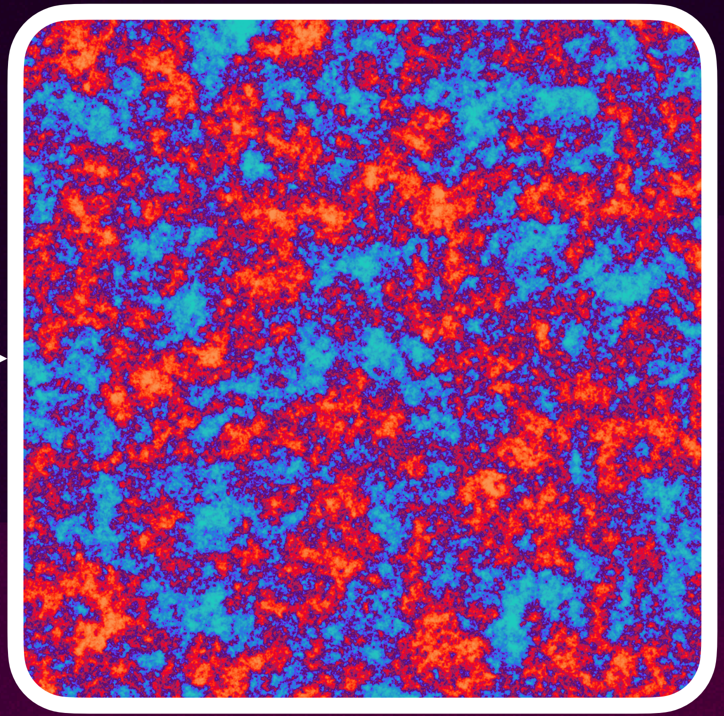
The structures in this map are linked to turbulence

Probing the turbulent motion with fluctuations



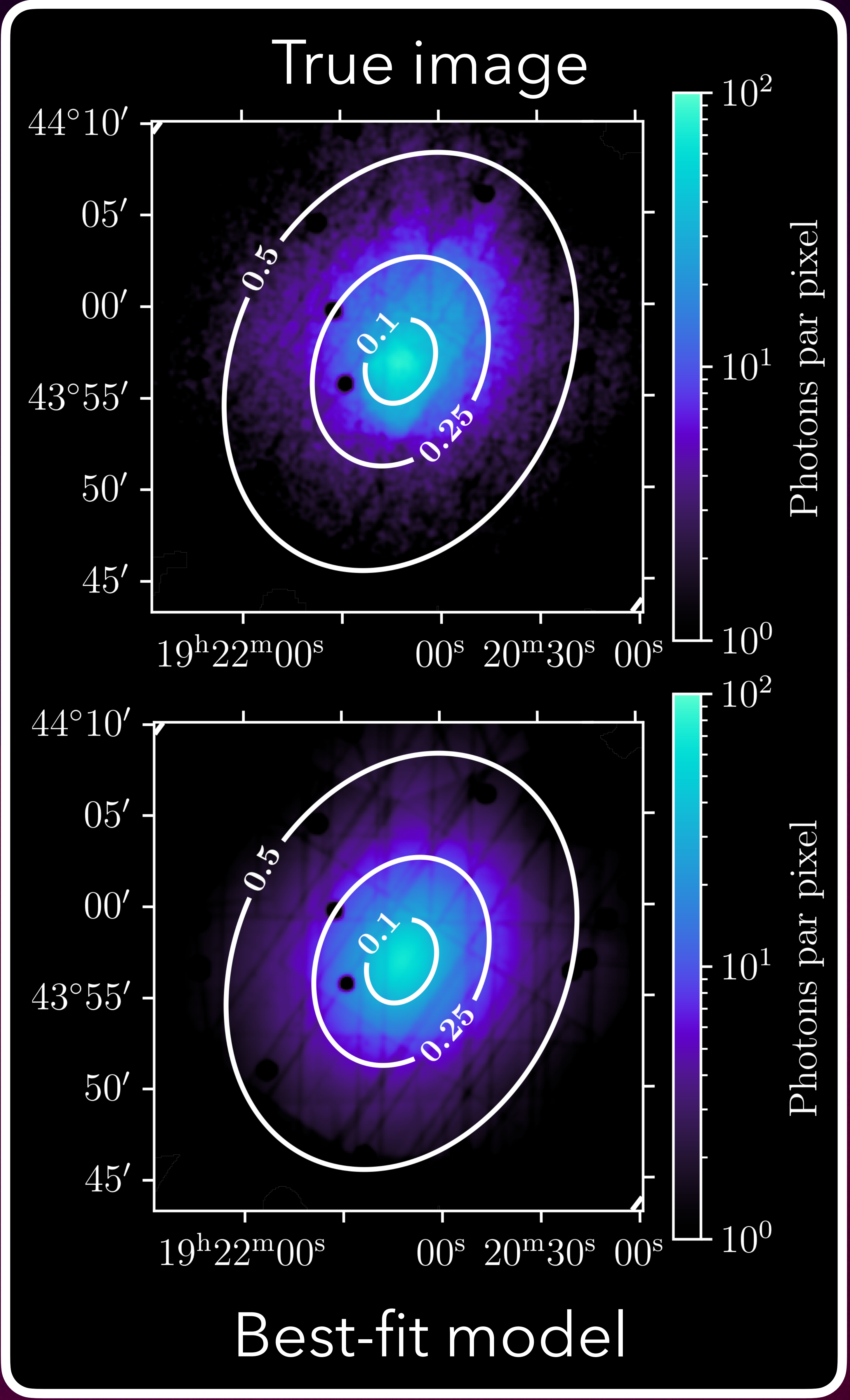
Assume that fluctuations are a GRF with Kolmogorov-like spectrum



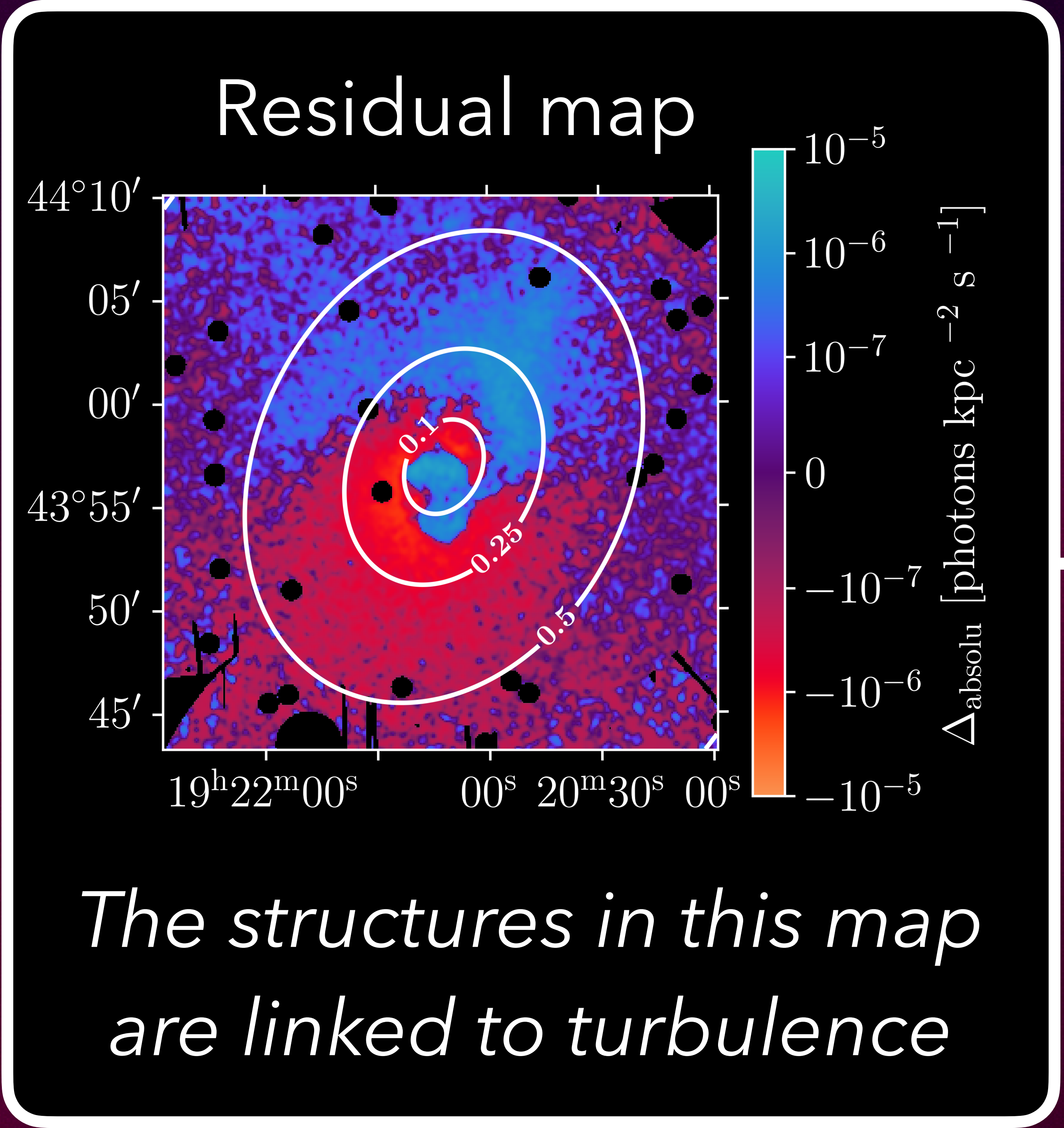


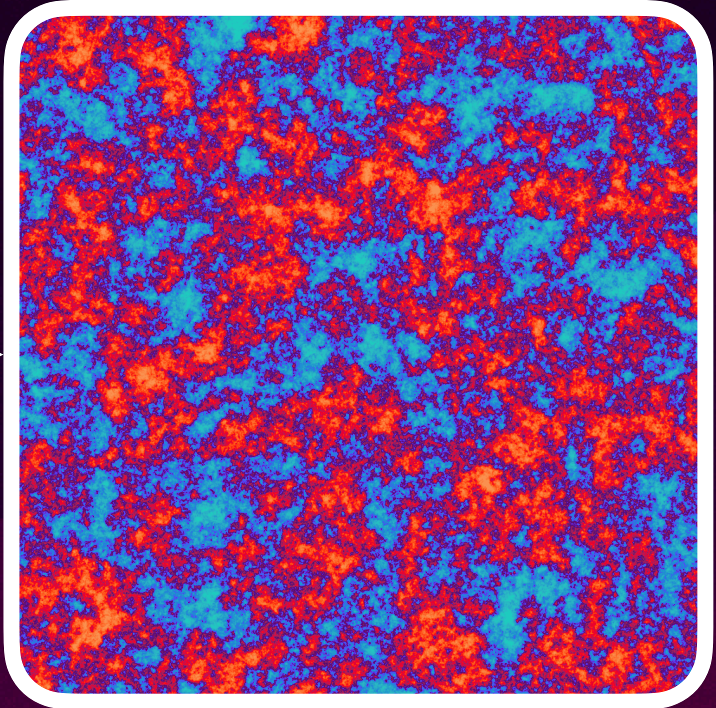
Mach number → hydrostatic bias

Probing the turbulent motion with fluctuations

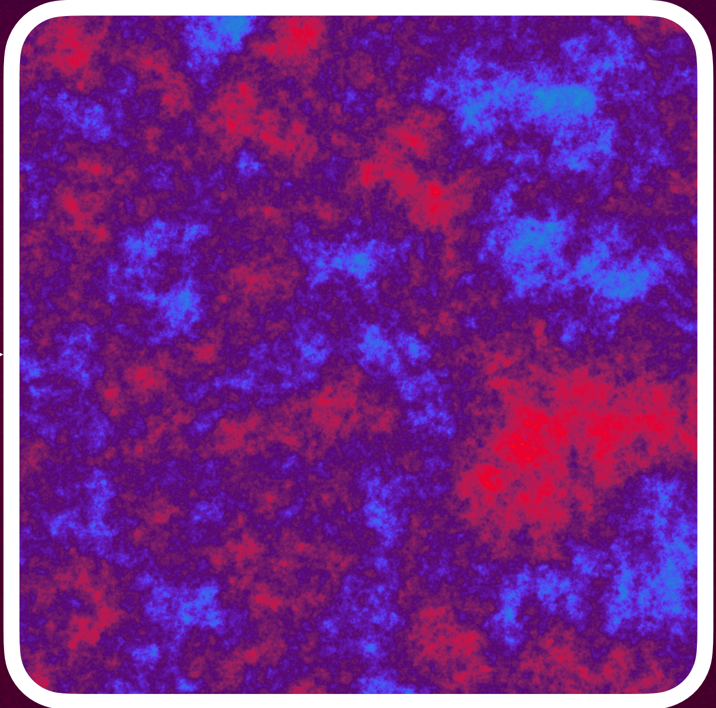


Assume that fluctuations are a GRF with Kolmogorov-like spectrum



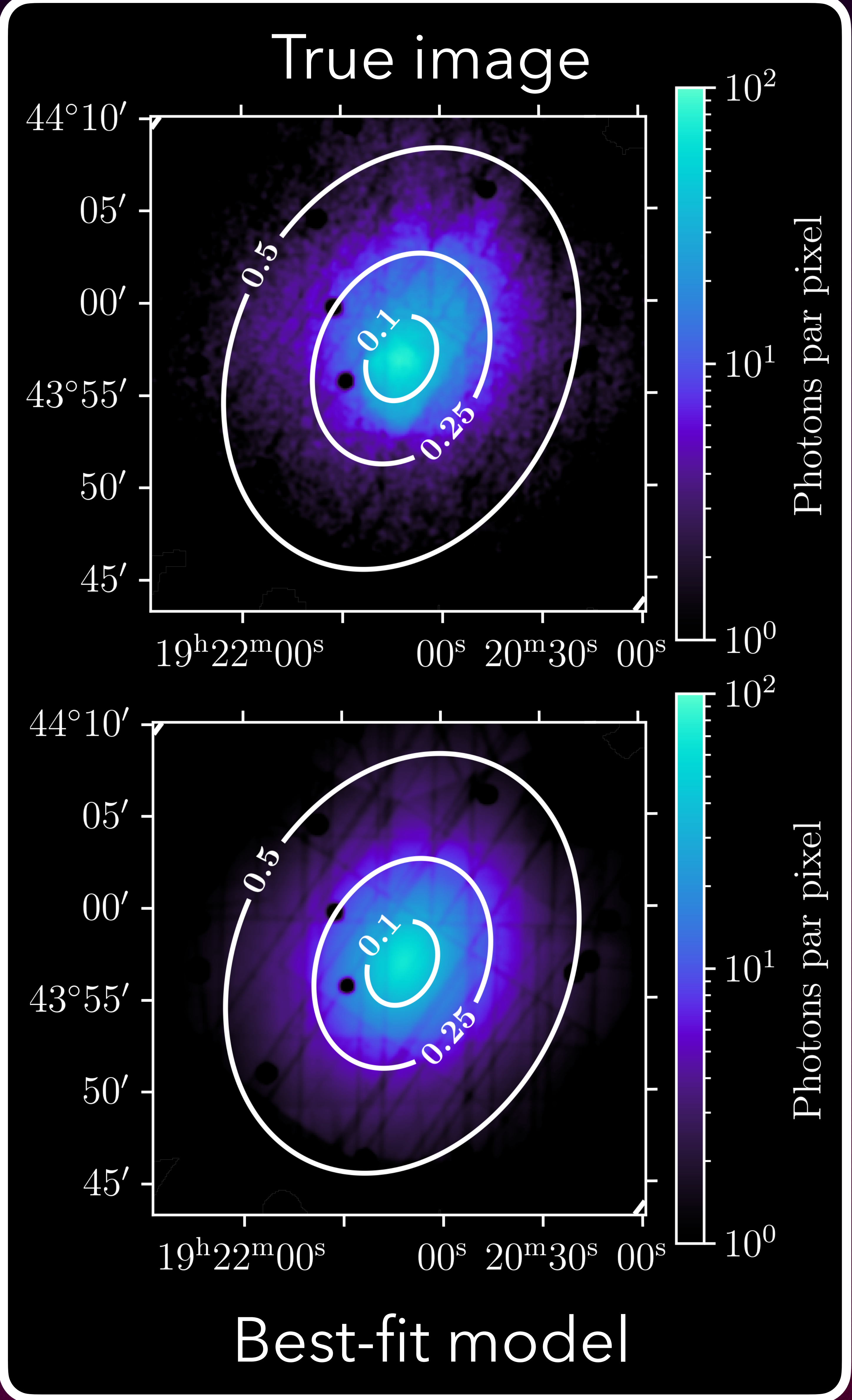


Mach number → hydrostatic bias

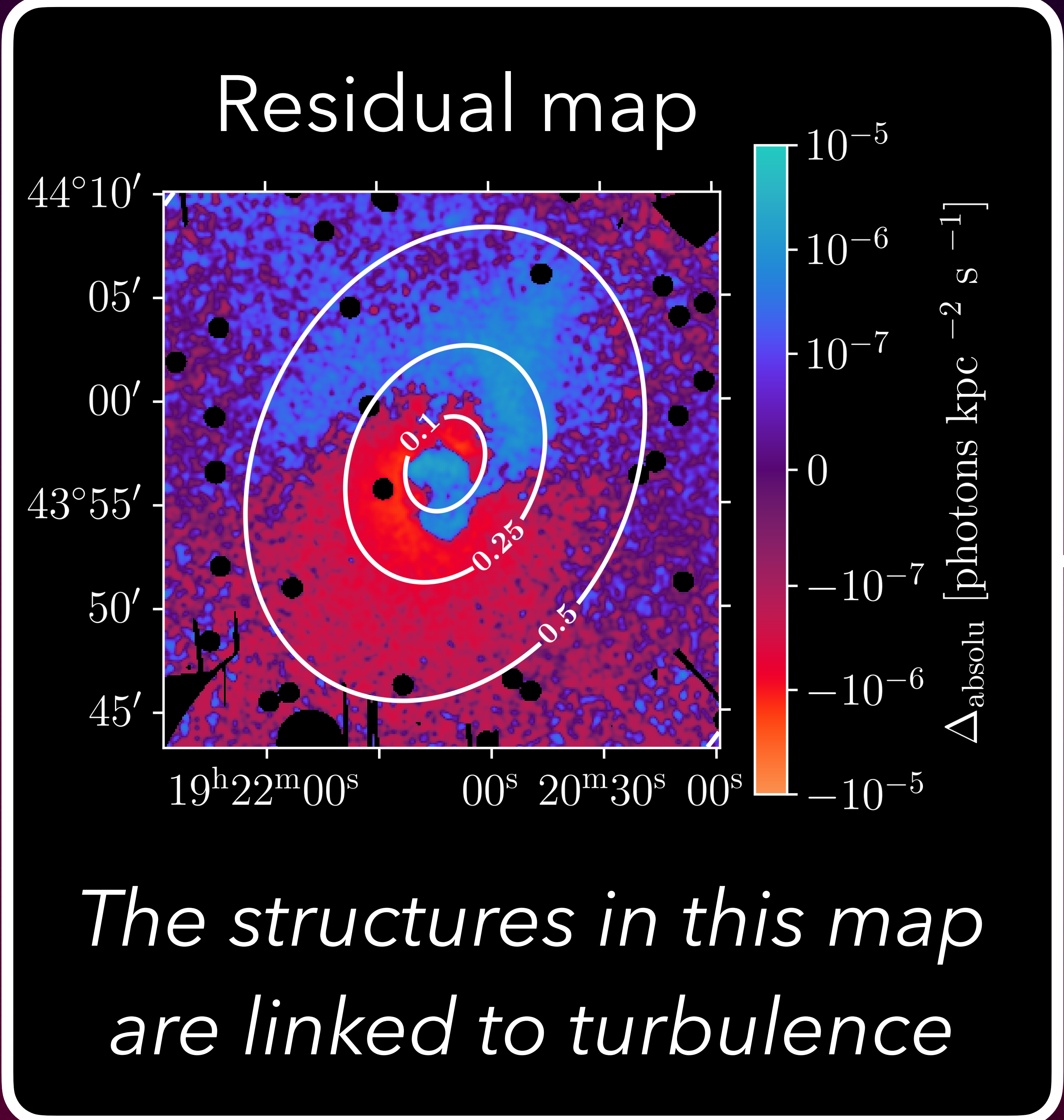


Injection scale → turbulence driver

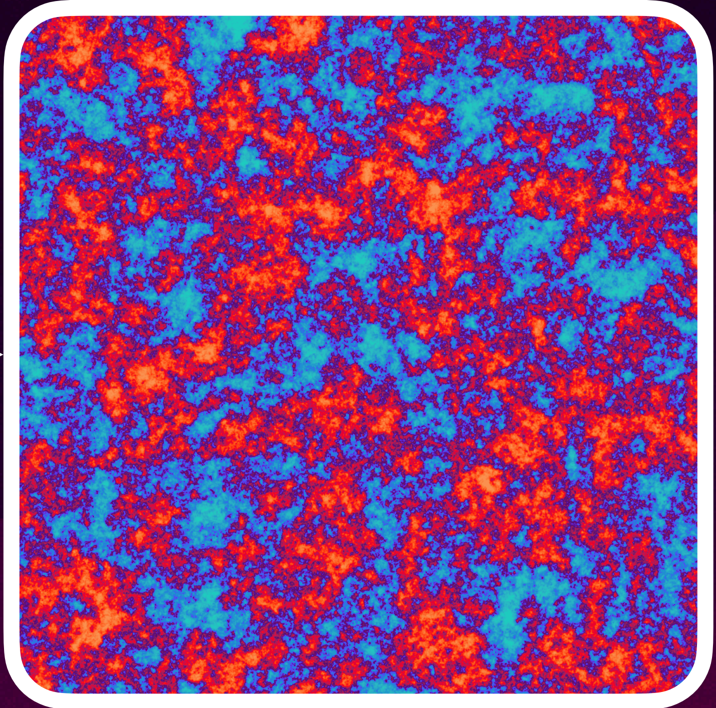
Probing the turbulent motion with fluctuations



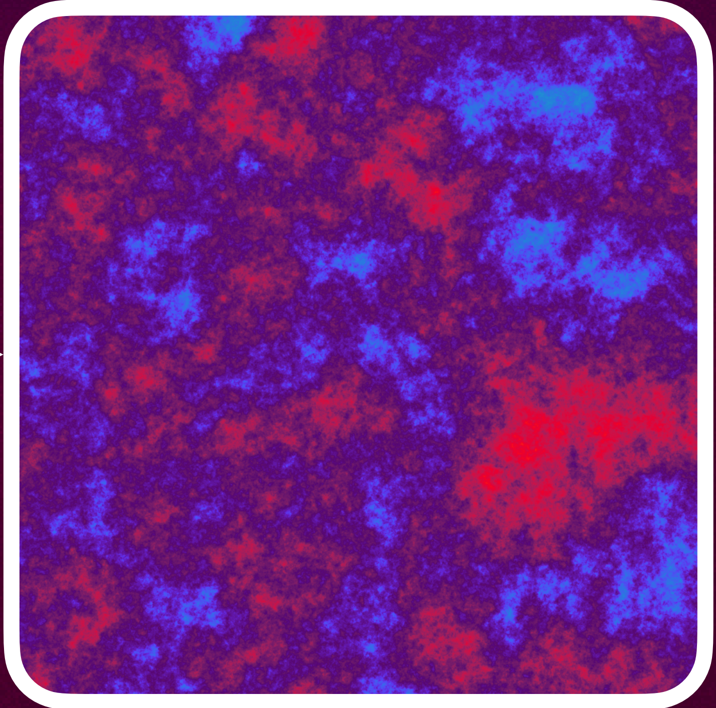
Assume that fluctuations are a GRF with Kolmogorov-like spectrum



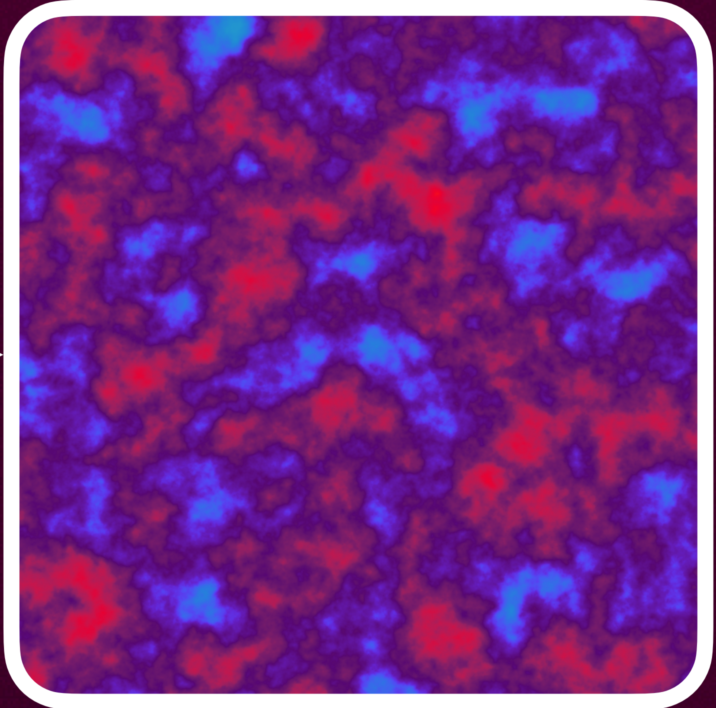
The structures in this map are linked to turbulence



Mach number → hydrostatic bias

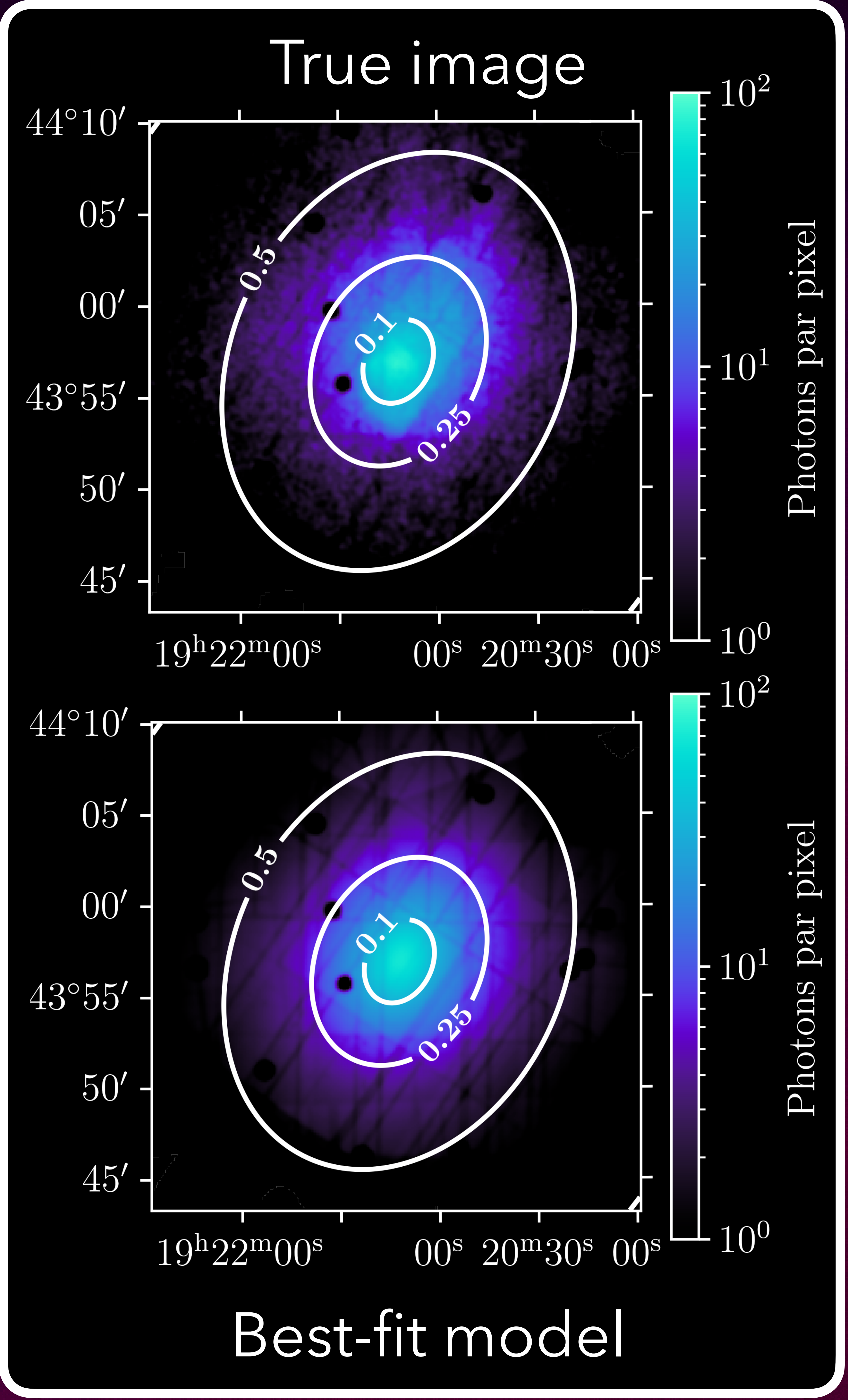


Injection scale → turbulence driver

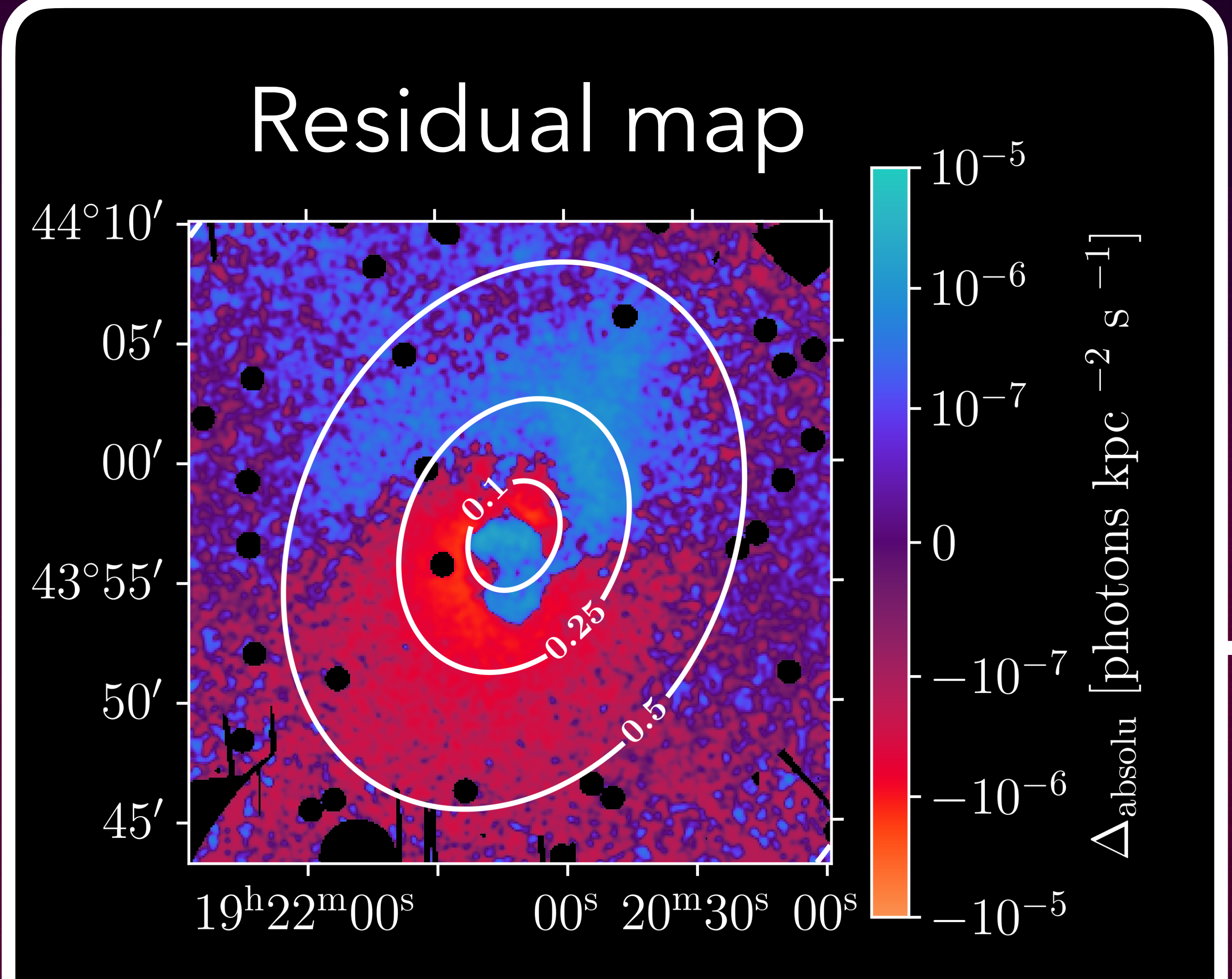


Cascading rate → gas physics

Probing the turbulent motion with fluctuations

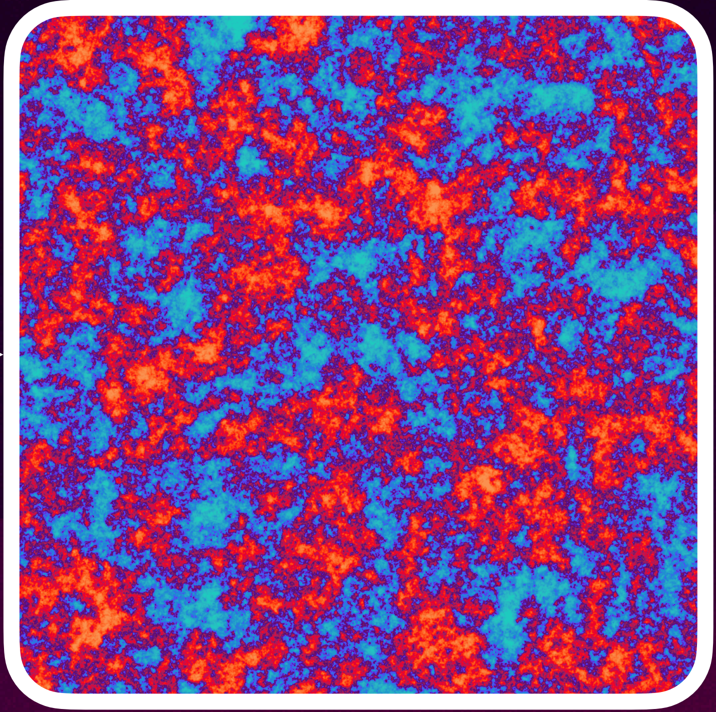


Assume that fluctuations are a GRF with Kolmogorov-like spectrum

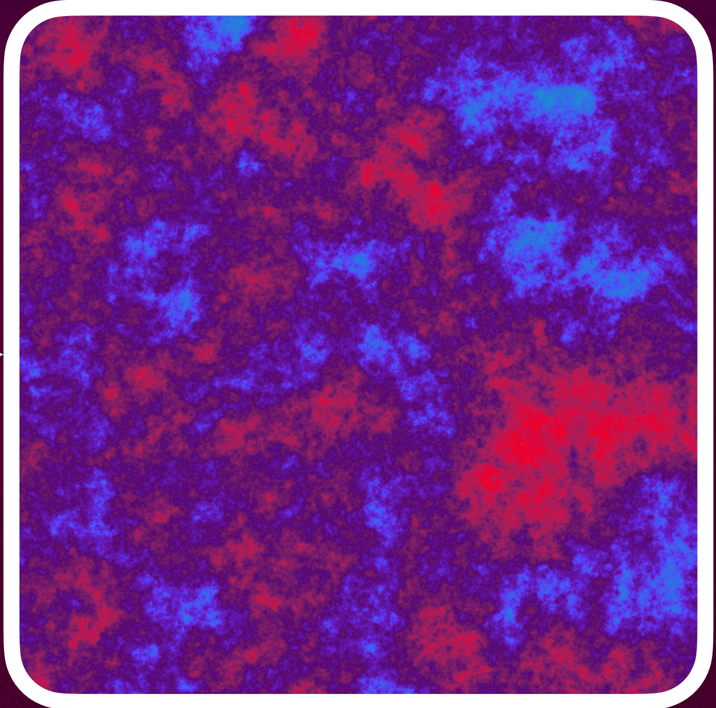


The structures in this map are linked to turbulence

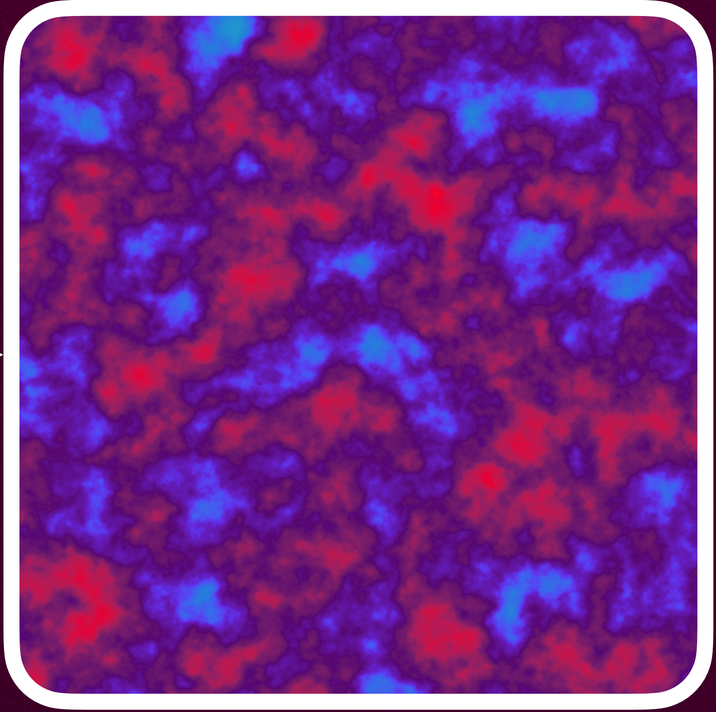
No likelihood because of sample variance (and masking)



Mach number → hydrostatic bias



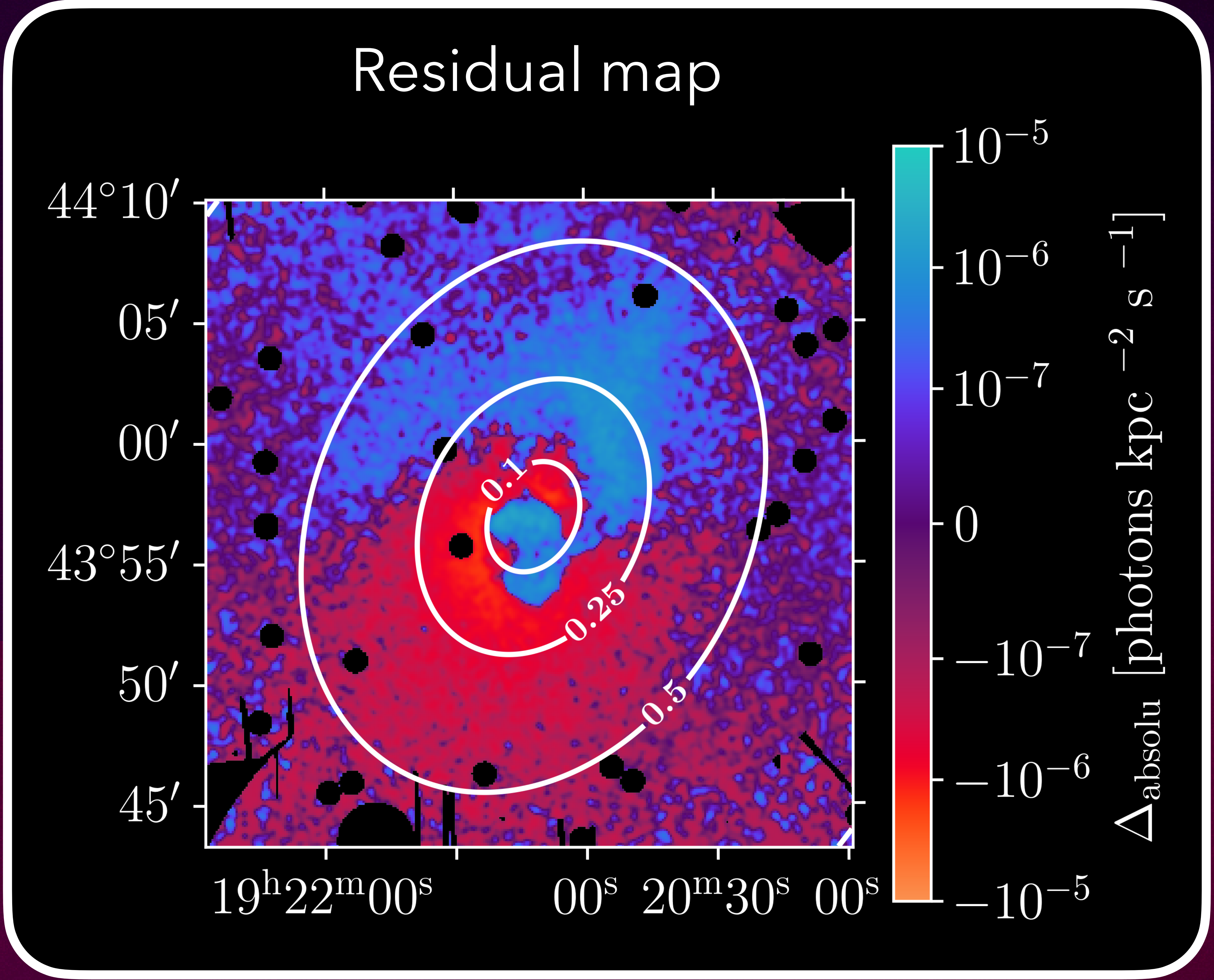
Injection scale → turbulence driver



Cascading rate → gas physics

Crafting an observable for the fluctuation map

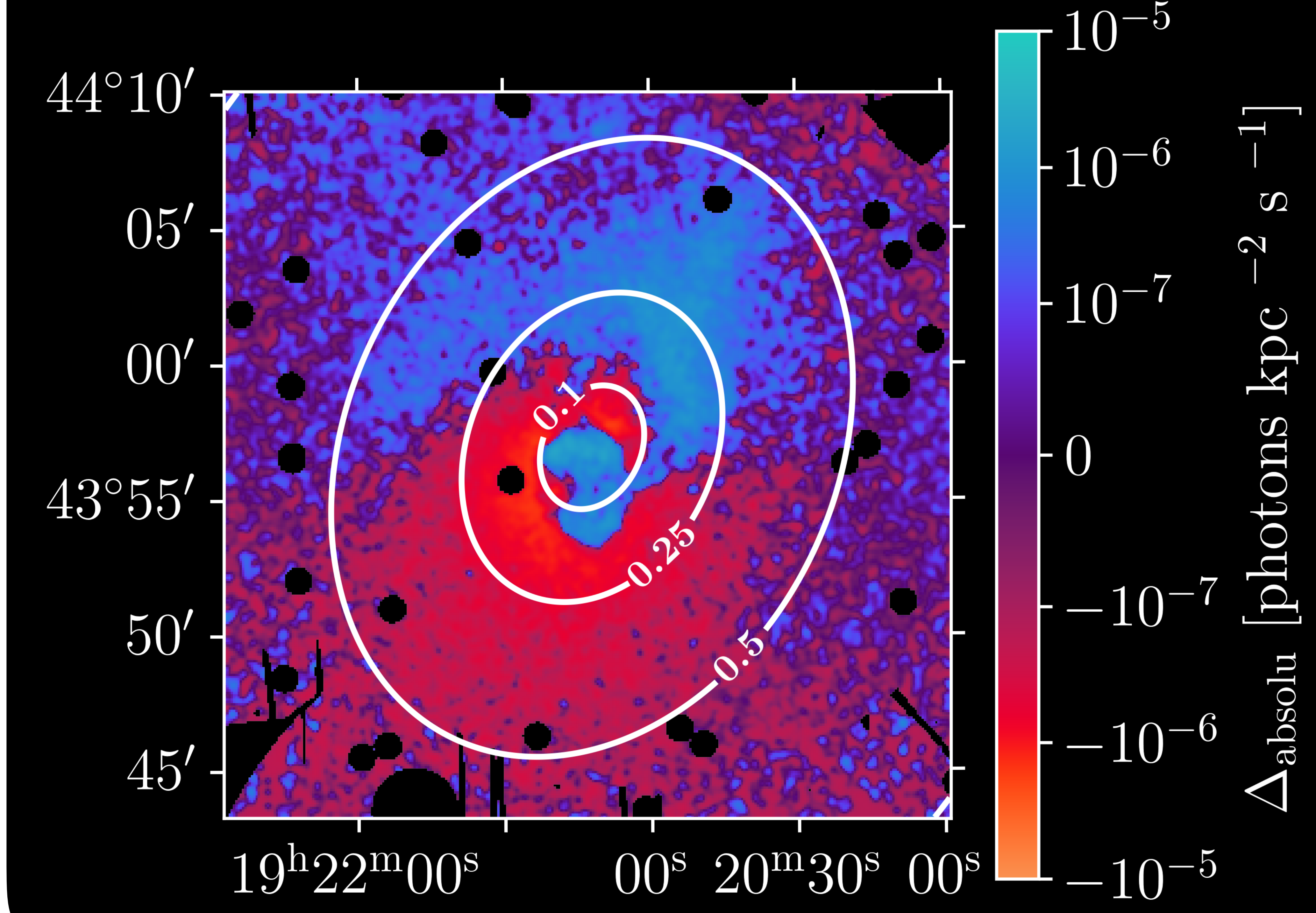
Crafting an observable for the fluctuation map



- Low interpretability
- High-dimension

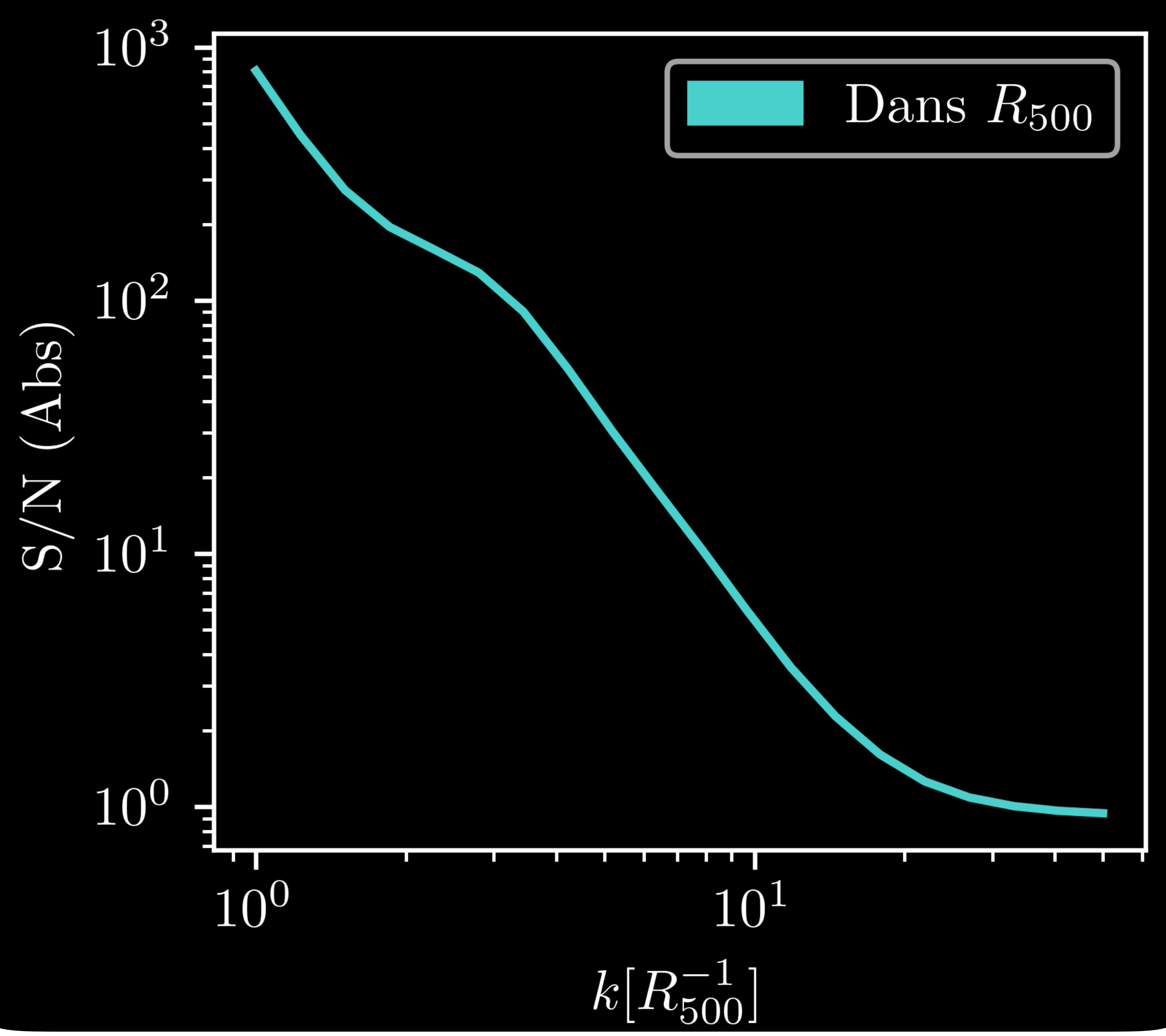
Crafting an observable for the fluctuation map

Residual map



Fourier transform with Mexican Hats (Arévalo + 2012)

Power spectrum (-ish)



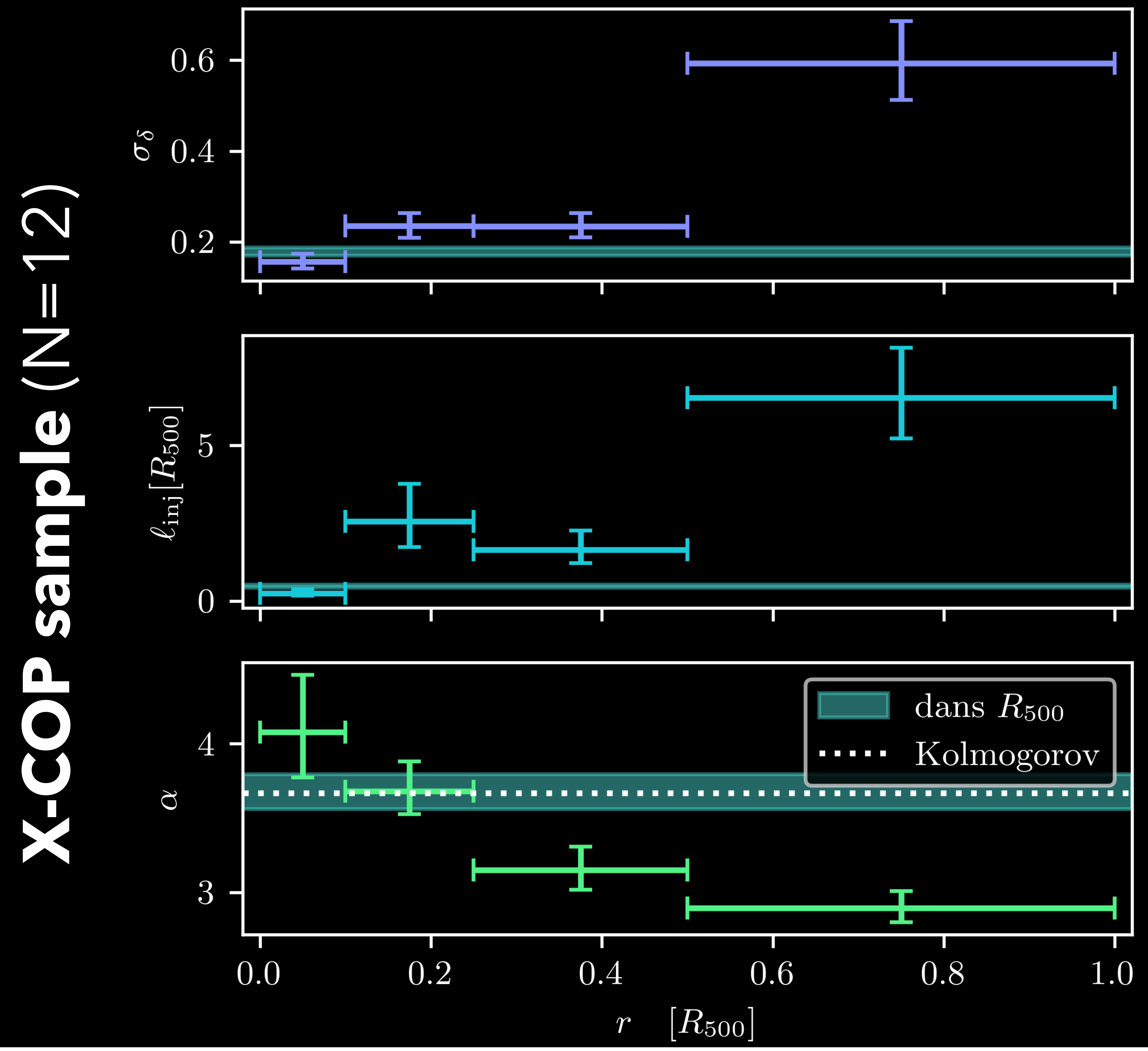
- Low interpretability
- High-dimension

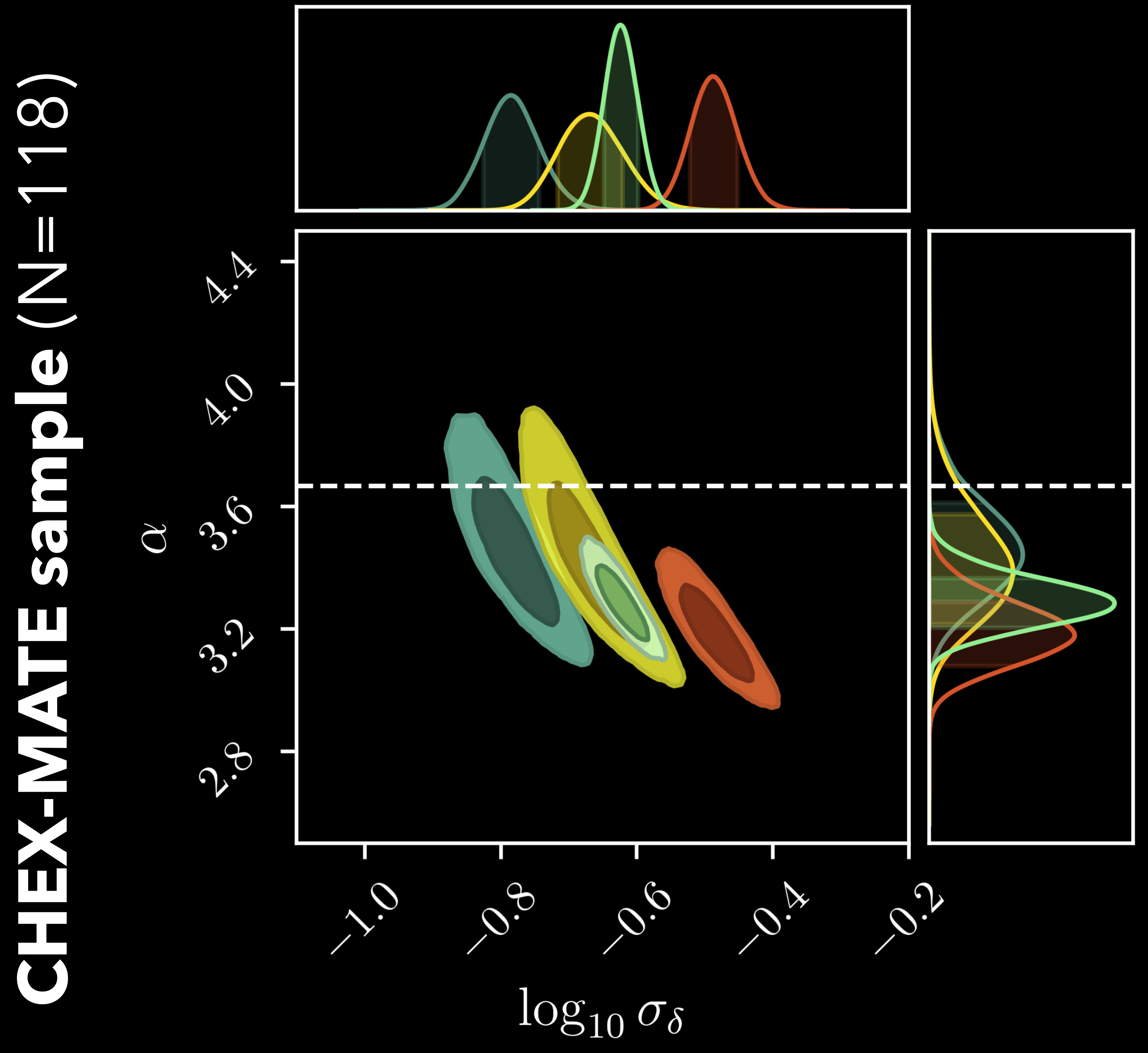
- High interpretability
- Low-dimension

- SBI can learn a likelihood function for many clusters using simulated fluctuation spectra
- Doing so, it automatically **marginalize** over the fluctuation variance
- These likelihoods can be combined to perform **survey over cluster samples**

- SBI can learn a likelihood function for many clusters using simulated fluctuation spectra
- Doing so, it automatically **marginalize** over the fluctuation variance
- These likelihoods can be combined to perform **survey over cluster samples**

Apply it to two cluster samples





Openings on SBI & Clusters

- **Direct Observations**
  - SBI has been successfully applied on true XRISM data in the Coma Cluster (Eckert & al 2025)
  - X-IFU prospective analyses (see Alexei's talk!)

- **SZ Fluctuations**
  - Work leaded by R. Adam on NIKA2 clusters (check PITSZI)
  - Coma fluctuations with Planck revisited (B. Sigal)

Conclusions

SBI can solve inference problems where the likelihood is **intractable** while being **much faster** than regular inference. It turns inference problems in feature engineering problems.

Relevant use cases

- We achieved high-resolution spectroscopy with SBI using physically motivated summary statistics for the **X-ray spectra from XRISM/Resolve and newAthena/X-IFU**.
- We successfully used SBI to probe **turbulence in the ICM**. It enabled large scale study of the X-ray fluctuations in both the X-COP and CHEX-MATE cluster samples.

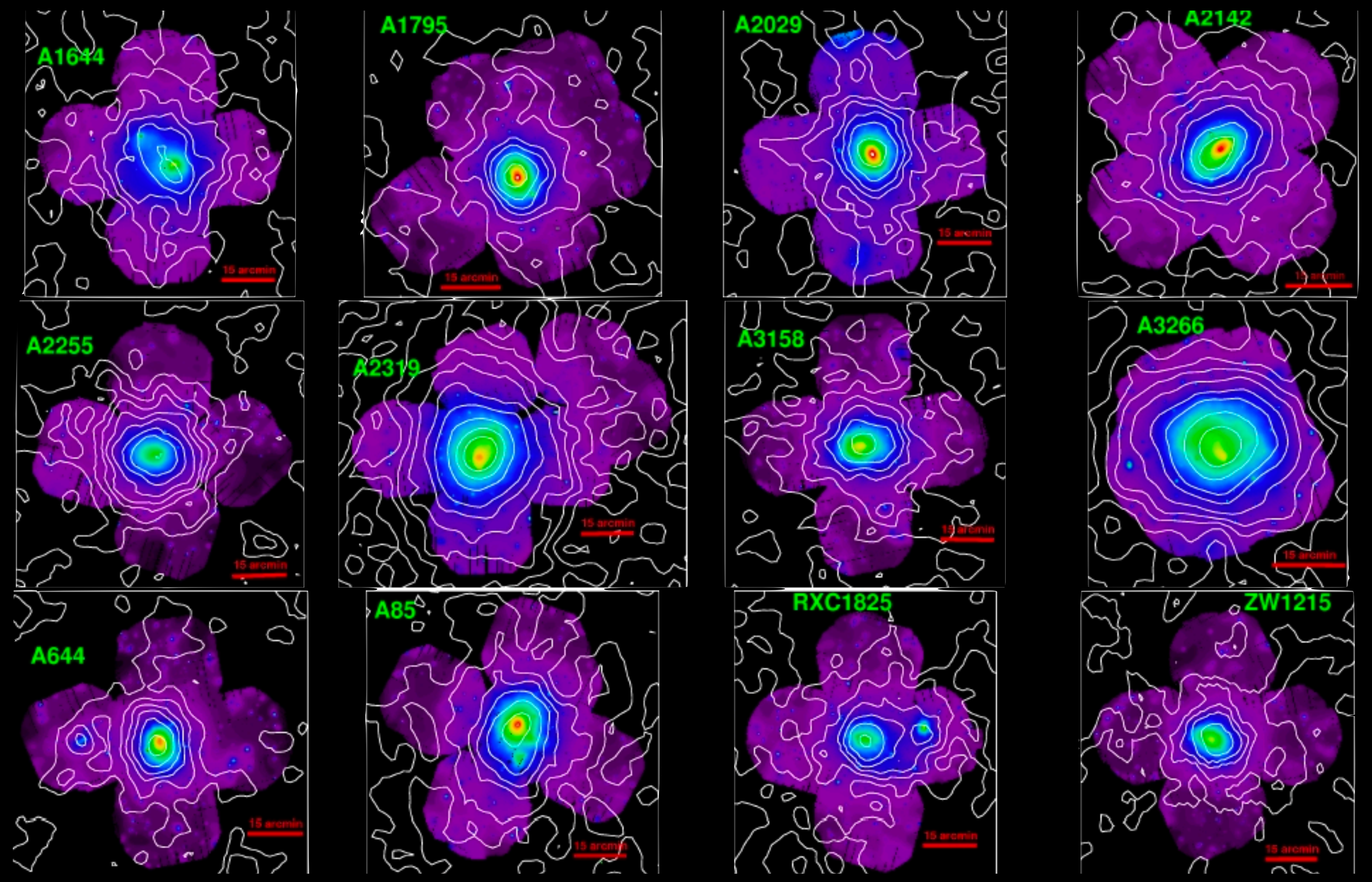
Backup

- SBI can learn a likelihood function using simulated fluctuation images
- Doing so, it automatically **marginalize** over the fluctuation variance
- These likelihoods can be combined to perform **survey over cluster samples**

- SBI can learn a likelihood function using simulated fluctuation images
- Doing so, it automatically **marginalize** over the fluctuation variance
- These likelihoods can be combined to perform **survey over cluster samples**

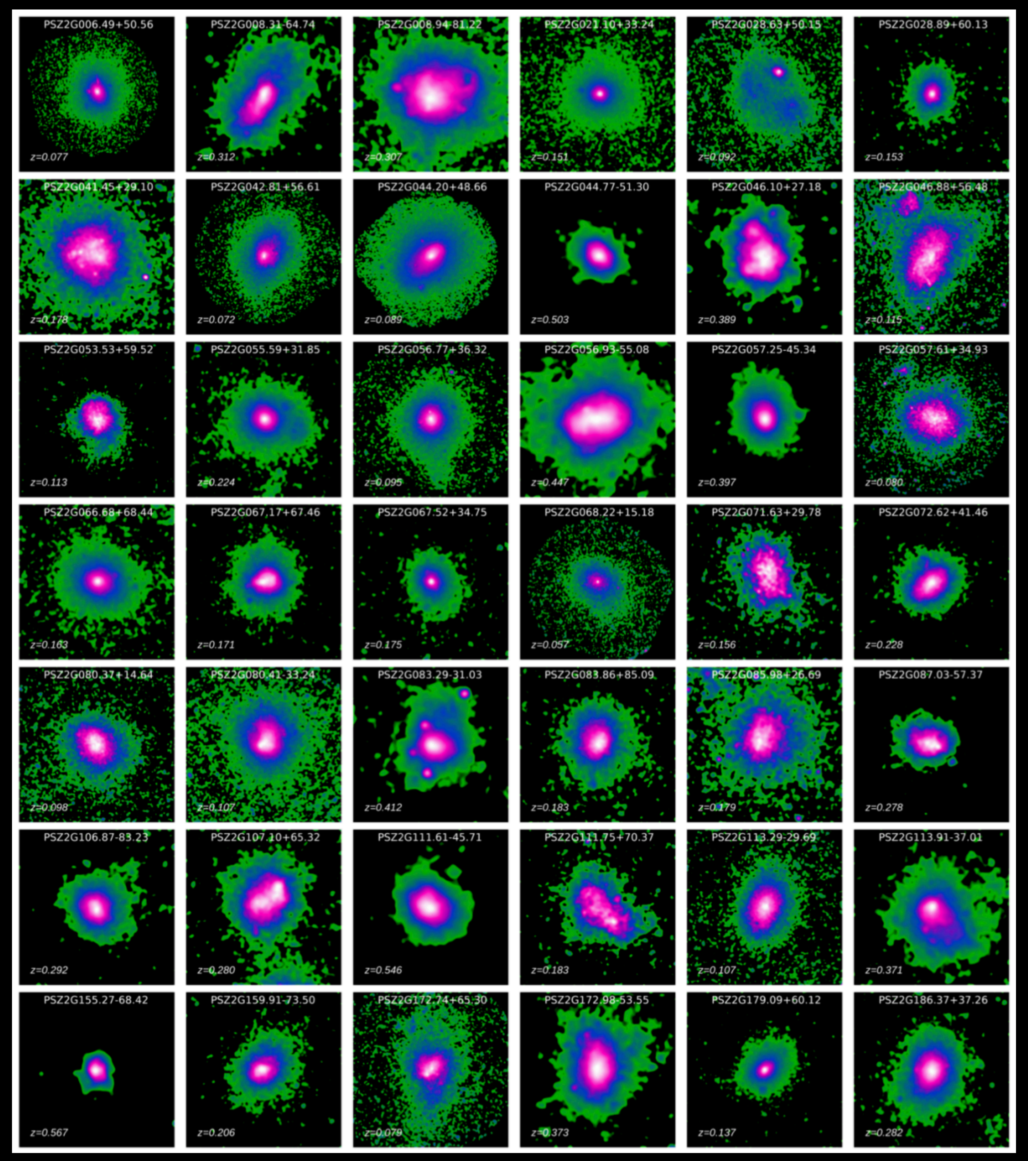
Apply it to two cluster samples

X-COP (Eckert & al. 2017)



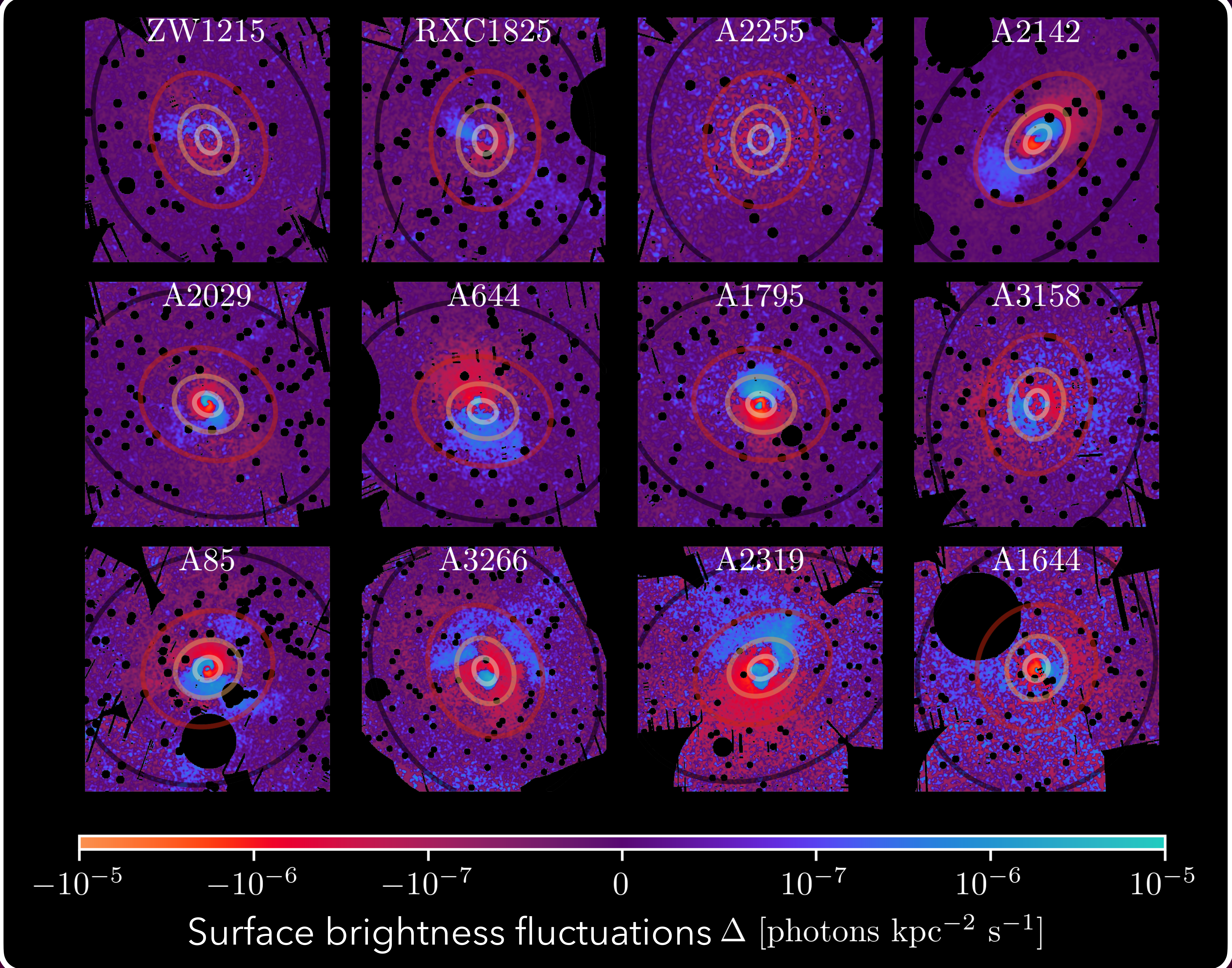
12 massive, nearby clusters observed up to R_{200} ($z < 0.07$, $M \sim 10^{15} M_\odot$)

CHEX-MATE (CHEX-MATE Collaboration, 2021)



- 118 clusters in the local Universe
- Homogeneous measurements up to R_{500}

$z < 0.6$,
$[2 \sim 20]\ 10^{14} M_\odot$

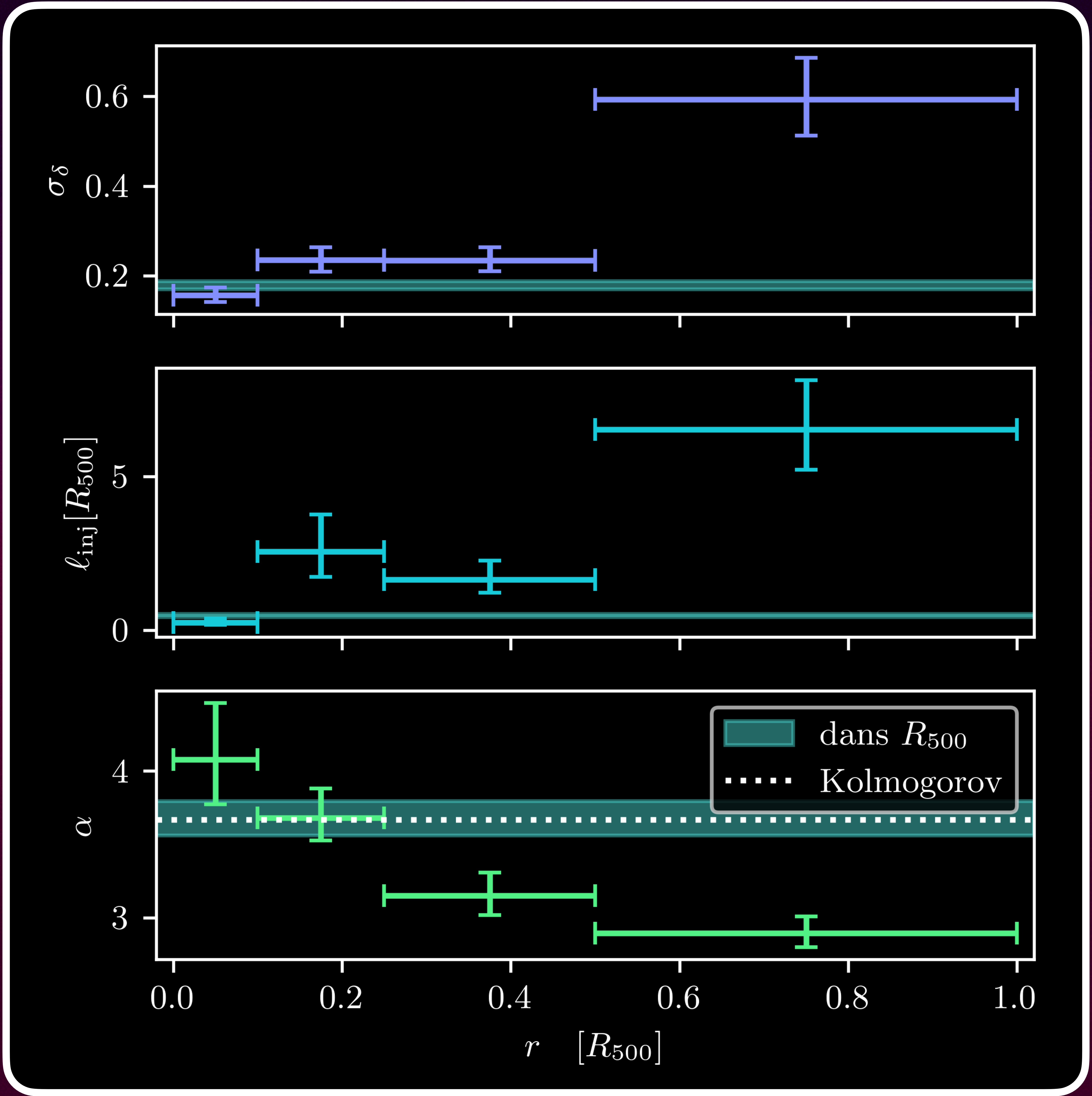


Region	Radius
(I)	$0 < r < R_{500}/10$
(II)	$R_{500}/10 < r < R_{500}/4$
(III)	$R_{500}/4 < r < R_{500}/2$
(IV)	$R_{500}/2 < r < R_{500}$

Split the analysis in 4 regions

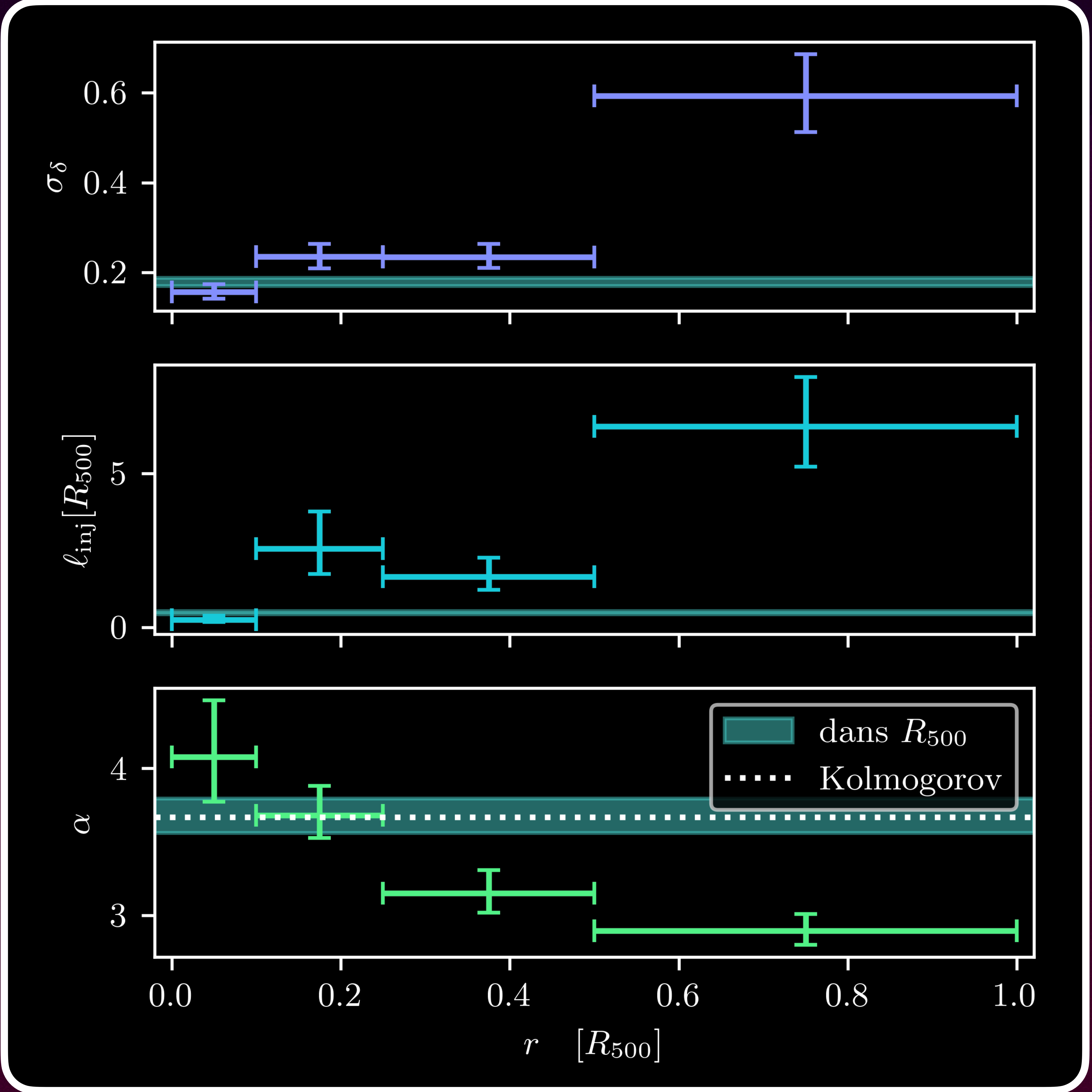
Radial evolution in X-COP

Radial evolution in X-COP



Radial evolution in X-COP

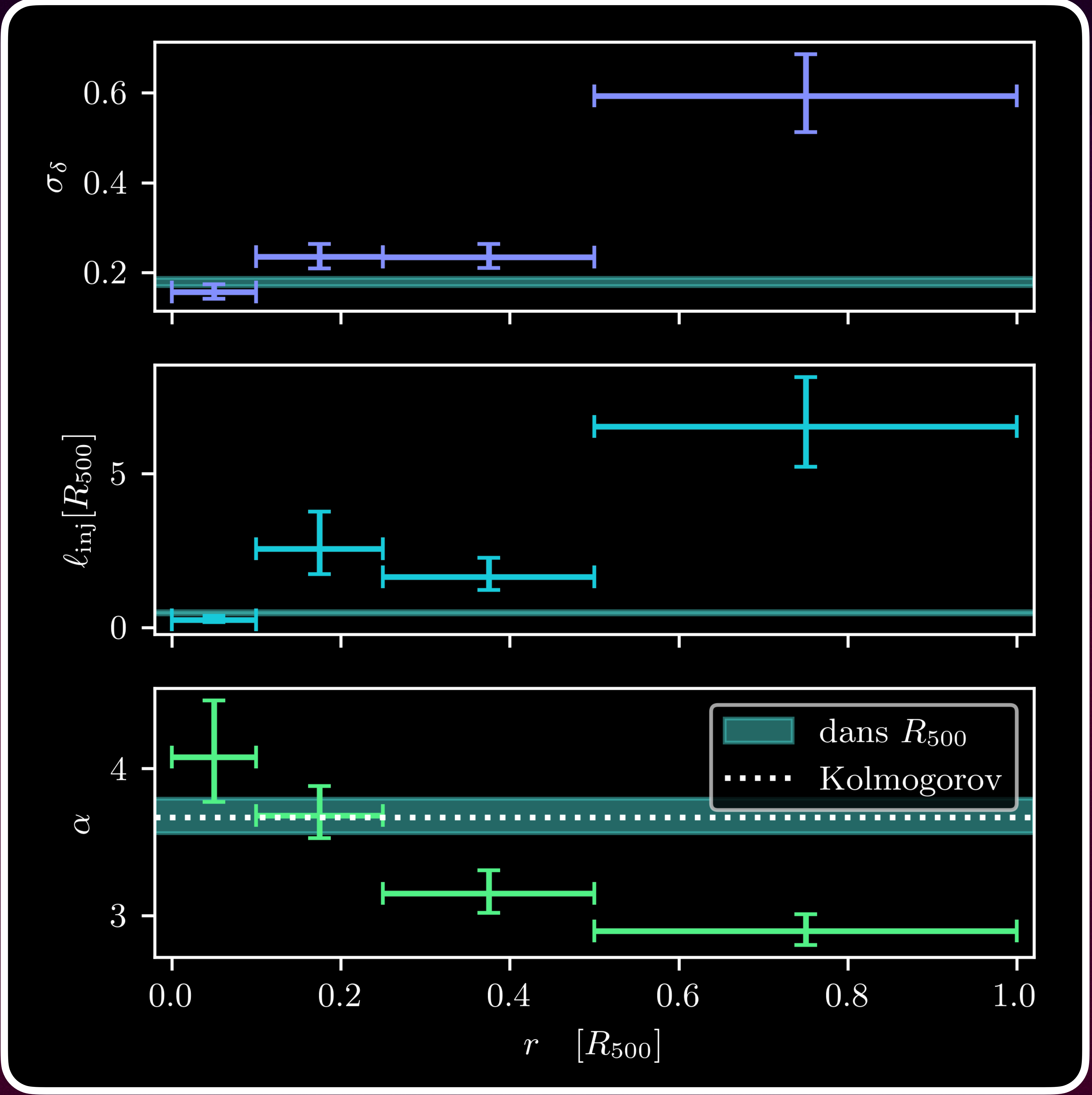
- **Profile :** the normalisation increases with radius → the overall disturbance increases in external regions
- **Global :** $\mathscr{M} \sim 0.1$, subsonic



Radial evolution in X-COP

- **Profile :** the normalisation increases with radius → the overall disturbance increases in external regions
- **Global :** $\mathscr{M} \sim 0.1$, subsonic

- **Profile :** the injection scale increases with radius → transition between feedback, sloshing and merging
- **Global :** dominated by central region

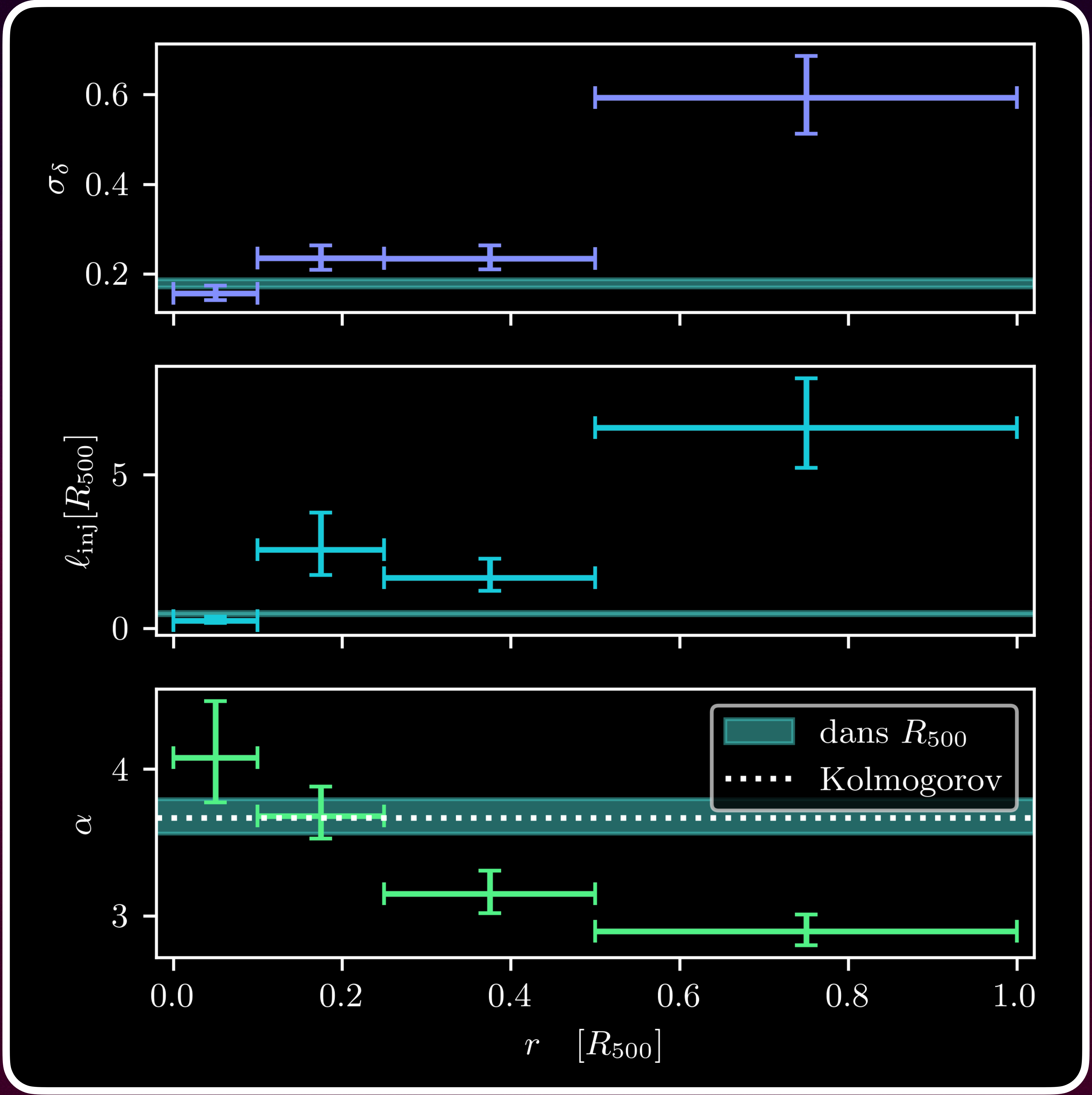


Radial evolution in X-COP

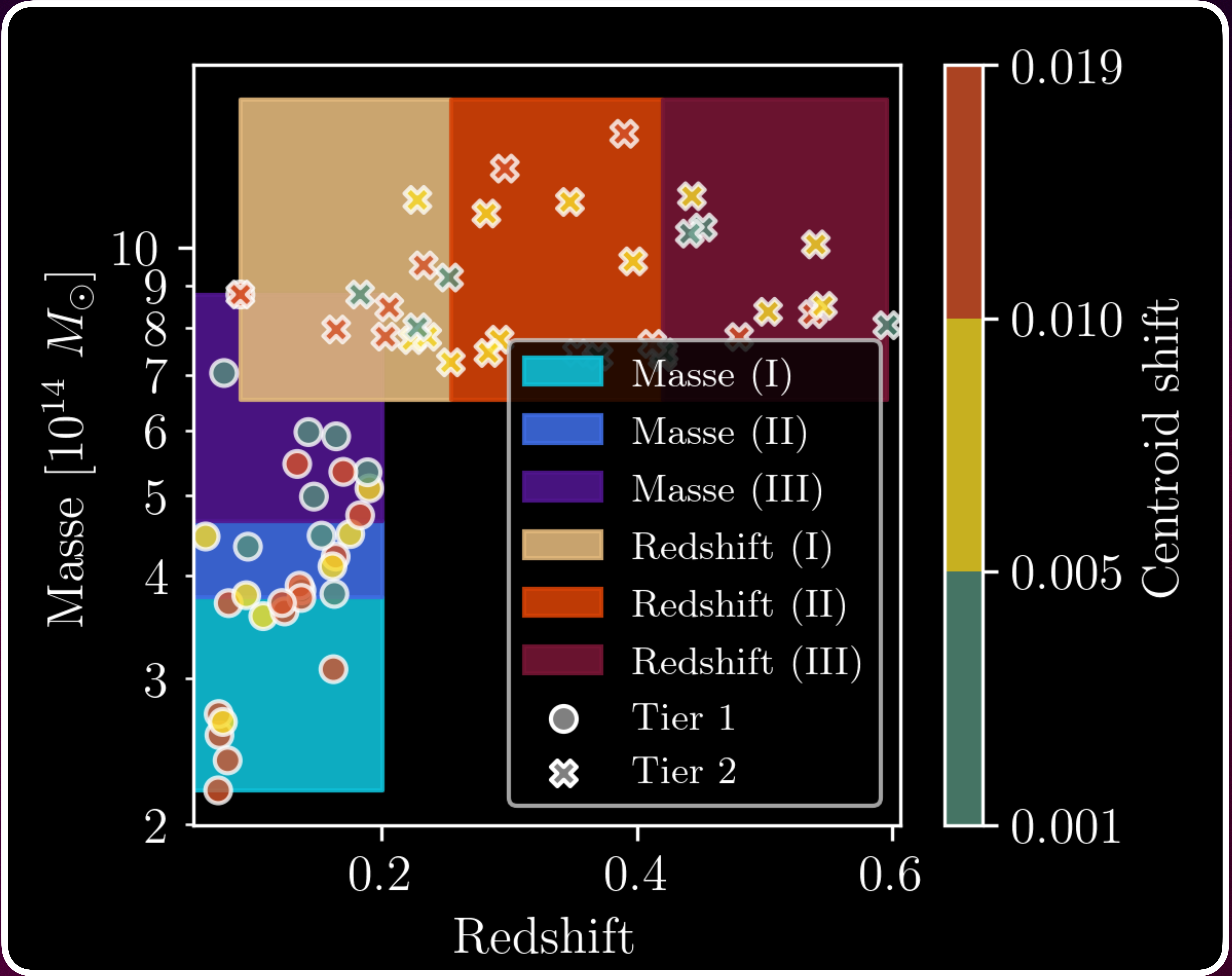
- **Profile :** the normalisation increases with radius → the overall disturbance increases in external regions
- **Global :** $\mathscr{M} \sim 0.1$, subsonic

- **Profile :** the injection scale increases with radius → transition between feedback, sloshing and merging
- **Global :** dominated by central region

- **Profile :** the spectral slope decreases with radius → transition between structured and noisy fluctuations
- **Global :** Kolmogorov-like!



Sample study with CHEX-MATE

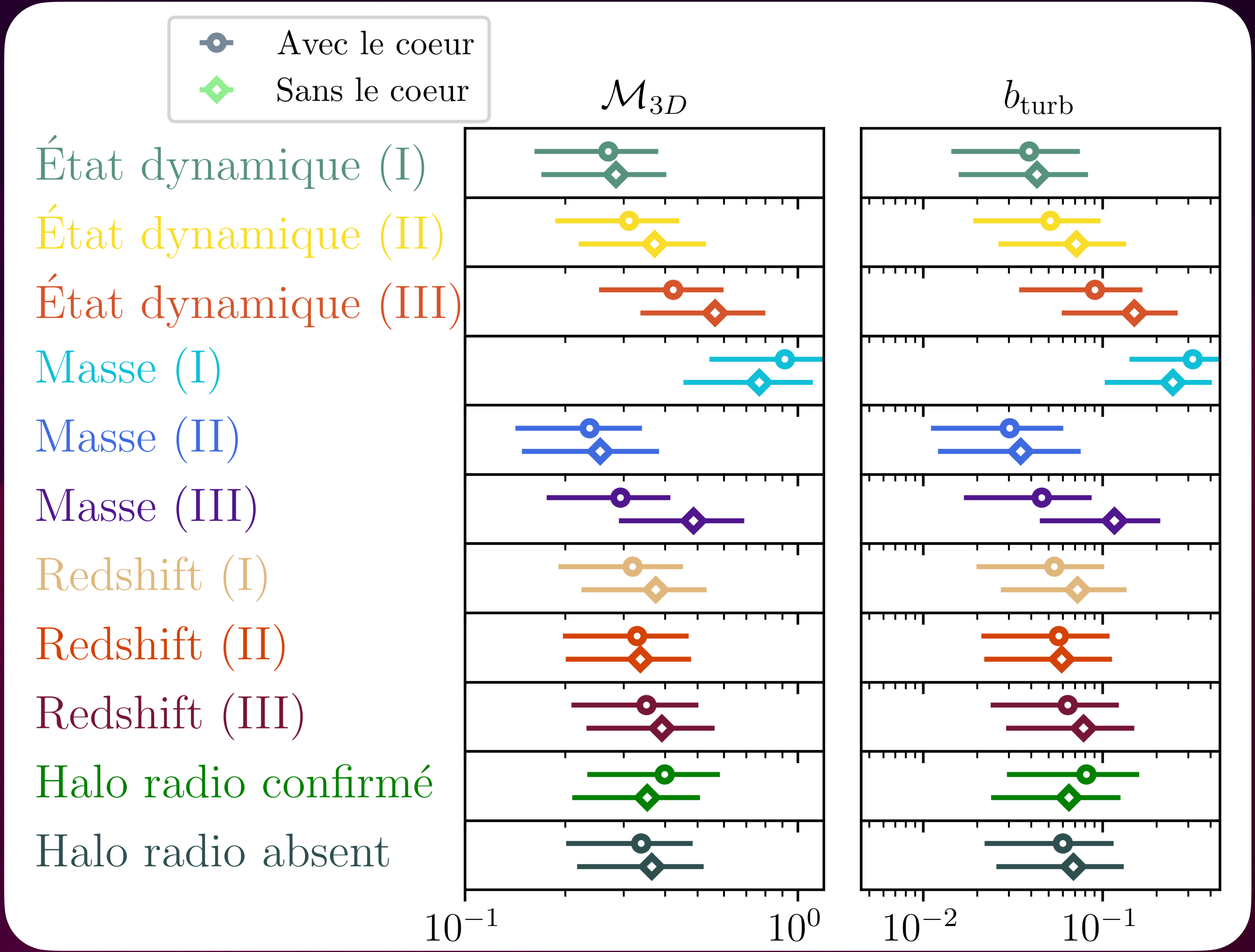


Investigate the link between cluster properties and turbulence

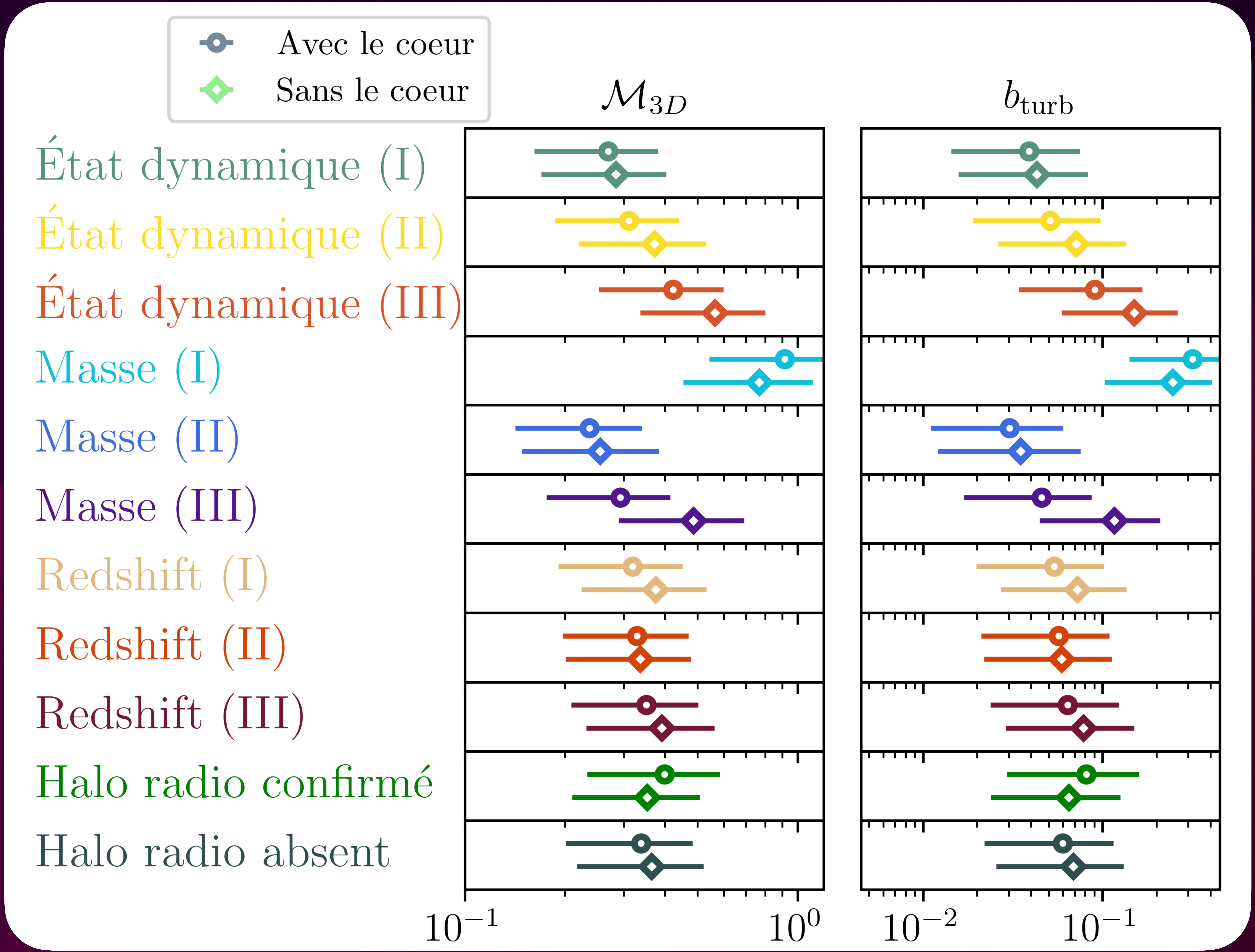
- Study on 64 cluster, after cleaning the sample from the most irregular ones
- Subdivide in three sub-samples in mass, redshift and dynamic state

Relaxed ($w < 0.005$)
In-between ($w < 0.01$)
Disturbed ($w < 0.02$)
Low mass ($< 4 \times 10^{14} M_\odot$)
In-between
High mass ($> 5 \times 10^{14} M_\odot$)
$z < 0.2$
$z < 0.4$
$z < 0.6$
Splitting on dynamical state
Splitting on mass
Splitting on redshift
Kolmogorov
α
4.4
4.0
3.6
3.2
2.8
−1.0
−0.8
−0.6
−0.4
−0.2
$\log_{10} \sigma_\delta$

Turbulence & hydrostatic mass bias



Turbulence & hydrostatic mass bias



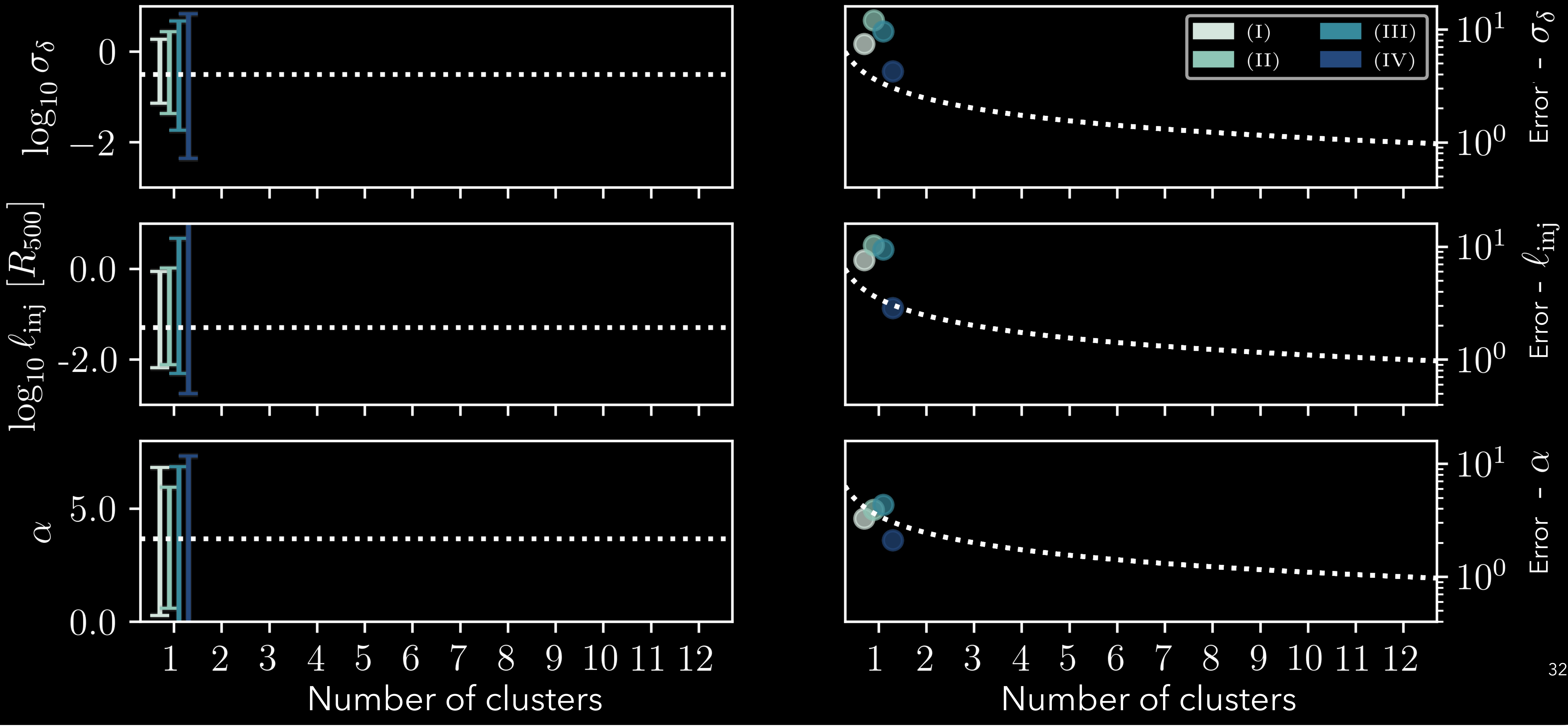
$\mathcal{M}_{3D} \sim 0.3 - 0.5$

$b_{turb} \sim (9 \pm 6)\,\%$

Coherent with direct and indirect observations, and numerical simulations

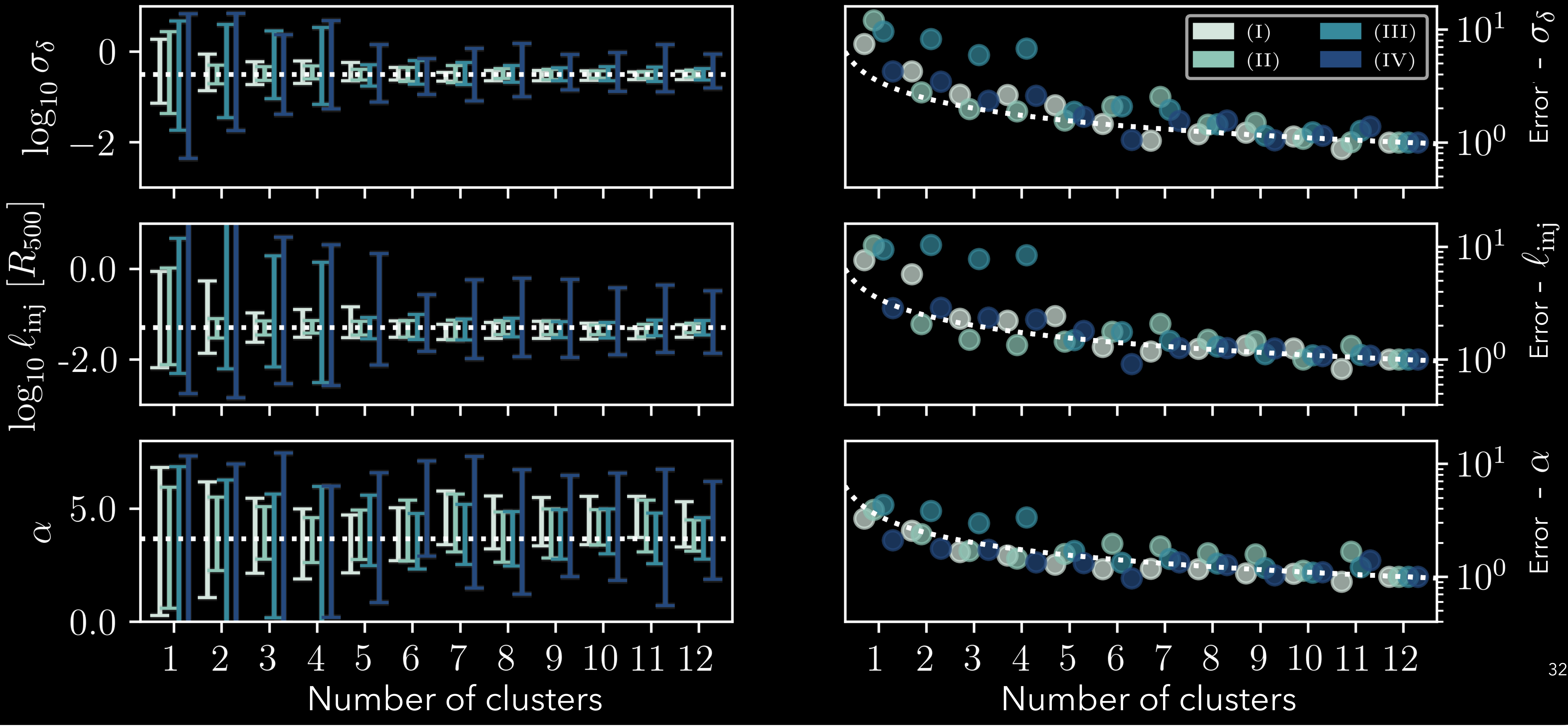
Validating SBI for galaxy clusters

(I)	$0 < r < R_{500}/10$
(II)	$R_{500}/10 < r < R_{500}/4$
(III)	$R_{500}/4 < r < R_{500}/2$
(IV)	$R_{500}/2 < r < R_{500}$



Validating SBI for galaxy clusters

(I)	$0 < r < R_{500}/10$
(II)	$R_{500}/10 < r < R_{500}/4$
(III)	$R_{500}/4 < r < R_{500}/2$
(IV)	$R_{500}/2 < r < R_{500}$

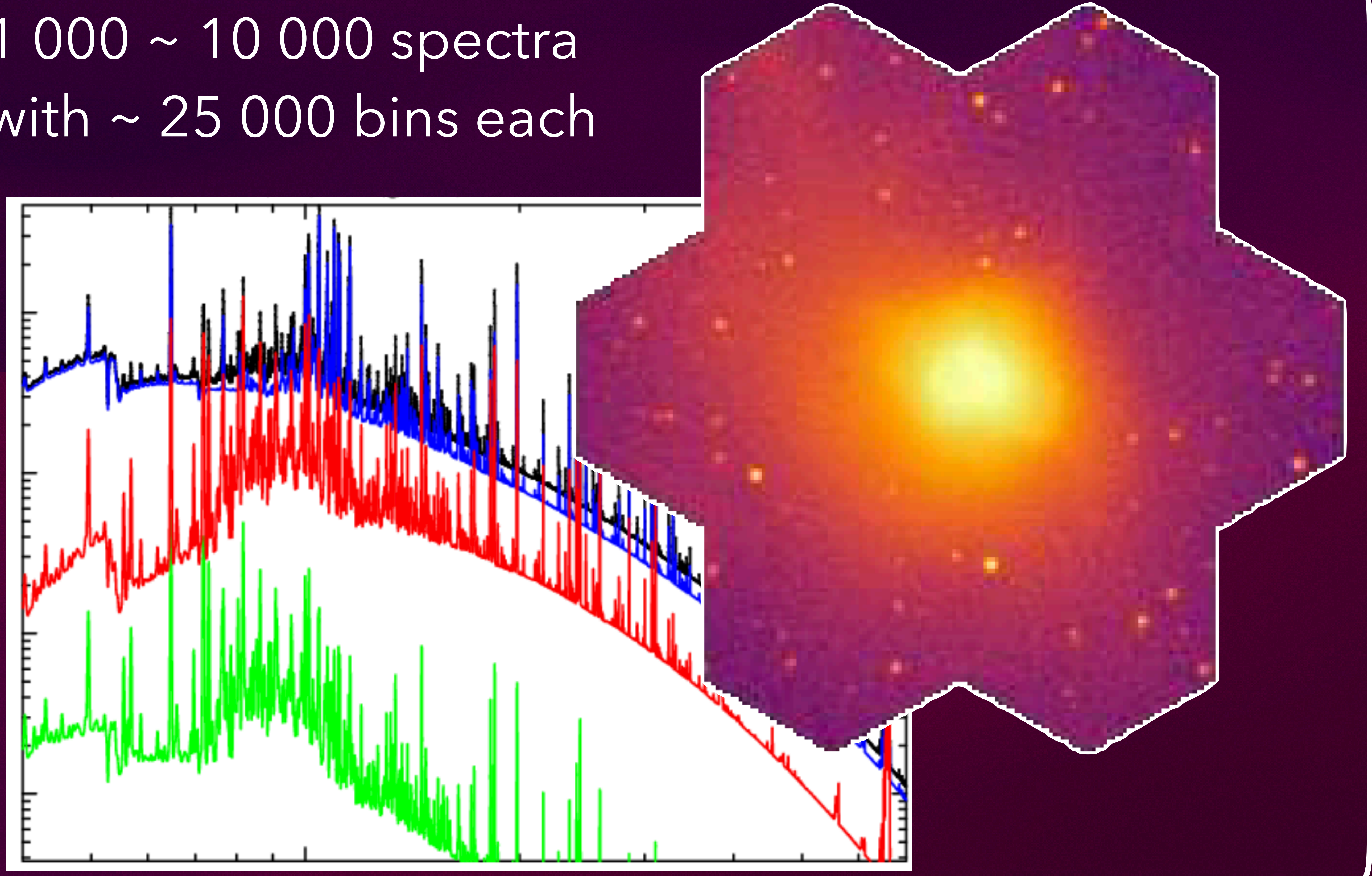


What is it to fit an X-IFU cube ?

X-IFU : high resolution spectrometer in X-ray (~2038)

Mock X-IFU mosaic of a close cluster (z<0.1)

1 000 ~ 10 000 spectra
with ~ 25 000 bins each

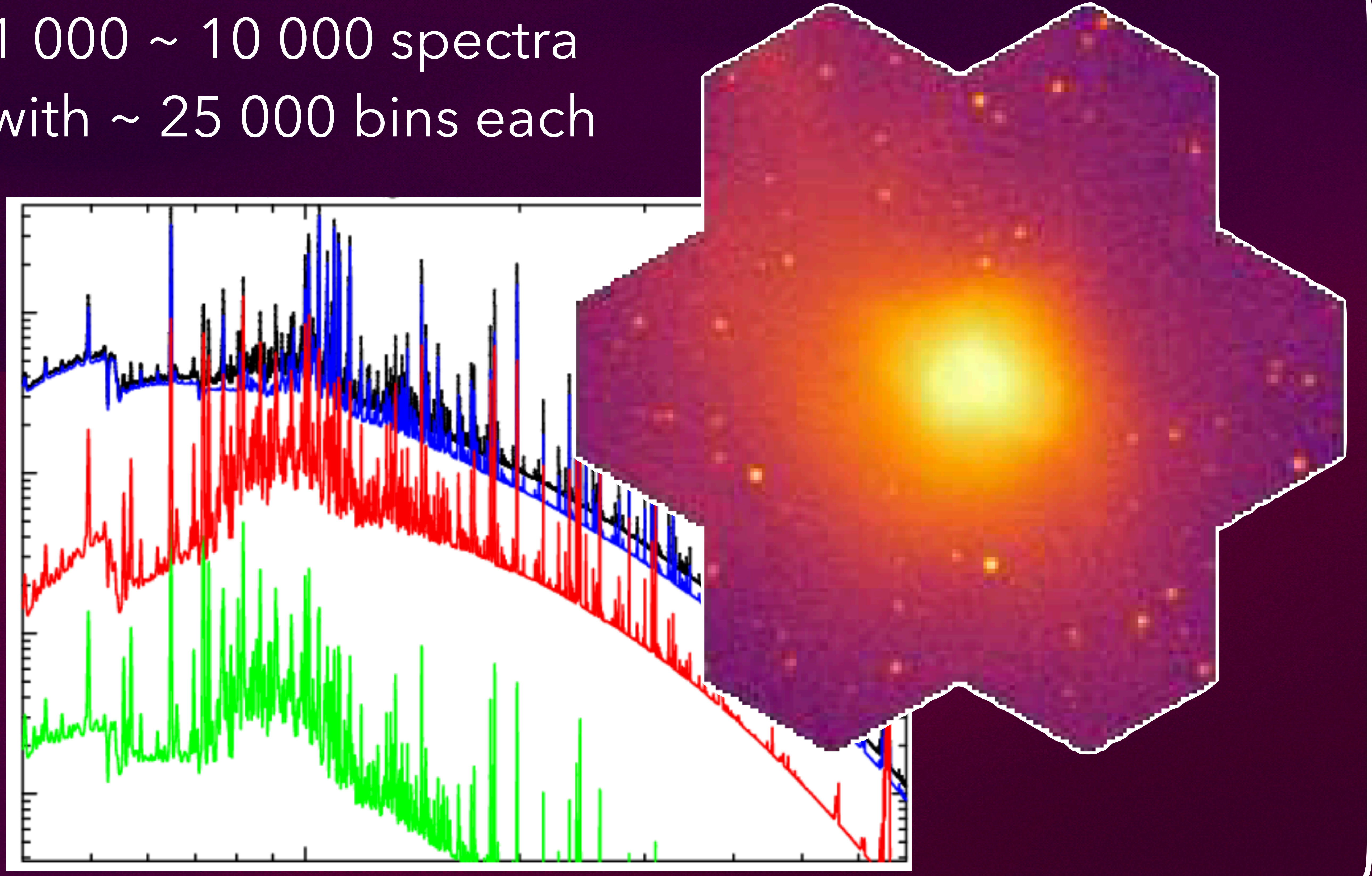


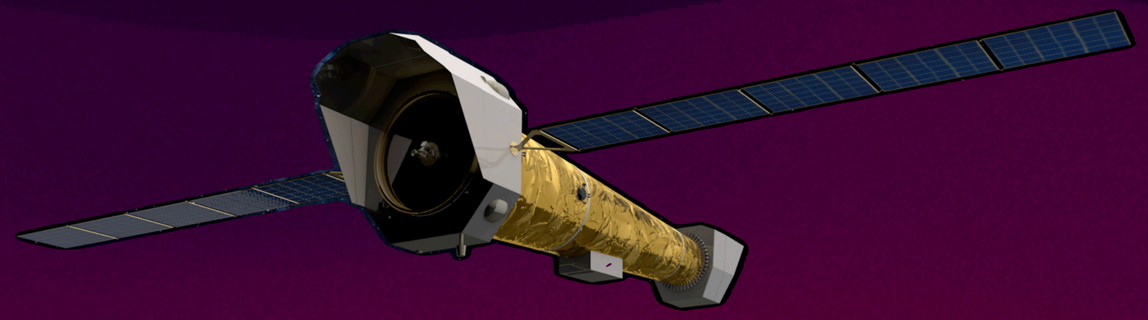
What is it to fit an X-IFU cube ?

X-IFU : high resolution spectrometer in X-ray (~2038)

Mock X-IFU mosaic of a close cluster (z<0.1)

1 000 ~ 10 000 spectra
with ~ 25 000 bins each

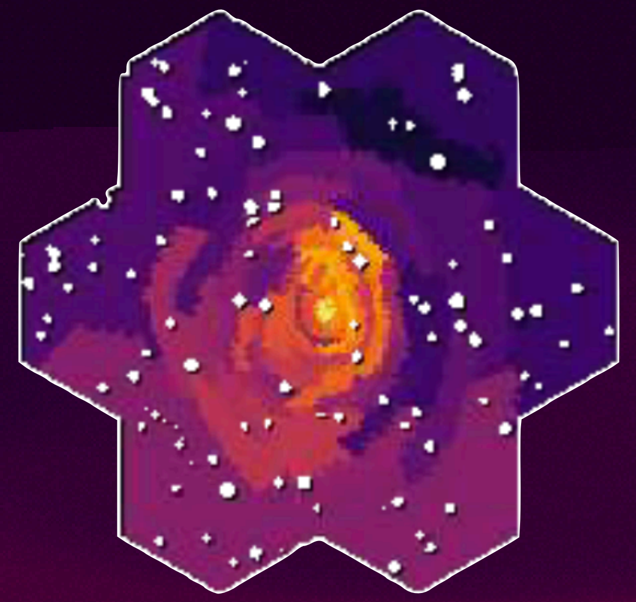




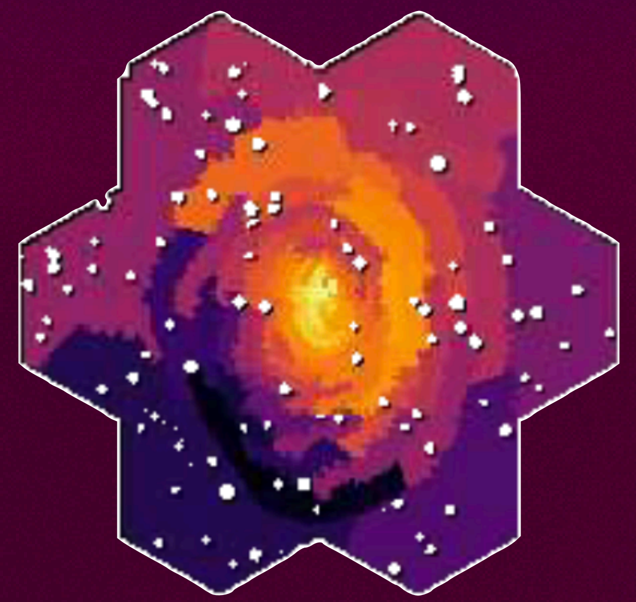
Spatial binning
Model fitting

~ 100k CPU Hours

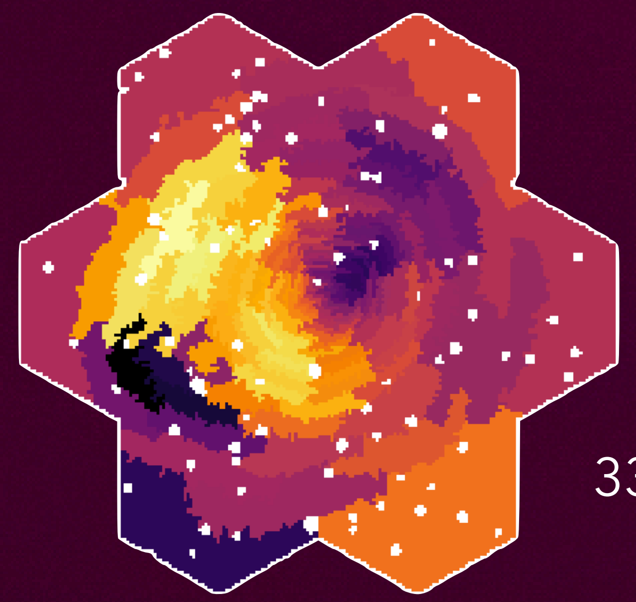
Oxygen



Temperature

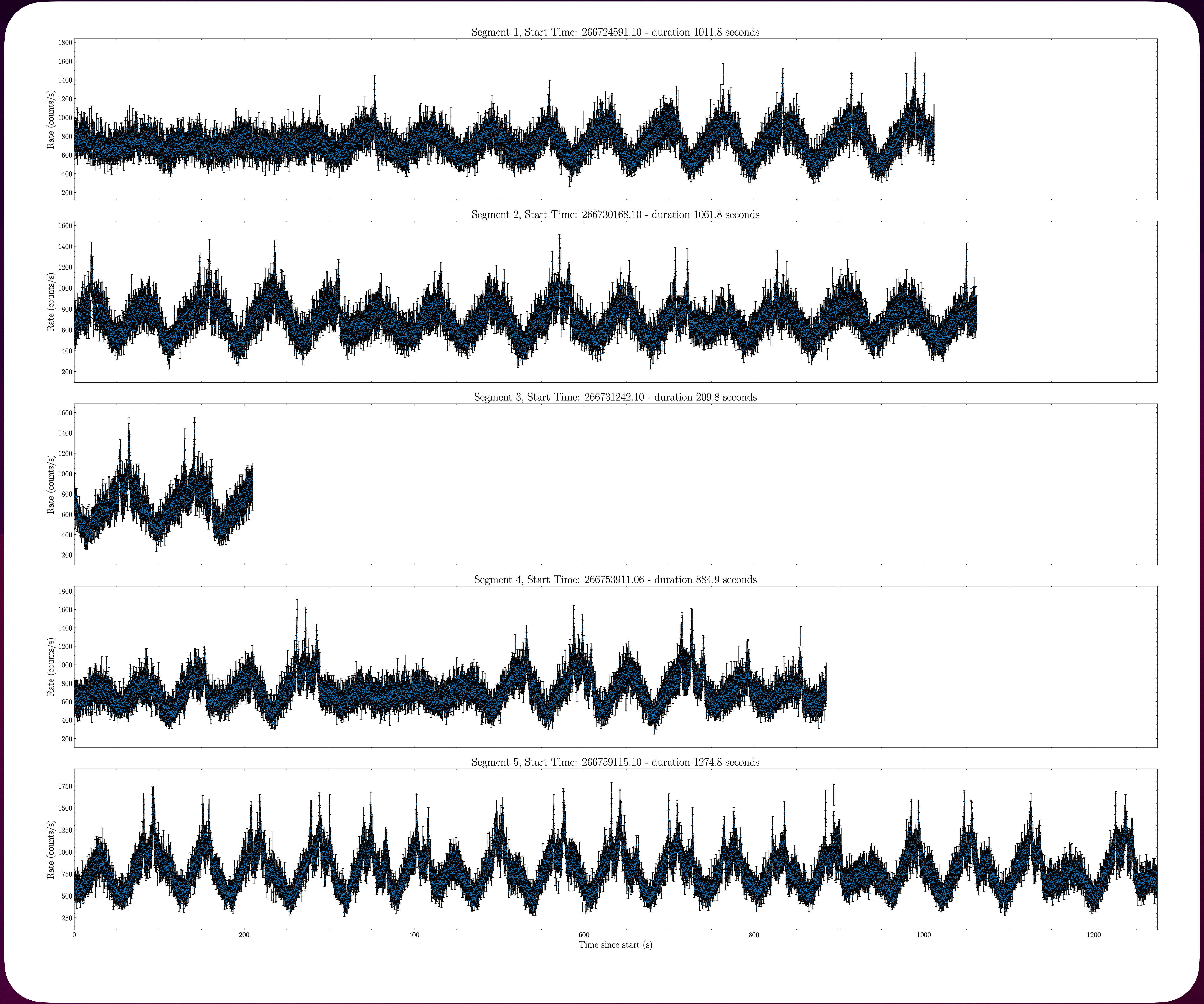


Bulk motion



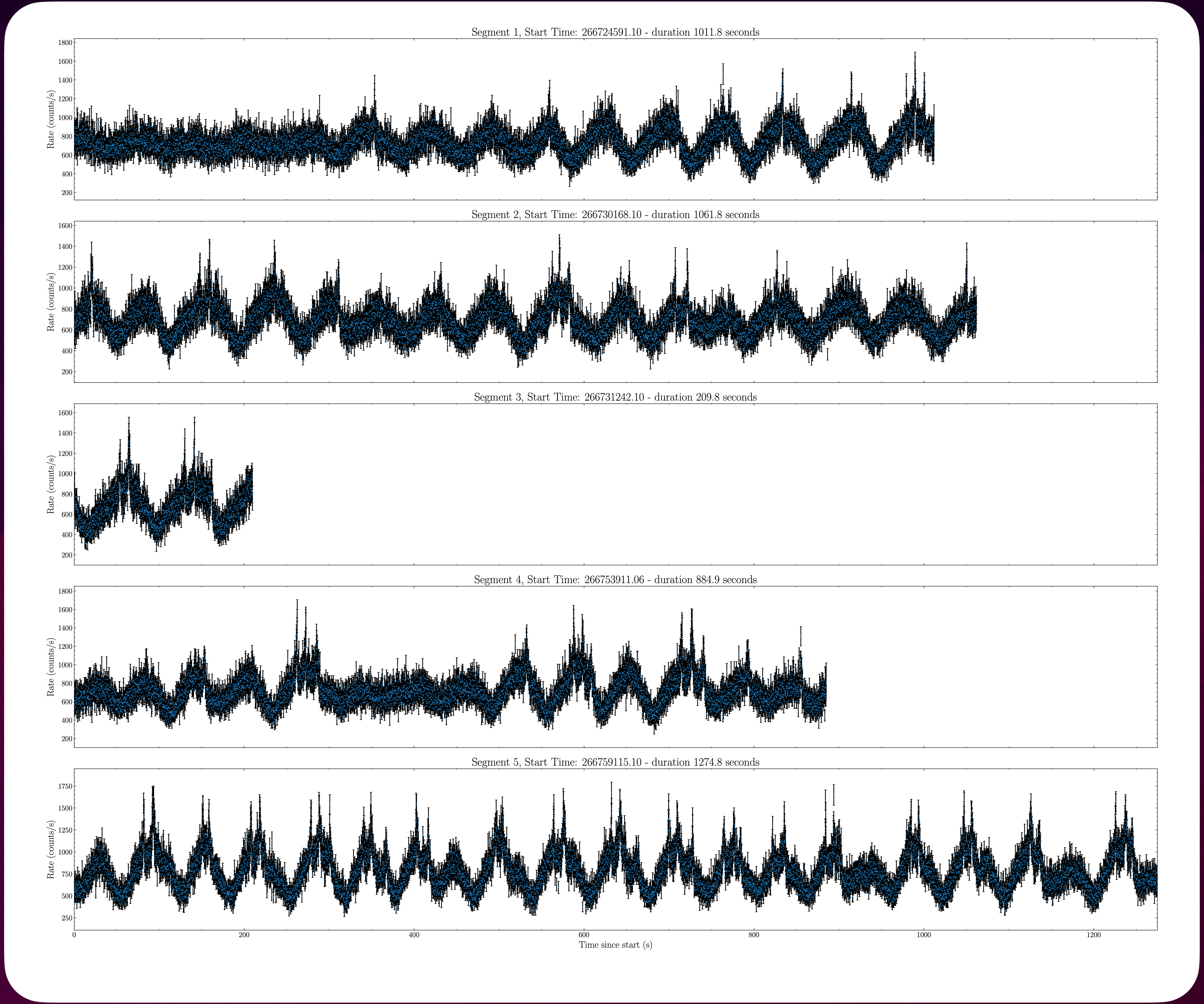
Single round inference with true data

Single round inference with true data



Example : time-resolved spectroscopy

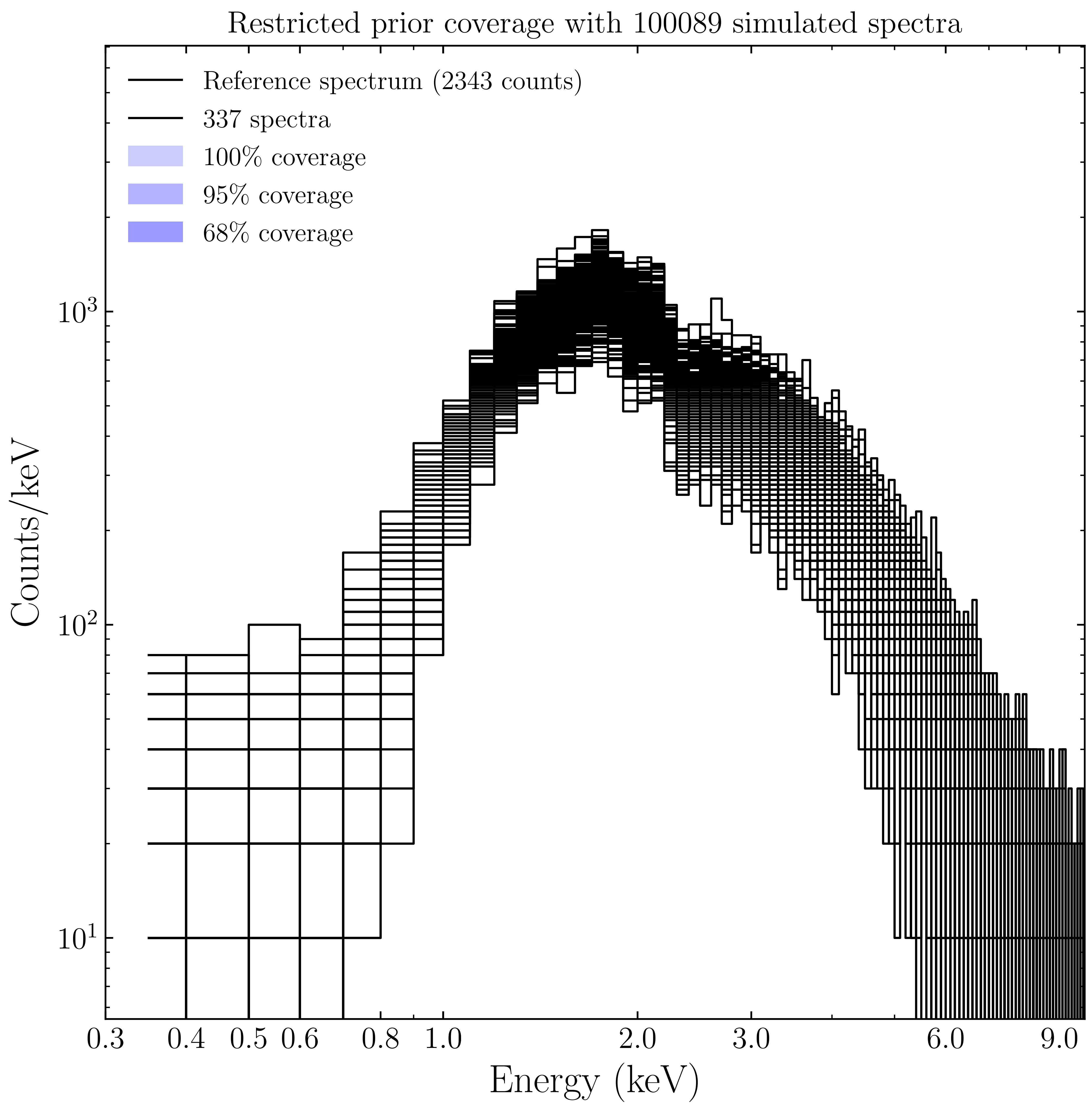
Single round inference with true data



Example : time-resolved spectroscopy

- Split observation of bright sources in hundreds/thousands of smaller observations and study the variability

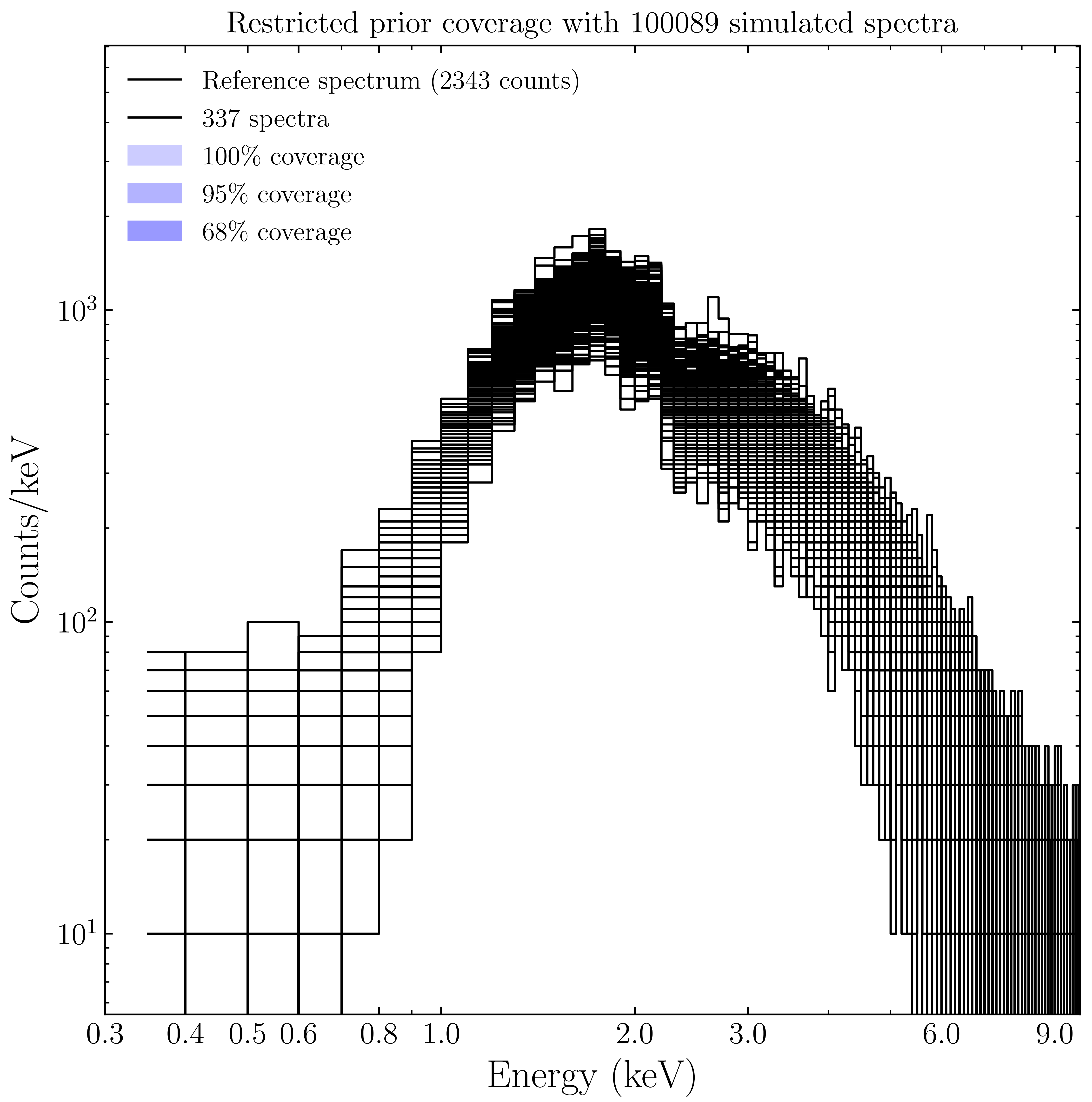
Single round inference with true data



Example : time-resolved spectroscopy

- Split observation of bright sources in hundreds/thousands of smaller observations and study the variability
- Prior set to cover all the observations

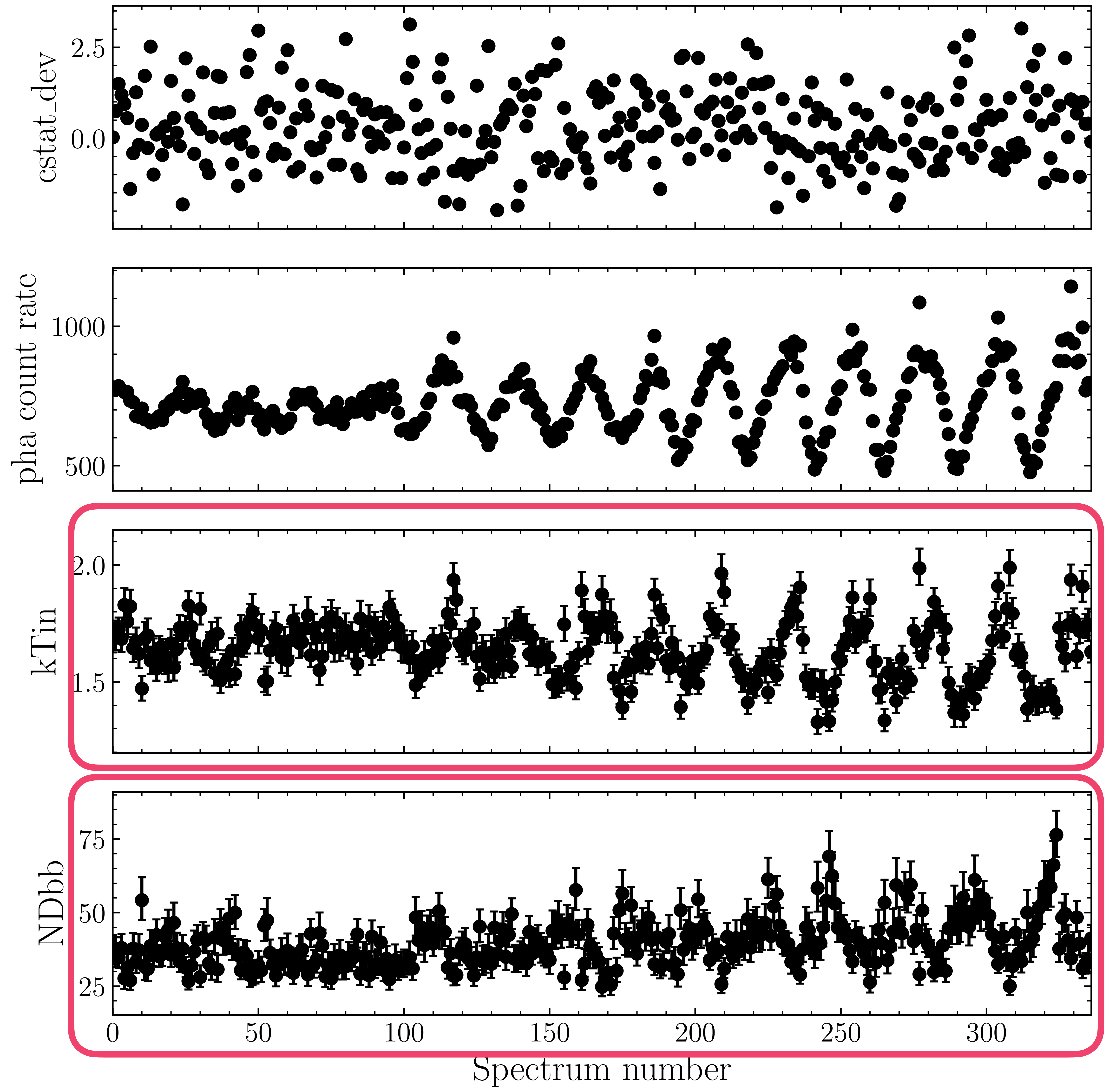
Single round inference with true data



Example : time-resolved spectroscopy

- Split observation of bright sources in hundreds/thousands of smaller observations and study the variability
- Prior set to cover all the observations
- Training a 3 parameter model using 10^5simulations (absorbed thermal emission from an accretion disk)

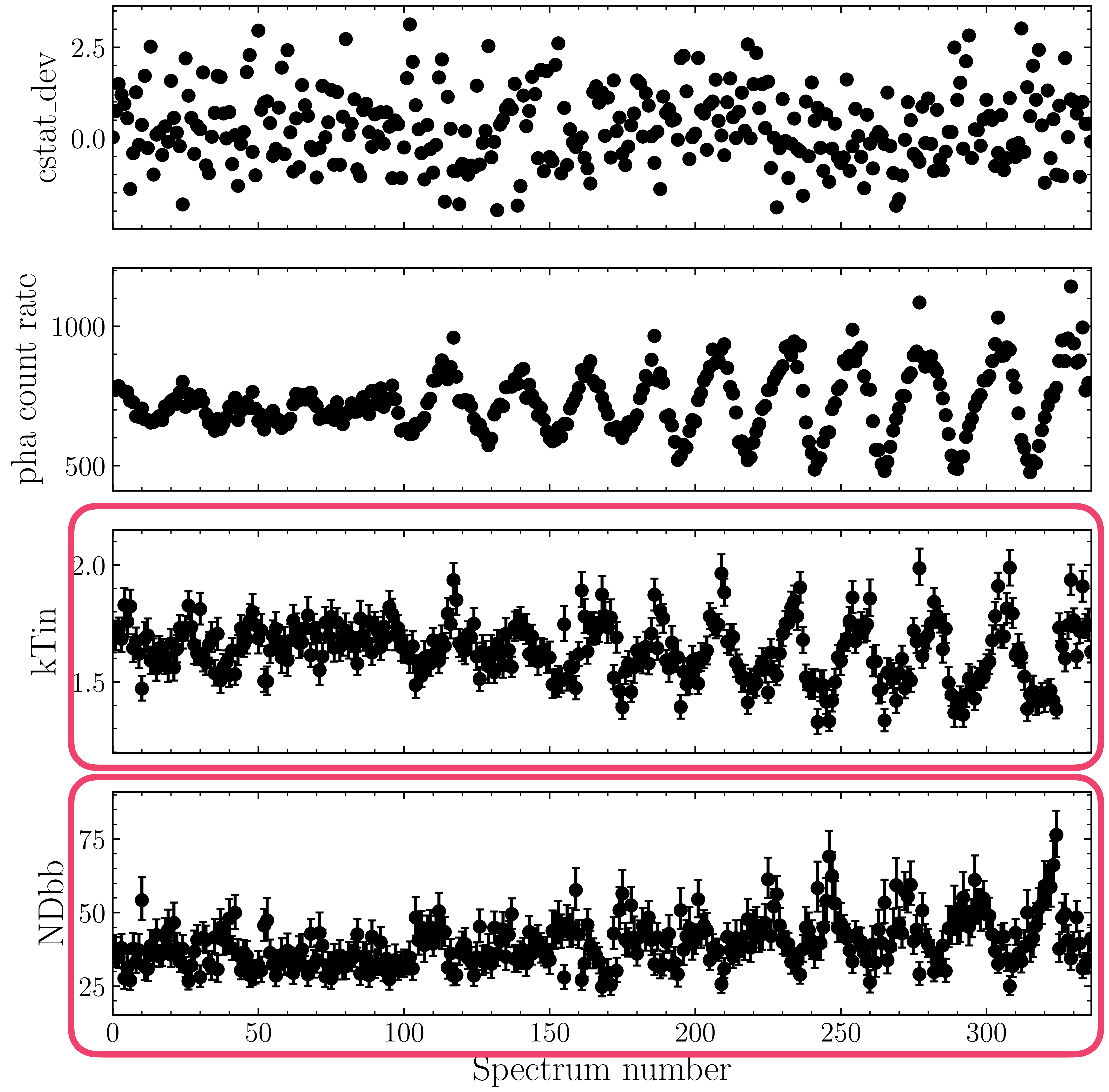
Single round inference with true data



Example : time-resolved spectroscopy

- Split observation of bright sources in hundreds/thousands of smaller observations and study the variability
- Prior set to cover all the observations
- Training a 3 parameter model using 10^5simulations (absorbed thermal emission from an accretion disk)
- 2min of training + 40sec of inference

Single round inference with true data

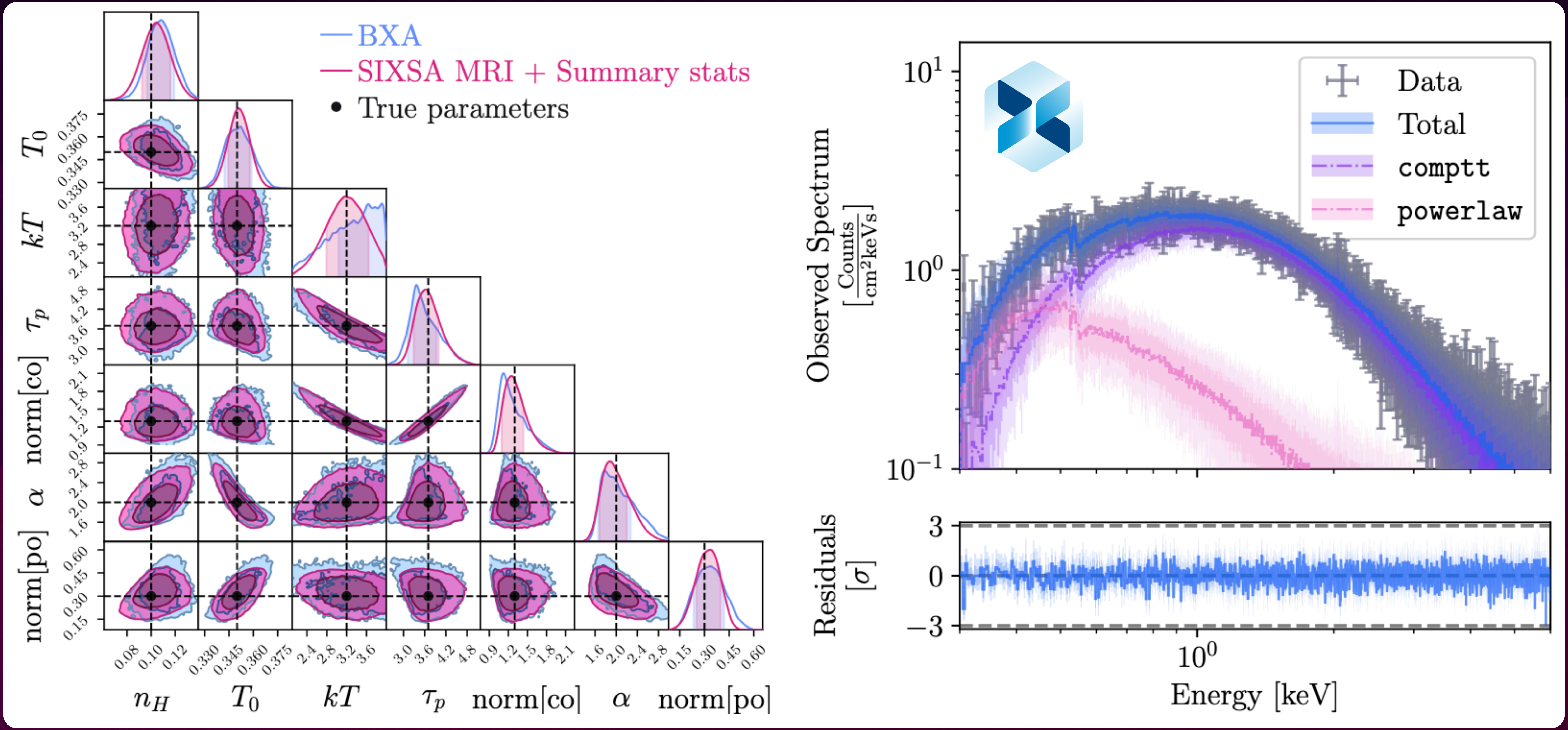


Example : time-resolved spectroscopy

- Split observation of bright sources in hundreds/thousands of smaller observations and study the variability
- Prior set to cover all the observations
- Training a 3 parameter model using 10^5simulations (absorbed thermal emission from an accretion disk)
- 2min of training + 40sec of inference

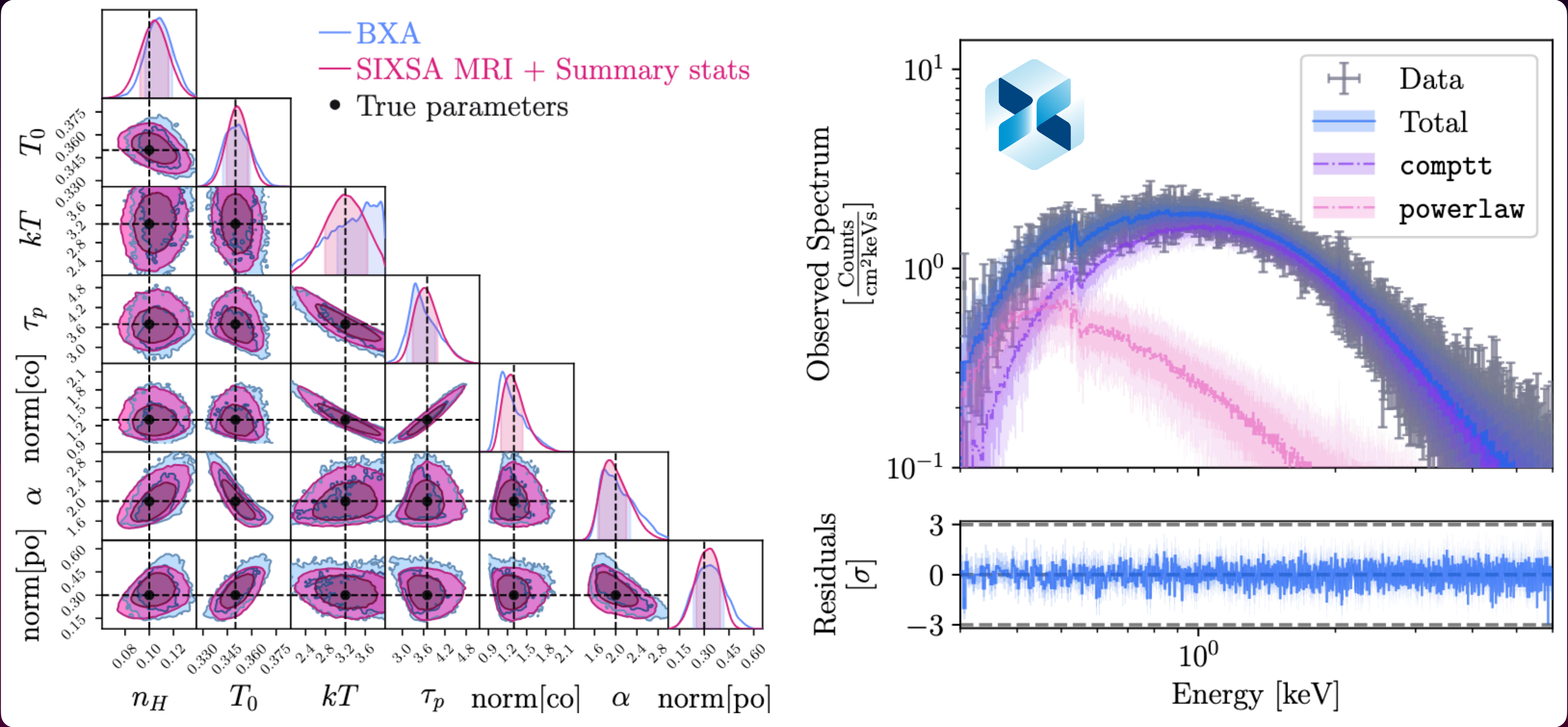
Up to 3 orders of magnitude faster than comparable methods if applied on a full X-IFU FOV

Multiple round inference with X-IFU



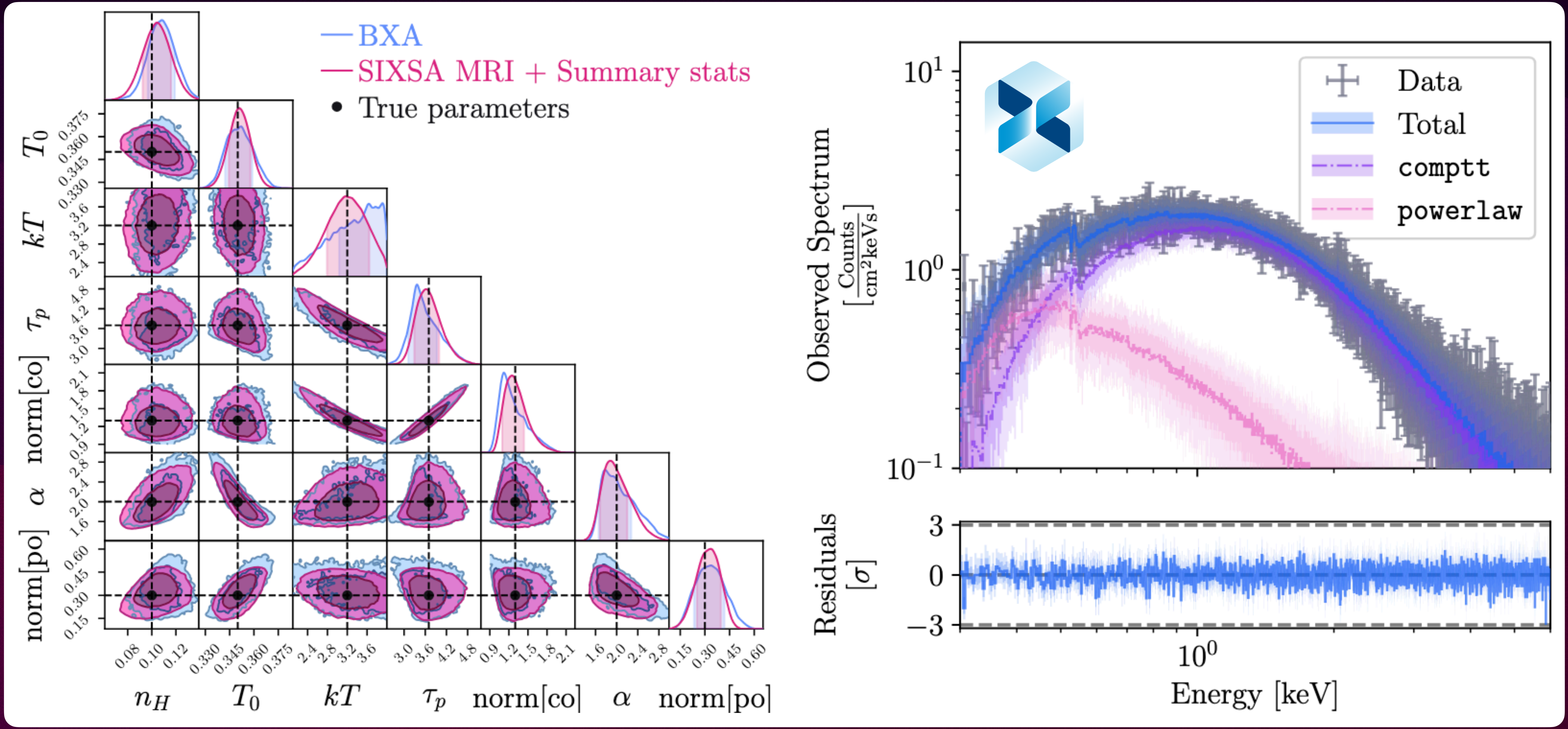
- Two components spectrum with X-IFU on a low-count regime (7 parameters)

Multiple round inference with X-IFU

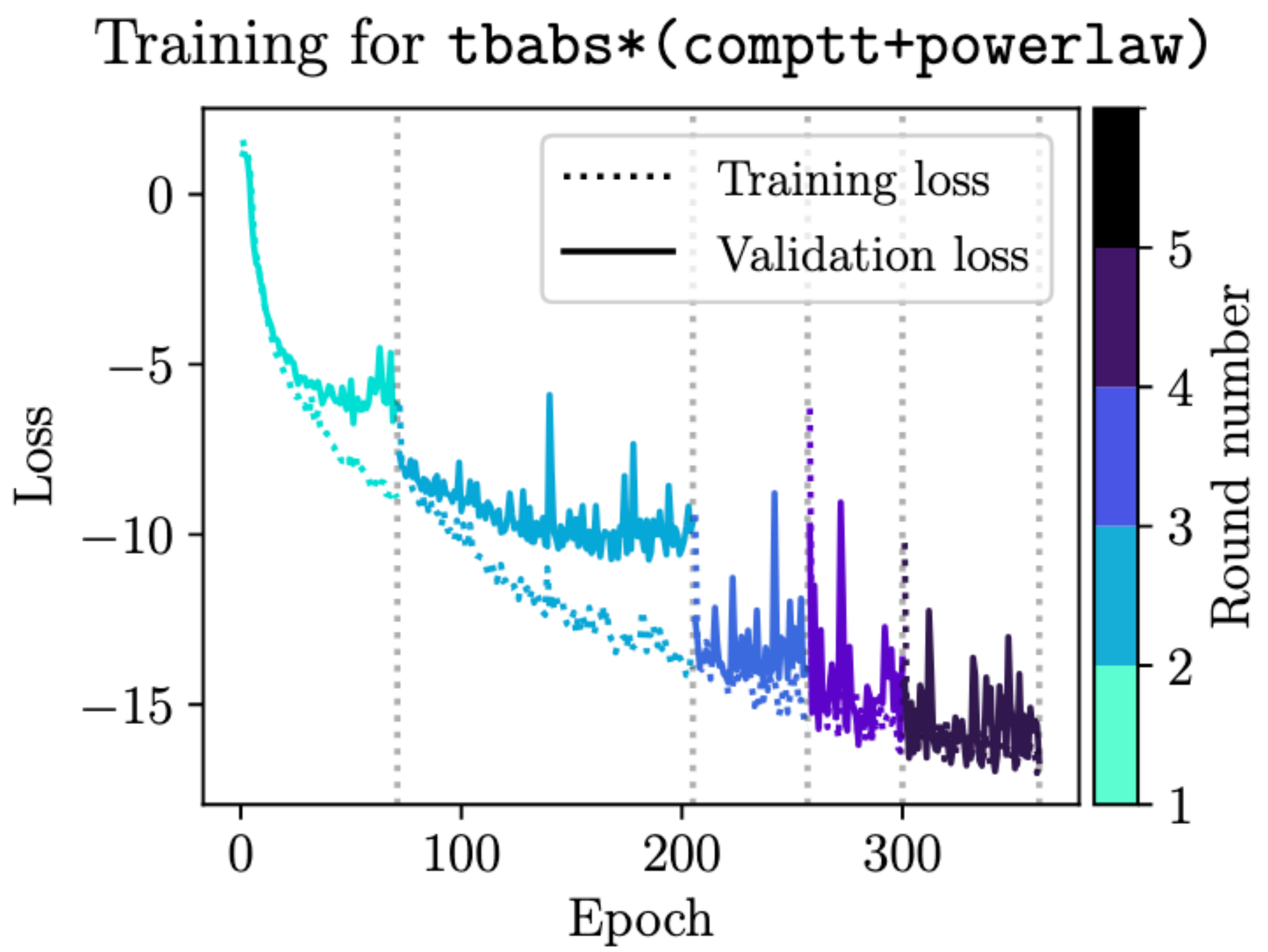


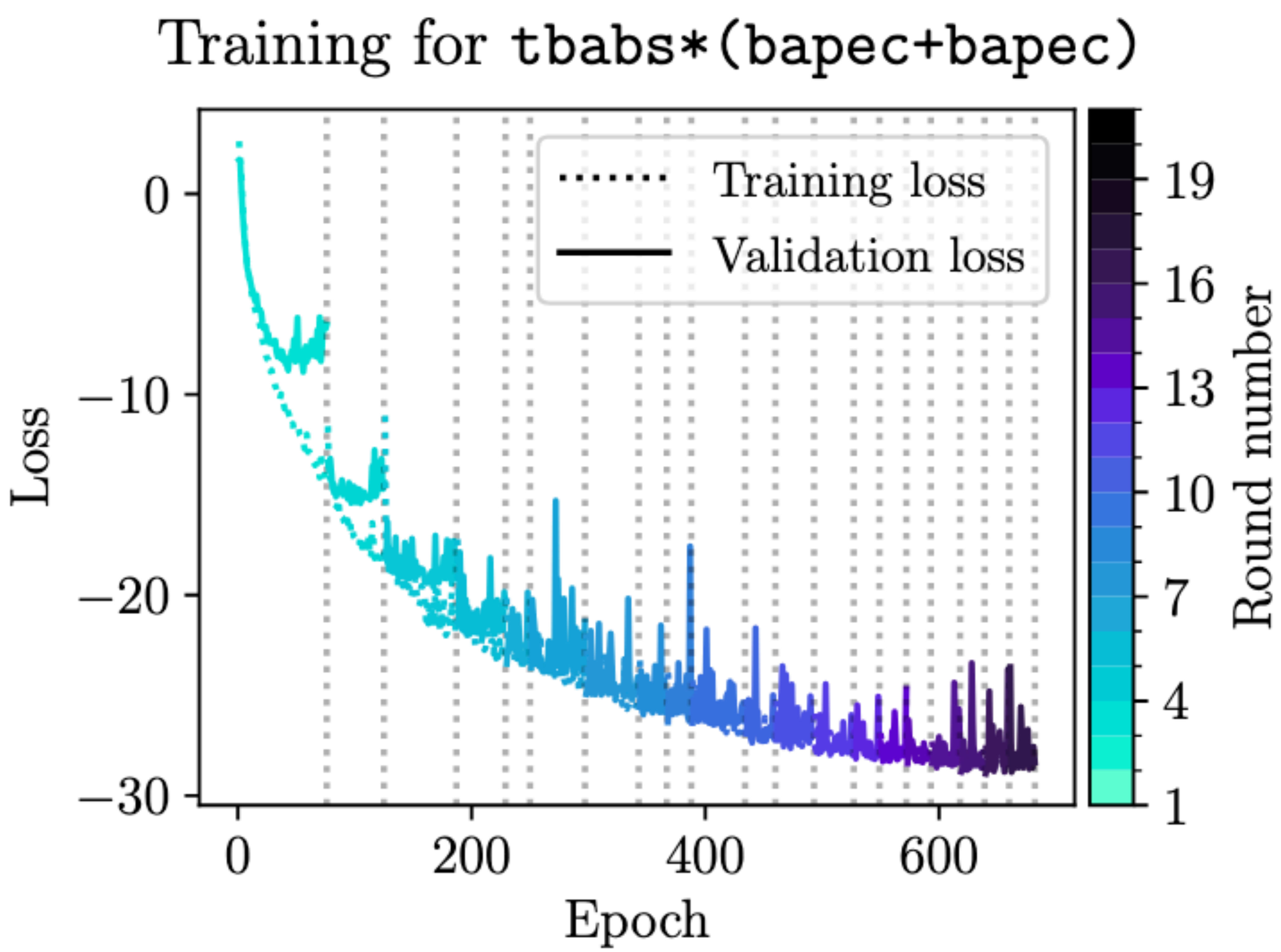
- Two components spectrum with X-IFU on a low-count regime (7 parameters)
- MRI on high resolution data is equally performant as SOTA for Bayesian Inference

Multiple round inference with X-IFU



- Two components spectrum with X-IFU on a low-count regime (7 parameters)
- MRI on high resolution data is equally performant as SOTA for Bayesian Inference
- It requires ~250 few simulations than SOTA and is ~2 order of magnitude faster


Training for tbabs*(comptt+powerlaw)
Training loss
Validation loss
Loss
Epoch
Round number
0
−5
−10
−15
0
100
200
300
1
2
3
4
5


Training for tbabs*(bapec+bapec)
Training loss
Validation loss
Loss
Epoch
Round number
0
−10
−20
−30
0
200
400
600
1
4
7
10
13
16
19