This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101034319 and from the European Union – NextGenerationEU.

NeuroMCT: Fast Monte Carlo Tuning with Generative Machine Learning in the JUNO Experiment

Arsenii Gavrikov[a] on behalf of the JUNO Collaboration
University & INFN Padova, Italy; arsenii.gavrikov@pd.infn.it

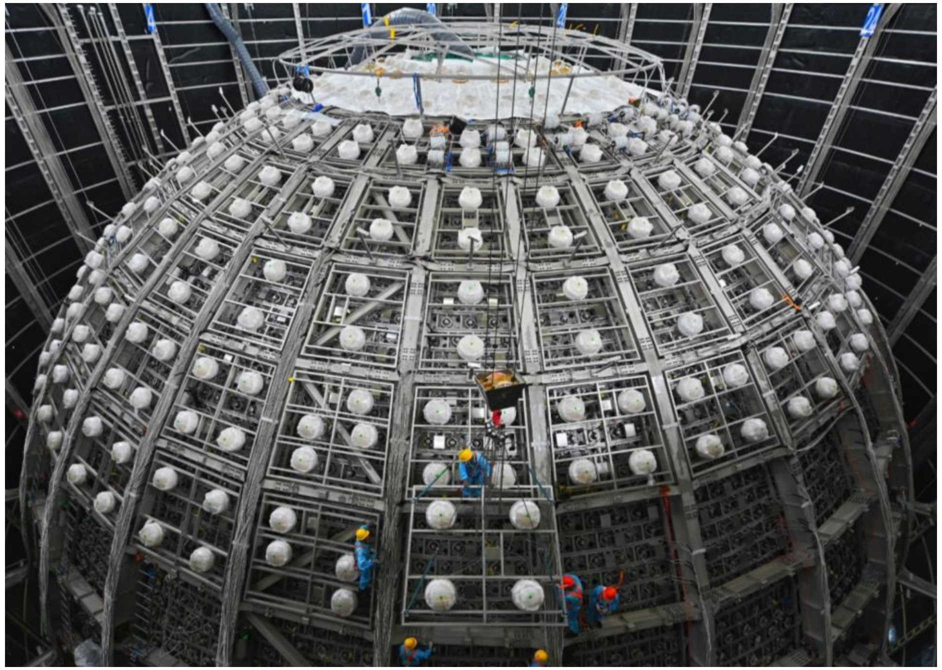
21/10/2024, Kraków, Poland
Conference on Computing in High Energy and Nuclear Physics 2024

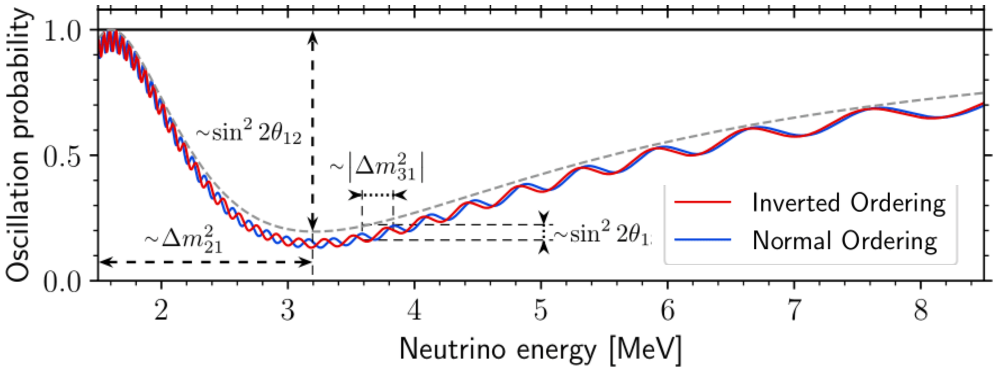
Introduction

The JUNO experiment

More details in the backup

- Jiangmen Underground Neutrino Observatory (**JUNO**) [1]:
    - **multi-purpose neutrino experiment** located in China
    - made of **20 kt liquid scintillator (LS), acting both as:**
        - **the interaction medium**
        - **the detection medium**
    - ~78% photo-coverage by **photo-multiplier tubes**
    - at the **latest stage** of its construction
- Neutrino **energy is measured through calorimetry** of final state leptons
- Main goals of JUNO [2, 3]:
    - **neutrino mass ordering** with **3σ** in ~6-7 years of data-taking
    - **sub-percent measurements** of the following **oscillation parameters:** $\sin^2\theta_{12}, \Delta m^2_{21}, \Delta m^2_{31}$
- The goals require to keep **energy-related systematic uncertainties below 1%**





[1] JUNO Collaboration 2016 *J. Phys. G: Nucl. Part. Phys.* **43** 030401
[2] JUNO Collaboration 2022 *Chinese Phys. C* **46** 123001
[3] JUNO Collaboration 2024 *arXiv:2405.18008*

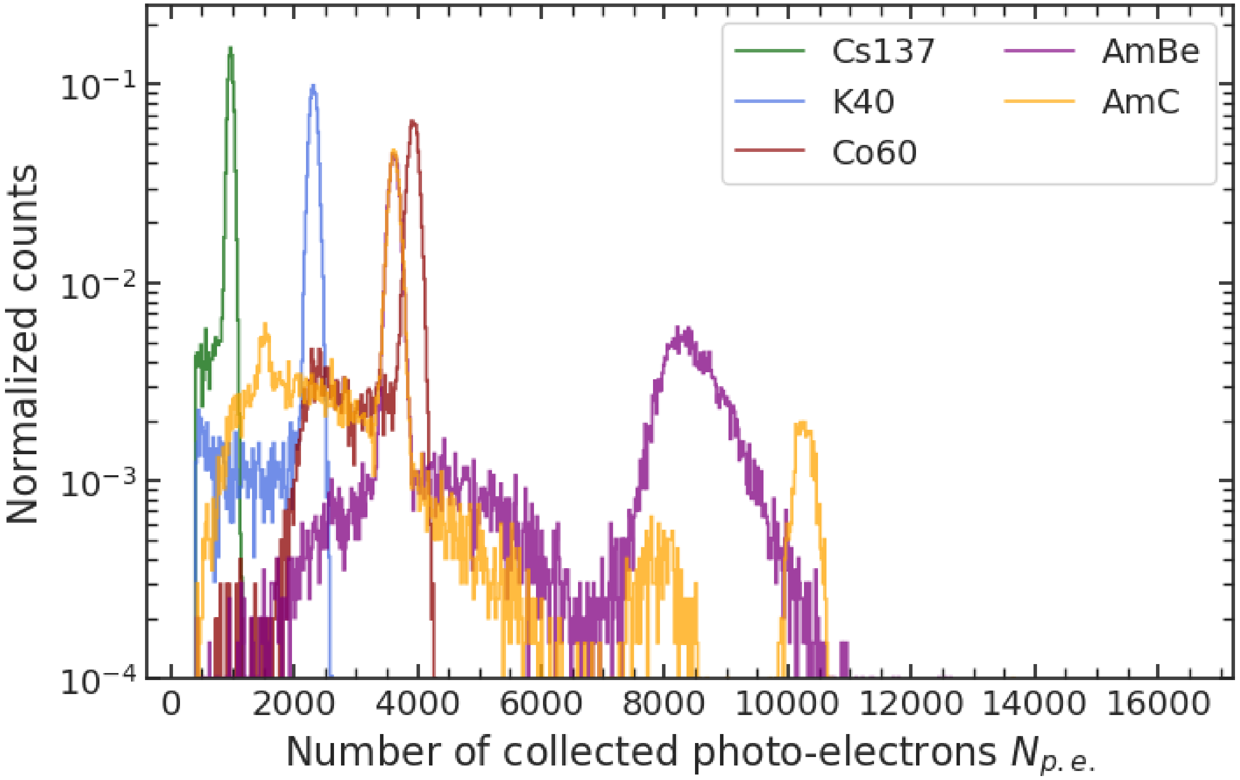
Physics challenge

- **LS emits visible light** when ionized by crossing charged particles
- **Relation between light detected** (NPE) and **energy deposited** in the LS is described **by several parameters [1]**
- **Tuning these parameters** to have JUNO MC matching real data is **pivotal to control systematic uncertainties**
- We use JUNO calibration campaign to tune the parameters

MC tuning:
Adjusting key parameters of the LS in the simulation

Monte Carlo (MC) simulation software

5 calibration sources*



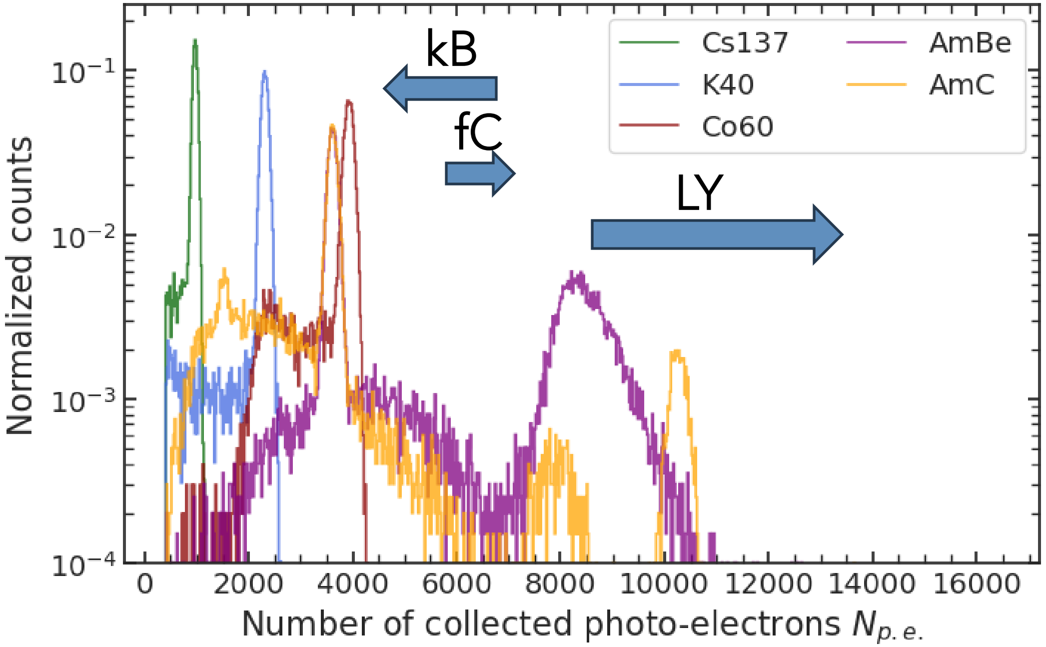
*more details in the backup

[1] JUNO Collaboration 2024 arXiv:2405.17860

MC tuning strategies

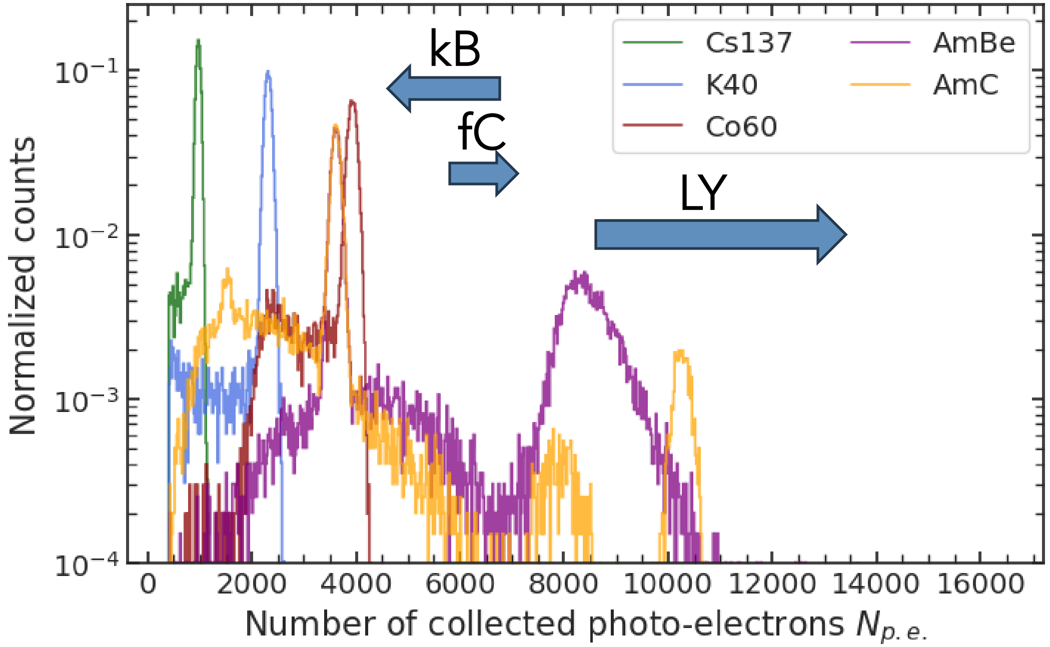
MC tuning strategies

- **The key parameters** of the LS to be tuned:
  - the Birks' constant: **kB** (a material-dependent quenching factor)
  - Cherenkov yield factor: **fC** (an effective parameter to adjust Cherenkov light yield)
  - Scintillation light yield per 1 MeV: **LY**
- All the parameters are highly correlated and so **multiple calibration sources are adopted** to break the correlations



MC tuning strategies

- **The key parameters** of the LS to be tuned:
  - the Birks' constant: **kB** (a material-dependent quenching factor)
  - Cherenkov yield factor: **fC** (an effective parameter to adjust Cherenkov light yield)
  - Scintillation light yield per 1 MeV: **LY**
- All the parameters are highly correlated and so **multiple calibration sources are adopted** to break the correlations



- **How to tune** the parameters?
- Find the set of parameters **minimizing distance** (chi2, likelihood) **between simulated calibration data and a reference dataset**
- Best parameter values found **through a fit** (optimizer / sampler)

MC tuning strategies

- **How to tune** the parameters?
- **The most straightforward approach** would be...

At **each step** of the fit, **we generate a lot of events with full MC** to build spectra of the sources with the corresponding $kB_g\ fC_g\ LY_g$

Full MC calib. data with:
$kB_g\ fC_g\ LY_g$

Real calib. data with:
$kB_r,\ fC_r,\ LY_r$

Fit
$D(\text{full MC data, real data}) \longrightarrow \min$

Best parameters

MC tuning strategies

- **How to tune** the parameters?
- **The most straightforward approach** would be…

At **each step** of the fit, **we generate a lot of events with full MC** to build spectra of the sources with the corresponding $kB_g\ fC_g\ LY_g$

Full MC calib. data with:
$kB_g\ fC_g\ LY_g$

Real calib. data with:
$kB_r,\ fC_r,\ LY_r$

Fit
D(full MC data, real data) $\longrightarrow$ min

Best parameters

Impractical and slow. Can we replace it with a surrogate model?

MC tuning strategies

Full MC samples with **diff. values** of parameters

- How **effectively and precisely** estimate the parameters?
- We propose **a fast MC tuning method** based on **Machine Learning** (ML):
  - Use a surrogate model to generate **artificial spectra to be compared with the reference spectra** during the fit

MC tuning strategies

Full MC samples with **diff. values** of parameters

Training

Fast ML generator of energy spectra ***E**** for **diff. values** of parameters

*Able to **interpolate** in the parameters space*

- How **effectively and precisely** estimate the parameters?
- We propose **a fast MC tuning method** based on **Machine Learning** (ML):
  - Use a surrogate model to generate **artificial spectra to be compared with the reference spectra** during the fit

**represented by amount of light collected*

MC tuning strategies

Full MC samples with **diff. values** of parameters

Training

Fast ML generator of energy spectra ***E*** * for **diff. values** of parameters

*Able to **interpolate** in the parameters space*

- How **effectively and precisely** estimate the parameters?
- We propose **a fast MC tuning method** based on **Machine Learning** (ML):
  - Use a surrogate model to generate **artificial spectra to be compared with the reference spectra** during the fit

Gen. calib. data with:
$kB_g\ fC_g\ LY_g$

Real calib. data with:
$kB_r,\ fC_r,\ LY_r$

*represented by amount of light collected

MC tuning strategies

- How **effectively and precisely** estimate the parameters?
- We propose **a fast MC tuning method** based on **Machine Learning** (ML):
  - Use a surrogate model to generate **artificial spectra to be compared with the reference spectra** during the fit

Full MC samples with **diff. values** of parameters

Training

Fast ML generator of energy spectra ***E**** for **diff. values** of parameters

*Able to **interpolate** in the parameters space*

Gen. calib. data with: $kB_g\ fC_g\ LY_g$

Real calib. data with: $kB_r,\ fC_r,\ LY_r$

At **each step** of the fit, **we generate a lot of events with an ML generator** to build spectra of the sources with the corresponding $kB_g\ fC_g\ LY_g$

Fit

$D(\text{gen. data, real data}) \longrightarrow \min$

**represented by amount of light collected*

MC tuning strategies

- How **effectively and precisely** estimate the parameters?
- We propose **a fast MC tuning method** based on **Machine Learning** (ML):
  - Use a surrogate model to generate **artificial spectra to be compared with the reference spectra** during the fit

Full MC samples with **diff. values** of parameters

Training

Fast ML generator of energy spectra ***E**** for **diff. values** of parameters

*Able to **interpolate** in the parameters space*

Gen. calib. data with: kB_g fC_g LY_g

Real calib. data with: kB_r, fC_r, LY_r

At **each step** of the fit, **we generate a lot of events with an ML generator** to build spectra of the sources with the corresponding kB_g fC_g LY_g

Fit

D(gen. data, real data) $\longrightarrow$ min

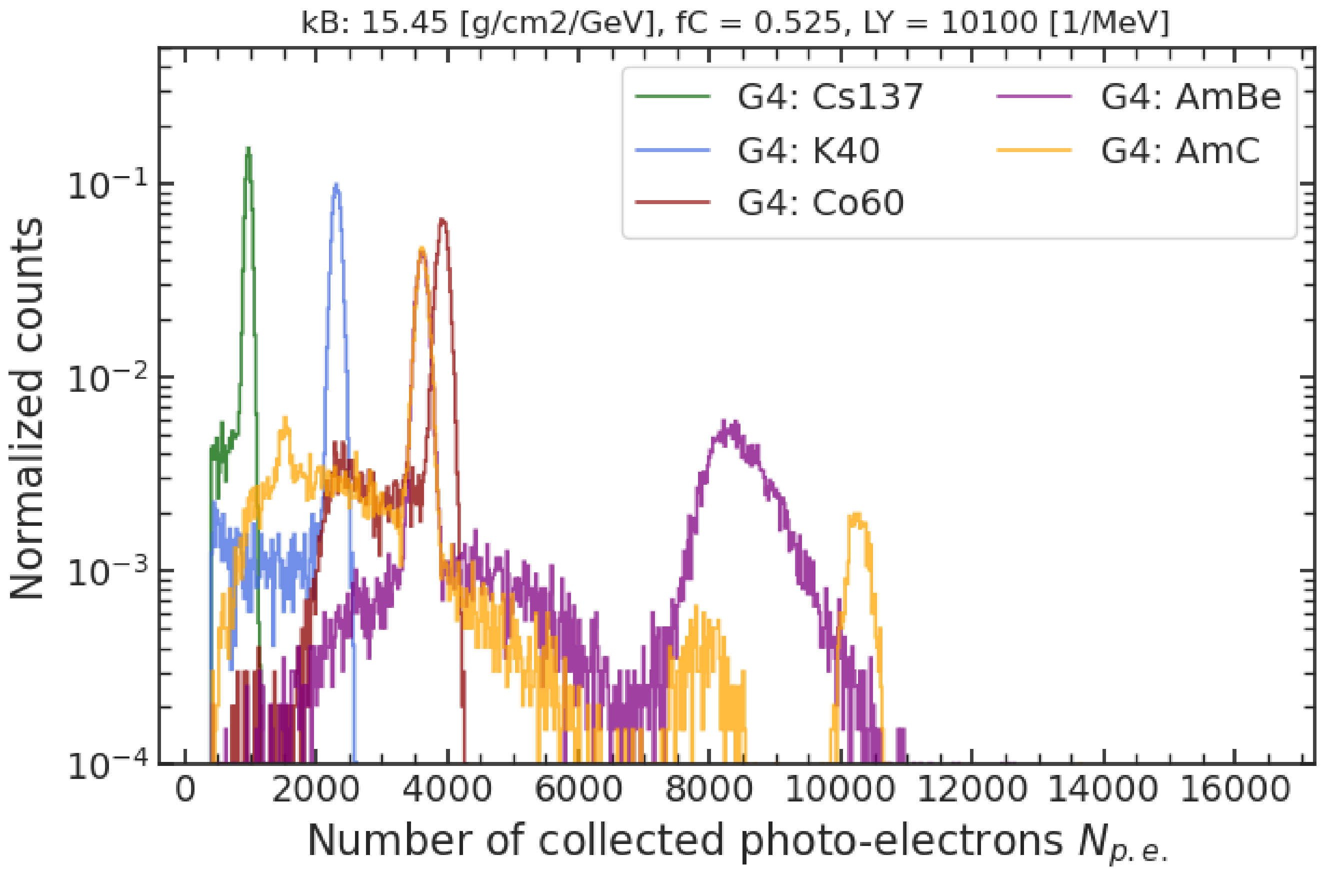
Best parameters

**represented by amount of light collected*

Data description

Data description

- JUNO employs sources emitting **neutrons and gammas at different energies**
- Each source is deployed alone and it results in an **energy spectrum measured in NPE**
- Spectra of all sources need to be analyzed simultaneously to **grasp LS energy response**



How parameters impact the calibration data?

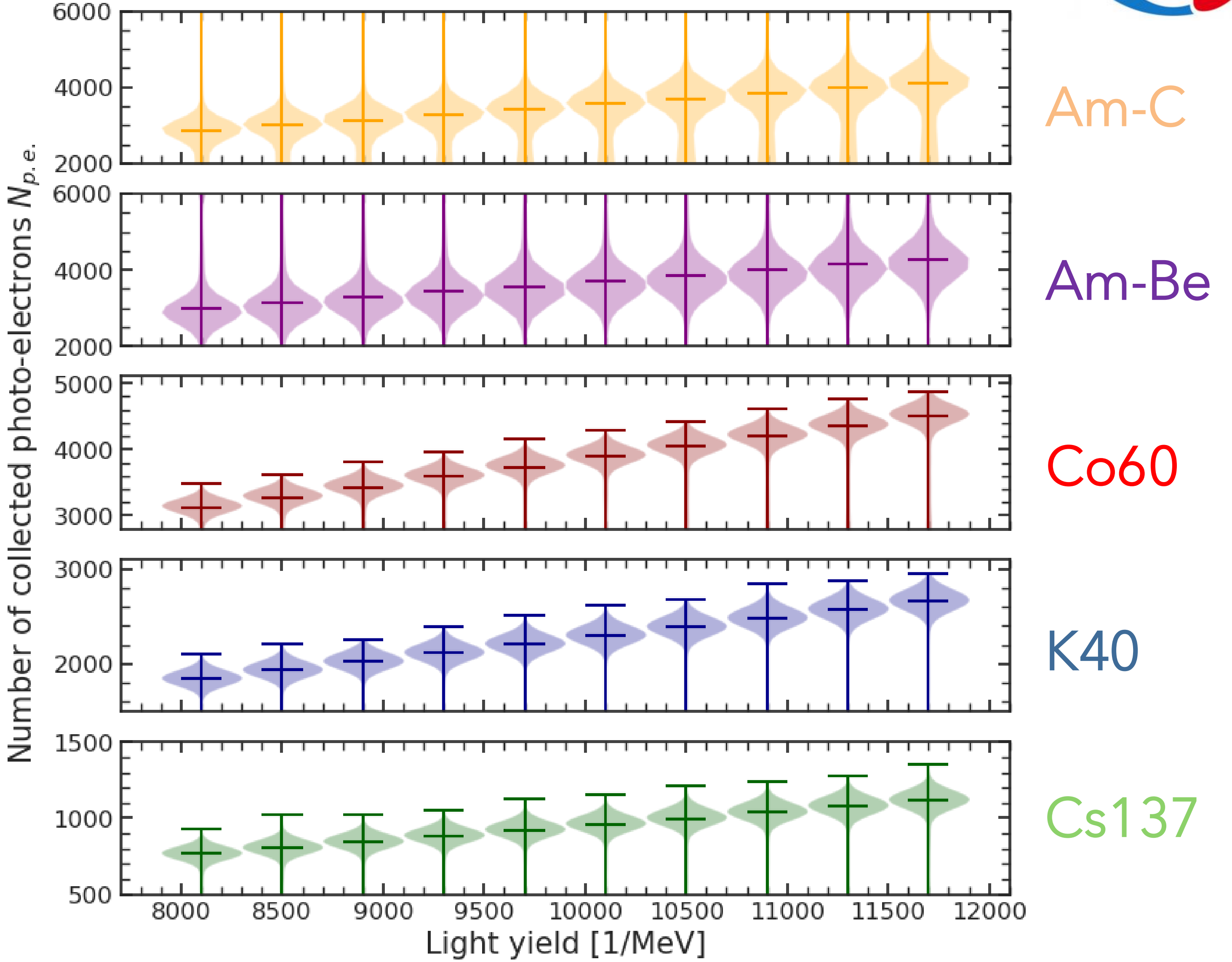
How LS parameters impact the calibration data

LY effect

- kB and fC are fixed:
  - kB = 15.45 [g/cm^2/GeV]
  - fC = 0.525
- LY is varying

- Light yield is the most influential parameter
- All sources are highly affected

Only main peaks are shown



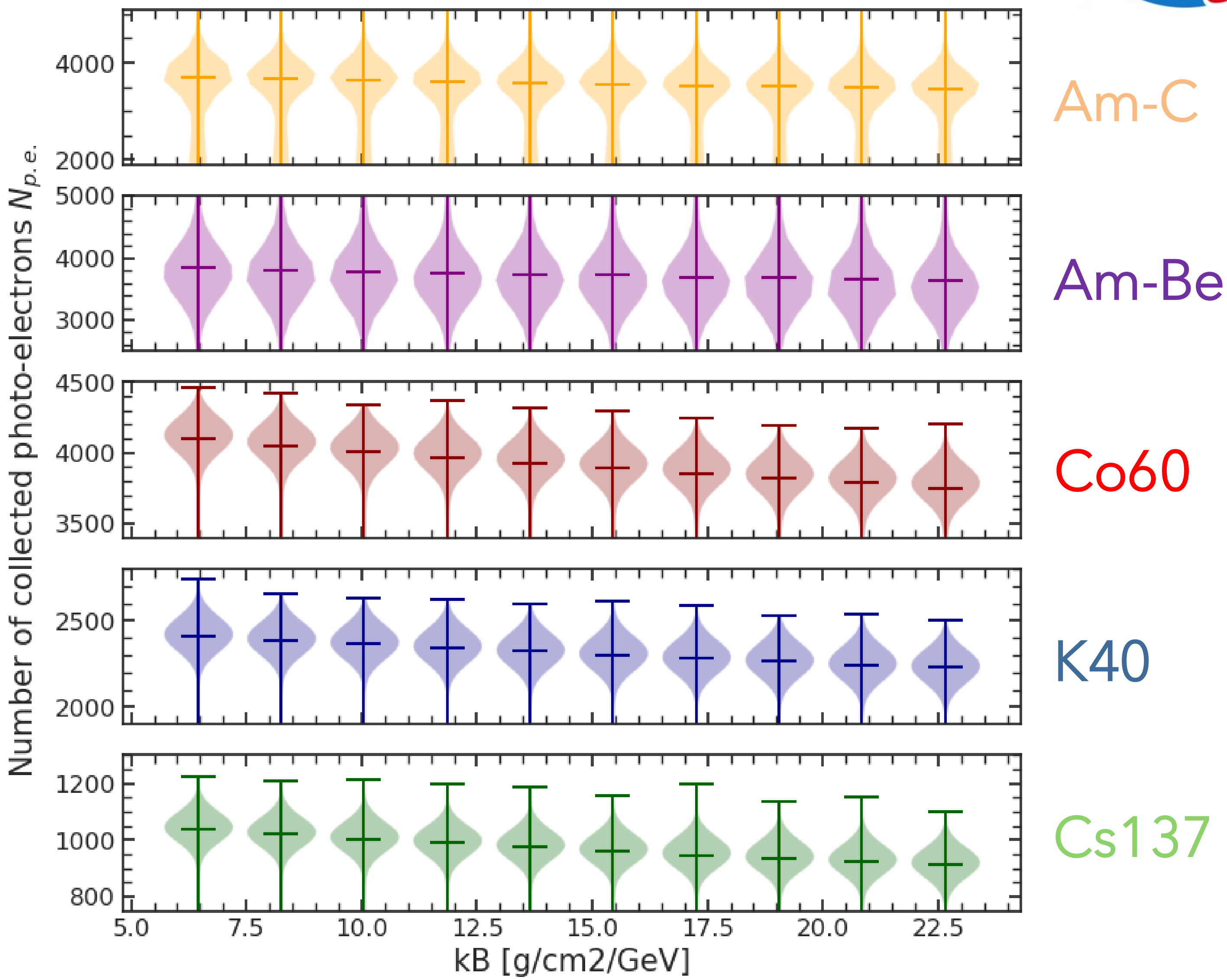
How LS parameters impact the calibration data

kB effect

- LY and fC are fixed:
  - LY = 10100 [1/MeV]
  - fC = 0.525
- kB is varying

- kB effect is smaller than LY and **anticorrelated with the photo peak**
- All sources are affected

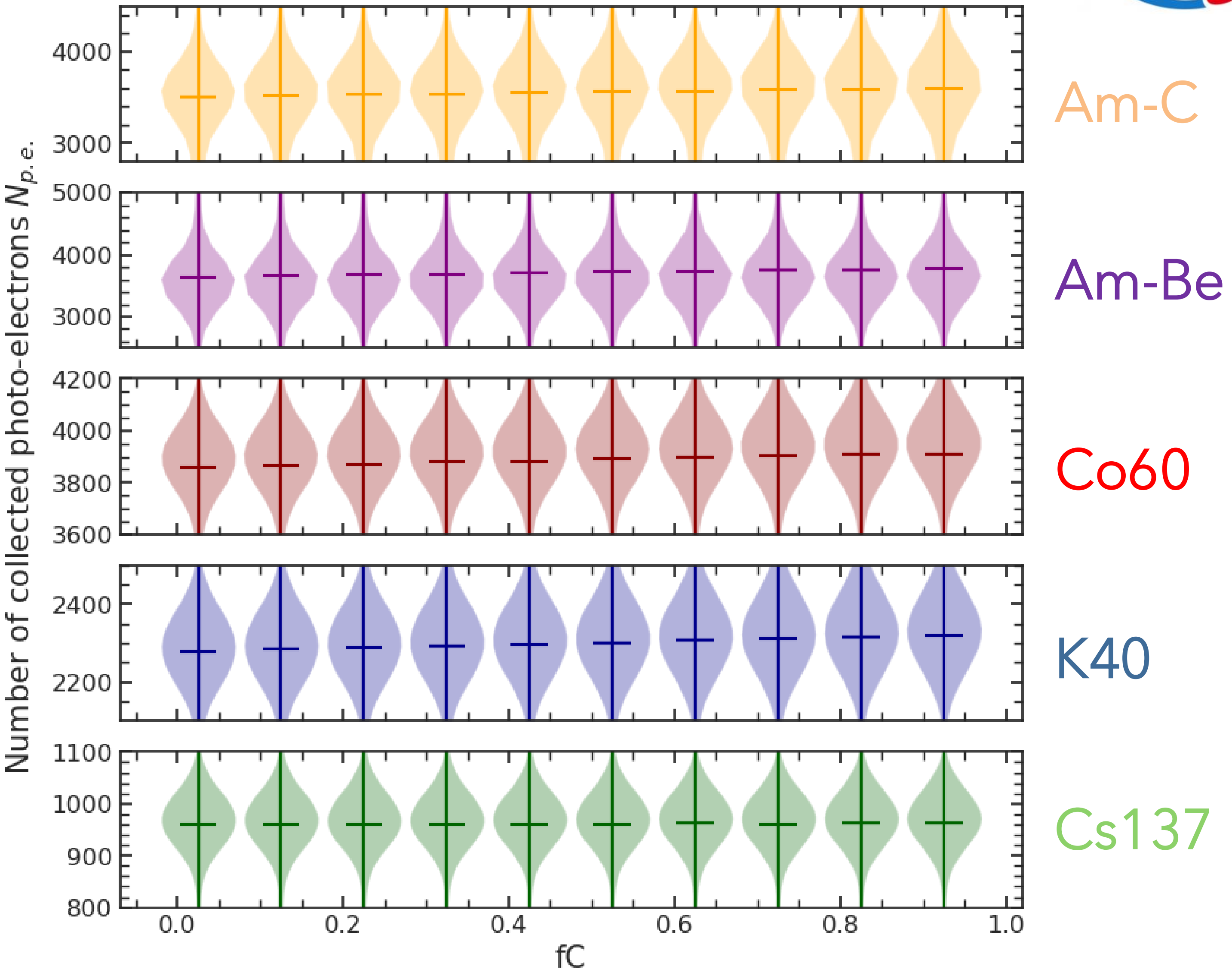
Only main peaks are shown



How LS parameters impact the calibration data

fC effect

- kB and LY are fixed:
  - kB = 15.45 [g/cm²/GeV]
  - LY = 10100 [1/MeV]
- fC is varying



- fC has **a minor effect** to the spectra
- Cs137 **is not affected at all**
- **Slight effect** for Co60 and K40

Only main peaks are shown

Data: training + validation

Data: training + validation

Huge dataset with full MC simulation:

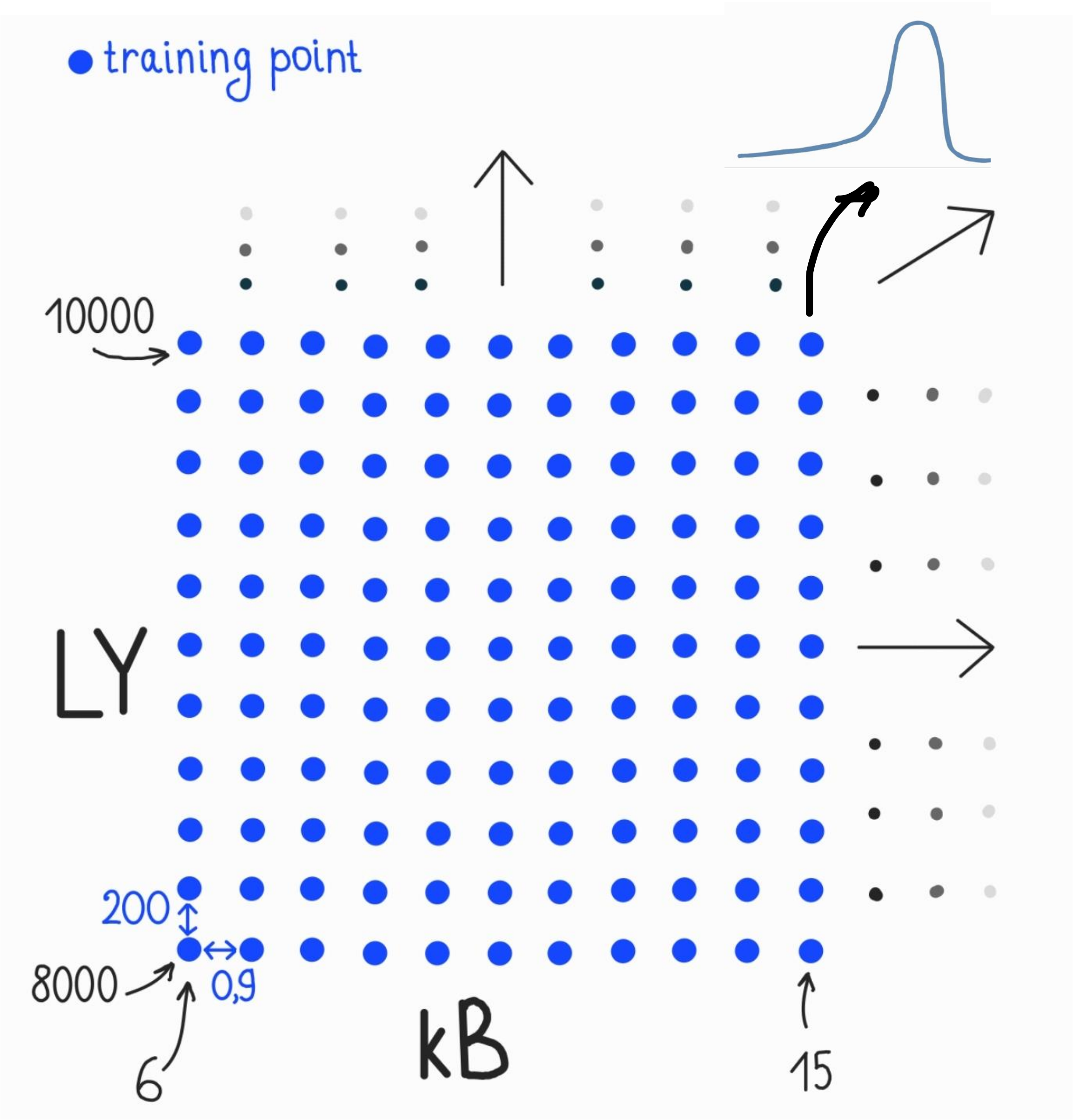
- Discrete grid of the parameters
- Per each of the sources:
    - Cs137; K40; Co60; AmBe; AmC

Training data; 21 points per param, 21^3 combinations:

1. kB: [6, 6.9, …, 24]
2. fC: [0, 0.05, …, 1]
3. LY: [8000, 8200, …, 12000]

- For each point **10k events**
- **~600M** events in total
- **A few millions** of CPU hours for the production

LY and kB example



Data: training + validation

Huge dataset with full MC simulation:

- Discrete grid of the parameters
- Per each of the sources:
    - Cs137; K40; Co60; AmBe; AmC

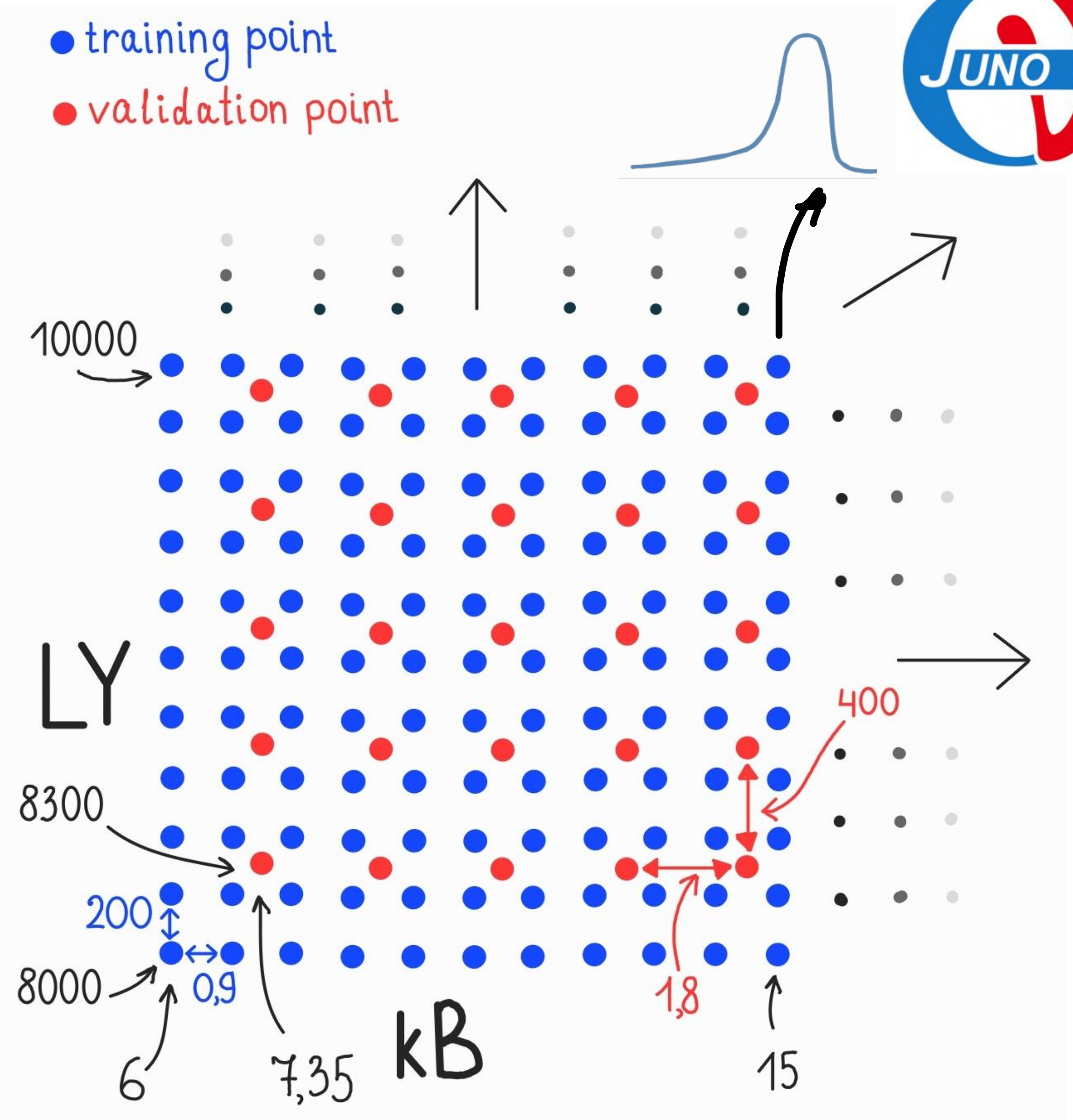
Training data; 21 points per param, 21^3 combinations:

1. kB: [6, 6.9, …, 24]
2. fC: [0, 0.05, …, 1]
3. LY: [8000, 8200, …, 12000]

Validation data; 10 points per param, 10^3 combinations:

1. kB: [7.35, 9.15, …, 24)
2. fC: [0.075, 0.175, …, 1)
3. LY: [8300, 8700, …, 12000)

this dataset is used to **validate** the model during training and to **optimize its hyperparameters**…



For each point **10k events**

Data: testing datasets

Testing the ML output (I)

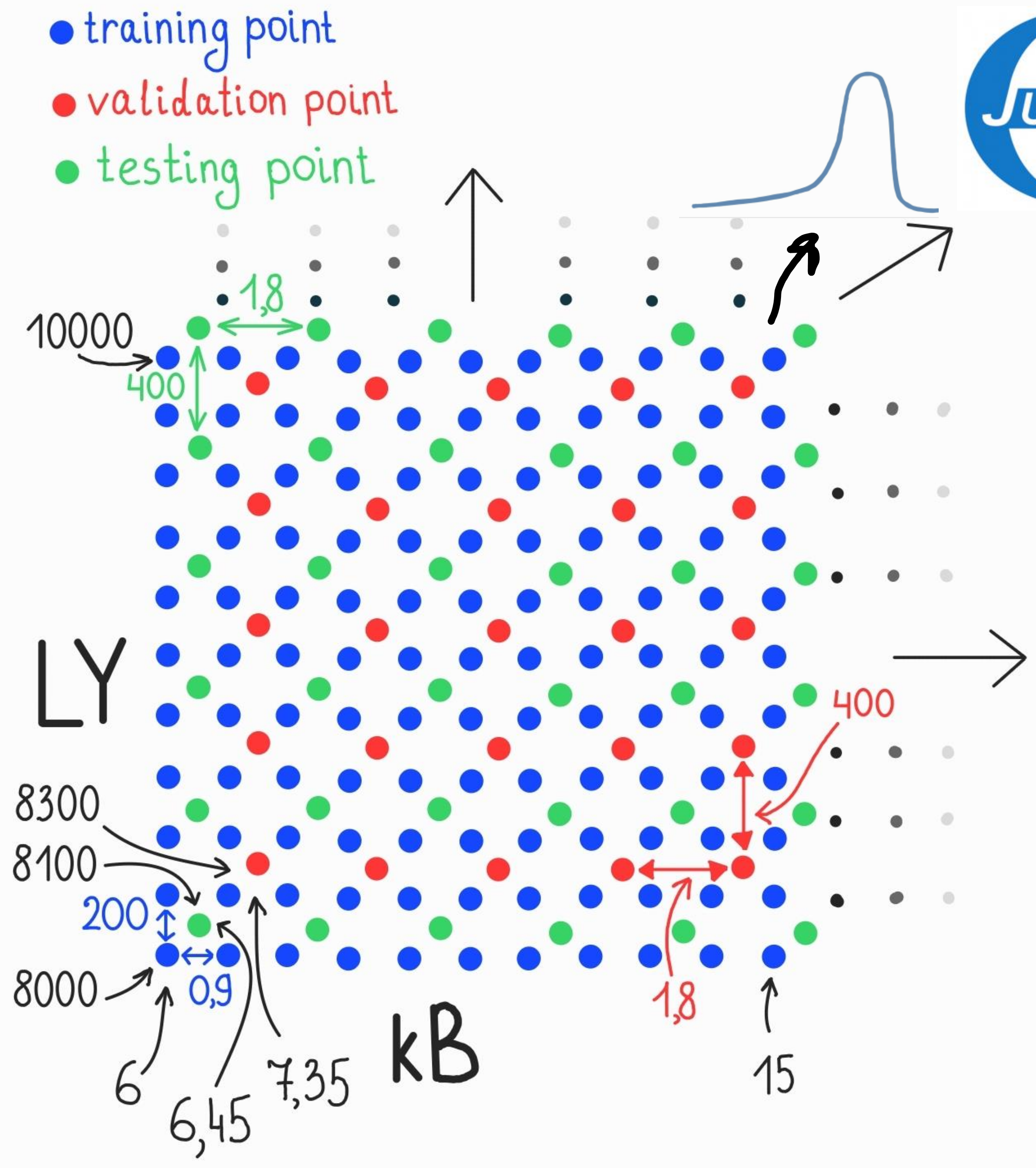
Huge dataset with full MC simulation:

- Discrete grid of the parameters
- Per each of the sources:
    - Cs137; K40; Co60; AmBe; AmC

Testing data №1; 10 points per param, 10^3 combinations:

1. kB: [6.45, 8.25, …, 24)
2. fC: [0.025, 0.125, …, 1)
3. LY: [8100, 8500, …, 12000)

this dataset is used to check the bias of the model across all the points of the grid…



For each point **10k events**

Testing the ML output (II)

Huge dataset with full MC simulation:

- Discrete grid of the parameters
- Per each of the sources:
  - Cs137; K40; Co60; AmBe; AmC

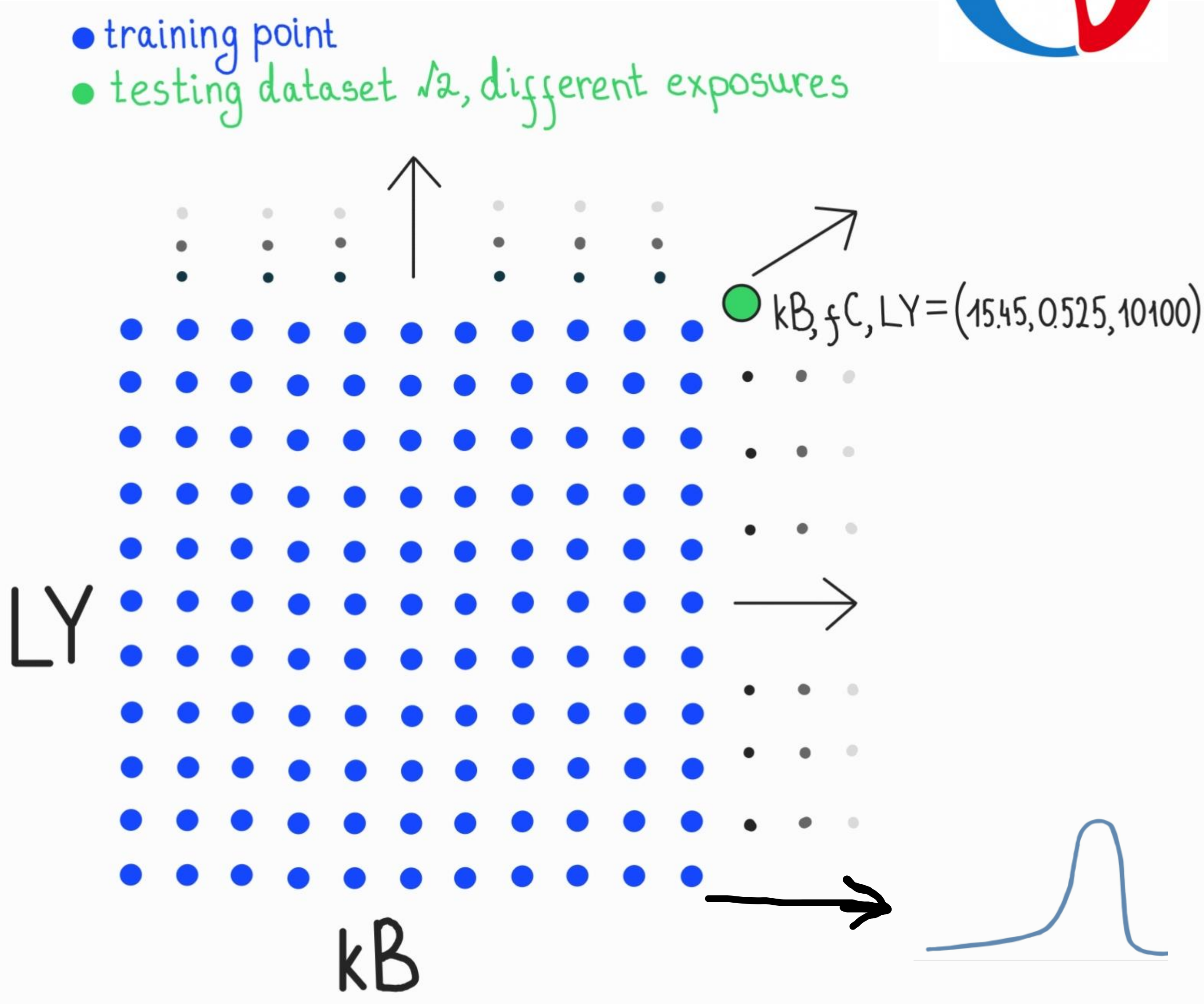
Testing data **№**2; a single point:

1. kB: 15.45 [g/cm^2/GeV]
2. fC: 0.525
3. LY: 10100 [1 / MeV]

Different exposures in numbers of events per source:

- 1k; 2k; 5k; 10k; 25k
- 1k datasets with diff. seeds per each exposure

this dataset is used to perform the systematic uncertainty analysis of the model...



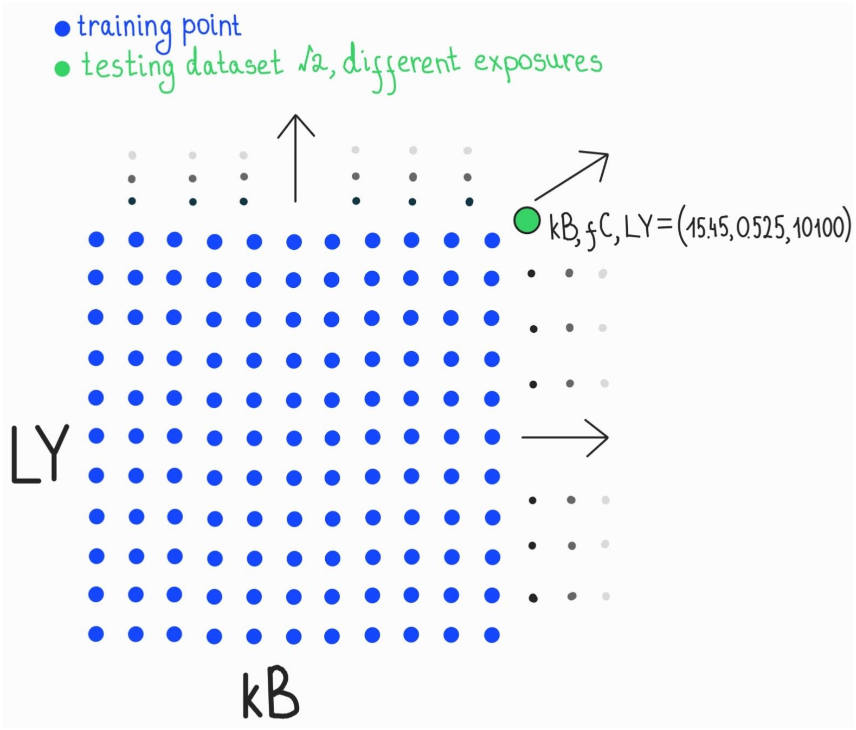
Testing the ML output (II)

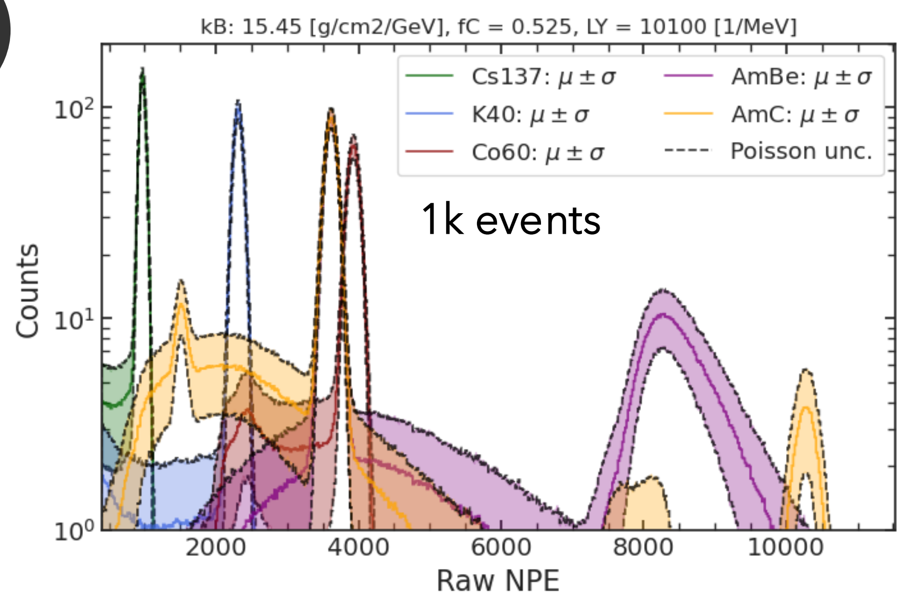
Testing data №2; a single point:

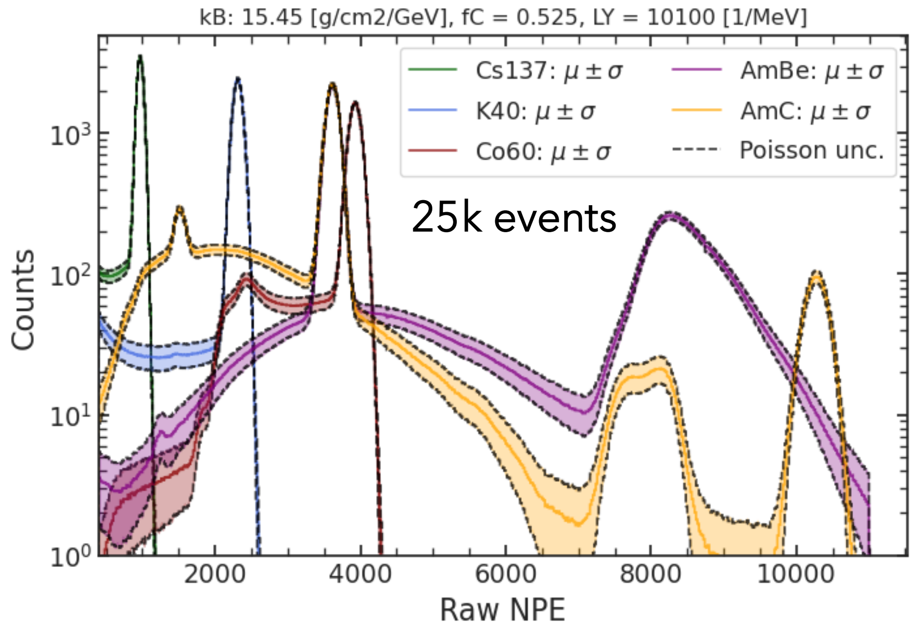
1. kB: 15.45 [g/cm^2/GeV]
2. fC: 0.525
3. LY: 10100 [1 / MeV]

Different exposures in numbers of events per source:

- 1k; 2k; 5k; 10k; 25k
- 1k datasets with diff. seeds per each exposure







ML models

ML models

learns unique mapping between the three parameters and a source type and an event rate λ_i in each bin

+ fast and reliable model
- requires *pre-defined binning*

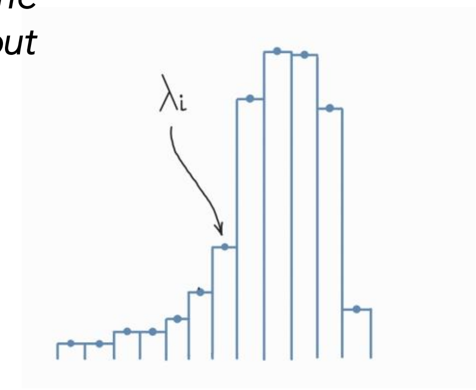
Conditions:
kB, fC, LY + source type S

Multi-output regressor

Aims to directly learn **a mapping** from the parameters and a source type to an event rate λ_i

We use a small **Transformer-based** model as **the Regressor**

*Produced spectra is always **the same for the same input parameters***



ML models

learns unique mapping between the three parameters and a source type and an event rate λ_i in each bin

+ fast and reliable model
- requires pre-defined binning

Conditions:
kB, fC, LY + source type S

Noise vector
(usually, multidim. normal distribution)

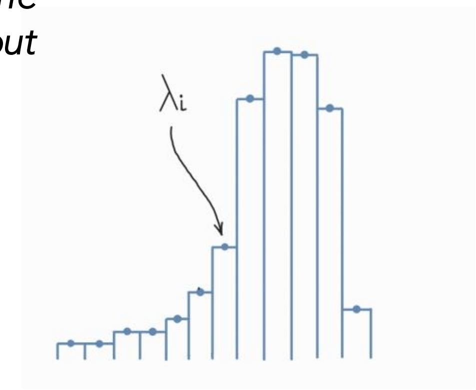
+ potentially better generalize
+ posterior
+ no pre-defined binning

Multi-output regressor

Aims to directly learn **a mapping** from the parameters and a source type to an event rate λ_i

We use a small **Transformer-based** model as **the Regressor**

*Produced spectra is always **the same for the same input parameters***



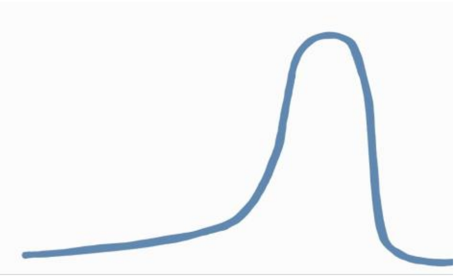
ML generator

Aims to learn the **conditional probability** of the energies* for a given set of parameters and a source type:

$$P(\vec{E} \mid \vec{kB}, \vec{fC}, \vec{LY}, \vec{S})$$

We use Generative Adversarial Networks (**GAN**) as the ML generator

Sampling energies under conditions: $\{kB_i, fC_i, LY_i, S_i\}$

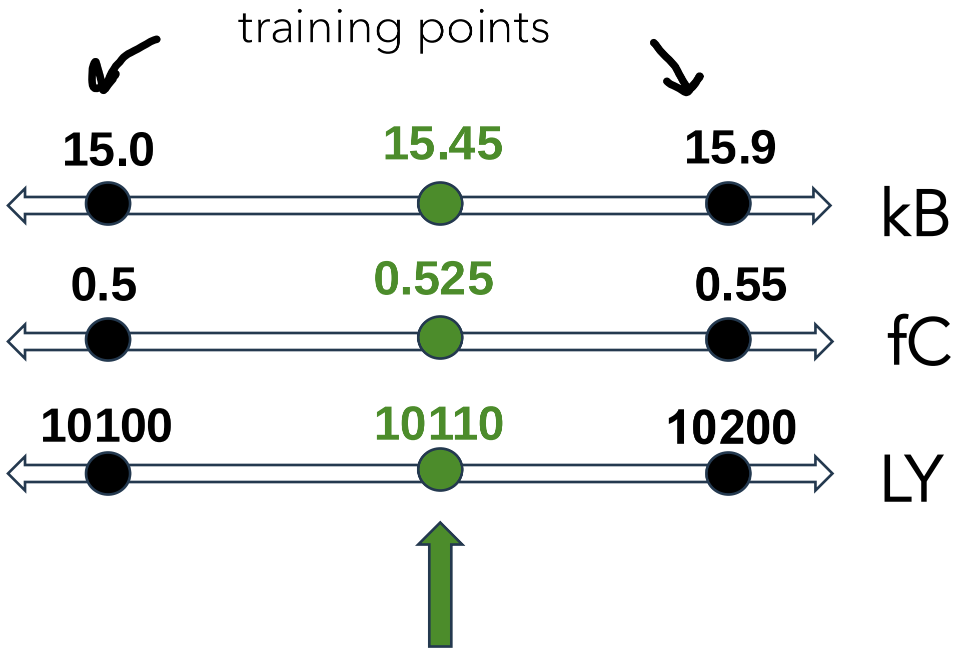


Final goal
As an Intermediate step: producing rates

*represented by amount of light collected

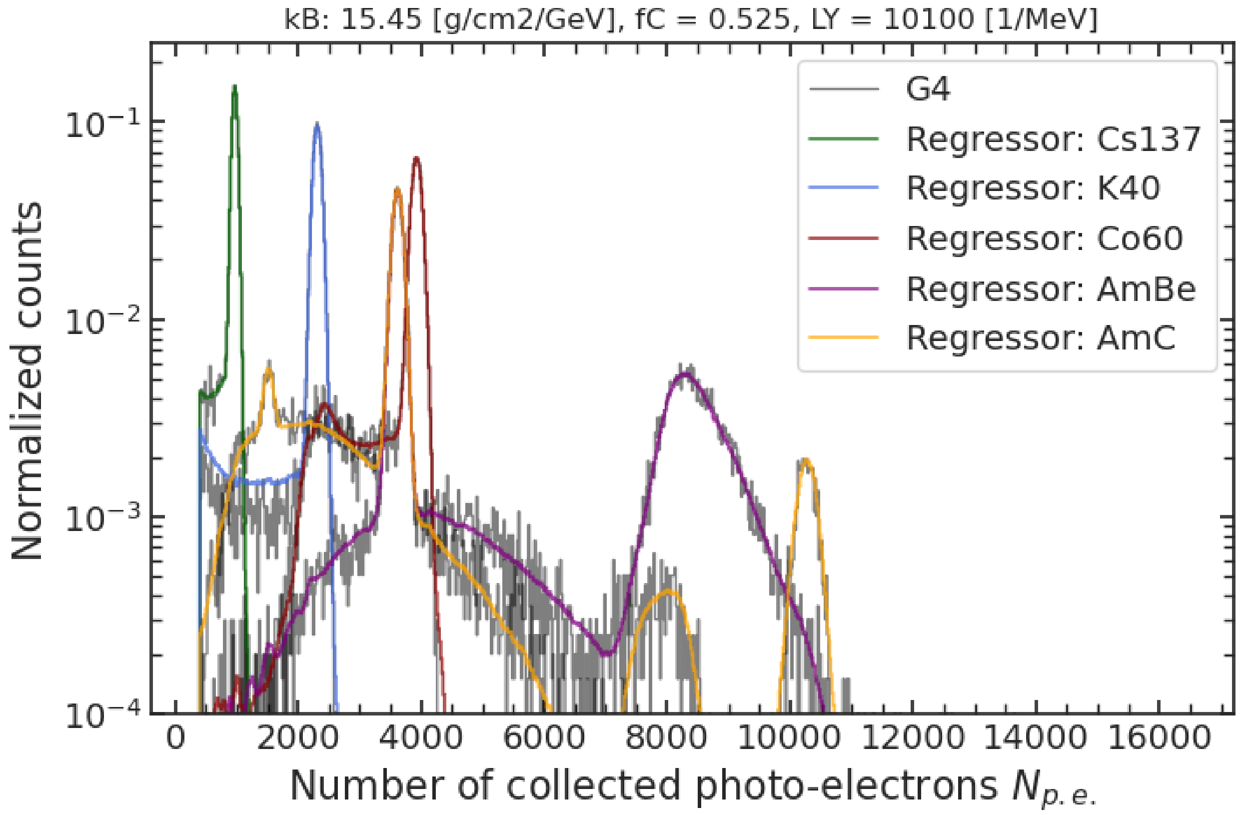
Models' performance

Regressor performance on calibration spectra



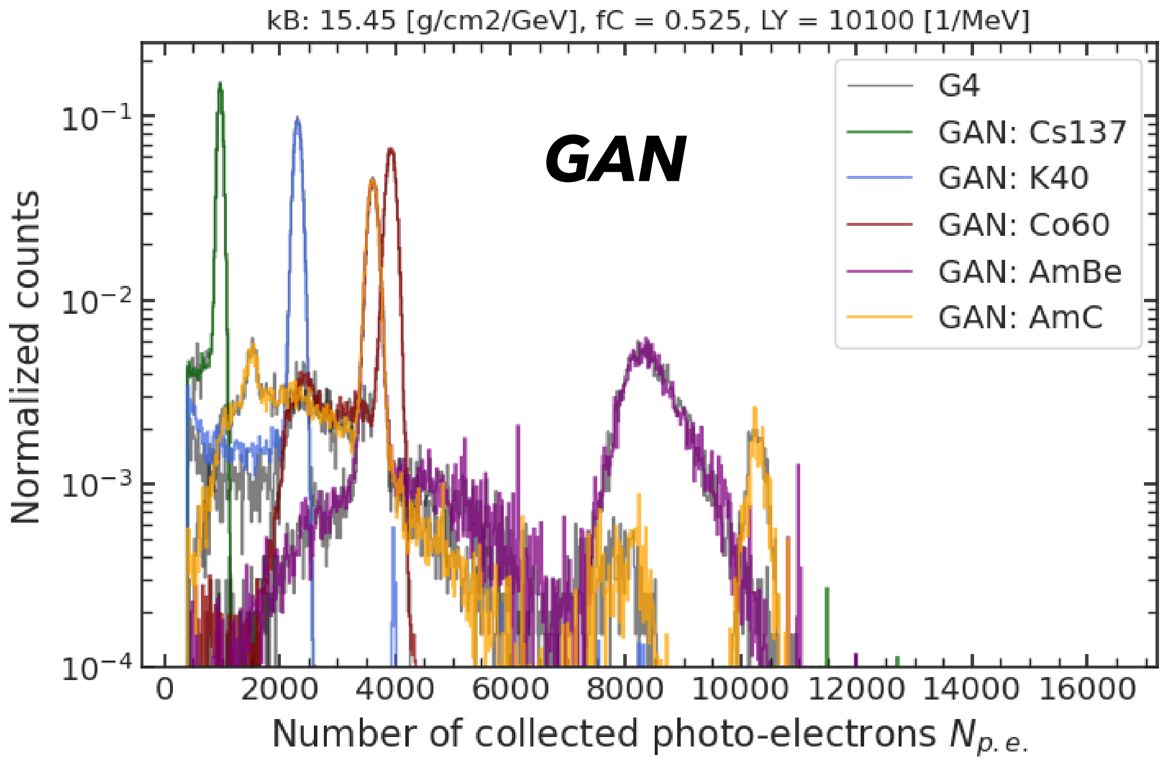
the testing dataset №2 point is between the points from the training dataset: ***interpolation***

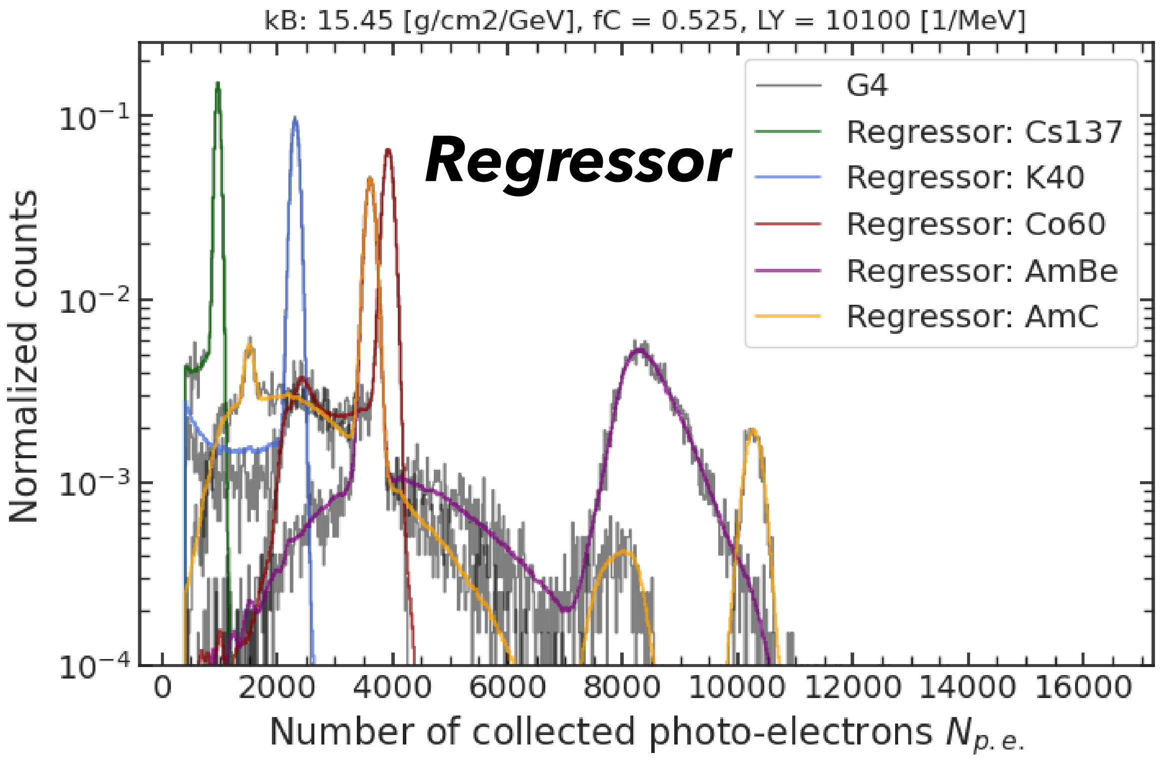
Model provides rates for ***any* continuous** values of the parameters



Interpolation with the **Regressor** model:
Smooth and denoised

GAN and Regressor performance on calibration spectra





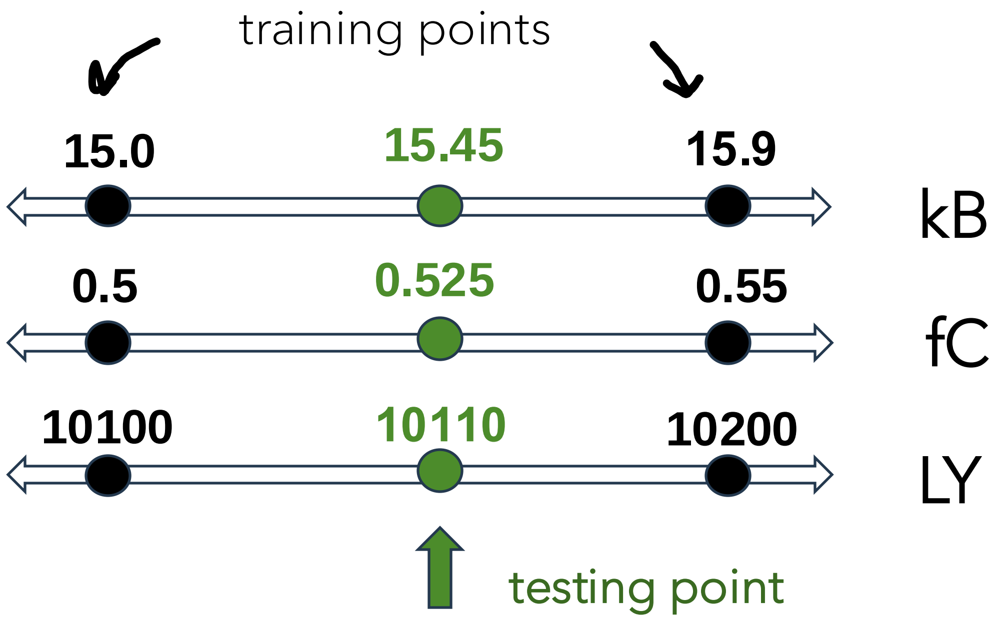
Interpolation with the GAN model: smooth in the peaks, struggles in the very low statistics regions

Interpolation with the Regressor model: Smooth and denoised

Parameter estimation

Precision and accuracy of parameter estimation

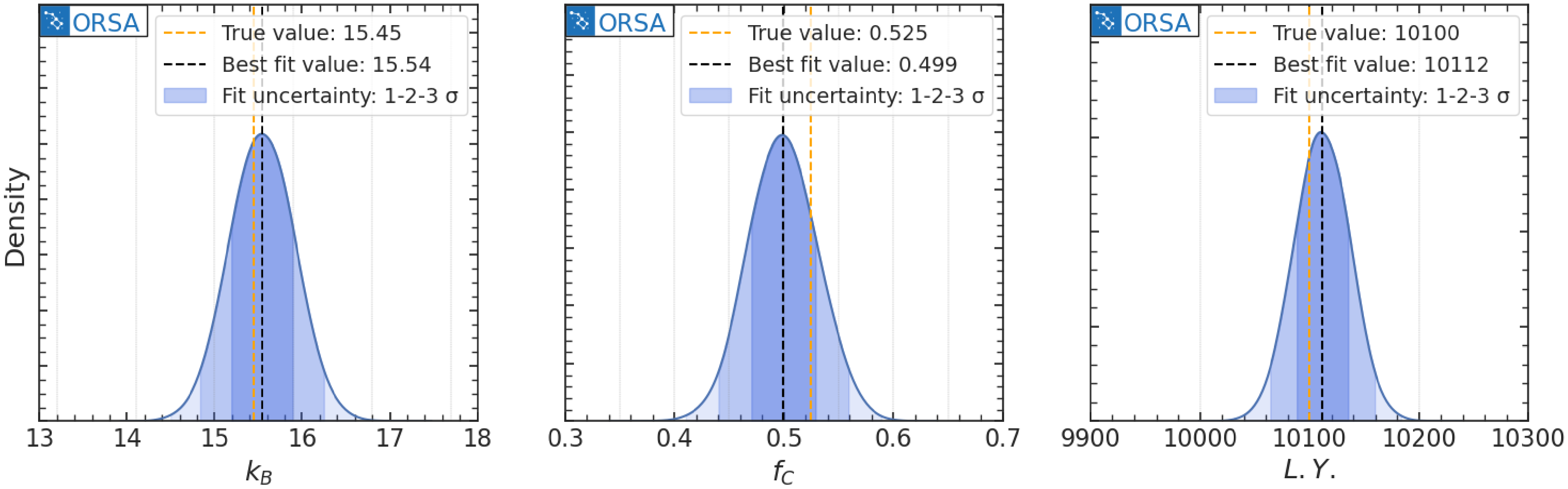
- Bin-to-bin LogPoisson as **the cost function**
- Markov-chain Monte Carlo (**MCMC**) method
- Estimate the **kB, fC, LY parameters** (using ORSA [1])
- **Explores full phase space, provides full posterior**
- Parameters estimation for the all sources: combined fit
- Shows correlation between the parameters



[1] A. Serafini, Accelerating Unbinned Likelihood Computations in JUNO with GPU Parallelization (2024)

Precision and accuracy of parameter estimation

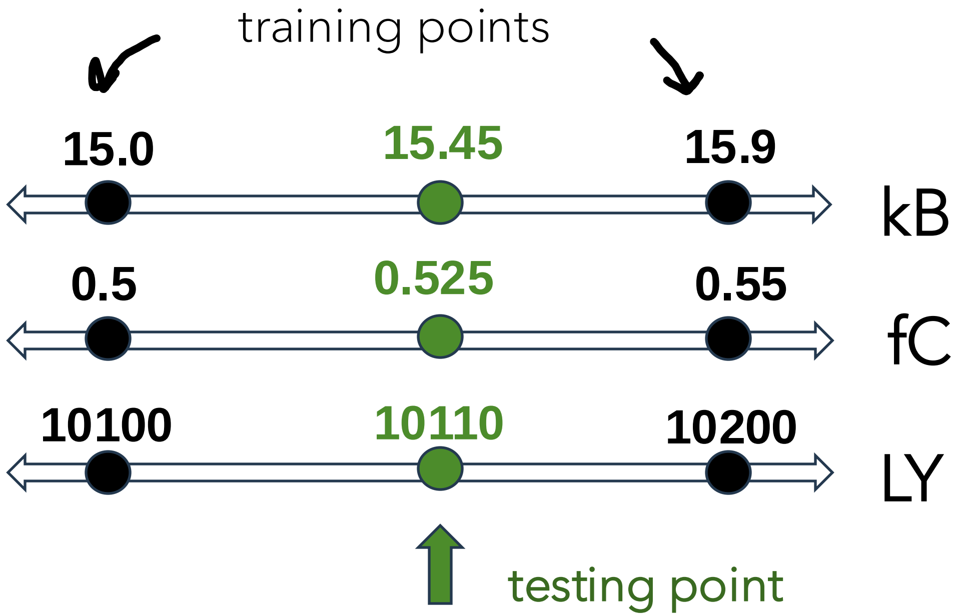
- Bin-to-bin LogPoisson as **the cost function**
- Markov-chain Monte Carlo (**MCMC**) method
- Estimate the **kB, fC, LY parameters** (using ORSA [1])
- **Explores full phase space, provides full posterior**
- Parameters estimation for the all sources: combined fit
- Shows correlation between the parameters



[1] A. Serafini, Accelerating Unbinned Likelihood Computations in JUNO with GPU Parallelization (2024)

Precision and accuracy of parameter estimation

- Bin-to-bin LogPoisson as **the cost function**
- Markov-chain Monte Carlo (**MCMC**) method
- Estimate the **kB, fC, LY parameters** (using ORSA [1])
- **Explores full phase space, provides full posterior**
- Parameters estimation for the all sources: combined fit
- Shows correlation between the parameters

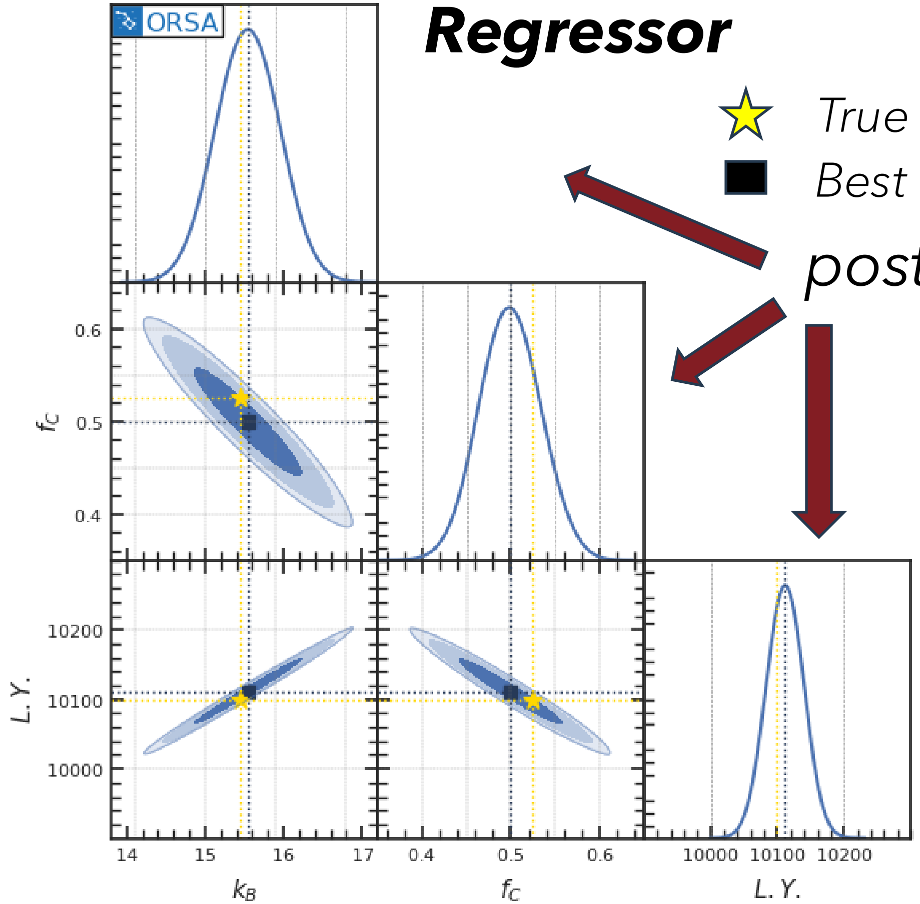


Regressor

☆ *True values*

■ *Best fit value*

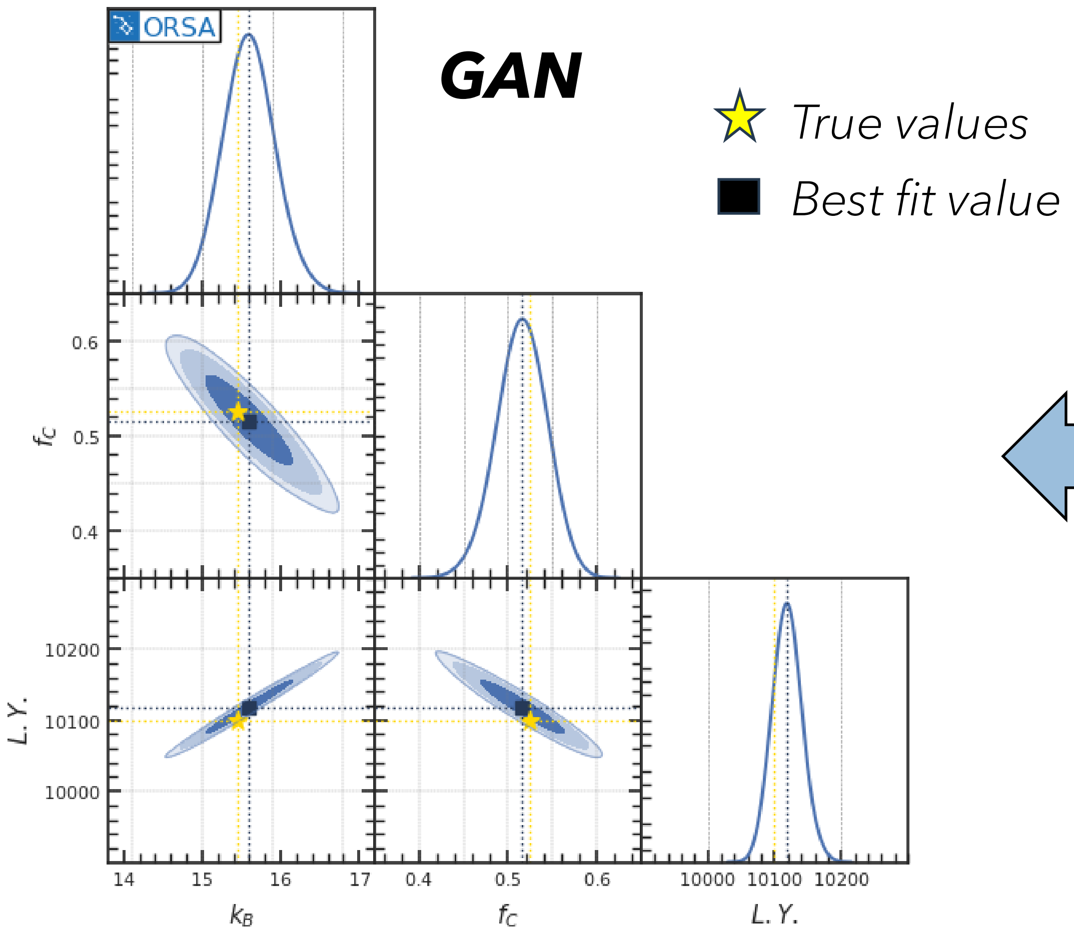
posteriors

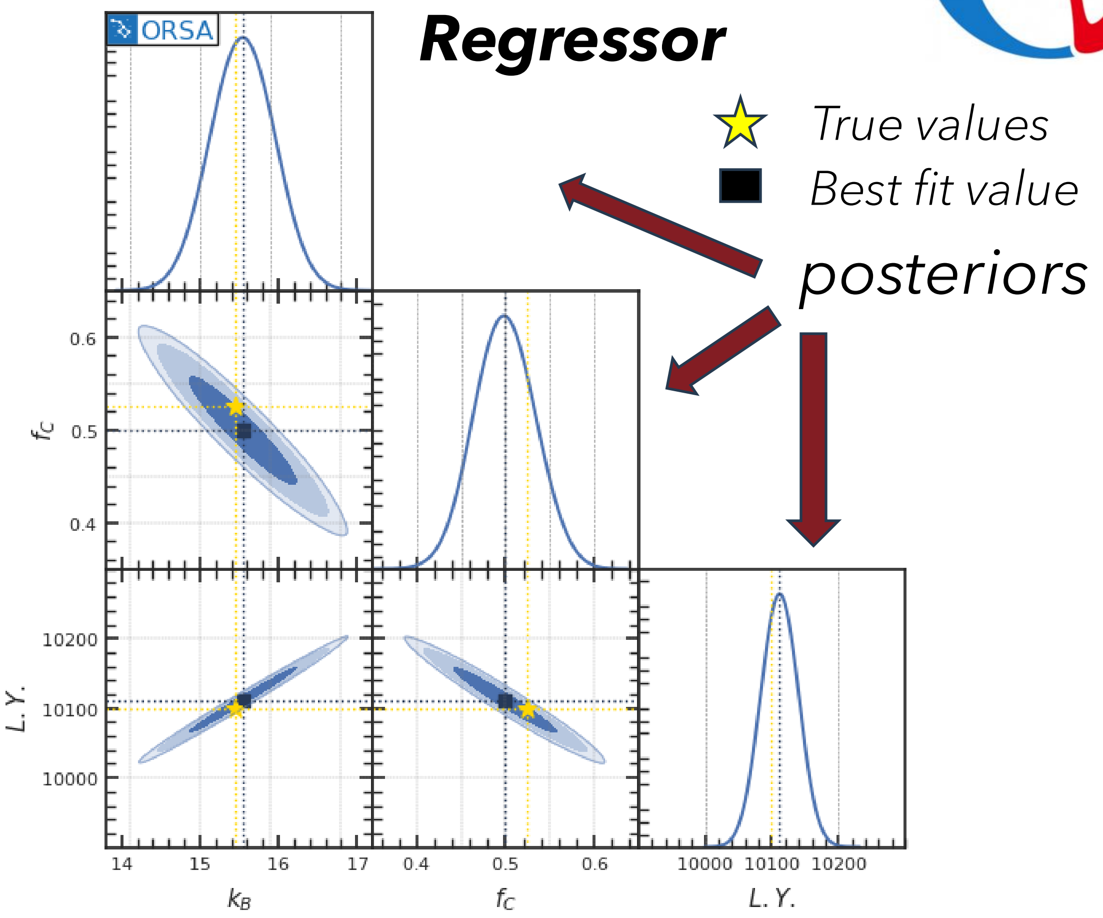


- Parameters estimation combined:
  - kB: **15.54 +- 0.35** | 15.45 [g/cm^2/GeV]
  - LY: **10112 +- 24** | 10100 [1/MeV]
  - fC: **0.499 +- 0.030** | 0.525

[1] A. Serafini, Accelerating Unbinned Likelihood Computations in JUNO with GPU Parallelization (2024)

Parameter estimation: GAN vs Regressor





- Parameters estimation combined:
  - kB: **15.58 +- 0.28** | 15.45 [g/cm^2/GeV]
  - LY: **10118 +- 20** | 10100 [1/MeV]
  - fC: **0.516 +- 0.025** | 0.525

- Parameters estimation combined:
  - kB: **15.54 +- 0.35** | 15.45 [g/cm^2/GeV]
  - LY: **10112 +- 24** | 10100 [1/MeV]
  - fC: **0.499 +- 0.030** | 0.525

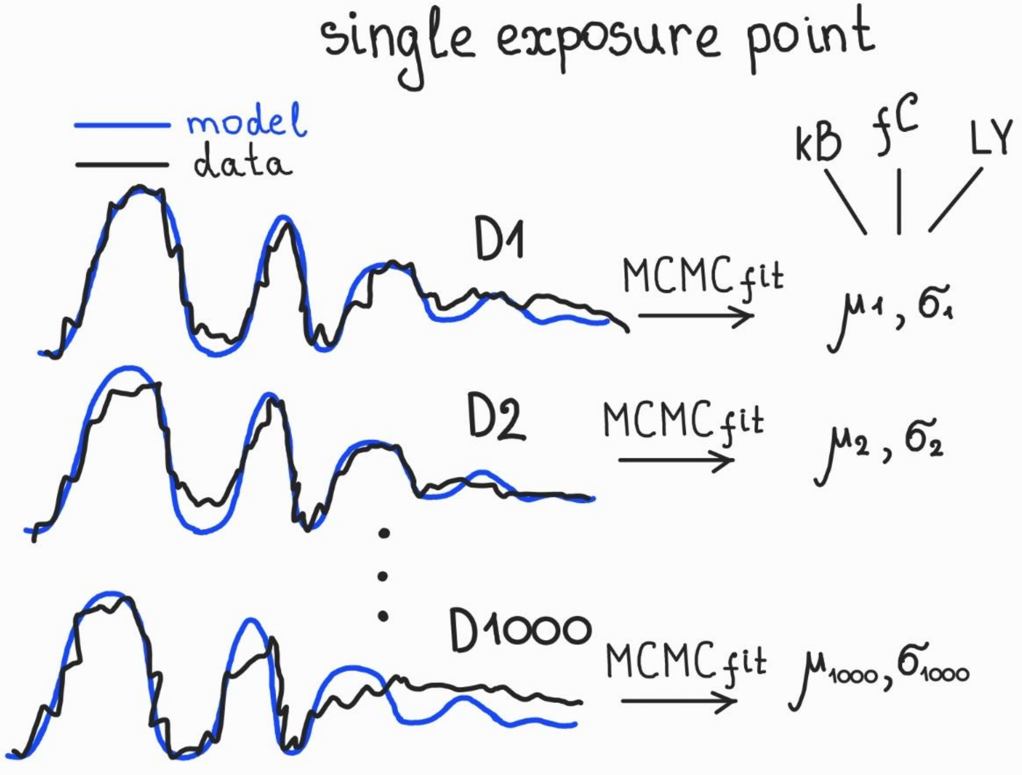
How to evaluate ML-driven systematic uncertainty?

How to evaluate ML-driven systematic uncertainty?

- To perform systematic uncertainty estimation analysis, we use the testing dataset 2:
  - Unseen during training point in the parameter space: kB, fC, LY = (15.45, 0.525, 10100)
  - 5 different exposures: 1k, 2k, 5k, 10k, 25k events
  - 1000 datasets with different JUNOSW generator seed per each exposure

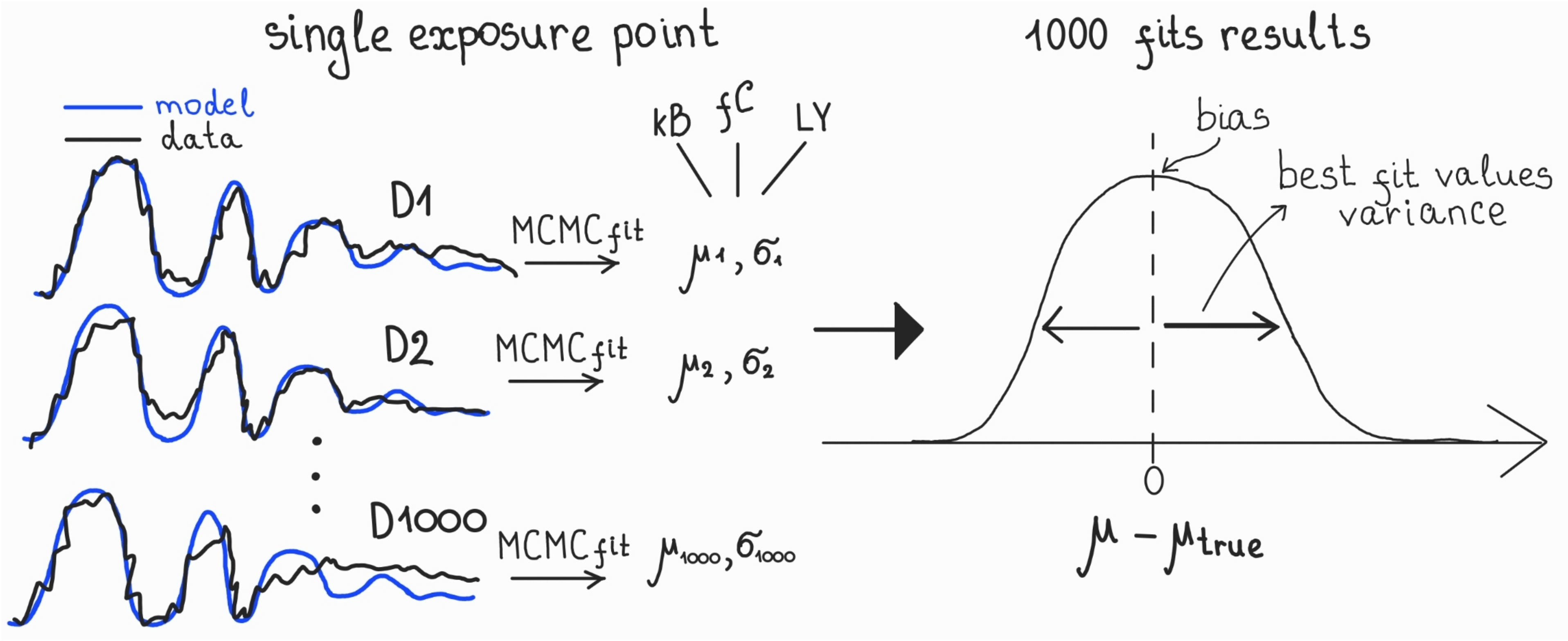
How to evaluate ML-driven systematic uncertainty?

- To perform systematic uncertainty estimation analysis, we use the testing dataset 2:
  - **Unseen during training point** in the parameter space: kB, fC, LY = (15.45, 0.525, 10100)
  - **5 different exposures**: 1k, 2k, 5k, 10k, 25k events
  - **1000 datasets with different JUNOSW generator seed** per each exposure



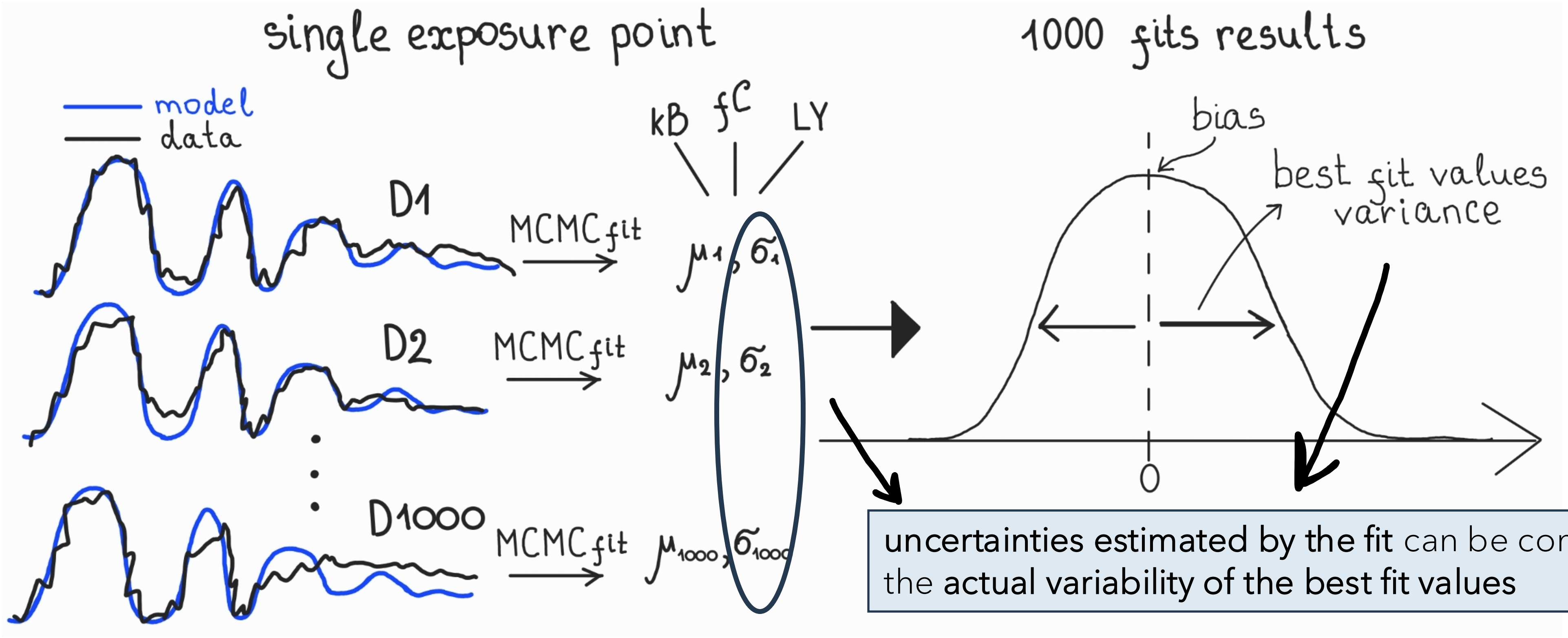
How to evaluate ML-driven systematic uncertainty?

- To perform systematic uncertainty estimation analysis, we use the testing dataset 2:
  - **Unseen during training point** in the parameter space: kB, fC, LY = (15.45, 0.525, 10100)
  - **5 different exposures**: 1k, 2k, 5k, 10k, 25k events
  - **1000 datasets with different JUNOSW generator seed** per each exposure



How to evaluate ML-driven systematic uncertainty?

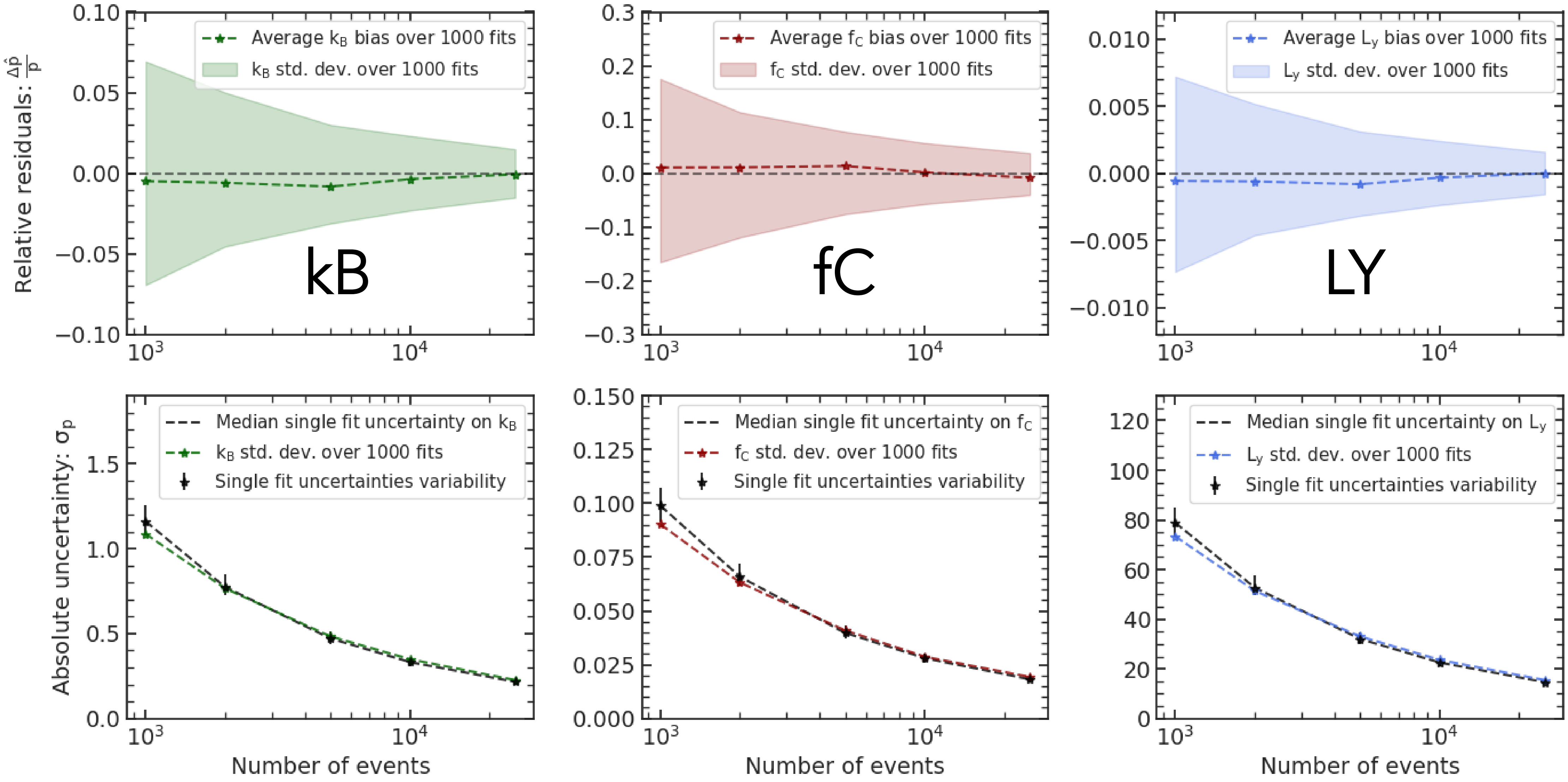
- To perform systematic uncertainty estimation analysis, we use the testing dataset 2:
  - **Unseen during training point** in the parameter space: kB, fC, LY = (15.45, 0.525, 10100)
  - **5 different exposures**: 1k, 2k, 5k, 10k, 25k events
  - **1000 datasets with different JUNOSW generator seed** per each exposure



uncertainties estimated by the fit can be compared with the **actual variability of the best fit values**

Uncertainty on the best fit parameters

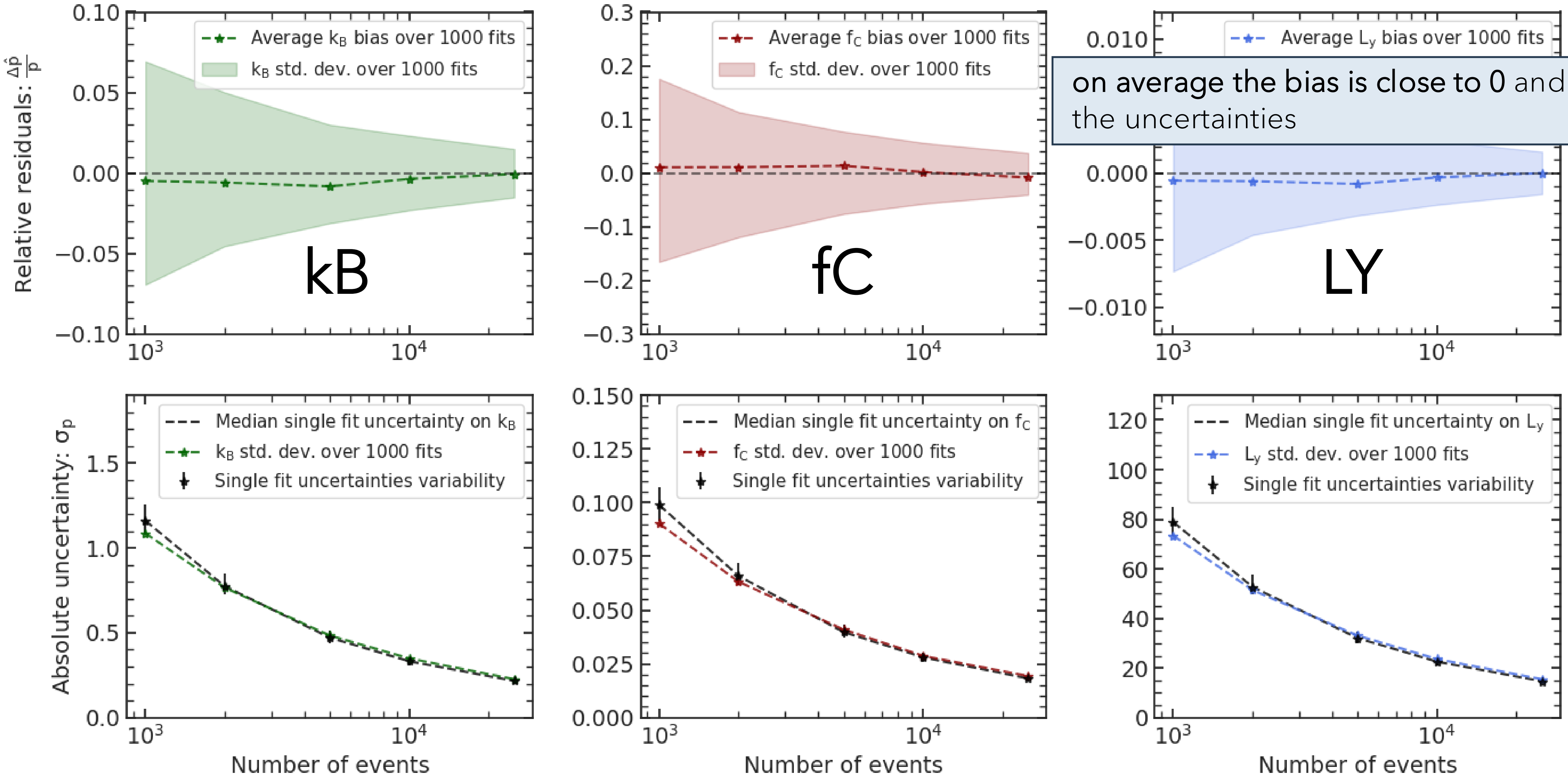
Regressor



GAN in the backup

Uncertainty on the best fit parameters

Regressor

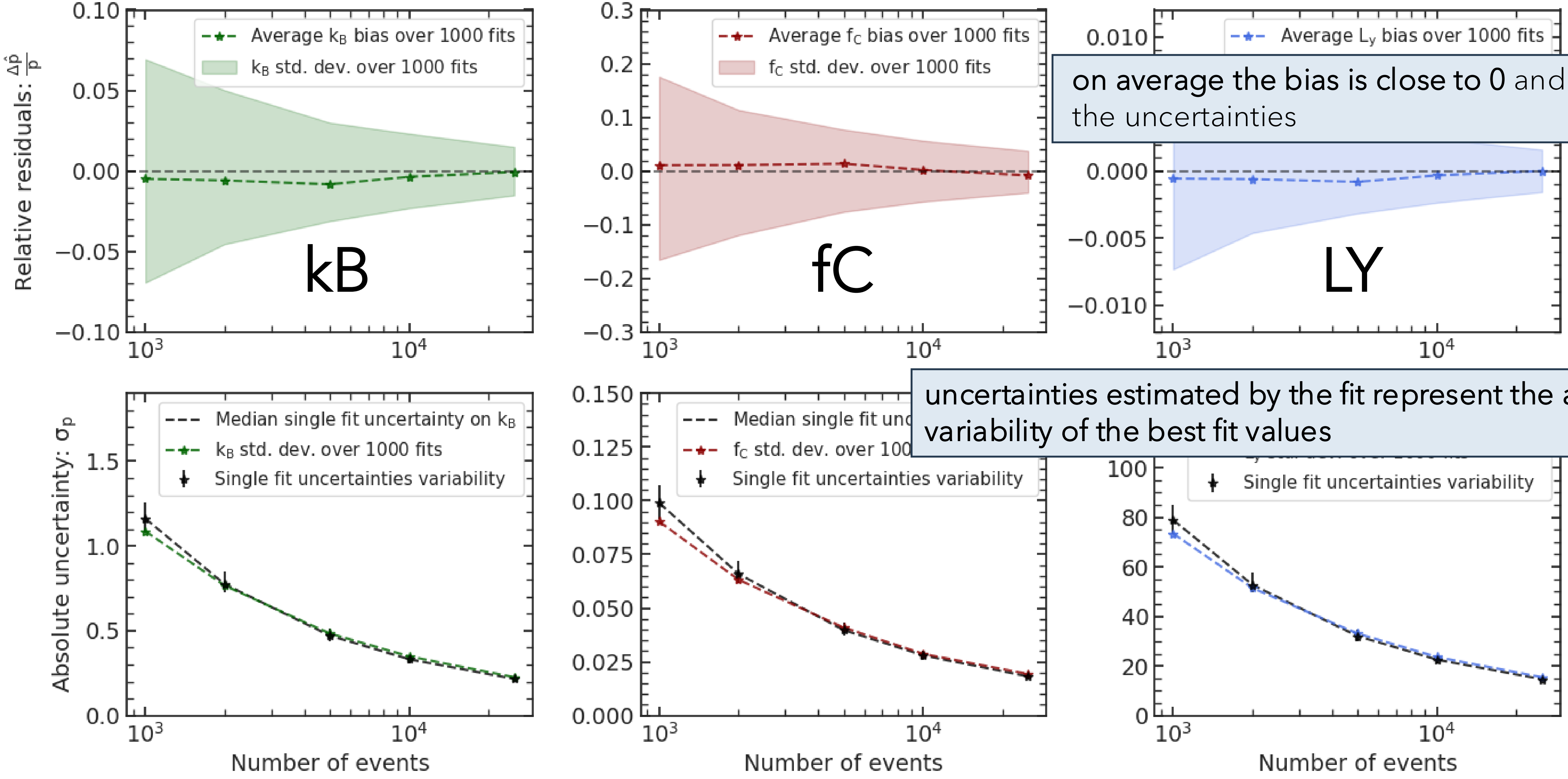


on average the bias is close to 0 and within the uncertainties

GAN in the backup

Uncertainty on the best fit parameters

Regressor



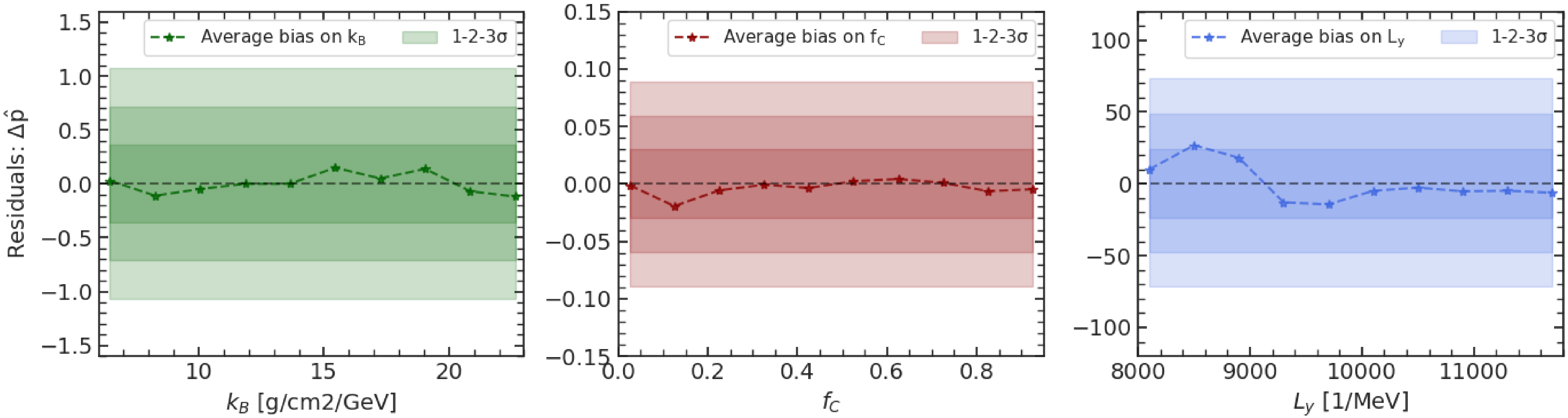
on average the bias is close to 0 and within the uncertainties

uncertainties estimated by the fit represent the actual variability of the best fit values

GAN in the backup

Uncertainty on the best fit parameters

- Using **the testing dataset 1,** one can check the bias across **different points**:
  - Run MCMC fits per each testing point of the dataset
  - Compare bias with the uncertainty **obtained by the previous analysis** for the 10k exposure point
  - **Biases are within the uncertainty**



Summary

Summary

- **An ML-based method** of MC tuning for the JUNO experiment is under development
  - Multi-output **Regressor** and **GAN** are studied
  - **Realistic dataset**: full sources simulation
  - **Based on raw number of photo-electrons**: no dependence on a reconstruction algorithm

Summary

- **An ML-based method** of MC tuning for the JUNO experiment is under development:
  - Multi-output **Regressor** and **GAN** are studied
  - **Realistic dataset**: full sources simulation
  - **Based on raw number of photo-electrons**: no dependence on a reconstruction algorithm

- Models' performances quantified:
  - **uncertainties estimated by the fit represent the actual variability of the best fit values**
  - **on average the bias is close to 0** and within the uncertainties

mostly limited by data sample statistics

- Regressor:
  - can **retrieve parameters at ~% level** kB (2.3%) fC (6.0%) LY (0.20%) with 10k-events
- GAN:
  - can **retrieve parameters at ~% level** kB (1.8%) fC (4.8%) LY (0.19%) with 10k-events

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101034319 and from the European Union – NextGenerationEU.

Thank you!

Backup

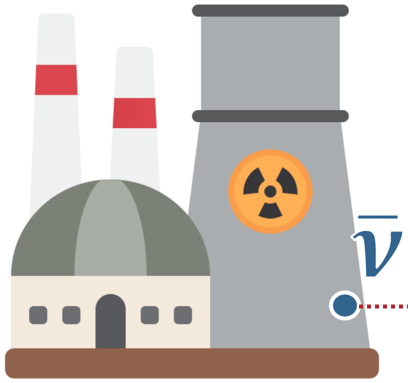
Slide credits A. Serafini

The JUNO detection process

JUNO will measure the **antineutrinos** ($\bar{\nu}_e$) generated in the fissions occurring in 8 nuclear cores at 52.5 km

The **detection** is based on a charged current interaction named Inverse Beta Decay (**IBD**) on protons (p)

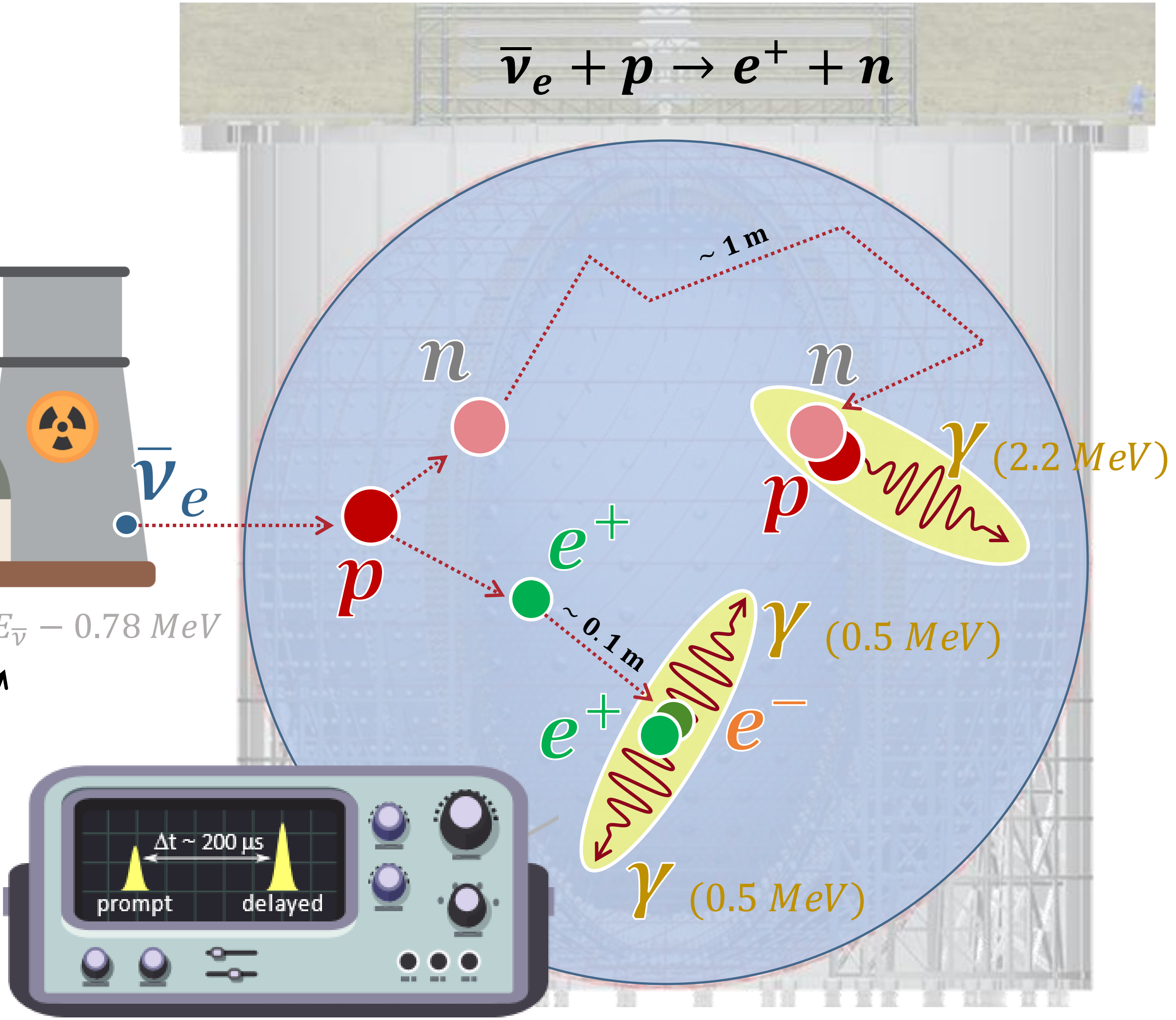
→ **sensitive only to electron $\bar{\nu}_e$**



Detection relies on a **double coincidence:**

- **prompt** signal: positron (e^+) annihilation
- **delayed** signal: neutron (n) capture

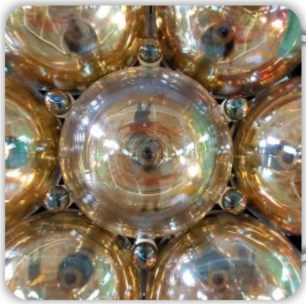
→ **strong handle against most backgrounds**

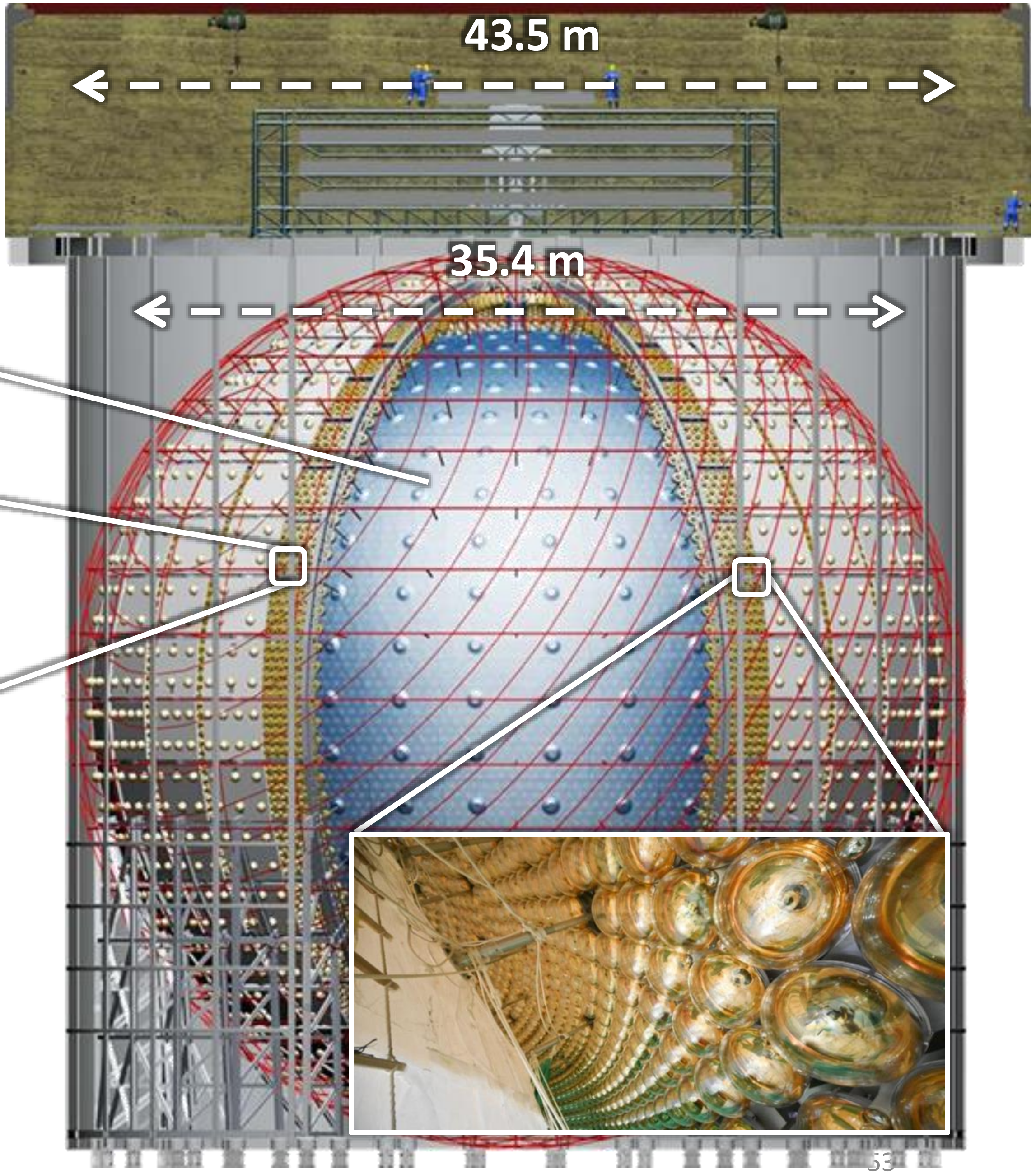


The JUNO detector

Main requirements:

- **high statistics**
  → 20 kton of liquid scintillator acrylic sphere
- **<3% energy resolution @ 1 MeV**
  → photocoverage ~78%
- **energy-scale systematics below 1%**
  → 17612 20" Large-PMT
  → 25600 3" Small-PMT





	Target mass [kton]	Energy resolution	Light yield [PE/MeV]
Daya Bay	0.02	8%/√E	160
Borexino	0.3	5%/√E	500
KamLAND	1	6%/√E	250
JUNO	**20**	**3%/√E**	**~1600**

[Prog. Part. Nucl. Phys. 2021.103927]

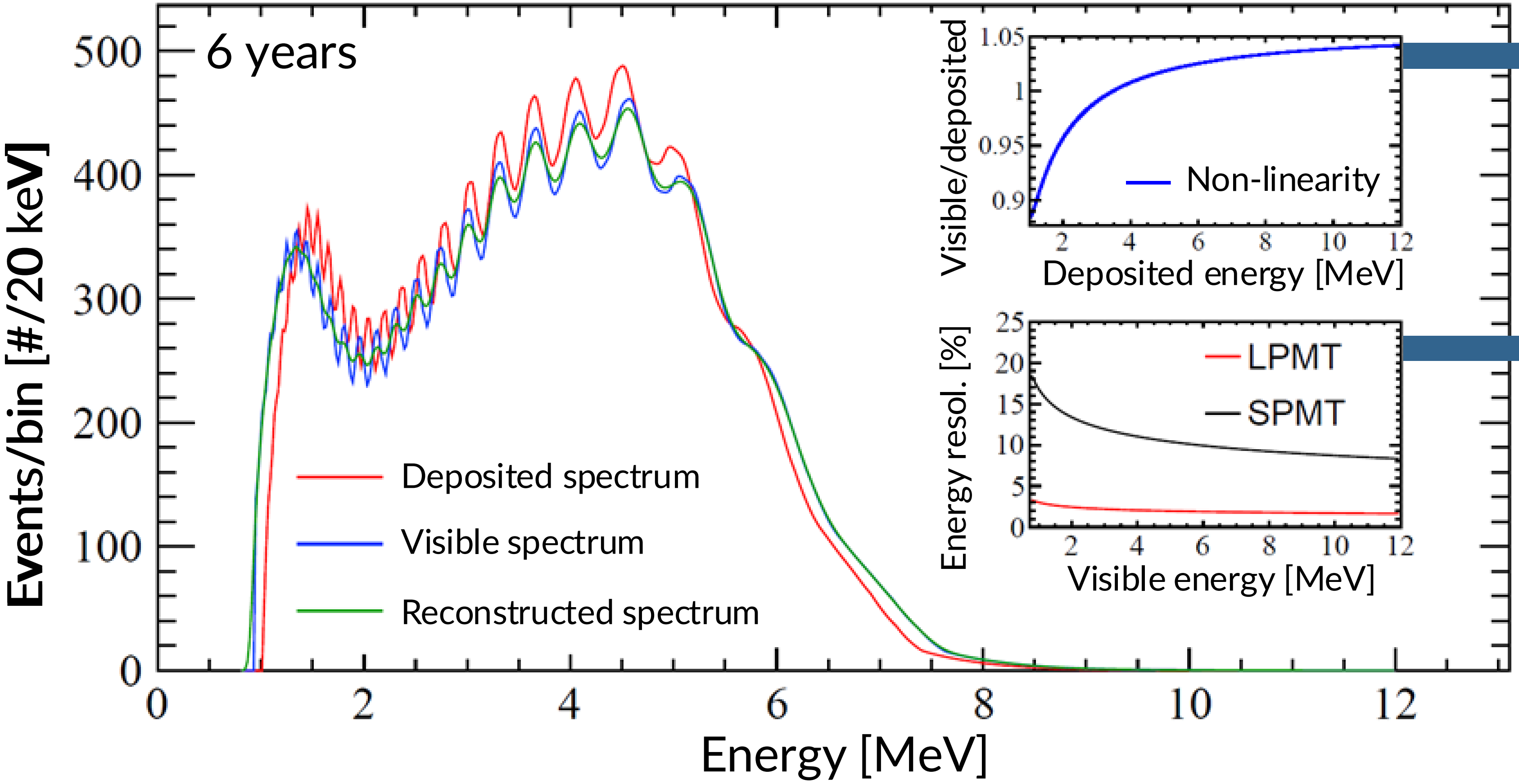
Slide credits A. Serafini

Detector response: what JUNO actually sees

interaction — light emission — light detection

E_ν → E_{dep} → E_{vis} → E_{rec}

Antineutrino energy → Deposited energy → Visible energy → Reconstructed energy



Calibration campaigns

- **automated** multiple-**position** and multi-**source calibration** (link)
- **periodic calibration** campaigns
- **dual-calorimetry** system (link)

Energy resolution

$$\frac{\sigma}{E} = \sqrt{\left(\frac{a}{\sqrt{E}}\right)^2 + b^2 + \left(\frac{c}{E}\right)^2}$$

a **Stochastic term**: light yield (from source calibration)

b Dominated by **non-uniformity** (from multi-source calibration)

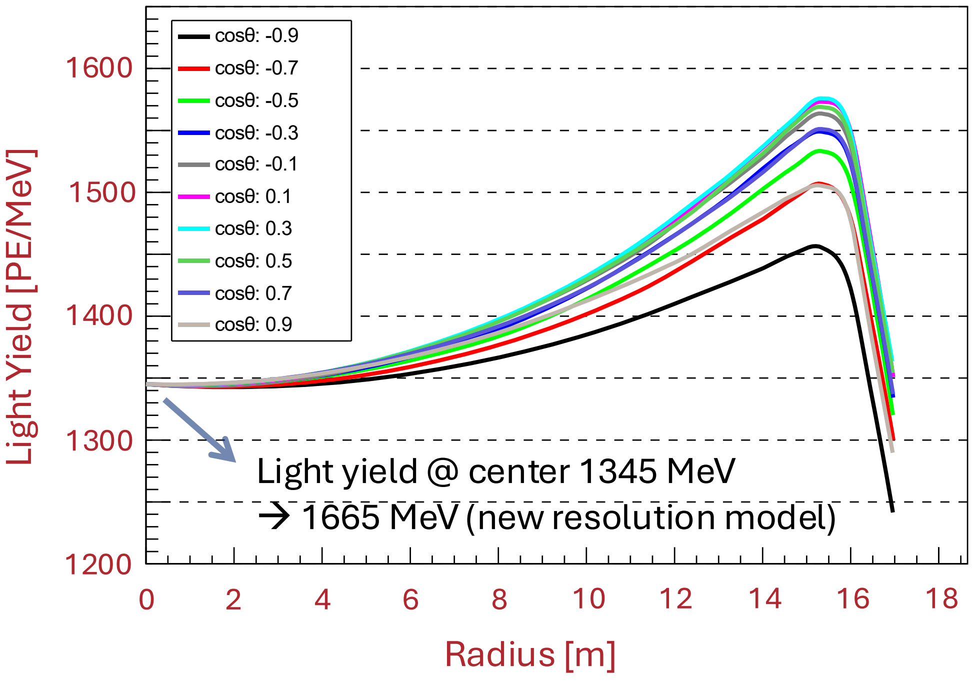
c PMT **dark noise**

Slide credits A. Serafini

Other non-linearities

Detector non-uniformity

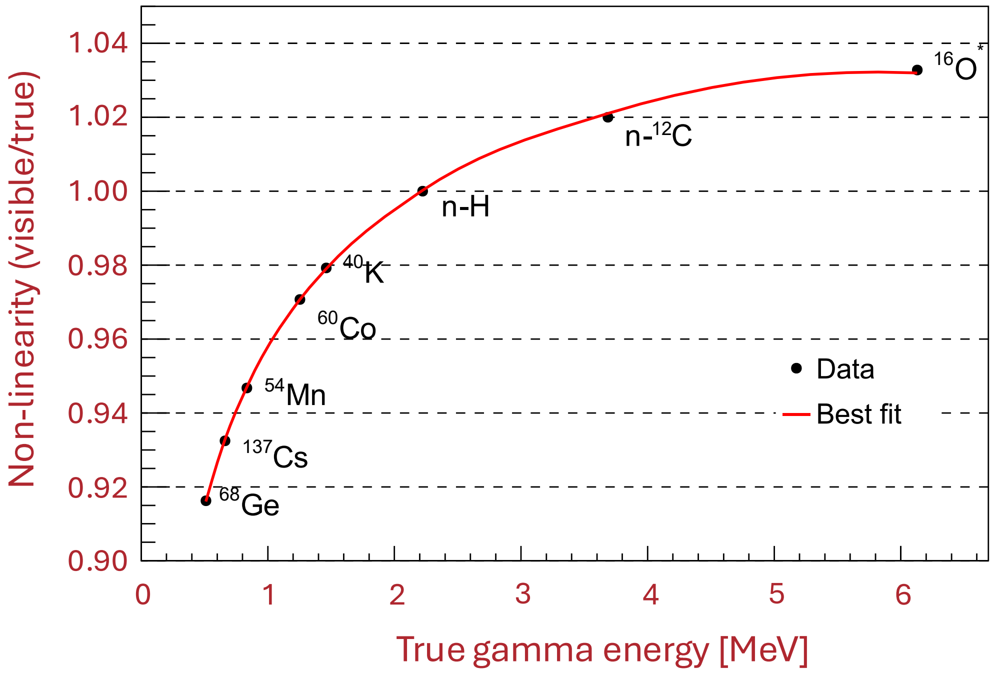
The detector response to the same charge deposition depends on the position at which the event occurs and needs to be properly characterized.



Liquid scintillator non-linearity

Light emission has an intrinsic non-linearity because of:

- Birks' quenching effect in scintillation photon yield;
- Velocity-dependent Cherenkov emission.



Slide credits D. Basilico

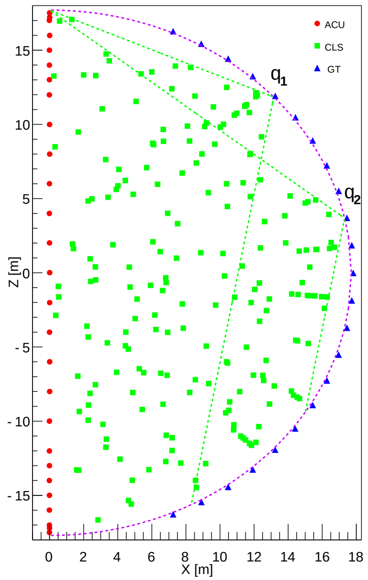
Calibration of the JUNO detector

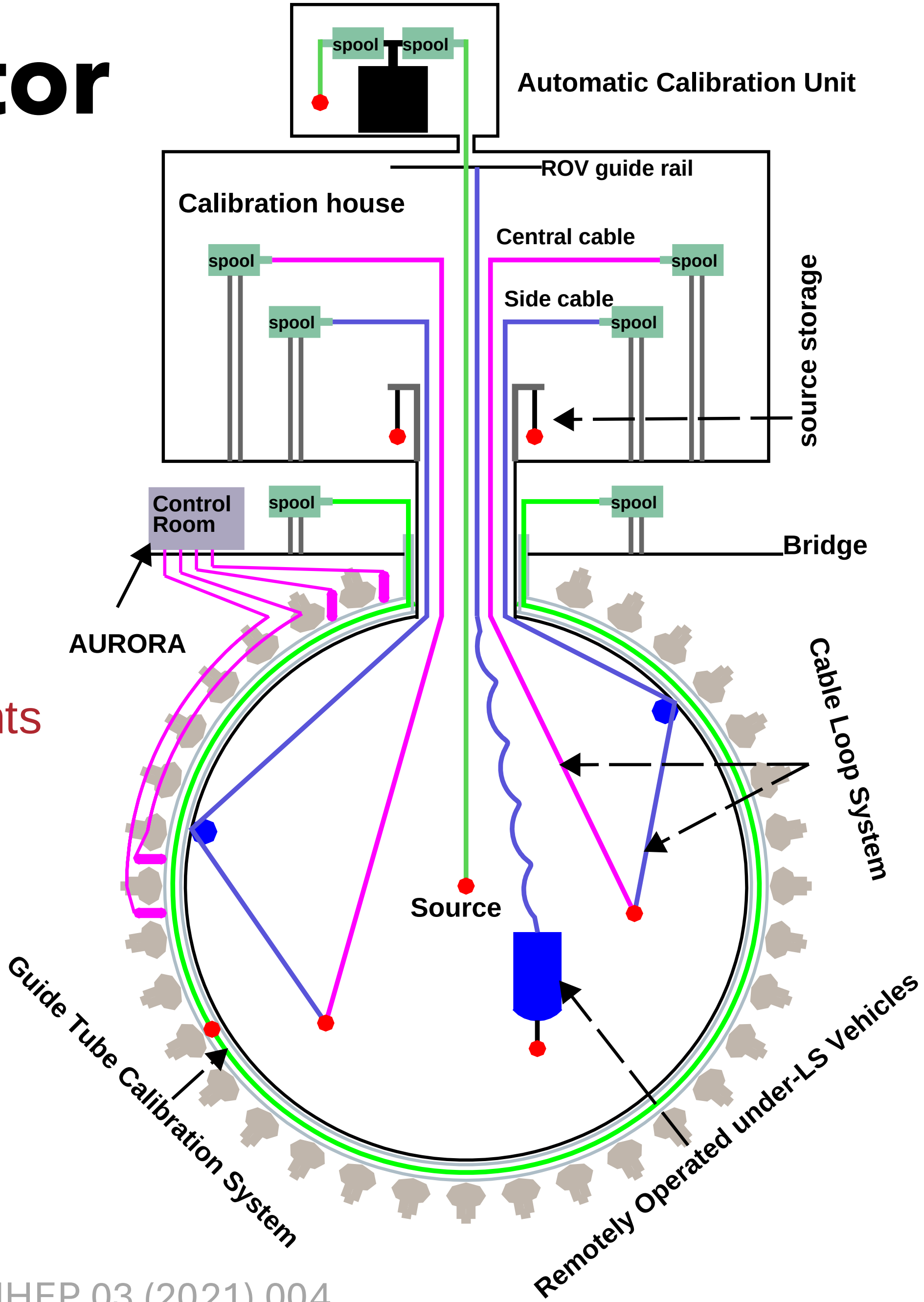
Radioactive sources (100-200 Hz) + Laser sources

- 1D: Automatic Calibration Unit (ACU)
- 2D: Cable Loop System (CLS)
- 3D: Remotely Operated under-LS Vehicles (ROV)
- Boundary: Guide Tube Calibration System (GTCS)

Sources/Processes	Type	Radiation
^{137}Cs	γ	0.662 MeV
^{54}Mn	γ	0.835 MeV
^{60}Co	γ	1.173 + 1.333 MeV
^{40}K	γ	1.461 MeV
^{68}Ge	e^+	annihilation 0.511 + 0.511 MeV
^{241}Am-Be	n, γ	neutron + 4.43 MeV (^{12}C*)
^{241}Am-^{13}C	n, γ	neutron + 6.13 MeV (^{16}O*)
(n,γ)p	γ	2.22 MeV
(n,γ)^{12}C	γ	4.94 MeV or 3.68 + 1.26 MeV

250 calibration points





JHEP 03 (2021) 004

Calibration strategy

JHEP 03 (2021) 004

Comprehensive calibration (250 points, ~48h)

→ basic understanding of the CD performance

Source	Energy [MeV]	Points
Neutron (Am-C)	2.22	250
Neutron (Am-Be)	4.4	1
Laser	/	10
^{68}Ge	0.511×2	1
^{137}Cs	0.662	1
^{54}Mn	0.835	1
^{60}Co	1.17+1.33	1
^{40}K	1.461	1
Total	/	/

Monthly calibrations (~100 points, ~11h)

→ monitor non-uniformity

System	Source	Points
ACU	Neutron (Am-C)	27
ACU	Laser	27
CLS	Neutron (Am-C)	40
GT	Neutron (Am-C)	23
Total	/	/

Weekly calibrations (~15 points, ~2.4h)

→ track variations in LY of LS, PMT gains, and electronics

Source	Energy [MeV]	Points
Neutron (Am-C)	2.22	5
Laser	/	10
Total	/	/

Uncertainty on the best fit parameters