



POLYTECHNIQUE  
MONTRÉAL

**IRSN**  
INSTITUT DE RADIOPROTECTION  
ET DE SÛRETÉ NUCLÉAIRE

  
**RÉPUBLIQUE  
FRANÇAISE**  
*Liberté  
Égalité  
Fraternité*

## PyNjoy2016: an Open Source System for Producing Cross Sections Libraries for DRAGON5 and SERPENT2

V. SALINO<sup>1,2</sup> and A. HÉBERT<sup>1</sup>

<sup>1</sup> École Polytechnique de Montréal, Institut de Génie Nucléaire, Canada

<sup>2</sup> Institut de Radioprotection et Sûreté Nucléaire (IRSN), PSN-RES/SNC/LN, France

M&C 2023 - The International Conference on Mathematics and Computational Methods Applied to  
Nuclear Science and Engineering · Niagara Falls, Ontario, Canada · August 13 – 17, 2023

# Presentation outline

(1) Introduction

Introduction

(2) Thermal scattering

Thermal scattering

(3) Impact of energy deposition model

Impact of energy  
deposition model

(4) Propagation of nuclear data uncertainties

Propagation of  
nuclear data  
uncertainties

(5) Conclusions

Conclusions

# Introduction : what is PyNjoy2016?

- Python wrapper around NJOY2016, in a modified and improved version
- Streamlines processing of complete nuclear data evaluations, producing **consistent**
  - multigroup libraries for DRAGON5, in Draglib format
  - continuous-energy libraries for SERPENT2, in mixed ACE+ENDF-6 format

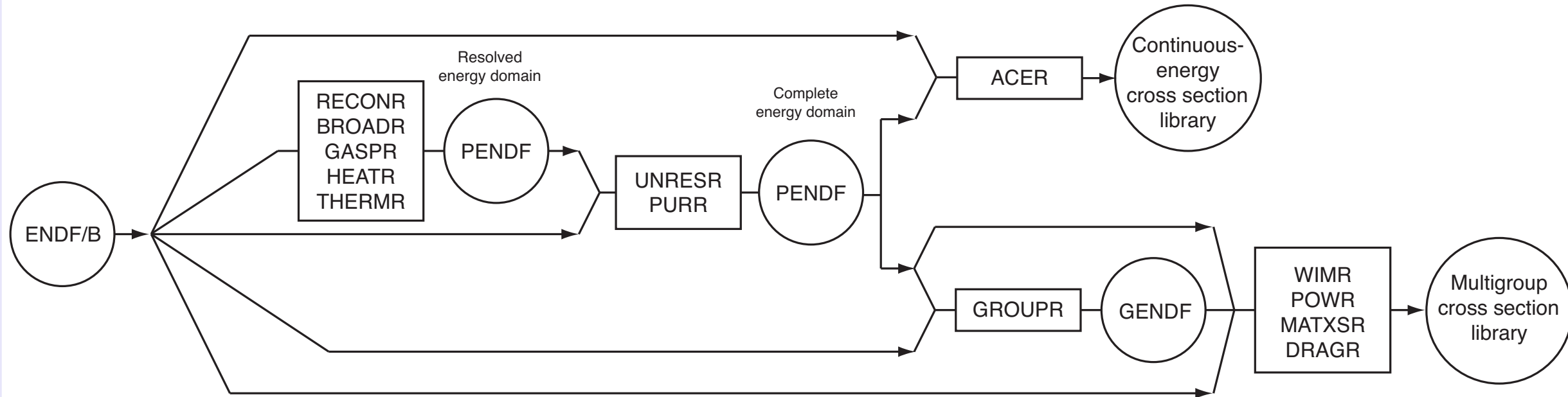


Figure : PyNjoy2016 data flow



# A consistent production of multigroup and continuous-energy libraries

## Introduction

## Thermal scattering

## Impact of energy deposition model

## Propagation of nuclear data uncertainties

## Conclusions

- Tempting to think that an evaluation can only result in a single **ACE file** (no multigroup approximation)
- But it does **depend on NJOY version** and **options**
- Almost inevitably, **different NJOY versions** and **options** used for multigroup library and for default ACE files (as delivered with Monte-Carlo codes)
- Implies **inconsistencies** when comparing deterministic and Monte-Carlo codes
- This additionnal **error** is most often incorrectly **attributed** to the deterministic code, on account of its **deterministic bias**, which is then **inaccurately evaluated**
- For example, **100 pcm difference** on  $k_{\text{eff}}$  between
  - ACE files produced with PyNjoy2016
  - ACE files delivered by default with SERPENT2based on JEFF-3.1.1 in both cases, on first start-up of Tihange-1
- Vast **majority** of comparisons between Monte-Carlo and deterministic codes in public and industrial literature are **affected** by these **inconsistencies**
- **Computing deterministic bias requires consistency in nuclear data processing**



# A lesson learned during non-regression verification : Effective multiplication factor discrepancies, versus PyNjoy2012

$\Delta \ln k_{\text{eff}}$  (pcm) on Rowland's PWR pincell, with DRAGON5 and 172 energy groups :

Burnup (MW·day per ton of initial heavy metal)	0	36,8	1000	2500	5000
(A) NJOY2016 with <b>118</b> points in THERMR (default)	15,9	15,5	4,7	8,2	10,5
(B) NJOY2016 with <b>357</b> points in THERMR	-0,4	-0,4	-0,2	-0,3	-0,3
(C) NJOY2016 with <b>357</b> points in THERMR and backporting the universal physical constants from NJOY2012	0	0	0	0	0
(D) NJOY99 with 357 points in THERMR	0	0	0	0	0

- Discrepancies mostly due to thermal scattering : NJOY uses an **insufficient discretization** of the incident energies, in **THERMR** (calcem subroutine)
- Can reach **hundreds of pcm** in more thermalized systems (Yamamoto and Sugimura, 2006)
- **Both deterministic and Monte-Carlo** codes are affected by this THERMR deficiency
- **Strongly recommend refining** this energy grid, **without waiting** for a complete overhaul of THERMR → meanwhile, NJOY users are getting a **biased answer**

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions

# First start-up of Tihange-1, a 900 MWe PWR

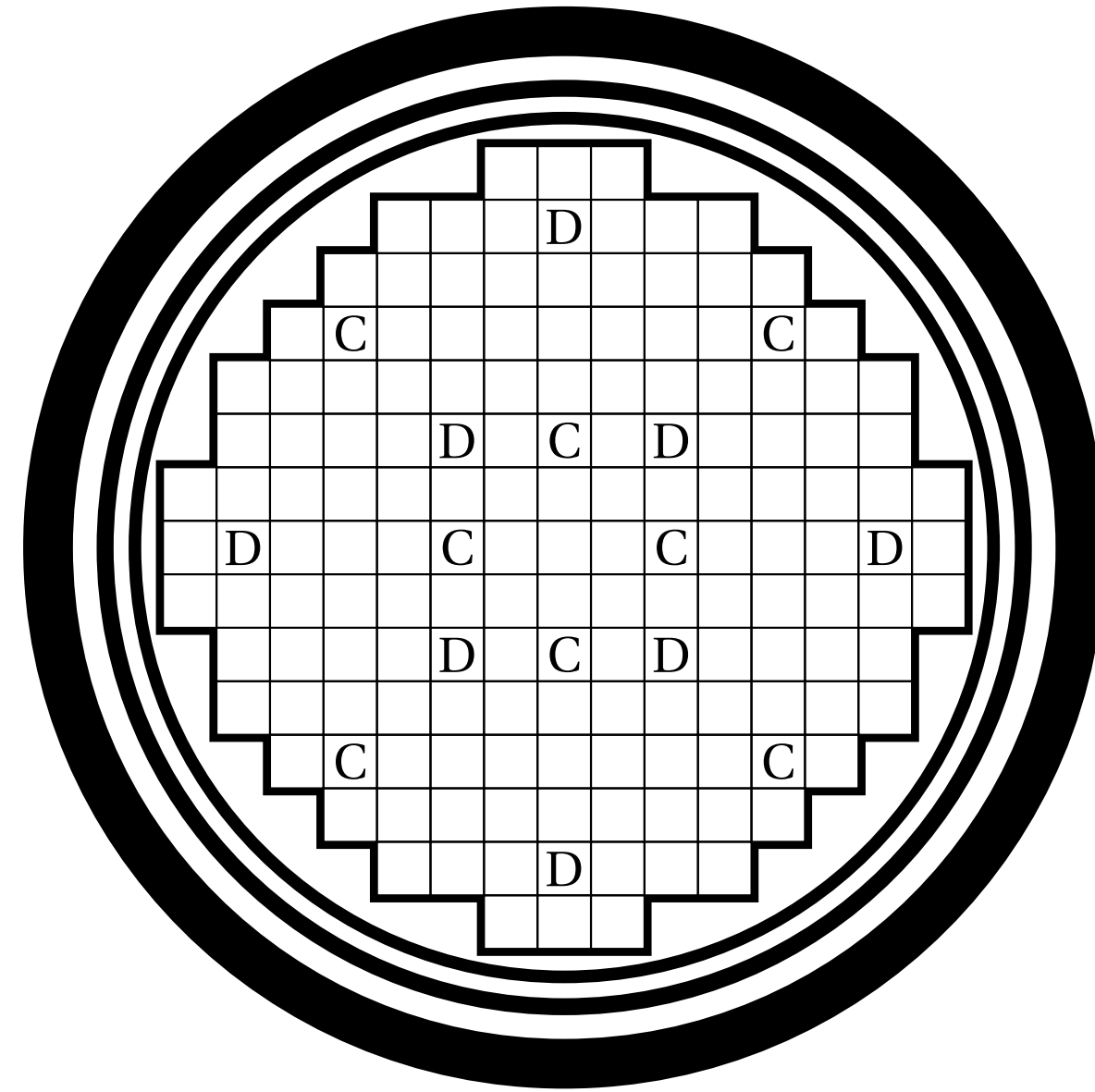
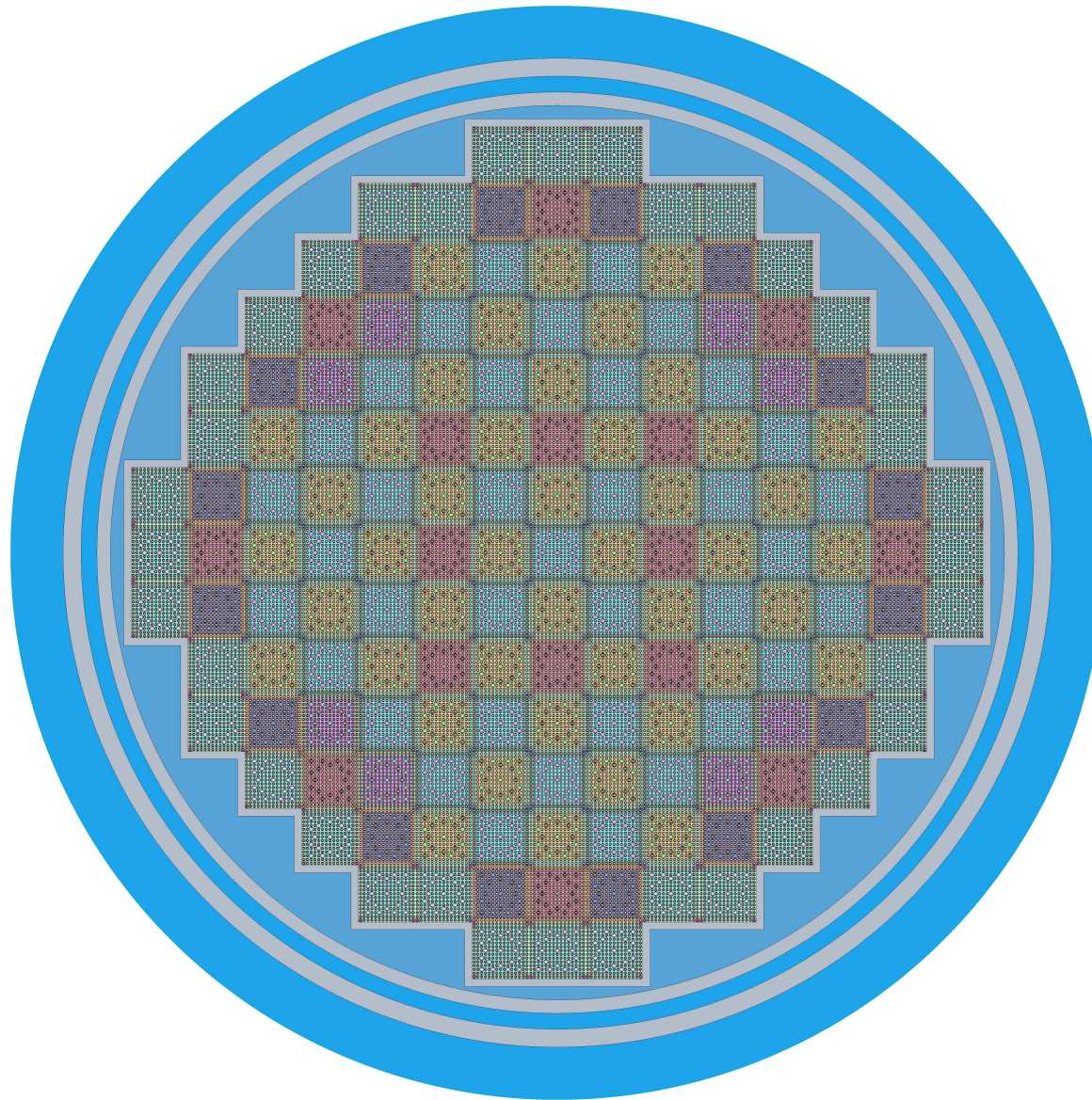
Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions



# Impact of energy deposition model (1/2)

## Default in SERPENT2 : an approximate energy is deposited only at fission sites

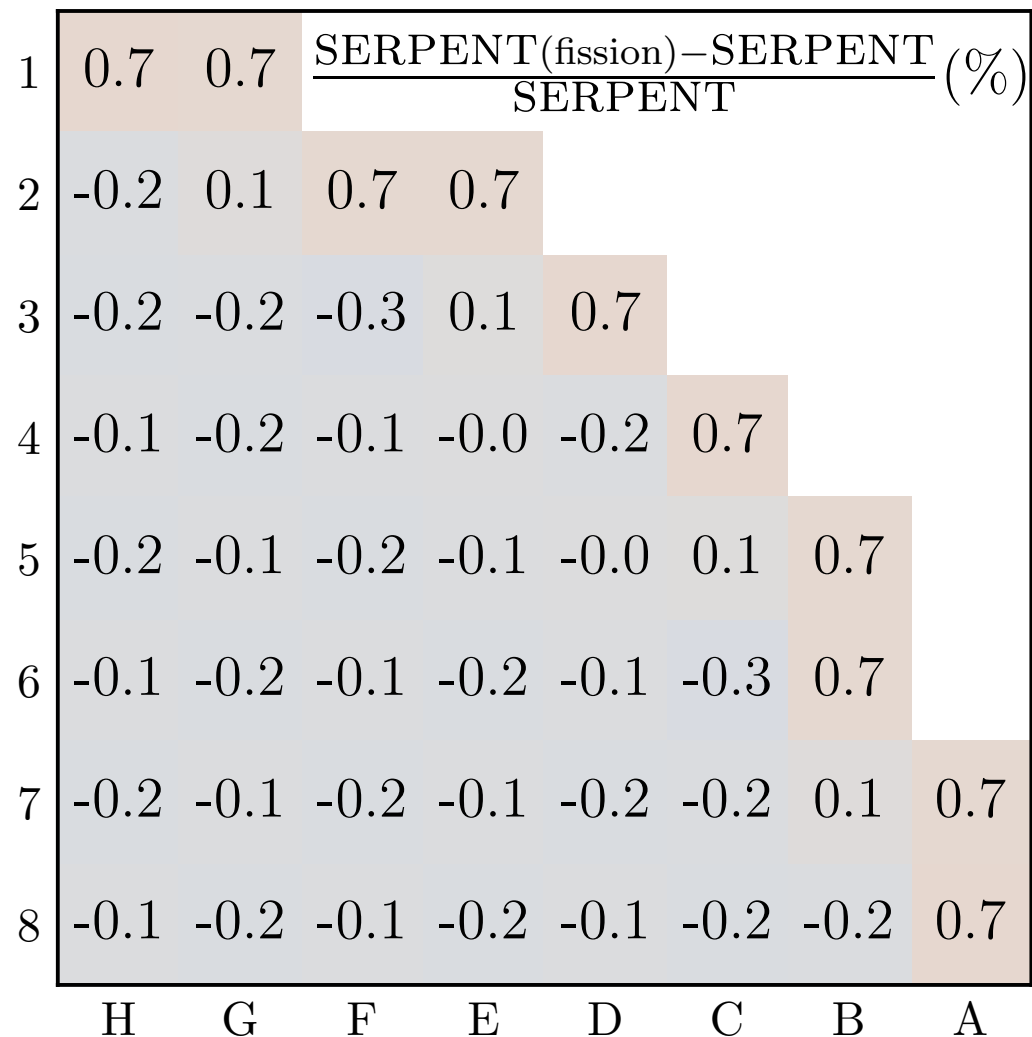
Introduction

Thermal scattering

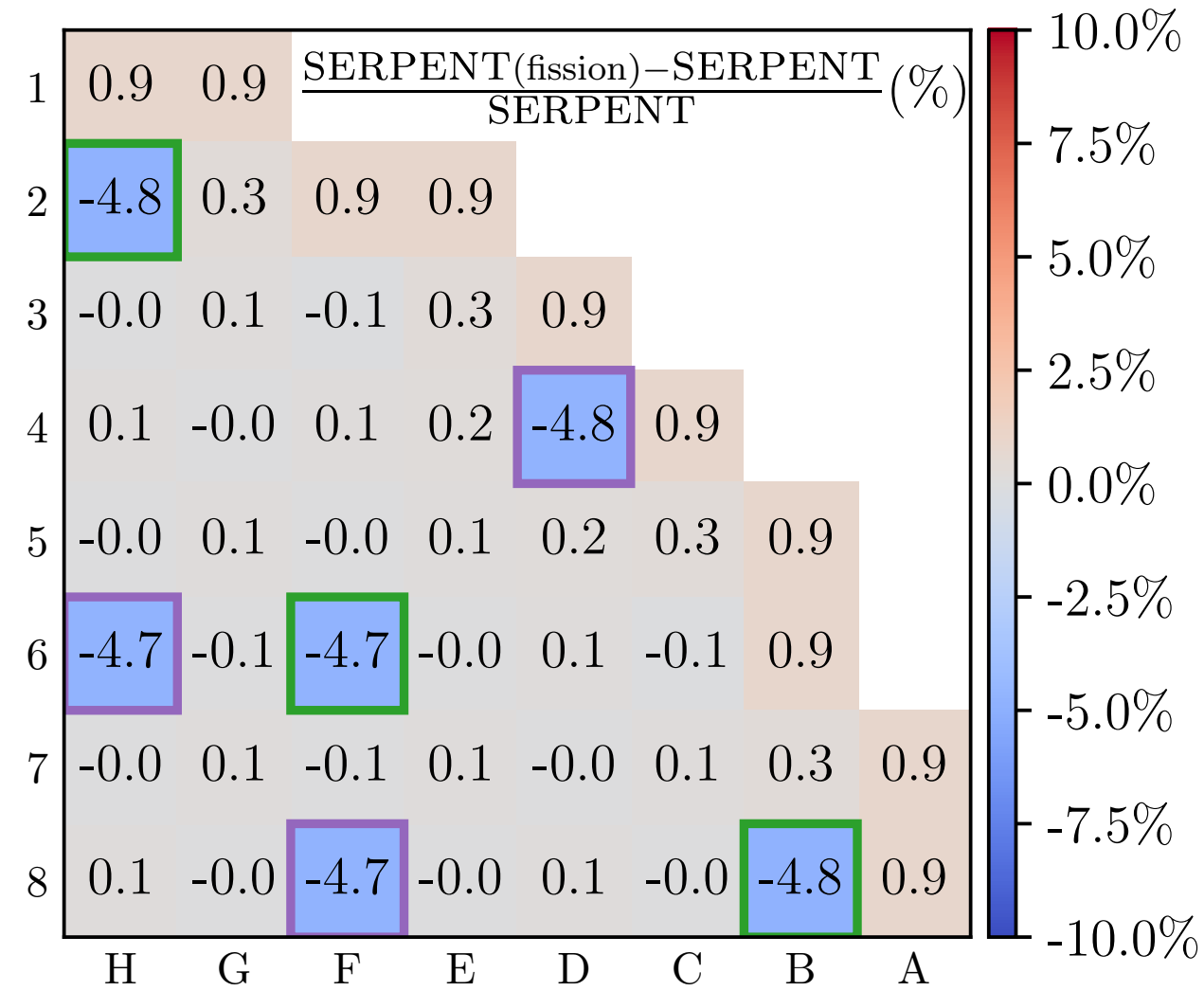
Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions



All rods out



Inserted control rods (purple and green boxes)

# Impact of energy deposition model (2/2)

## SERPENT2 advanced energy deposition model

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions

- When comparing codes, the most common attitude regarding the previous **approximate** model is to **ignore** its consequences
- Here again, a deterministic code is often blamed for this **error**, erroneously labeled as **deterministic bias**
- SERPENT2 also features an **advanced energy deposition model**
- For that, relies on files mixing ACE and ENDF-6 formats, but such **files** are **not widely available**
- PyNjoy2016 has the ability to produce such ACE+ENDF-6 files, to compute properly
  - **Power distribution**, even without depletion
  - **Depletion**, because burnup is an amount of energy deposited (MW·day) per unit mass (ton) → if incorrectly estimated, a drift will appear during depletion, even on  $k_{\text{eff}}$
- **Unresolved resonance range** has no tangible impact on our use case, but is nevertheless properly taken into account in PyNjoy2016 for potential future uses on systems with larger proportions of fast or epithermal neutrons
- PyNjoy2016 also supports energy deposition following **photon transport**



# Propagation of uncertainties related to nuclear data (1/2)

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions

1) **Perturbation theory** → some drawbacks

→ Difficult to propagate through **self-shielding** or a **multiphysics** coupling (transients, etc)

→ Cannot access the complete distribution, but **only the mean** and **standard deviation**

Advantage → very **fast** and **powerful**, as long as its drawbacks are not blocking issues for the desired use

2) **Total Monte-Carlo** (sampling)

→ Propagation through subsequent codes without modification or simplification of any kind (**self-shielding, multiphysics...**)

→ Access the **complete distribution** on a quantity of interest, without approximation

→ Reference solution, comes with higher **computational cost**

→ **Convergence** must be verified

## Propagation of uncertainties related to nuclear data (2/2)

PyNjoy2016 has capability to produce randomly **sampled ACE** and **DRAGLIB** files on multiple CPUs (in parallel), taking as a basis either :

- 1) **presampled TENDL** files in ENDF-6 format, sampled upstream of its nuclear physics models
    - **Continuously** varying uncertainties
    - **Not necessarily Gaussian**
    - But no TENDL files for most important **heavy** isotopes ( $^{233,235,238}\text{U}$ ,  $^{239}\text{Pu}$ ,  $^{232}\text{Th}$ ) and nor for lightest nuclei, consisting of **less than 20 nucleons**, due to optical model limitations
  - 2) ENDF-6 files **sampled** Gaussianly within multigroup **covariance** matrices, with on-the-fly calls to **SANDY** (Fiorito, 2017)
    - More **adventurous** path, since **covariance** formats are more **heterogeneous** than the homogeneous TENDL formats (derived from a single code chain)
- See companion presentation on Wednesday evening for such an uncertainty propagation

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions

## Shall we merge DRAGR into NJOY2016 main branch?

- NJOY2016 documentation says that "the **mission** of NJOY is to take basic data from the nuclear data library and **convert it** into the **forms needed** for applications"
- Therefore, we believe **DRAGR should be integrated** into NJOY2016 main branch
- Without such an inclusion, the NJOY community may become increasingly divided, having more different versions being used, instead of **benefiting from a community effect**
- Added a **non-regression test** similar to those of NJOY2016, but specific to DRAGR
  - Using the **same ENDF-6** files already used in the non-regression testing of NJOY2016 (does not burden its repository)
  - As **short** as possible : does not include all the isotopes of an evaluation (typically between 300 and 800 isotopes), however all possible use cases are covered
- **DRAGR** is thoroughly **documented**, in a  $\LaTeX$  section following NJOY documentation format
- **Same** path should apply to **resonant upscattering** RESKR module, originally from NECP-Atlas (J. Xu, T. Zu and L. Cao, Xi'an Jiaotong University, China, 2019)

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions

# Conclusions

- **PyNjoy2016** is dedicated to the production of :
  - **multigroup libraries** in DRAGLIB format, for the **DRAGON5** deterministic code
  - **continuous-energy ACE** files, supporting the deposition energy model of **SERPENT2**
- **Deterministic bias** of a calculation scheme based on DRAGON5 can now be precisely **calculated** by consistently comparing its results with SERPENT2
- Such a **consistency** is a rare feature provided by the simultaneous library production : the **same NJOY2016** run, in the **same version** and with the **same options** are used to produce both libraries
- As an example of its importance, **100 pcm** reactivity difference between ACE files delivered with SERPENT2 and those produced with PyNjoy2016 (for same nuclear data)
- Open nature of NJOY2016 implies that PyNjoy2016 can be fully **open source software** (BSD license)

[github.com/IRSN/PyNjoy2016](https://github.com/IRSN/PyNjoy2016)

- Usefully complementing the state-of-the-art **DRAGON5** deterministic lattice code, also open source and, in that sense, **unique**

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions





# Acknowledgments

- [Dimitri Rochman](#) (PSI) for his advices on TENDL, SANDY, covariances and more generally on Total Monte-Carlo
- [Riku Tuominen](#) (VTT) for sharing with us his `runnjoy_kermas.pl` Perl script that enables SERPENT2 precise energy deposition model
- [Nicolò Abrate](#) (Politecnico di Torino) for releasing his NDL software (Nuclear Data Library processing tools) which helped us ordering the NJOY module calls, in particular regarding PURR, HEATR and GASPR
- [Paul Romano](#) (ANL) for his fruitful remarks on THERMR

Introduction

Thermal scattering

Impact of energy  
deposition model

Propagation of  
nuclear data  
uncertainties

Conclusions



**POLYTECHNIQUE  
MONTREAL**

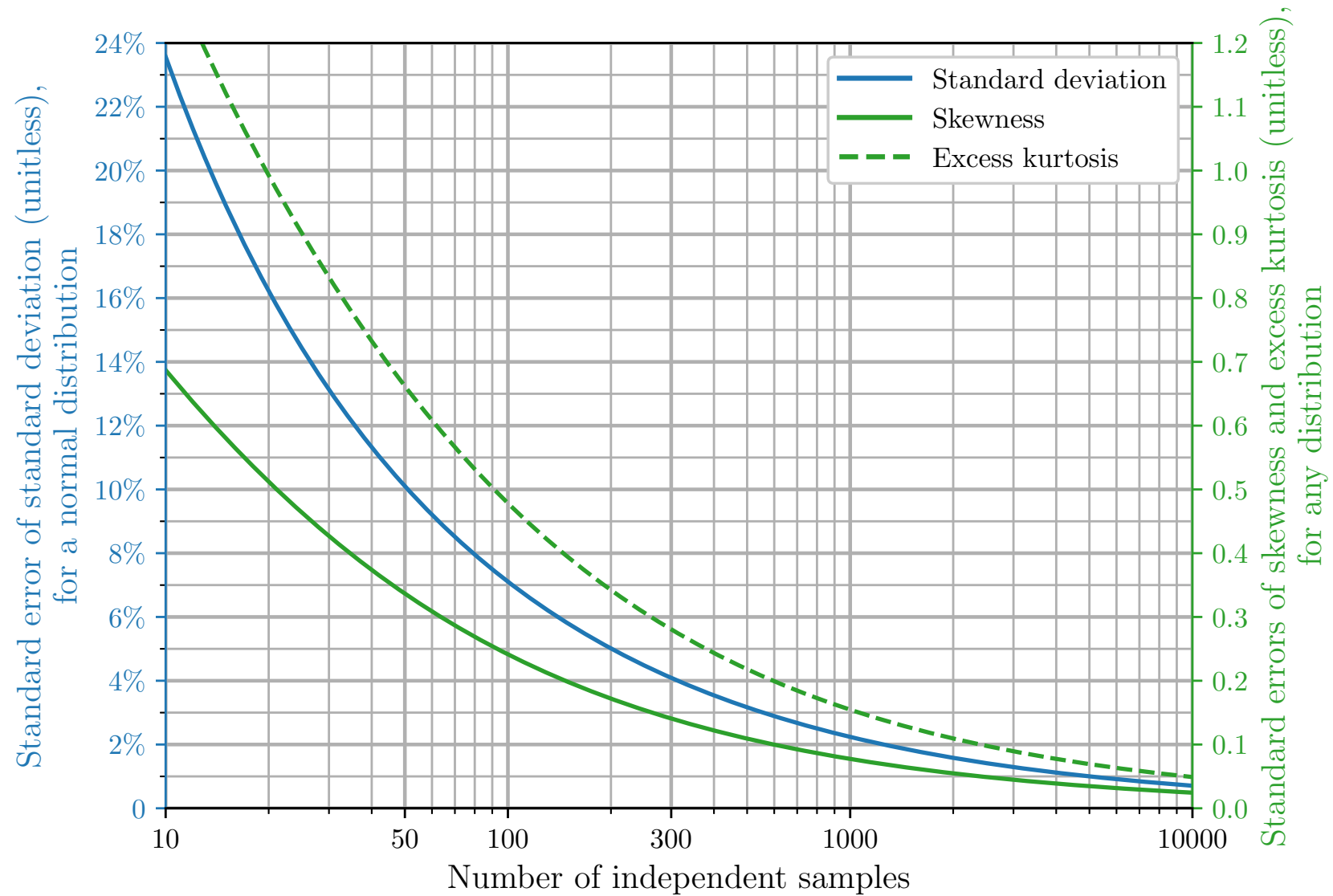


**RÉPUBLIQUE  
FRANÇAISE**

*Liberté  
Égalité  
Fraternité*

## Backup slides

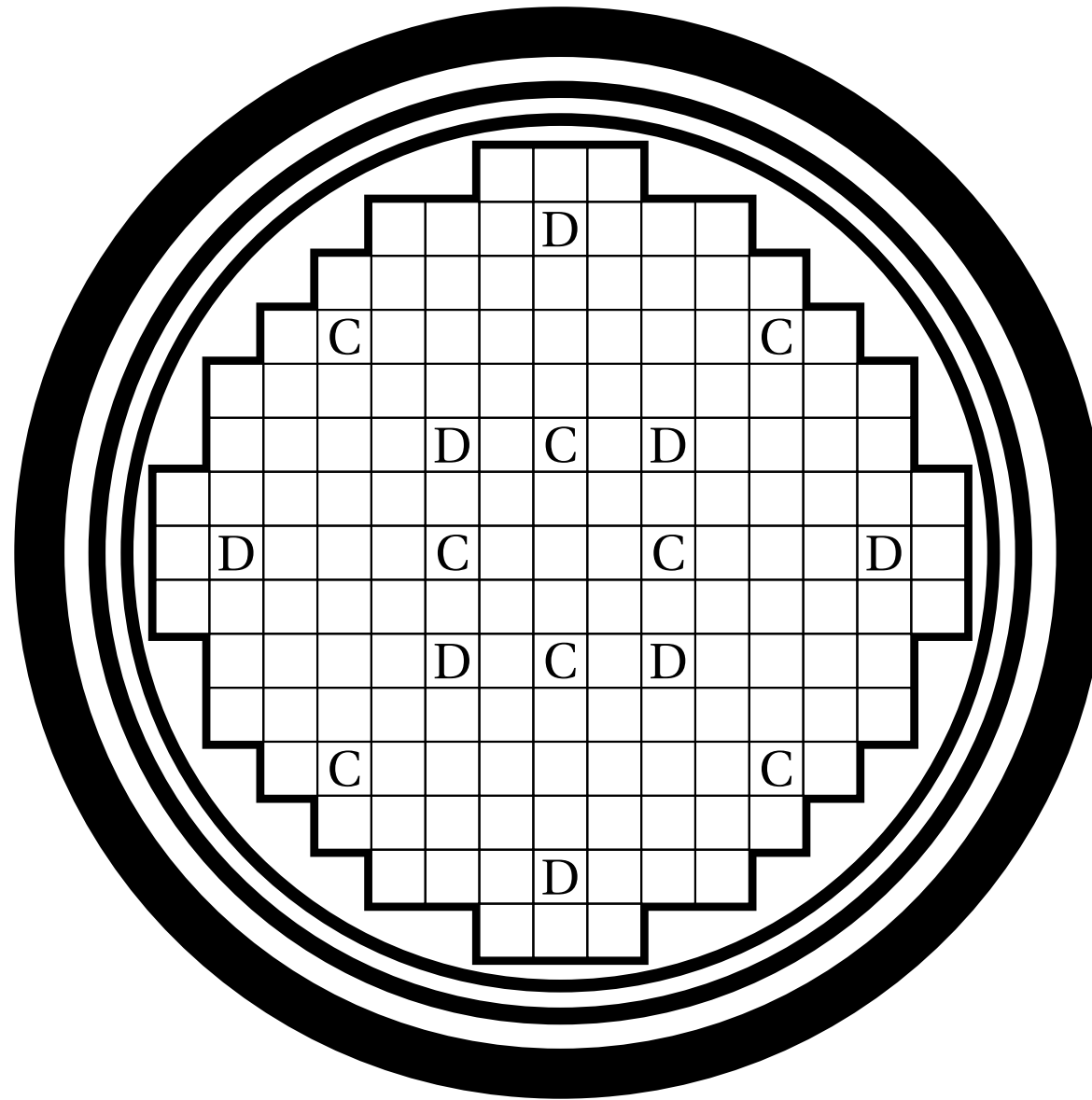
# Standard error on standard deviation (left axis), on skewness and on excess kurtosis (right axis)



# Importance of power distribution

- Most of the research in the field of uncertainty is focused on  $k_{\text{eff}}$ , yet, for reactors
  - nuclear safety depends **not much** on it
- Focusing on **power distribution**
  - nuclear safety depends **heavily** on it  
(KERKAR and PAULIN, 2008)

# C and D control rods banks in Tihange-1





## Equivalence methods for the reflector

- Equivalences used for fuel → not applicable to the reflector, as they typically conserve
  - diffusion properties (diffusion coefficients),
  - reaction rates (SPH)
- Conserving reaction rates **in** the reflector or neutron scattering **in** the reflector is of little interest
- Rather wishes to maintain its reflective properties against an **external source** of neutrons
- Specific equivalences → determining cross sections and diffusion coefficients preserving reflective properties of a (transport) reference

## Lefebvre-Lebigot method (1/2)

→ Analytical solution of the current **at the interface** ( $x = 0$ )

$$\frac{J_1}{\phi_1} = \sqrt{D_1 (\Sigma_{a1} + \Sigma_{1 \rightarrow 2})} \quad (1)$$

and

$$\frac{J_2}{\phi_1} = \frac{\phi_2}{\phi_1} \sqrt{D_2 \Sigma_{a2}} - \frac{\Sigma_{1 \rightarrow 2} \sqrt{D_1 D_2}}{\sqrt{\Sigma_{a2} D_1} + \sqrt{(\Sigma_{a1} + \Sigma_{1 \rightarrow 2}) D_2}} \quad (2)$$

- After analytical study, numerical experiments with a  $S_N$  transport code on a representative heterogeneous traverse
- Reflector applicable for all fuel bundles,  $\forall$  temperature, burnup, etc.
  - reflector response for fuel varying as such

1) Fast group →  $J_1 / \phi_1$  is constant

$$R_1 = \frac{J_1}{\phi_1} \quad (3)$$

## Lefebvre-Lebigot method (2/2)

### 2) Thermal group

→  $J_2/\phi_1$  depends on the fuel

$$\frac{J_2}{\phi_1} = -R_3 + R_2 \frac{\phi_2}{\phi_1} \quad (4)$$

→ **5 unknowns** and **3 parameters** ( $R_1$ ,  $R_2$  et  $R_3$ ) calculated in  $S_N$ ,  
Lefebvre and Lebigot suggest to equalize  $D_g$  :

$$D_1^{\text{fuel}} \approx 1.3 \text{ cm}$$

$$D_1^{\text{reflector}} = \mathbf{1.3 \text{ cm}}$$

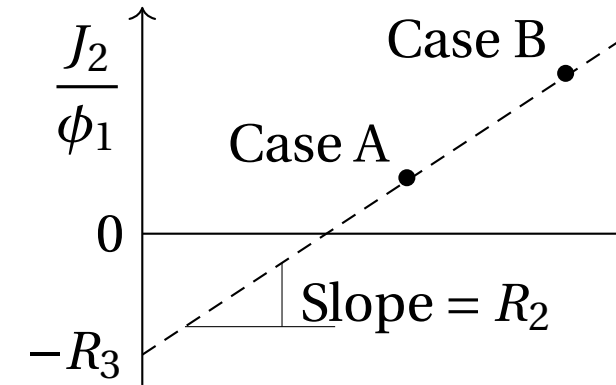
$$D_2^{\text{fuel}} \approx 0.4 \text{ cm}$$

$$D_2^{\text{reflector}} = 0.4 \text{ cm} \quad (5)$$

$$\Sigma_{a2} = \frac{(R_2)^2}{D_2}$$

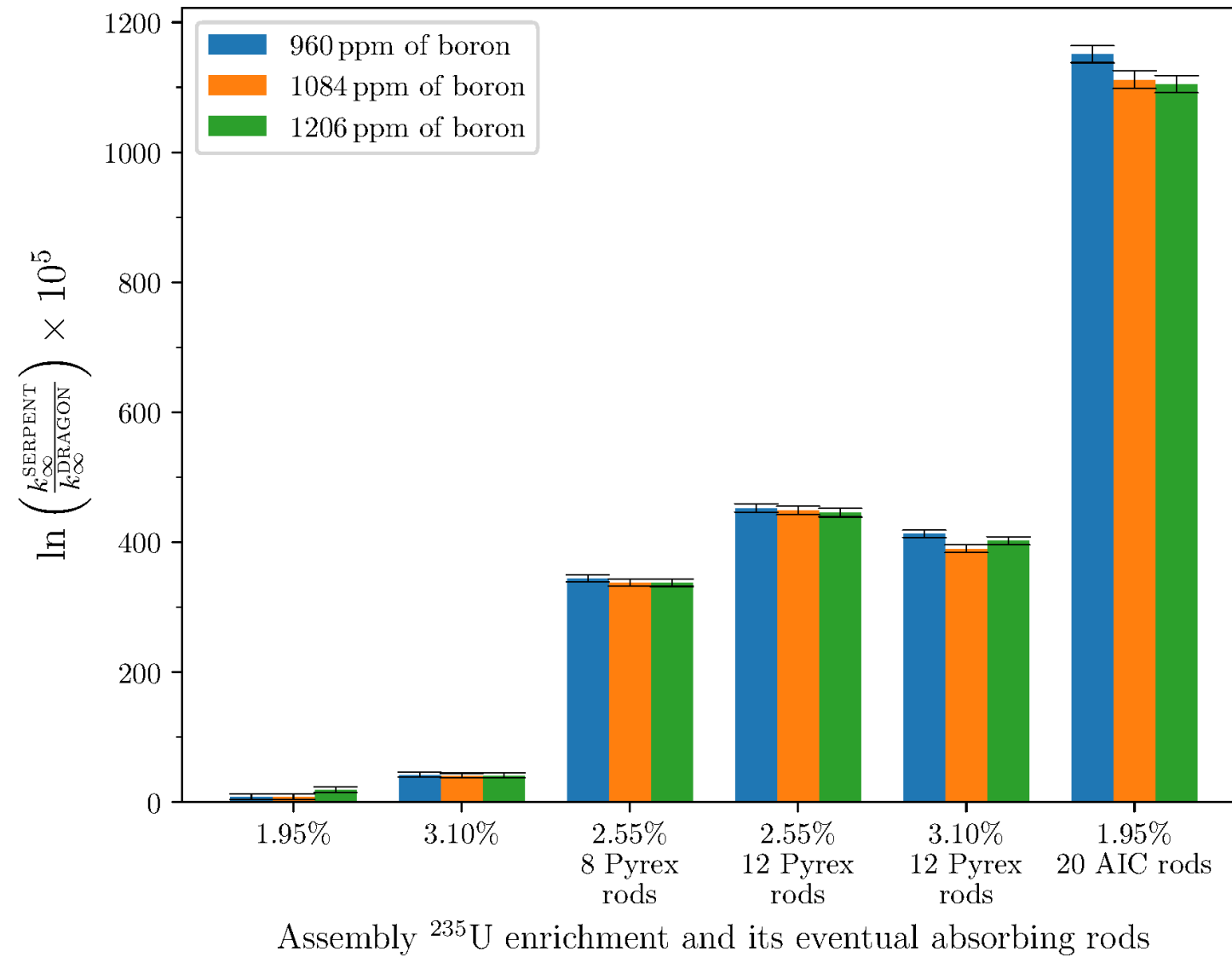
$$\Sigma_{1 \rightarrow 2} = R_3 \left( \frac{R_1}{D_1} + \sqrt{\frac{\Sigma_{a2}}{D_2}} \right) \quad (6)$$

$$\Sigma_{a1} = \frac{(R_1)^2}{D_1} - \Sigma_{1 \rightarrow 2} \quad (7)$$

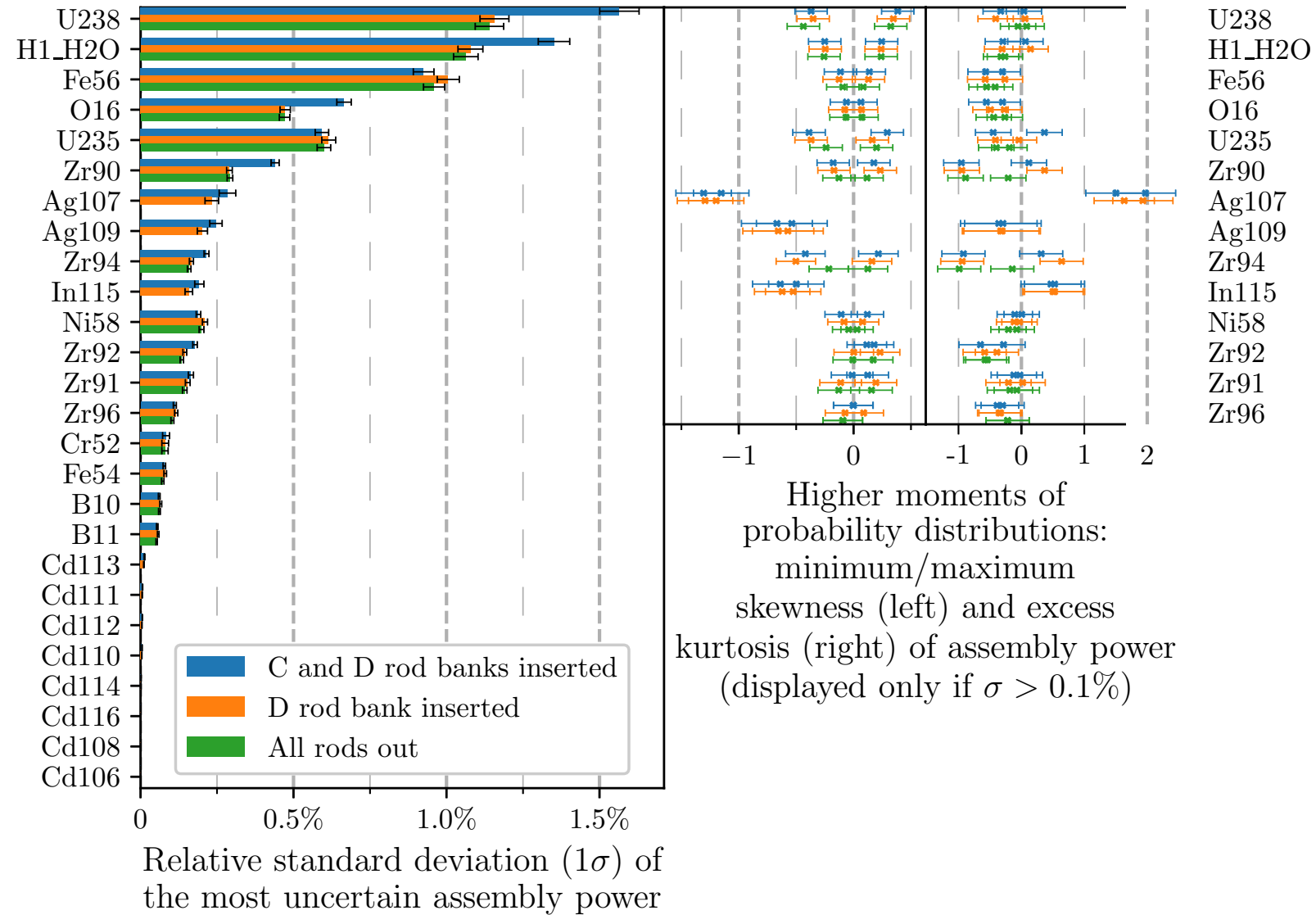




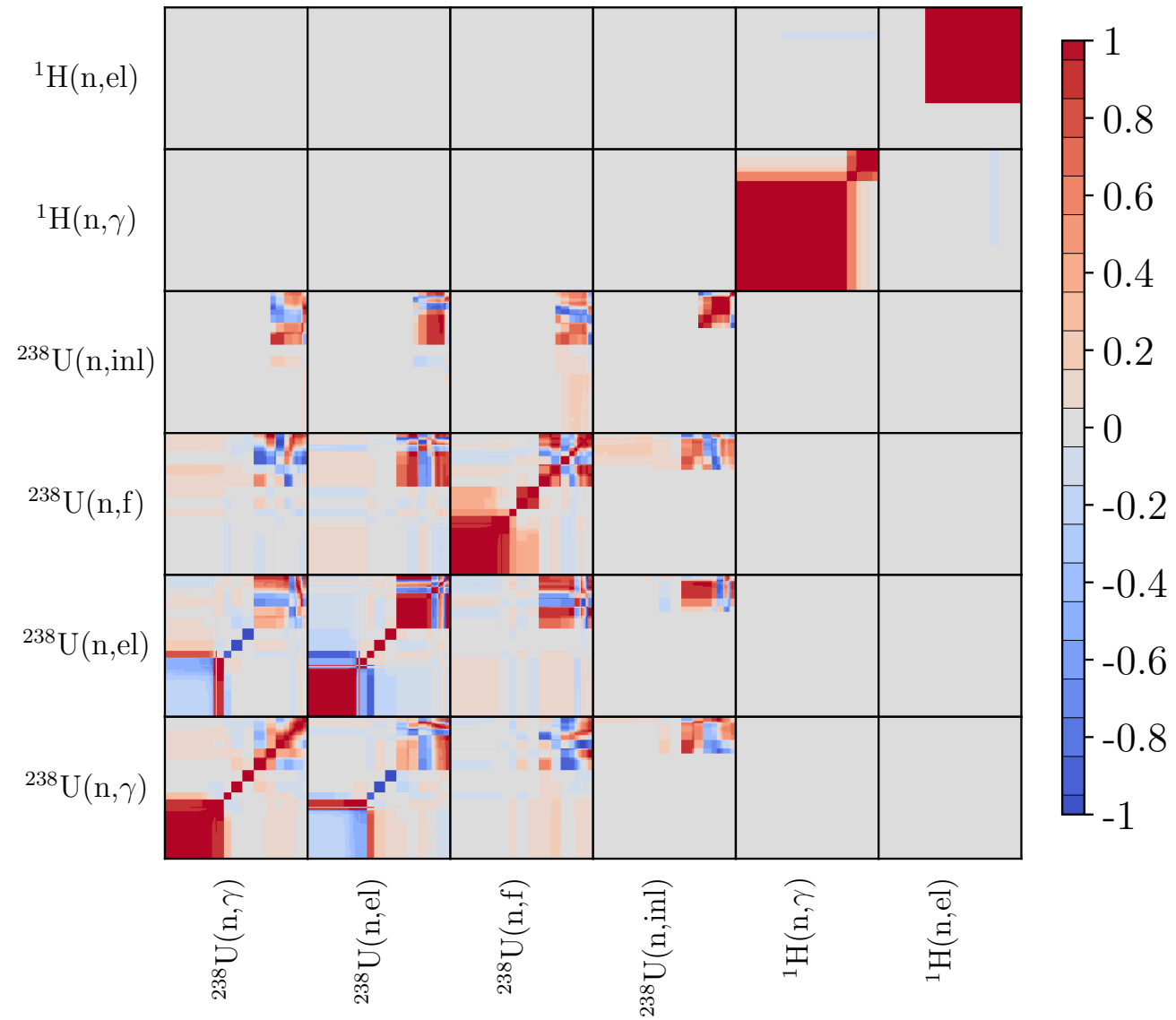
# $k_{\infty}$ discrepancy between Serpent and Dragon (pcm), for assemblies during 1<sup>st</sup> start-up of Tihange-1



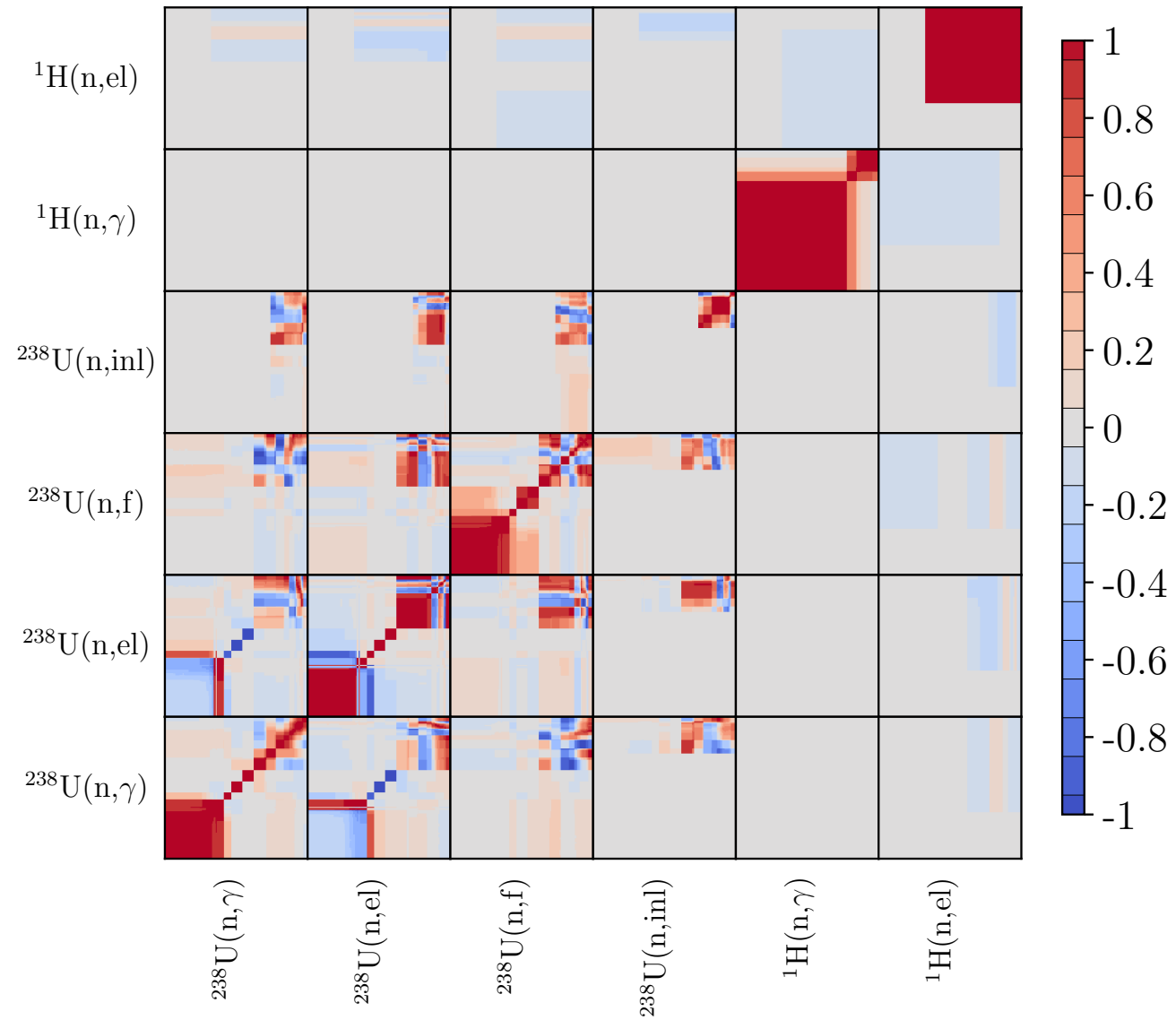
# Ranking of uncertainties in Tihange-1, by isotope



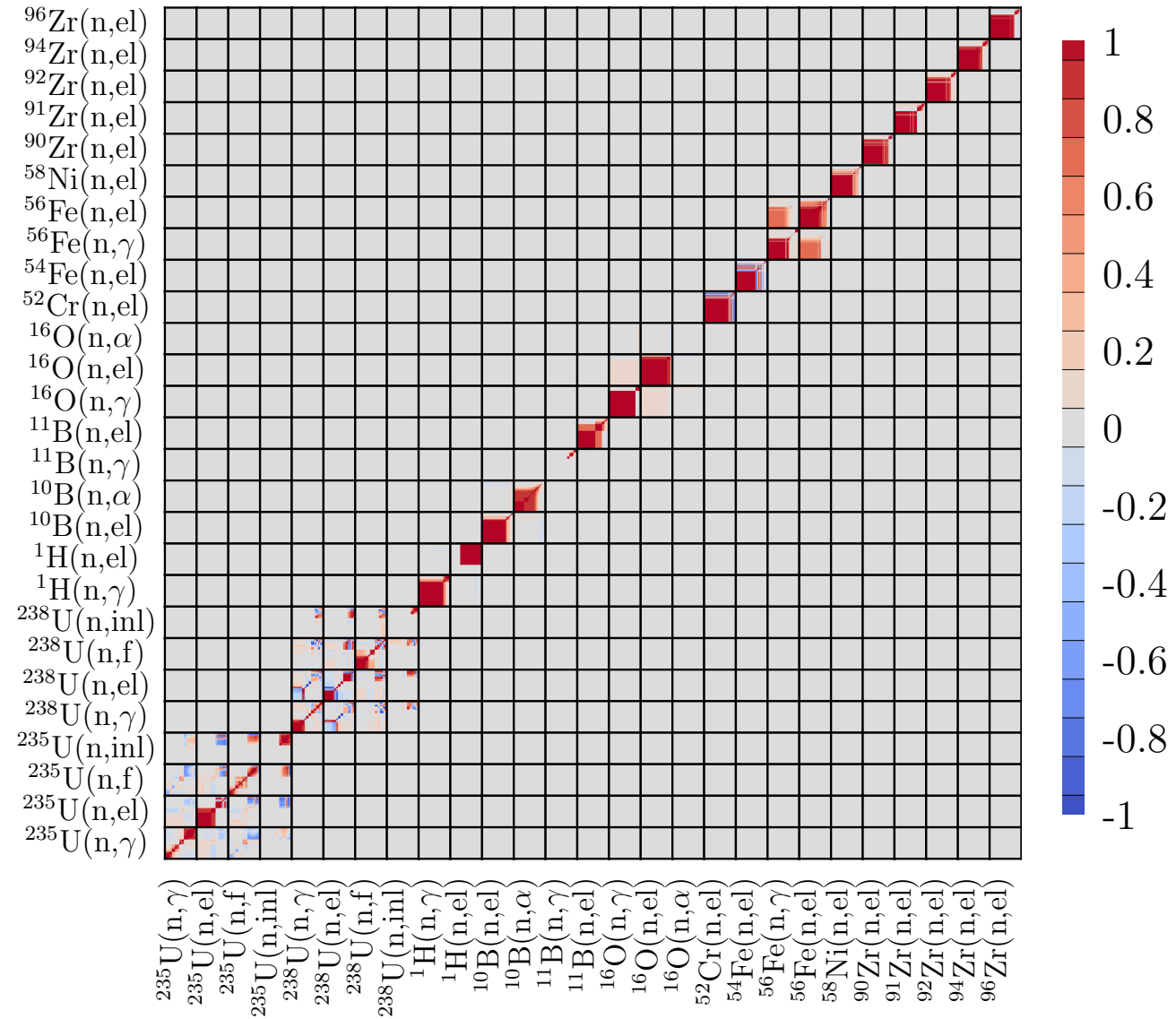
# Prior correlations between the uncertainties of hydrogen and uranium 238



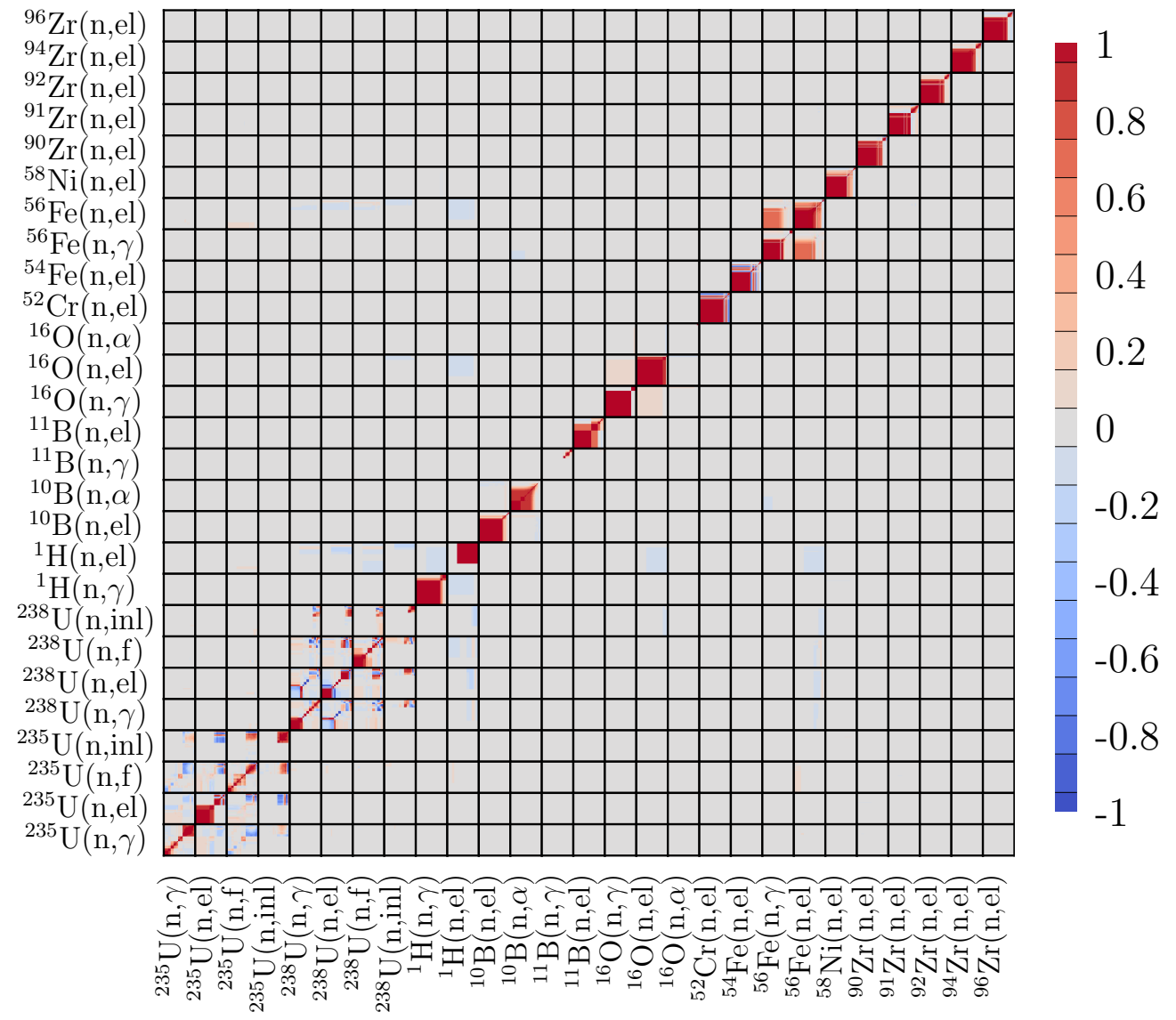
# Posterior (BFMC) correlations between the uncertainties of hydrogen and uranium 238



# Prior correlations between the uncertainties of different isotopes



# Posterior (BFMC) correlations between the uncertainties of different isotopes



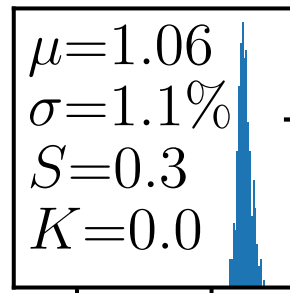
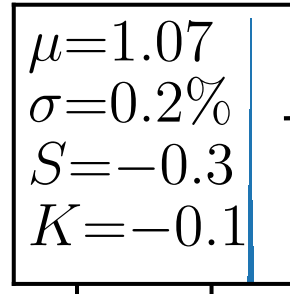
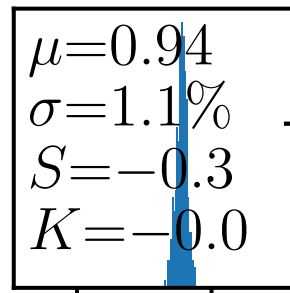
## Evaluation of nuclear data **uncertainties**

- 1) JEFF-3.3 or ENDF-B/VIII.0 evaluations → covariance matrices
  - Multigroups → piecewise constant
  - Sometimes missing or incomplete (minor isotopes)
- 2) TENDL-2019 evaluation → some various benefits
  - Very **complete uncertainties** (angular distributions, etc), continuously varying, not necessarily Gaussian
  - Above all, **single code** for production → high format homogeneity → **safer** and faster
  - TALYS optical models → **no evaluation** for less than 20 nucleons (hydrogen, boron, oxygen...) → JEFF-3.3
  - **No evaluation** either for some heavy isotopes, the most important ones : uranium 235, 238... → JEFF-3.3

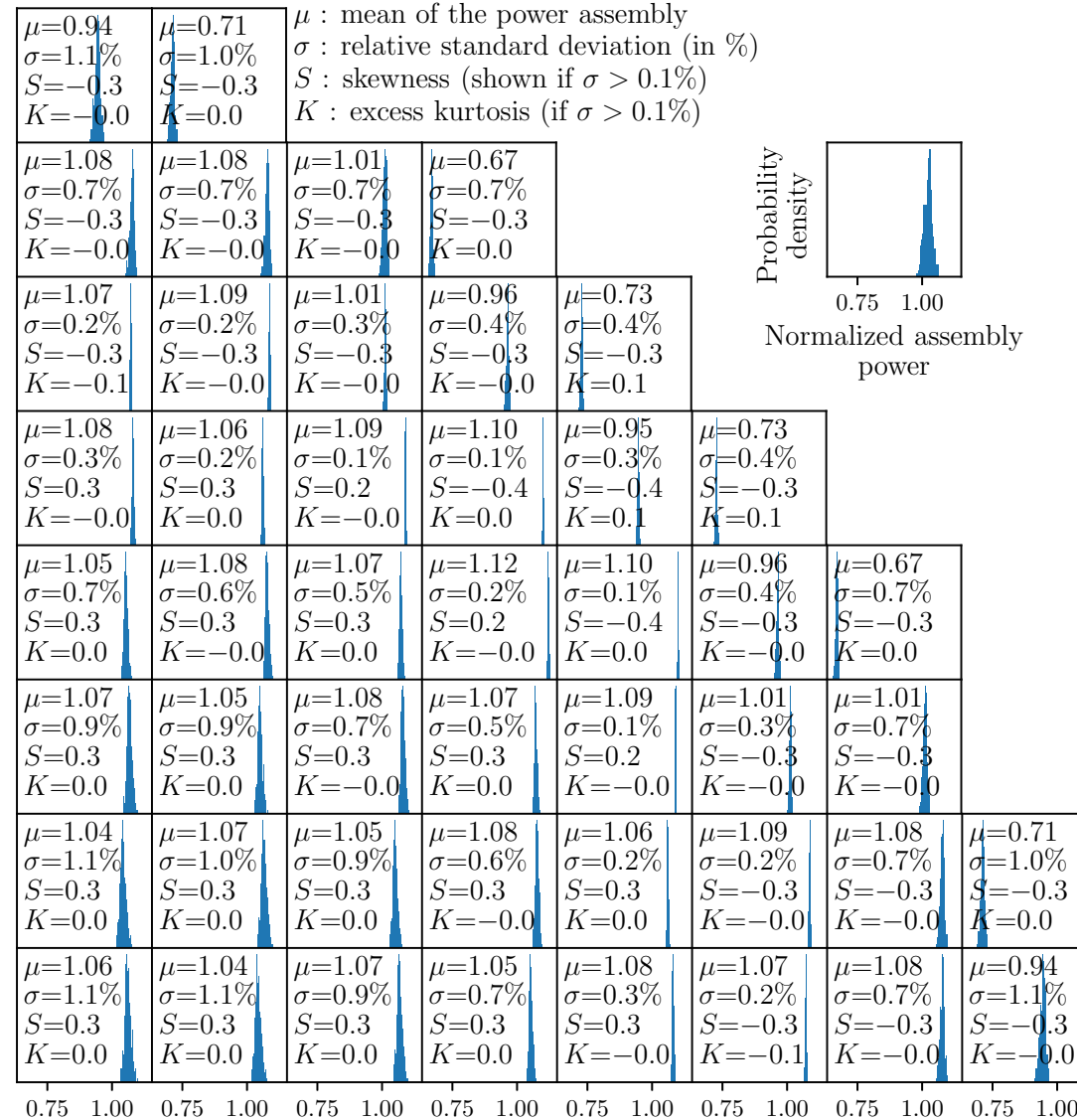
**The idea is not to demonstrate the qualities or defects of JEFF-3.3 or TENDL-2019 : just a choice**

# Propagation of uncertainties from uranium 238, JEFF-3.3

## 300 samples of nuclear data (with SANDY)



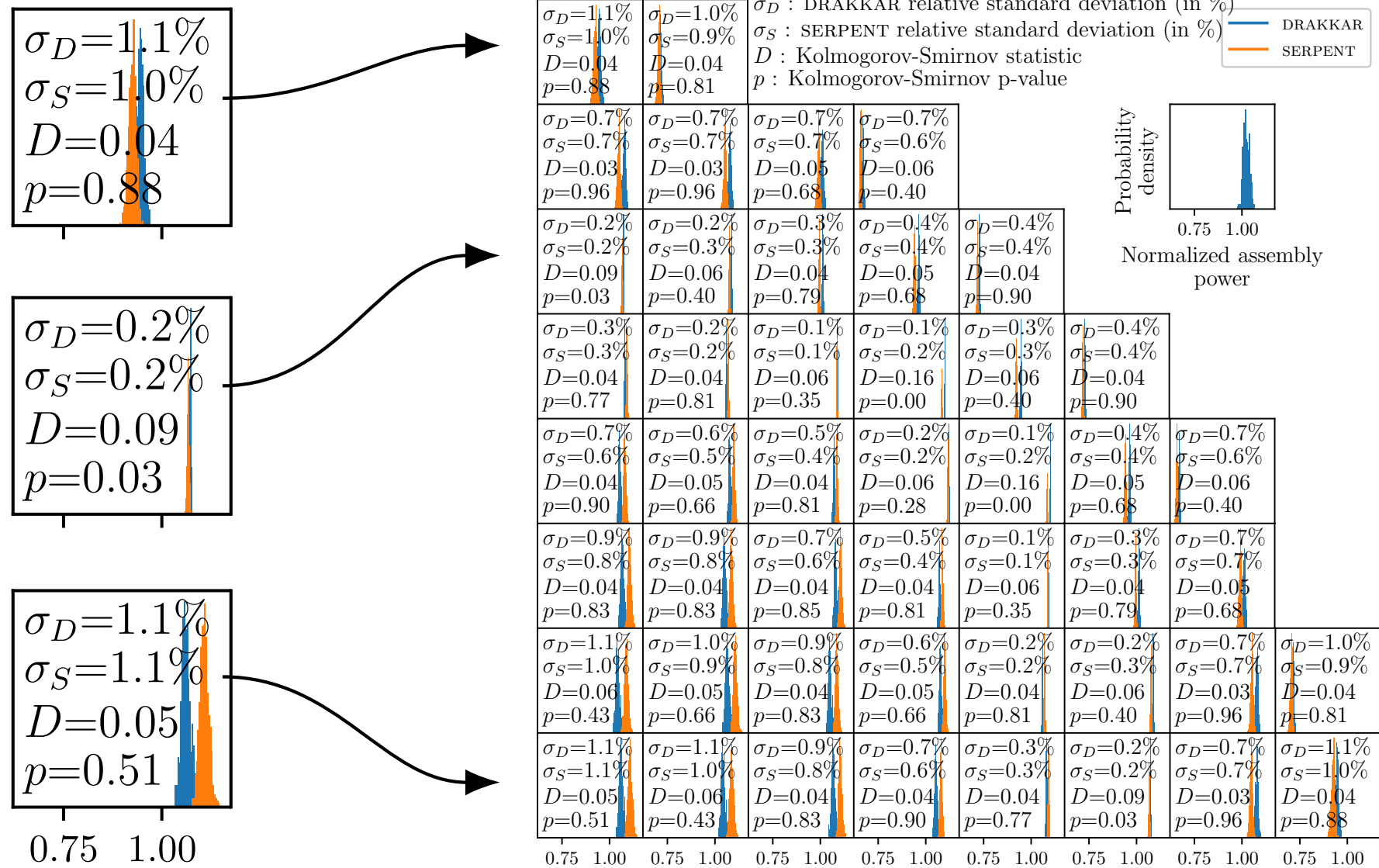
0.75 1.00



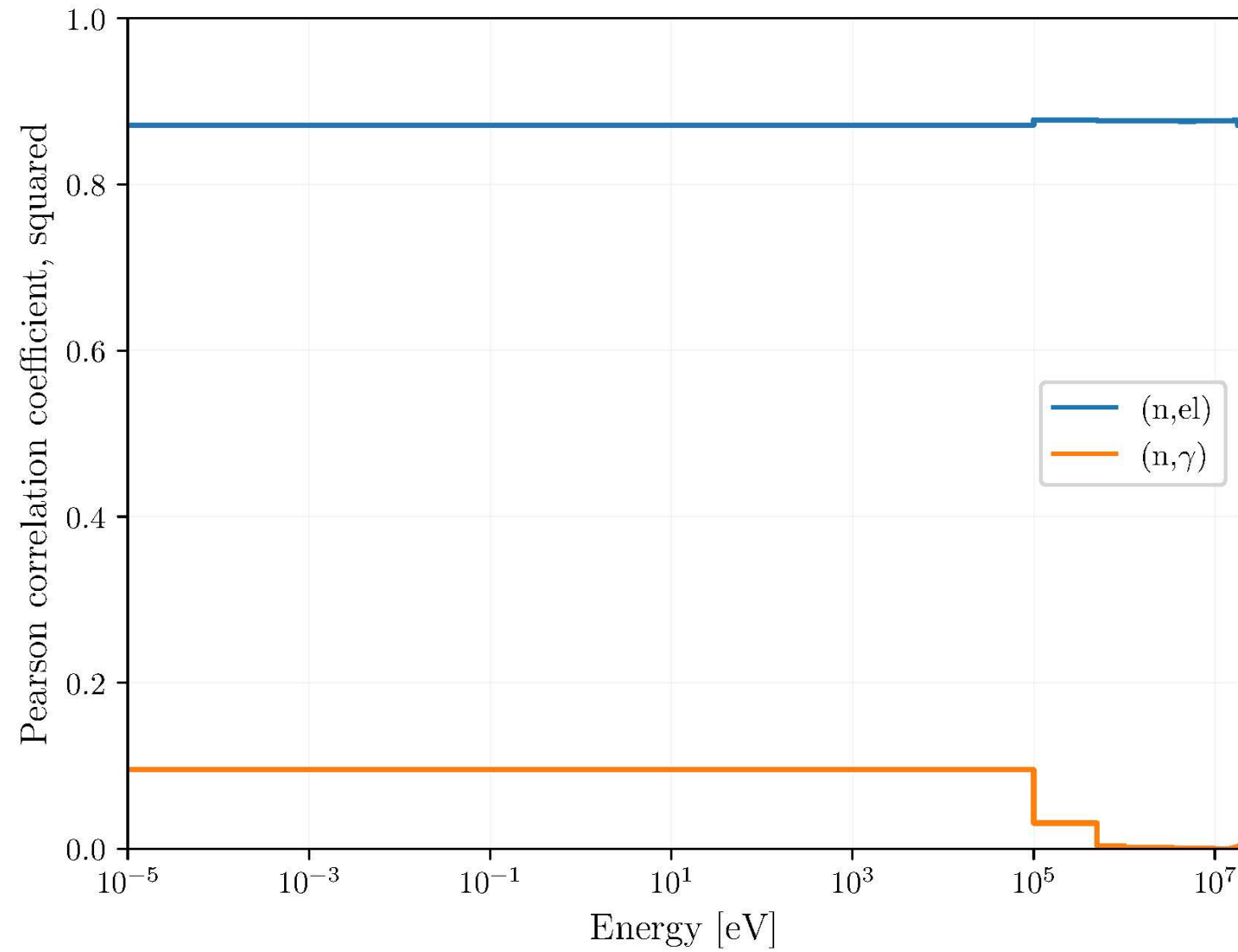


# Propagation of uncertainties from uranium 238, JEFF-3.3

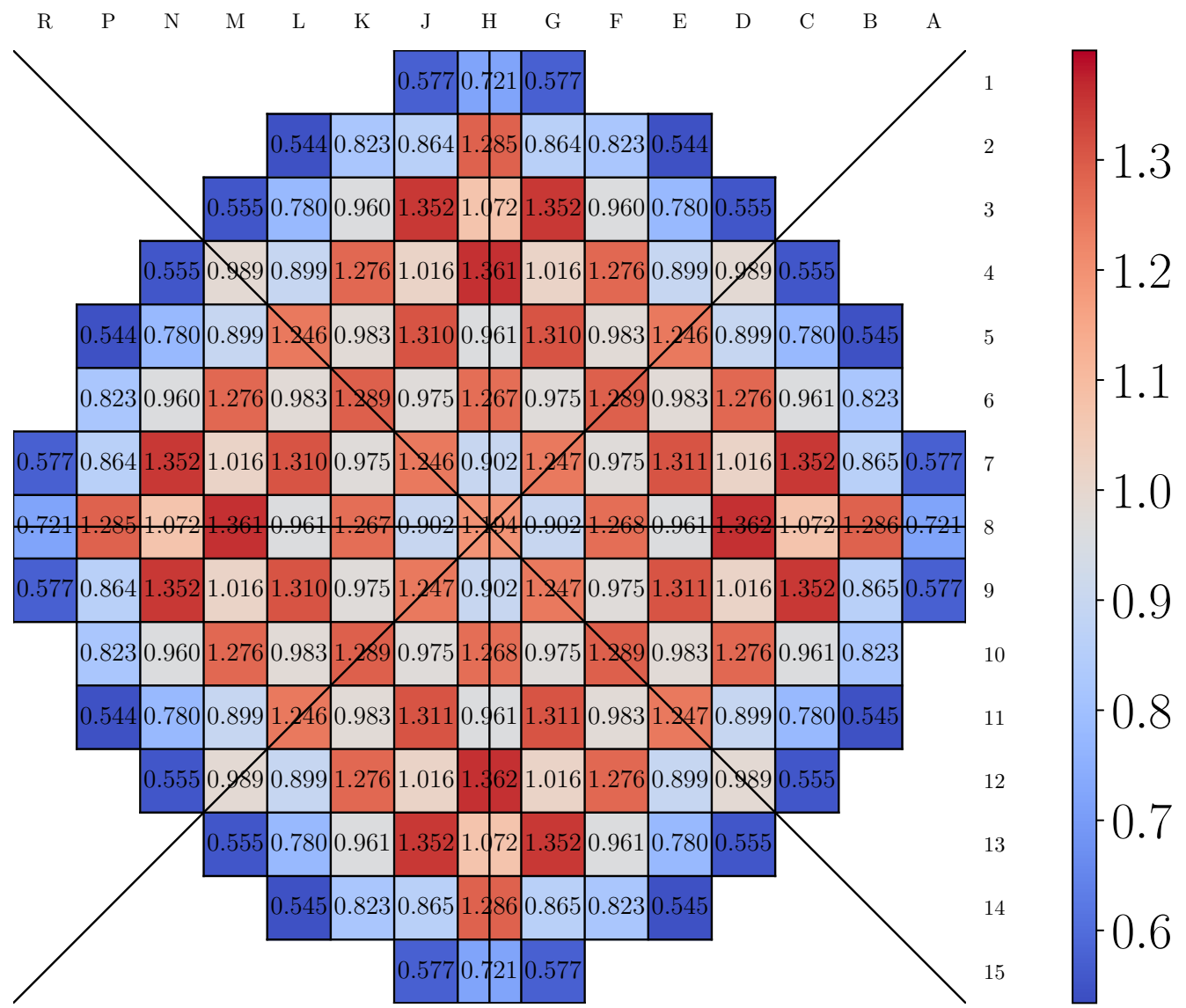
## Deterministic and Monte-Carlo comparison



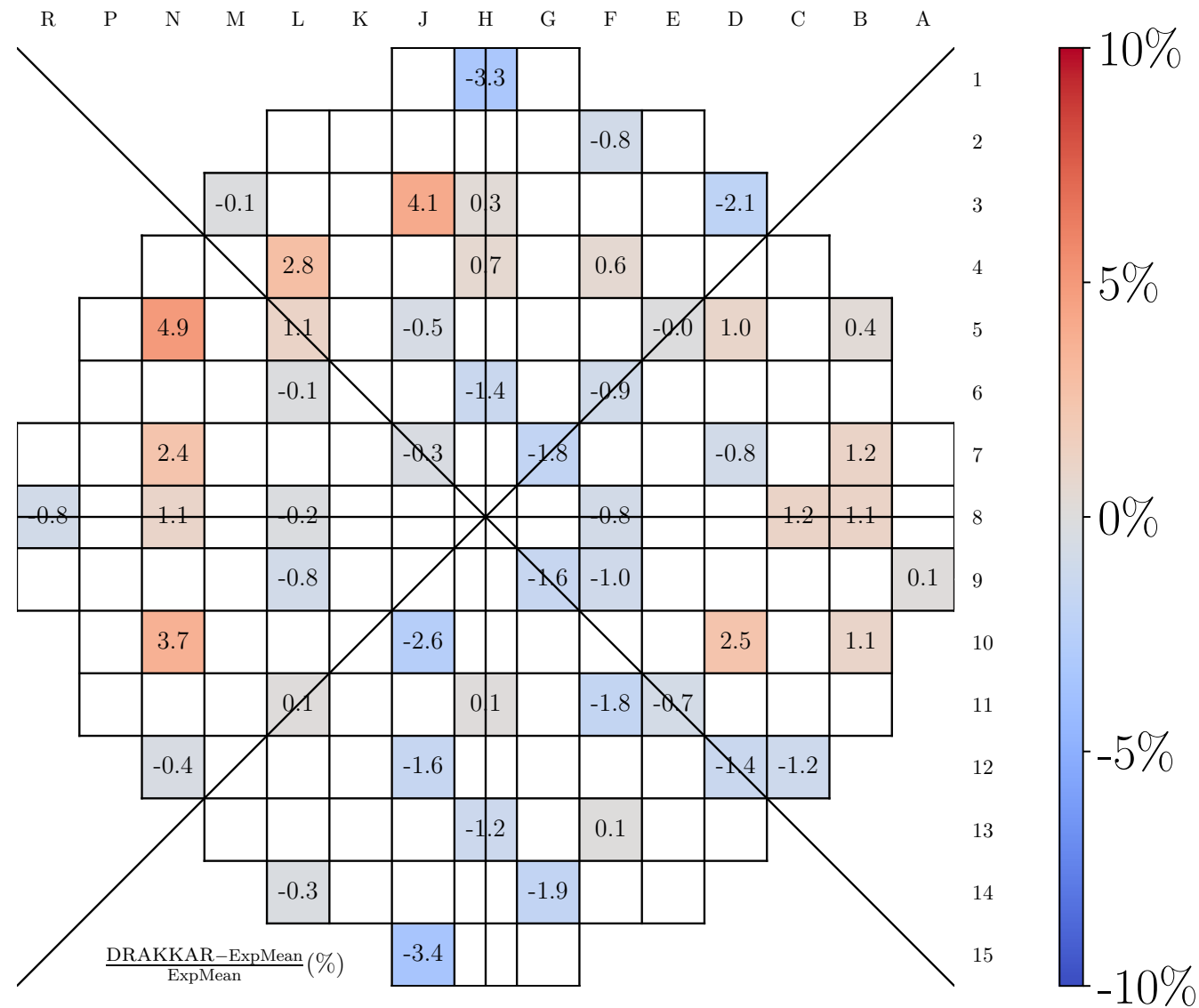
# Hydrogen uncertainty decomposition with respect to reaction and energy



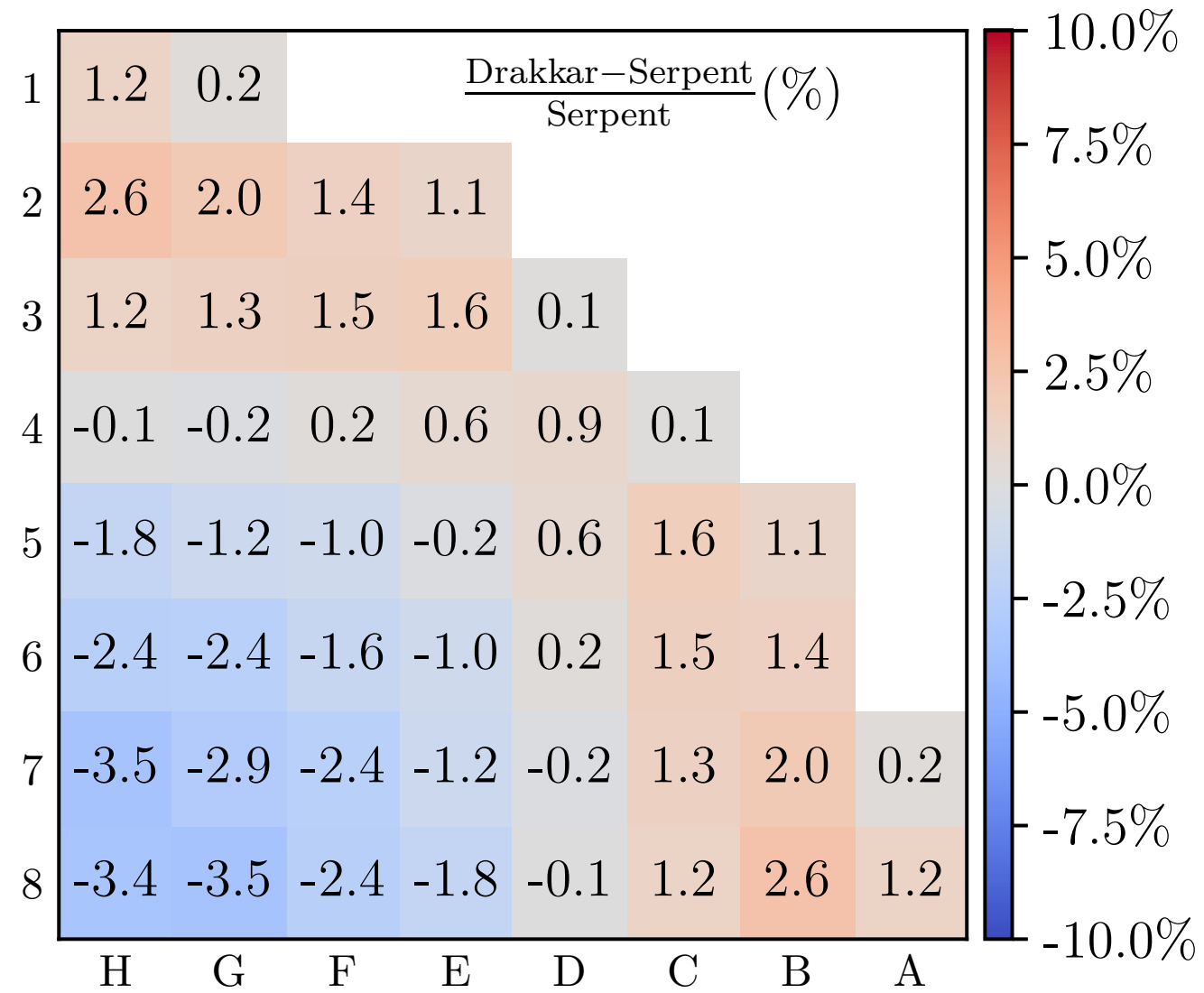
# Drakkar detector responses (JEFF-3.3)



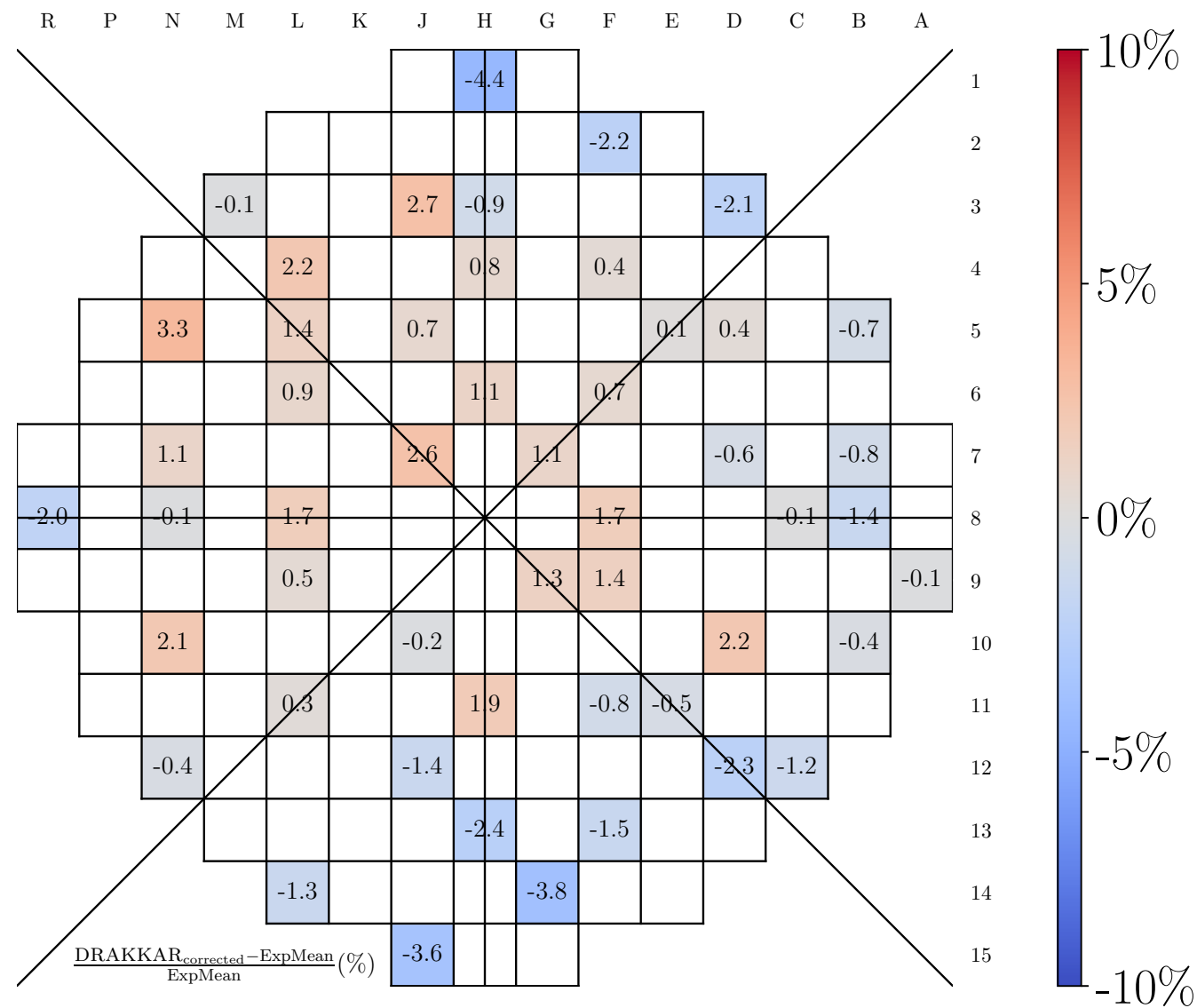
# Discrepancy between Drakkar (JEFF-3.3) and experimental mean



# Power distribution discrepancy between Drakkar and Serpent, on Bugey-2 and Fessenheim-1 and 2

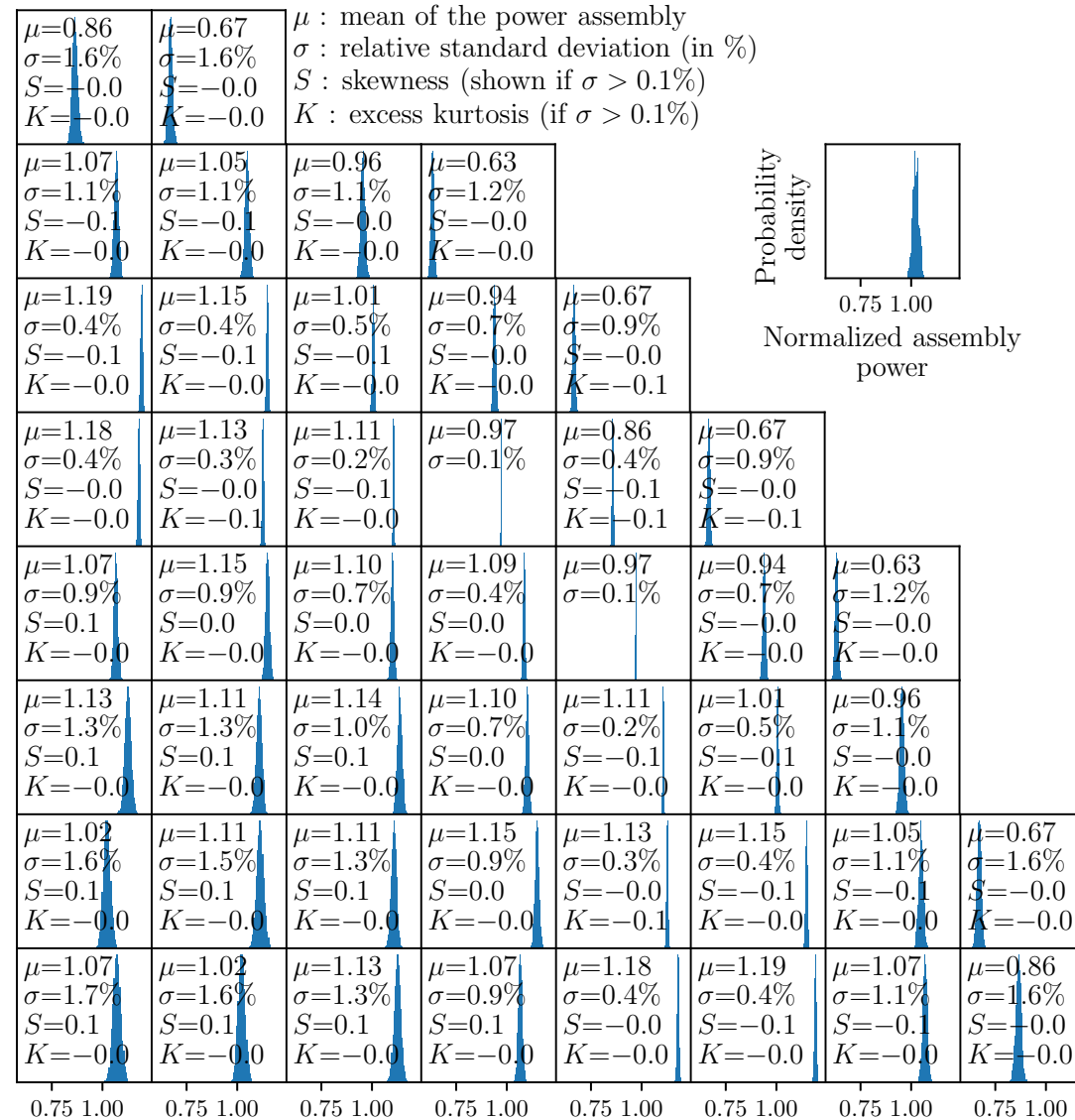
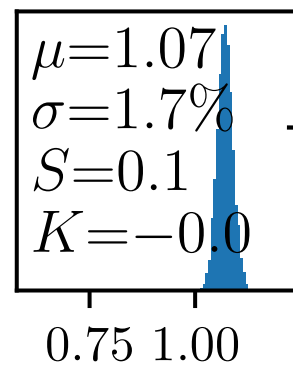
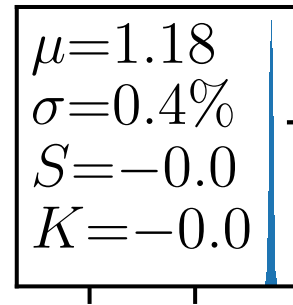
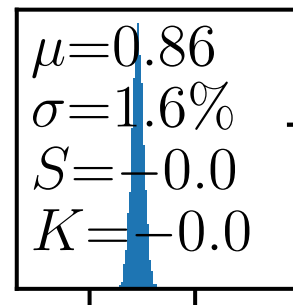


# Discrepancies between Drakkar (corrected from deterministic bias, JEFF-3.3) and experimental mean



# Simultaneous sampling : interactions between isotopes

## Power distribution



## Adjustment with BMC method (Bayesian Monte-Carlo)

- Proximity between the measurement and a sample  $k$  of nuclear data estimated with

$$\chi_k^2 = \sum_{i=1}^{n_{\text{mes}}} \left( \frac{C_{i,k} - E_i}{\sigma_{E_i}} \right)^2 \quad (8)$$

- Lack of uncertainty estimated by experimentalists
  - expert opinion, for lack of a better choice

$$\sigma_E = 2\% \quad (9)$$

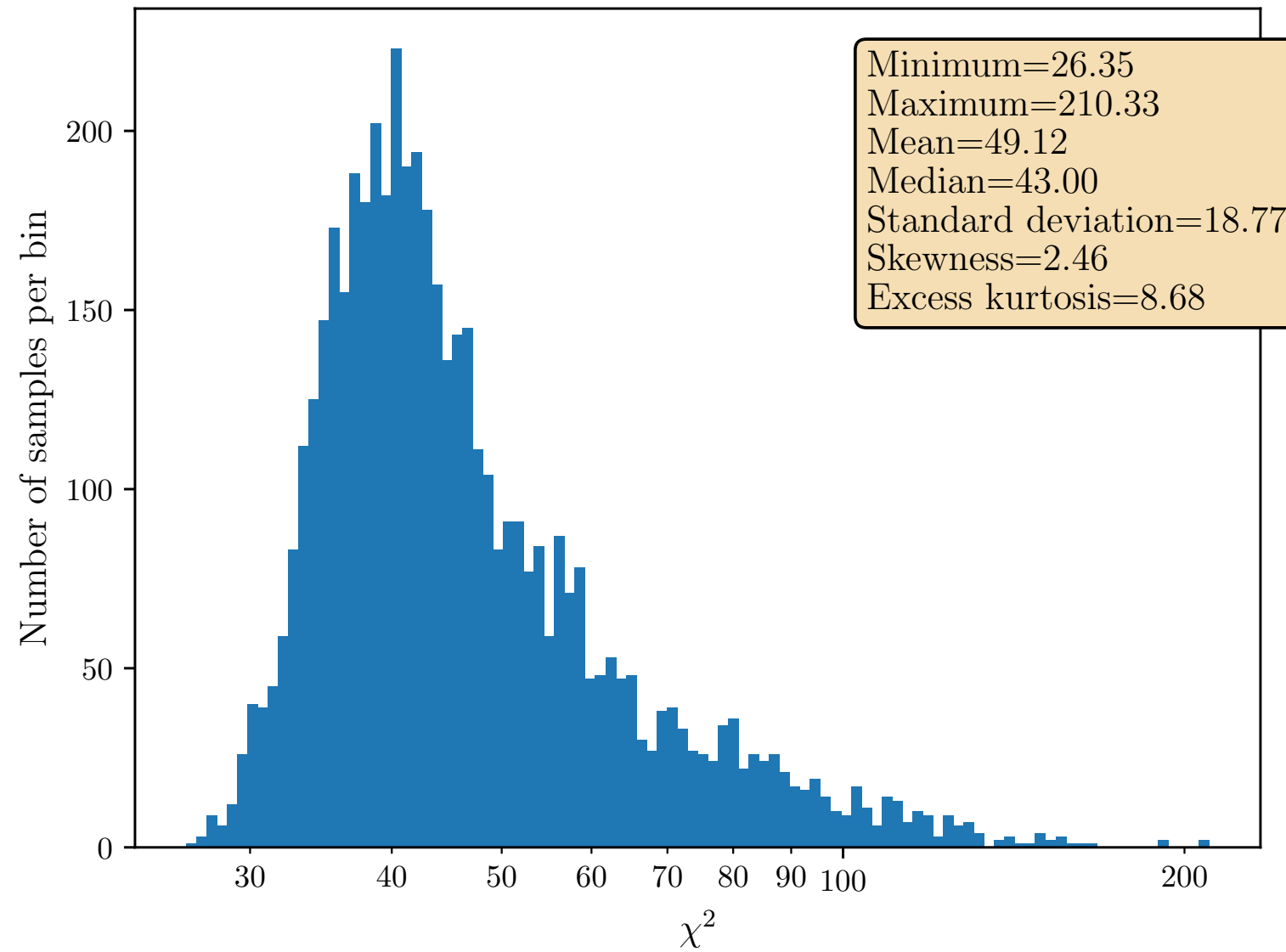
- Each nuclear data sample  $k$  → weight

$$w_k = \frac{L(\mathbf{p}_k, \mathbf{x})}{\sum_{\kappa=1}^n L(\mathbf{p}_\kappa, \mathbf{x})} = \frac{\exp(-\chi_k^2/2)}{\sum_{\kappa=1}^n \exp(-\chi_\kappa^2/2)} \quad (10)$$

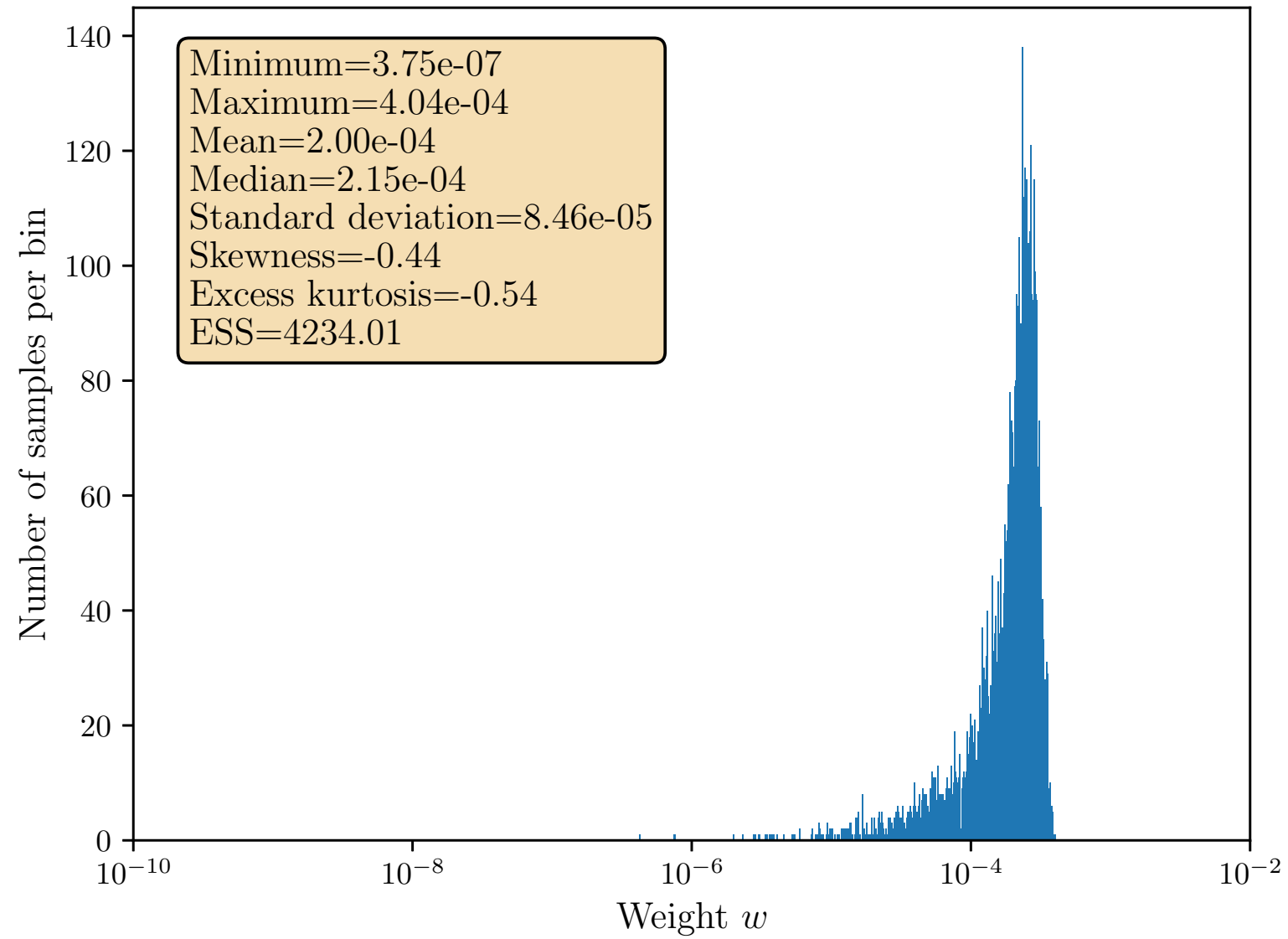
where  $L(\mathbf{p}_k, \mathbf{x})$  → likelihood of the  $\mathbf{x}$  experimental observations for a  $k$  sample of  $\mathbf{p}$  nuclear data



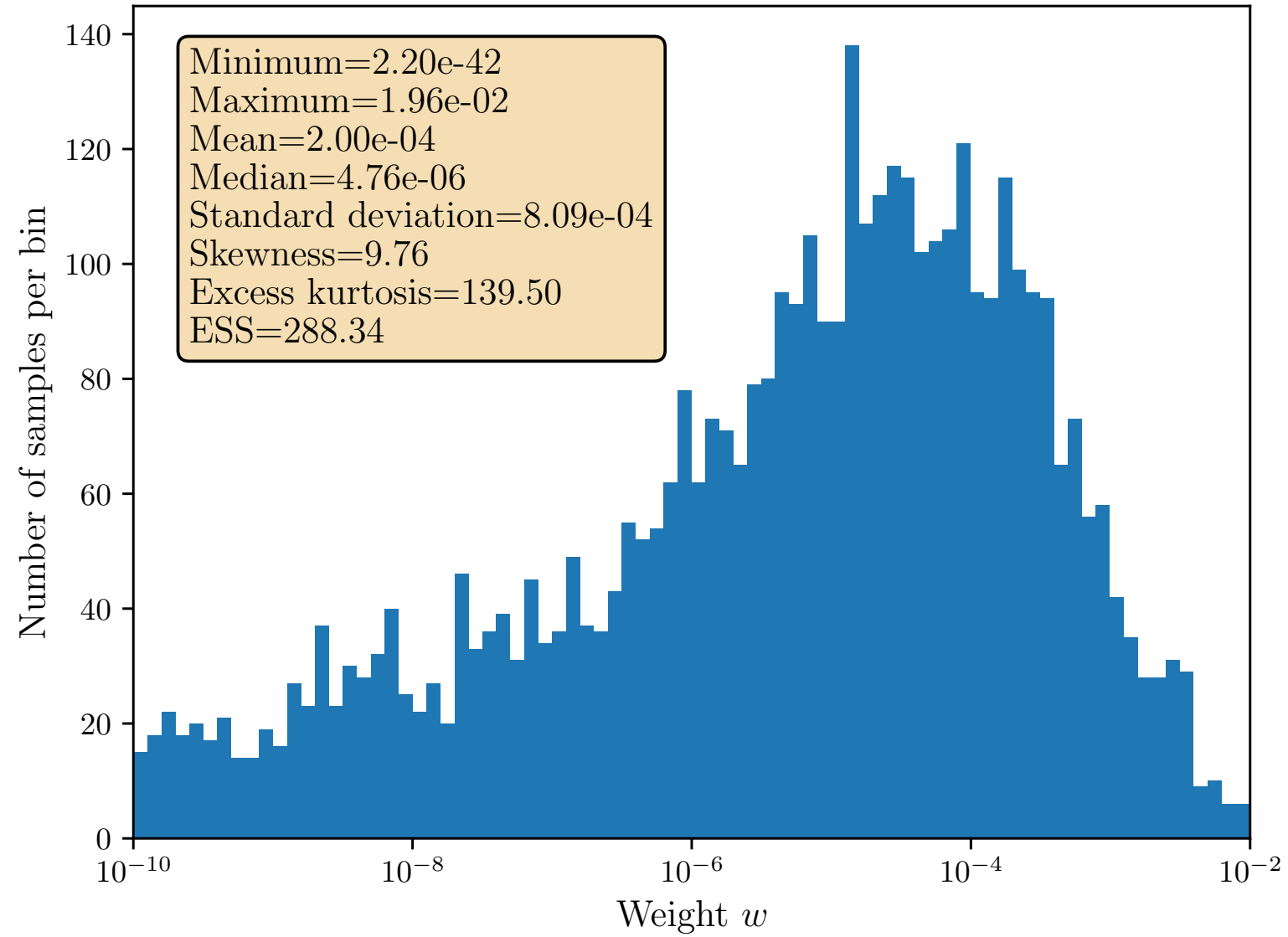
# $\chi^2$ histogram



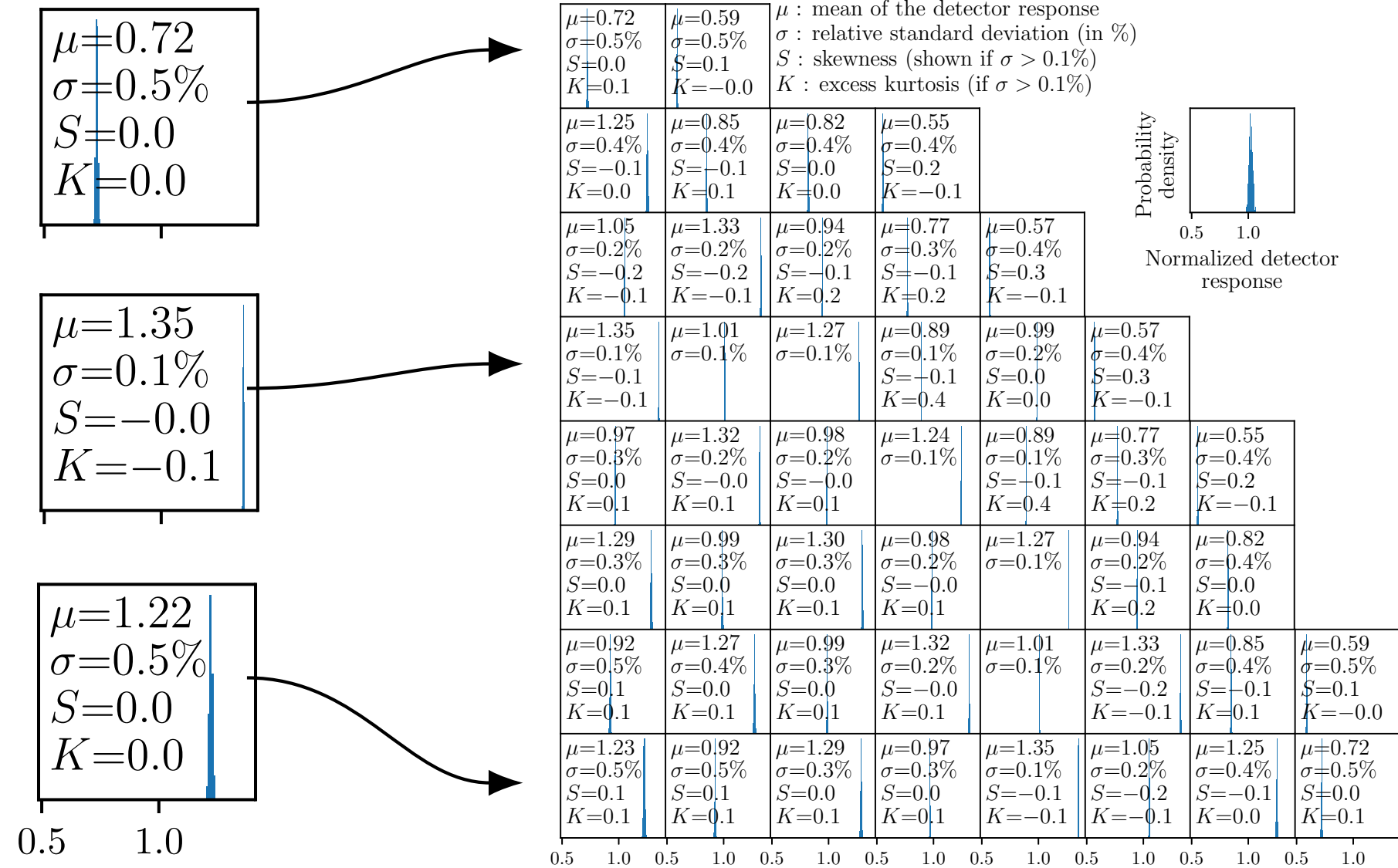
# $w$ weight histogram obtained with BFMC



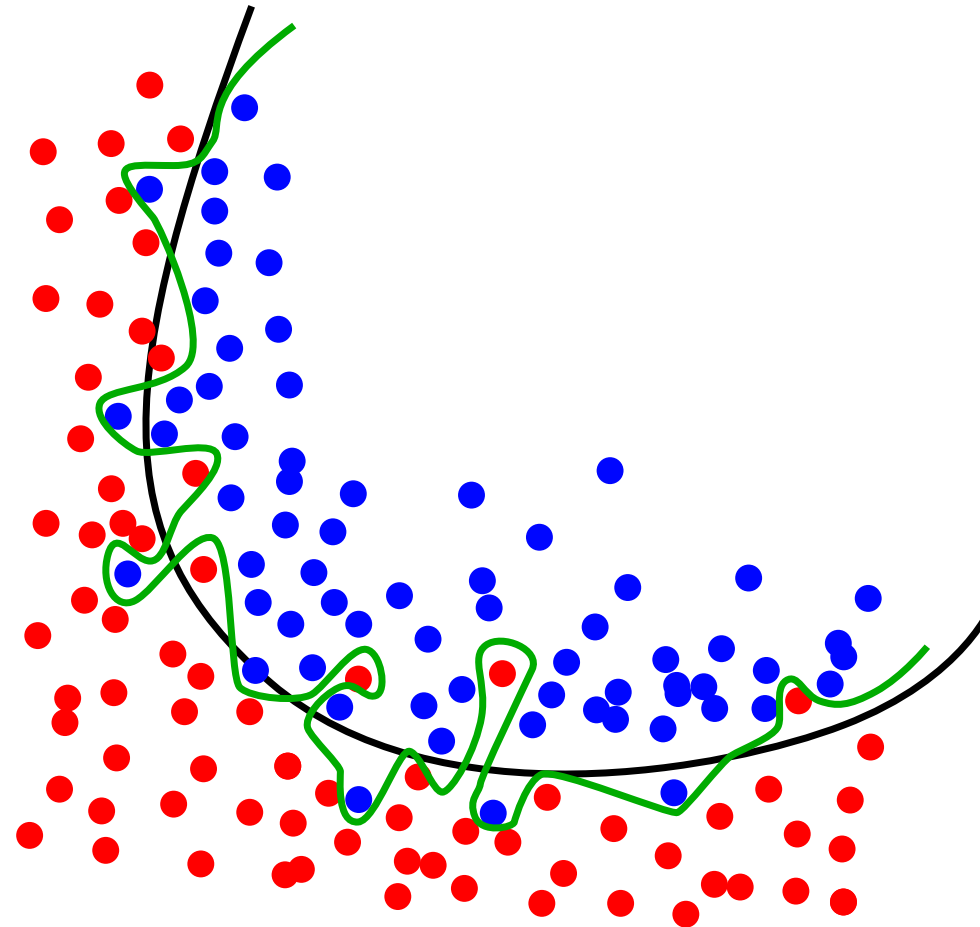
# $w$ weight histogram obtained with BMC



# Posterior (BMC) detector responses

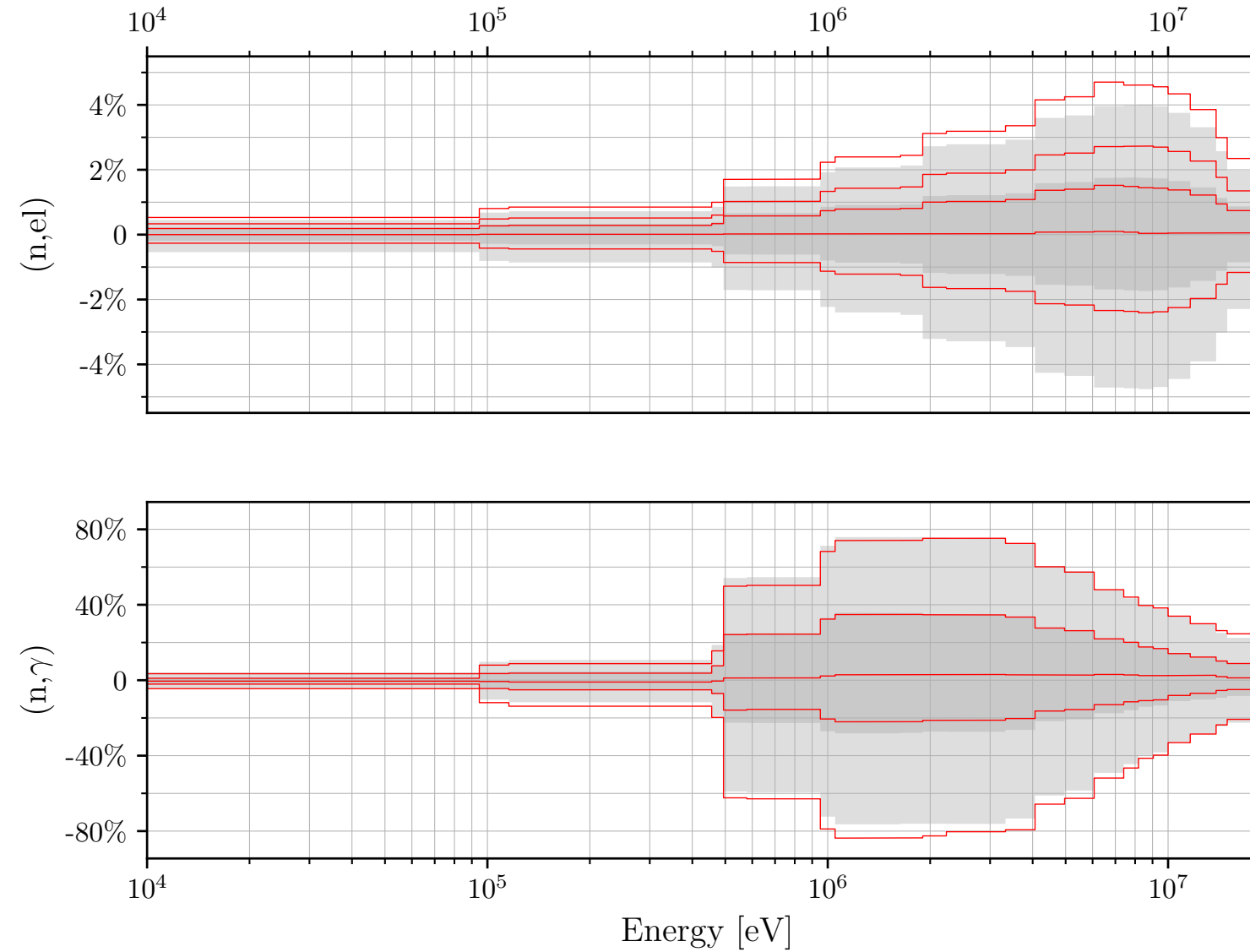


# Overfitting illustration, in green

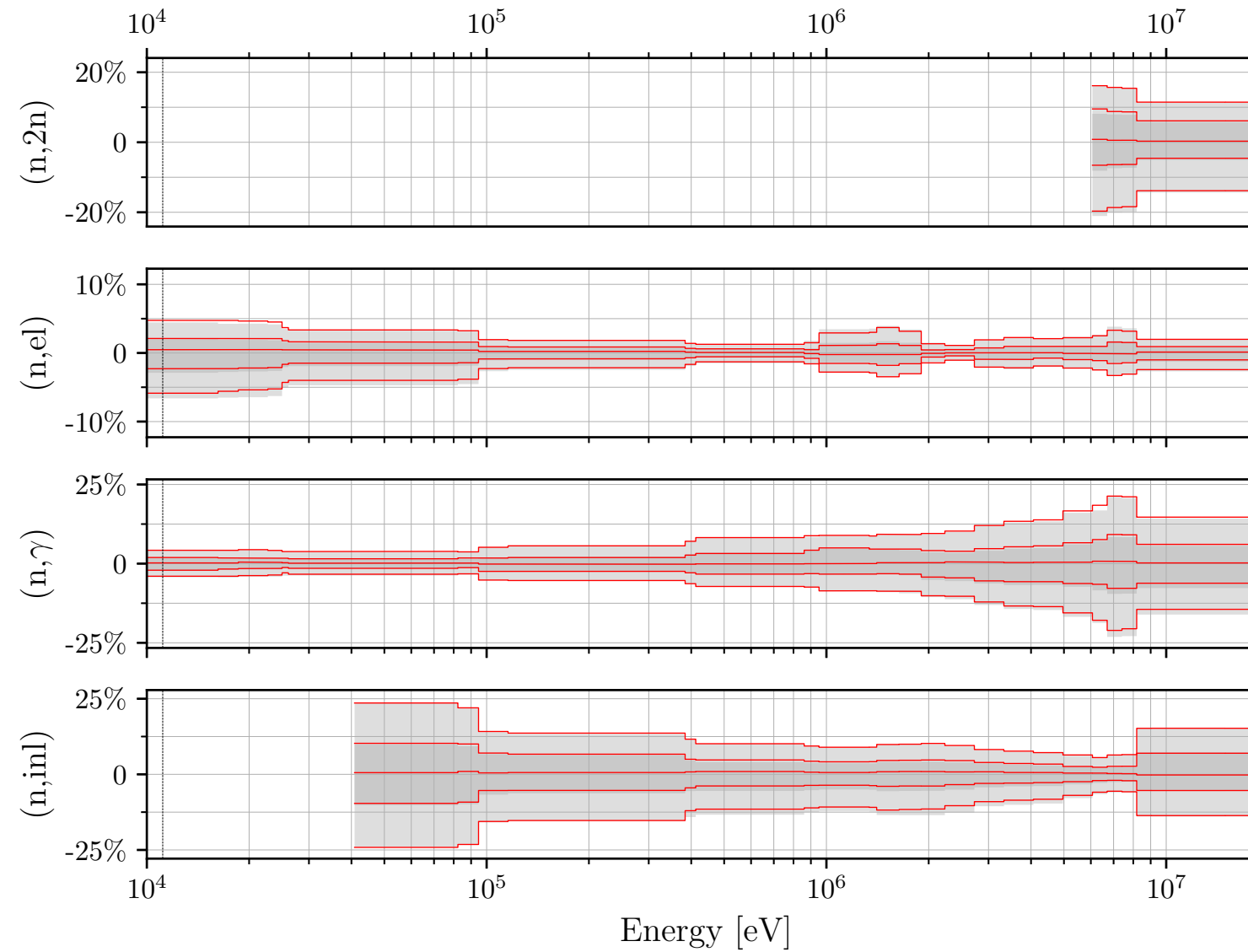


Credits : I. ICKE, Creative Commons BY-SA 4.0 <https://commons.wikimedia.org/wiki/File:Overfitting.svg>

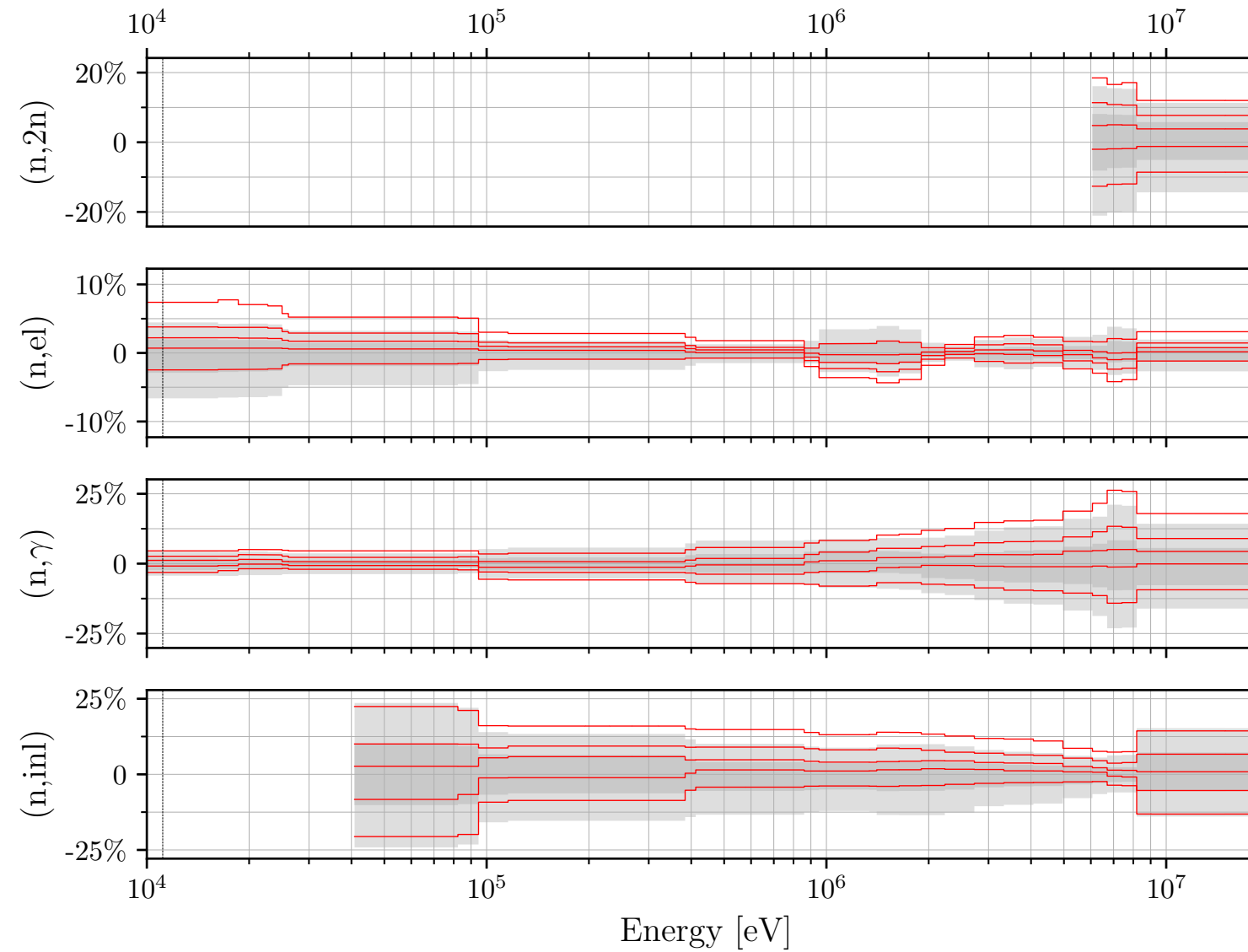
# Modifications (red) to five prior quantiles (grey) of hydrogen cross sections, with BMC



# Modifications (red) to five prior quantiles (grey) of uranium 238 cross sections, with BFMC

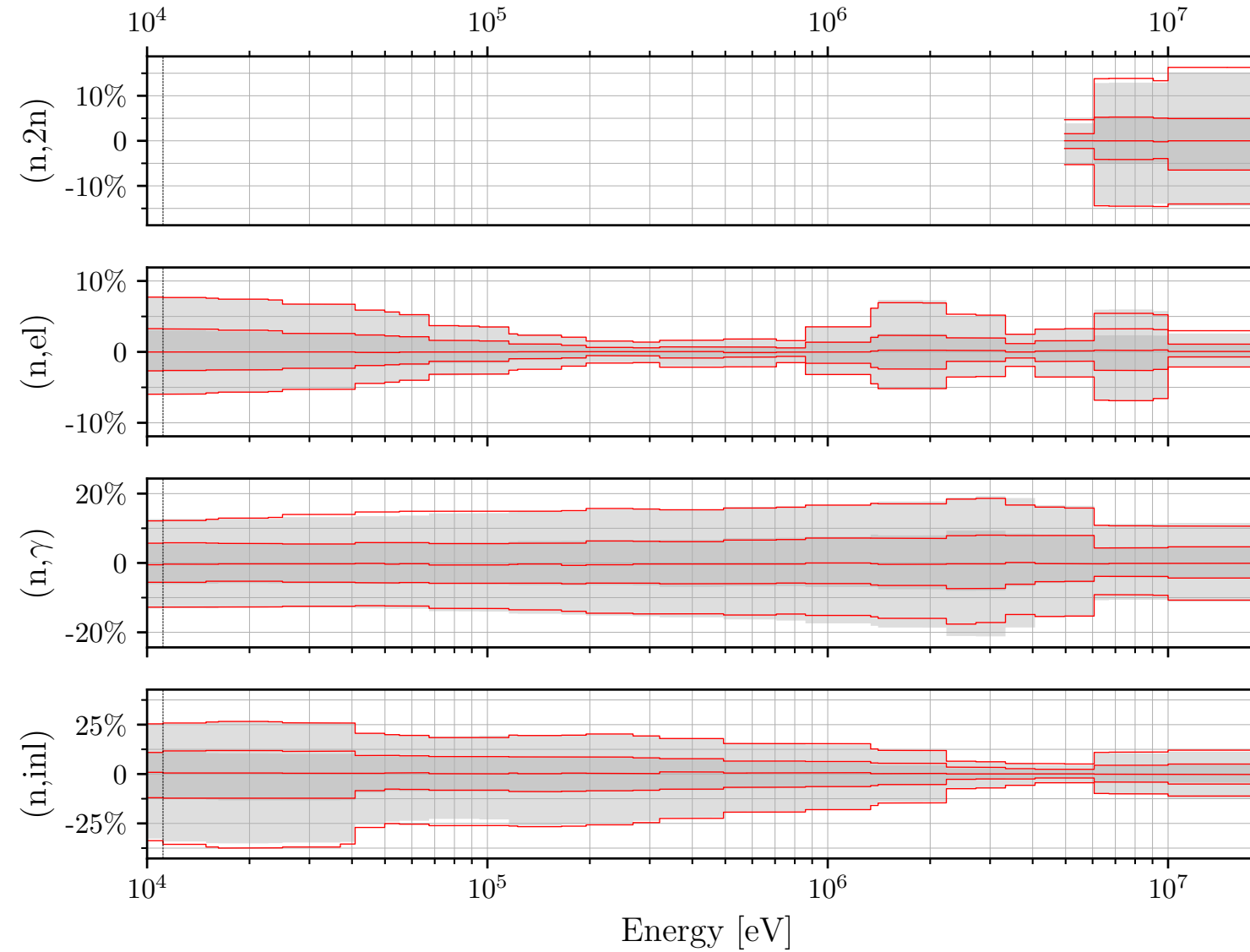


# Modifications (red) to five prior quantiles (grey) of uranium 238 cross sections, with BMC

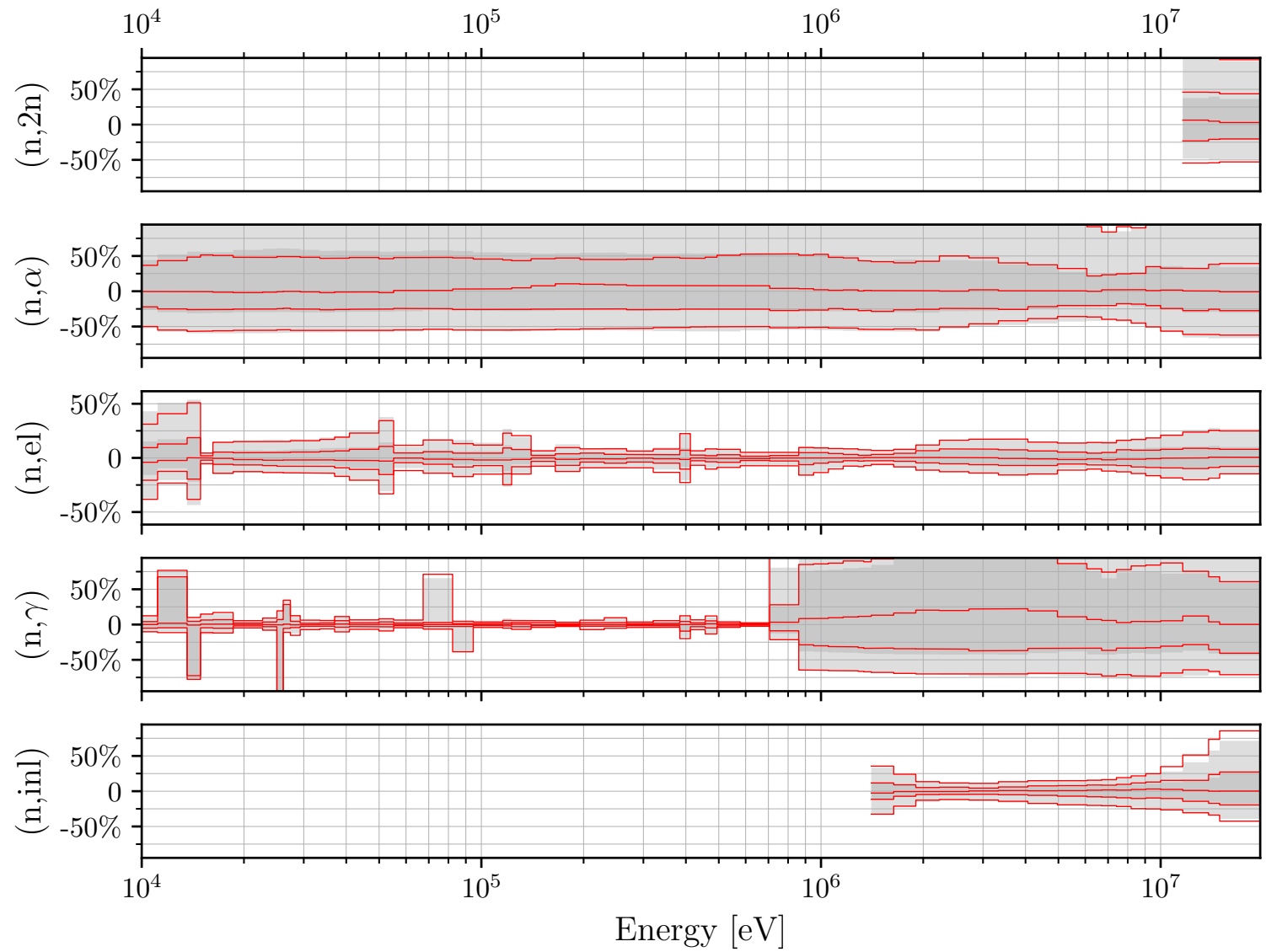




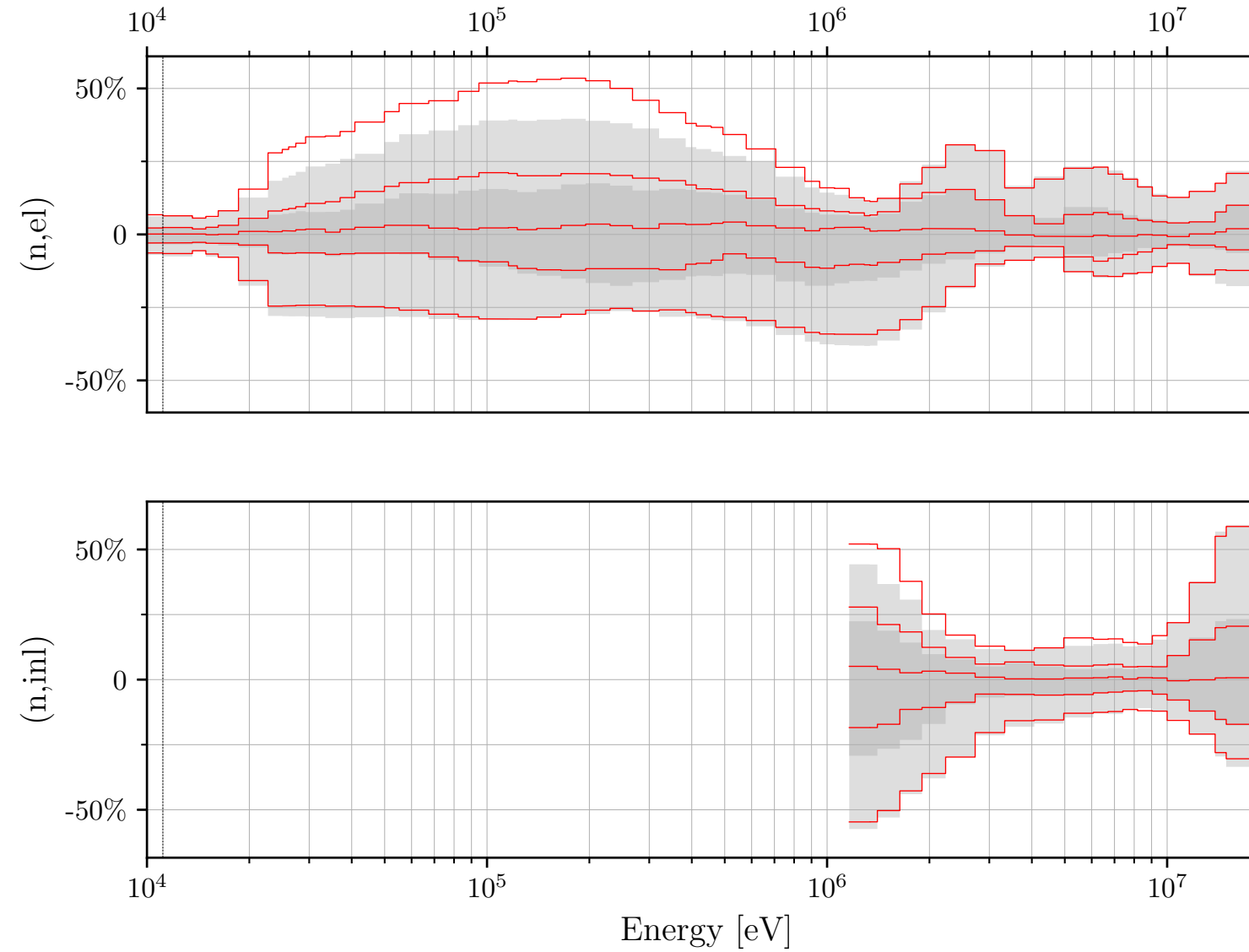
# Modifications (red) to five prior quantiles (grey) of uranium 235 cross sections, with BMC



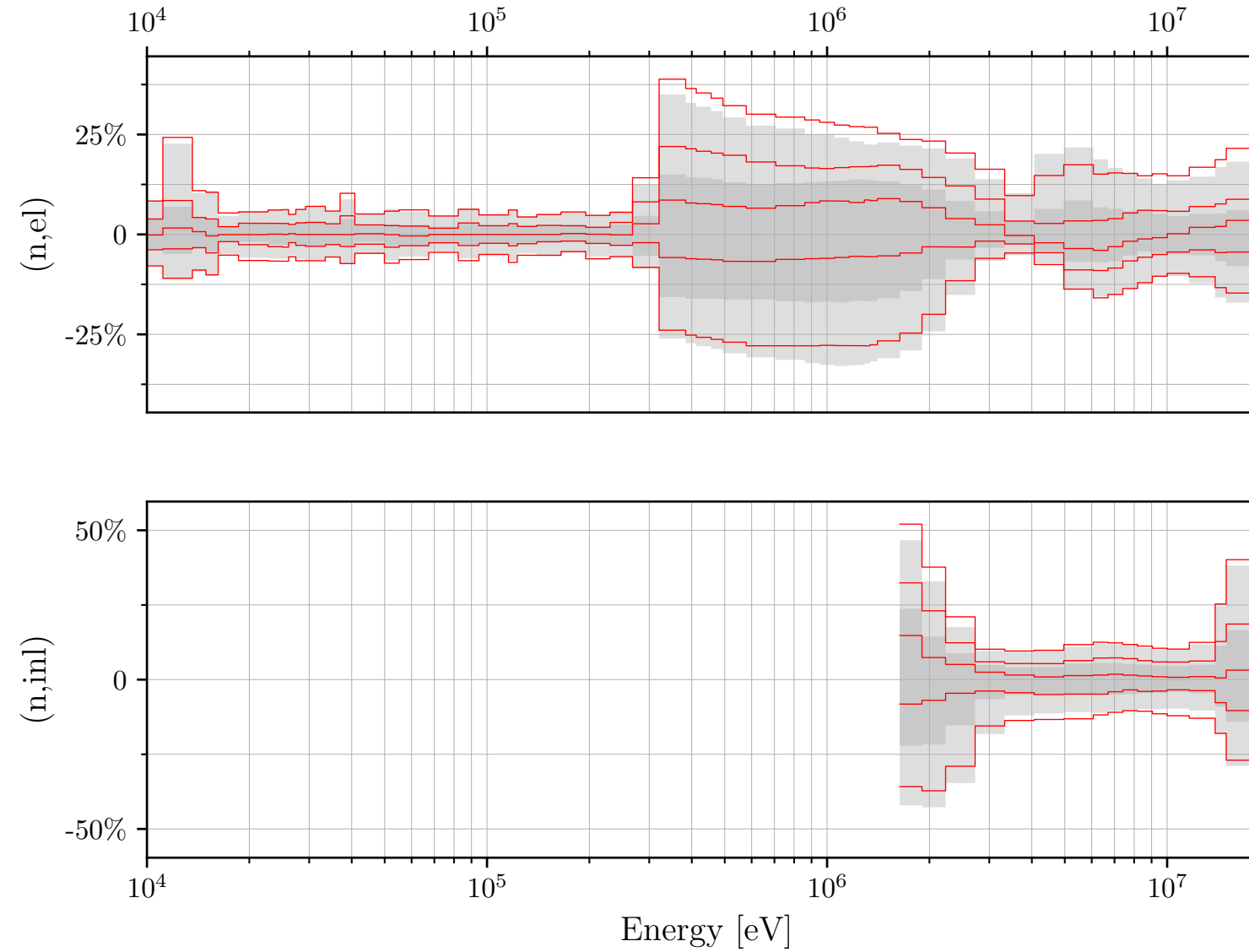
# Modifications (red) to five prior quantiles (grey) of nickel 58 cross sections, with BMC



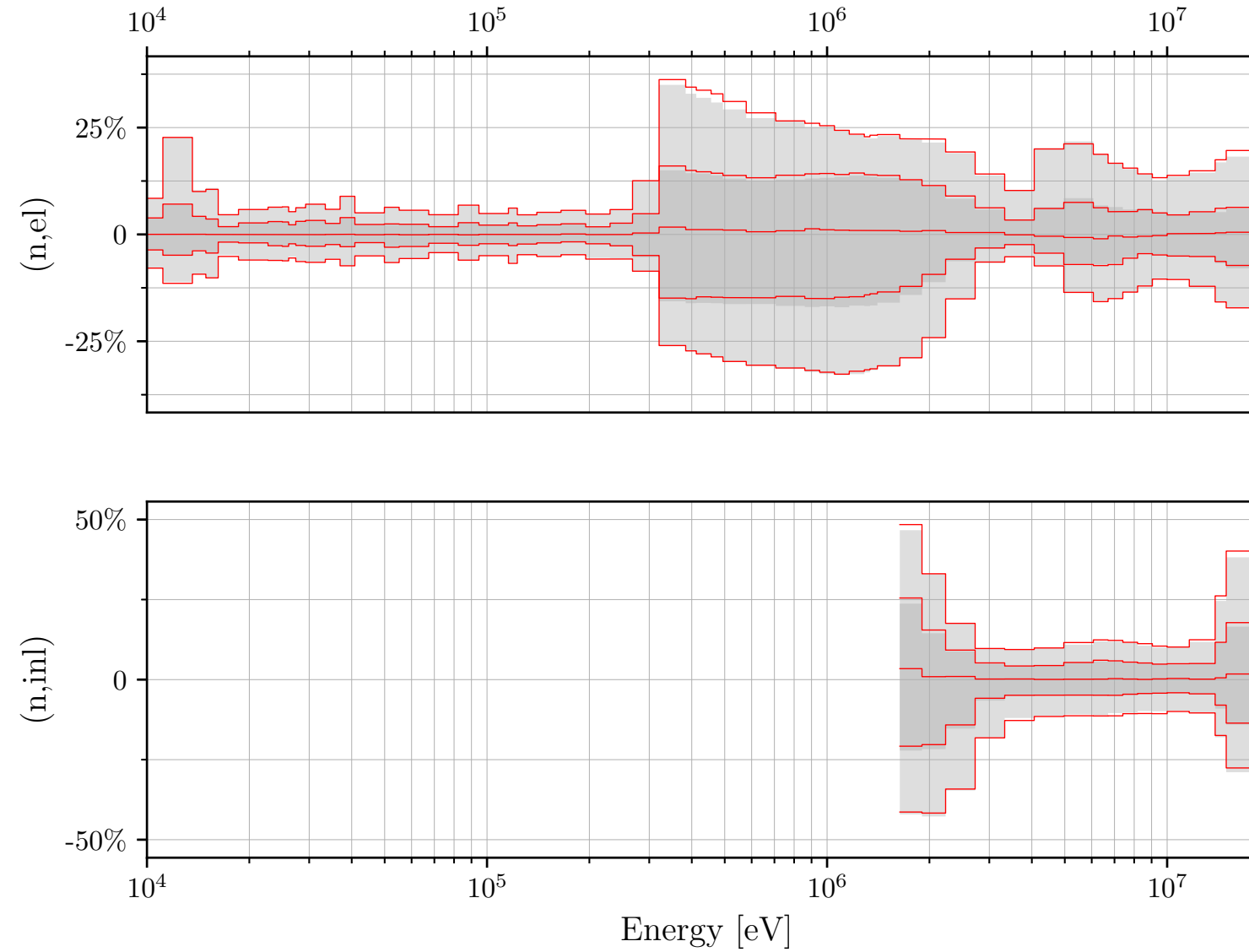
# Modifications (red) to five prior quantiles (grey) of zirconium 91 cross sections, with BMC



# Modifications (red) to five prior quantiles (grey) of zirconium 90 cross sections, with BMC



# Modifications (red) to five prior quantiles (grey) of zirconium 90 cross sections, with BFMC



# Pseudorandom number generator

- “The generation of random numbers is too important to be left to chance” (COVEYOU)
- Codes such as **SERPENT** (since version 2.1.0) and MCNP5
  - **Linear congruential generator**

$$X_{n+1} = (a \times X_n + c) \mod m \quad (11)$$

- Nuclear data sampling → **SANDY** → NumPy
  - Since NumPy version 1.17, **permuted congruential generator** (PGC64)
  - Before NumPy 1.17, Mersenne Twister
- Random **selection** of nuclear data
  - Here, relies on Python 3.6.9 → **Mersenne Twister**