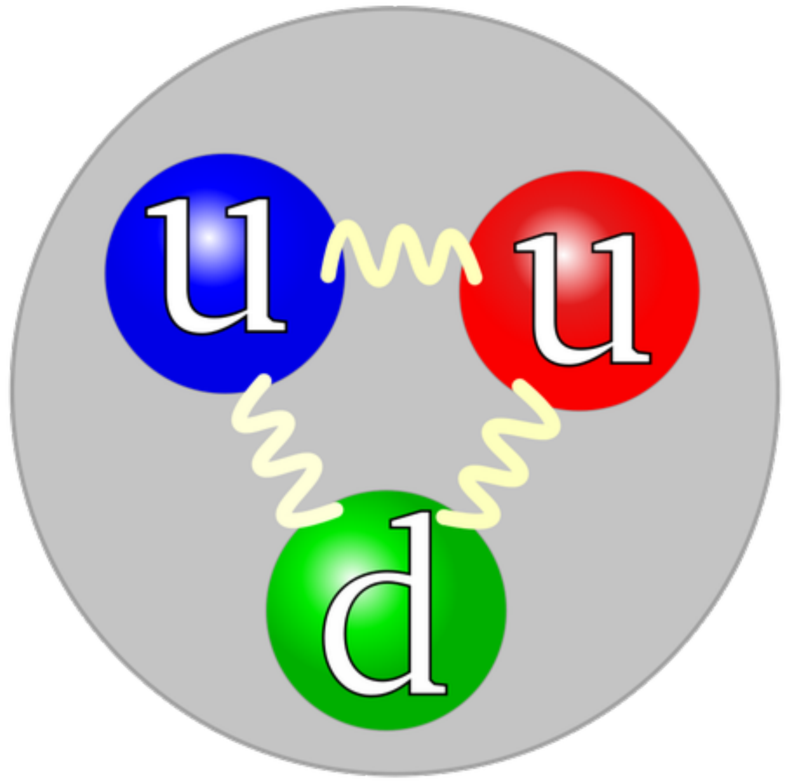
Evaluating Neural Network Uncertainty Estimation with Inconsistent Training Data

Author: Giovanni De Crescenzo

Based on work with: Andrea Barontini & Mark N. Costantini

Parton Distribution Functions: PDFs



- QCD: strong interaction theory
- Fundamental fields are quarks and gluons (color confinement)
- Proton: bound state $\mapsto$ PDFs dependent observables
- PDFs: $\{f_i(x, Q^2)\}_{i} = 1,\ldots,n_{\text{fl}}$

Convolutional map

$$\sigma_{had} = \sum_{i,j} f_i \otimes \hat{\sigma}_{ij} \otimes f_j$$

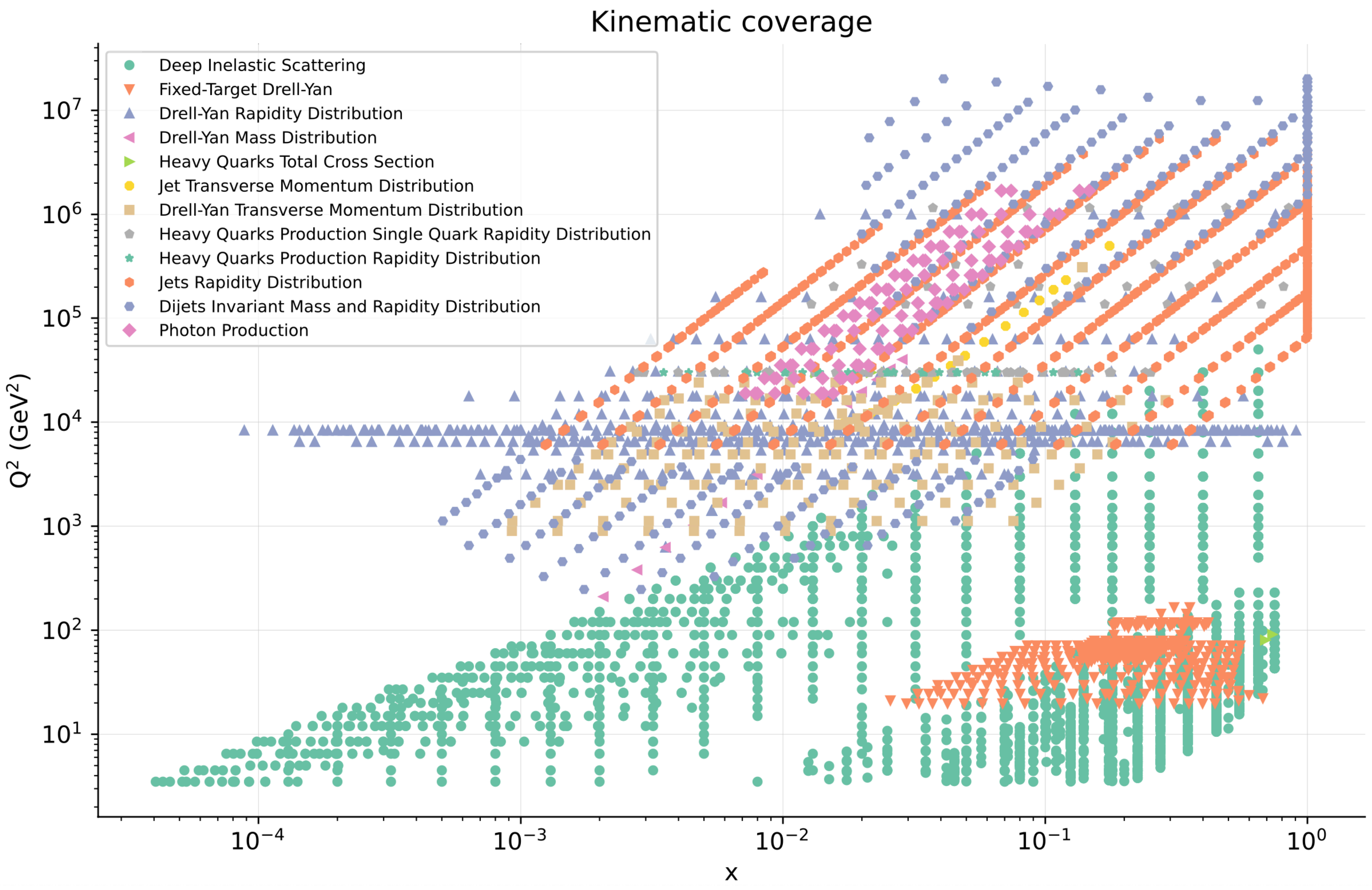
How to compute them?

NN parameterized regression + MC for uncertainty

Starting dataset and technical details

Dataset and PDFs features

- PDFs: vector function $(x, Q^2) \mapsto \mathbb{R}^{N_{fl}}$
- Starting dataset: 4000 datapoints (several “labels”: process type, kinematic region…)



- Input:
  - y_0 : *exp central values*
  - C_{exp} : *exp covariance matrix*
  - $y_0 = f + \eta$
  - $\eta \sim \mathcal{N}(0, C_{exp})$

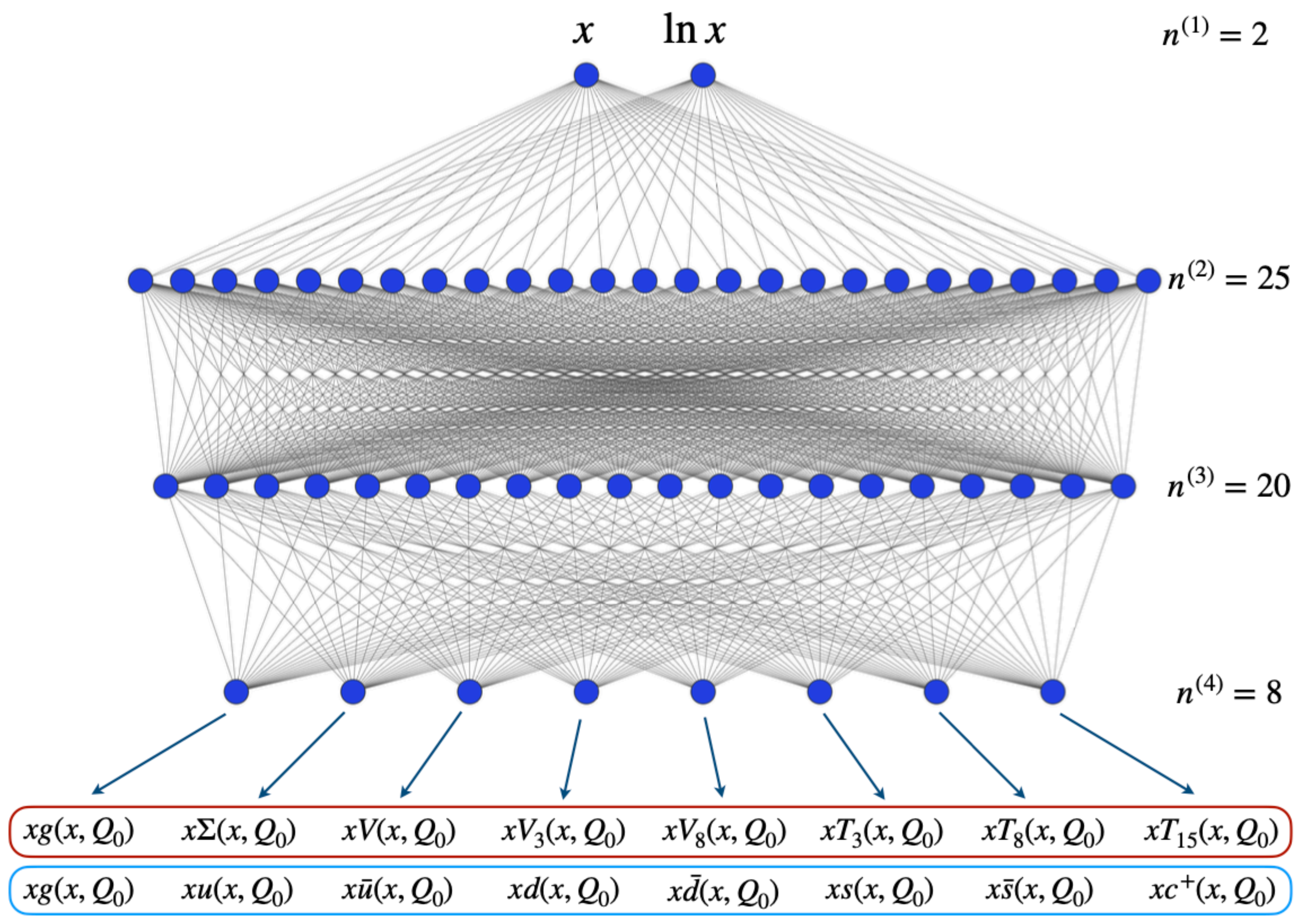
NN parameterization of PDFs and fitting

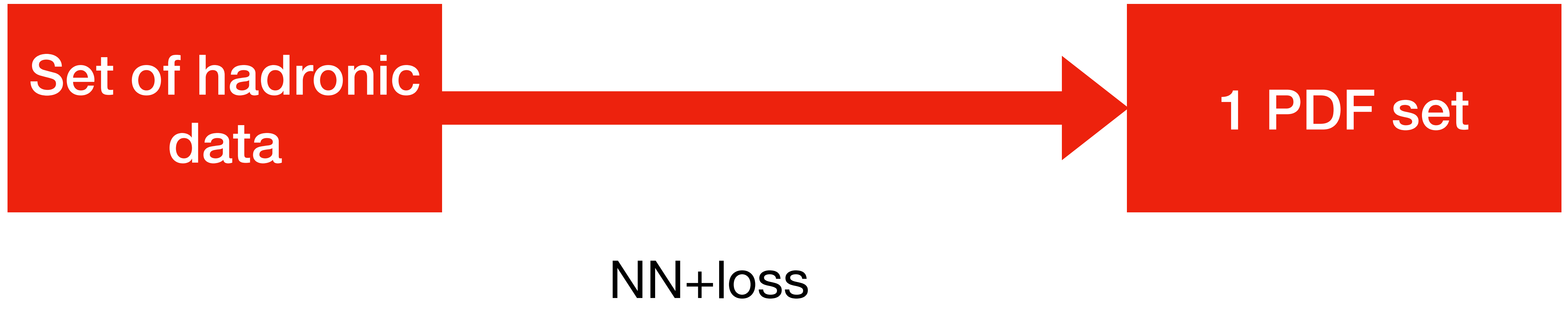
Single PDF fit

- PDFs parameterized at Q_0 energy scale (DGLAP $Q_0 \mapsto Q$)
- Training: loss minimization

$$\mathscr{L} = \chi^2 = (T-D)^T C_{\text{exp}}^{-1} (T-D)$$

- T: NN prediction
- D: data





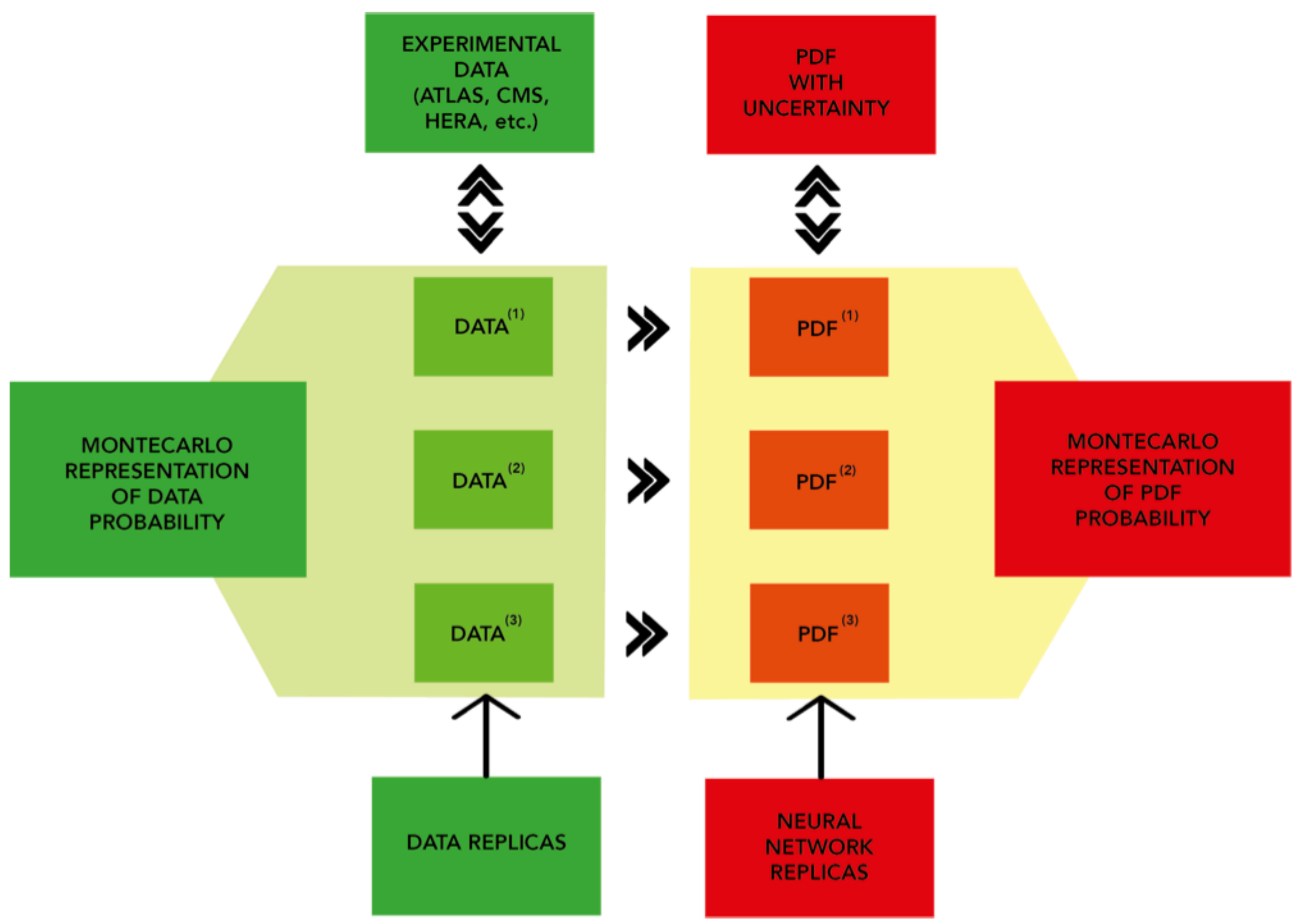
Uncertainty estimation

- PDF *uncertainty estimation*
- Sample of MC replicas of input data:
  $y^r := y_0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, C_{\text{exp}})$
- This yields an *ensemble* of replicas, which gives information on PDF uncertainty

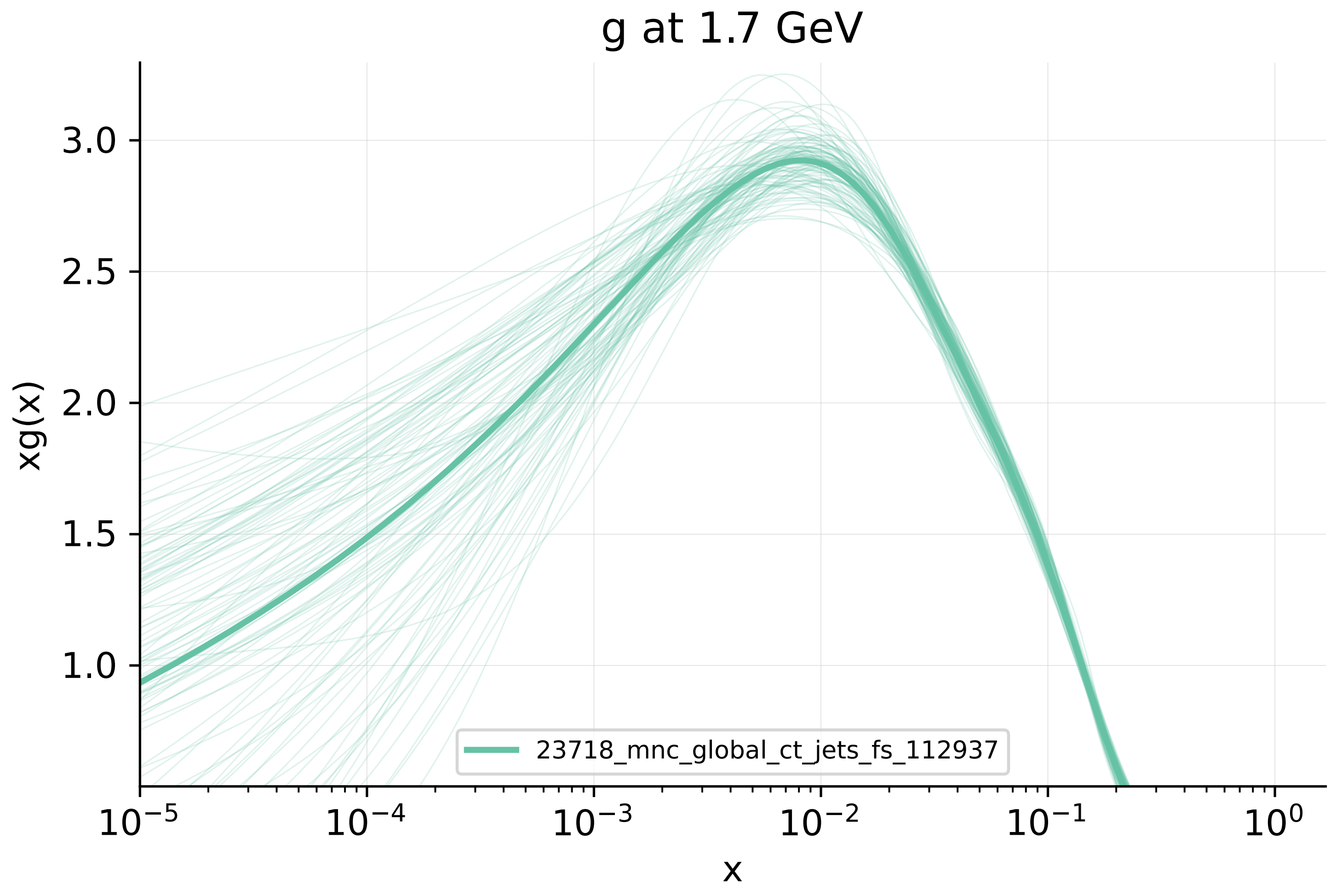
[arXiv:2111.05787]

C_{exp} drives both replica generation and loss.

What happens if it is flawed?



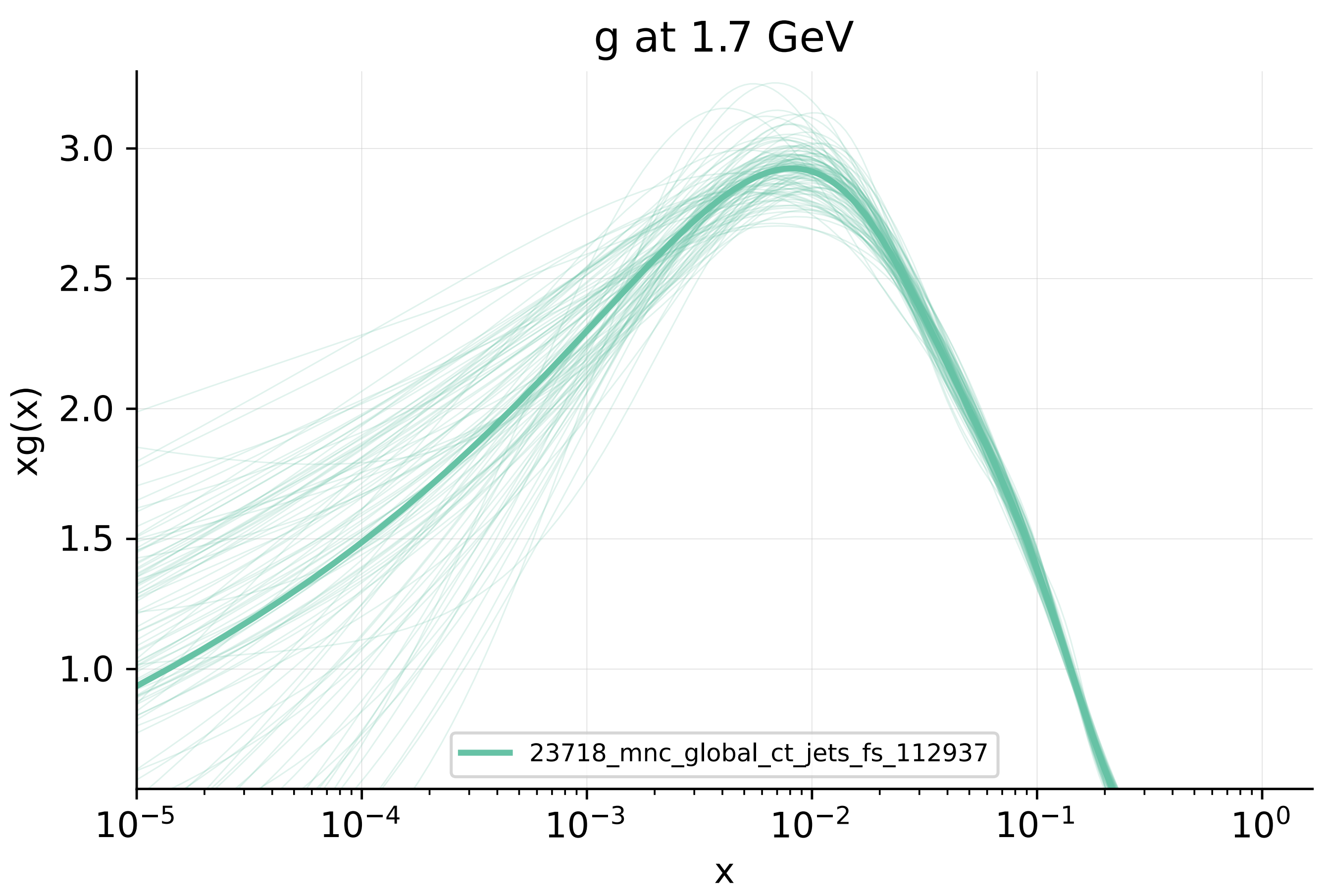
MC replica approach

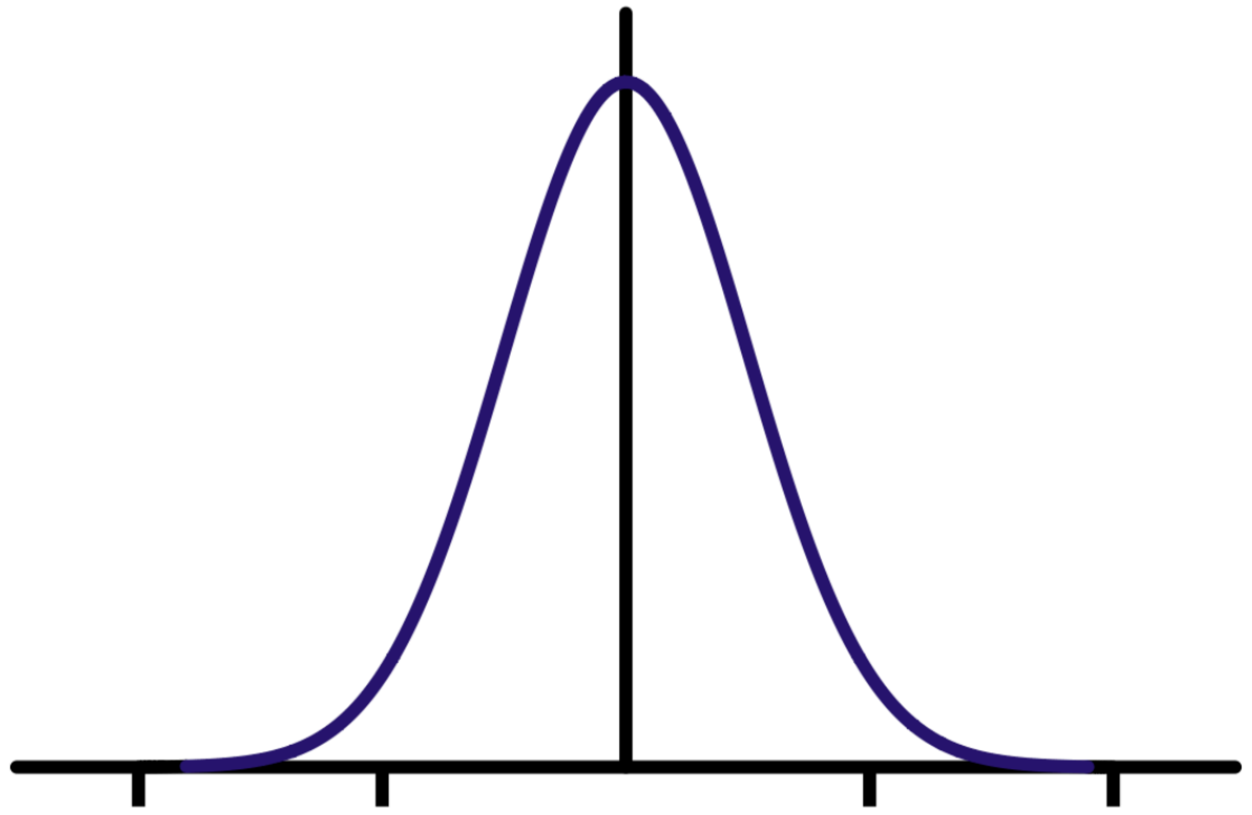


PDF output replicas

Faithfulness in uncertainty?

How do we check uncertainty faithfulness?





Introduce

- $\langle O_i \rangle :=$ central prediction;
- $\sigma(O_i) :=$ std deviation of prediction sample
- $\langle O_i \rangle \overset{!}{\sim} f_i + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma(O_i))$

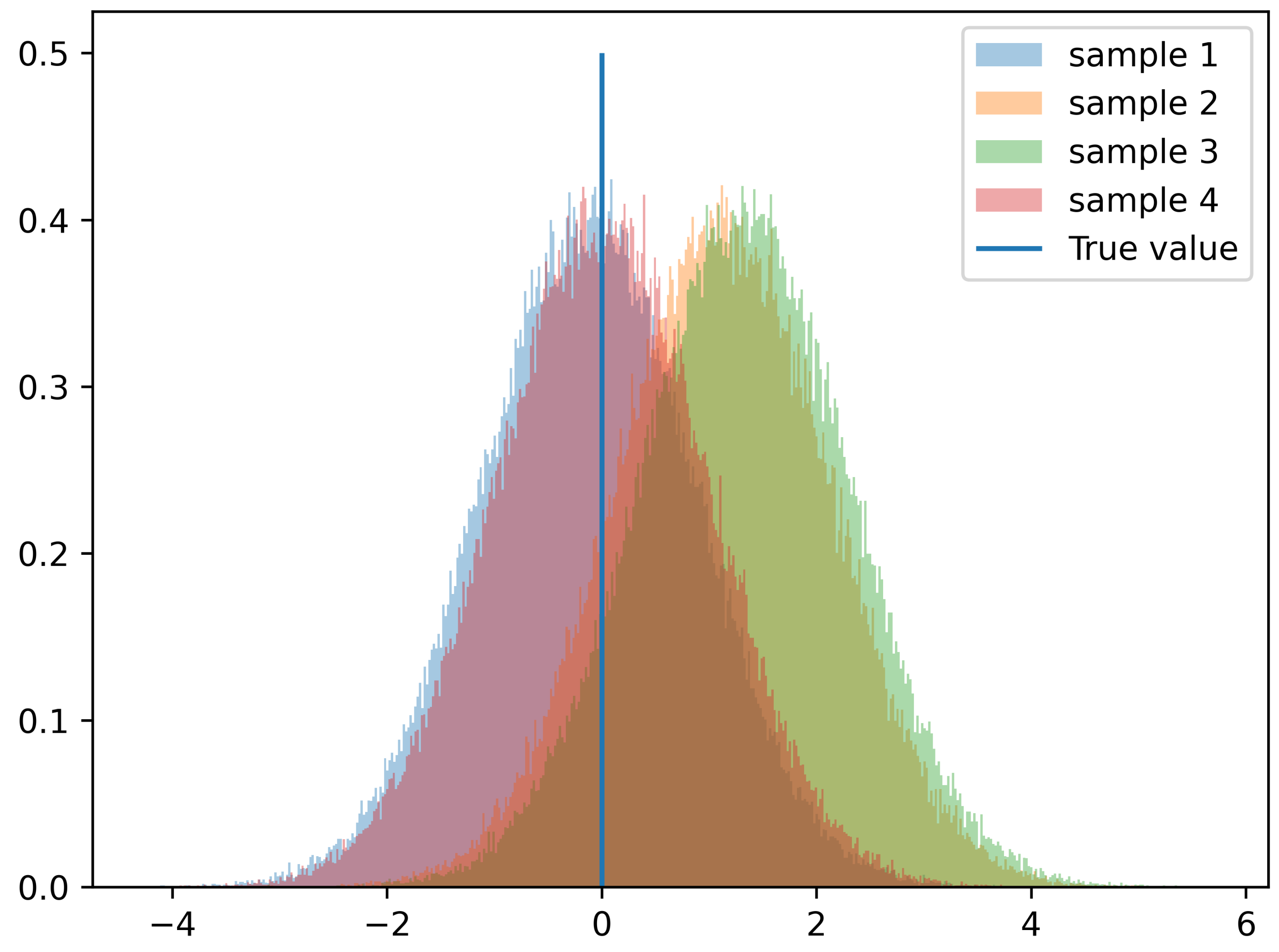
PDF sample → Prediction sample O_i

Forward map

Introduce method to check statement in a frequentist way

Closure testing: validation of the methodology

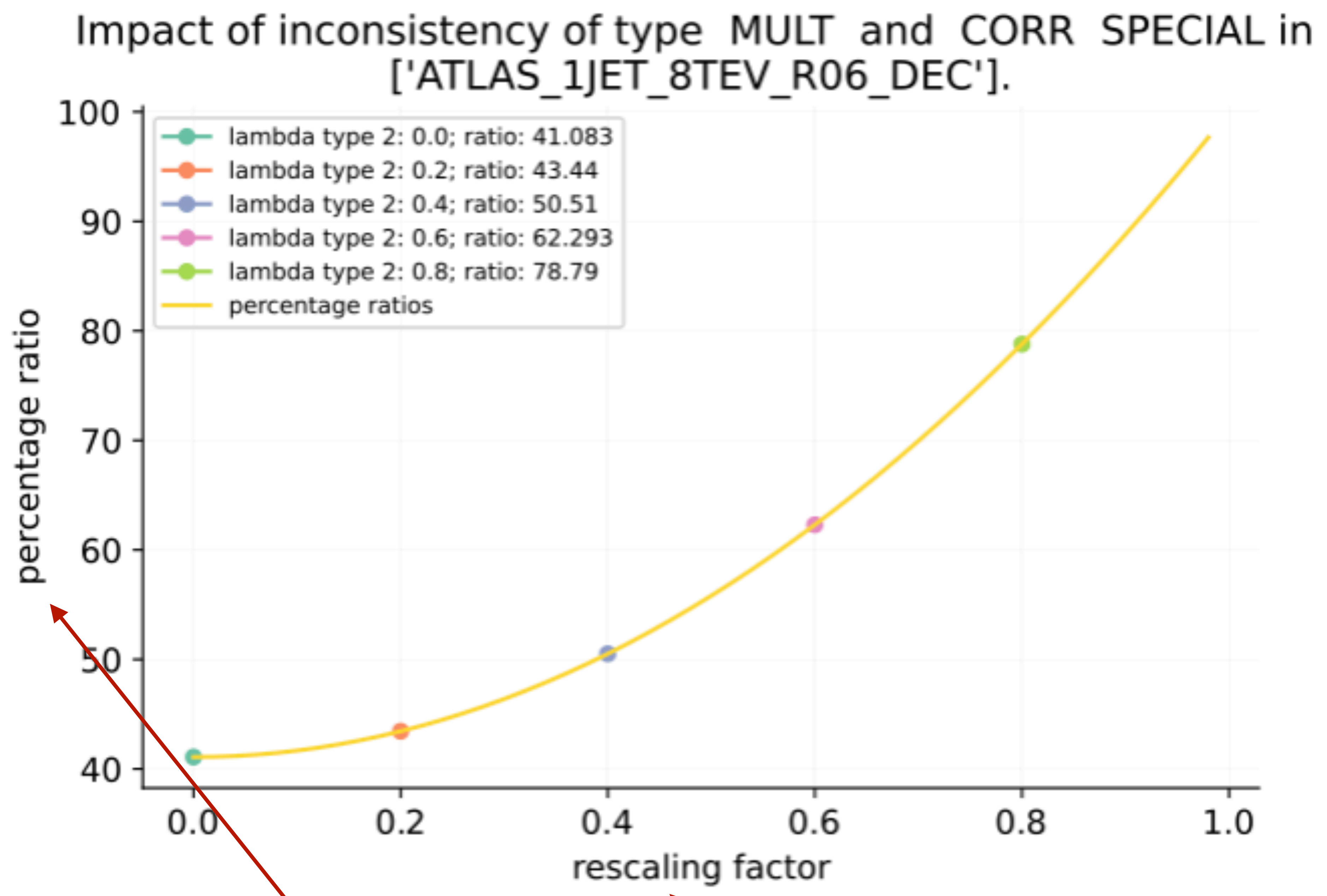
- Real situation: f true value is not known
- *Choose* f underlying truth (choose PDF set $\mapsto$ "true" dataset)
- Generate "runs of the Universe": $y_0^l \sim \mathcal{N}(f, C_{exp}) \mapsto y^{l,r} := y_0^l + \epsilon^{l,r}$ where $\epsilon^{l,r} \sim \mathcal{N}(0, C_{exp})$
- Several *independent* PDF samples $\mapsto$ several independent *predictions* samples



- $\langle O_i \rangle \mapsto \langle O_i \rangle^l$ for $l = 1, \ldots, n_{fits}$
- $\langle \langle O_i \rangle^l \rangle - f \overset{!}{=} 0$
- $\sigma\left(\frac{\langle O_i \rangle^l - f}{\sigma(O_i)} \right) \overset{!}{=} 1$

Inconsistent closure testing

- Inconsistent Closure testing: *simulate* C_{exp} is flawed
- "Real error": $\epsilon_1 \sim \mathcal{N}(0,C_1)$
- "Experimental error": $\epsilon_2 \sim \mathcal{N}(0,C_2)$
- $C_2 \neq C_1$; tune the difference with a "scale" parameter λ and exp label (JETS, DIS, DY…)
- $y_0^l := f + \epsilon_1^l \mapsto y^{l,r} := y_0^l + \epsilon_2^{l,r}$



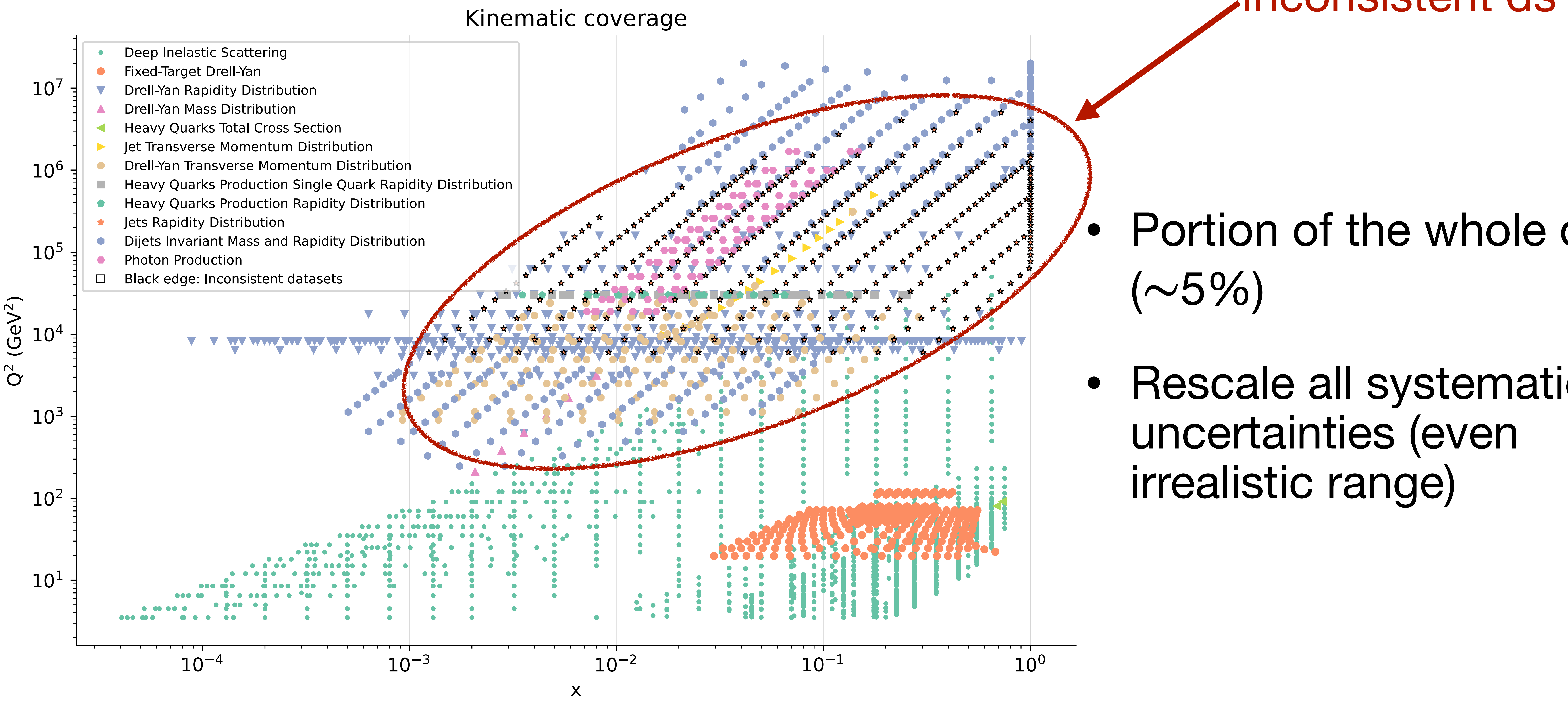
λ

$\frac{tr(C_2)}{tr(C_1)}$

- Check trend of std deviation: $\left[\sigma\left(\frac{\langle O_i\rangle^l - f}{\sigma(O_i)}\right)\right](\lambda)$

Inconsistent dataset

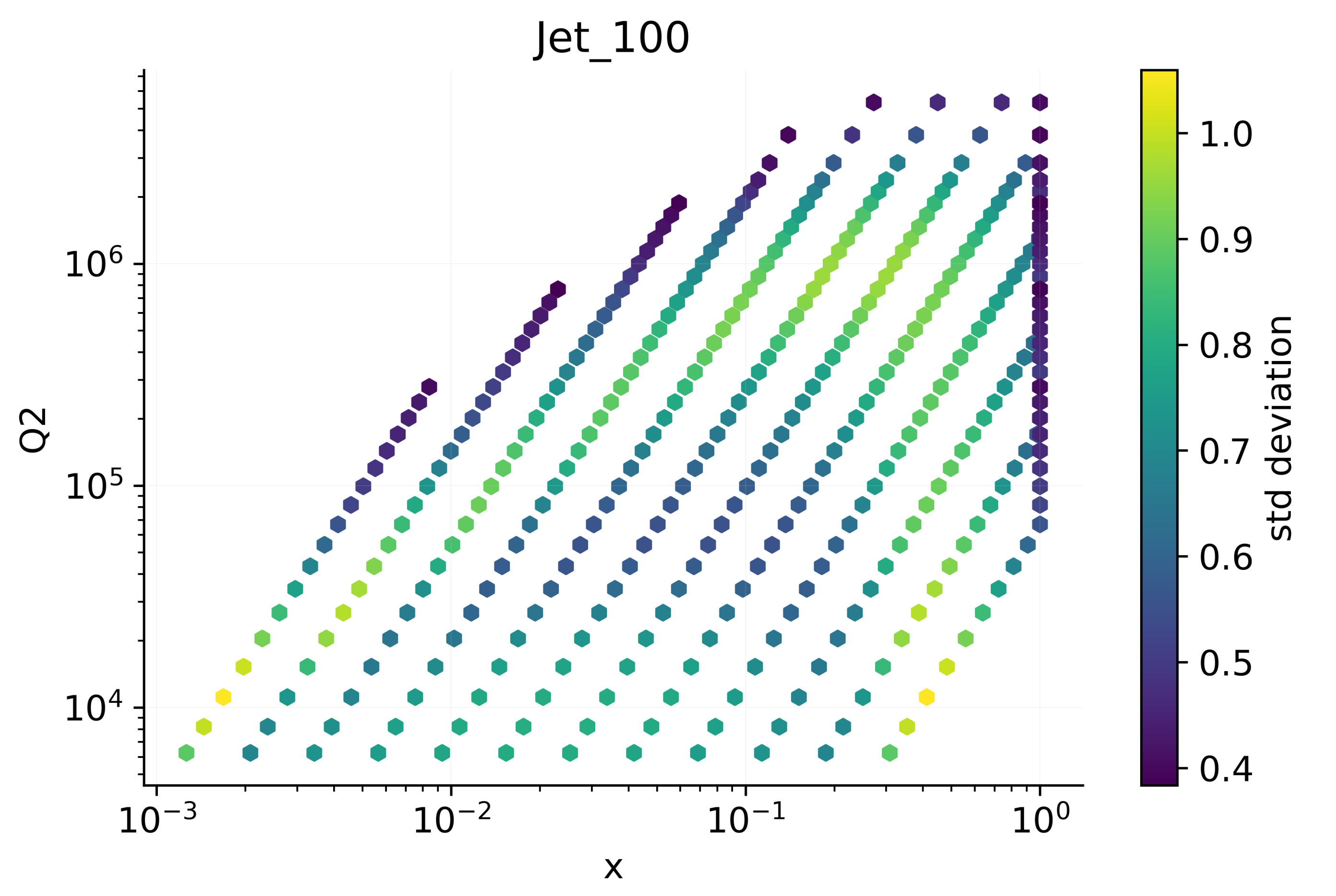
ATLAS SINGLE JET 8TEV



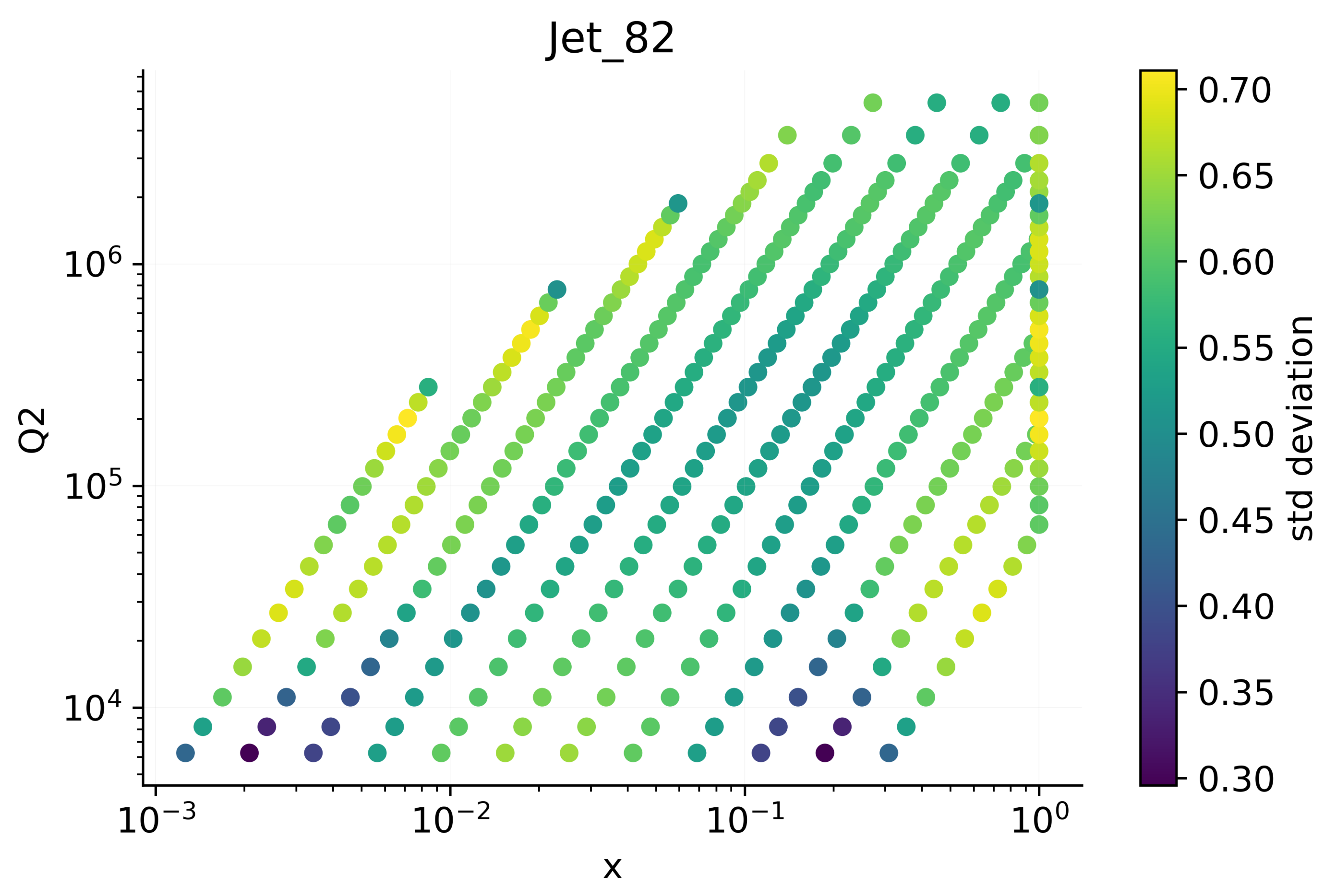
- Portion of the whole dataset (~5%)
- Rescale all systematic uncertainties (even irrealistic range)

Results: output trend of std deviation

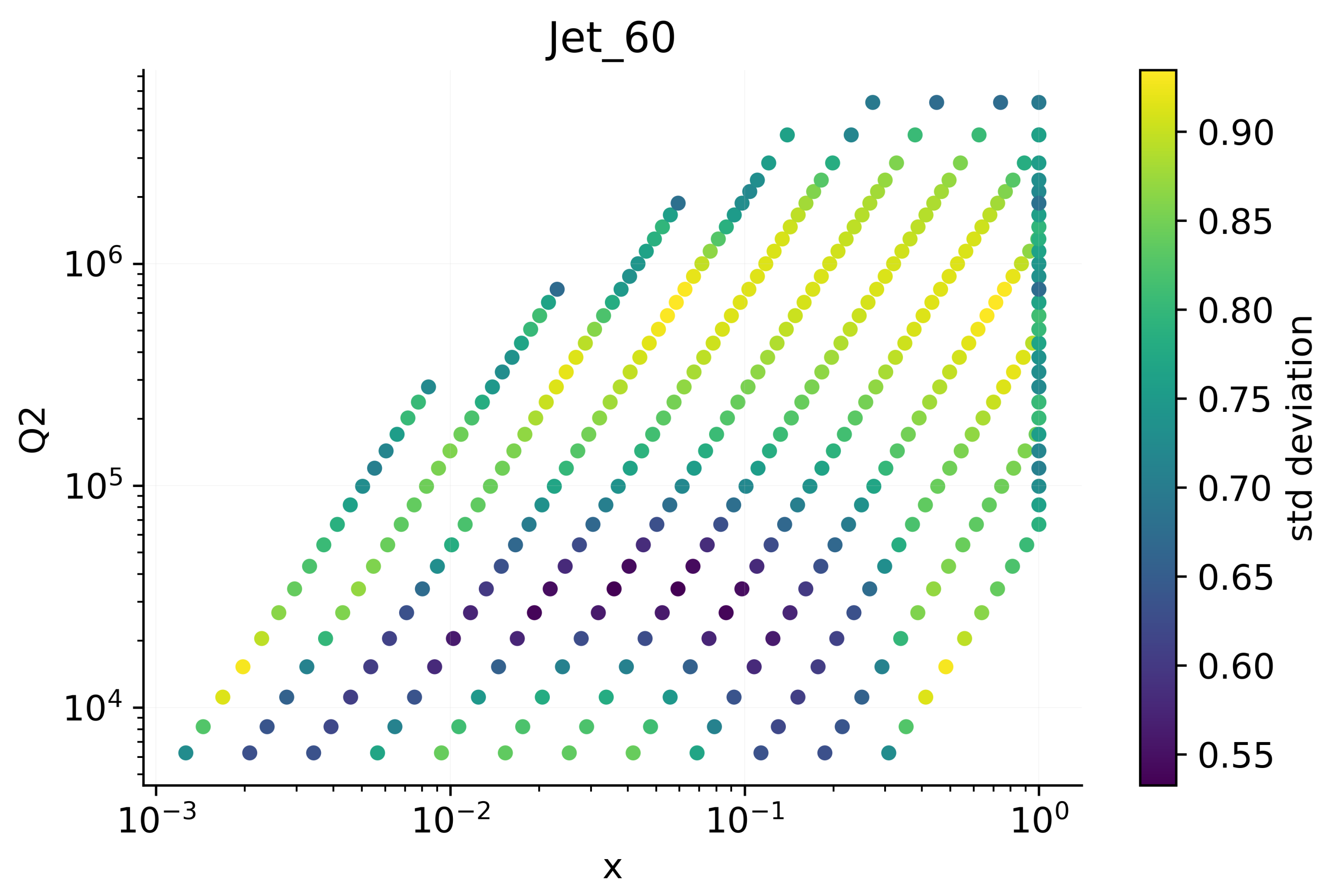
TESTING DS: CMS SINGLE JET 8TEV



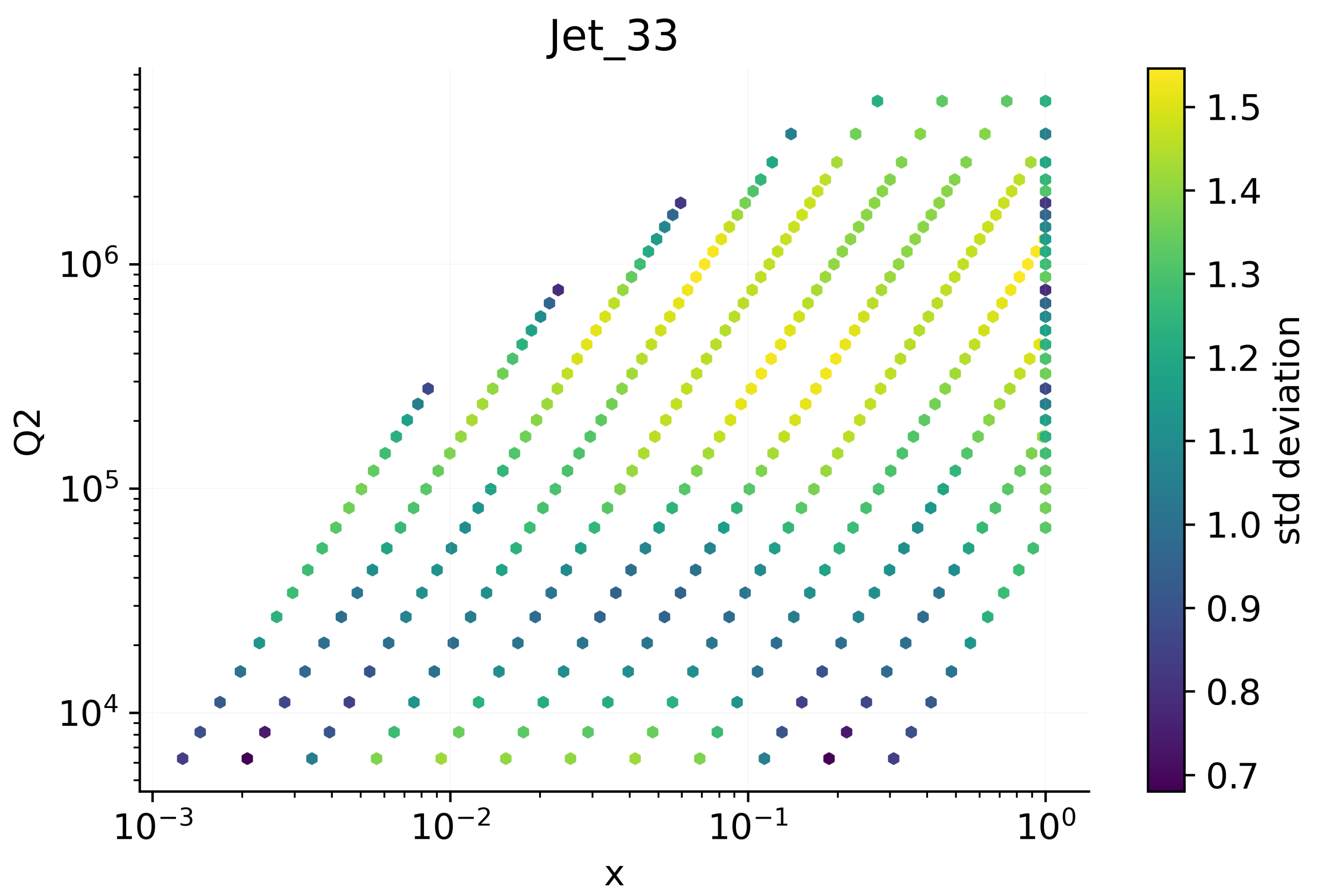
$\lambda = 1$: consistent



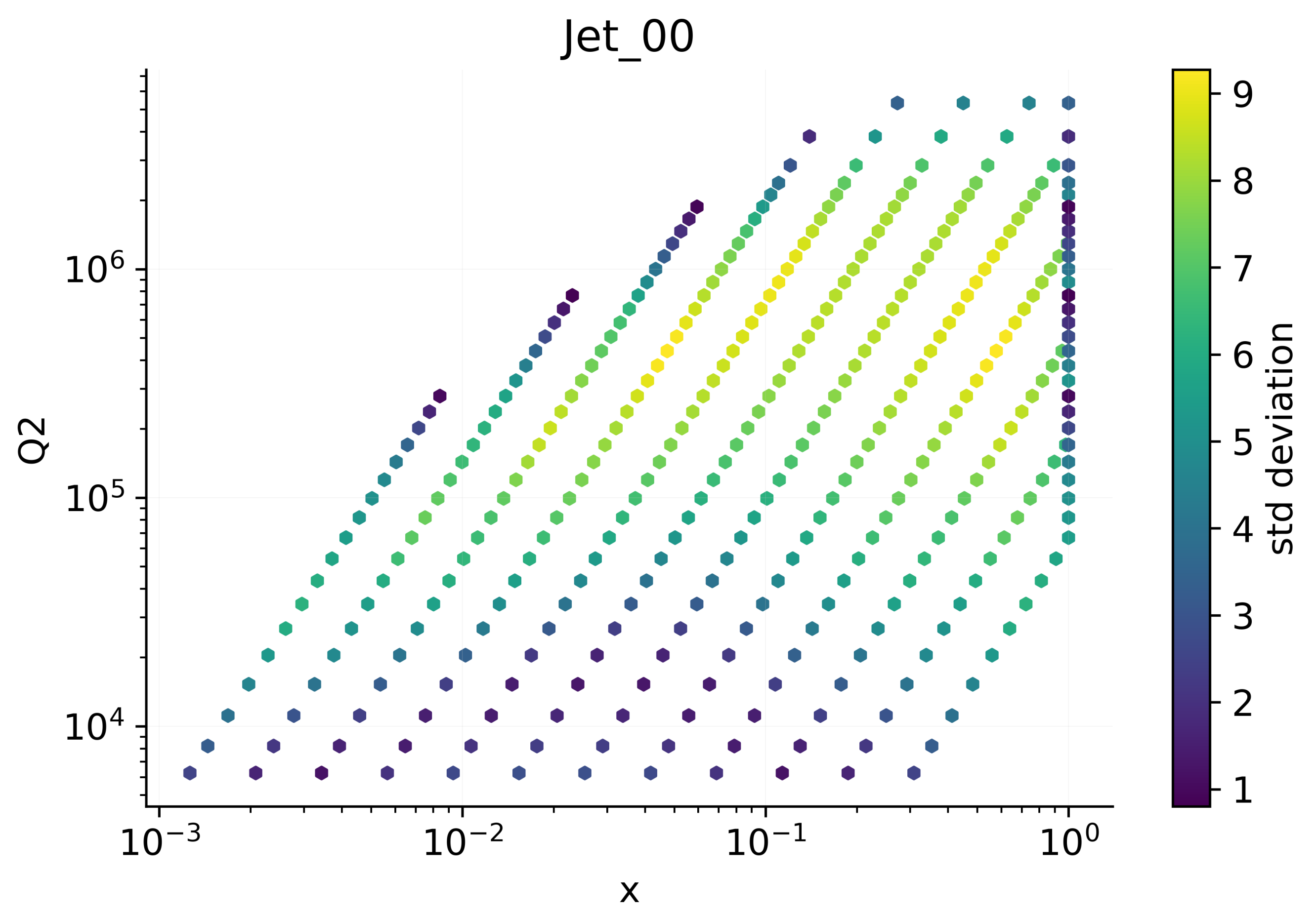
$\lambda = 0.82$



$\lambda = 0.60$



$\lambda = 0.33$



$\lambda = 0$: most inconsistent

- Output is evaluated on JET observables
- Input inconsistency also JET dataset
- Just to give an idea:
  - $\sigma > 1$: uncertainty underestimation
  - $\sigma < 1$: uncertainty overestimation

Overlook and summary

- Closure test setting can be used in any situation in which we have to deal with reliable uncertainty estimation
- “Hard” to detect inconsistencies
- “One-to-one” correspondence between input inconsistency and output performance

Next steps

- Study more cases
- Keep correlations into account in prediction space

Overlook and summary

- Closure test setting can be used in any situation in which we have to deal with reliable uncertainty estimation

- “Hard” to detect inconsistencies

- “One-to-one” correspondence between input inconsistency and output performance

Next steps

Thanks for your attention!

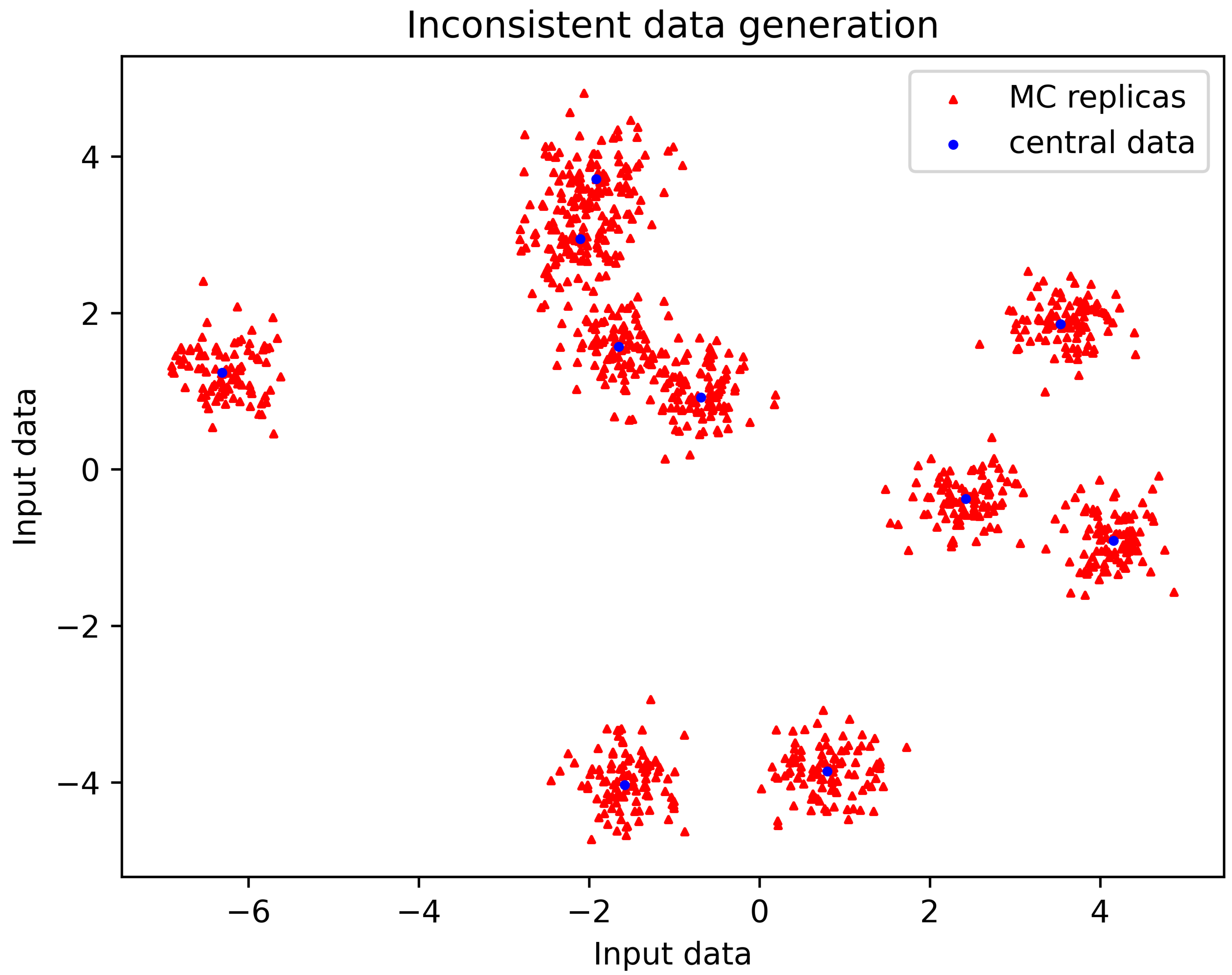
- Study more cases

- Keep correlations into account in prediction space

Backup slides
Definition of inconsistency

- C_{exp} is defined summing together different uncertainty sources
- $$cov_{ij} := \left(\sum_{k=1}^{N_{sys}} \sigma_{i,k}\sigma_{j,k} + F_i F_j \sigma_N^2 \right) + \delta_{i,j}\sigma_{i,t}^2$$
  - F_* experimental central value
  - $\sigma_{*,k}$ systematic uncertainties
  - σ_N overall normalization uncertainty
  - $\sigma_{i,t}$ uncorrelated uncertainty

- Inconsistent cov' defined as follows:
- $$cov'_{ij} := \left(\sum_{k=1}^{N_{sys}} \lambda_k\sigma_{i,k}\lambda_k\sigma_{j,k} + F_i F_j \sigma_N^2 \right) + \delta_{i,j}\sigma_{i,t}^2$$
  - Where $\lambda_k < 1$ if we affect the k-th uncertainty



Backup slides
Output correlations

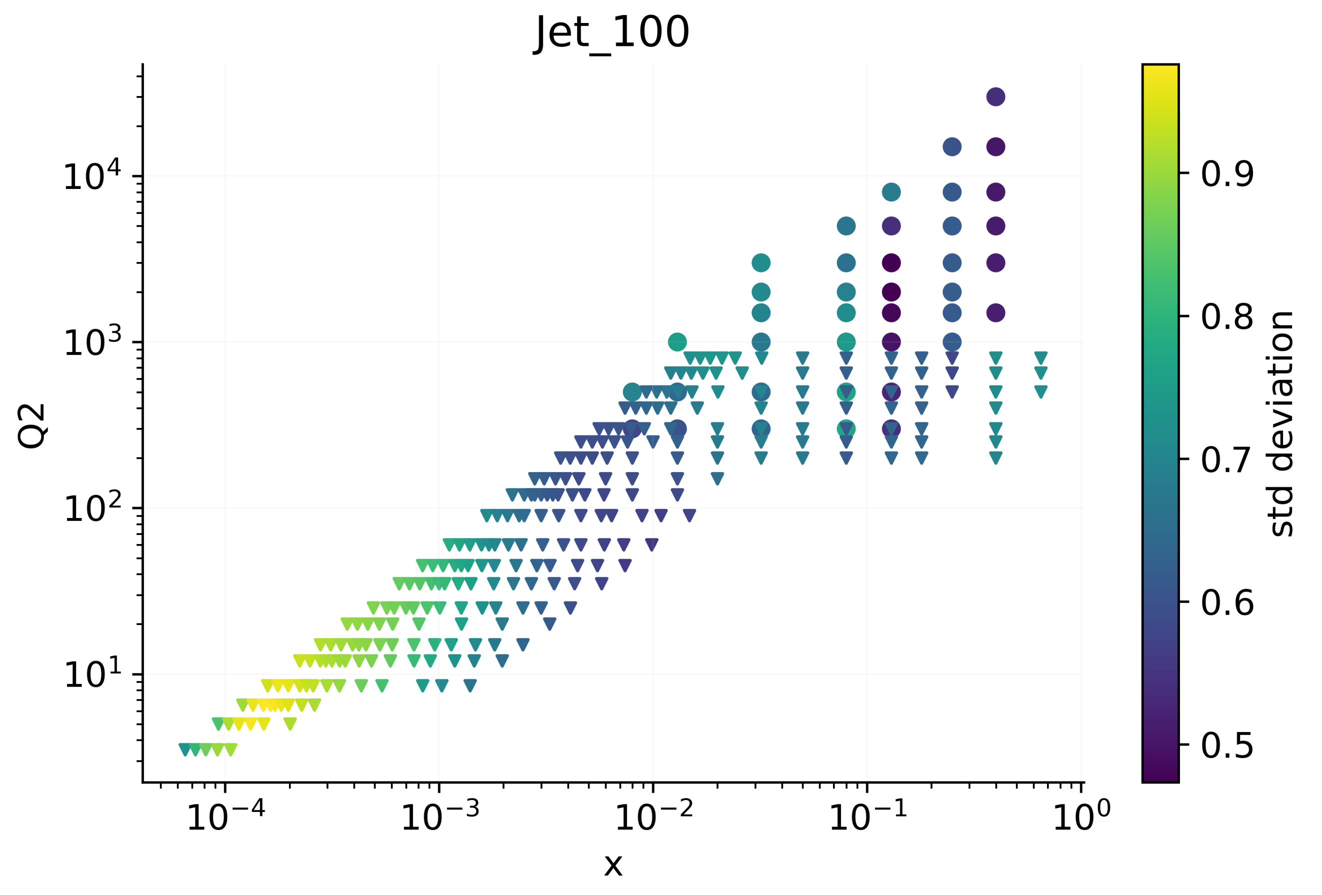
- What said up until now does not take into account correlations
- Correlations arise from the forward map itself since PDFs points are correlated
- Possible solution: estimate C_{rep} and use as figure of merit:

$$\chi^2(C_{rep}) := \Delta^T C_{rep}^{-1} \Delta$$

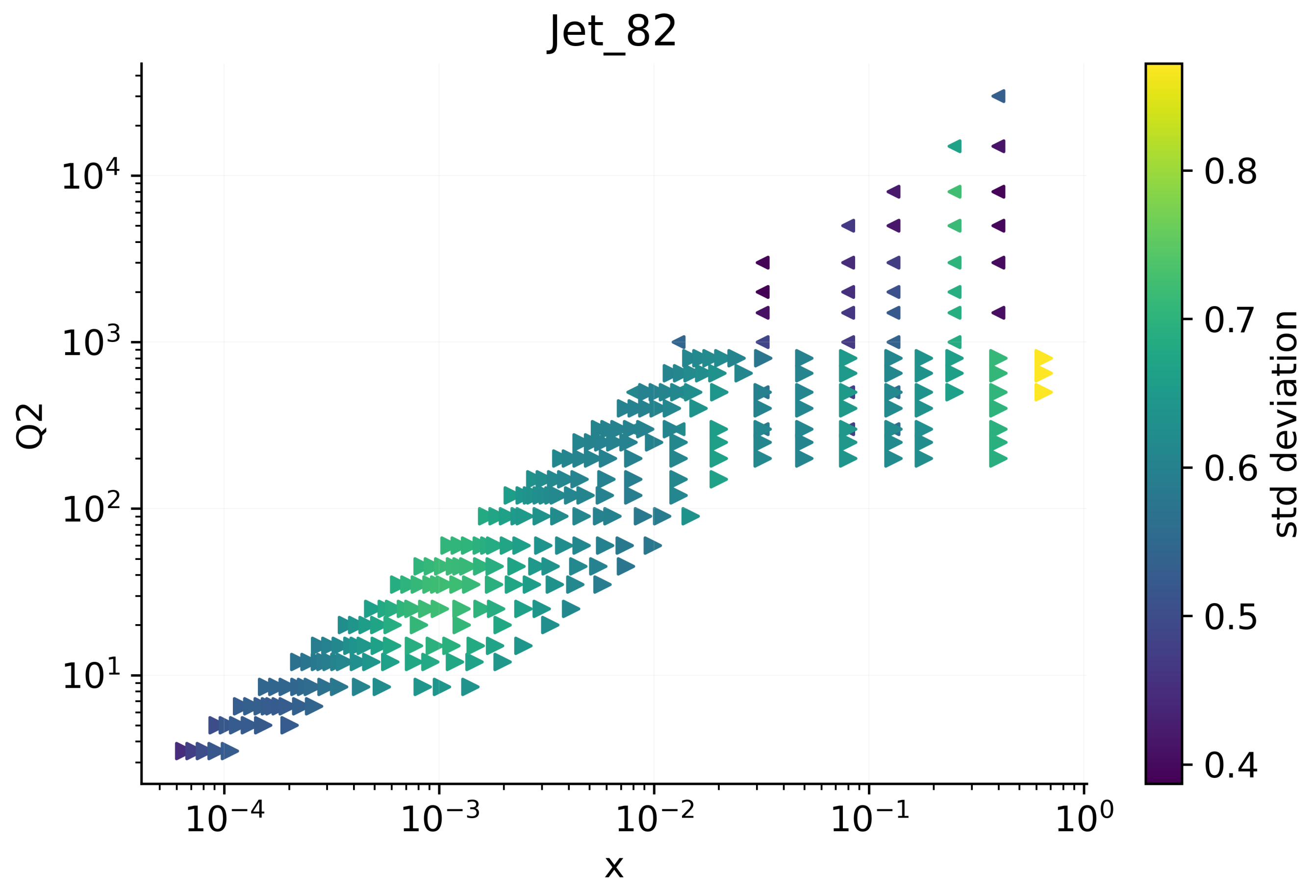
- C_{rep} ill-defined matrix! Regularization procedures like PCA need to keep into account tension between information loss/well-definition of C_{rep}

Backup results

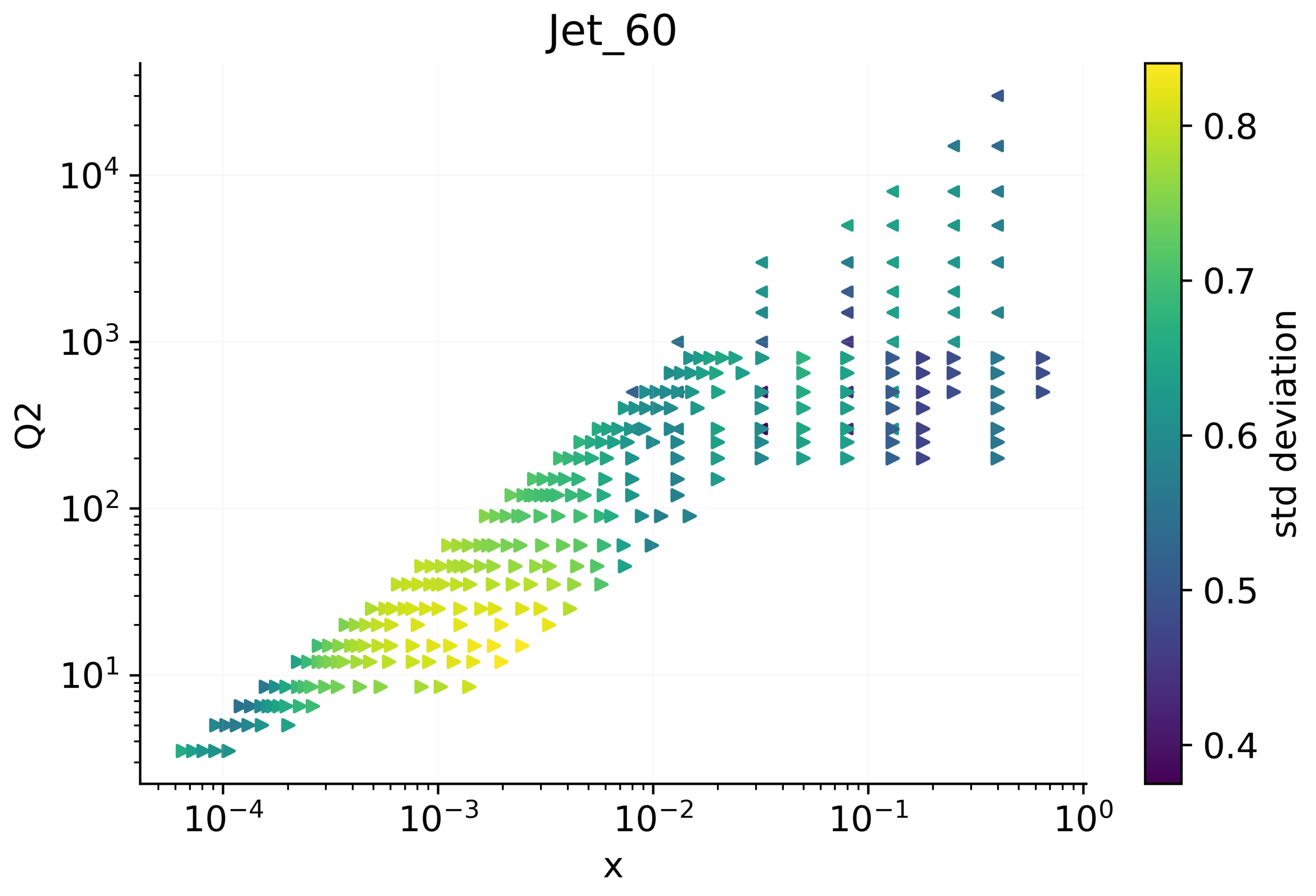
TESTING DS: DIS OBS (HERACC HERANC)



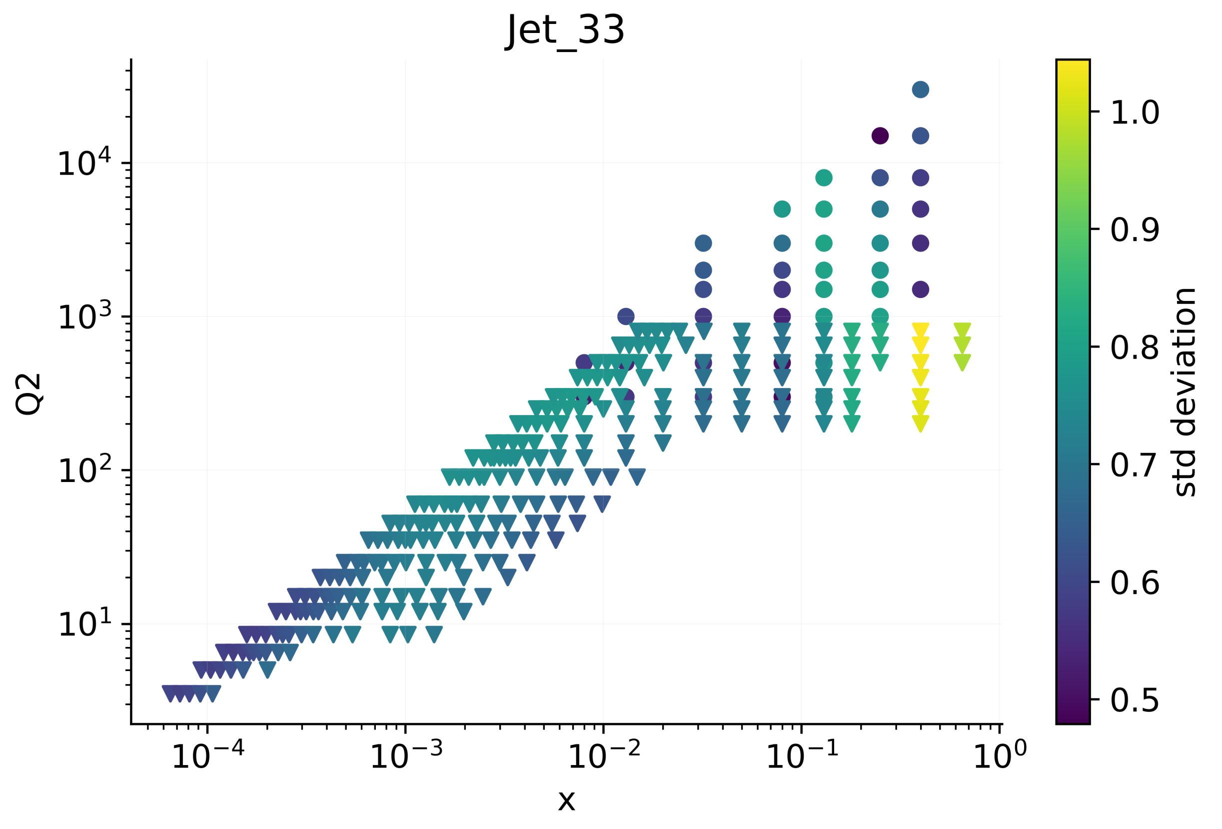
$\lambda = 1$: consistent



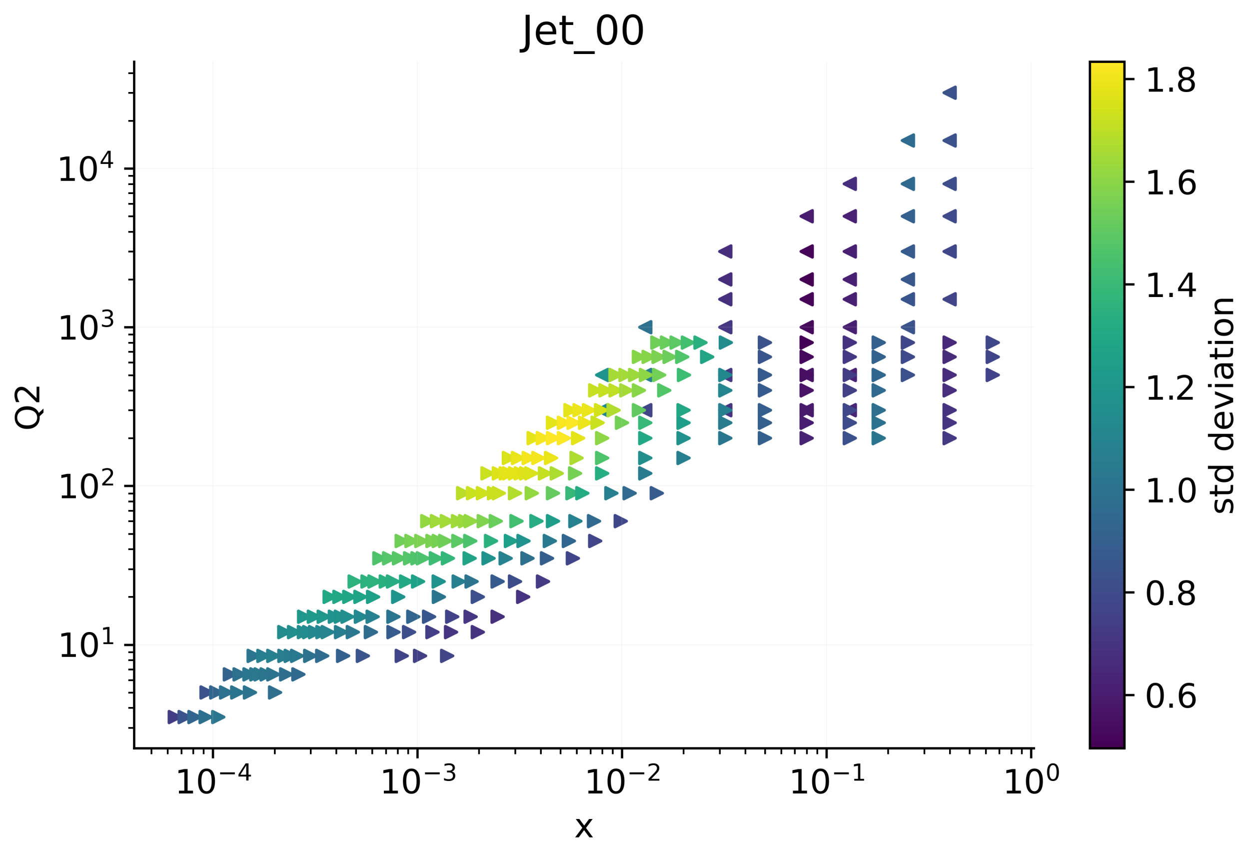
$\lambda = 0.82$



$\lambda = 0.60$



$\lambda = 0.33$



$\lambda = 0$: most inconsistent

- Output is evaluated on DIS observables
- Input inconsistency again ATLAS JET dataset