Chapter 2: A different perspective on uncertainty

What are the limits of replica distribution uncertainties?

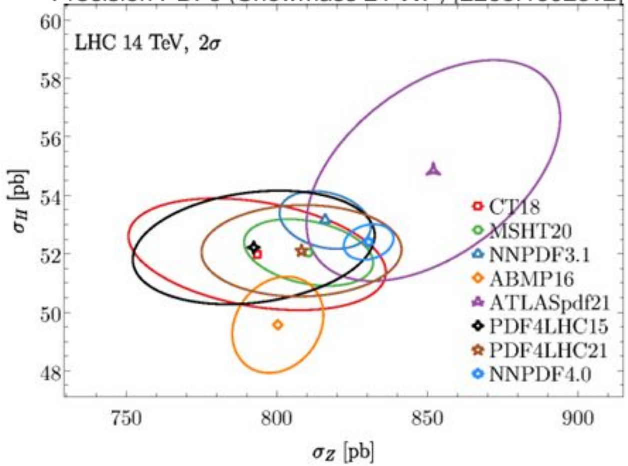
We’ve seen that new data doesn’t just shrink error bars, but shift means.

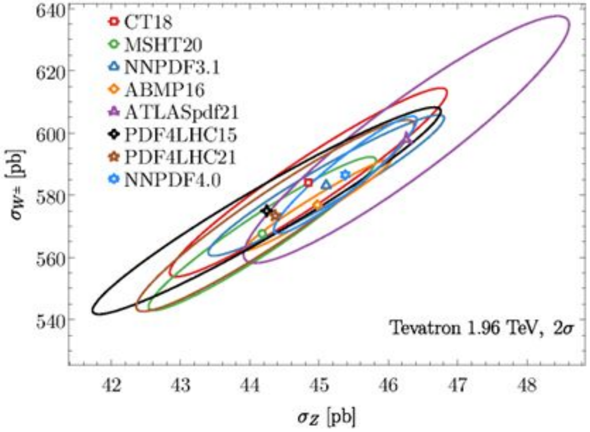
We’ve seen that adding flexibility to the model does the same

The tolerance puzzle

Why do groups fitting similar data sets obtain different PDF uncertainties?

Precision PDFs (Snowmass 21 WP) [2203.13923v2]





The answer has direct implications for high-stake experiments such as W boson mass measurement, tests of nonperturbative QCD models and lattice QCD, high-mass BSM searches, etc.

Comparisons of the latest PDF sets

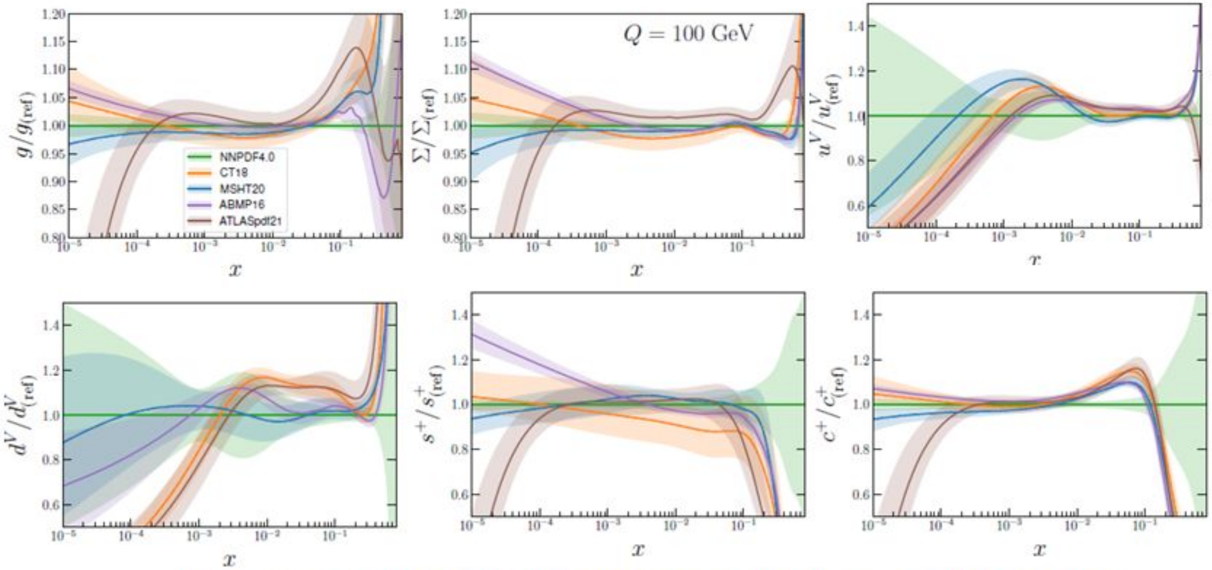
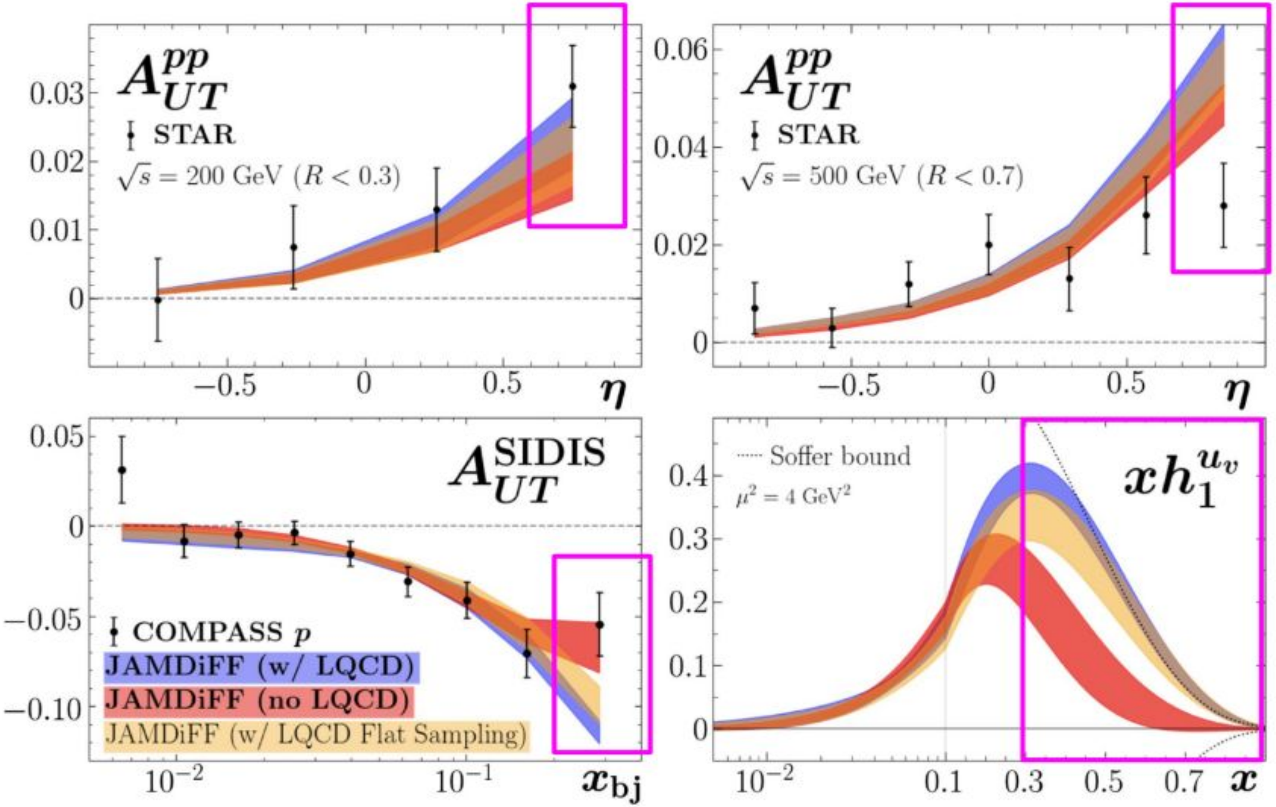


FIG. 2. Comparison of the PDFs at $Q = 100$ GeV. The PDFs shown are the N2LO sets of NNPDF4.0, CT18, MSHT20, ABMP16 with $\alpha_s(M_Z) = 0.118$, and ATLASpdf21. The ratio to the NNPDF4.0 central value and the relative 1σ uncertainty are shown for the gluon g, singlet Σ, total strangeness $s^+ = s + \bar{s}$, total charm $c^+ = c + \bar{c}$, up valence u^V and down valence d^V PDFs.



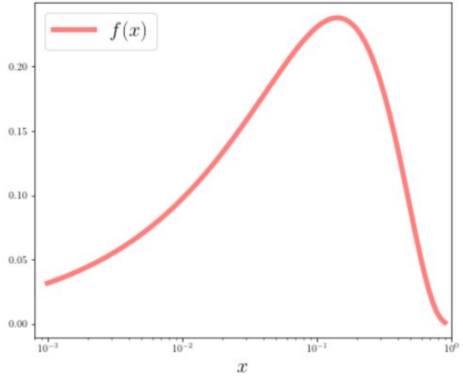
Pitonyak, Cocuzza, Metz, Prokudin, NS, `24 (PRL)
Cocuzza, Metz, Pitonyak, Prokudin, NS, Seidl `24 (PRL)
Cocuzza, Metz, Pitonyak, Prokudin, NS, Seidl `24 (PRD)

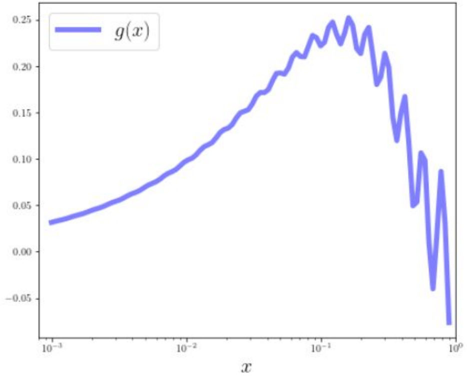
What is the constraining power of the data on the PDFs?

Data could be incompatible, or

Parameterized models lead to artificial certainty, especially at large *x*.

We explore models that look like this





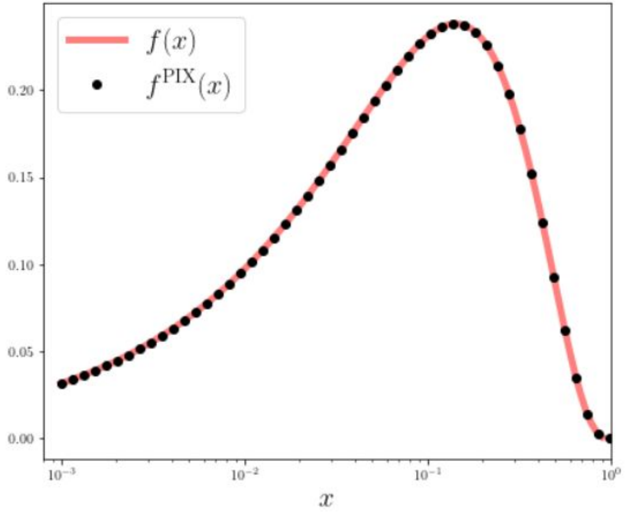
Not like this

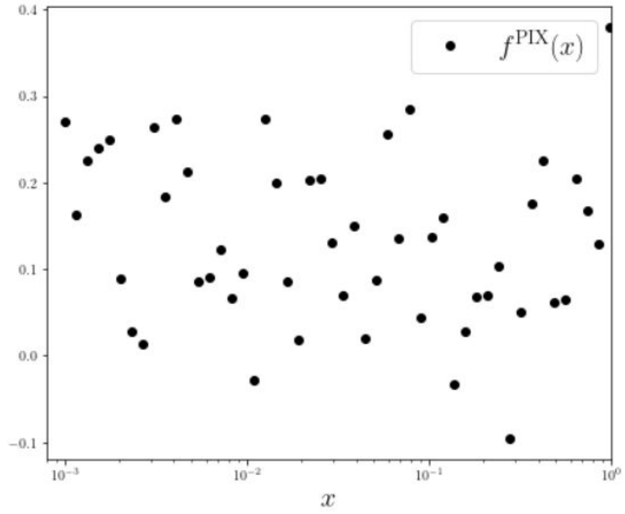
How might we explore the impact data has on input scale PDF without having to worry about model bias?

We need a universal function - test all models simultaneously

Pixelized PDFs

Instead of a parametric model, we ‘pixelize’ our input scale PDF. Each pixel is its own free parameter.

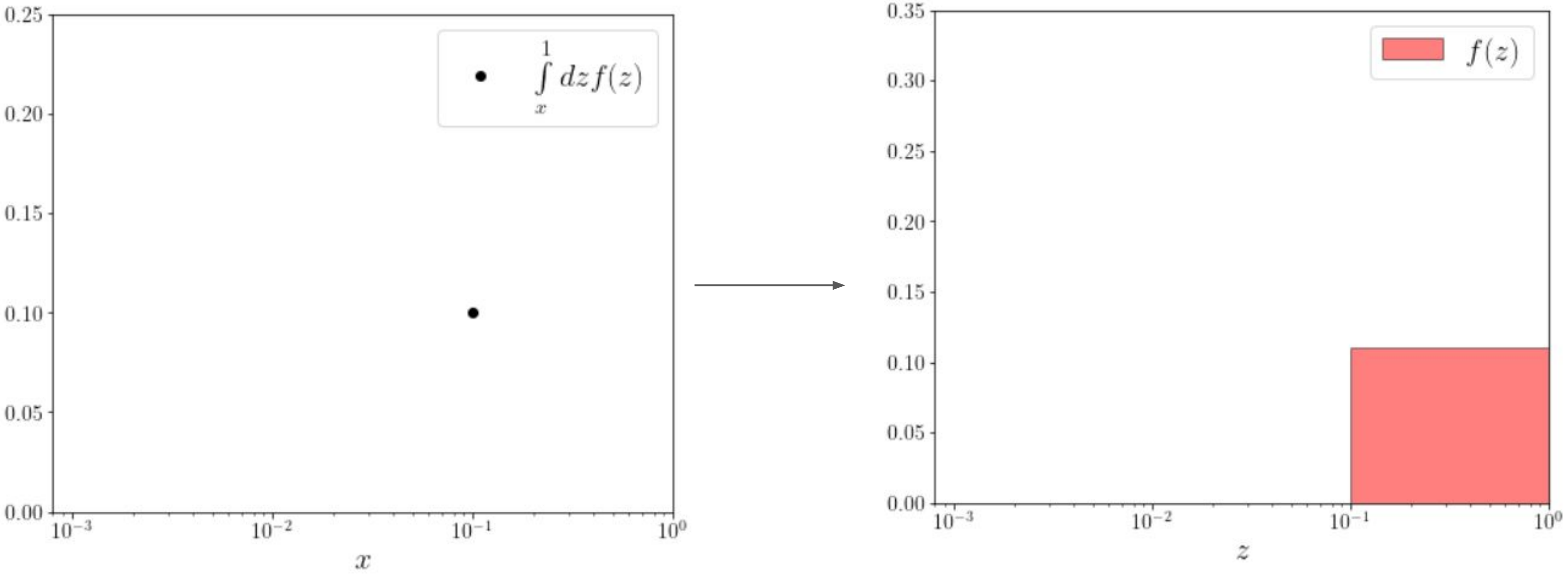




This is a universal function approximator, at the cost (benefit?) of introducing a finite resolution in *x*. Doesn’t remove all model bias, but it is much more flexible than the parametric models used so far, especially because it’s easy to increase flexibility by adding more pixels

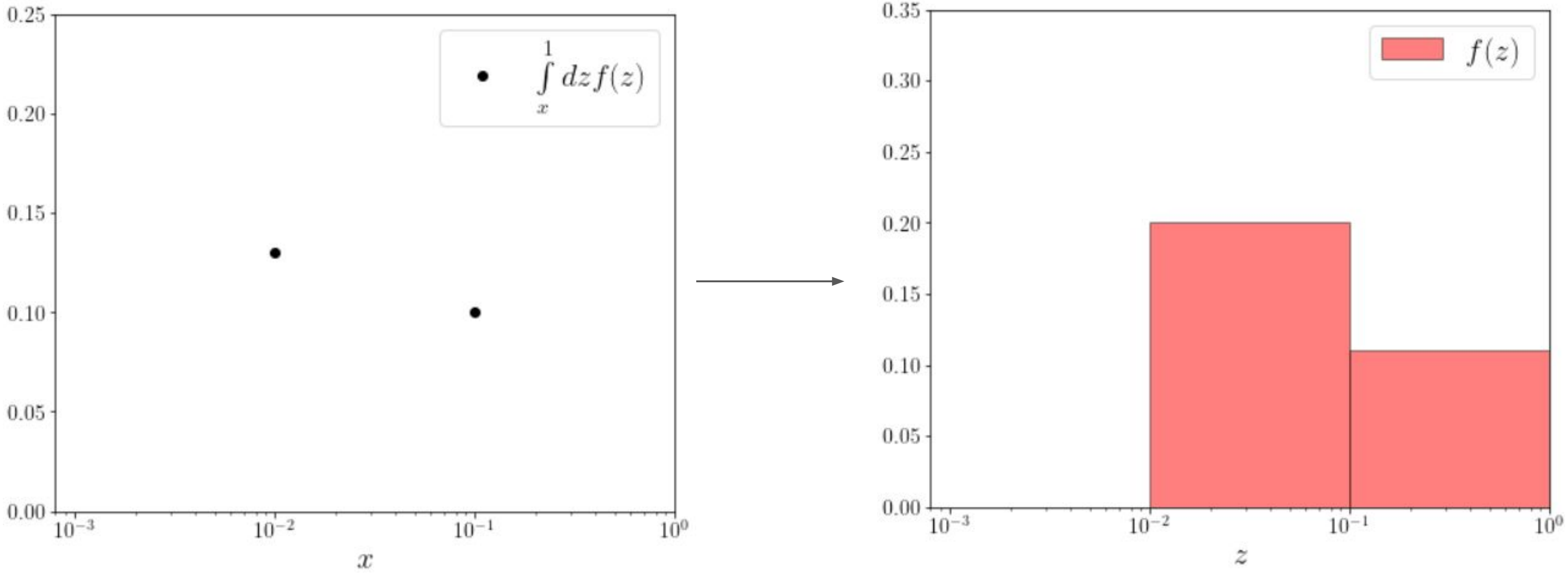
In convoluted observables, what can we learn?

Consider the most trivial convolution: the partial moment. What can we infer from it?



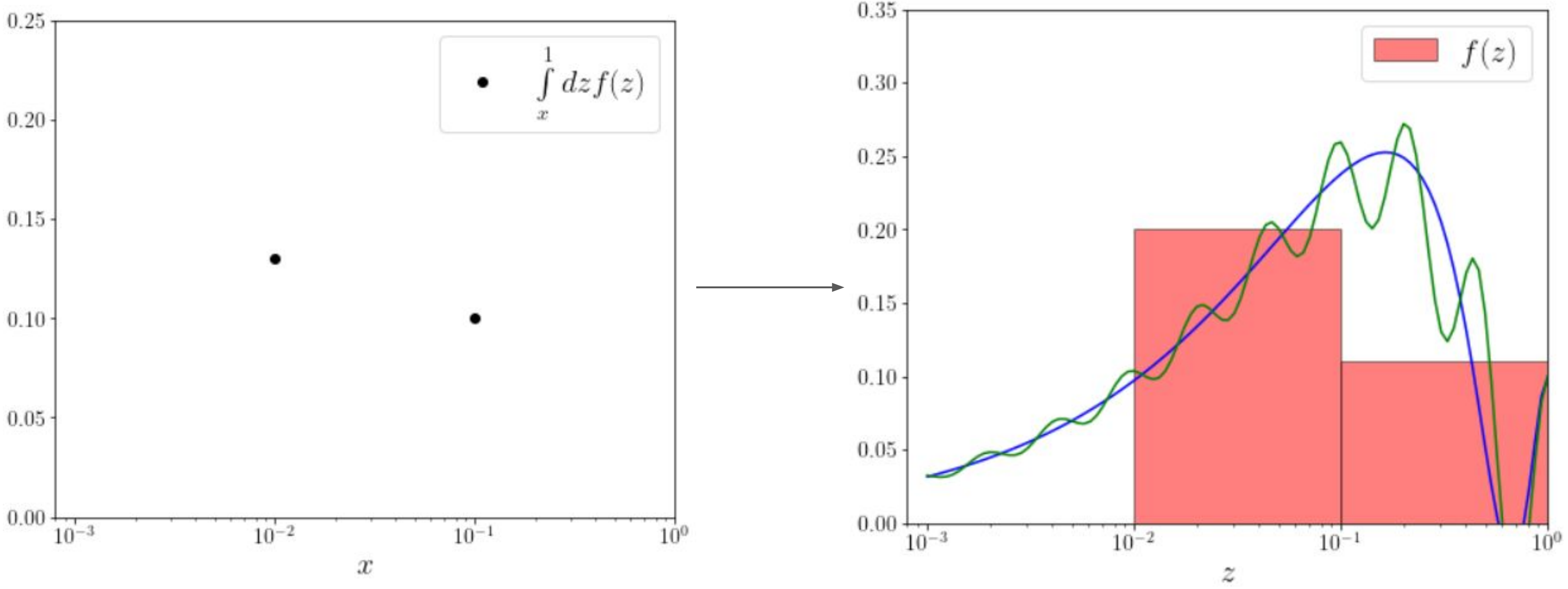
In convoluted observables, what can we learn?

Consider the most trivial convolution: the partial moment. What can we infer from it?



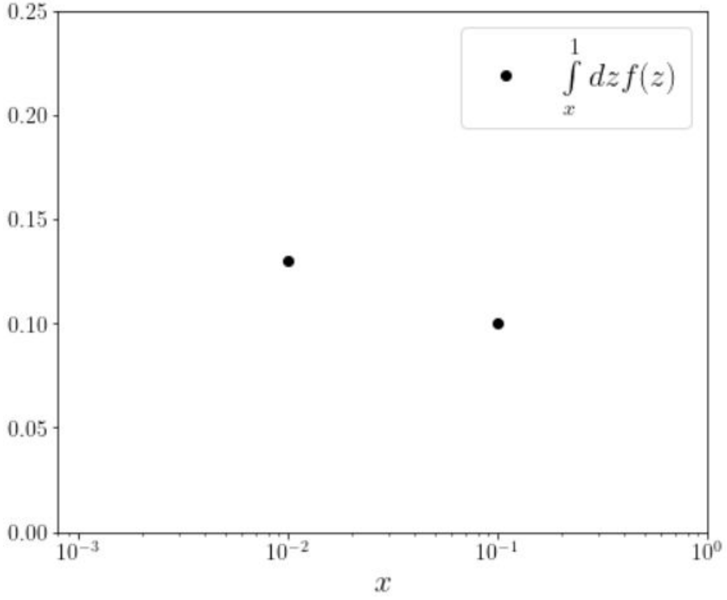
The area under the curve is constrained, not the exact form

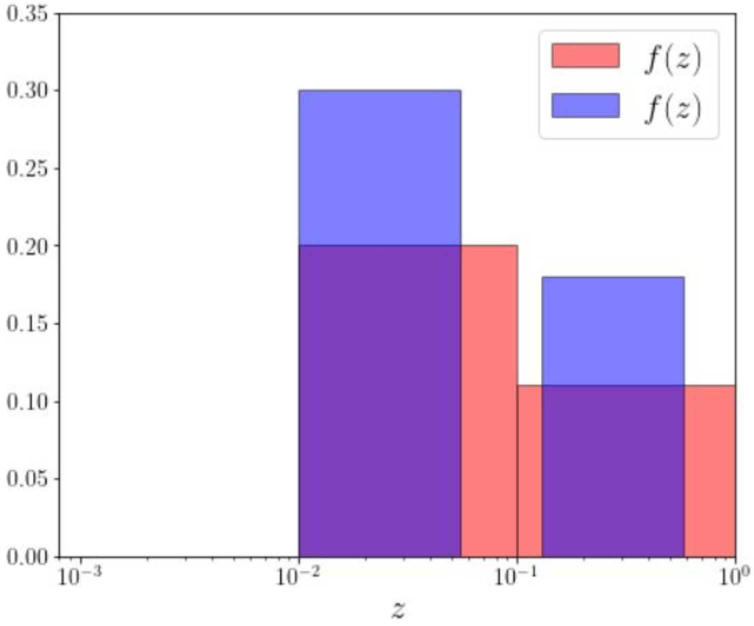
Isomorphic to the average value of the function between two *x* points being constrained



What do we expect for the distribution of valid replicas?

Consider loading all the information into individual pixels

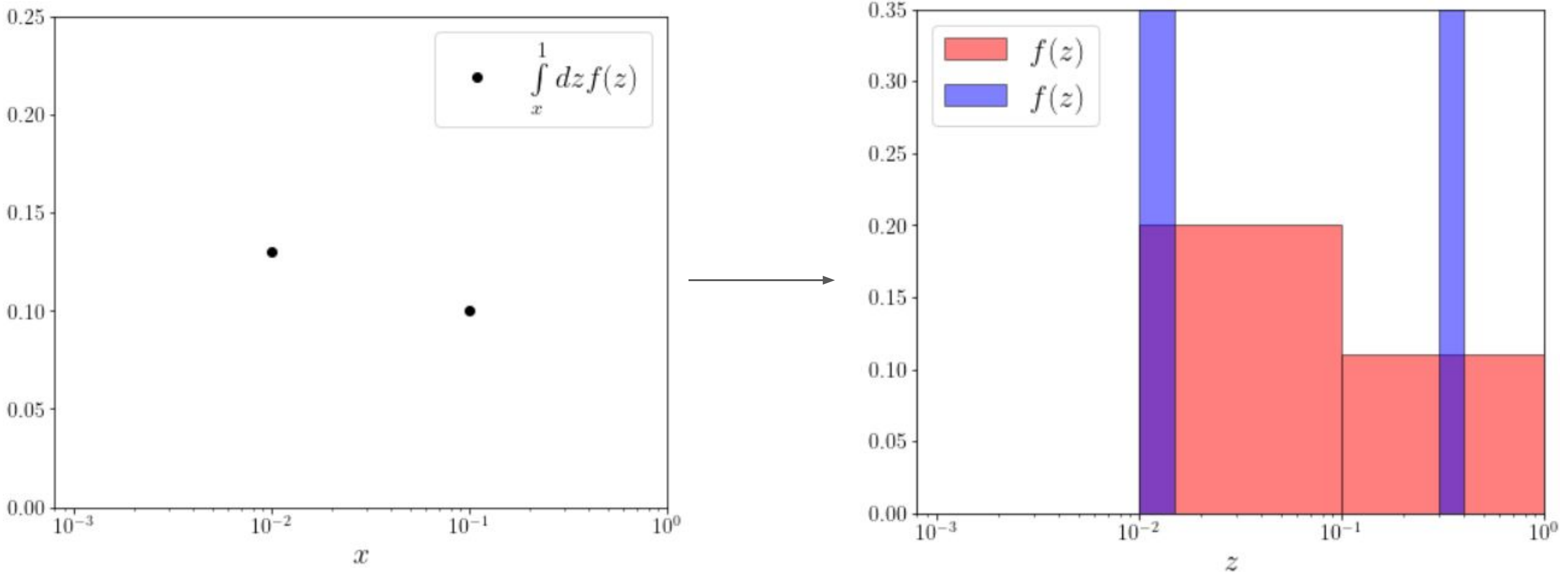




Set some pixels to 0, let the other pixels pick up the rest of the area

What do we expect for the distribution of valid replicas?

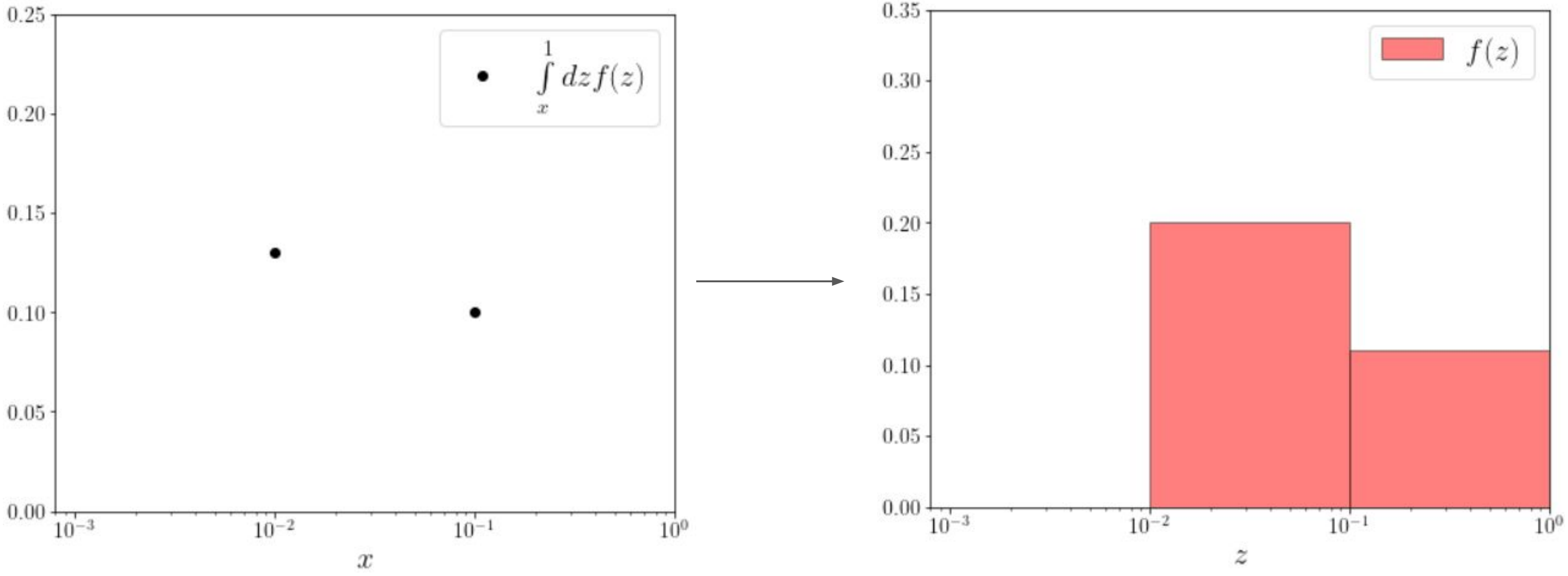
The finer your grid, the wider the distribution of replicas!



This implies an infinite uncertainty for infinitely flexible models, even when data has zero uncertainty!

Data tell us about the resolution of PDFs: $\langle f(x_i<x<x_{i+1})\rangle$

These histograms are the real constraints of data



How can we infer the PDF resolution from DIS data?

- The convolutions involved in computing observables are much more complex

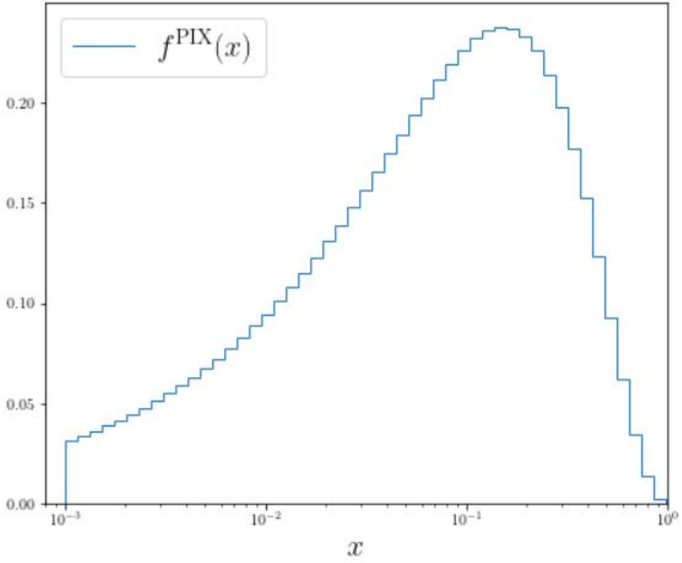
$$F_2(x, Q^2) = x \sum_q e_q^2 \int_x^1 \frac{dz}{z} \, C_q(x/z, \alpha_s) f_q(x, Q^2)$$

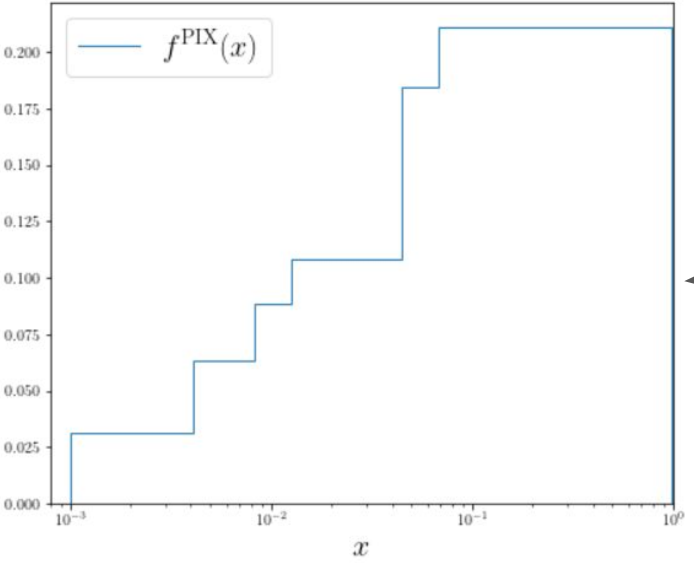
$$\frac{\partial}{\partial \ln Q^2} f_i(x, Q^2) = \frac{\alpha_s}{2\pi} \int_x^1 \frac{dz}{z} p_{ij}(x/z, Q^2) f_i(x, Q^2)$$

- I don’t know how to infer the resolution analytically
- But I have an algorithm that starts with a high resolution fit and gradually lowers the resolution until the quality of the fit begins to degrade

How to determine the lowest resolution

Compare high and low resolution fit, if both give good chi^2 (within some tolerance) then we have a better idea of how low the resolution really is

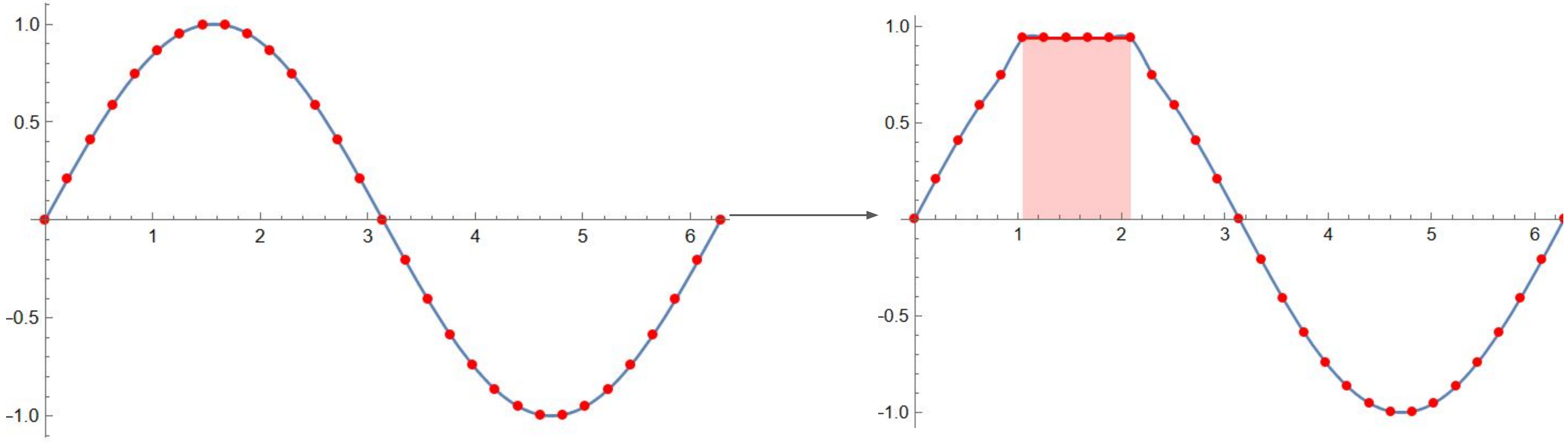




More informative about our true state of knowledge

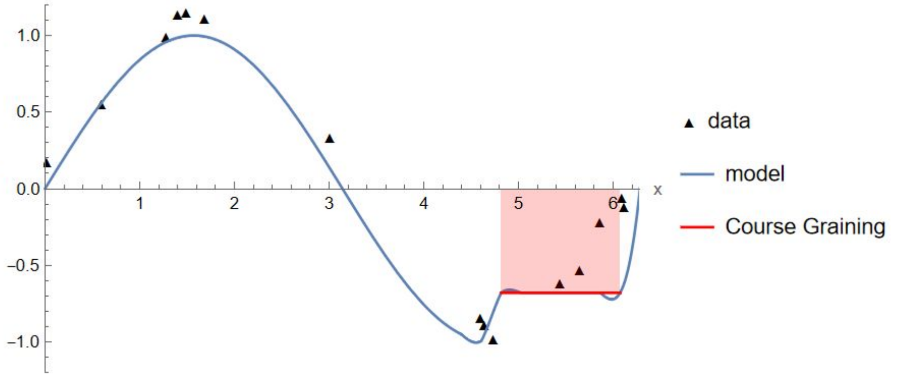
Bucketing

Take some sequential pixels, and replace them with their average

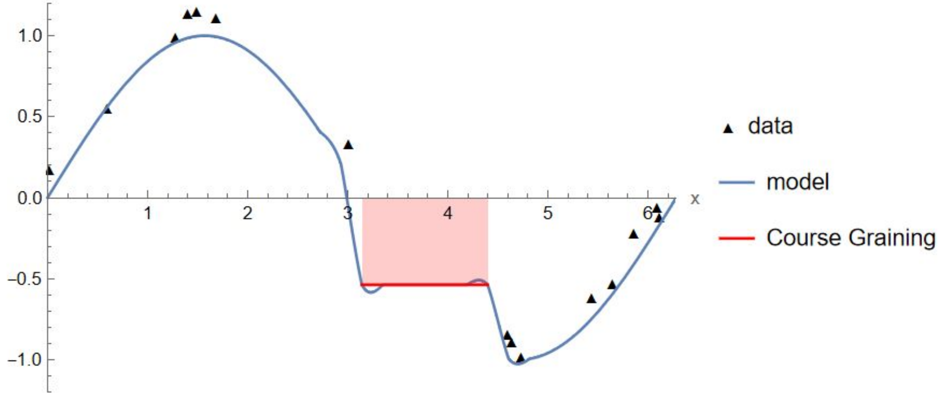


Reject the Bucket if it Harms the Chi^2

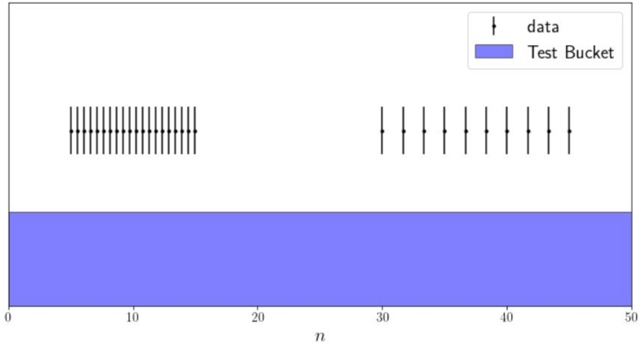
[χ^2 = , 5.33139],



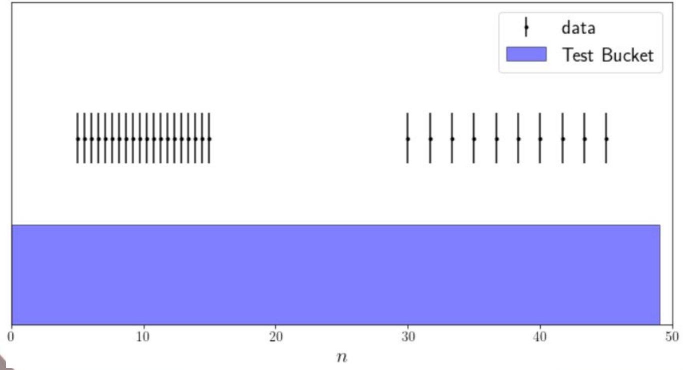
[χ^2 = , 1.79372],

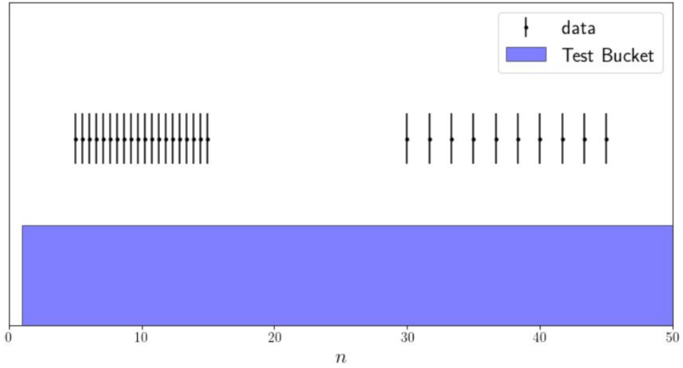


Recursively check all bucket sizes and positions



Width = 50

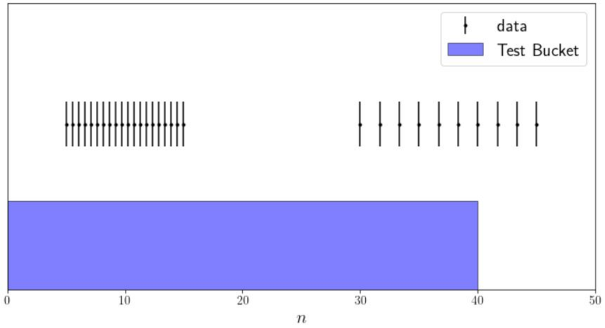


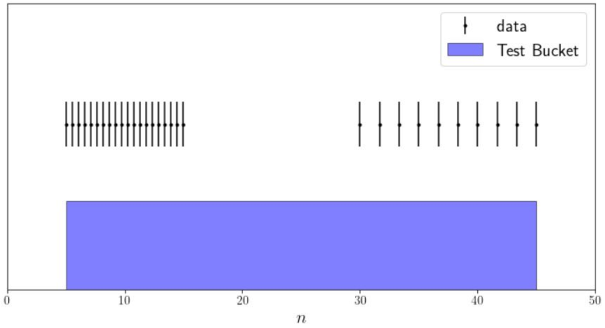


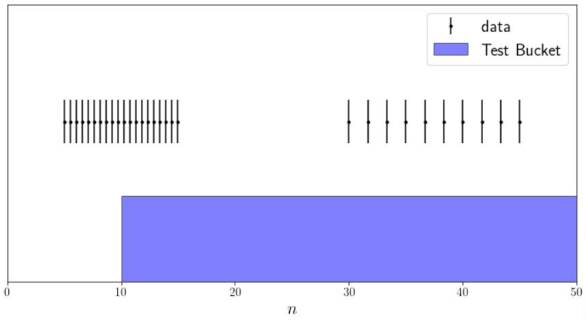
Width = 49

Recursively check all bucket sizes and positions

Width = 40

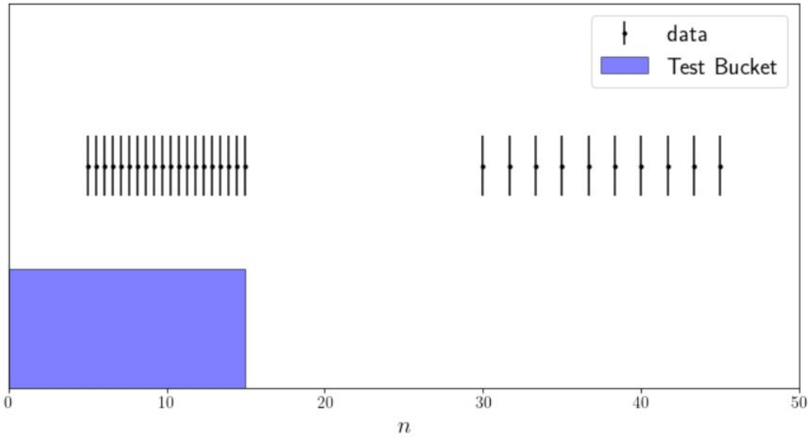


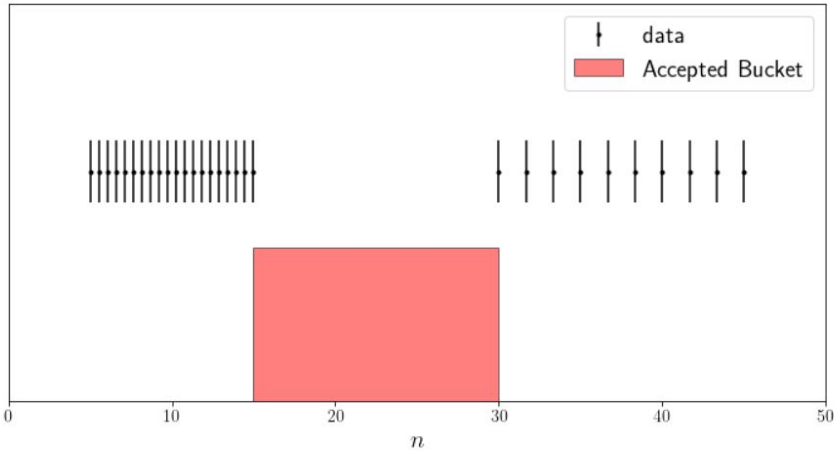




Recursively check all bucket sizes and positions

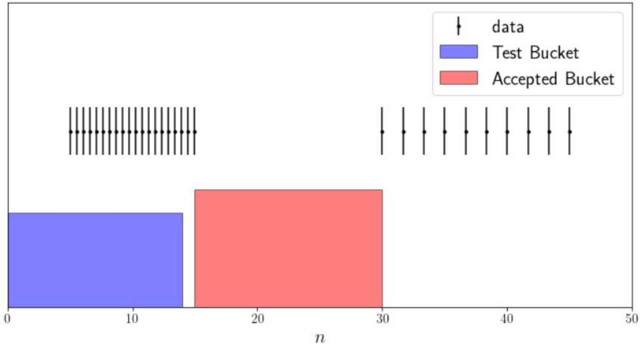
Width = 15

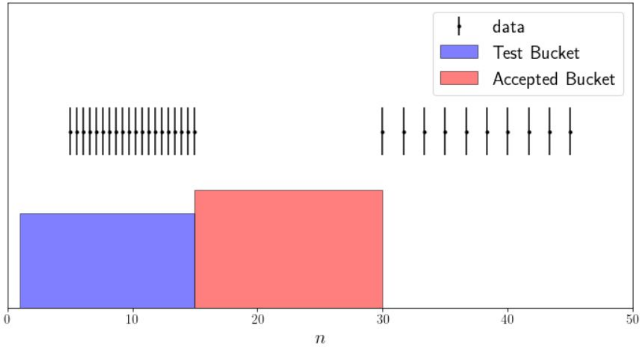


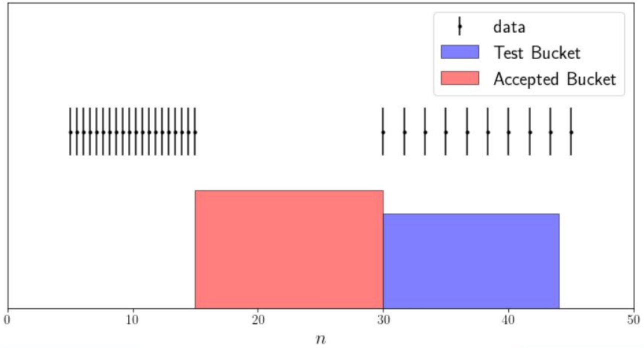


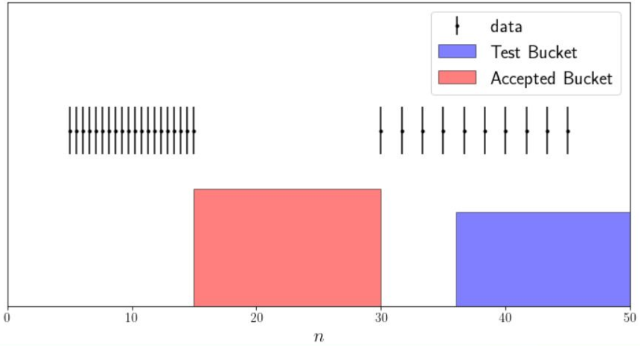
Recursively check all bucket sizes and positions

Width = 14



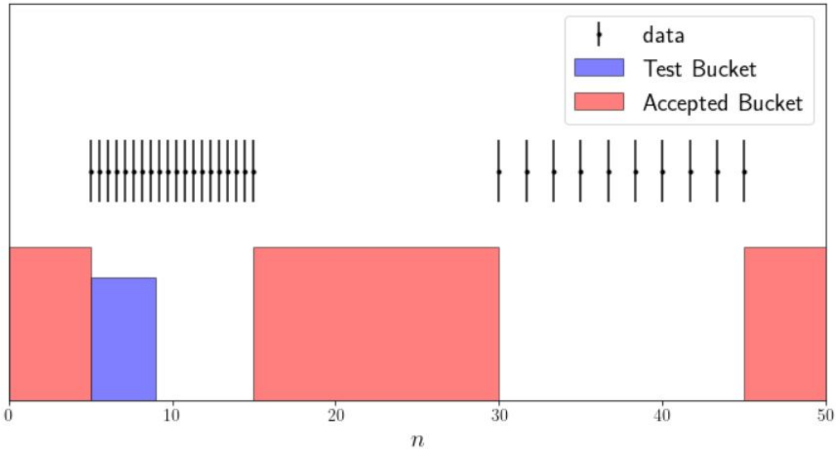


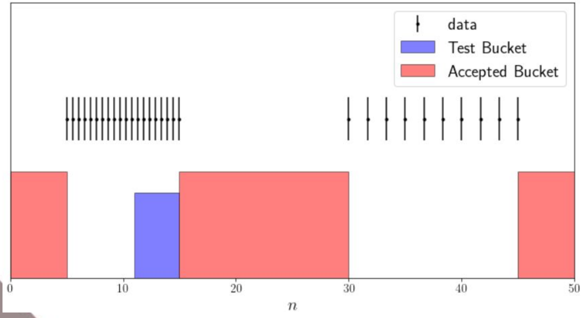


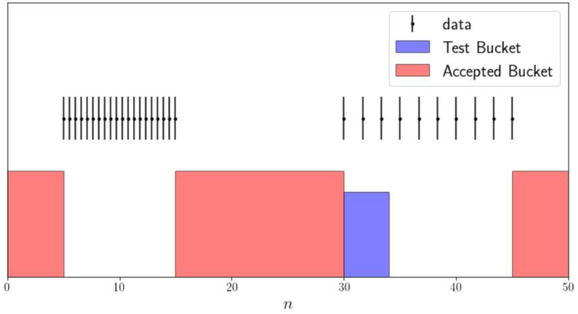


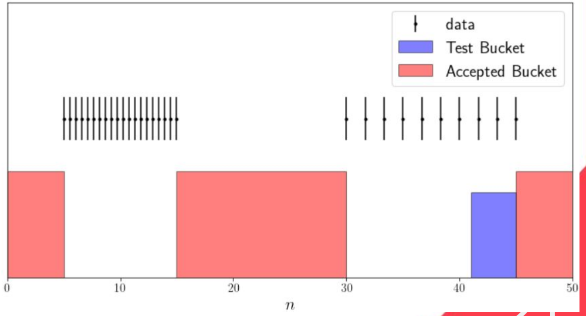
Recursively check all bucket sizes and positions

Width = 4

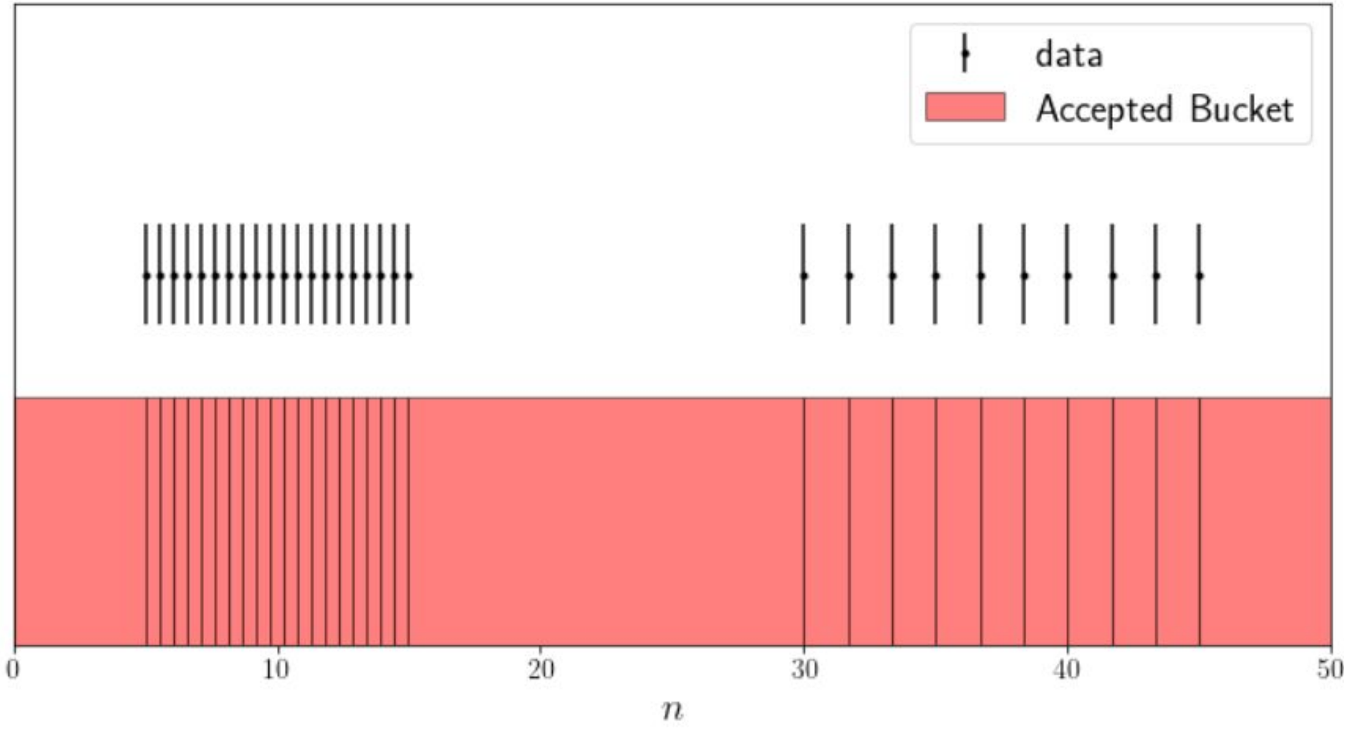




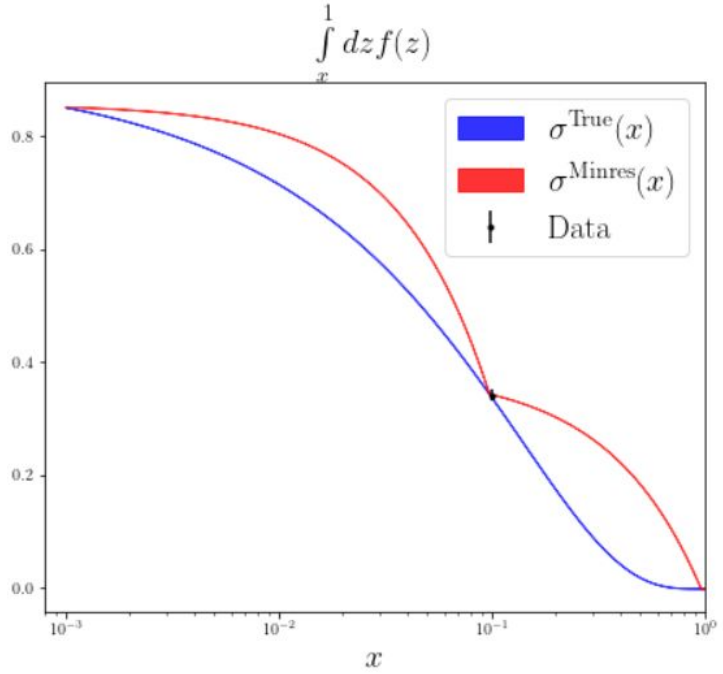


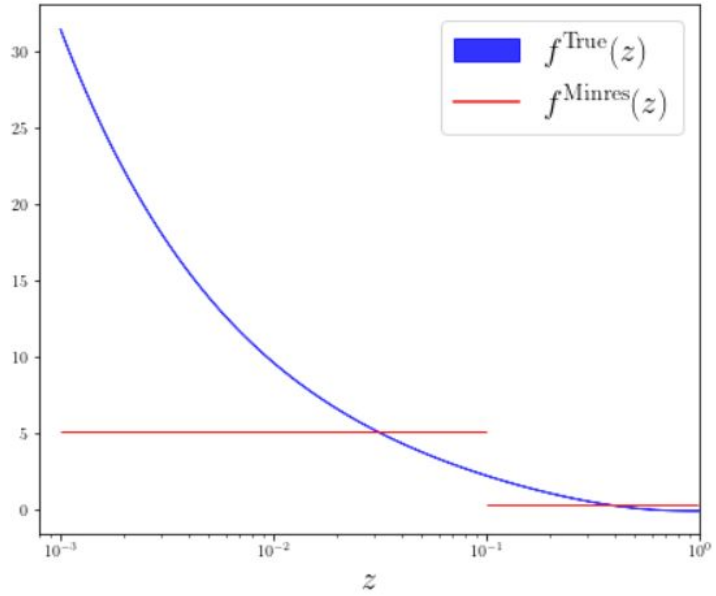


Recursively check all bucket sizes and positions

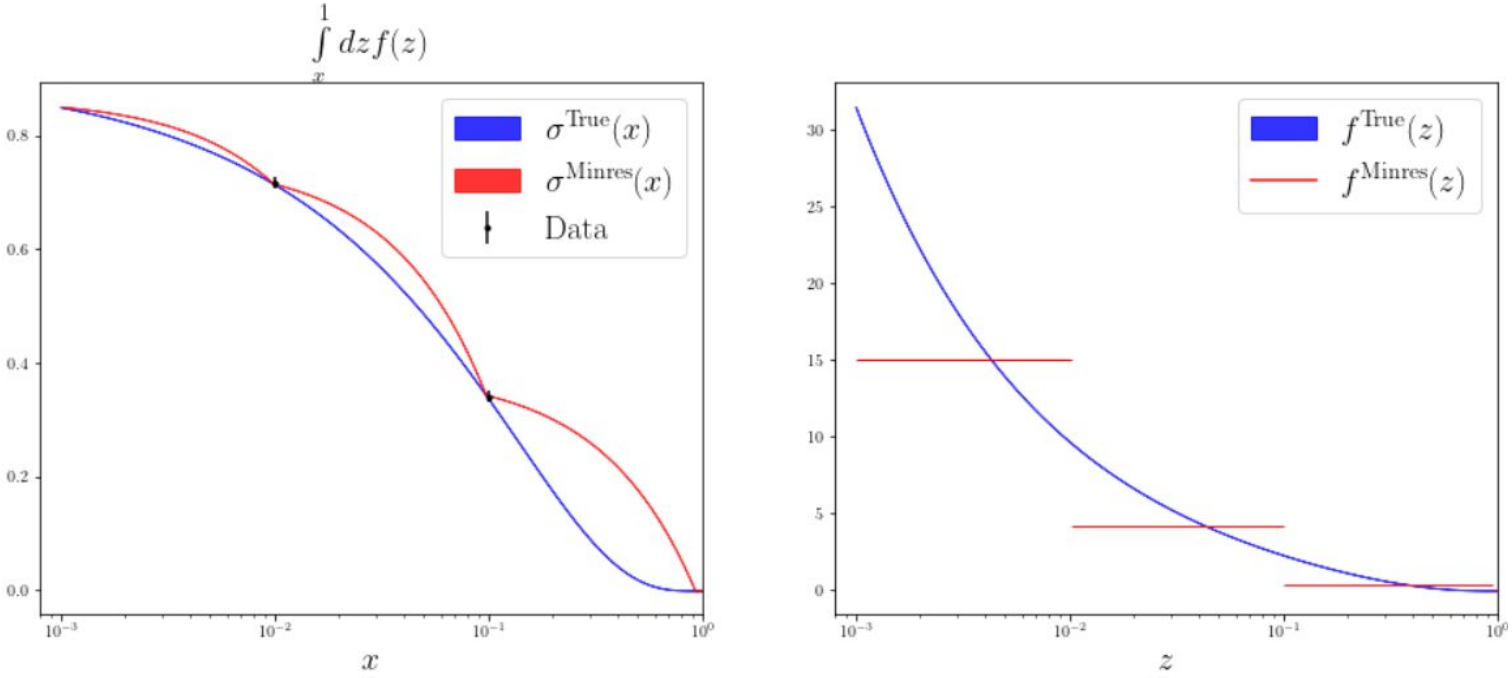


Example: Trivial Kernel





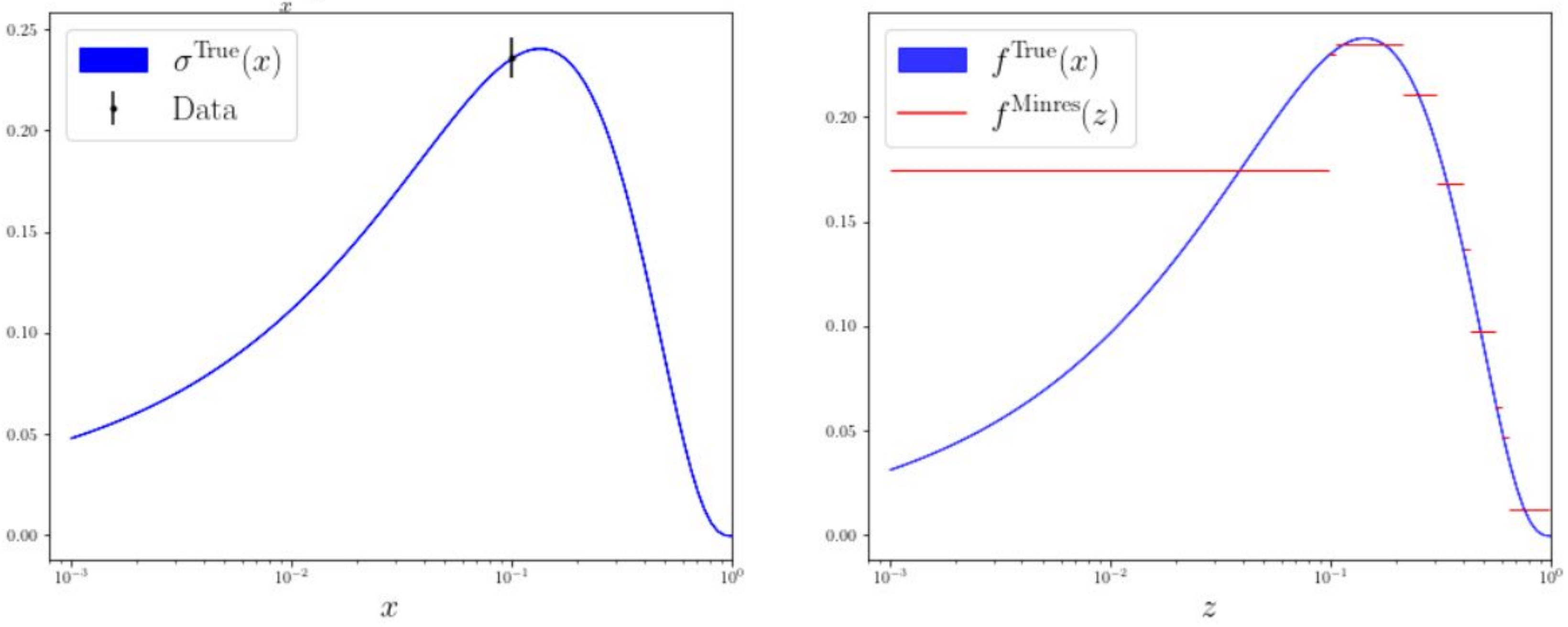
Example: Trivial Kernel



Does what we expect when I don’t show the bad examples

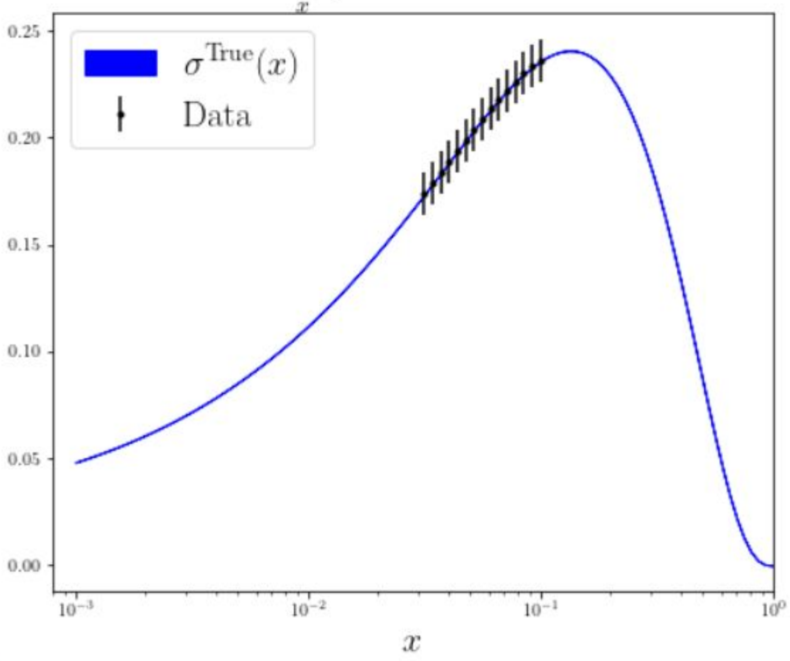
Example: Pgg Splitting Kernel

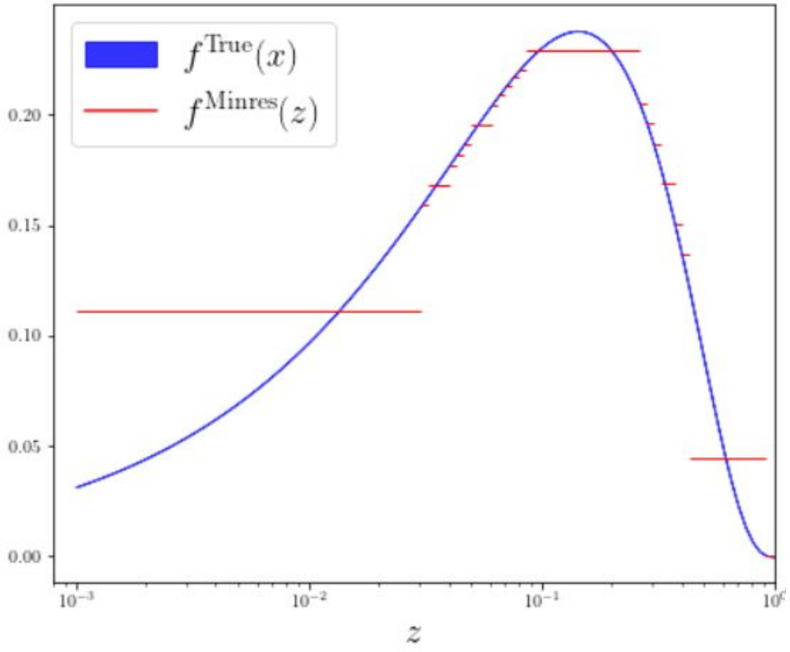
$$\int_x^1 \frac{dz}{z} K(z) f(x/z)$$



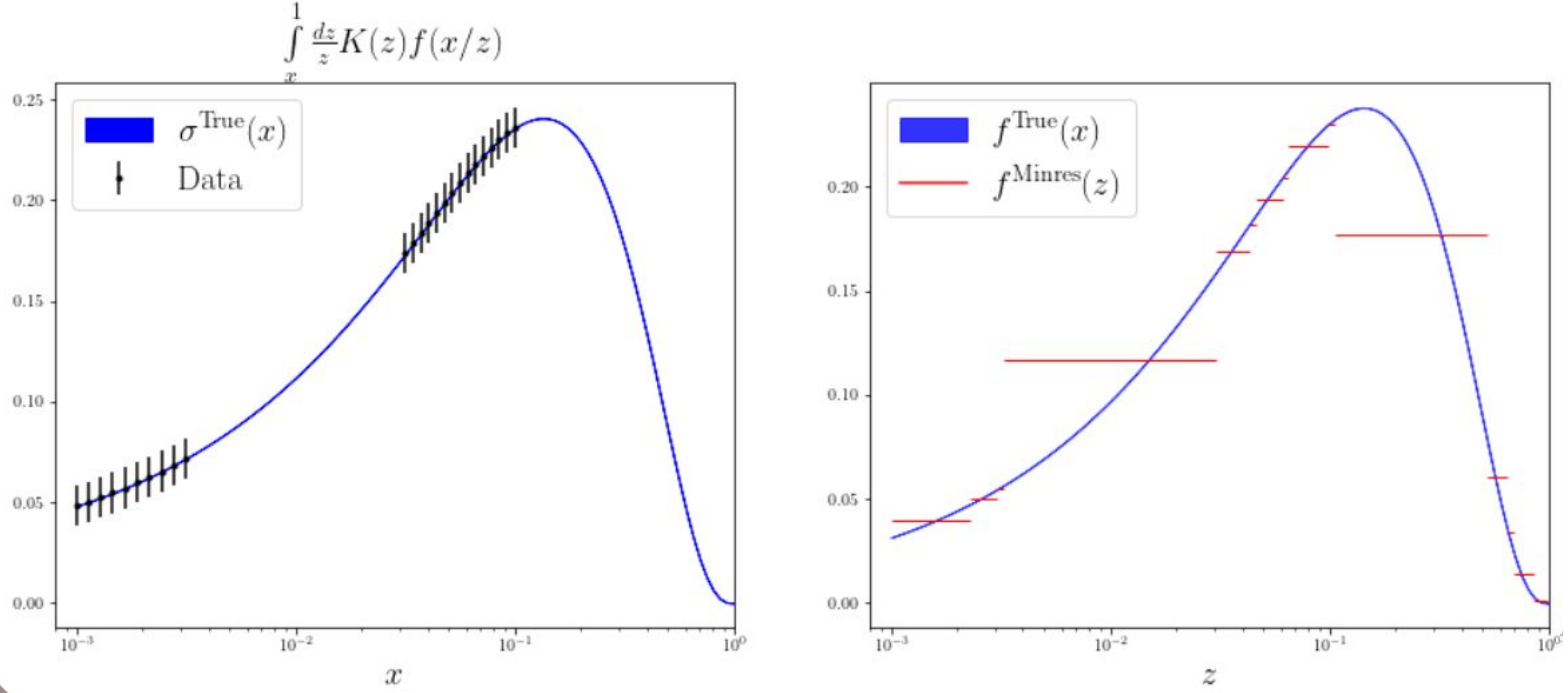
Example: Pgg Splitting Kernel

$$\int_x^1 \frac{dz}{z} K(z) f(x/z)$$





Example: Pgg Splitting Kernel



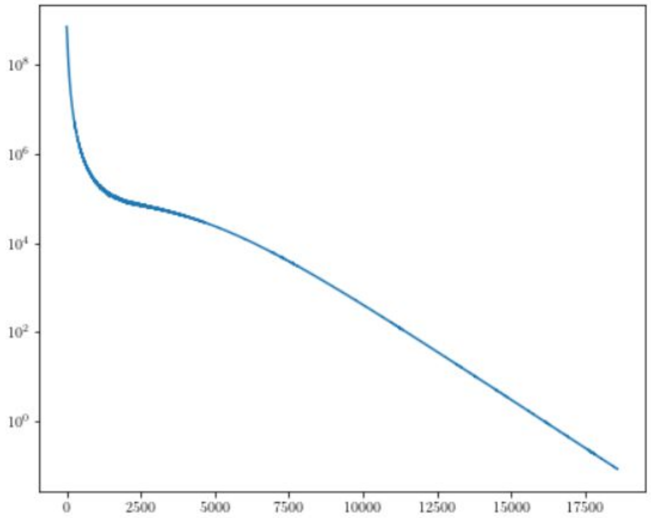
Incorporating Uncertainty of the Data

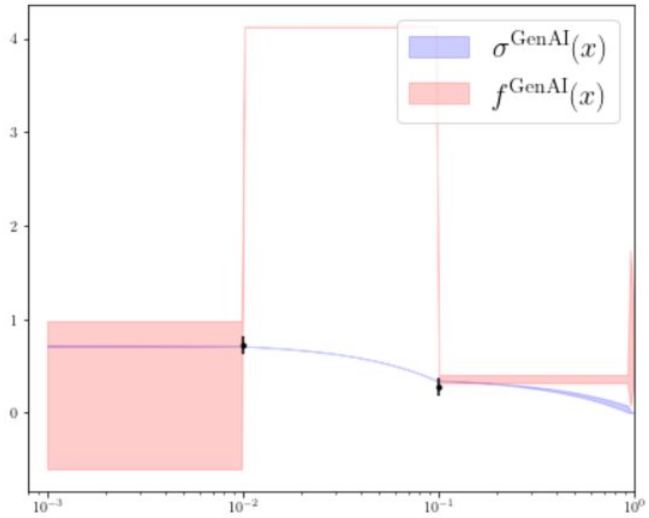
- Usually experimental uncertainties are incorporated into fits via bootstrapping/reshuffling of data
- On each reshuffle, a new set of parameters are trained in order to generate a new replica - the distribution of replicas give the statistical uncertainty of the model
- Now that we have a minimum resolution fit, our parameters are just the intensities of the low resolution pixels
- So we can apply reshuffling to ge uncertainties on the pixel intensities
- We have decorrelated the resolution and the uncertainty of the data

Fit Minimum Bias Model

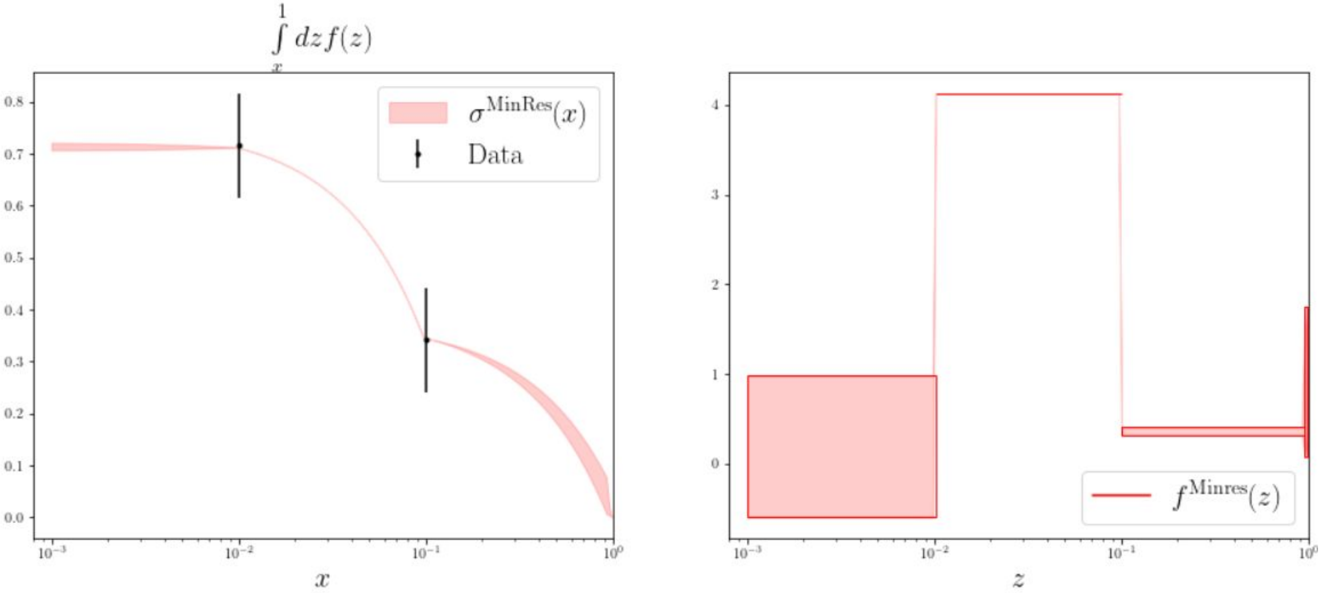
Our minimum bias model is given by the minimum resolution staircase function, where the height of each stair is a free parameter. Fit to the data

Example: Trivial Kernel





Example: Trivial Kernel



- Vertical uncertainties given by replica distribution

Additional flavors

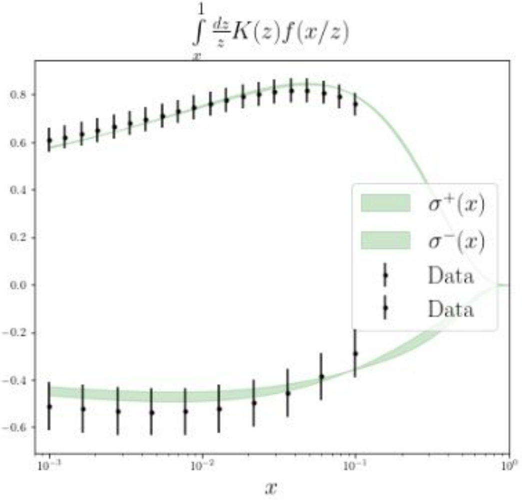
Loop over flavors is inserted after the loop over bucket widths.

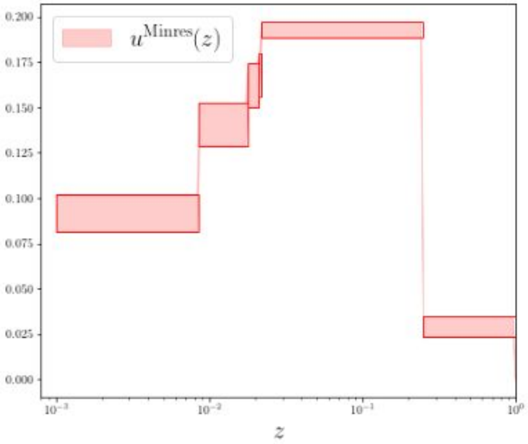
Example:

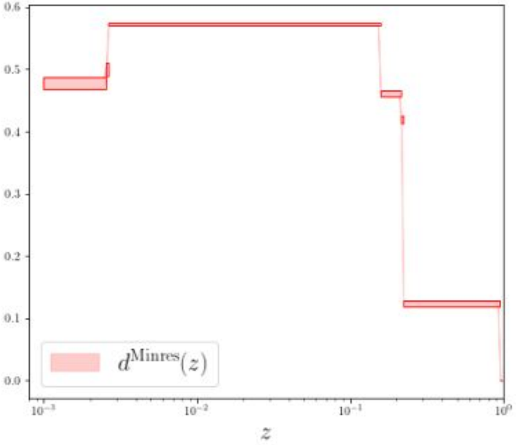
$$\sigma^{+}(x) = \int_{x}^{1} \frac{dz}{z} K(z)\,(u+d)(x/z)$$

$$\sigma^{-}(x) = \int_{x}^{1} \frac{dz}{z} K(z)\,(u-d)(x/z)$$

Preliminary!







Strengths and Weakness

Strengths:

- It exists
- Behaves reasonably
- Deterministic (and reproducible). Can serve as a cross-check for future algorithms

Weaknesses:

- Doesn't extend nicely to higher dimensions
- Greedy algorithm - finds local optima
- Doesn't quite pass all sanity checks

Other Methods

Kevin Braga’s Method

- Take individual pixels and adjust their position
- If this harms the chi2, insert more pixels
- If this doesn’t harm the chi2, merge pixels

Emil Constantinescu

- Subdivide pixels based on evolution - larger derivative = more subdivisions
- In our case derivative must be of chi2. Might be doable in our matrix evolution framework

Conclusions

- Model flexibility is somewhat isomorphic to replica uncertainty, so we should be careful about comparing different models
- Data can only constrain the resolution , or the average value, or integrals of the PDFs
- The resolution of PDFs is related to the distribution of data (in x (mostly))
- The uncertainty of data can then be turned into uncertainty of the average value of the PDFs within a bin - Decorolating data distribution from data uncertainty