Proposal

- Procedure
  - Generate SIDIS events with given parameters
  - Analyze the data, similar how the experimental data will be analyzed
  - Try to extract the parameters used to generate the data
- Realization
  - Developed a library that can generate physics events according to cross section calculated for given set of parameters.
  - Use this library to generate multiple samples with random set of parameters.
  - Use the generated samples to trail Neural Network on the sample distributions.
  - Generate data sample for one set of parameters, analyze the resulting physics events. Extract the parameters of model based on physics observable distributions.

SIDIS cross section

$$\frac{d\sigma}{dx\,dy\,d\phi\,dz\,d\phi_h\,dP_\perp^2} = \frac{\alpha^2}{xyQ^2}\frac{y^2}{2(1-\epsilon)}\left(1+\frac{\gamma^2}{2x}\right)\times K$$

$$K = \left(F_{UU,T} + \epsilon F_{UU,L} + \sqrt{2\epsilon}(1+\epsilon)\cos\phi_h F_{UU}^{\cos\phi_h} + \epsilon\cos(2\phi_h)F_{UU}^{\cos 2\phi_h}\right)$$

$$F_{UU,T} = x f_1(x) D_1(z)\frac{1}{\pi P_T}e^{-\frac{p_T^2}{P_T}}$$

$$f_1(x) = (1-x)^{C_1}x^{C_2}$$

$$D_1(z) = C_3(1-z)^{C_4}$$

$$P_T = z^2 < k_T > + < P_T >$$

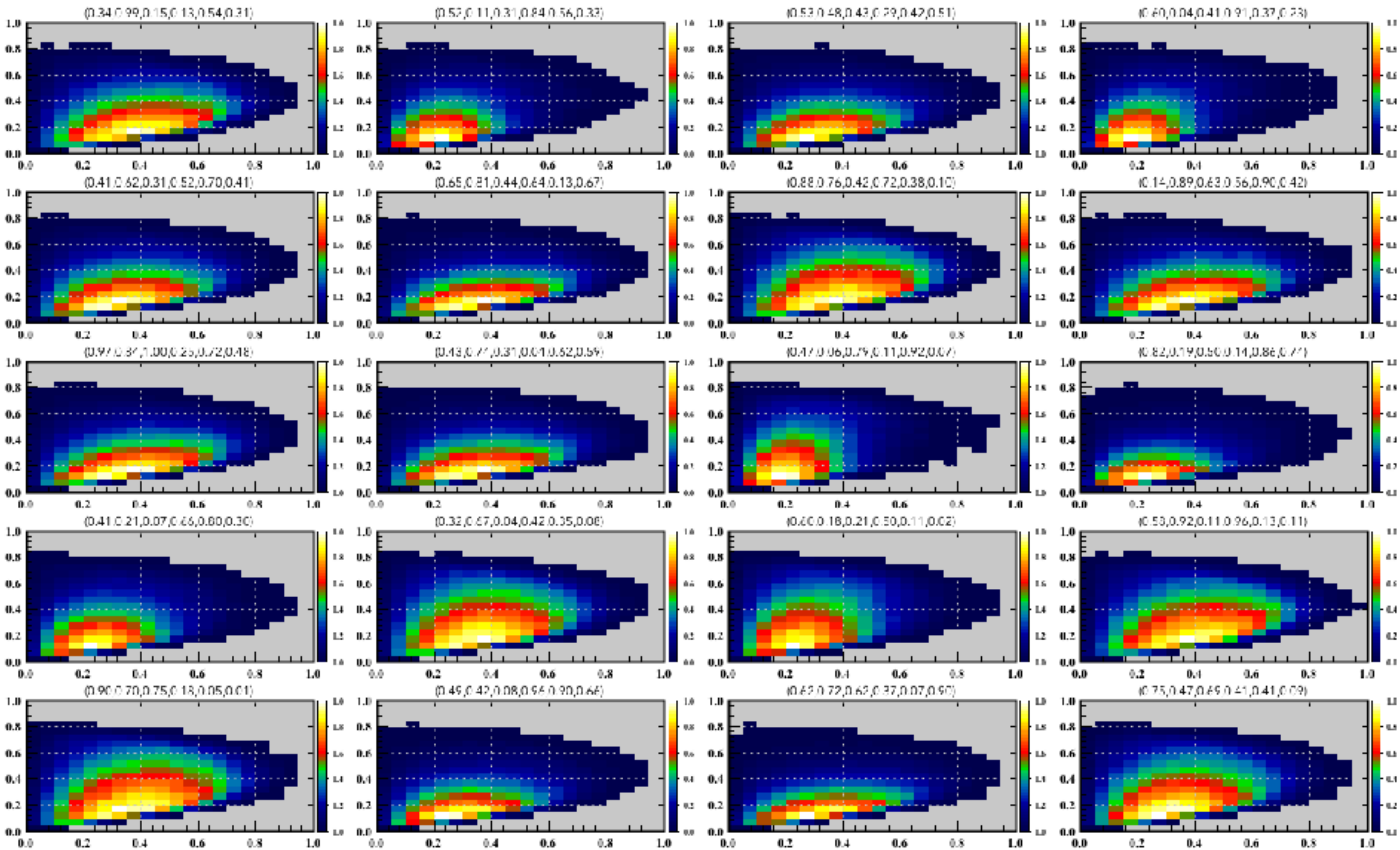
P	Value	Min	Max
<KT>	0.33	0.05	0.4
<PT>	0.16	0.05	0.4
C1	3.00	2.00	5.00
C2	-1.313	-2.00	-1.00
C3	0.80	0.50	1.00
C4	2.00	1.00	4.00

What was done

- Training Neural network
  - Generated training sample (z vs pt) for one kinematical bin in (Q2,xb)
  - The training sample 2D histogram was normalized to 1 (important)
  - Neural network was trained on resulting distributions
- Fitting the data:
  - Generated data with random parameters
  - Analyze data to extract z vs pt distribution
  - Extract the parameters from Neural Network and compare to the random parameters the data was generated with

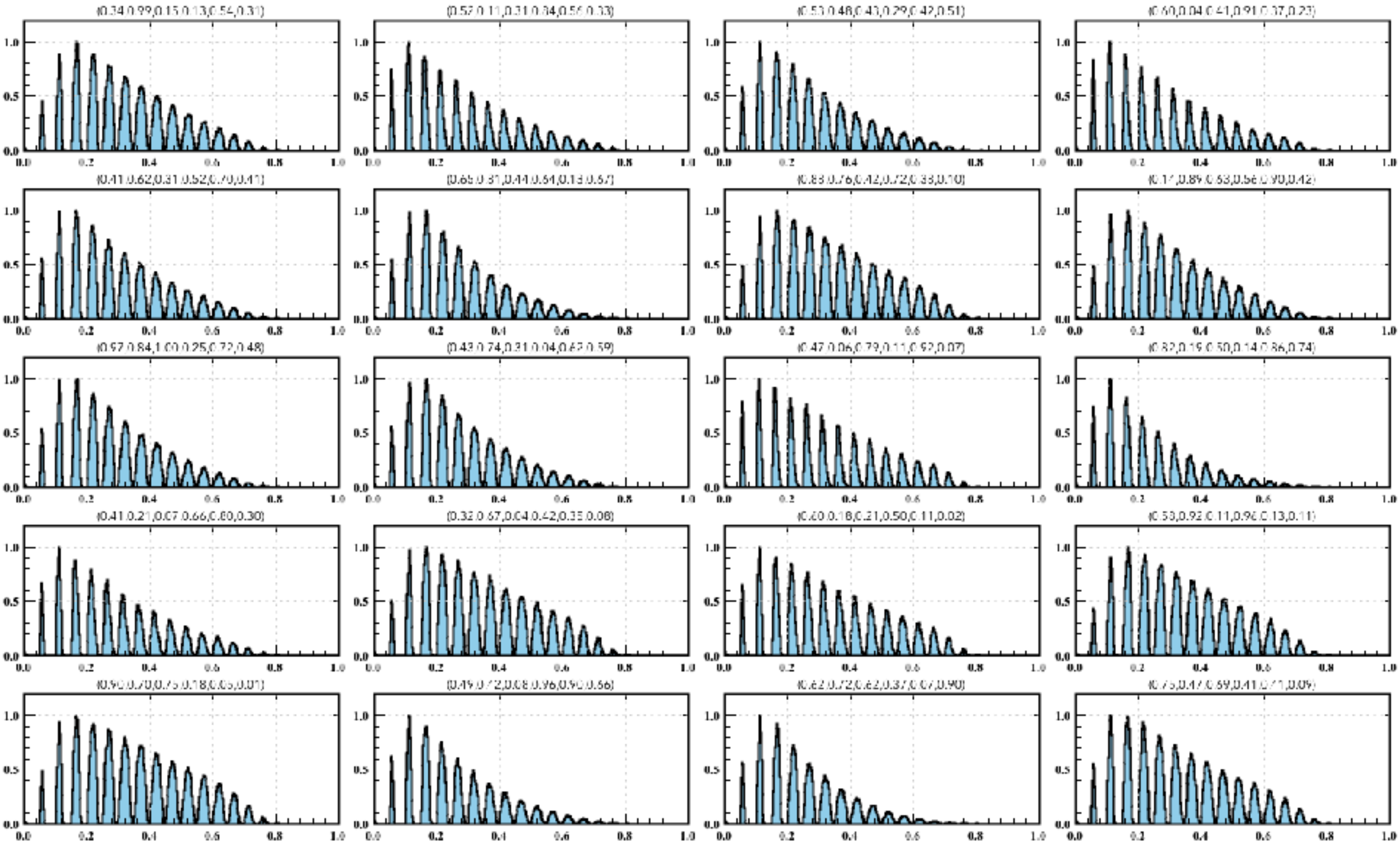
Training Sample

- generated samples (2000) with 200K events in each
- training sample is used to train neural network



Training Sample 1D

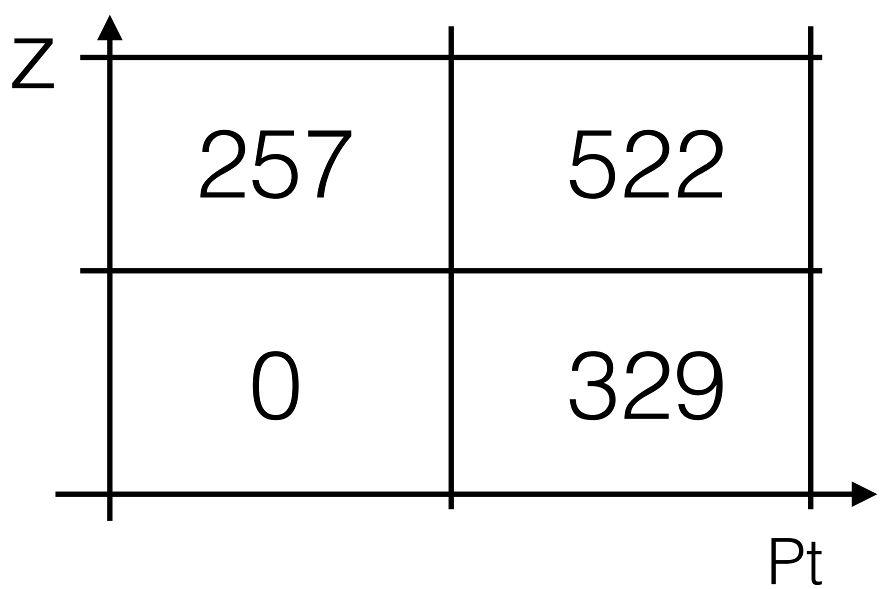
- 1D distributions of pt vs Z for each training sample for random parameter set

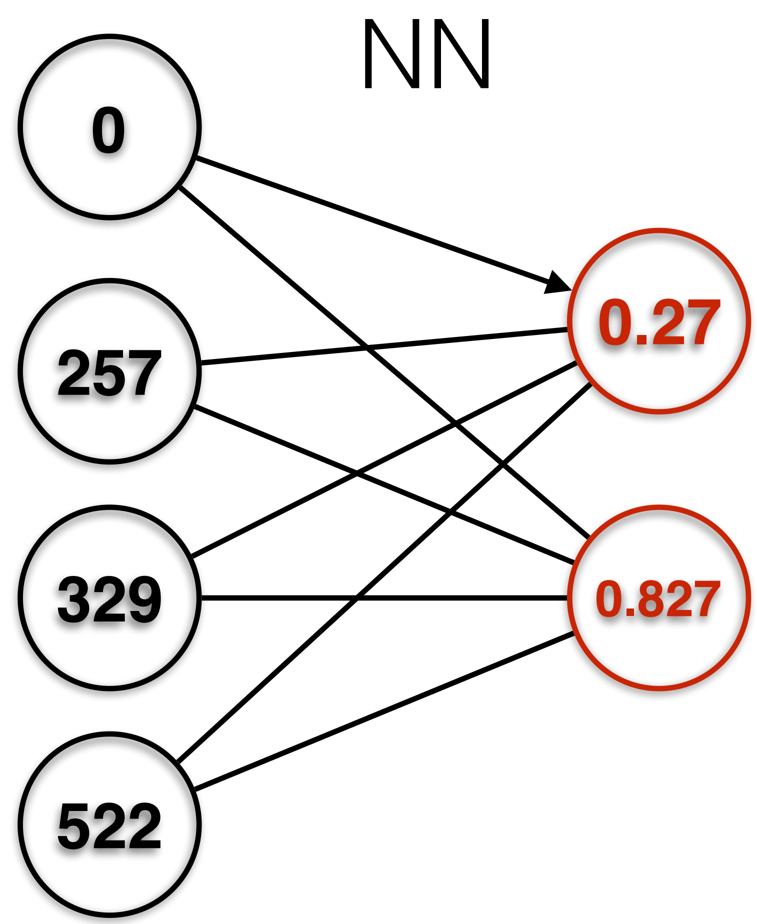


Neural Network

- Generate events with given values of parameters
- Fill histogram for given variables (pt and z in this case)
- Construct input nodes for neural network from the histogram
- Output nodes are the values are parameters

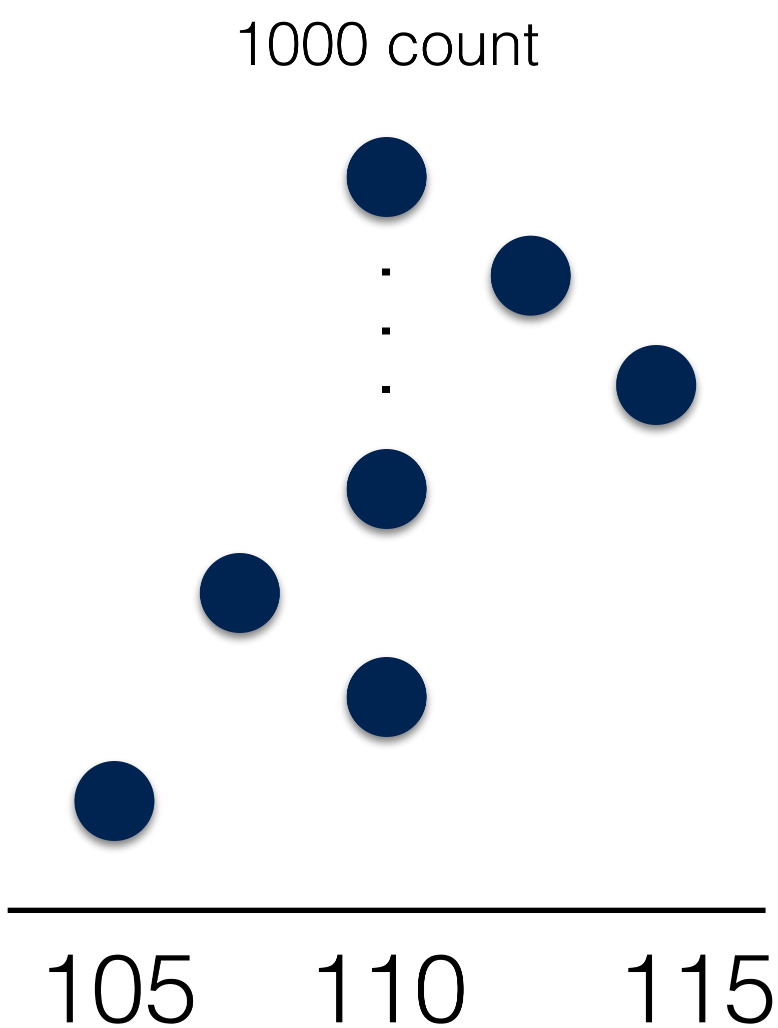
Generated c1=0.27, c2=0.827

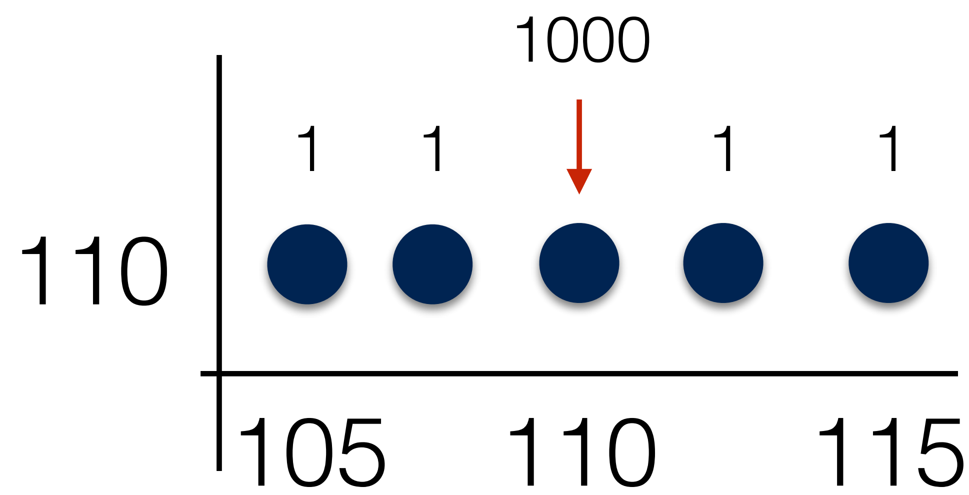




Data plotting

- Plotting packages do not represent the density of distribution well
- It can be misleading to interpret widths of any distribution based on scatter plot





RMS=3.5 **BY EYE**

RMS=0.25 **REAL**

Error Analysis

- Neural Networks work with nodes approximating their values with SIGMOID functions.
- The functions valid range is 0.0-1.0, that's why parameters have to be rescaled to this range
- Recovered values might seem to have large errors in the lower end especially (misleading)
- Physics distributions are sensitive to REAL values of these parameters

Example

parameter a: 7.0 - 12.0
scaled to 0.0-1.0 : a': min = 7.0, width = 5.0

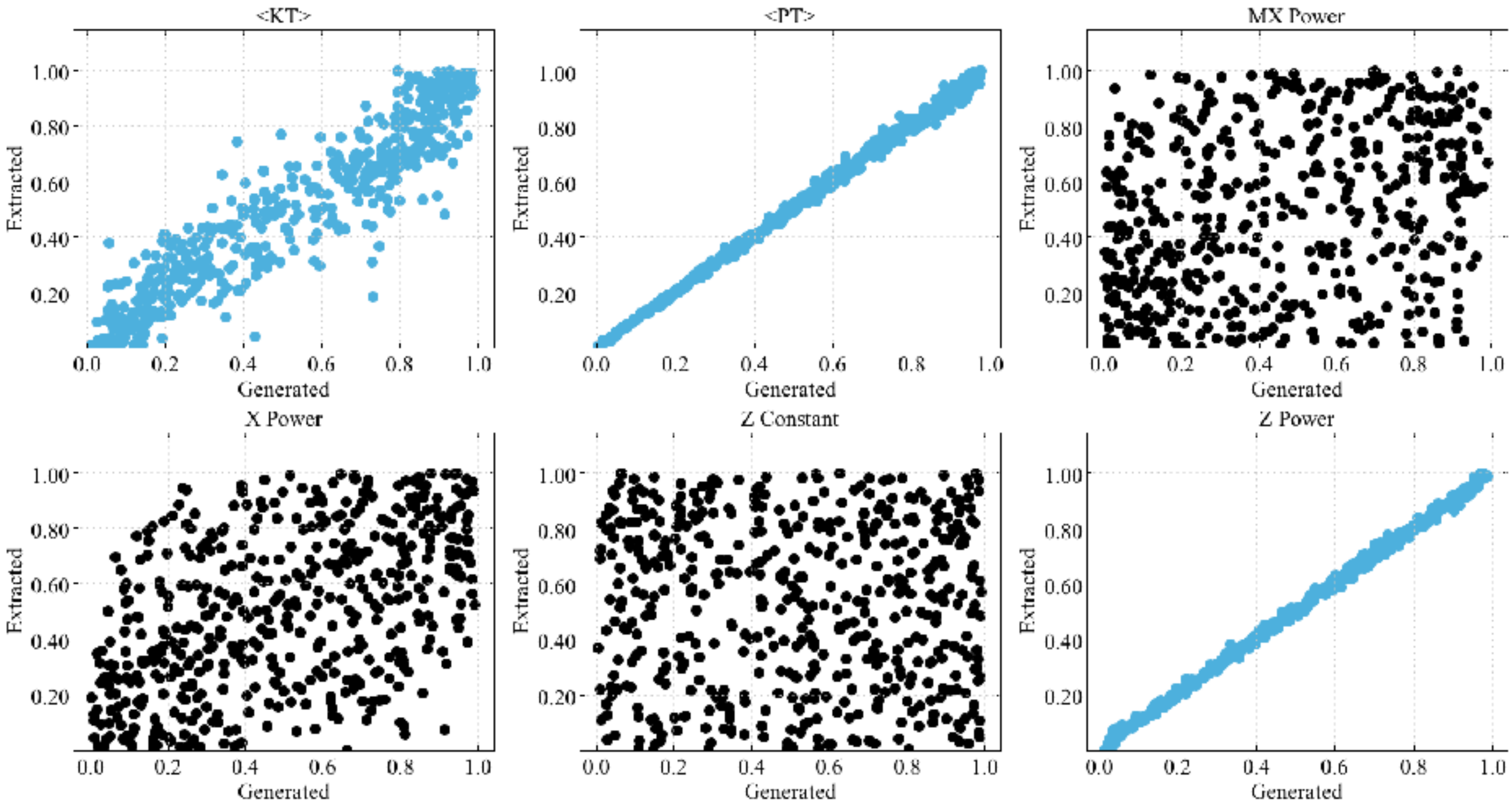
$$a' = \frac{a - 7.0}{5.0}$$

recovered NN result (a'=0.05, a' expected 0.03)
error = 0.66 (**66% error**)

Reality

a = 7.25,a expected 7.15, error = 0.013986 (**1.3%**)

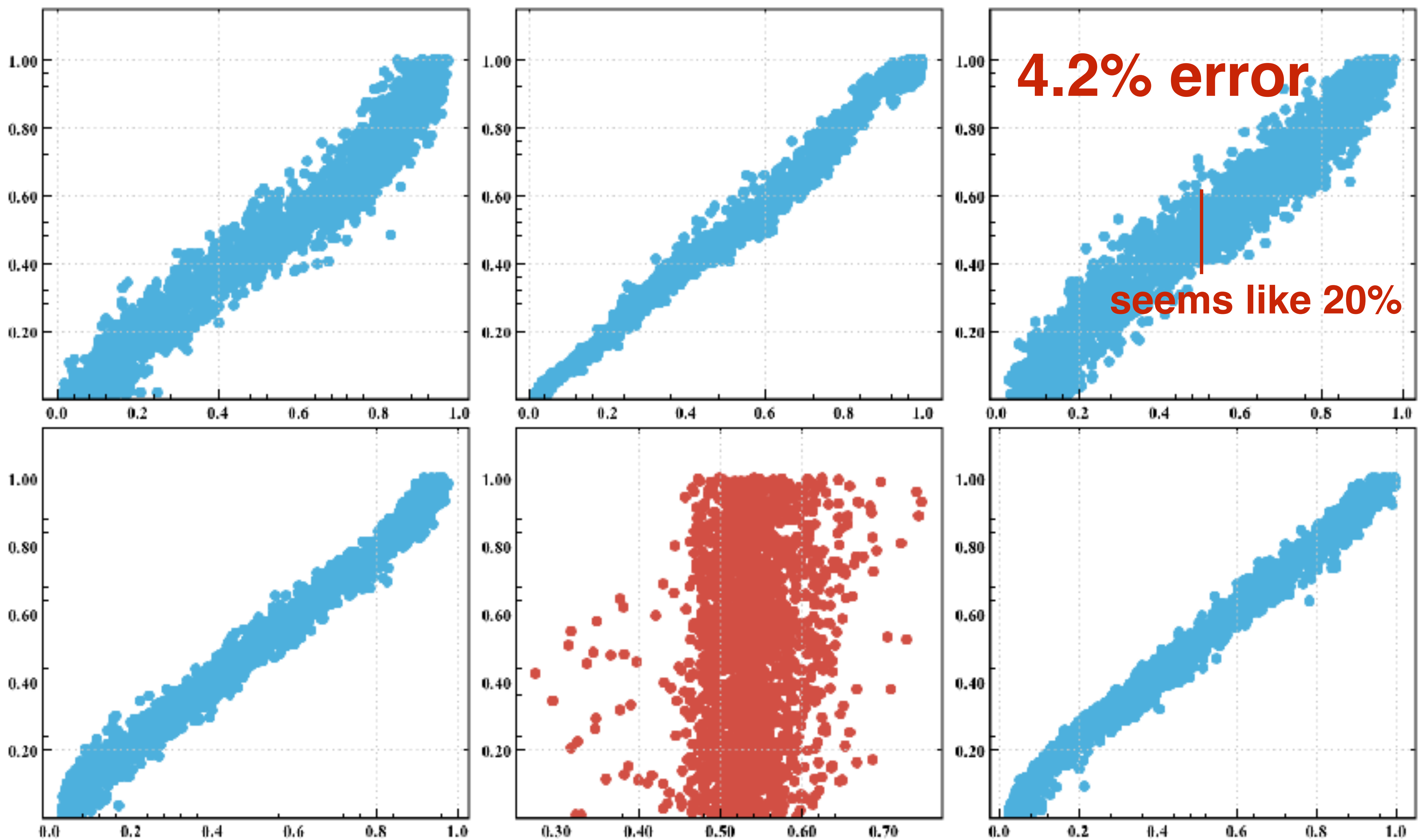
Results



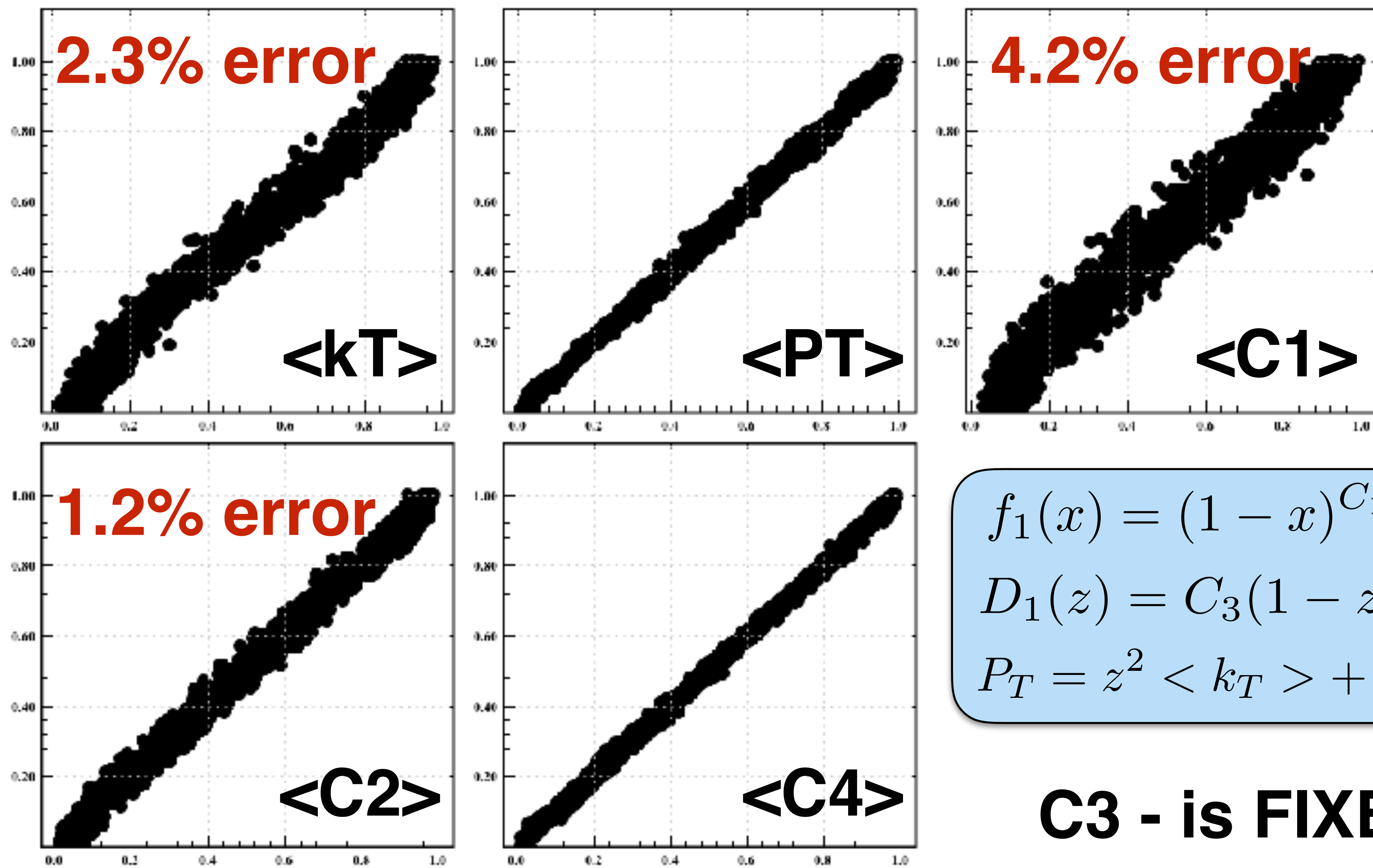
<KT>
<PT>
MX Power
X Power
Z Constant
Z Power
Extracted
Generated

Results

- Simulated with x {0.0-1.0}, pt {0.0-1.0}, z {0.0-1.0}
- Input data distribution x%z%pt (1000 nodes, 10x10x10)



Recent results



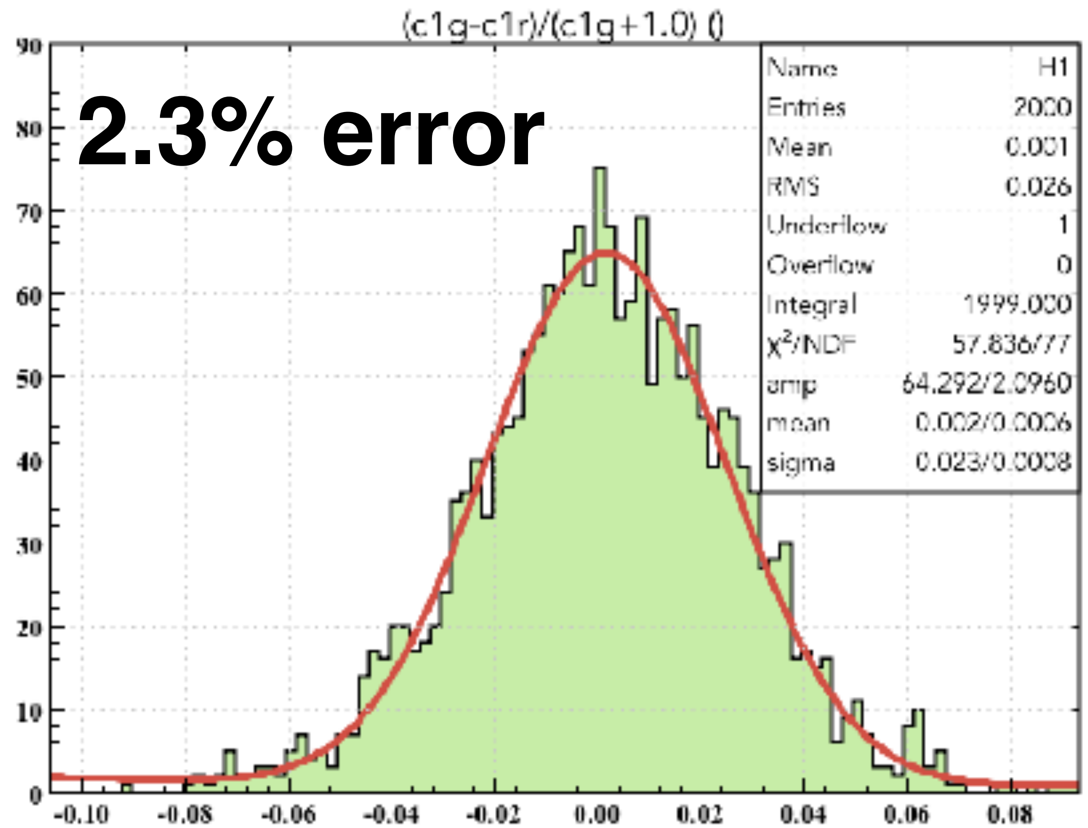
$$f_1(x) = (1-x)^{C_1} x^{C_2}$$

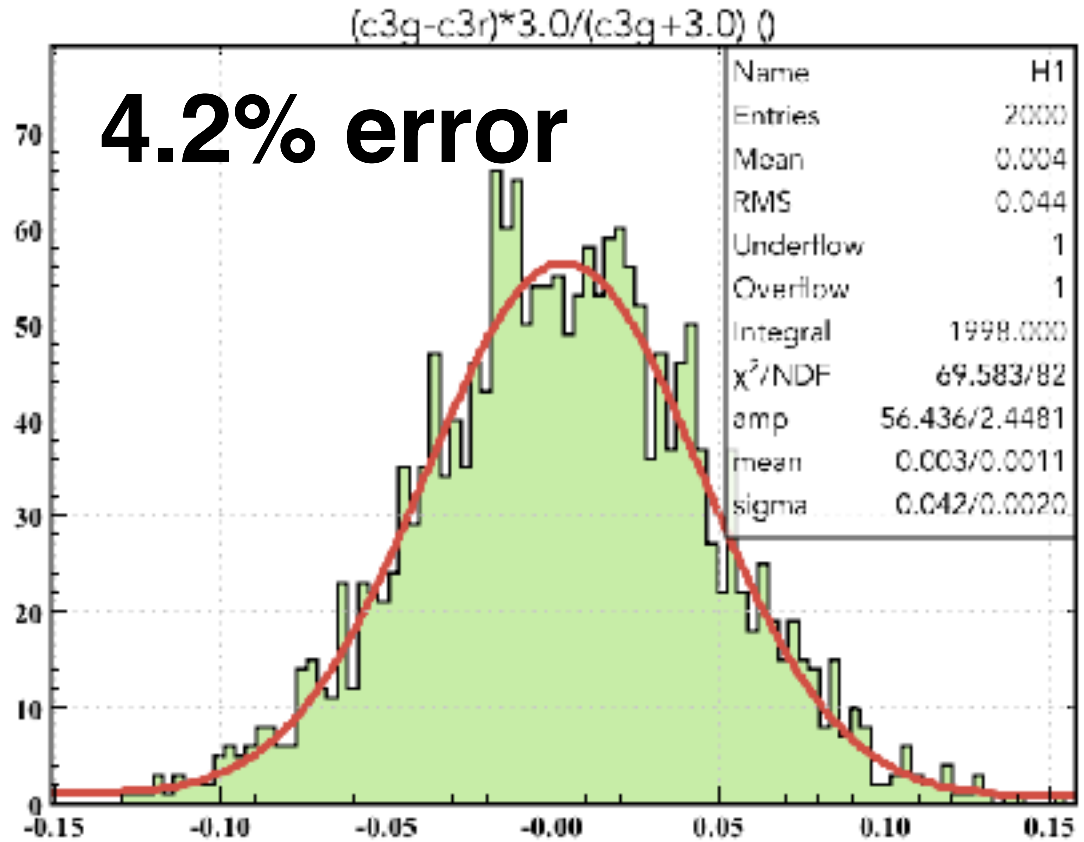
$$D_1(z) = C_3(1-z)^{C_4}$$

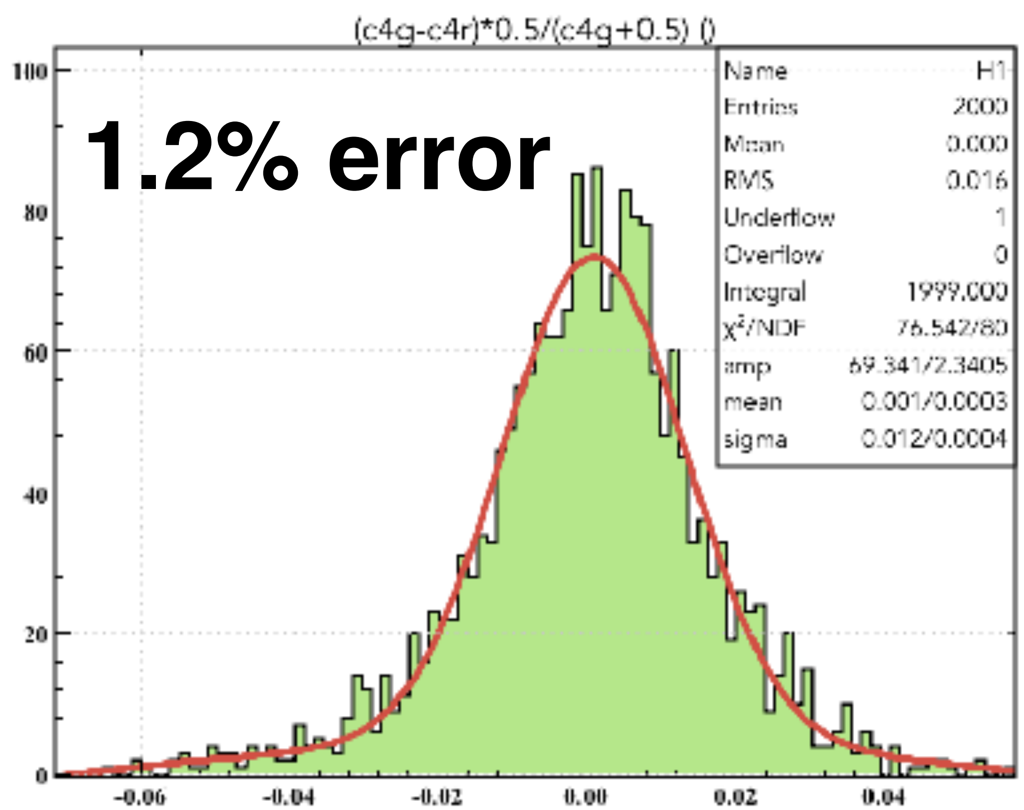
$$P_T = z^2 < k_T > + < P_T >$$

C3 - is FIXED

Uncertainty







Future work

- Implement CLAS12 detector FastMC into simulations
- Implement cosine structure functions
- Use z, pt and phi distribution to determine the parameters
- Use convolutional neural network to reduce dimensionality