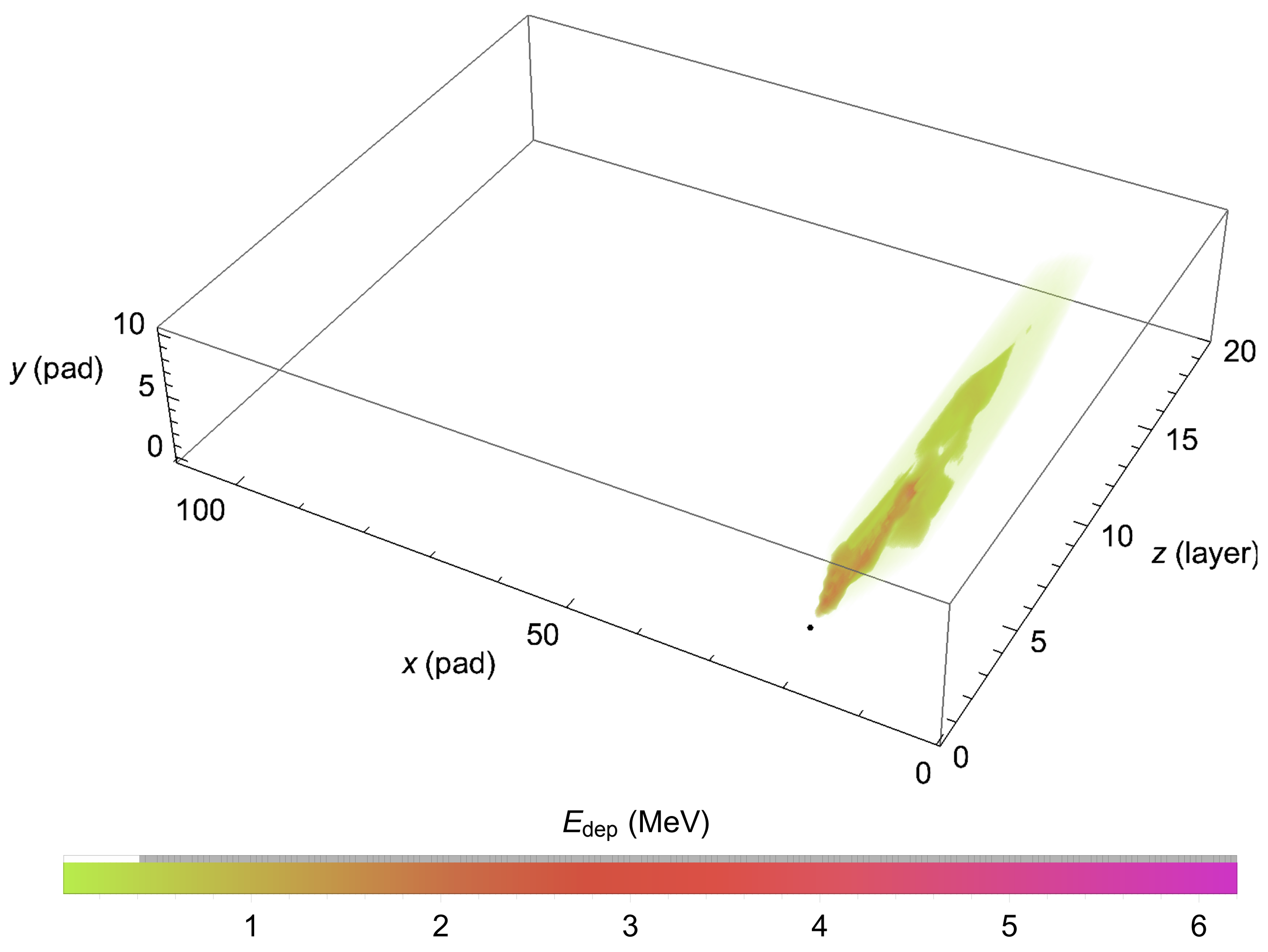
Neural Network
in ECAL Reconstruction
(status report)

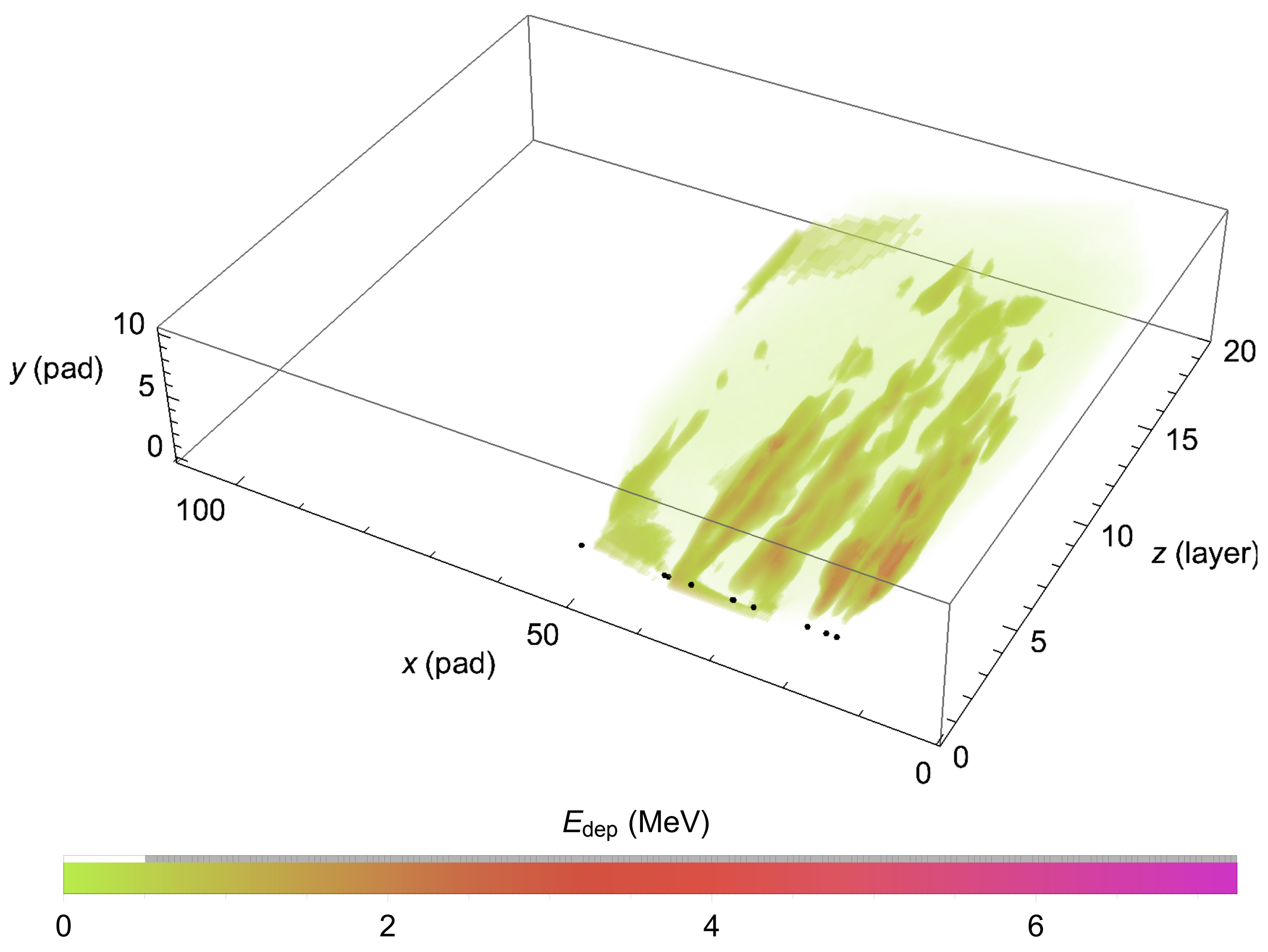
Shan Huang & Elihu Sela

4 August 2021

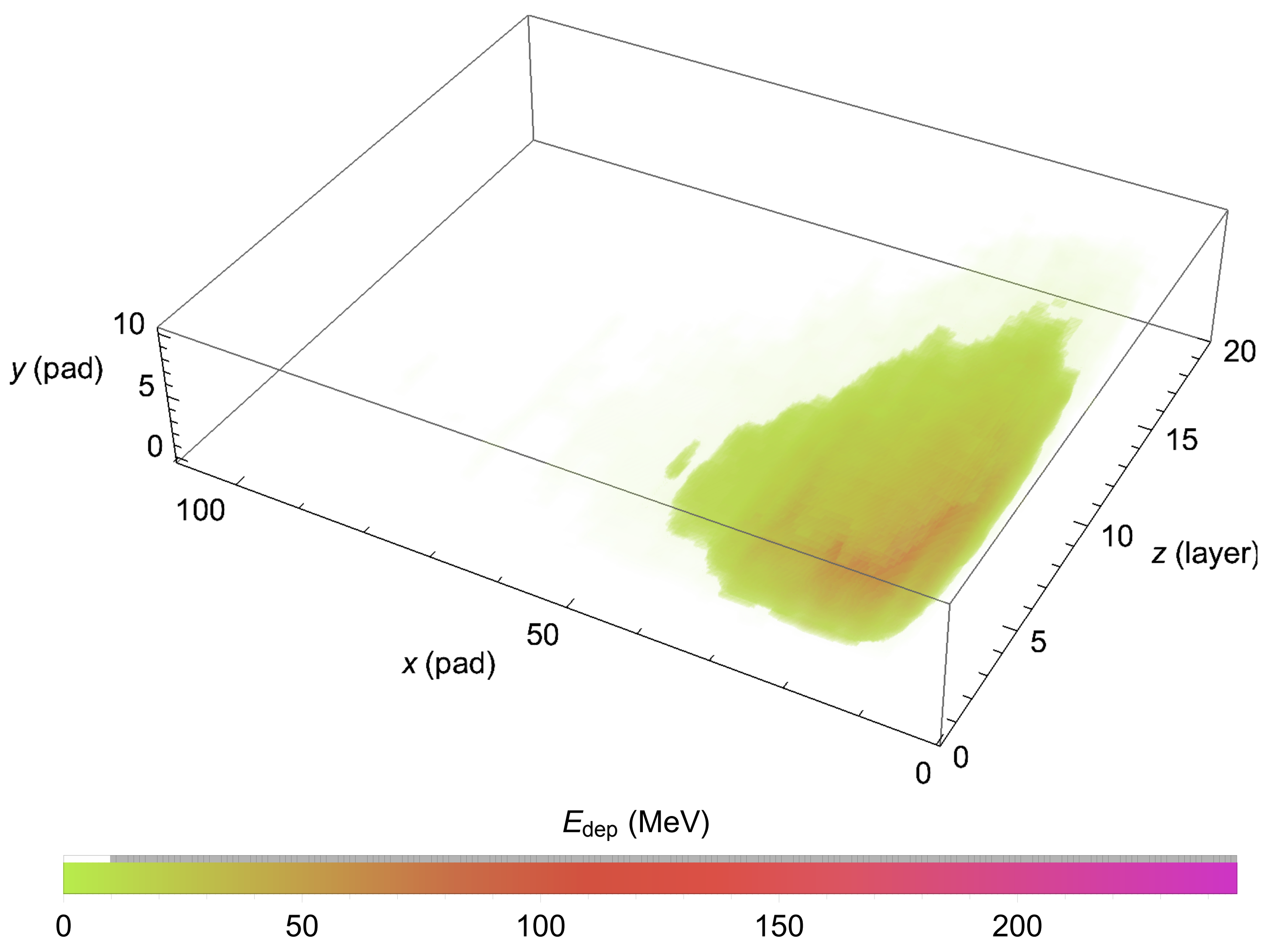
E_{dep} image in ECAL



Single-positron shower
"cold" ECAL



10-positron shower
"warm" ECAL



1000-positron shower
"hot" ECAL

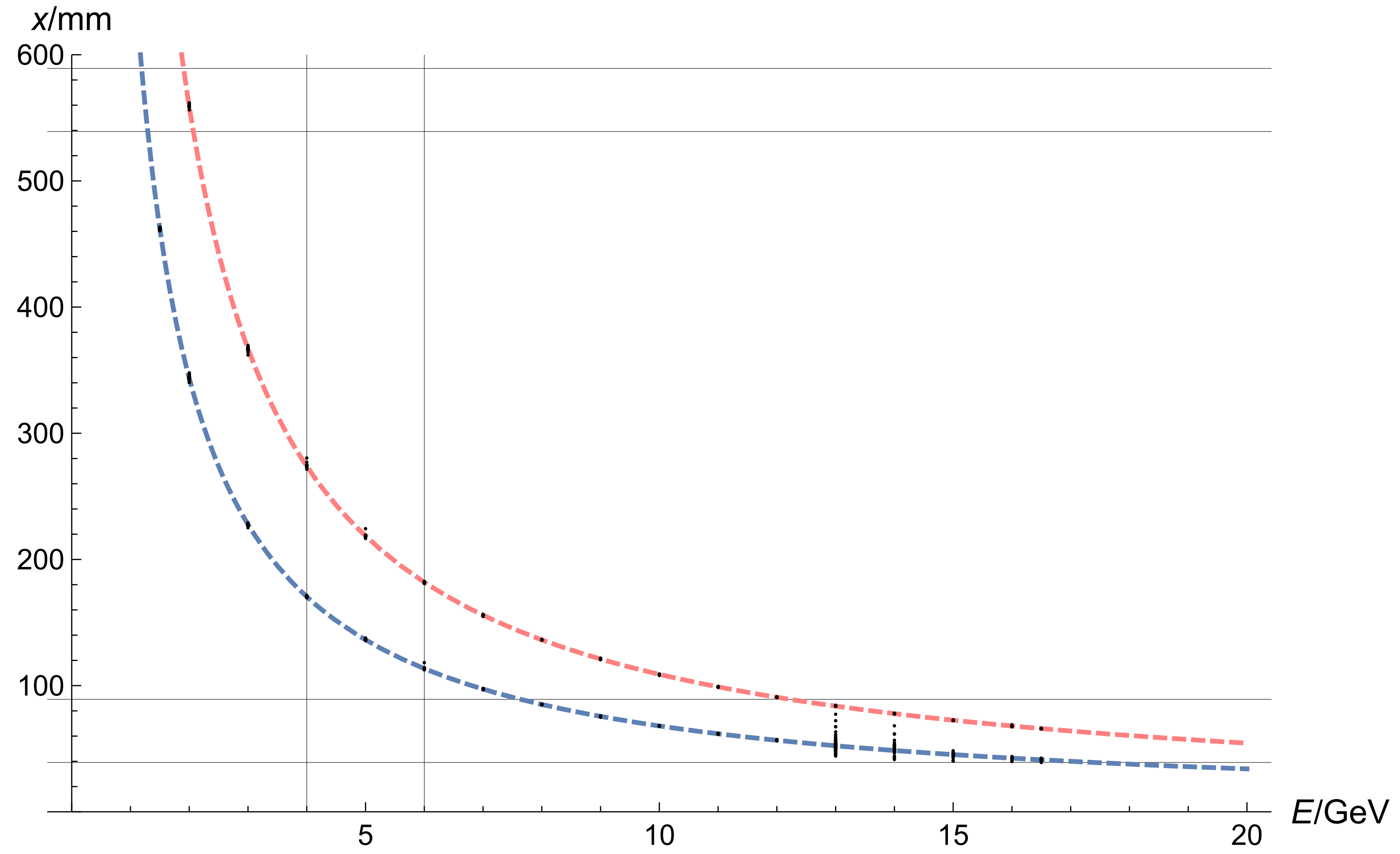
Questions to be answered:

- Where do the positrons enter ECAL? (*geometry un-related*)
- What are the energies of those positrons? (*depend on the geometry*)

for e-laser geometry (blue curve)

$$x * E_0 = 680.058 \text{ mm GeV}$$

$$- dx * E_0^2 = dE * 680.058 \text{ mm GeV}$$



Summary of Situation

"Cold" ECAL

- Dataset
  - ✓ positrons with given energies
  - ✓ positrons from signal files
  - no background
- Conventional methods
  - ✓ linear weighting
    (sigma = 0.1~1 mm/10~100 MeV)
  - ✓ logarithmic weighting
    (rms = 0.01~1 mm/10 MeV)
  - correction of angular effects
- Neural networks
  - "training separately" venue
    (rms = 0.001~0.1 mm)
  - mix training venue (rms = 450 MeV)

"Warm" ECAL

- Dataset
  - ✓ positron combinations from signal files
  - adding background
- Conventional methods
  - clustering (testing)
  - reconstruction for cluster
- Neural networks
  - training for energy list

"Hot" ECAL

- Dataset
  - ✓ 907 signal files
  - artificial generated from spectrums, plus BG
- Conventional methods
  - ✓ energy flow (max 10% error)
- Neural networks
  - training for energy histogram

COLD ECAL

Position Reconstruction: weighting methods

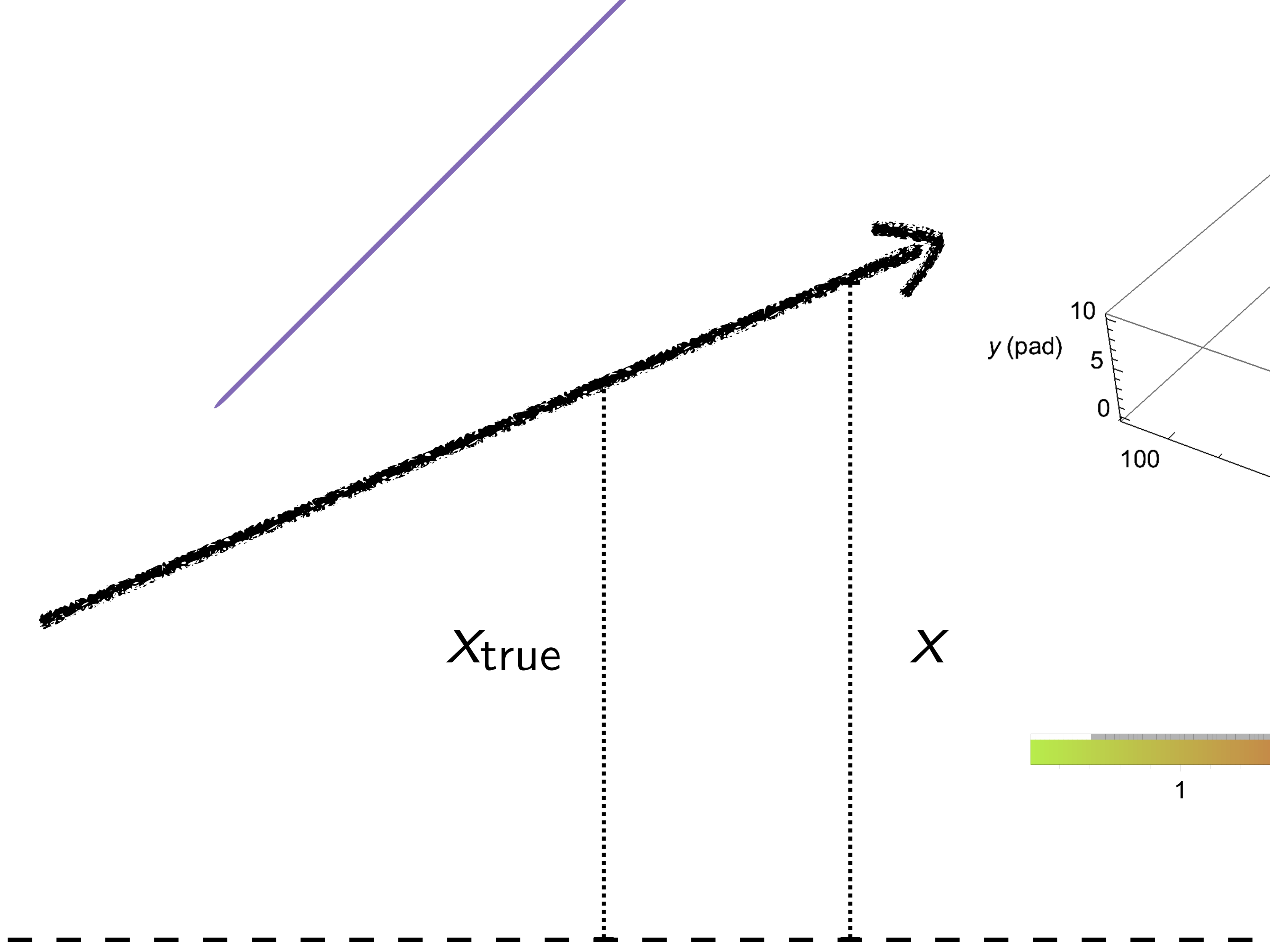
Give pads a E_{dep}-related weight

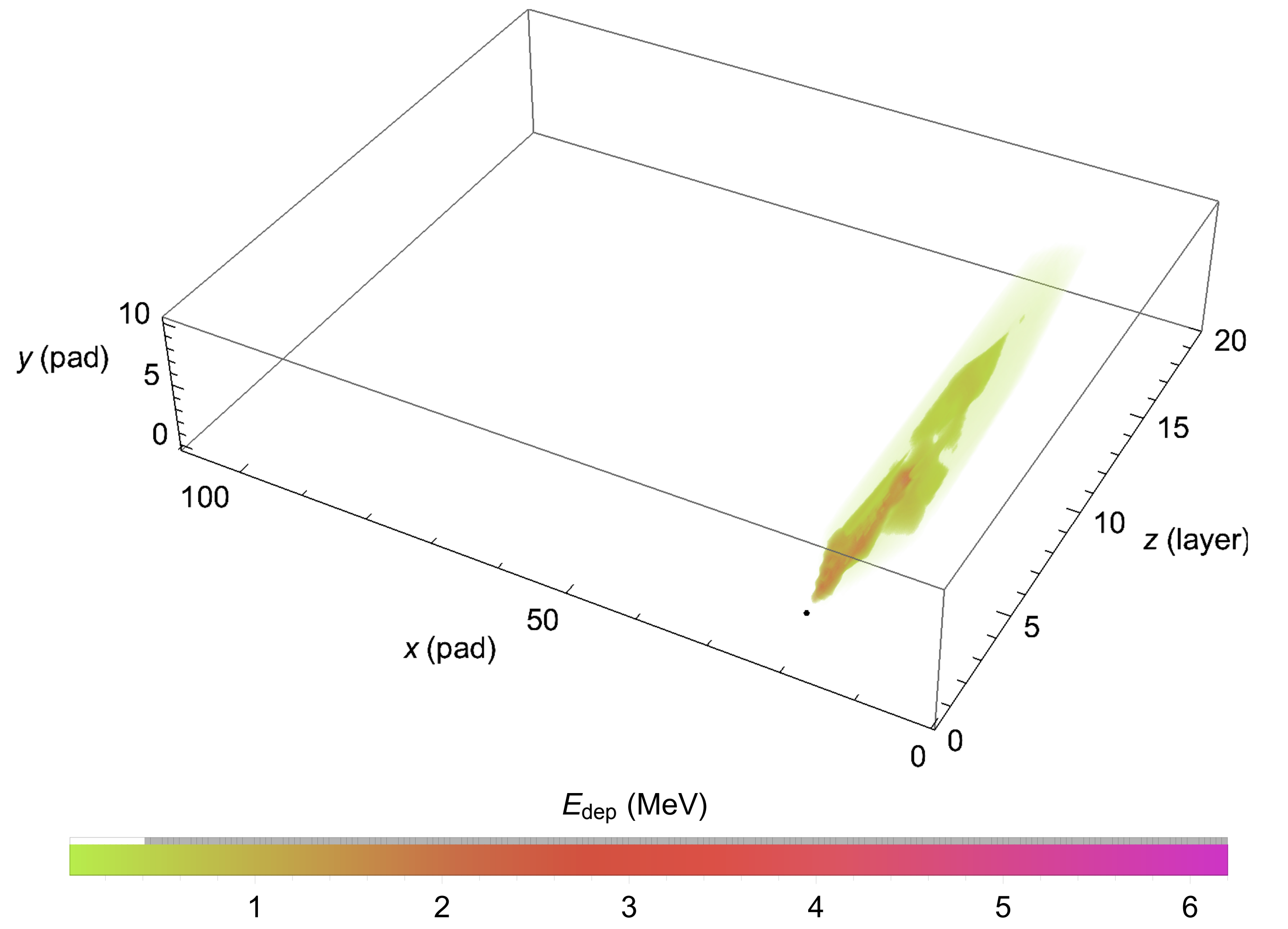
- Linear weighting method
  - weights are proportional to <u>E_{dep}</u>
  - not sensitive nor robust enough
- Logarithmic weighting method
  - weights are proportional to <u>$\log(E_{dep}/\text{threshold})$</u>
  - sensitive to pads with energy deposit near the threshold
  - immune from disturbance under the threshold

1 *As a result, with a well-chosen threshold, log-weighting method performs better*

- Positions in x-axis suffer from angular effects
- This effects should/can be eliminated with sophistic approaches (not-yet-all-found)

2 *Use y-coordinates as benchmarks for neural network methods*

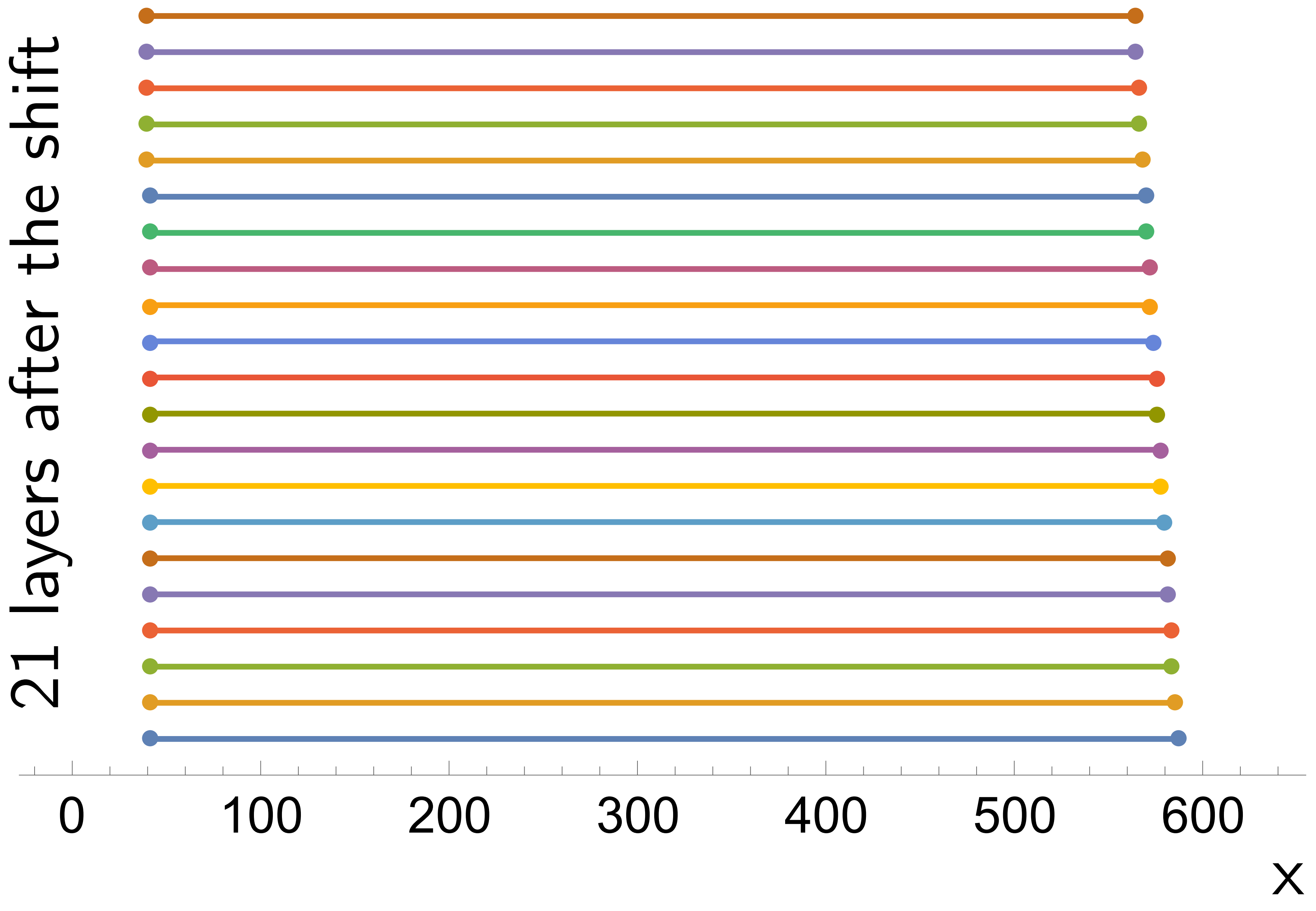


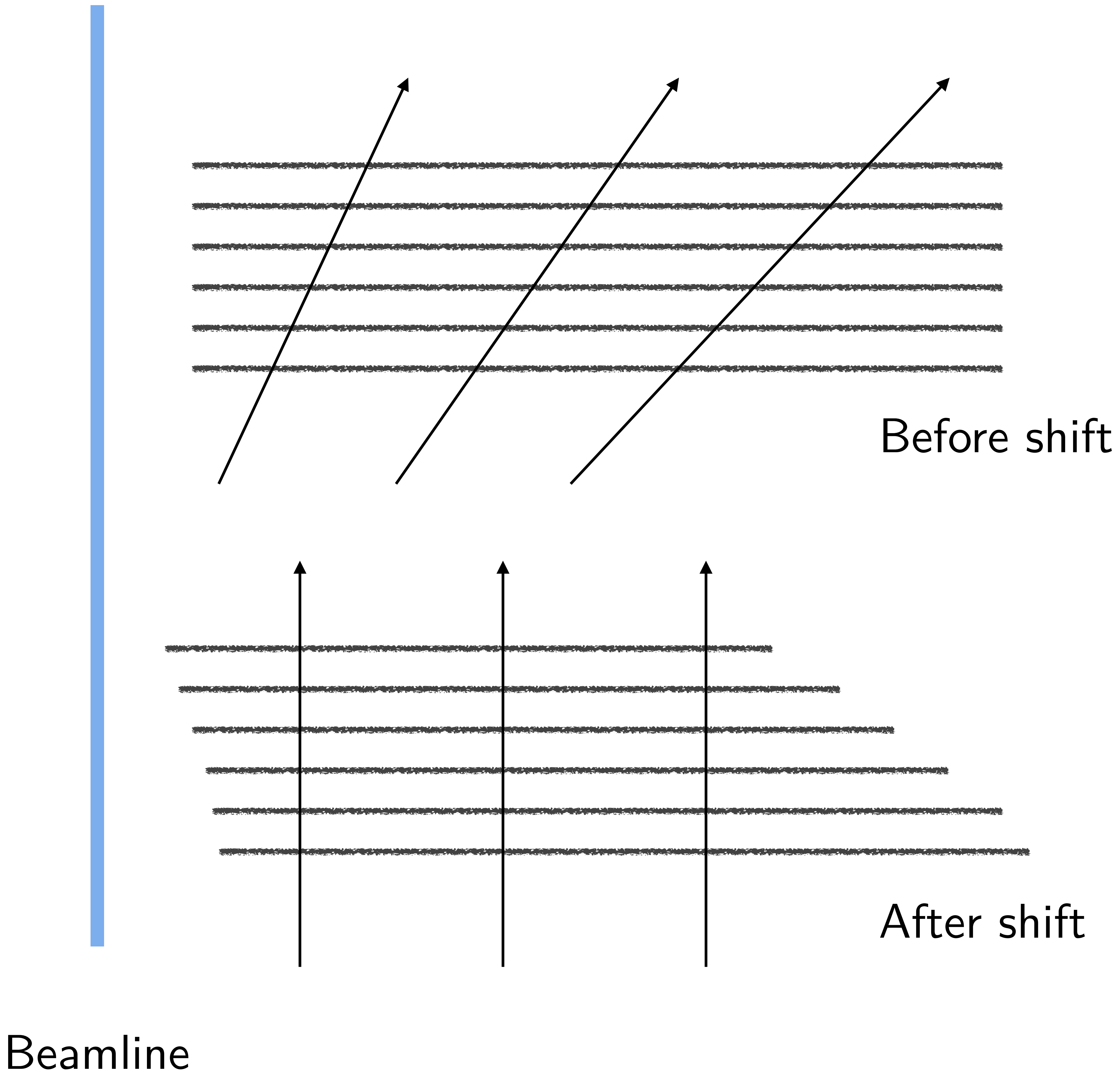


Position Recon.: correction on x coordinates

An example of the "approaches" to eliminate systematic bias by angular effects

- Layer-shifting to "straighten" the showers





Single shower: neural network training

- Basic idea:
  - treat E_{dep} distribution as 3D image
  - link the images with "true" values, including E_0 and x_{true}
- The definition of x_{true}
  i. ~~position of e^+ hit on ECAL~~
  ii. ~~position of the particle with highest energy after pre showering/scattering~~
  iii. position linked to e^+ energy
- Training targets
  i. the true position
  ii. positron's energy
  - different targets may bring different results, because of the E_0-x_{true} unlinearity

Available dataset

- Positrons with fixed energy
  - 130k showers
  - discrete from 2 to 14 GeV per 1 GeV
  - 10k per energy
- "Real-world" MC signal positron (907 files)
  - 914k showers
  - continuous in energy spectrum
  - uneven distribution follows the spectrum

Training venues

- Train on the discrete ones and test with the continuous ones
- Train on the continuous ones only

Single shower: NN training venues

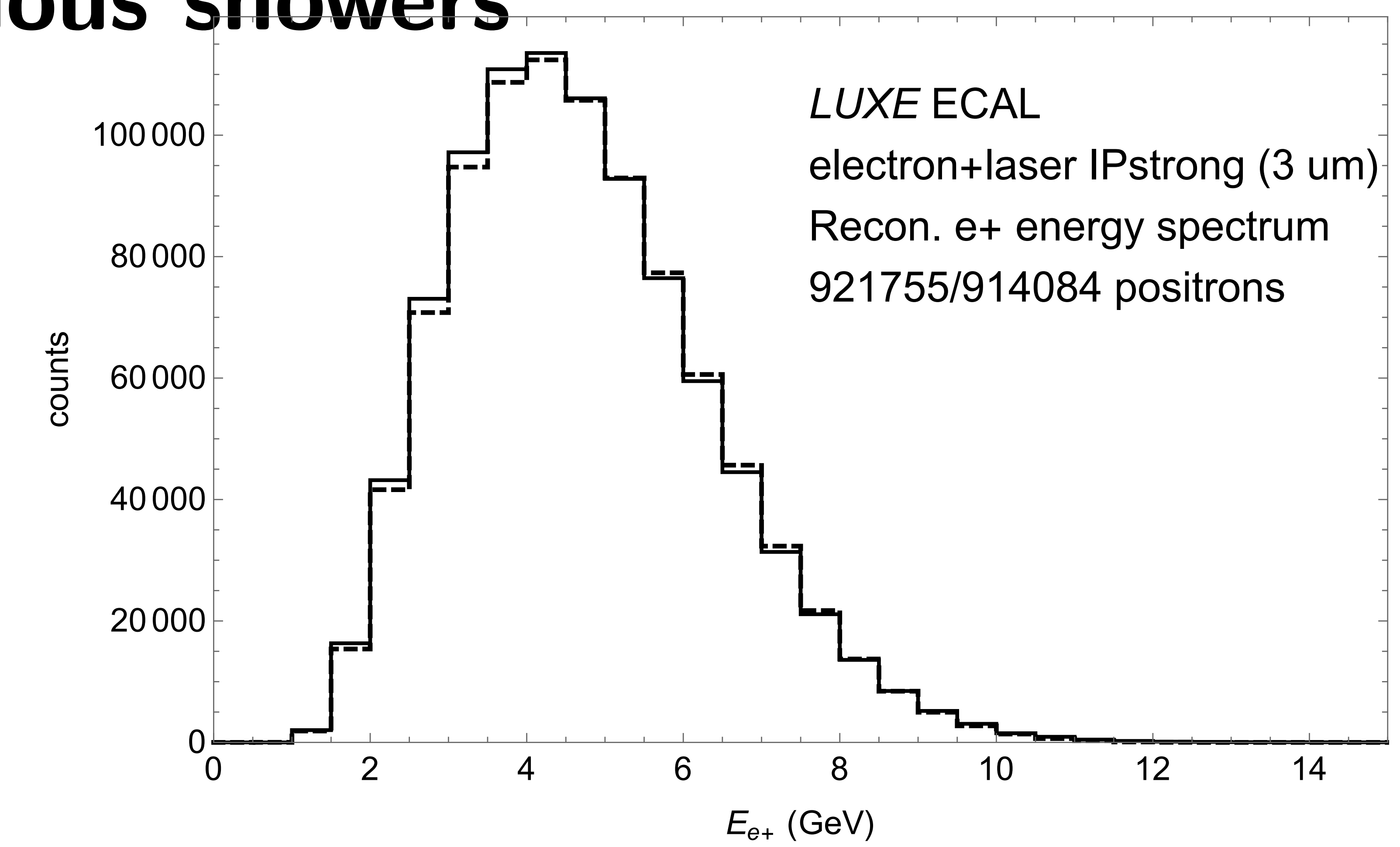
- The definition of x_{true} makes the discrete training venue "unreliable"
- Possible fix: to train all the 130k discrete showers together
  ~~"Training combining" strategy:~~
  ~~train each energy separately then combine their results~~

std = RMS of bias

ResNet10						
Gev	dX	dX_std	TrueX	dx/X	dY	dY_std
2	-1.235	0.032	343.592	-0.00359	0	0.002
3	-0.827	0.026	227.733	-0.00363	0	0.001
4	-0.649	0.02	170.46	-0.00381	0	0.001
5	-0.524	0.018	136.244	-0.00385	0	0.001
6	-0.443	0.014	113.481	-0.0039	0	0.001
7	-0.392	0.013	97.24	-0.00403	0	0.001
8	-0.334	0.014	84.07	-0.00397	0	0.001
9	-0.302	0.014	75.608	-0.00399	-0.001	0.001
10	-0.268	0.01	68.042	-0.00394	0	0.001
11	-0.249	0.014	61.852	-0.00403	0	0.001
12	-0.23	0.008	56.695	-0.00406	0	0.001
13	-0.208	0.009	52.332	-0.00397	0	0.001
14	-0.192	0.007	48.592	-0.00395	0	0.001

- The mixed training venue:
  - train with the 130k discrete showers plus some (6k) continuous showers
  - test with the continuous ones
- Result (energy training target):
  - average bias -16.9 MeV
  - RMS of bias 450 MeV
- **Possible fix: to train with more continuous showers**

energy spectrum
of 914k positrons



Single-positron shower recon.: benchmark

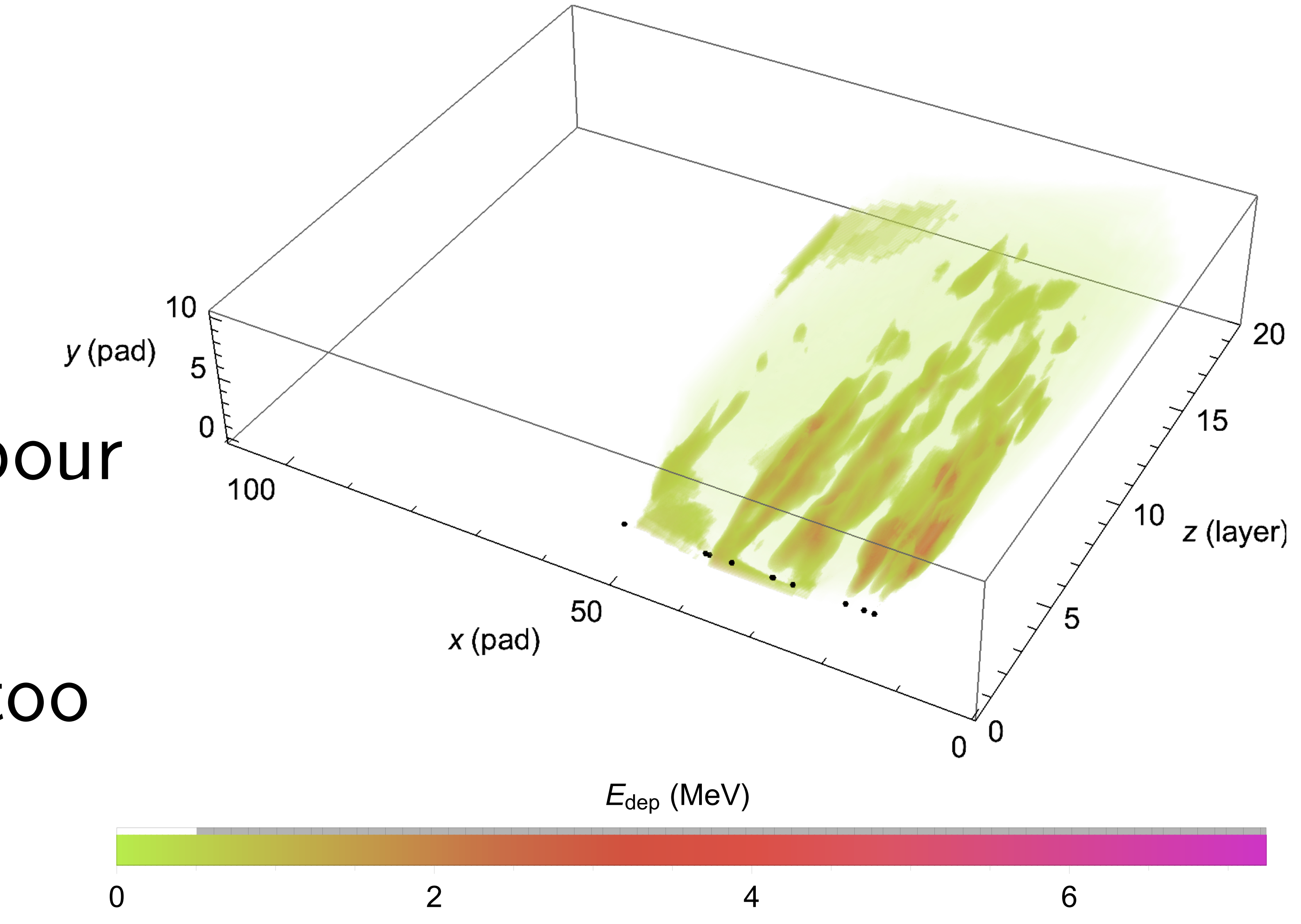
	energy (GeV)	2	3	4	5	6	7	8	9	10	11	12	13	14
Linear weight	**sigma_y (mm)**	0.942	0.764	0.652	0.594	0.538	0.504	0.461	0.438	0.423	0.398	0.375	0.361	0.355
	sigma_E (MeV)	5.541	10.11	15.34	21.84	28.48	36.31	43.38	52.17	62.20	70.81	79.40	89.71	102.3
Log weight (cut at 2.5%)	**rms_y**	0.896	0.538	0.300	0.189	0.130	0.087	0.054	0.037	0.035	0.022	0.016	0.013	0.012
	sigma_E	5.270	7.120	7.058	6.948	6.882	6.269	5.082	4.407	5.147	3.914	3.388	3.231	3.459
NN	**sigma_x**				0.0040	0.0030	sigma coming from gaussian fitting of bias_x/y instead of the root-mean-square							
	sigma_E				0.1470	0.1588								

WARM ECAL

Clustering: “the (local) highest wins all”

General idea

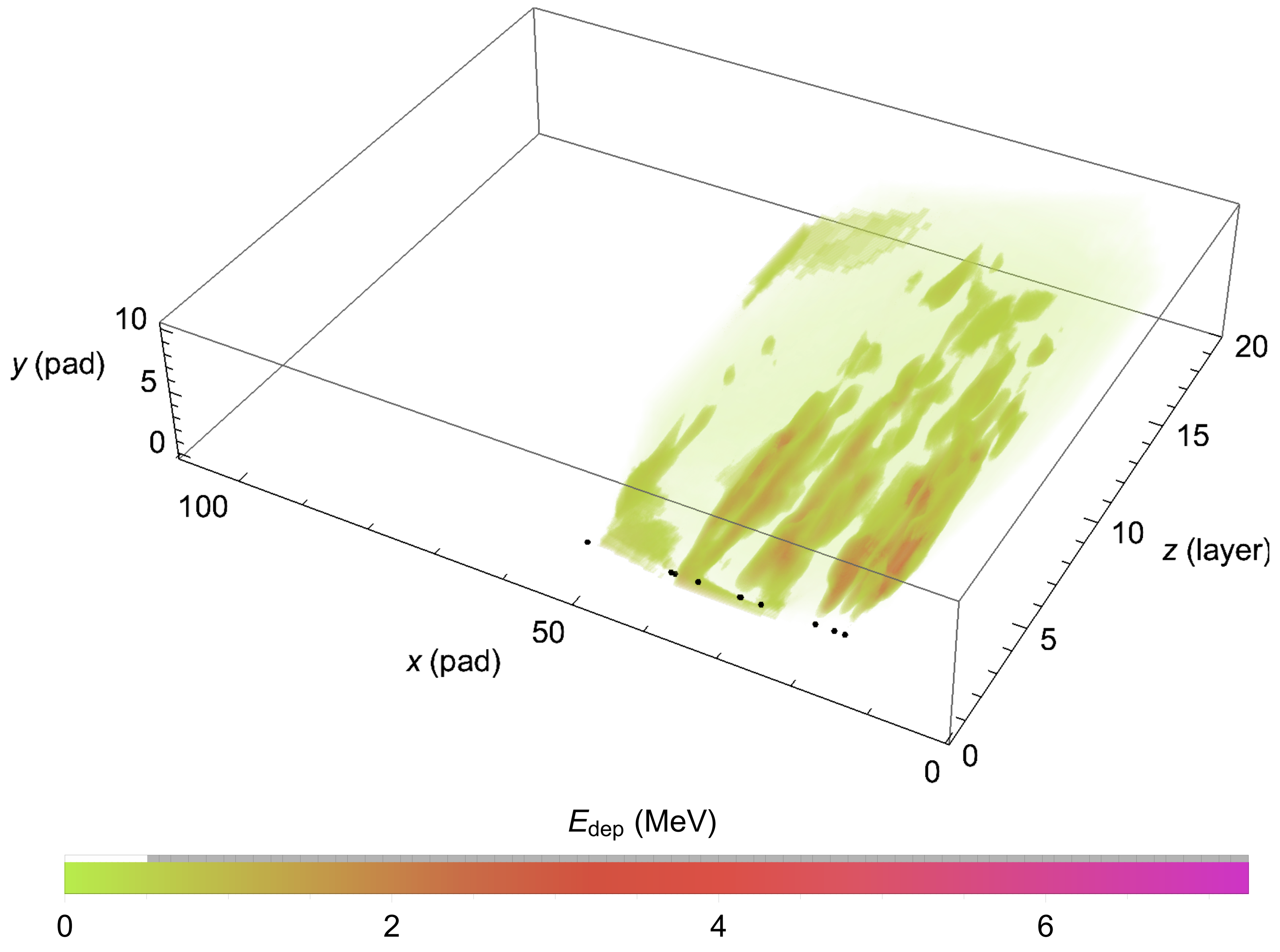
1. Clustering: Separate a big shower into small showers
   - each cluster “centred” around the local highest
   - absorb the clusters with low energy into their closest neighbour
2. Single shower methods
   - with proper logarithmic threshold, clusters will not change too much from a “real” shower
   - ***will the NN still work for clusters?***



TODO

- clustering code
- resolution test for clusters

NN for a few positrons



{Ea, Eb, Ec, Ed, …}

General idea

- The list of positron energies as training target
  - length? (10 for now)

Dataset generation

- Randomly pick upto 10 positrons from the general dataset
- Mix them with background (creating using the same method)
  - BG dataset: 45k files, need 150 files for a BG

TODO

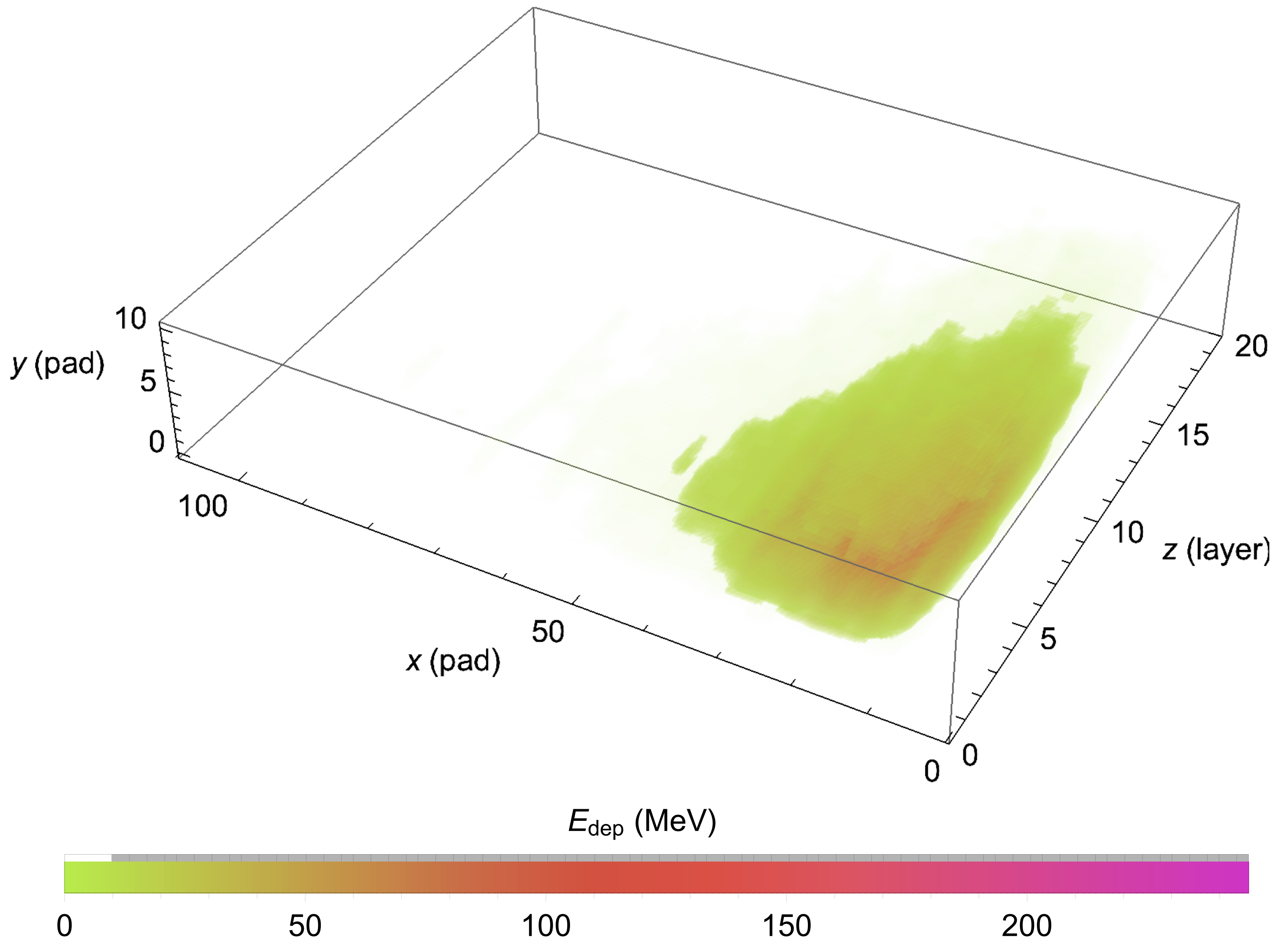
- generate dataset with background
- train the NN

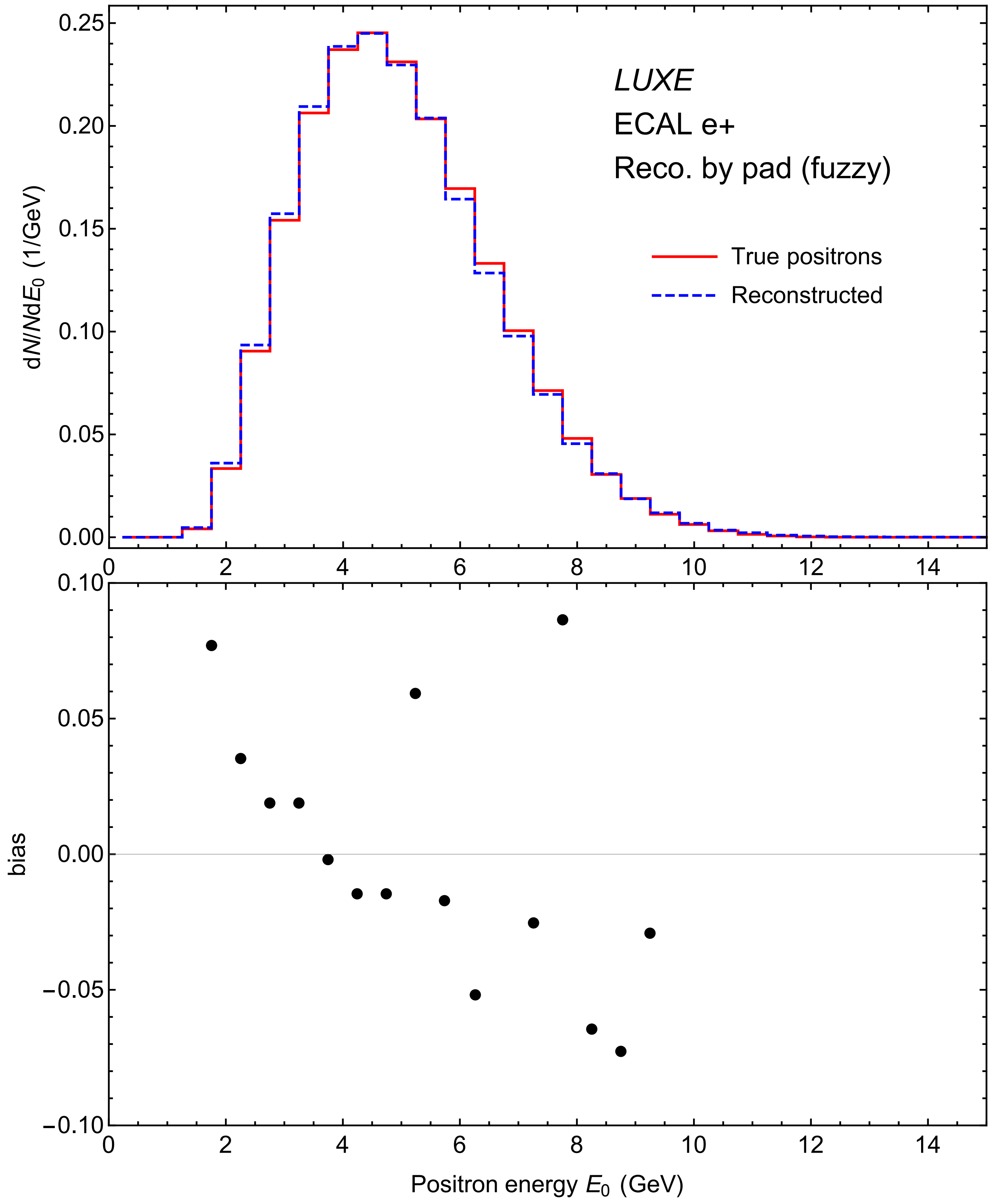
HOT ECAL

Energy flow

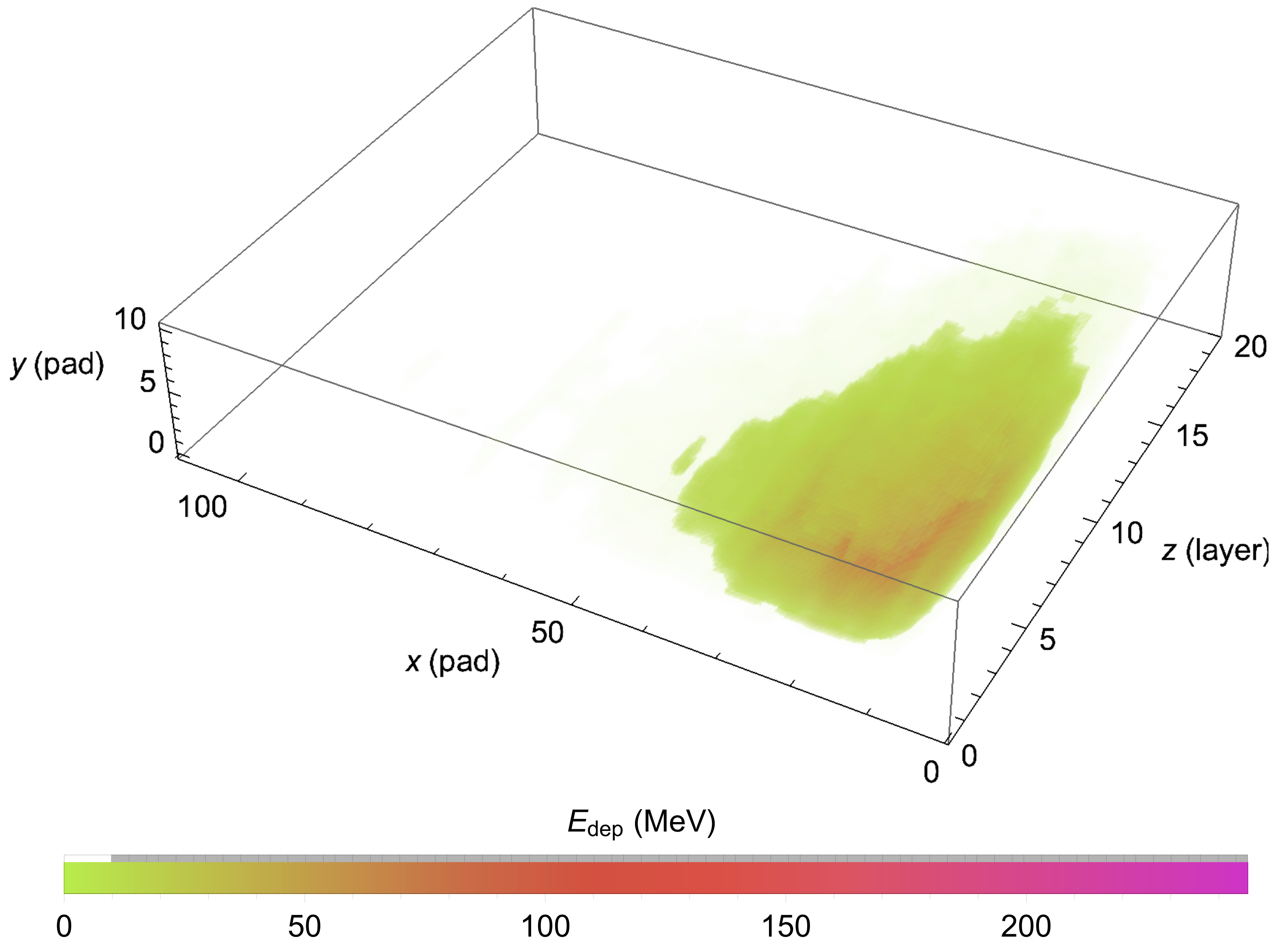
General idea

1. Convert E_{dep} to E_{dep} density
   - assuming in a pad, energy deposited averagely
   - obtain a function of E_{dep} density over e^+ energy (the “truth”)
   - use E_{dep} to estimate the corresponding e^+ number
   - get a rough energy spectrum function
2. Binning
   - to remove the roughness brought by ECAL pad
   - to make the result comparable to other methods
   - binning range: 0 to 15 GeV, every 0.2 or 0.5 GeV
   - result: less than 10% error





NN for a few positrons



General idea

- The histogram of positron energies as training target
  - binning range: 0 to 15 GeV
  - bin size? (0.5 GeV for now, testing for 0.2 GeV bin)

$\{N_{bin1}, N_{bin2}, N_{bin3}, N_{bin4}, \ldots\}$

Dataset generation

- 907 signal files (expanding)
- Use (1) energy spectrum and (2) multiplicity distribution to generate artificial dataset, based on inverse CDF
  1. number of positron in a shower
  2. positron's energies
  3. picking-up from main dataset
  4. combining into one shower

TODO

- generate dataset with background
- generate artificial dataset
- train the NN
- estimate NN's uncertainty

BACKUP

Single-positron shower recon.: benchmark

	energy (GeV)	2	3	4	5	6	7	8	9	10	11	12	13	14
Linear weight	sigma_x (mm)	1.134	0.931	0.783	0.697	0.679	0.607	0.535	0.495	0.508	0.437	0.398	0.387	0.409
	sigma_E (MeV)	6.670	12.32	18.42	25.62	35.94	43.74	50.35	58.96	74.70	77.75	84.28	96.17	117.9
Log weight (cut at 2.5%)	sigma_x	0.892	0.691	0.375	0.337	0.438	0.447	0.161	0.136	0.693	0.116	0.101	0.110	0.968
	sigma_E	5.247	9.145	8.823	12.389	23.186	32.208	15.152	16.199	101.903	20.639	21.386	27.336	278.988
NN	sigma_x				0.0040	0.0030								
	sigma_E				0.1470	0.1588								

Position recon.: x-direction

	energy (GeV)	2	3	4	5	6	7	8	9	10	11	12	13	14
Linear weight	**bias_x (mm)**	0.733	0.531	0.234	0.067	0.446	-0.013	0.386	0.272	0.119	0.097	0.237	0.262	0.770
	sigma_x	1.134	0.931	0.783	0.697	0.679	0.607	0.535	0.495	0.508	0.437	0.398	0.387	0.409
Log weight (cut at 2.5%)	**bias_x**	0.787	0.785	-0.095	-0.514	1.224	-0.859	0.587	0.156	-0.564	-0.881	-0.713	-1.210	0.480
	sigma_x	0.892	0.691	0.375	0.337	0.438	0.447	0.161	0.136	0.693	0.116	0.101	0.110	0.968
NN	**bias_x**				-0.007	-0.023								
	sigma_x				0.004	0.003								

Position recon.: y-direction

	energy (GeV)	2	3	4	5	6	7	8	9	10	11	12	13	14
Linear weight	**bias_y (mm)**	0.017	-0.006	0.000	0.009	0.007	0.004	-0.003	0.000	0.009	0.002	0.003	0.002	0.005
	sigma_y	0.942	0.764	0.652	0.594	0.538	0.504	0.461	0.438	0.423	0.398	0.375	0.361	0.355
Log weight (cut at 2.5%)	**bias_y**	0.017	-0.010	-0.002	0.003	0.000	0.002	-0.001	0.000	0.000	0.000	0.000	0.000	0.000
	rms_y	0.896	0.538	0.300	0.189	0.130	0.087	0.054	0.037	0.035	0.022	0.016	0.013	0.012
NN	**bias_y**				0.0001	-0.0010								
	sigma_y				10^{-6}	10^{-5}								