European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world

GENERALITAT VALENCIANA

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA, INDUSTRIA Y COMPETITIVIDAD

UNIÓN EUROPEA
Fondo Europeo de Desarrollo Regional (FEDER)
Una manera de hacer Europa

EXCELENCIA MARÍA DE MAEZTU
EXCELENCIA SEVERO OCHOA
CSIC
Consejo Superior de Investigaciones Científicas

Machine Learning in PET Imaging with

Reconstruction and Machine Learning in Neutrino Experiments

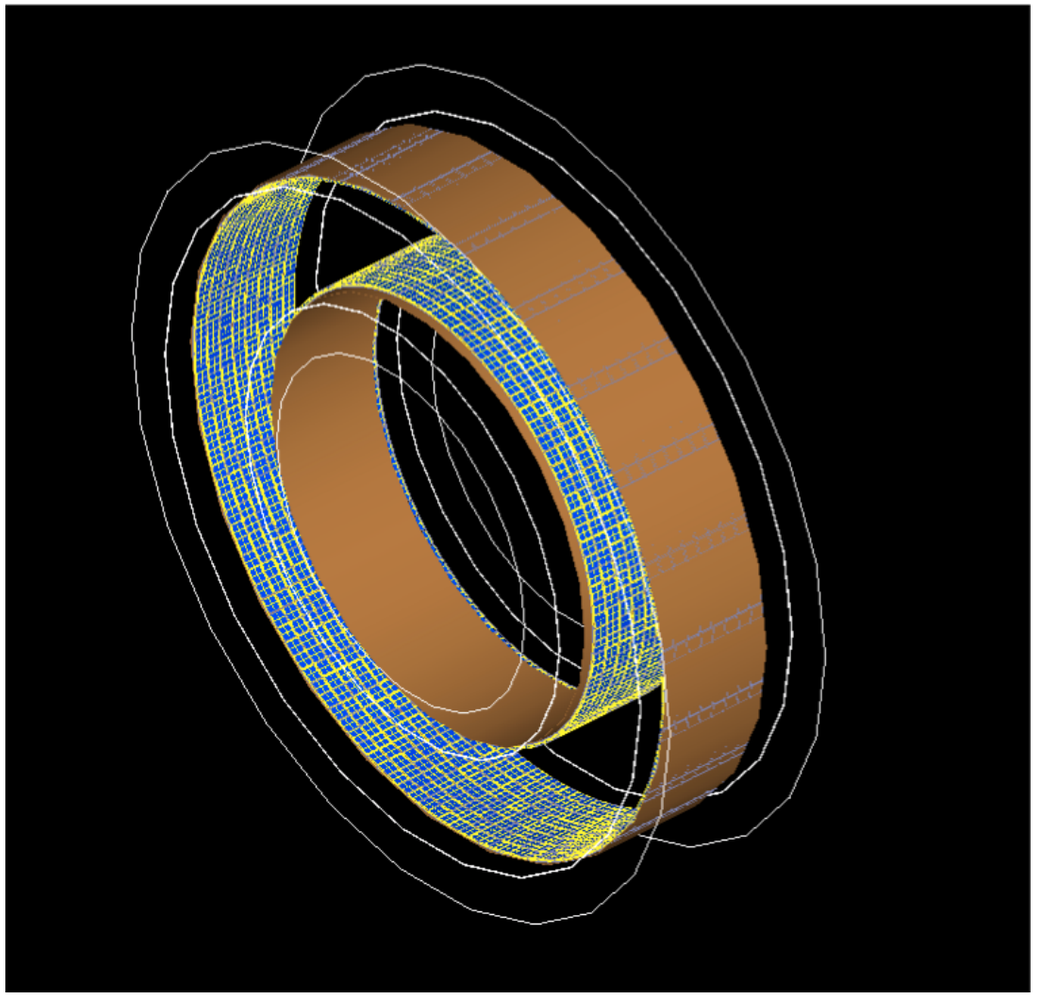
DESY Hamburg
September 19, 2019

J.M. Benlloch-Rodríguez[1], J.V. Carrión[1], P. Ferrario[1,2], R. Gadea[3], J.J. Gómez-Cadenas[2], V. Herrero-Bosch[3], M. Kekic[1], **J. Renner[1,4]**, C. Romo-Luque[1]
(PETALO Collaboration)

[1]*IFIC/Universitat de València*
[2]*DIPC/Ikerbasque*
[3]*Universitat Politècnica de València*
[4]*IFGAE / Universidade de Santiago de Compostela*

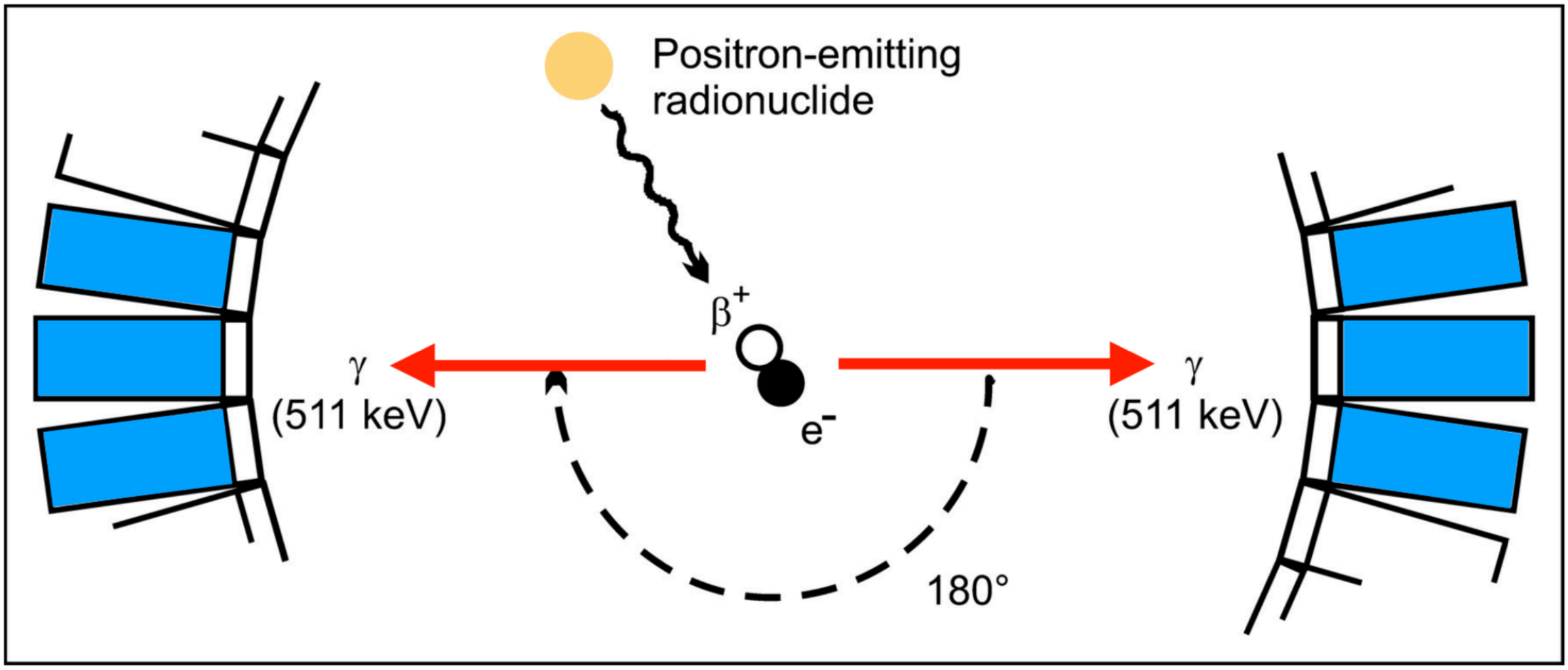
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
VNIVERSITAT DE VALÈNCIA
IFIC INSTITUT DE FÍSICA CORPUSCULAR
ikerbasque Basque Foundation for Science
DIPC
IGFAE Instituto Galego de Física de Altas Enerxías
USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Positron **E**mission **T**ime of flight **A**pparatus based on **L**iquid xen**O**n



- PET imager using liquid xenon (LXe) as detection medium
- Currently in construction phase
- Xenon offers fast scintillation (potential to use time-of-flight) and high light yield
- Ring-shaped homogeneous volume with one or two faces instrumented with SiPMs sensitive to xenon scintillation (VUV)

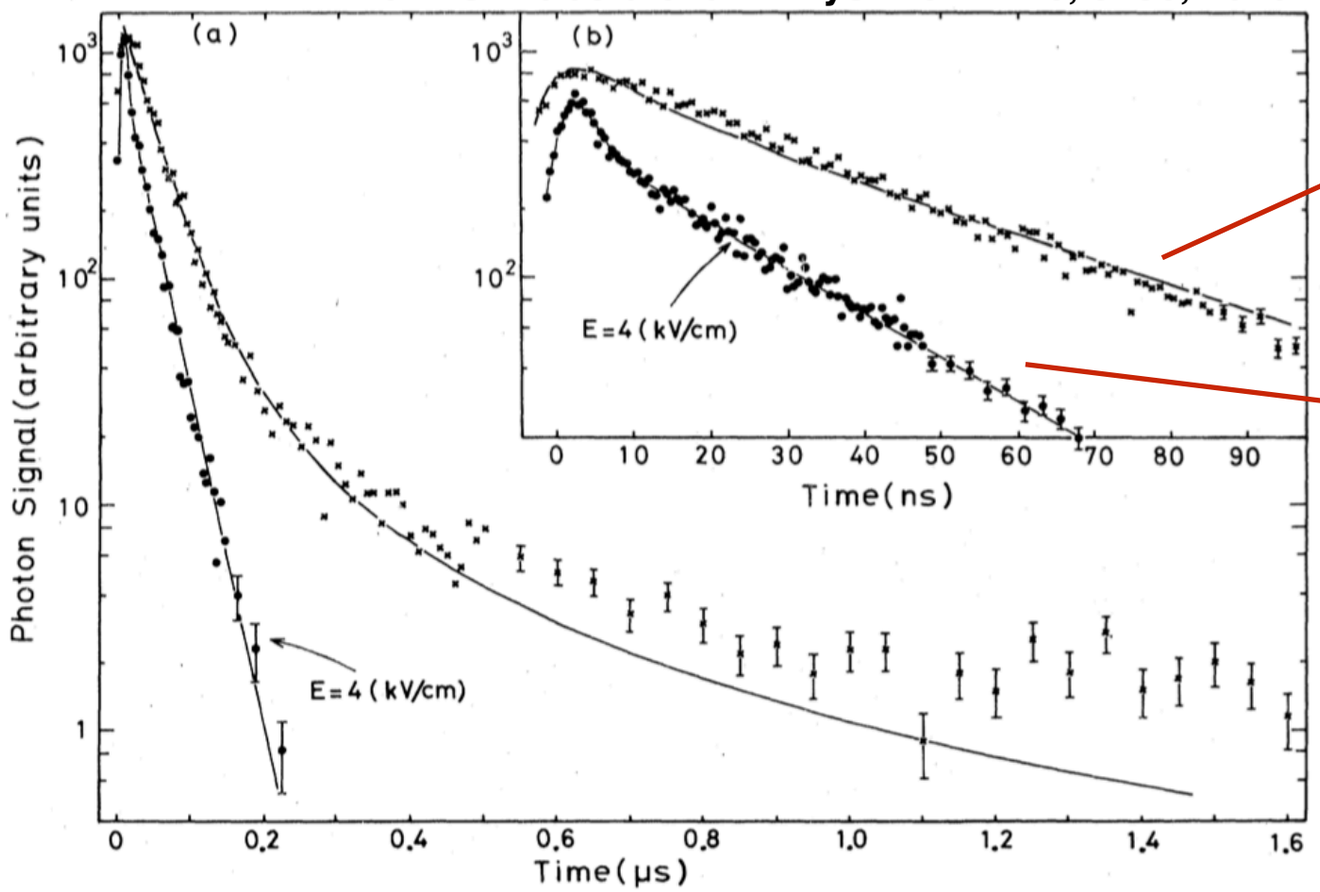
PET imaging



- Glucose analogue (such as ^{18}F-FDG) injected into patient, doped with radioactive isotope (such as ^{18}F) yielding positrons; metabolized by cells and accumulates
- Positron annihilation gives two collinear 511 keV gammas
- These gammas detected and the line along which they were emitted (the line of response, LOR) is reconstructed; full image deduced from many LORs

Why LXe?

From: **S. Kubota et al. Phys. Rev. B 20, 3486, 1979.**



Primary scintillation + recombination

Recombination quenched via application of an electric field

PETALO will see all scintillation (no applied electric field - ionization will not be read out)

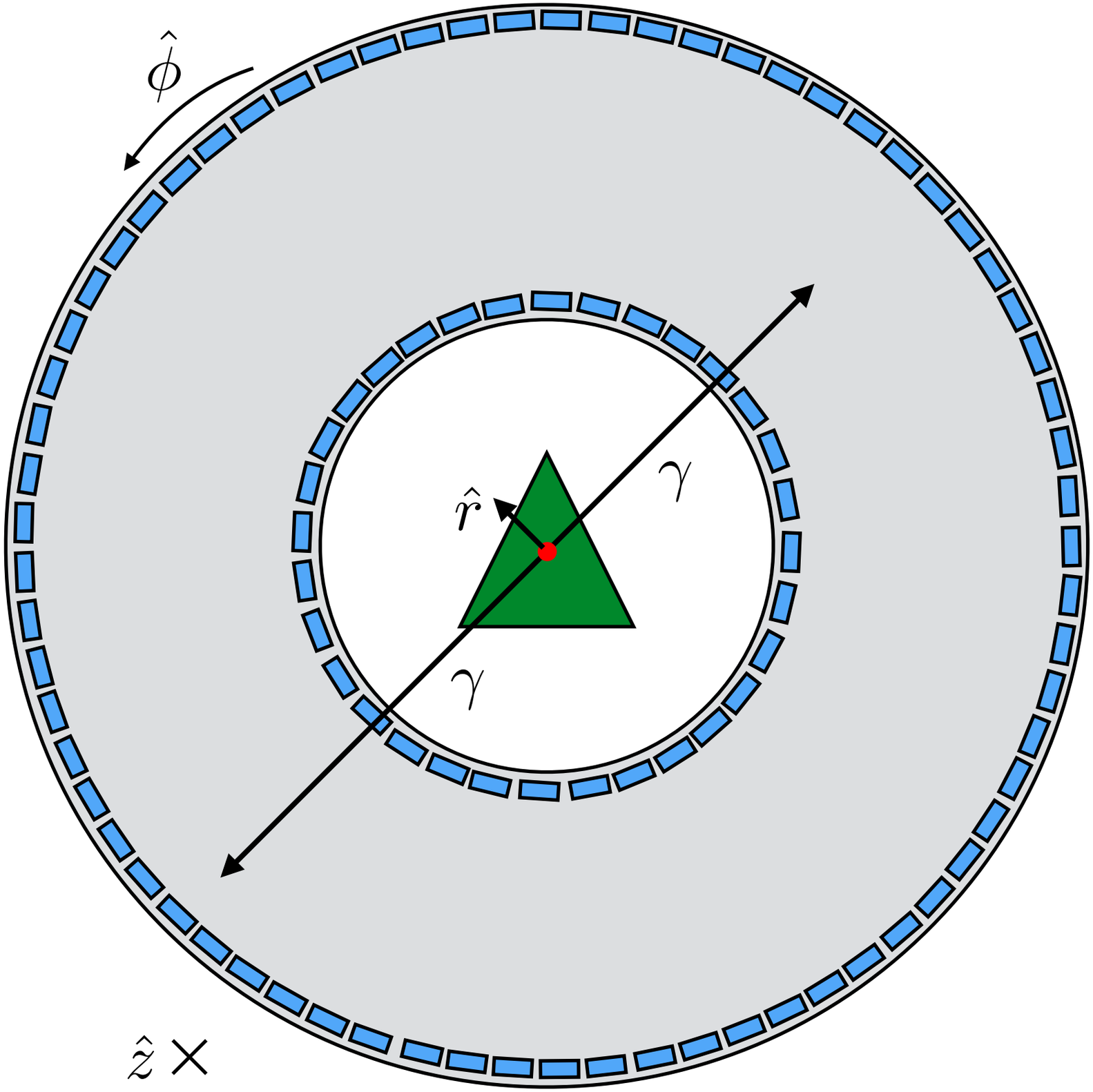
- **Fast scintillation**:
  - molecular state responsible for fast-transition has lifetime of 2.2 ns
  - potential for time-of-flight (TOF) measurements
- **High scintillation yield:** ~ 30000 photons / 511 keV gamma ray
- **Relatively cheap** vs. standard crystals such as LSO (Lu_2SiO_5[Ce])
- **Liquifies at near-atmospheric pressure**: SiPMs operate with low noise at LXe temperatures

Reconstruction in PETALO

<u>2 main components</u>

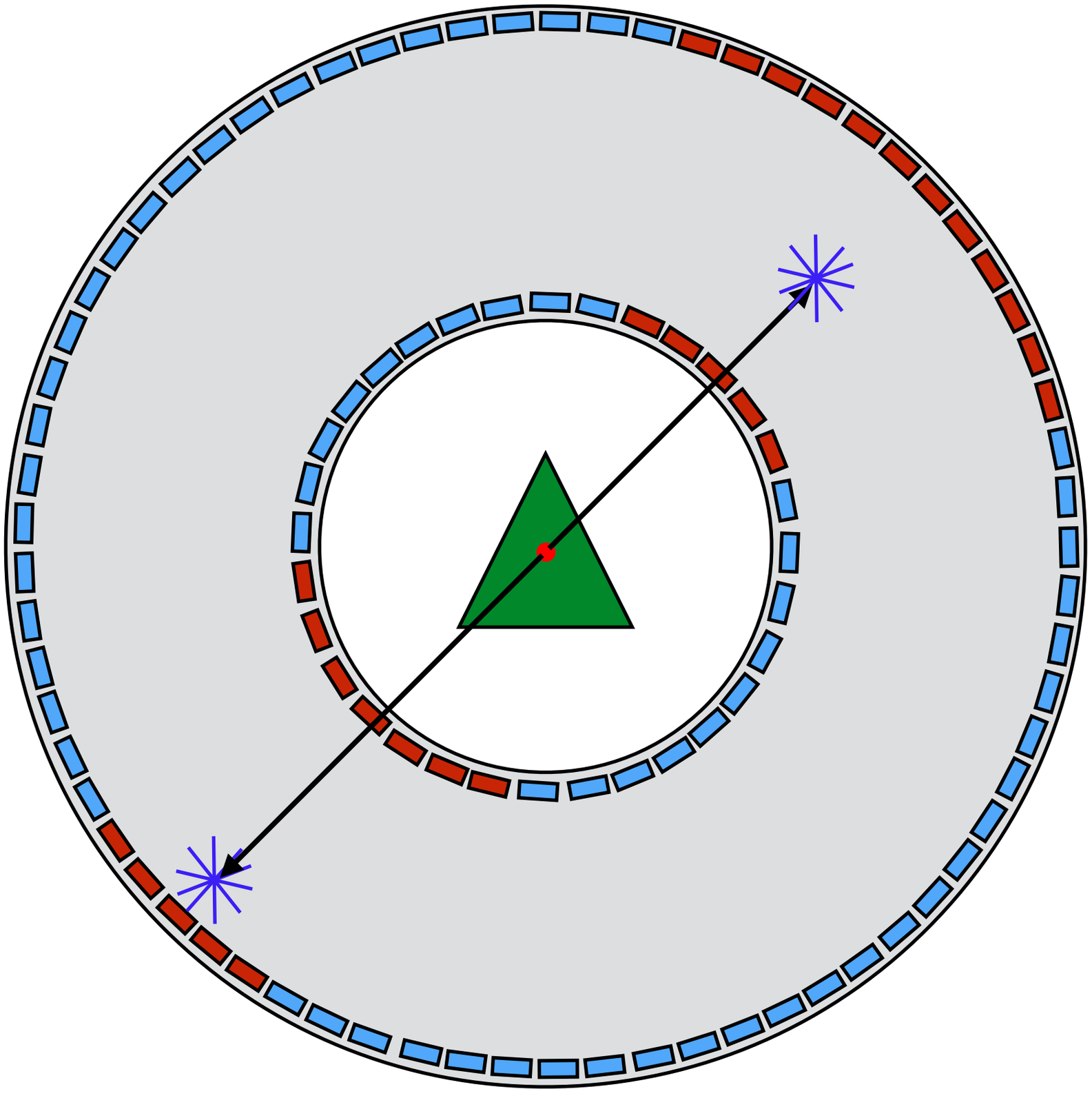
1. Reconstruction of γ-ray interactions **(focus of this talk)**
   - Pattern of light deposited on SiPMs —> reconstructed locations
   - Potentially use machine learning to identify Compton events
2. PET image reconstruction
   - Standard algorithms
   - Include time-of-flight information

PETALO concepts



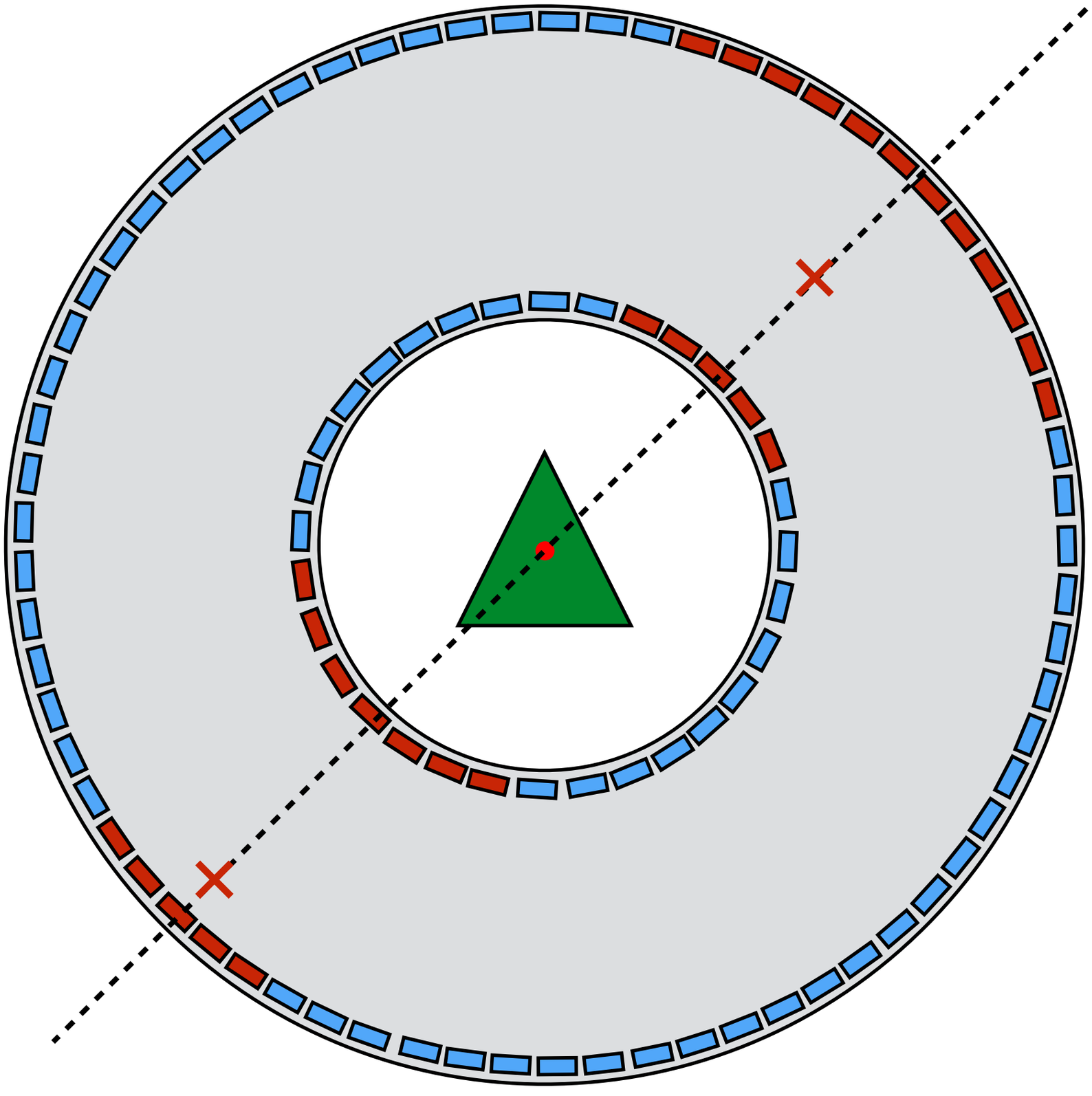
- Back-to-back 511 keV gamma rays are emitted from a point in the measured volume

PETALO concepts



- Back-to-back 511 keV gamma rays are emitted from a point in the measured volume
- The gamma rays interact in the LXe, producing VUV scintillation which is detected by the SiPMs in the vicinity

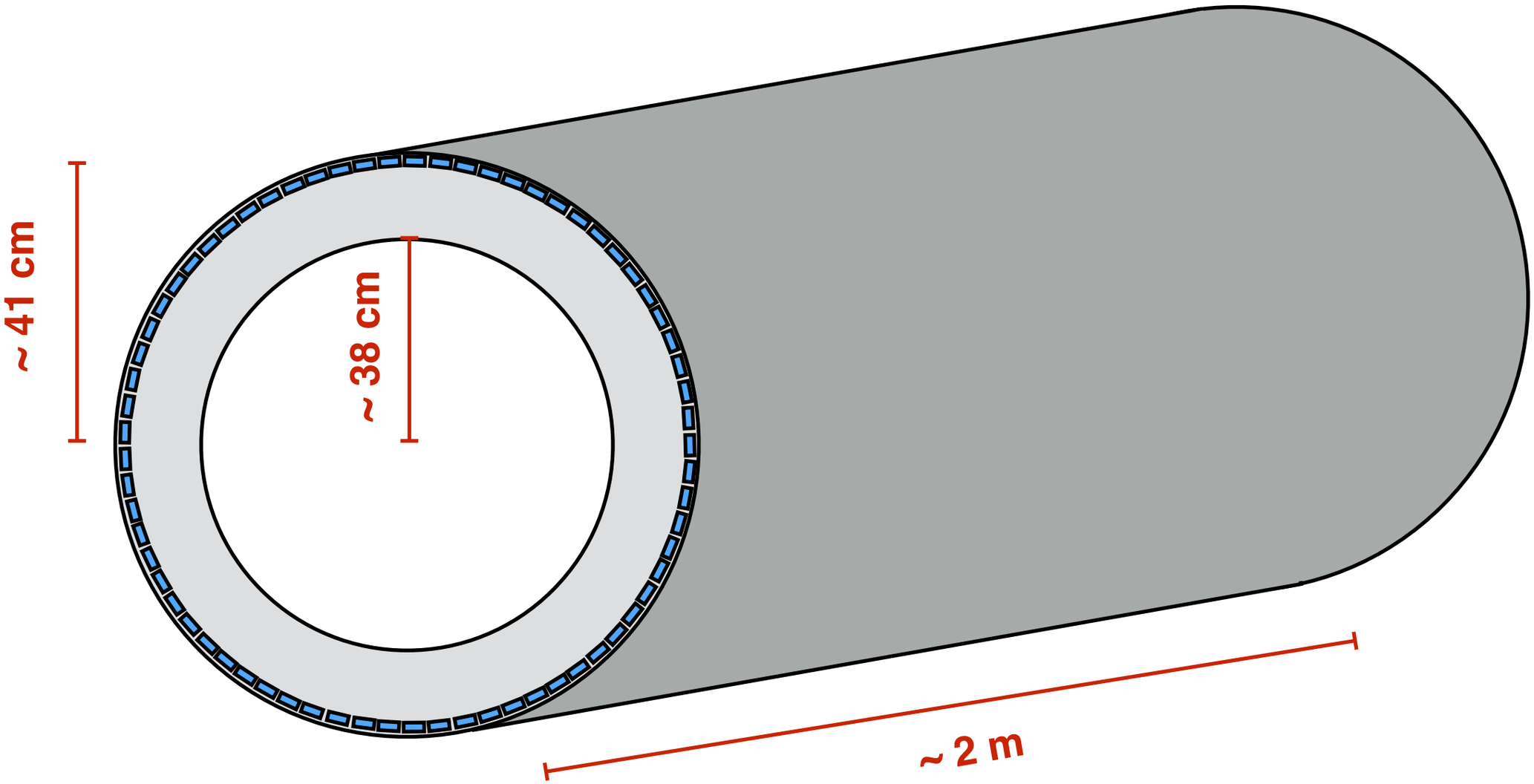
PETALO concepts



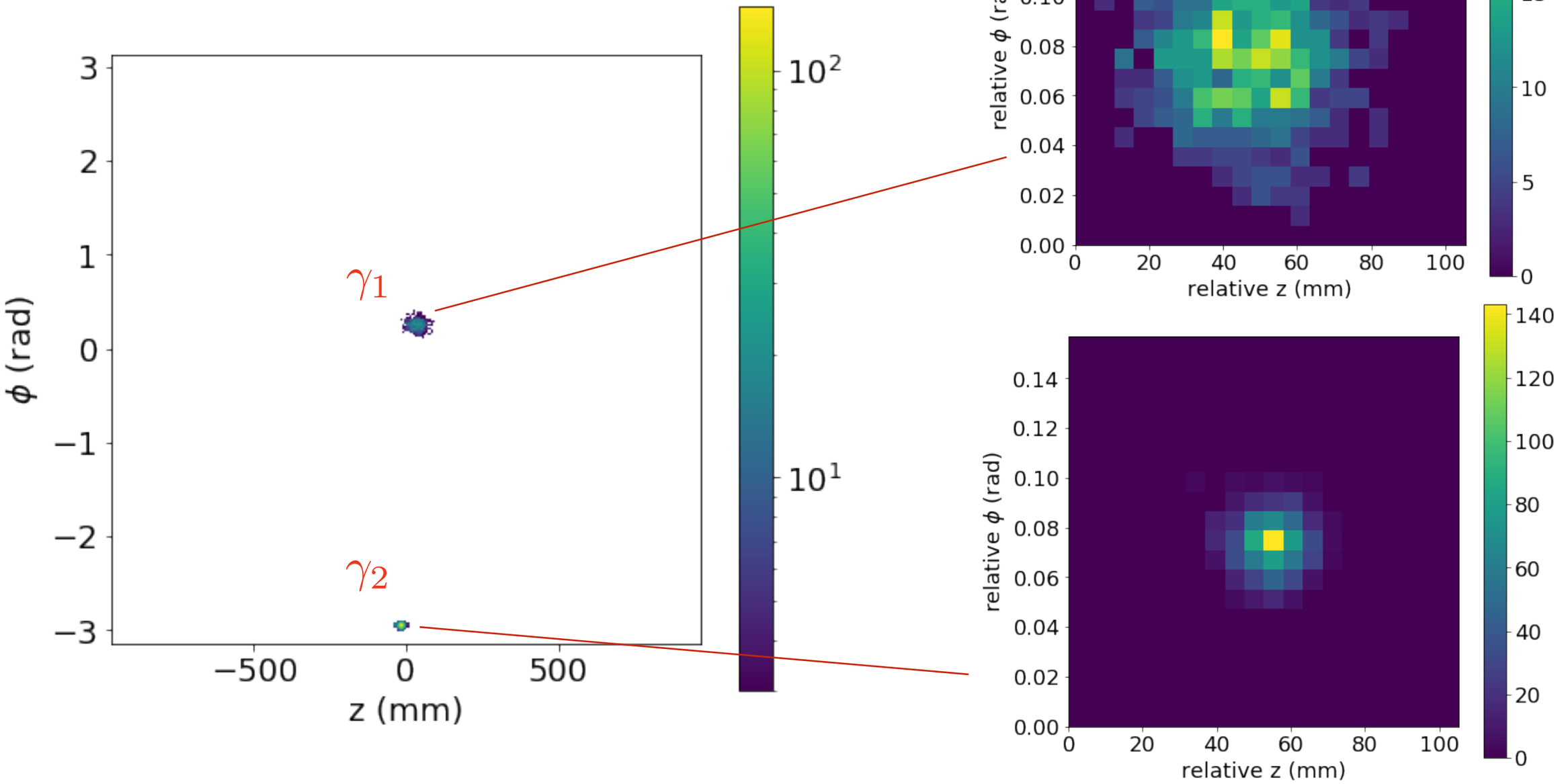
- Back-to-back 511 keV gamma rays are emitted from a point in the measured volume
- The gamma rays interact in the LXe, producing VUV scintillation which is detected by the SiPMs in the vicinity
- The interaction points are reconstructed from the light pattern detected on the SiPMs, and together they form a line of response (LOR)

Reconstruction of (r, ϕ, z)

- **Geant4-based Monte Carlo:** “full-body” configuration
  - Z-extent of approximately 1.94 m
  - Inner radius ~38 cm, outer radius ~41 cm
  - Inside of outer ring instrumented with SiPMs
  - 7 mm SiPM spacing (278 rings of 368 SiPMs)



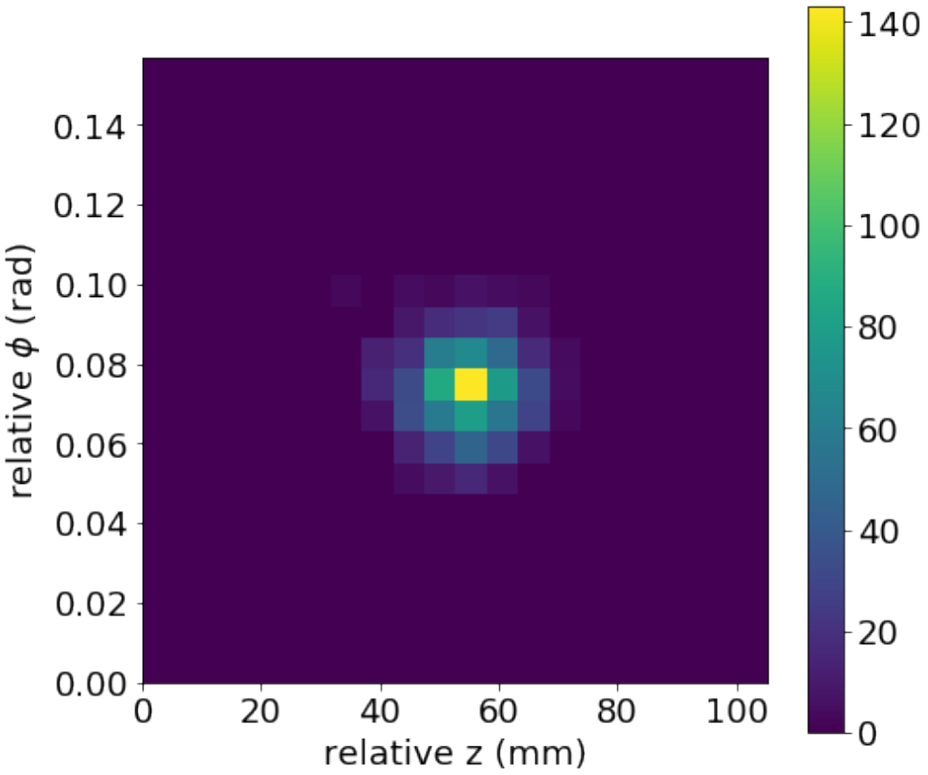
Reconstruction of (r, ϕ, z)



- “Unroll” the ring in (ϕ,z)
- Divide the sensor response into two regions for the two gammas
- Reconstruct (r, ϕ, z) for each gamma

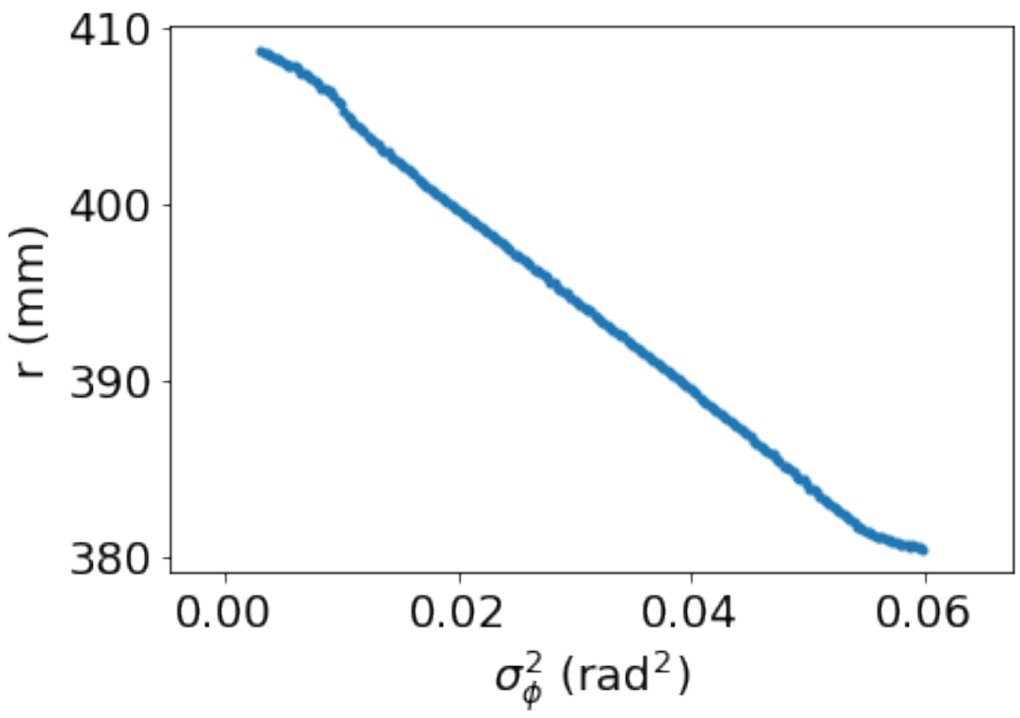
Reconstruction of (r, ϕ, z)

- Coordinates z and ϕ computed via weighed average



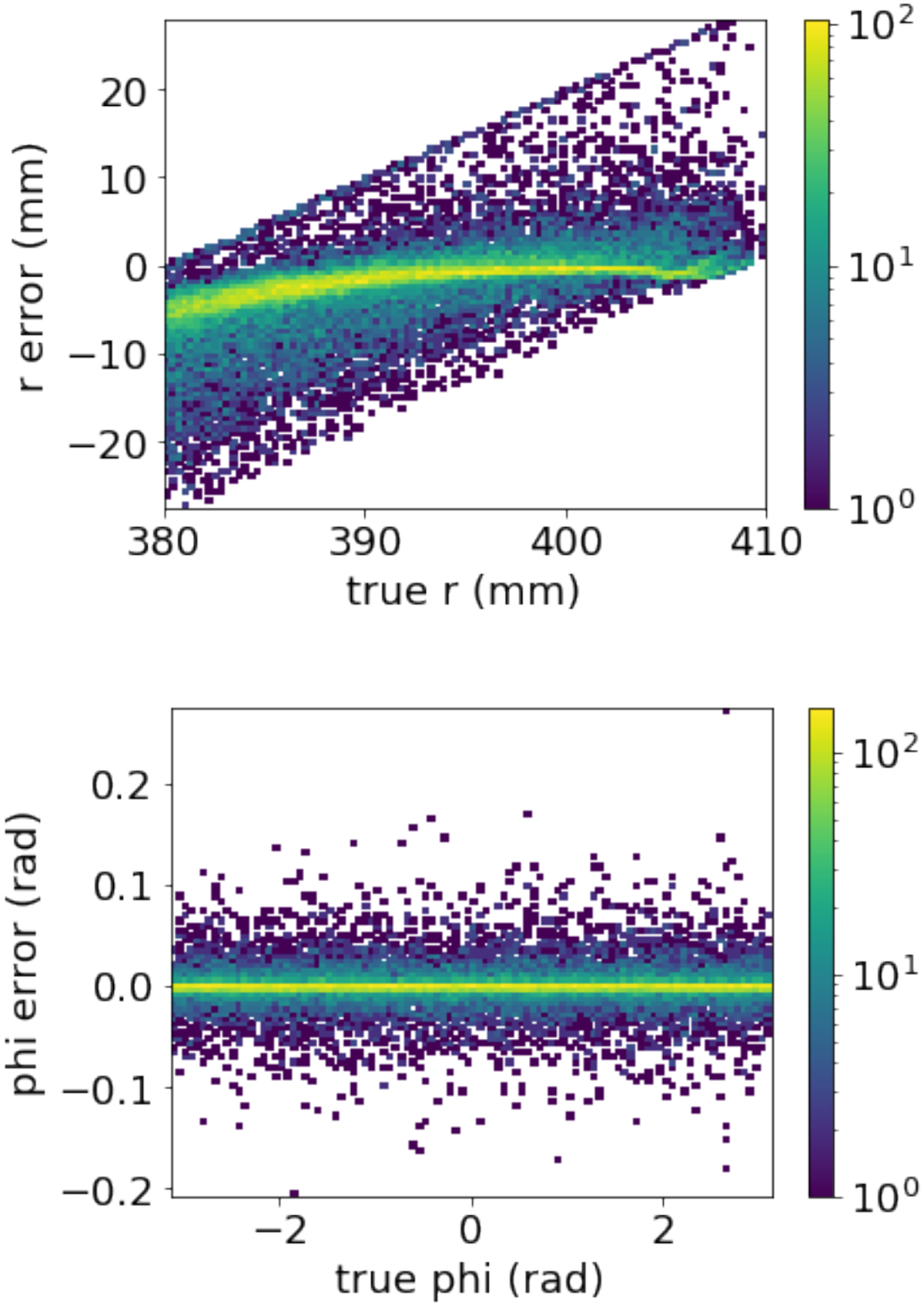
$$\phi = \frac{1}{Q} \sum_{q_i > q_{t,\phi}} \phi_i q_i \quad z = \frac{1}{Q} \sum_{q_i > q_{t,z}} z_i q_i$$

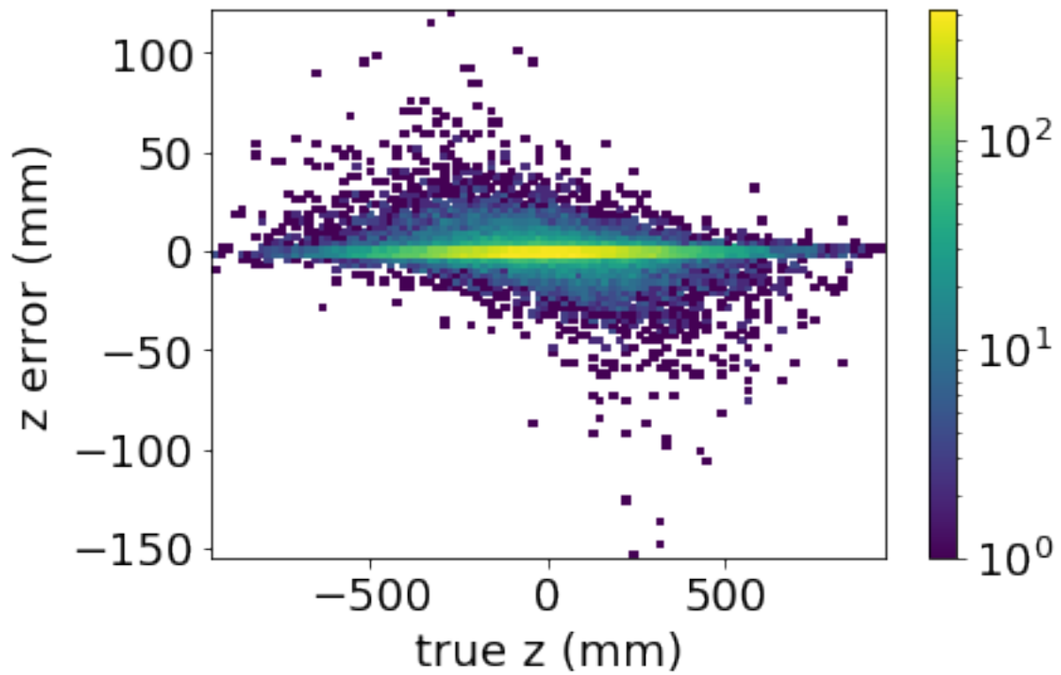
- Radial coordinate deduced from the variance of ϕ



*** For each coordinate, only SiPMs above a certain charge threshold are included in the weighted averages (presently using 4 photons)**

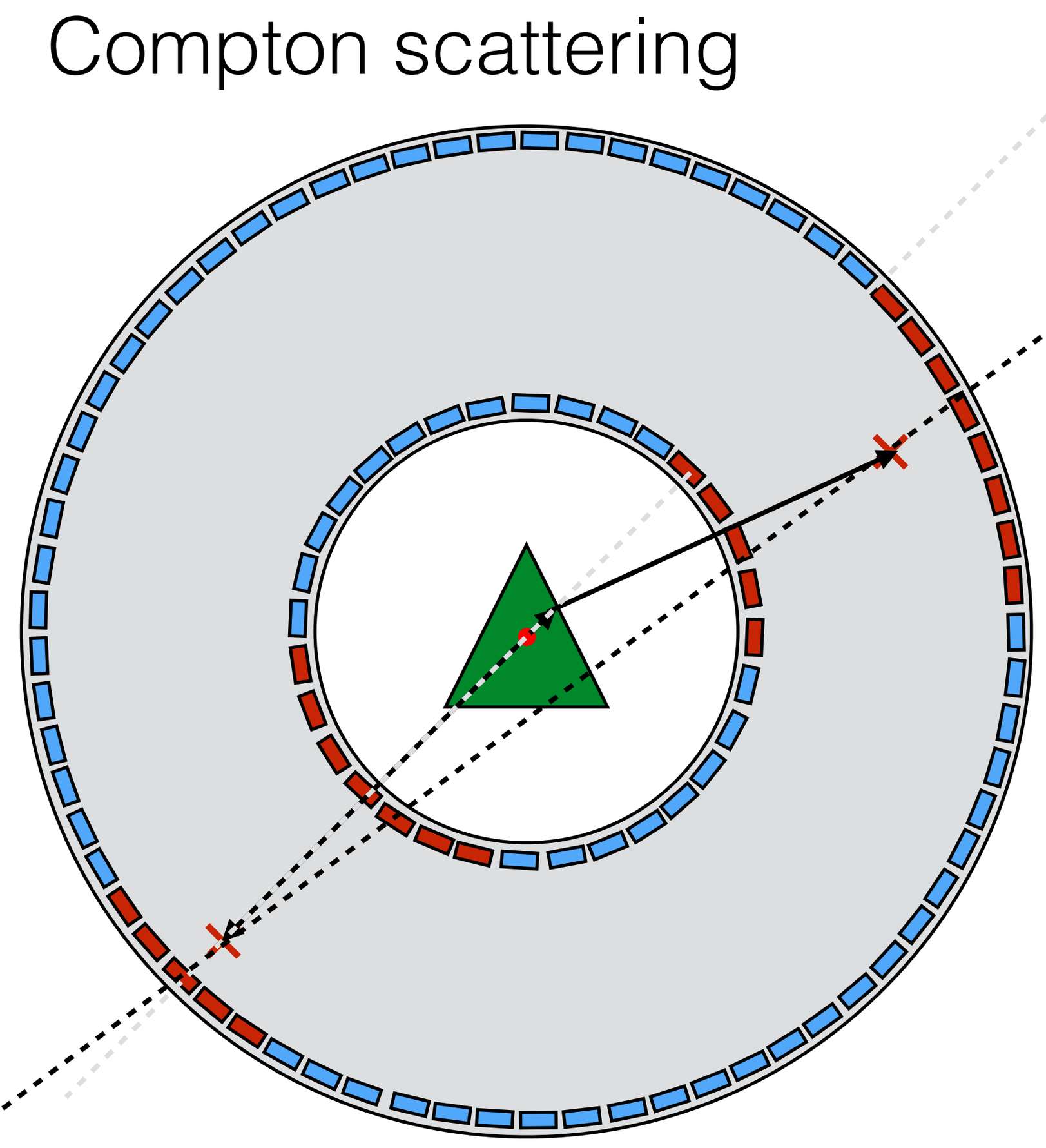
Reconstruction of (r, ϕ, z)





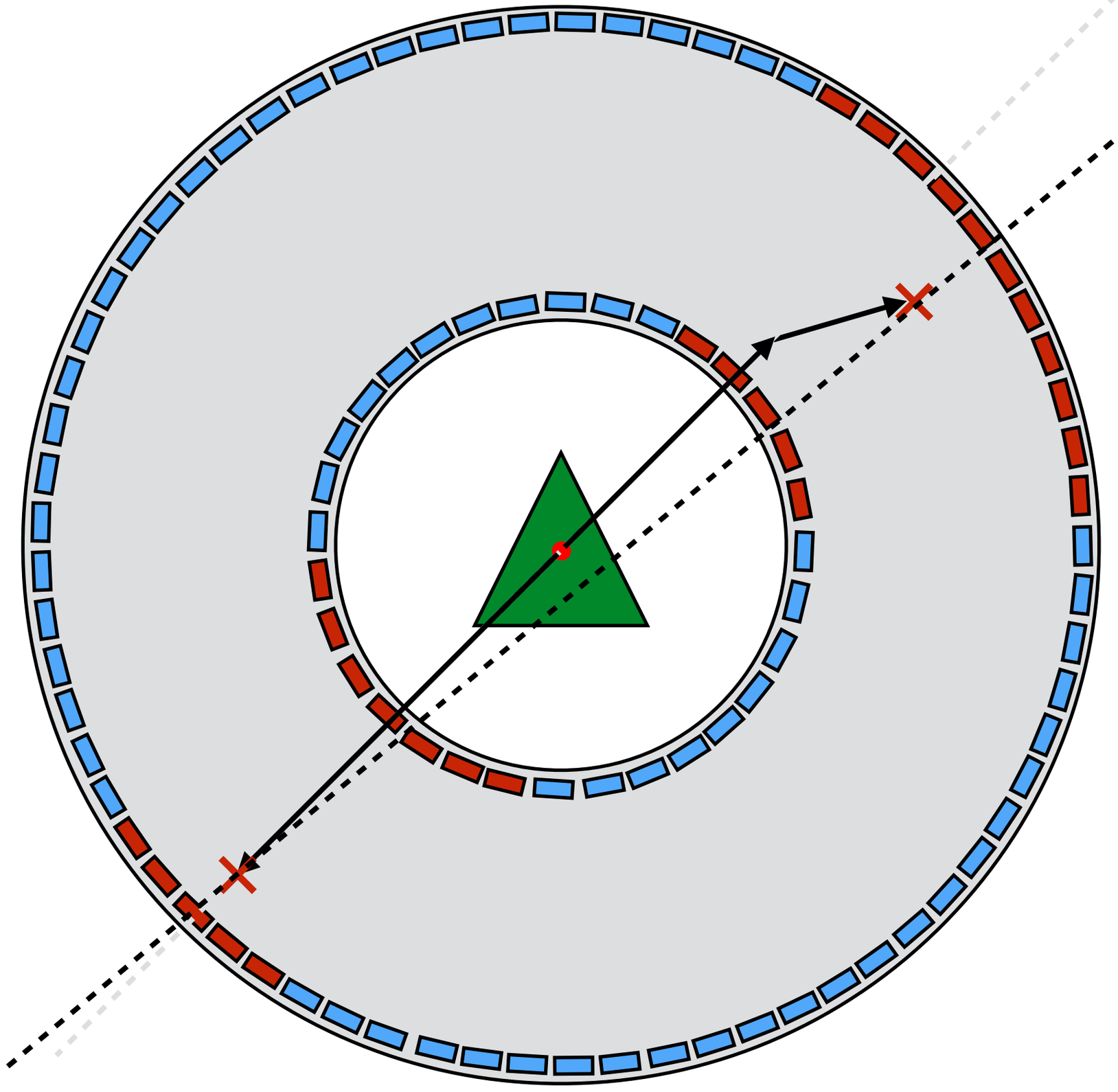
- ϕ in general is well-reconstructed
- r and z often affected significantly by problematic events (Compton scatters, for example)

Compton scattering



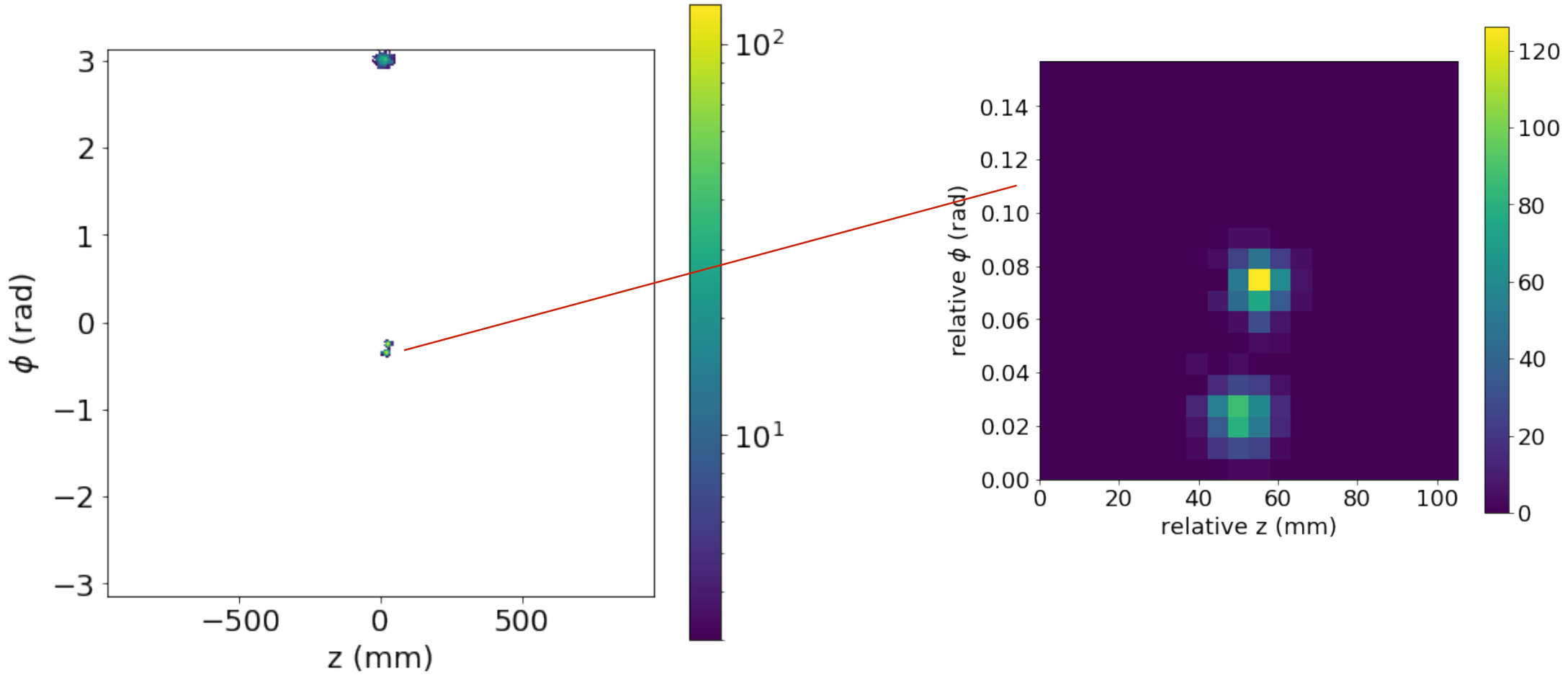
- A gamma ray may Compton scatter in the measured volume, losing some energy and changing its direction of travel
- Such events may be removed with an energy cut accepting only coincident 511 keV gammas

Compton scattering



- Note that a gamma ray may Compton scatter in the LXe but still deposit all its energy within the active volume
- These events are more difficult to remove

Compton scattering



- In this event, one gamma clearly Compton scattered: clear separation of depositions
- Often separation of depositions is not so clear
- One idea: **train a neural network to predict the reconstruction error in (r, z) of single gamma ray depositions**

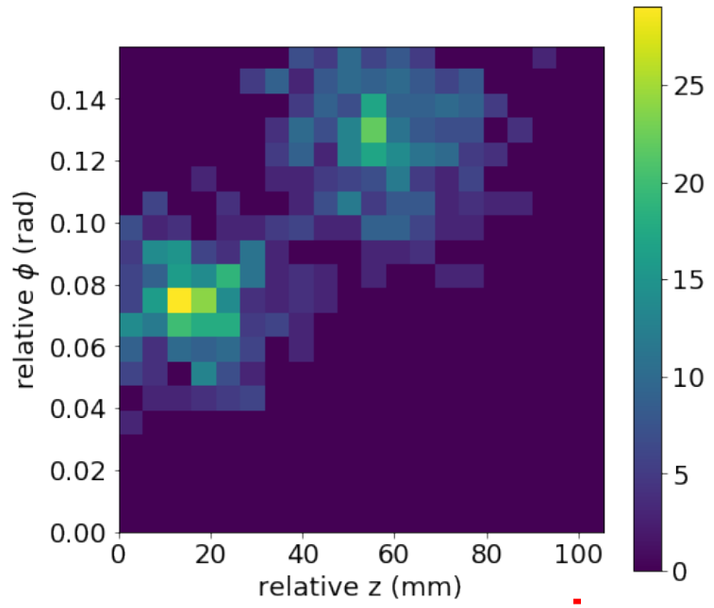
CNN-based approach

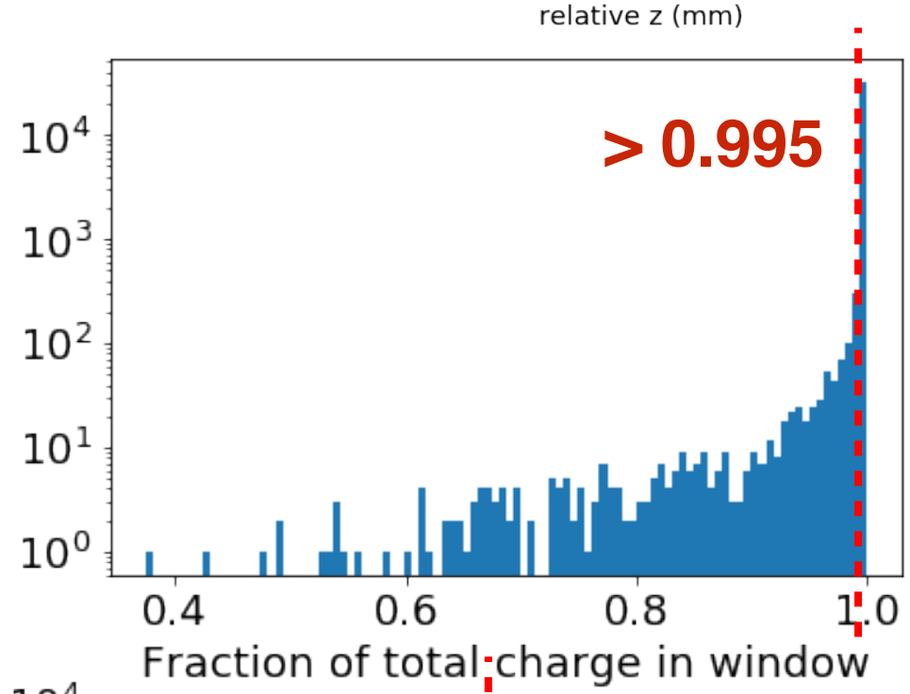
- Data samples are 20x20 SiPM “windows” of the event, centered on the (z,ϕ) bins with maximum summed SiPM charge

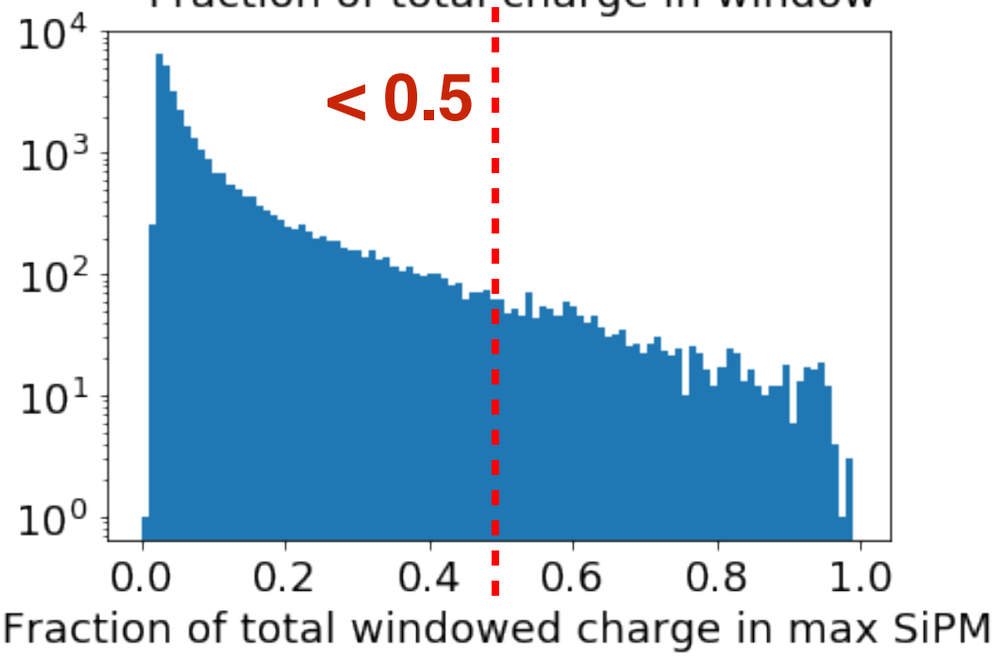
- **2 pre-processing cuts**

  1. Remove events with significant charge outside 20x20 window for each gamma
  2. Remove events with majority of charge in a single SiPM

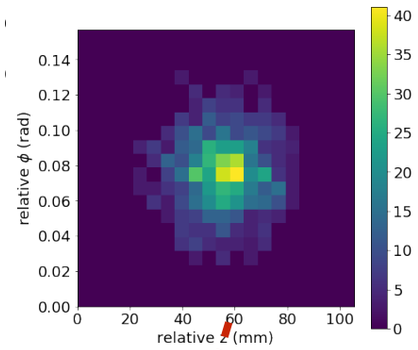
Each cut eliminates ~ 4% of events



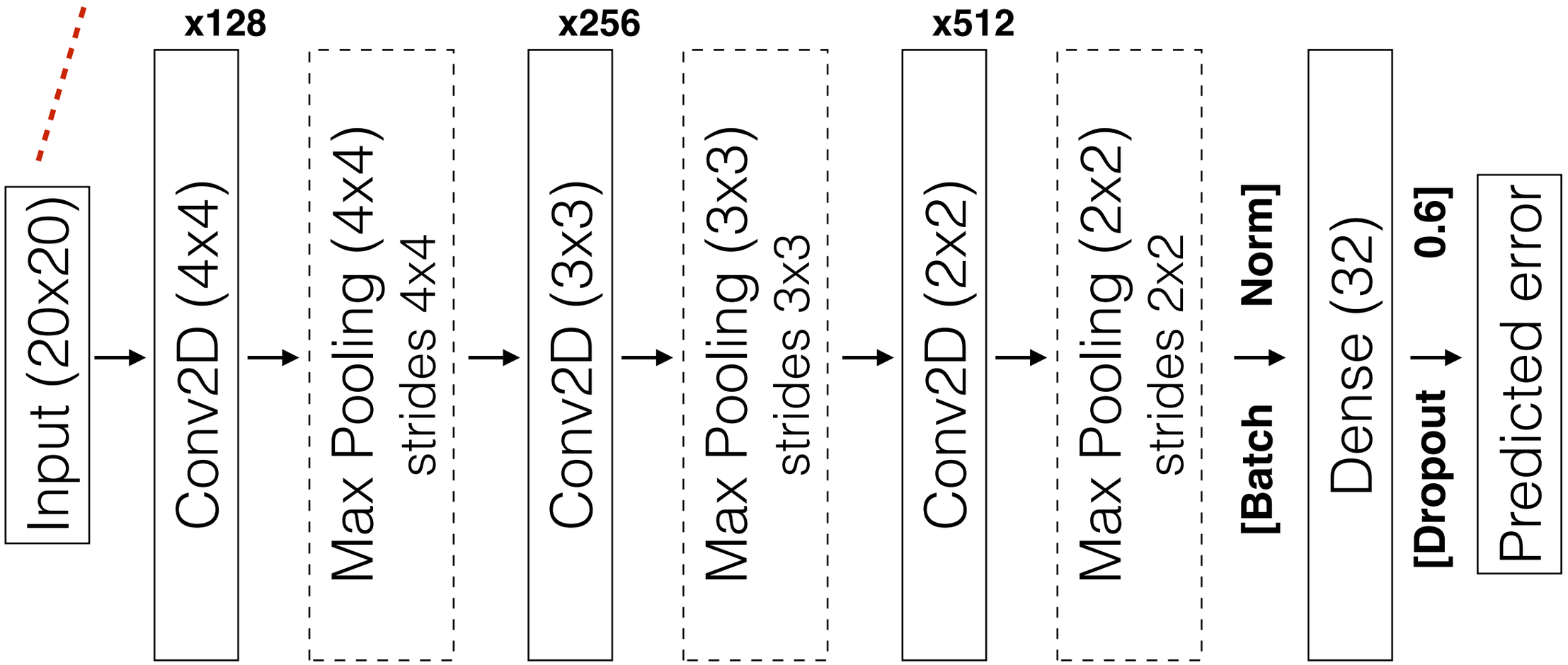




CNN-based approach



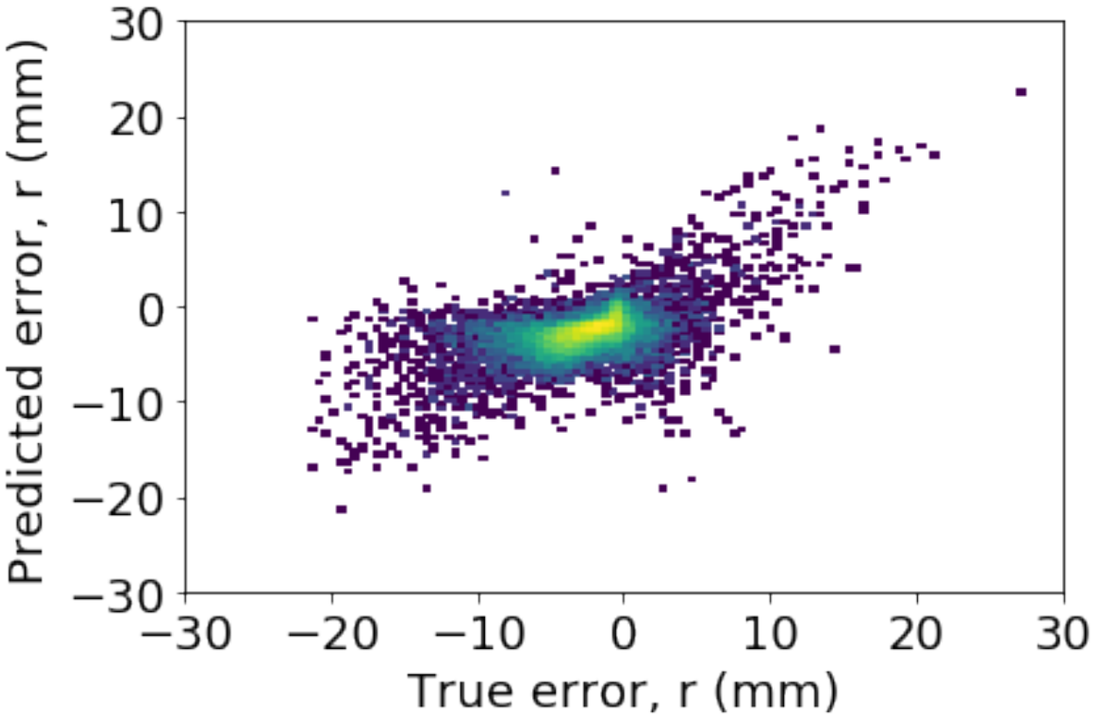
Basic CNN architecture (built in Tensorflow[1]/Keras[2])

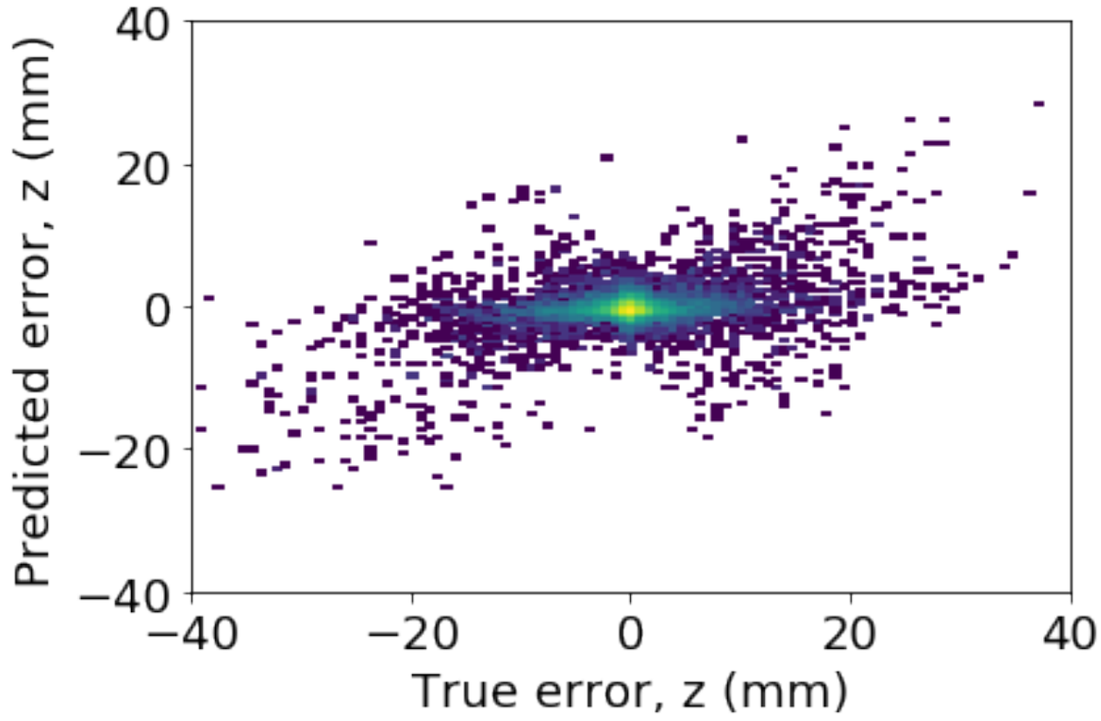


- [1]Tensorflow: https://www.tensorflow.org
- [2]Keras: https://keras.io

CNN-based approach

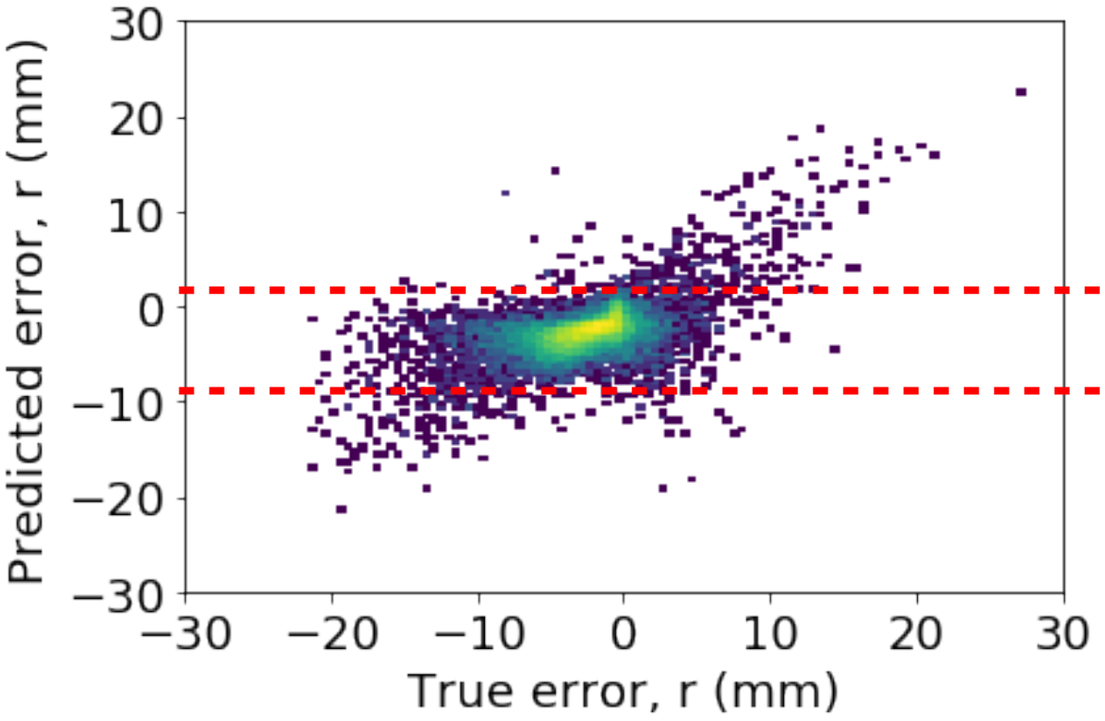
- CNN predictions (15k training, ~15k test events)

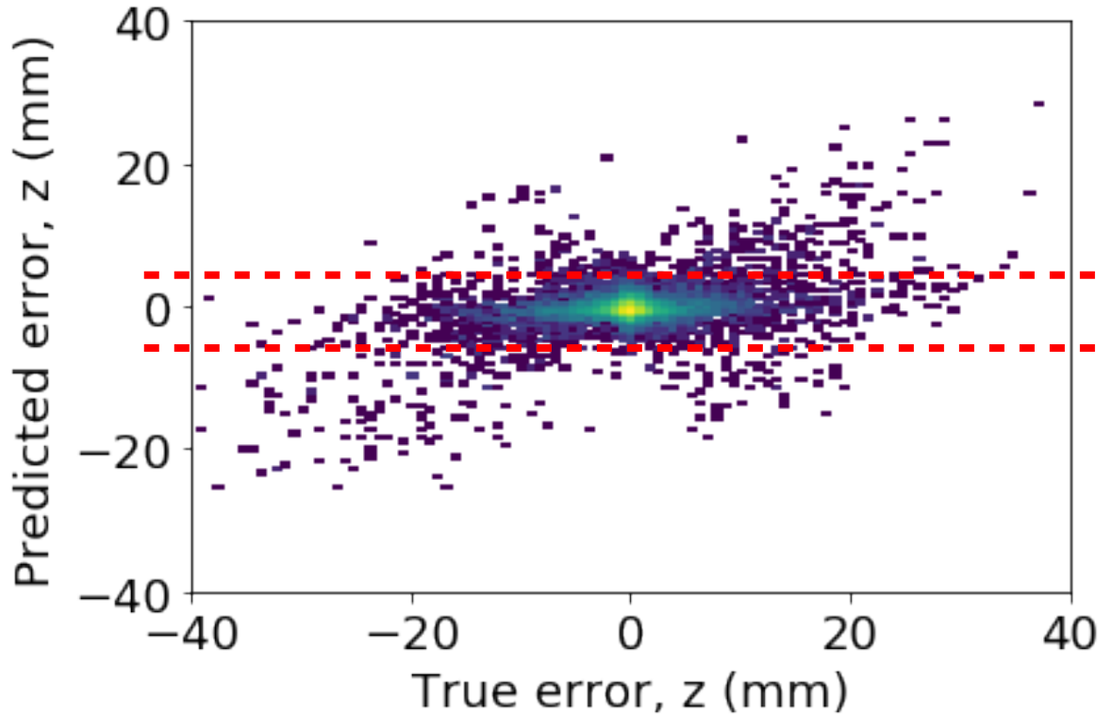




CNN-based approach

- CNN predictions (15k training, ~15k test events)

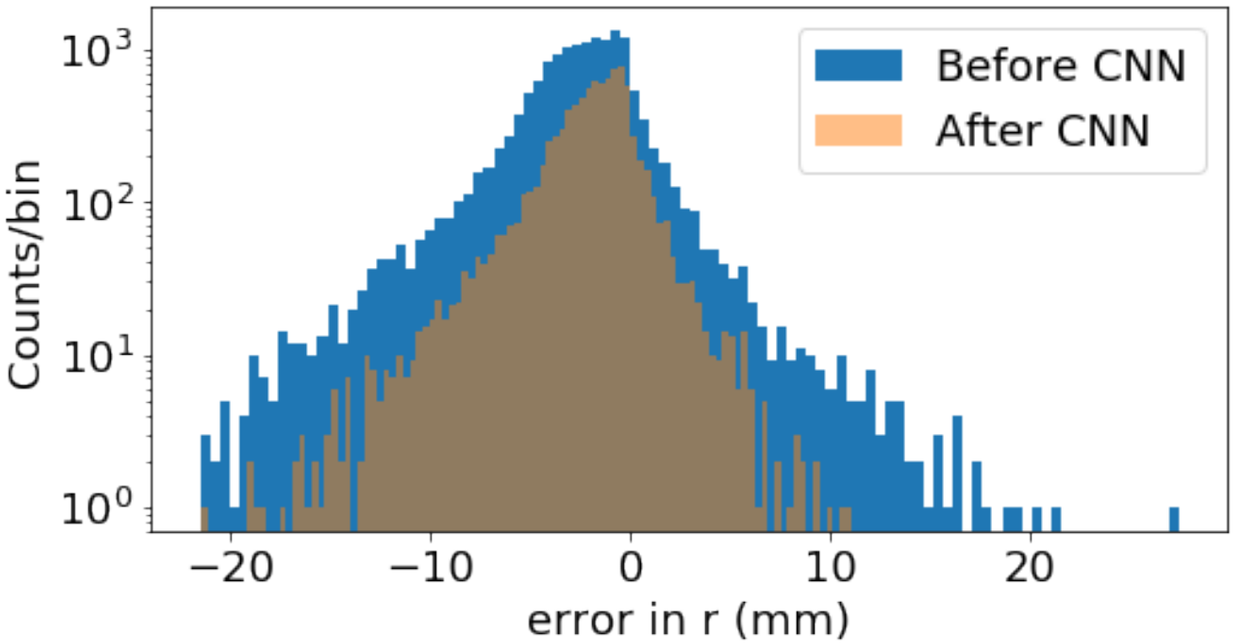


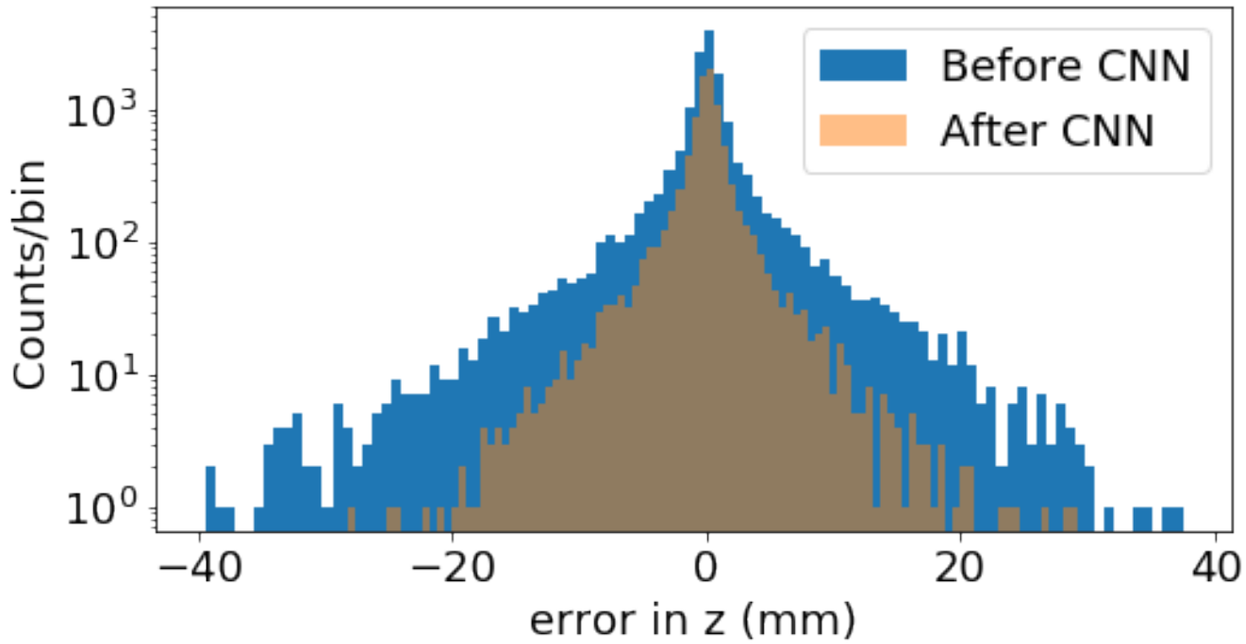


- Eliminate poorly reconstructed events by cutting on predicted error

CNN-based approach

- An example CNN cut (eliminates additional ~40% of total events in each case)





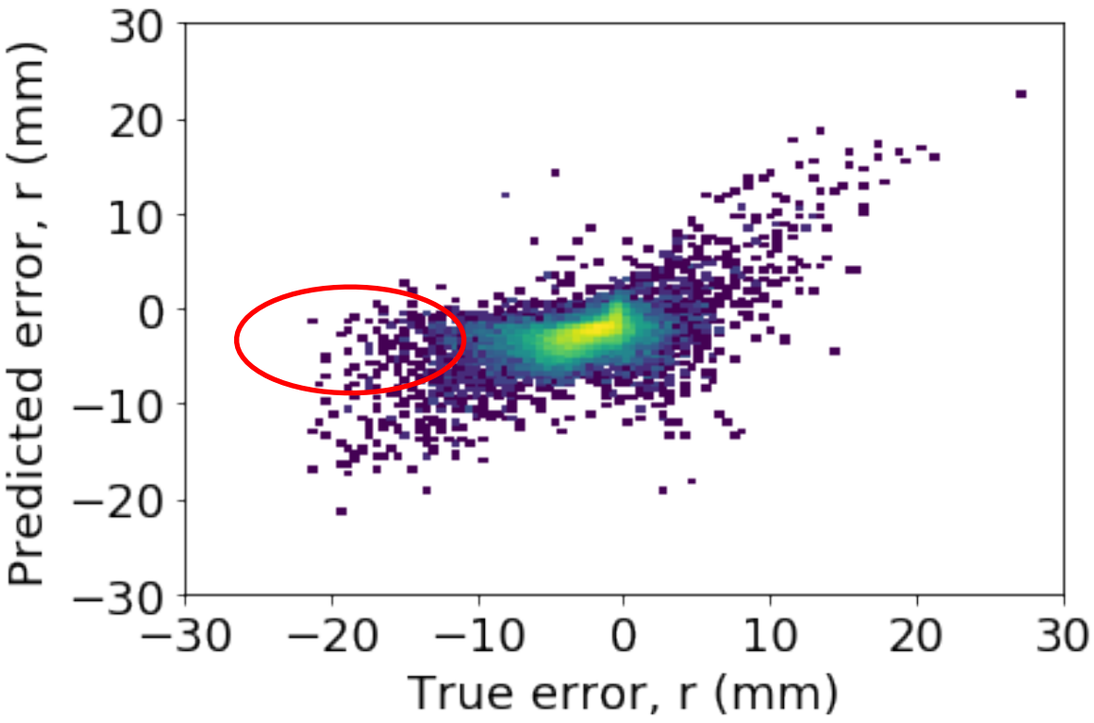
-3 mm < r-error (CNN) < 1 mm

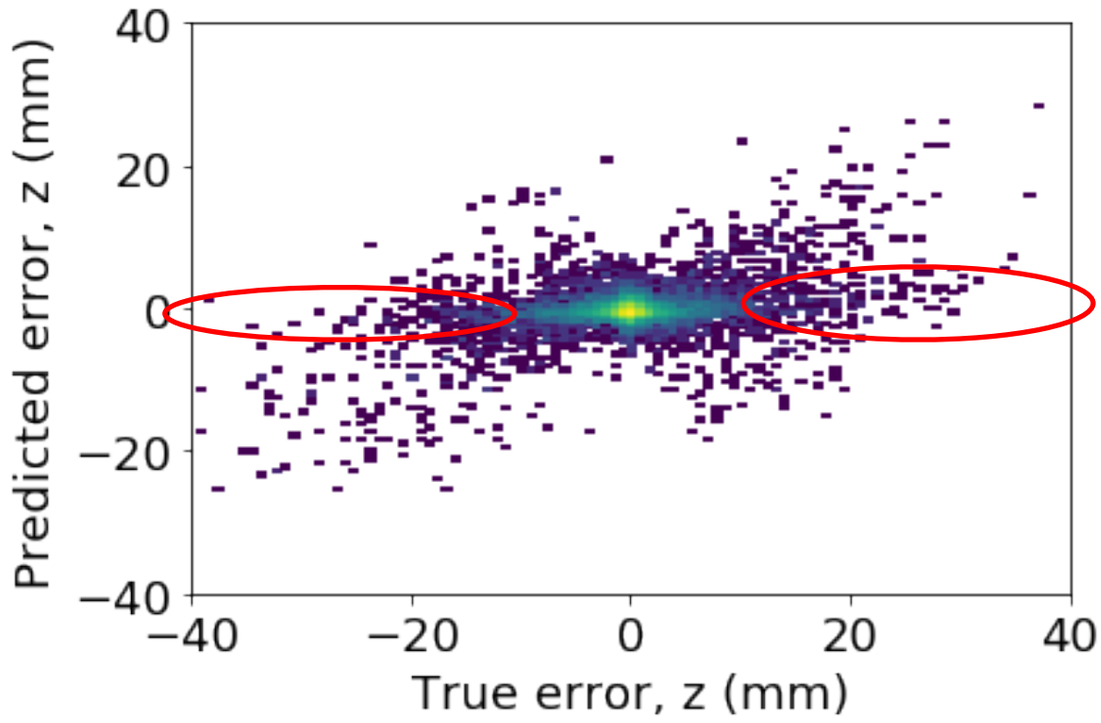
-0.75 mm < z-error (CNN) < 0.75 mm

- Few convincing gains from the r-coordinate
- z-coordinate looks more promising, but further evaluation needed

CNN-based approach

- CNN predictions (15k training, ~15k test events)

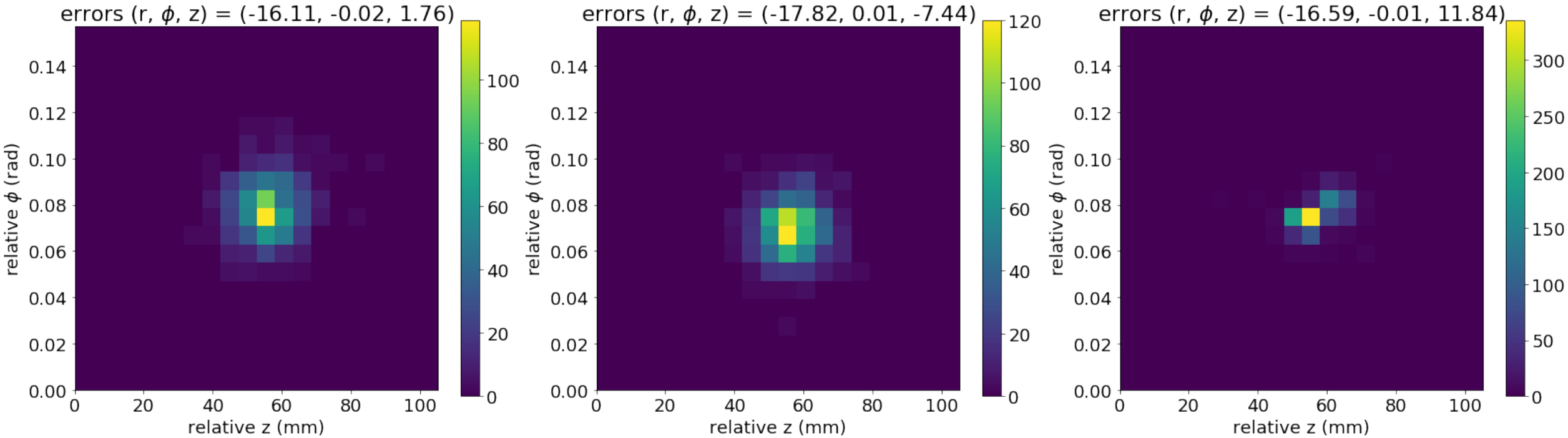




- Eliminate poorly reconstructed events by cutting on predicted error
- Some events still manage to get by the CNN

CNN-based approach

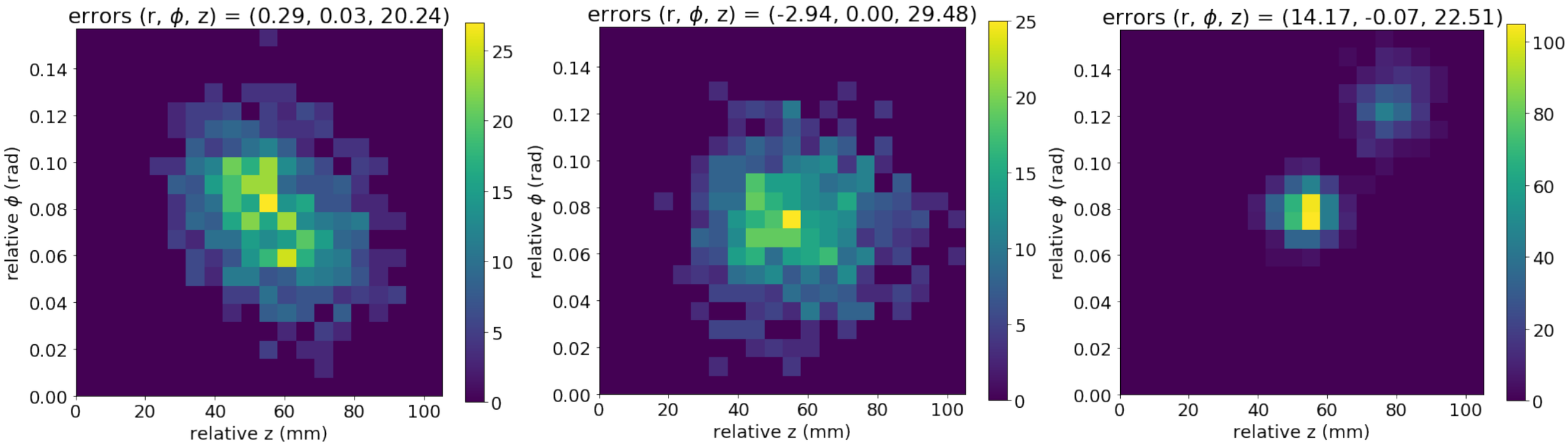
- Some events with true r-error < - 15 mm, but predicted r-error in the range (-5 mm, 5 mm)



- Many of these events don't look like Compton scatters
- The event on the far right may be a Compton that did not travel very far before interacting again (note the z-error is also relatively high)

CNN-based approach

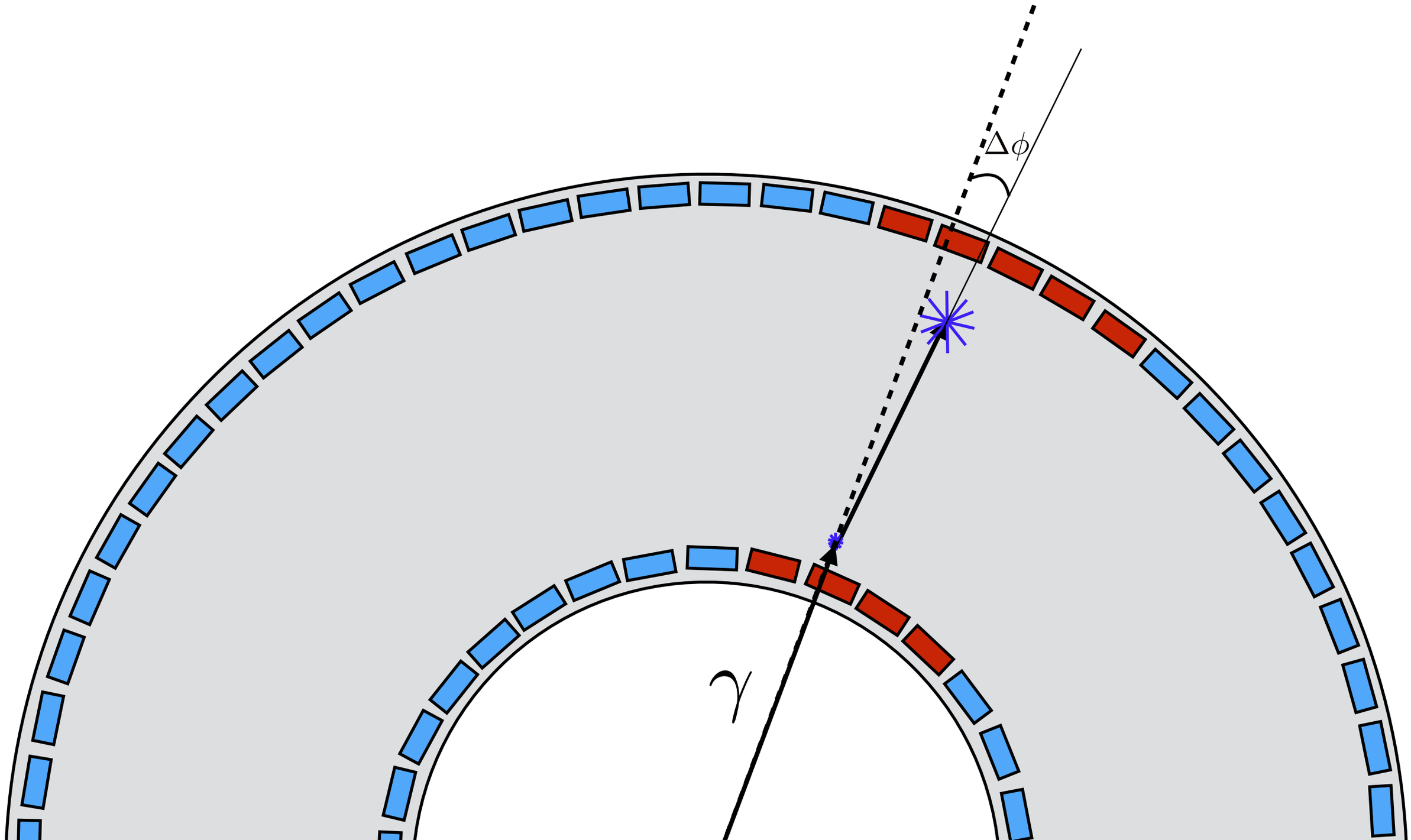
- Some events with true z-error > 20 mm, but predicted z-error in the range (-5 mm, 5 mm)



- Most of these events look disperse (interacted at low r)
- The event on the right is clearly a Compton that was missed

CNN-based approach

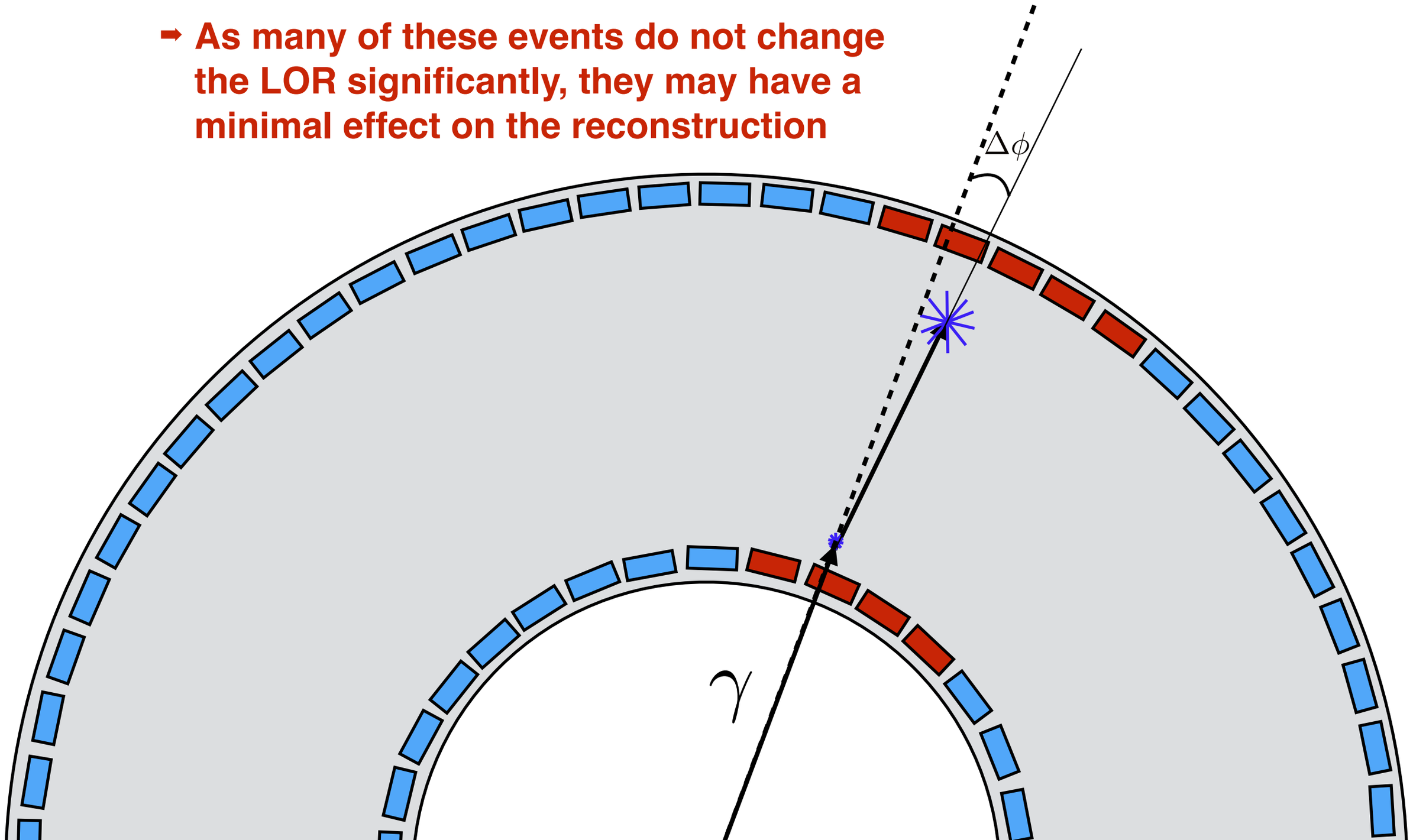
- A primary issue seems to be “soft” Comptons, which interact twice, depositing very little energy in the first interaction



CNN-based approach

- A primary issue seems to be “soft” Comptons, which interact twice, depositing very little energy in the first interaction

  ➡ **As many of these events do not change the LOR significantly, they may have a minimal effect on the reconstruction**



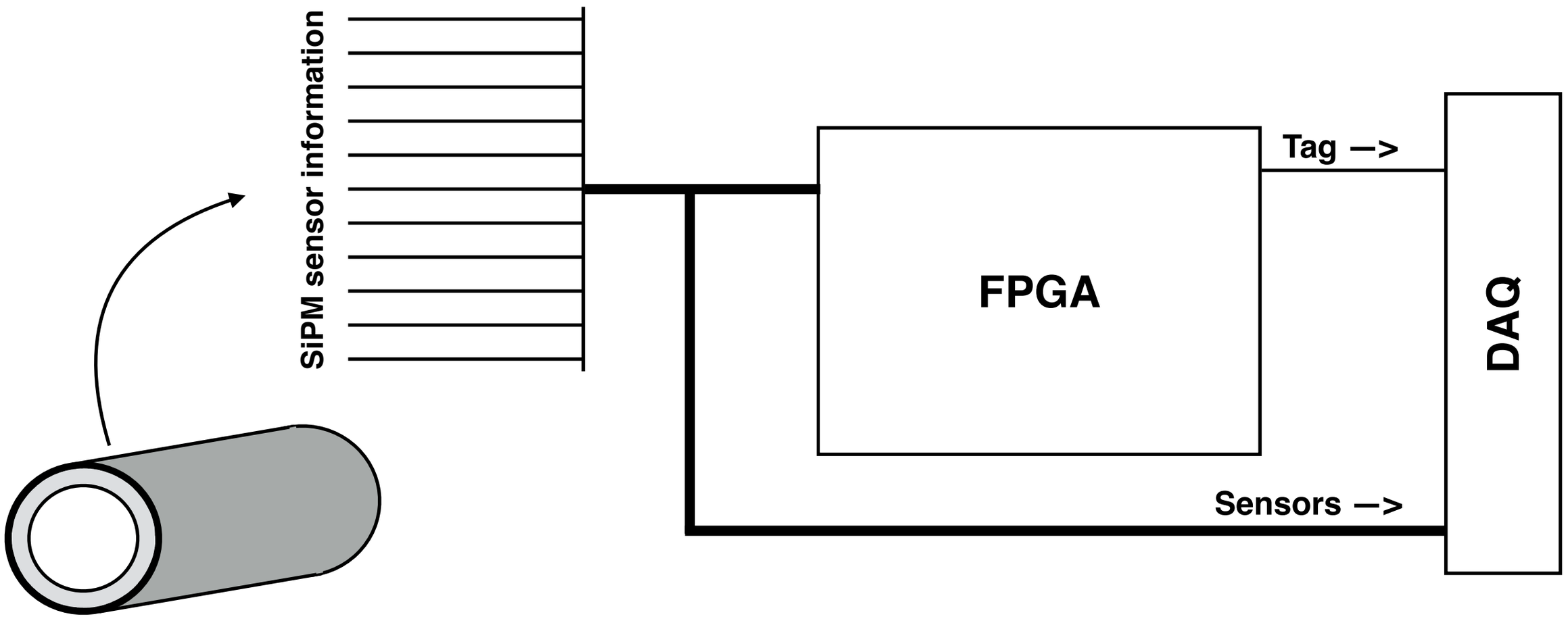
CNN-based approach

- **Summary**:

    - The CNN is likely to be useful, but several questions remain:
        - How much error can be tolerated?
        - Is the relative gain in eliminating a poorly reconstructed event greater than the cost of losing some good events?
        - Perhaps by re-labeling “soft” Compton events the CNN can be trained more precisely
    - Will need to test within the scheme of full PET reconstruction, eventually with TOF, to make a clear statement on this

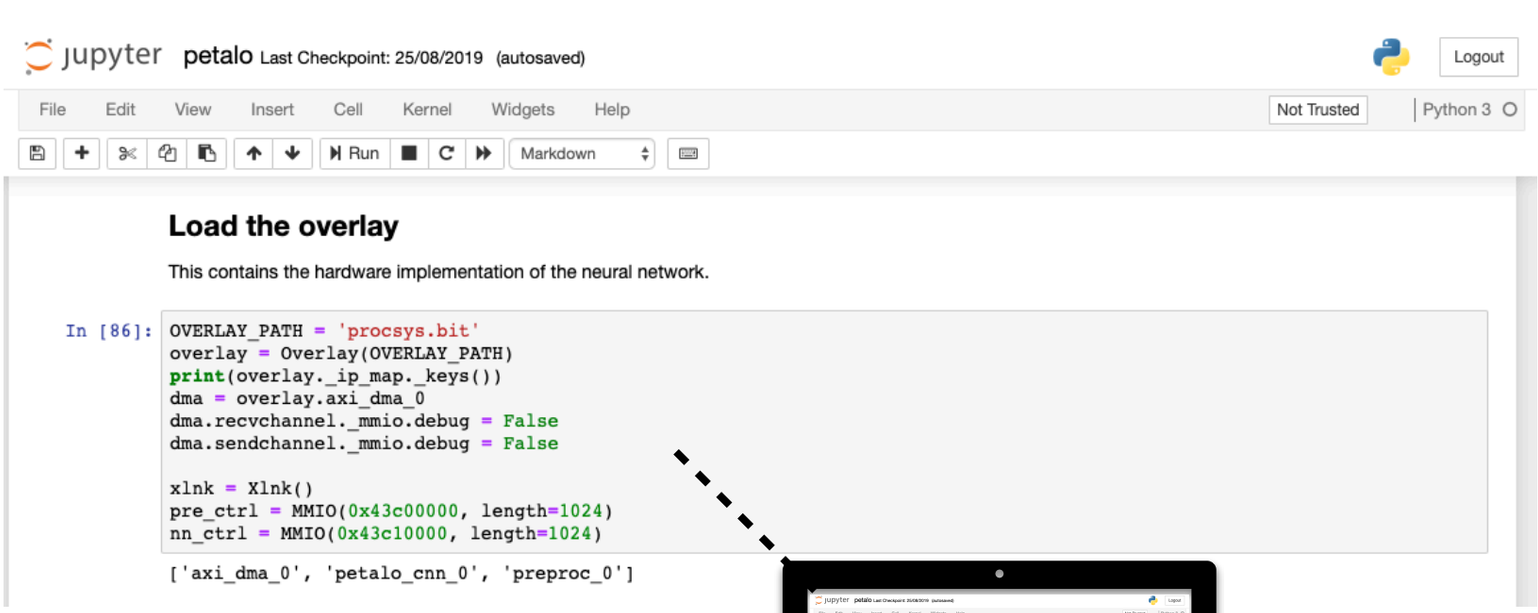
Hardware integration (prototype)

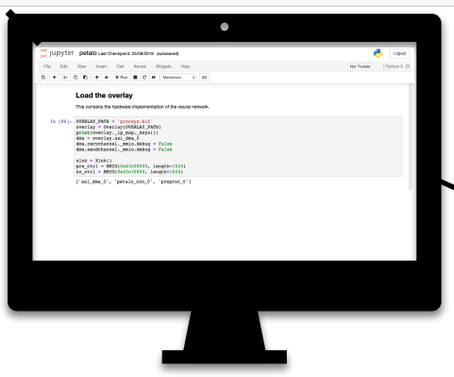
- Ideally, events could be tagged and potentially discarded *on-the-fly*
- Integrate CNN into hardware via an FPGA

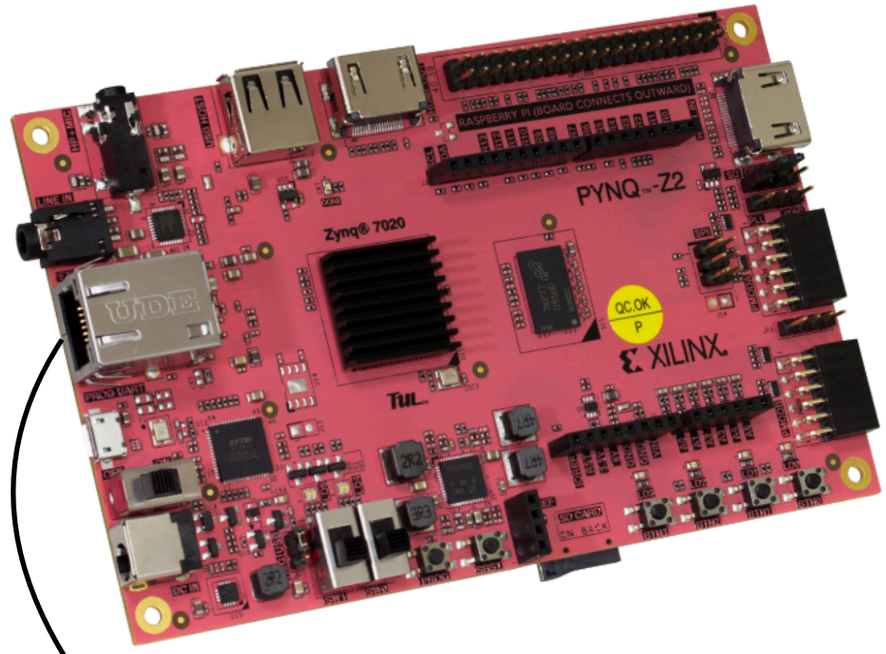


Hardware integration (prototype)

- Adapted code for CNN training and deployment on FPGA from the spooNN[1] project
- Uses high level synthesis (HLS) C++ libraries to generate hardware description language (HDL) code, which can be loaded into an FPGA
- Interact with FPGA prototype from Python/Jupyter using PYNQ[2]







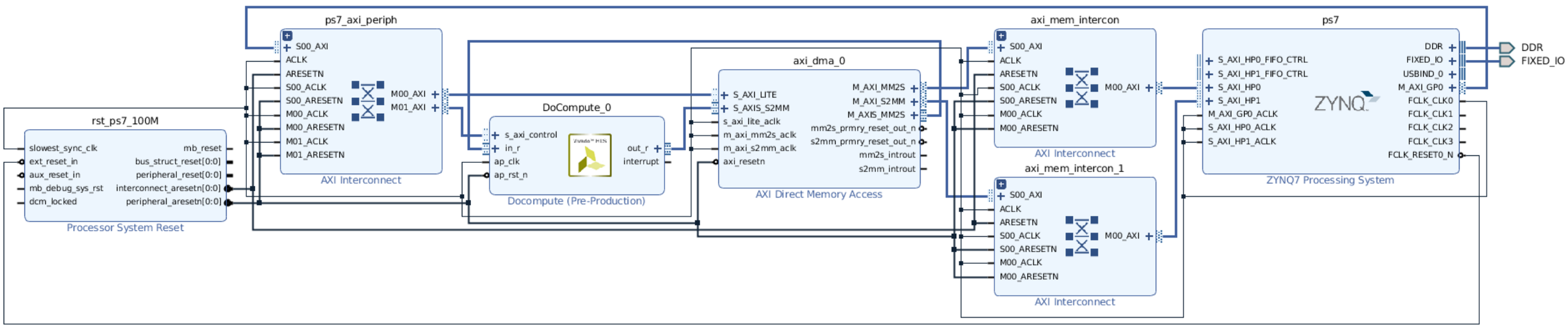
LAN

➡ [1]spooNN project: https://github.com/fpgasystems/spooNN

➡ [2]PYNQ: http://www.pynq.io

Hardware integration (prototype)

1. Train CNN using Tensorflow/Tensorpack[1]
   - Output weights to HLS arrays
2. Compile HLS to produce programmable logic (IP)
   - This can in principle be used in other designs (may need to adapt input/output interface)
3. Integrate IP into a PYNQ-compatible design

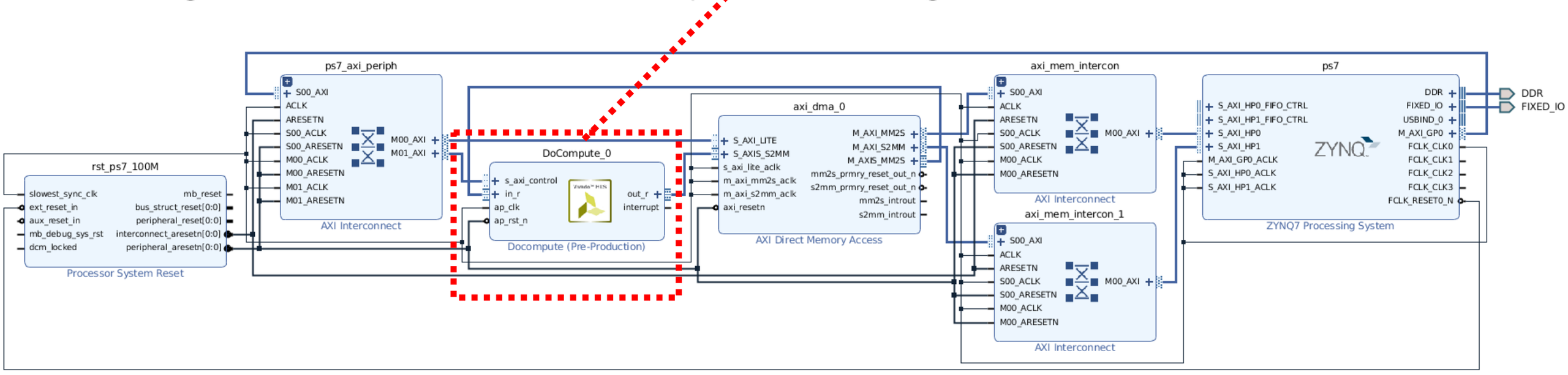


➡ [1]Tensorpack: https://github.com/tensorpack/tensorpack

Hardware integration (prototype)

```
const ap_uint<8*20> weights4[2][2] = {
{"0xf1088fe09de3d2133faeeff7d0fd0de53b51c0c7",
{"0xda6b735df2f70a000f70fdbeb0c7a5ea725210c5",
};
```

1. Train CNN using Tensorflow/Tensorpack[1]
   - Output weights to HLS arrays
2. Compile HLS to produce programmable logic (IP)
   - This can in principle be used in other designs (may need to adapt input/output interface)
3. Integrate IP into a PYNQ-compatible design

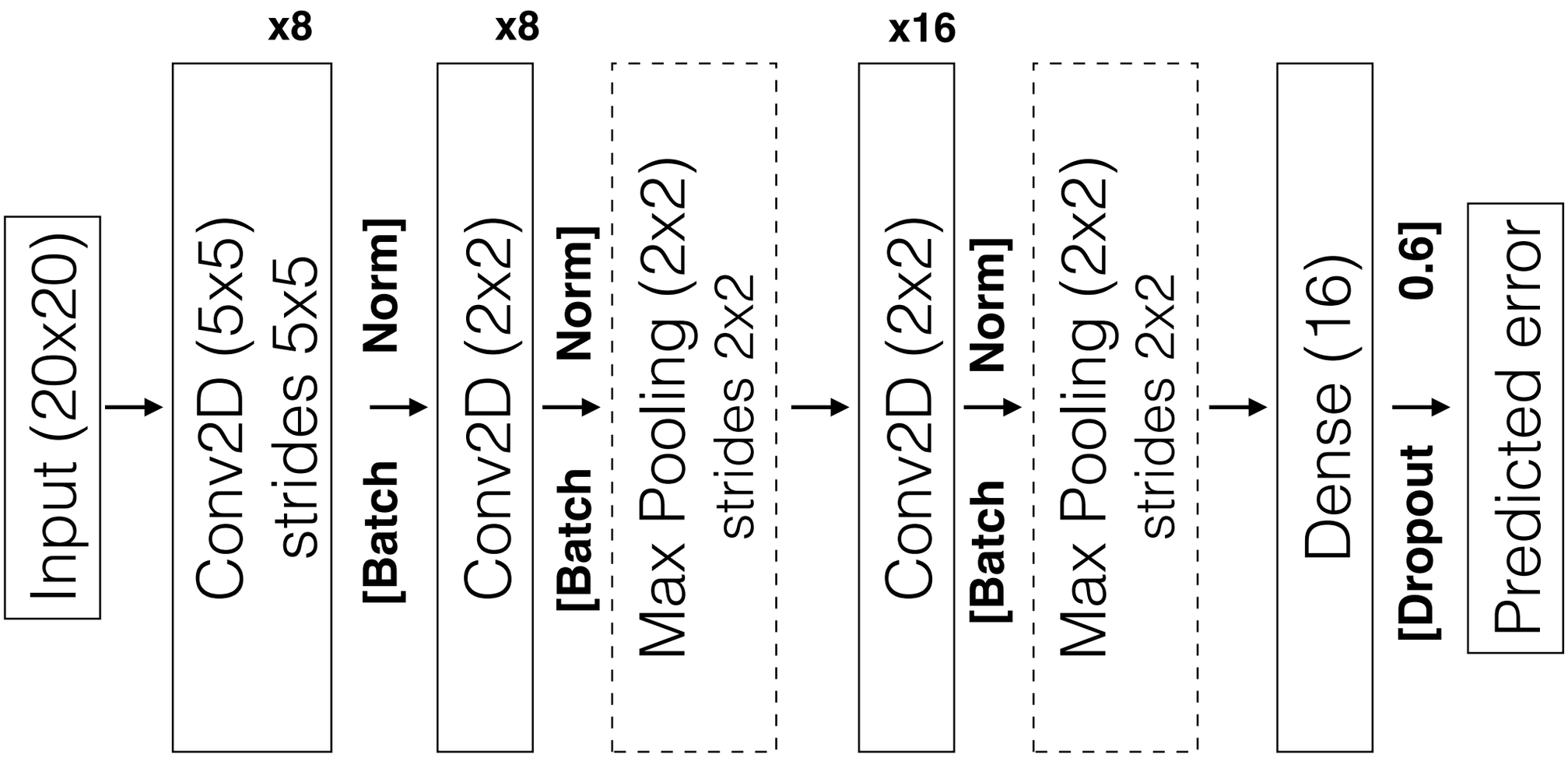


➡ [1]Tensorpack: https://github.com/tensorpack/tensorpack

Hardware integration (prototype)

- Attempting to meet timing restrictions of ~10 μs/event (compact CNN)

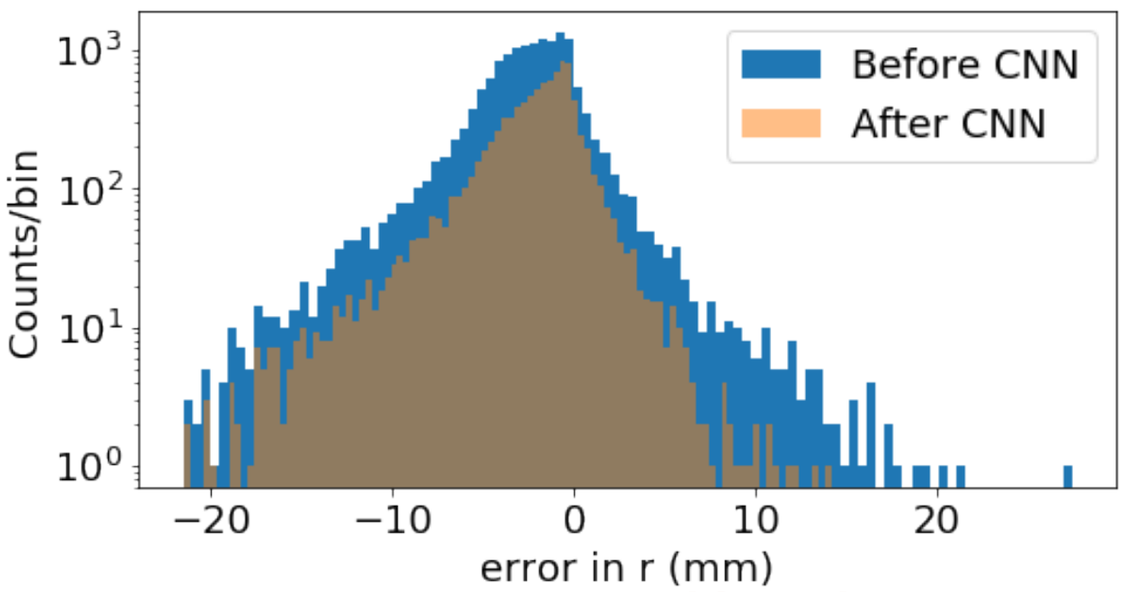
Smaller CNN (1434 parameters)



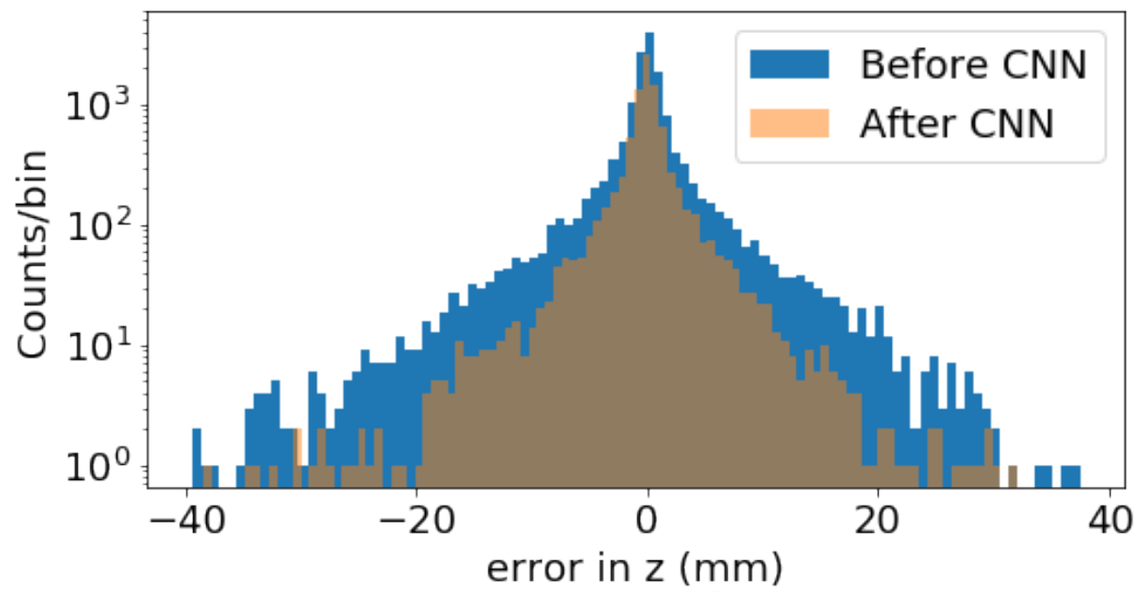
Hardware integration (prototype)

- Attempting to meet timing restrictions of ~10 μs/event (compact CNN)

Smaller CNN (1434 parameters)

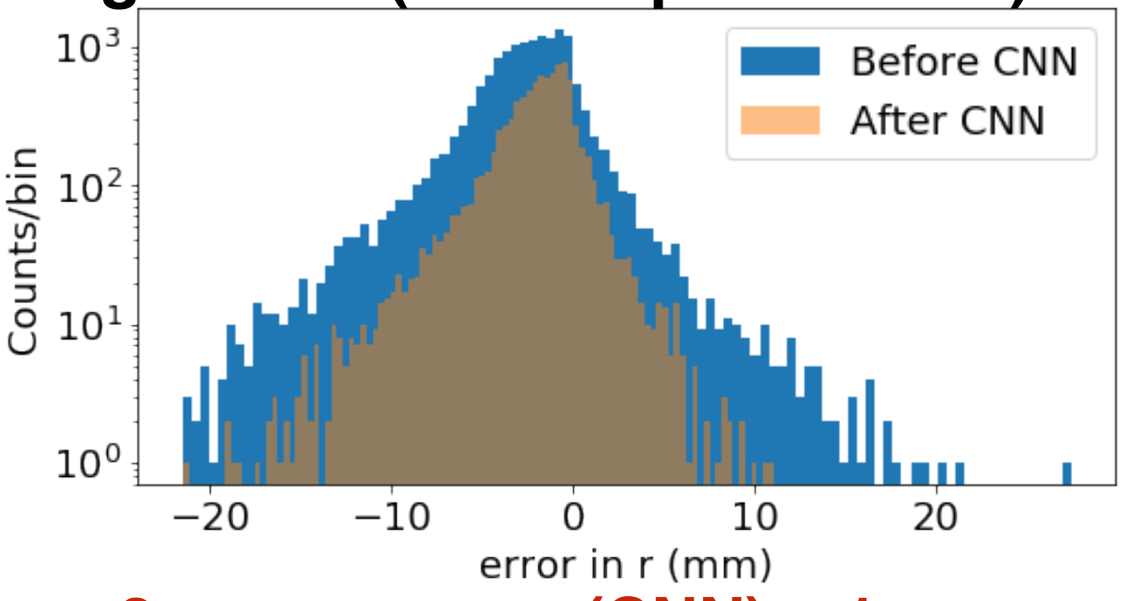


-3 mm < r-error (CNN) < 1 mm

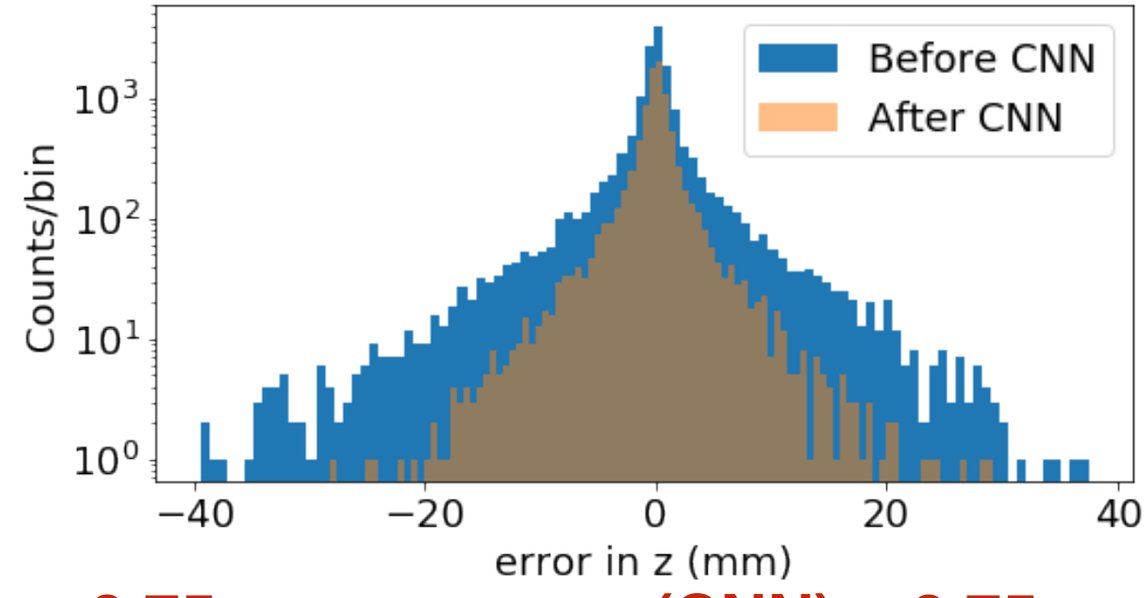


-1.4 mm < z-error (CNN) < 1.4 mm

Larger CNN (840674 parameters)



-3 mm < r-error (CNN) < 1 mm



-0.75 mm < z-error (CNN) < 0.75 mm

Hardware integration (prototype)

- Attempting to meet timing restrictions of ~10 μs/event (compact CNN)

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.742	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
862	2902	482	2522	dataflow

Detail

Instance

		Latency		Interval		
Instance	Module	min	max	min	max	Type
CONV2D_ACT_NoP_2_U0	CONV2D_ACT_NoP_2	43	53	35	53	dataflow
POOL2D_NoP_1_U0	POOL2D_NoP_1	15	27	14	26	dataflow
CONV2D_ACT_NoP_1_U0	CONV2D_ACT_NoP_1	74	137	67	137	dataflow
POOL2D_NoP_U0	POOL2D_NoP	45	85	34	74	dataflow
DENSE_NOACT_U0	DENSE_NOACT	9	9	10	10	dataflow
CONV2D_ACT_NoP_U0	CONV2D_ACT_NoP	861	2901	482	2522	dataflow
ReduceWidth_1_U0	ReduceWidth_1	402	402	402	402	none
AddLast_1u_U0	AddLast_1u_s	1	1	1	1	none
DENSE_ACT_U0	DENSE_ACT	4	4	5	5	dataflow
ExtractPixels_U0	ExtractPixels	9	9	9	9	none
AppendZeros_U0	AppendZeros	0	0	0	0	none

Loop

N/A

Xilinx Vivado timing report (net <u>similar</u> to small CNN)

- Reported times in clock cycles (10 ns)
- Slowest step appears to be the first convolutional layer

Hardware integration (prototype)

- Attempting to meet timing restrictions of ~10 μs/event (compact CNN)

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.742	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
862	2902	482	2522	dataflow

Detail

Instance

		Latency		Interval		
Instance	Module	min	max	min	max	Type
CONV2D_ACT_NoP_2_U0	CONV2D_ACT_NoP_2	43	53	35	53	dataflow
POOL2D_NoP_1_U0	POOL2D_NoP_1	15	27	14	26	dataflow
CONV2D_ACT_NoP_1_U0	CONV2D_ACT_NoP_1	74	137	67	137	dataflow
POOL2D_NoP_U0	POOL2D_NoP	45	85	34	74	dataflow
DENSE_NOACT_U0	DENSE_NOACT	9	9	10	10	dataflow
CONV2D_ACT_NoP_U0	CONV2D_ACT_NoP	861	2901	482	2522	dataflow
ReduceWidth_1_U0	ReduceWidth_1	402	402	402	402	none
AddLast_1u_U0	AddLast_1u_s	1	1	1	1	none
DENSE_ACT_U0	DENSE_ACT	4	4	5	5	dataflow
ExtractPixels_U0	ExtractPixels	9	9	9	9	none
AppendZeros_U0	AppendZeros	0	0	0	0	none

Loop

N/A

Xilinx Vivado timing report (net similar to small CNN)

- Reported times in clock cycles (10 ns)
- Slowest step appears to be the first convolutional layer

Summary

- Use of machine learning in evaluating PET events on-the-fly in PETALO
- An FPGA-based model with timing ~10 μs/event appears to be possible: needs to be quite small
- Many details still to be addressed:
  - impact of neural network in actual reconstruction
  - integration into final electronic readout

Additional Slides

CNN-based approach

- Errors after the 2 pre-processing cuts

1. Remove events with significant charge outside 20x20 window for each gamma
2. Remove events with majority of charge in a single SiPM