DEEP UNDERGROUND NEUTRINO EXPERIMENT

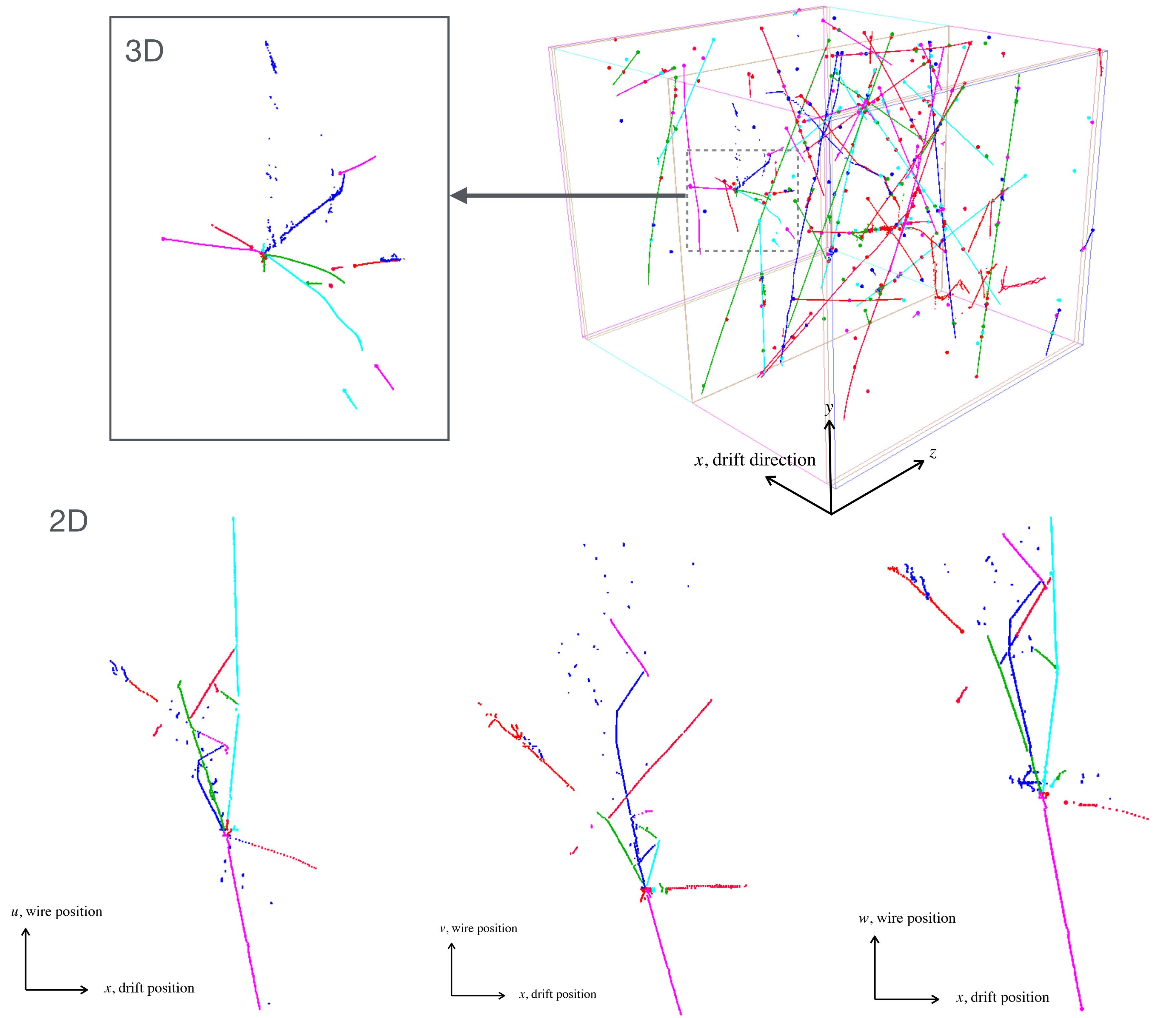
ProtoDUNE Test Beam Hierarchy Performance

Steven Green on behalf of the Pandora Team

24th September 2019

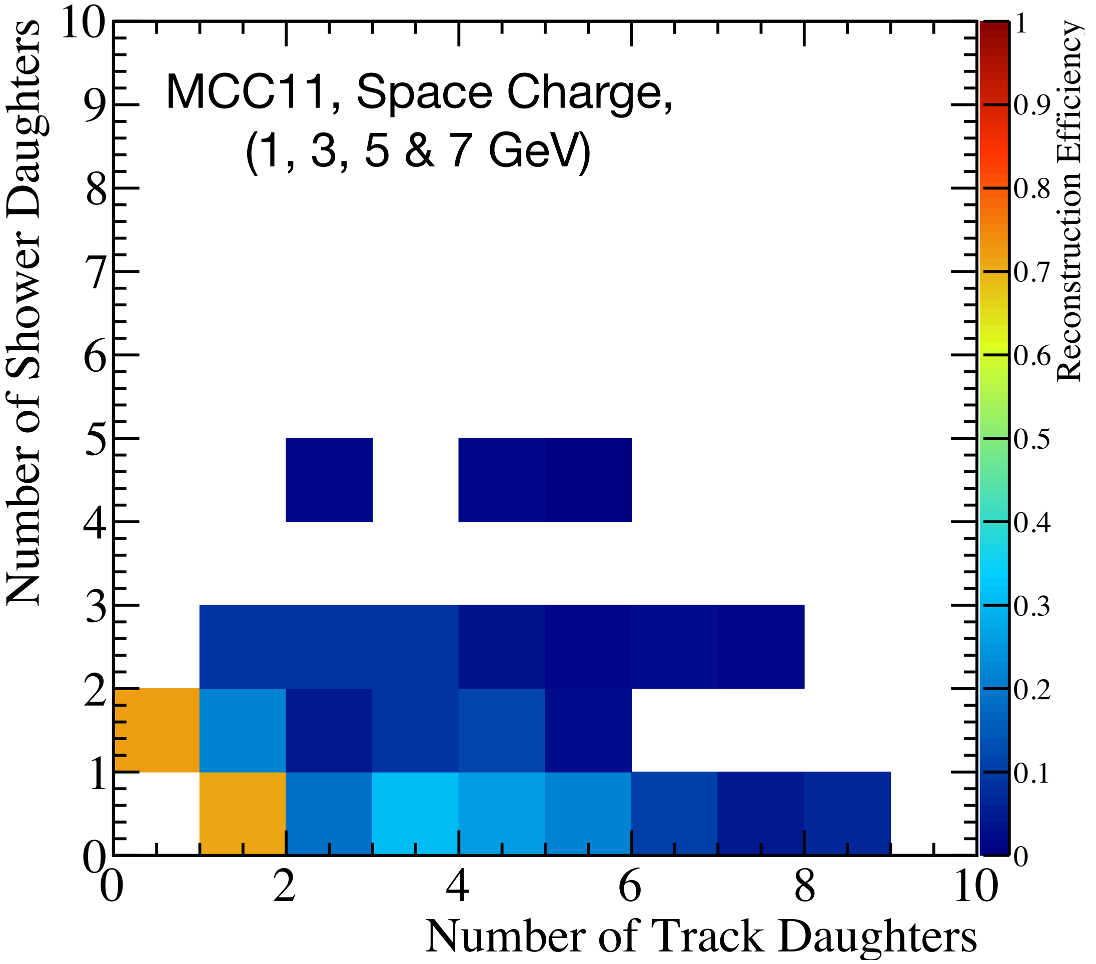
Aim:

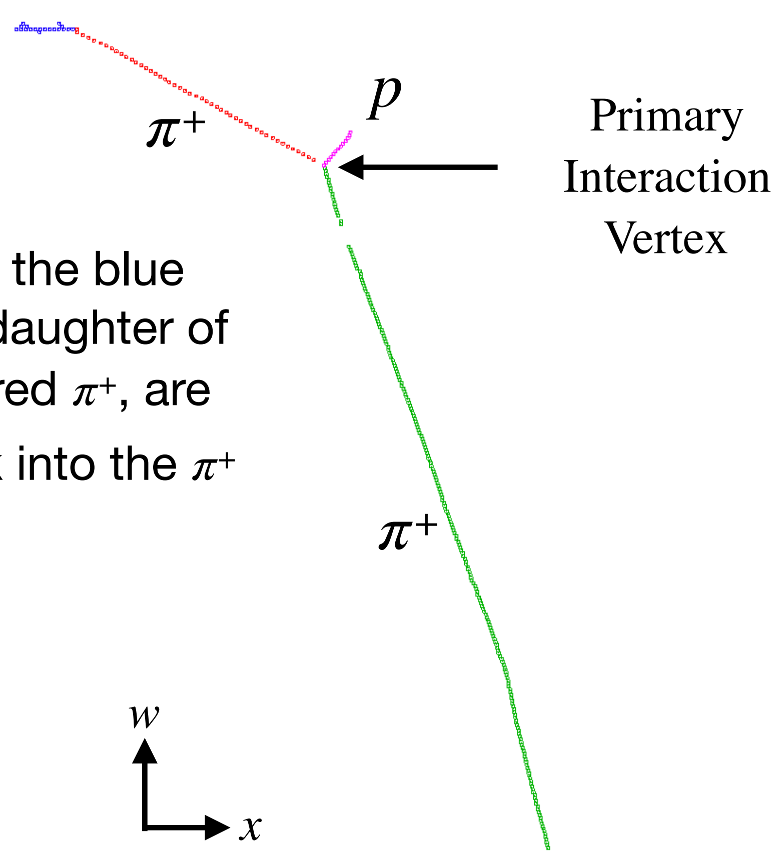
- Go into more detail on the Pandora ProtoDUNE hierarchy reconstruction efficiencies shown previously.
- Demonstrate a first attempt at incorporating deep learning into the Pandora reconstruction.



MCC11 Reconstructed Event with correctly tagged test beam particle

- The reconstructed particle hierarchy metric is very strict:
  - Every target MC particle in the event (*including the parent*) must to be reconstructed and matched to a single reconstructed particle.
  - Each particle has to be identified as a test beam particle and not a cosmic-ray muon.
  - The reconstructed particles must belong to the same hierarchy i.e. not split.

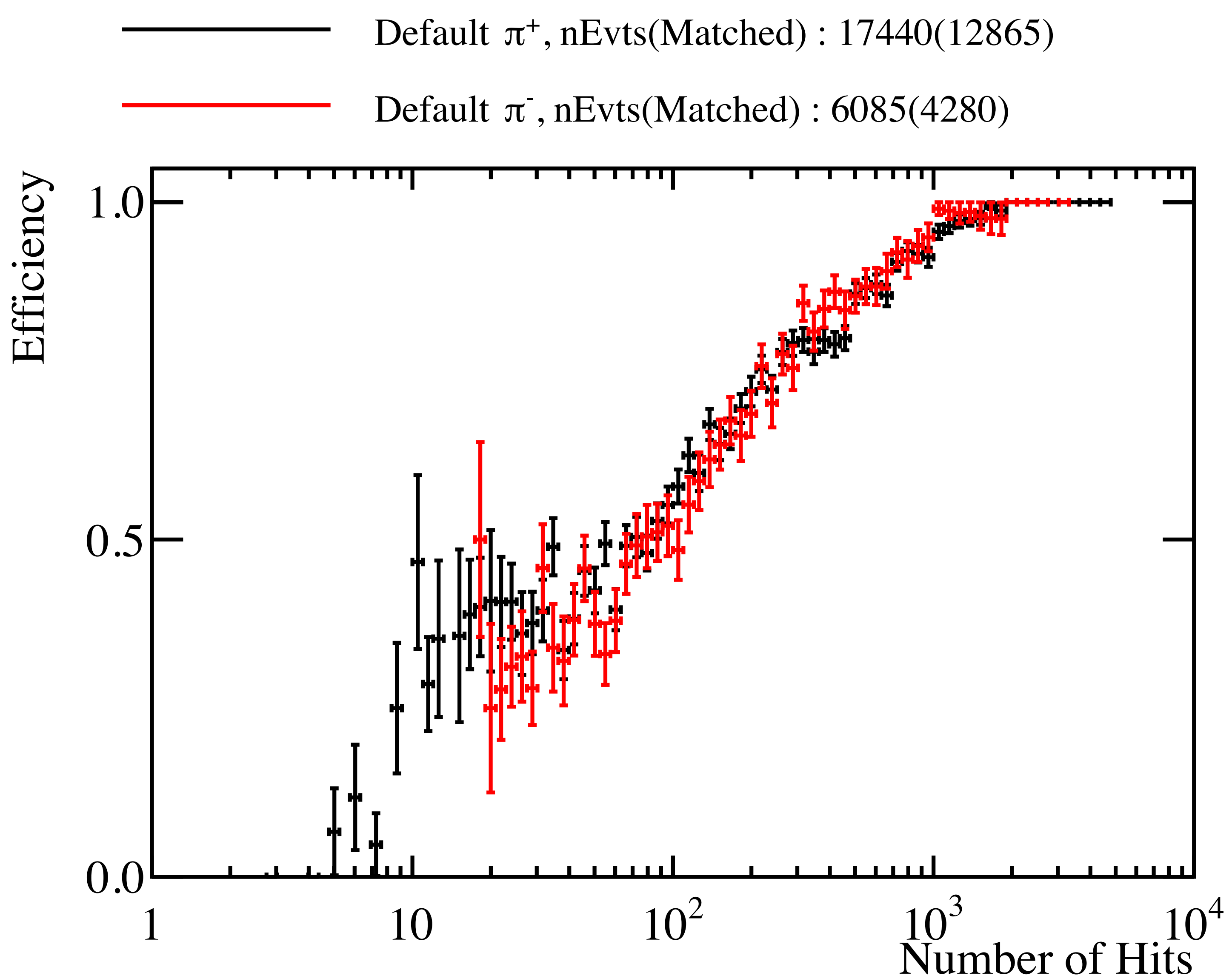


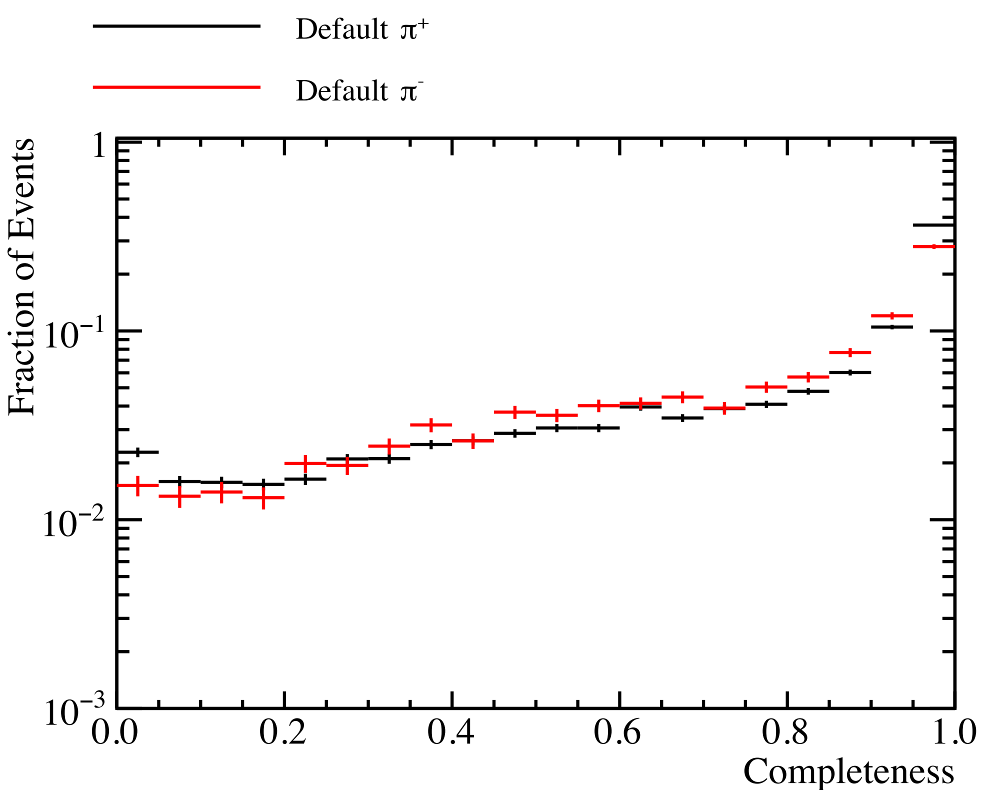


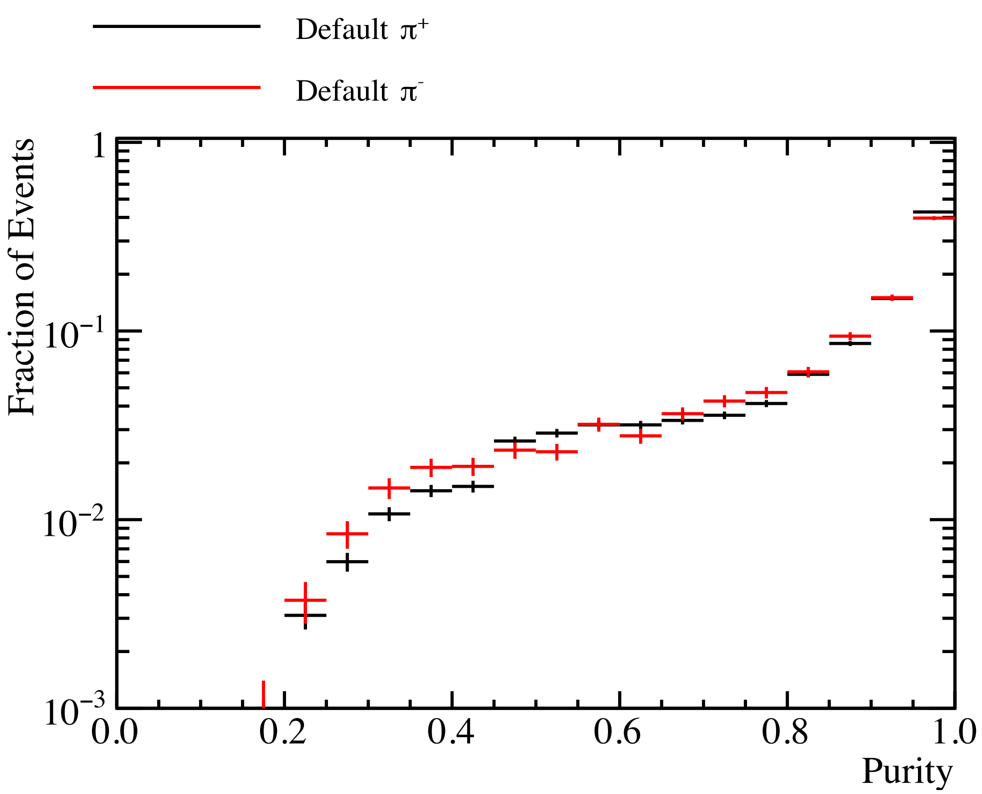
Hits from the blue particle, a daughter of the scattered π^+, are folded back into the π^+

Particle	MC Hits	Reco Hits	Shared Hits
Parent : π^+	674 (259, 159, 246)	670 (265, 160, 245)	668 (265, 158, 245)
Scattered: π^+	168 (58, 43, 67)	168 (58, 43, 67)	168 (58, 43, 67)
p	34 (8, 13, 13)	35 (11,11,13)	32 (8,11,13)

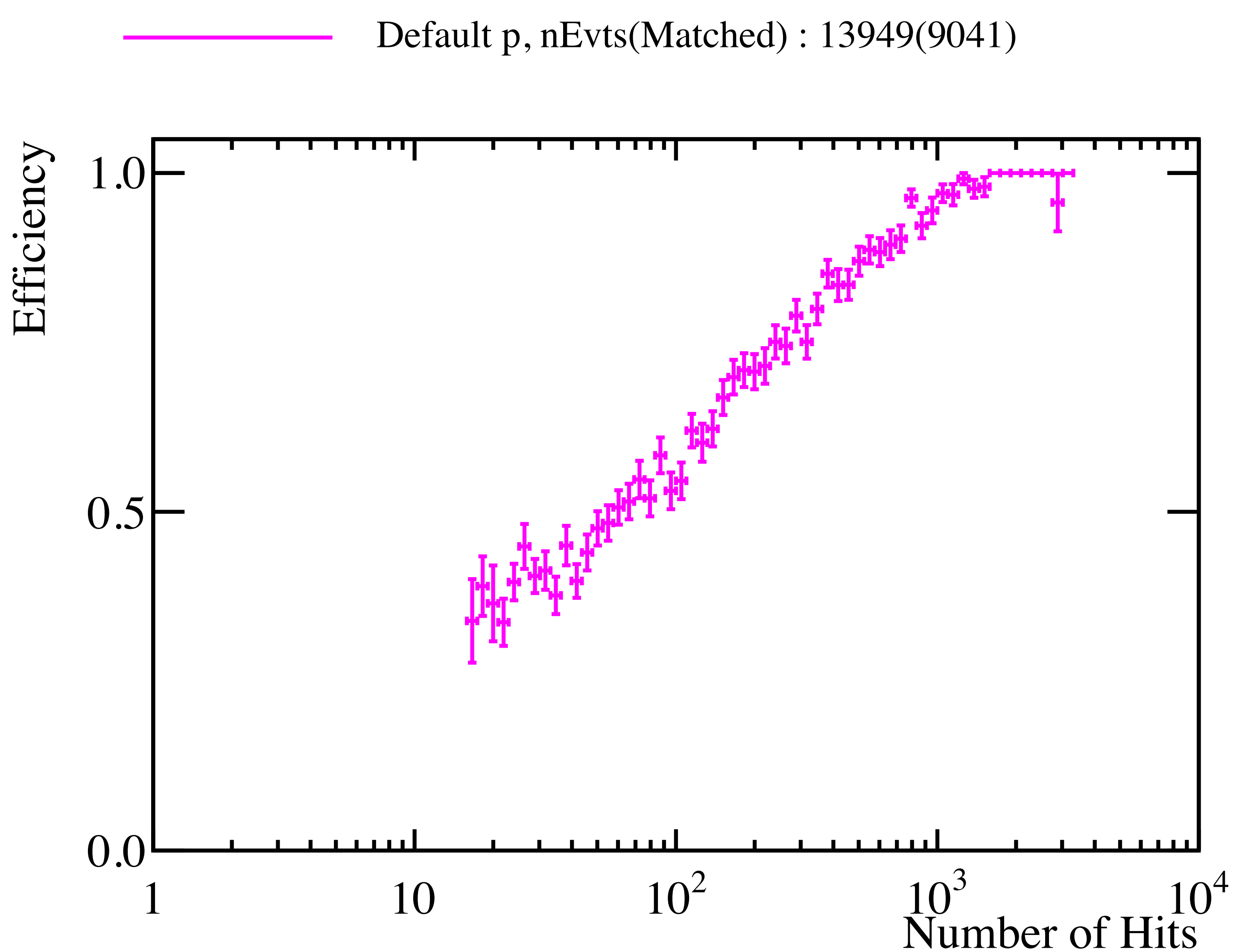
- The previous metric considers all particles in the interaction, but we can also look at individual particle types.
- Typically good reconstruction efficiency for tracks, especially for tracks with large numbers of hits.

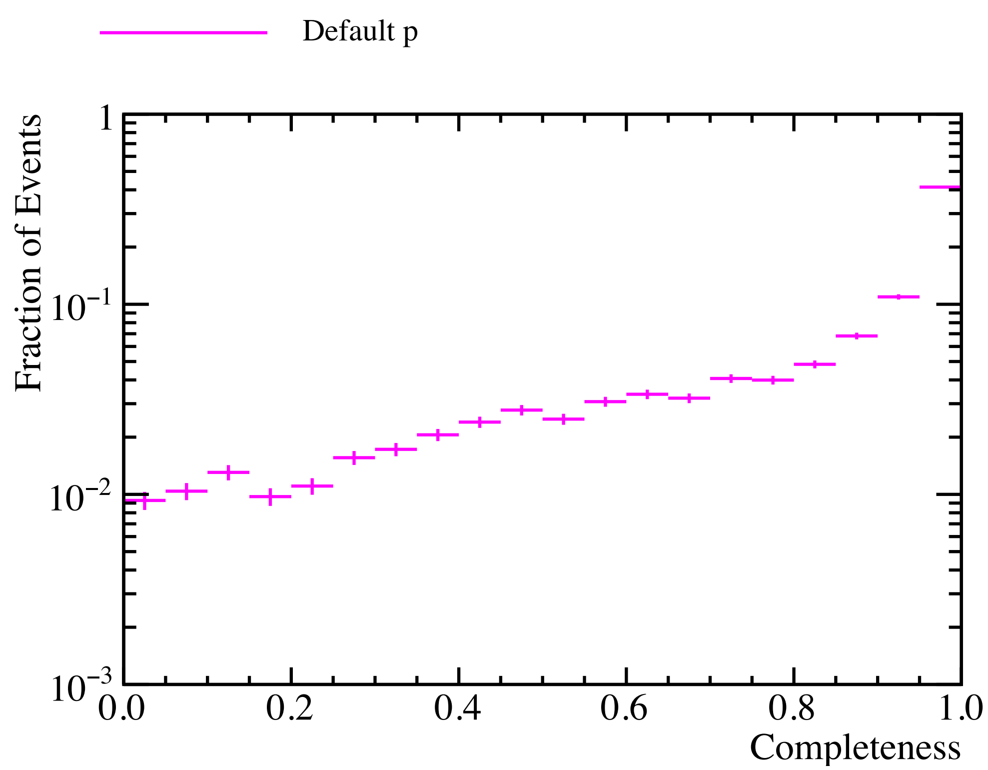


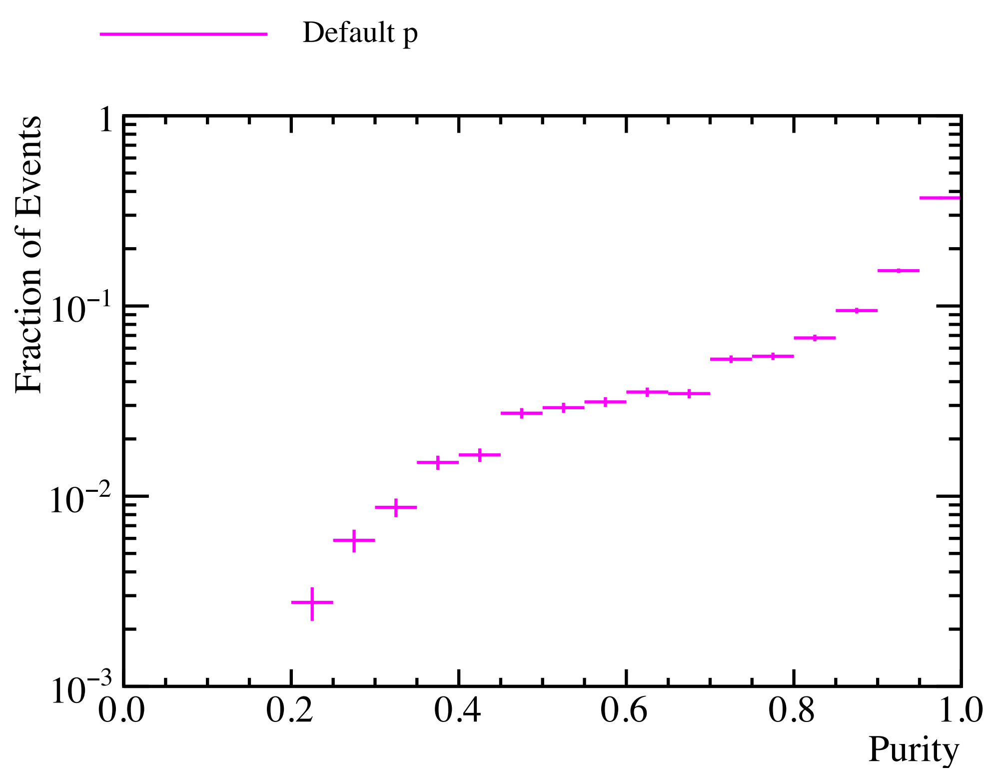




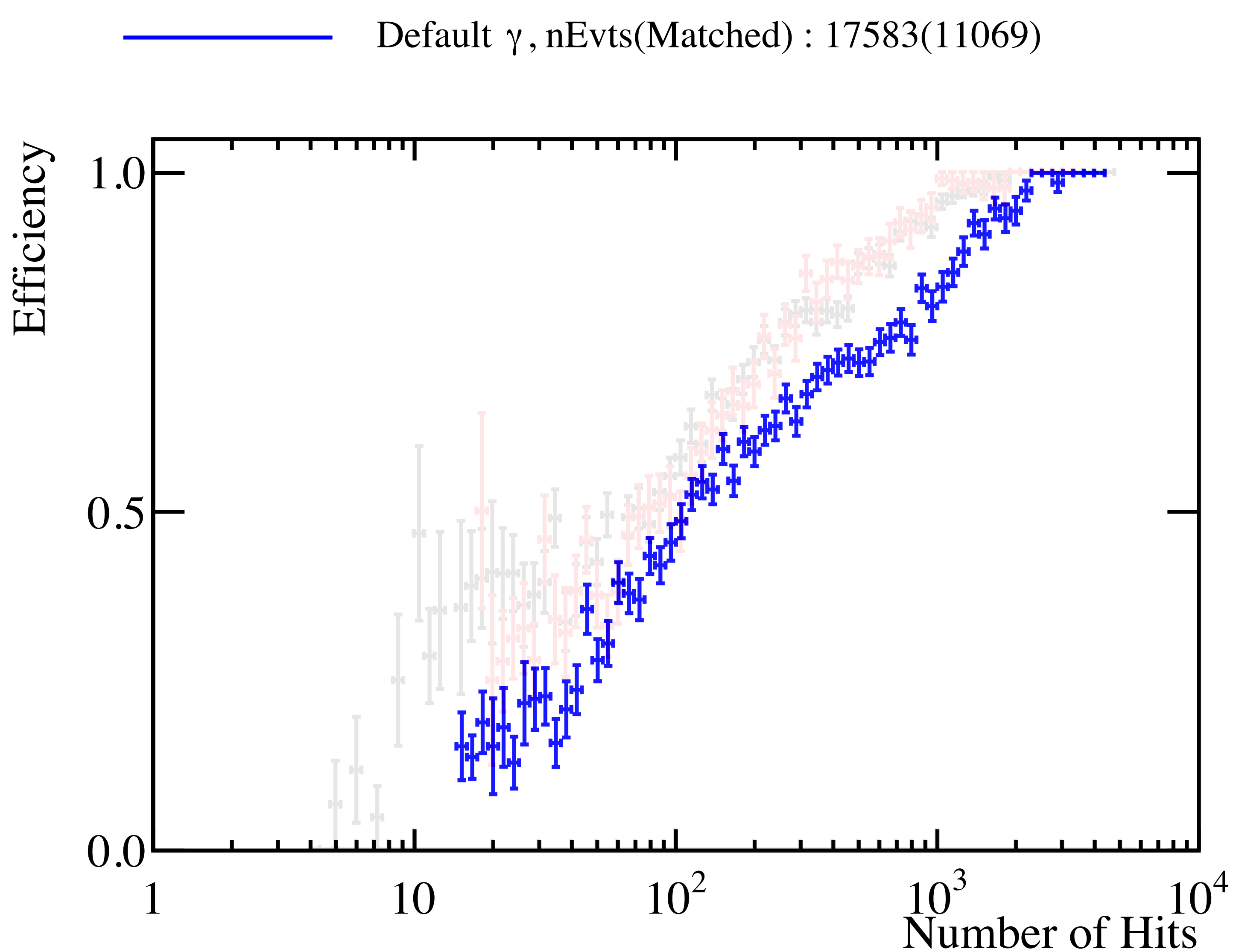
- Proton reconstruction hierarchy metrics mirror that seen for charged pions as both species are track like.

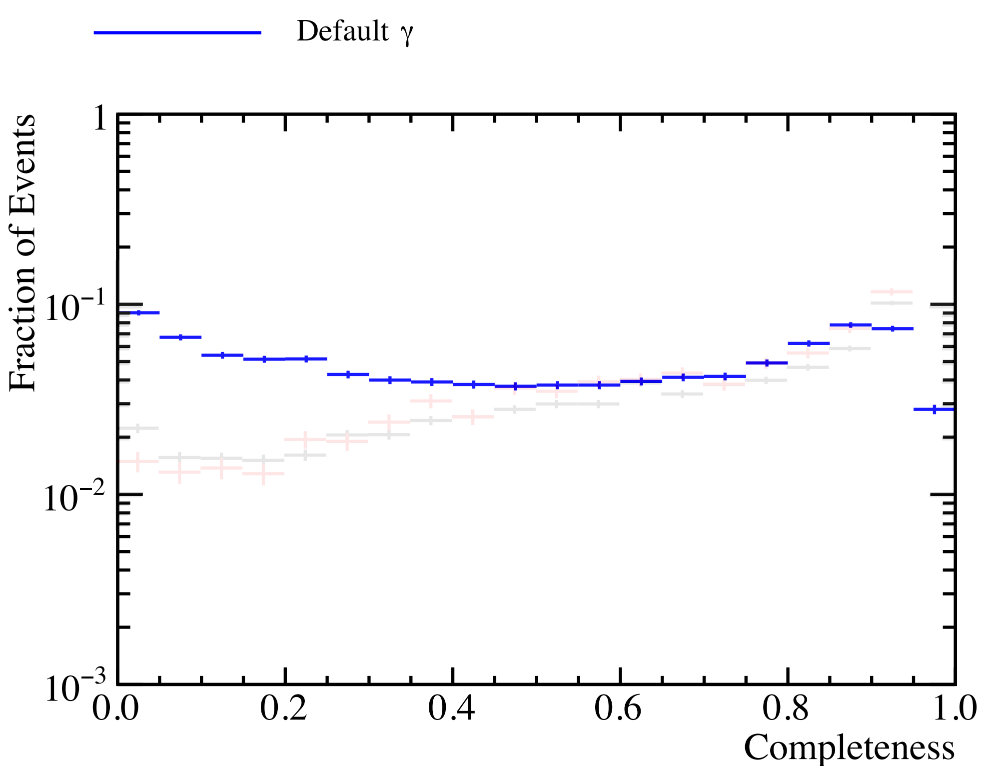


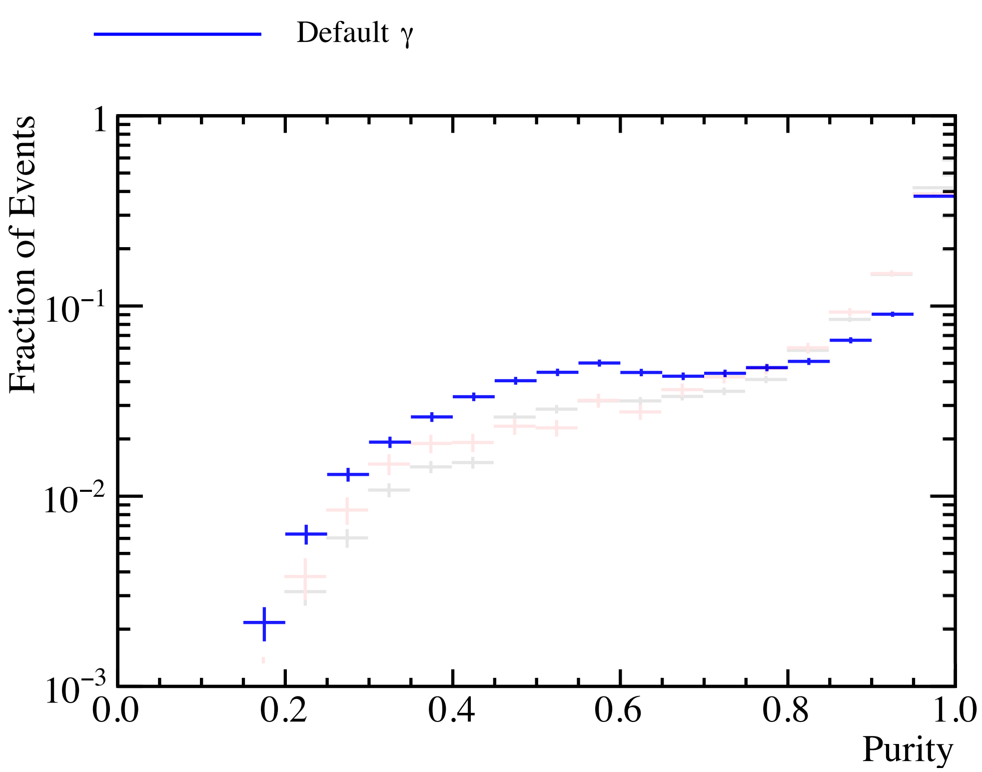




- There is a drop in reconstruction efficiency for ɣs.
- Mainly due to overlapping ɣs from boosted $\pi^0 \rightarrow$ ɣɣ events.
- There is also a peak at low completeness due to ɣs being reconstructed as tracks.

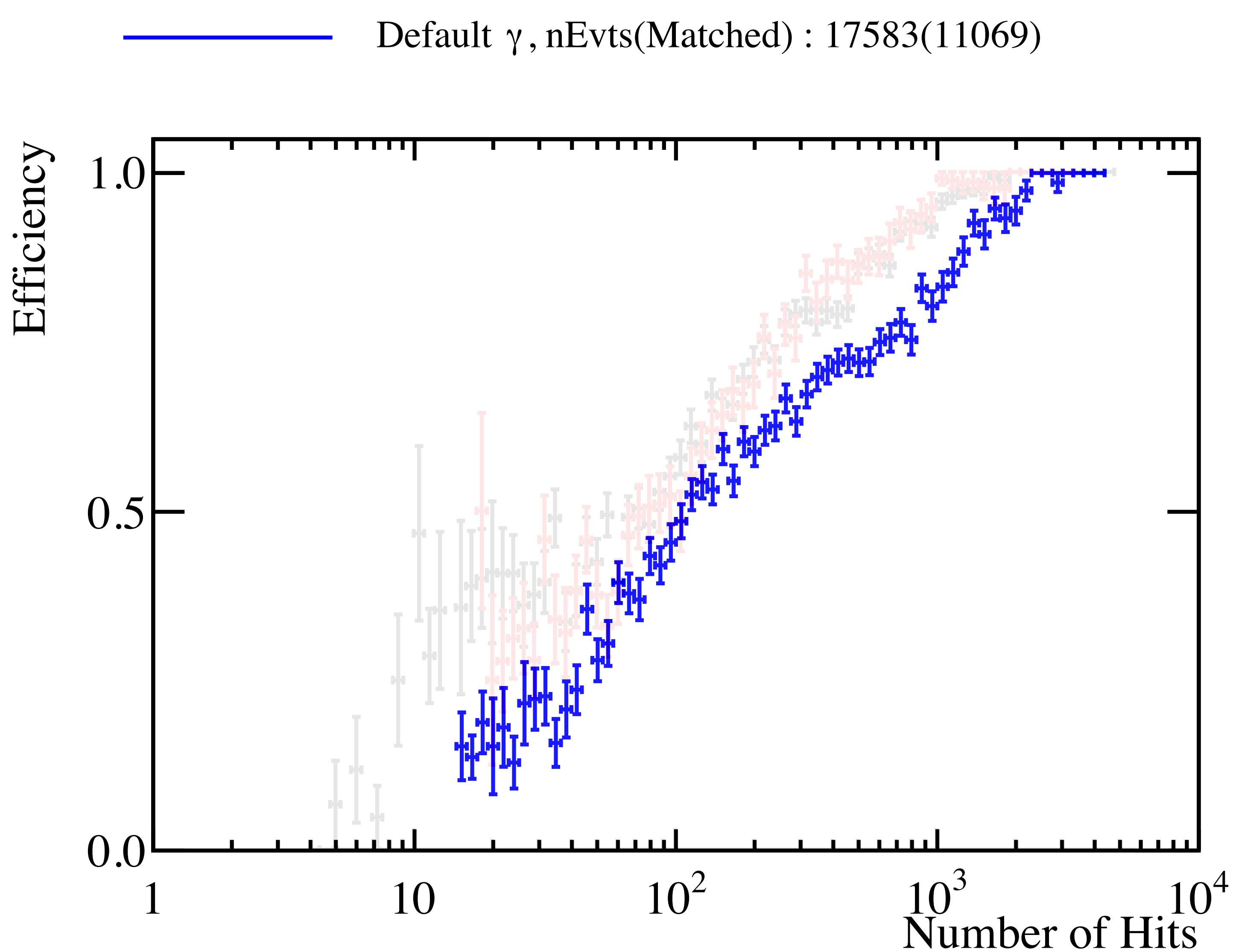


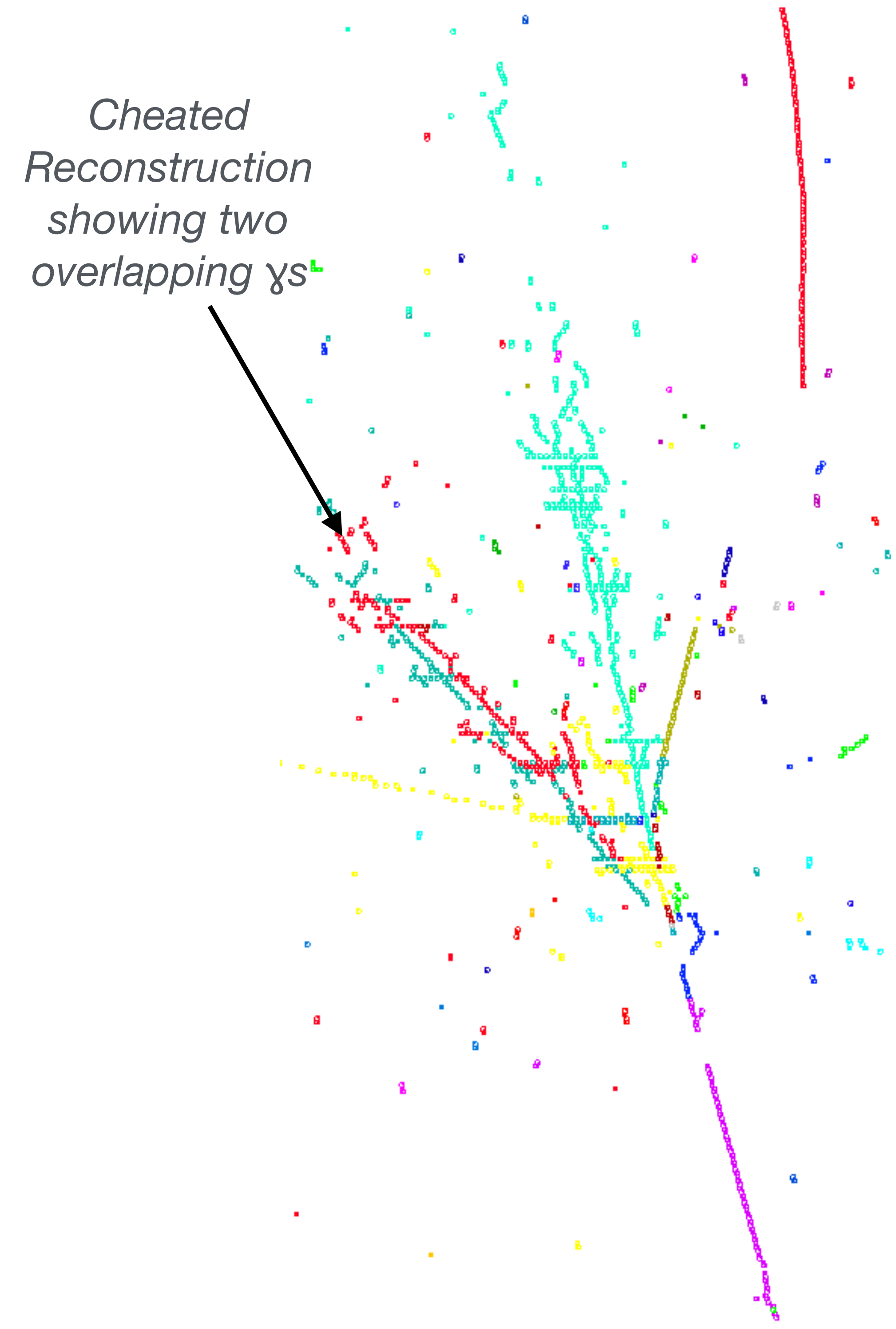




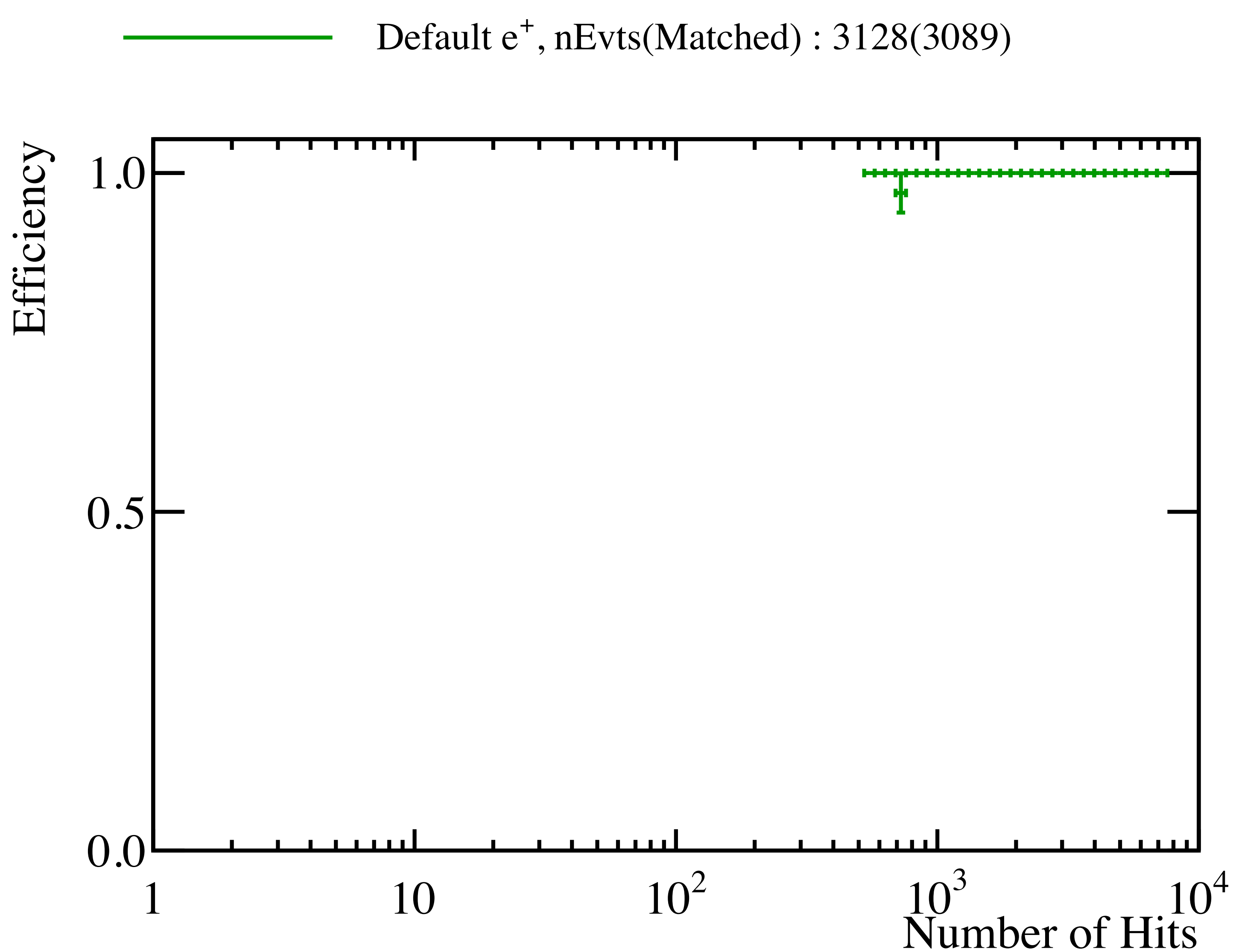
Test Beam Hierarchy : Photons

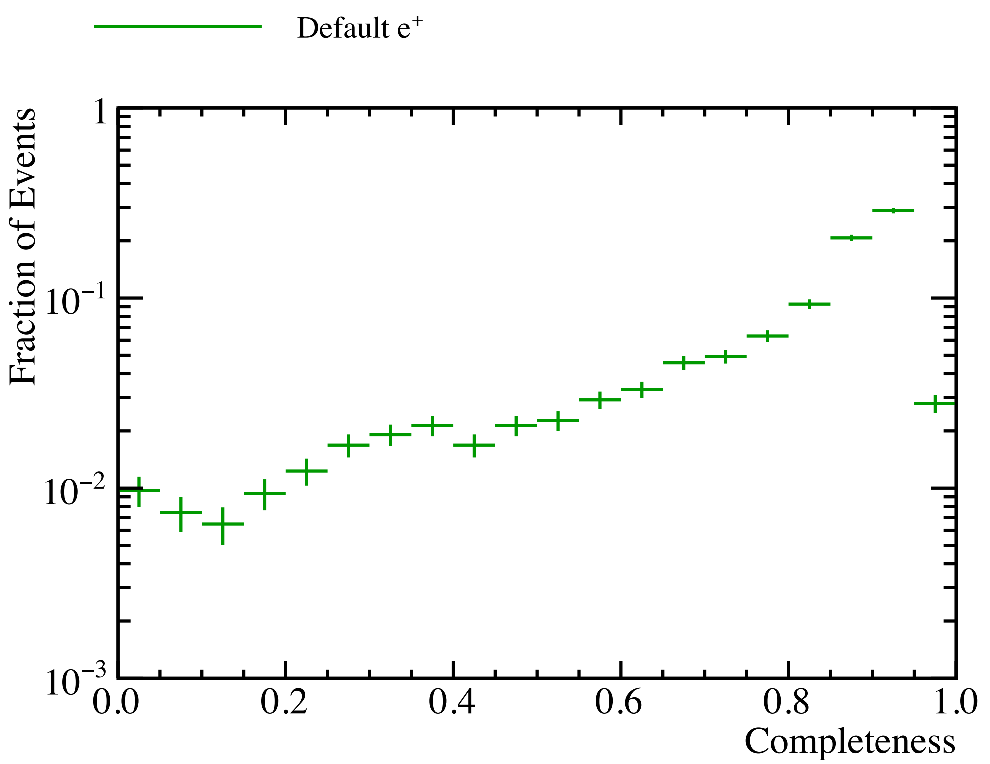
- There is a drop in reconstruction efficiency for ɣs.
- Mainly due to overlapping ɣs from boosted $\pi^0 \rightarrow$ɣɣ events.
- There is also a peak at low completeness due to ɣs being reconstructed as tracks.

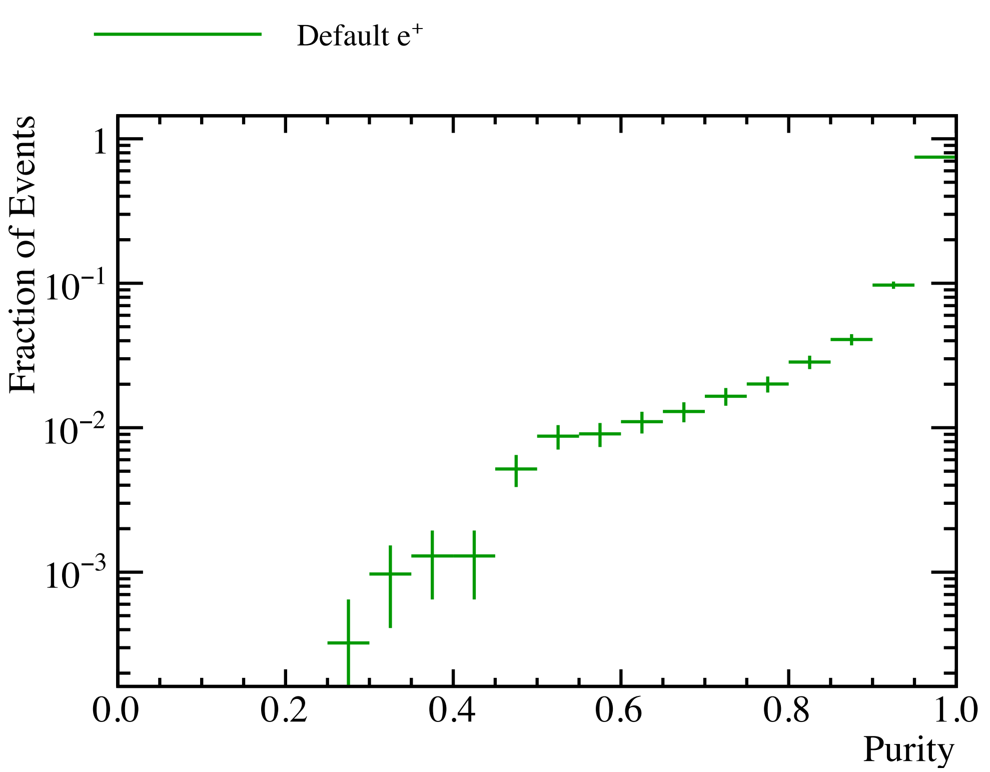




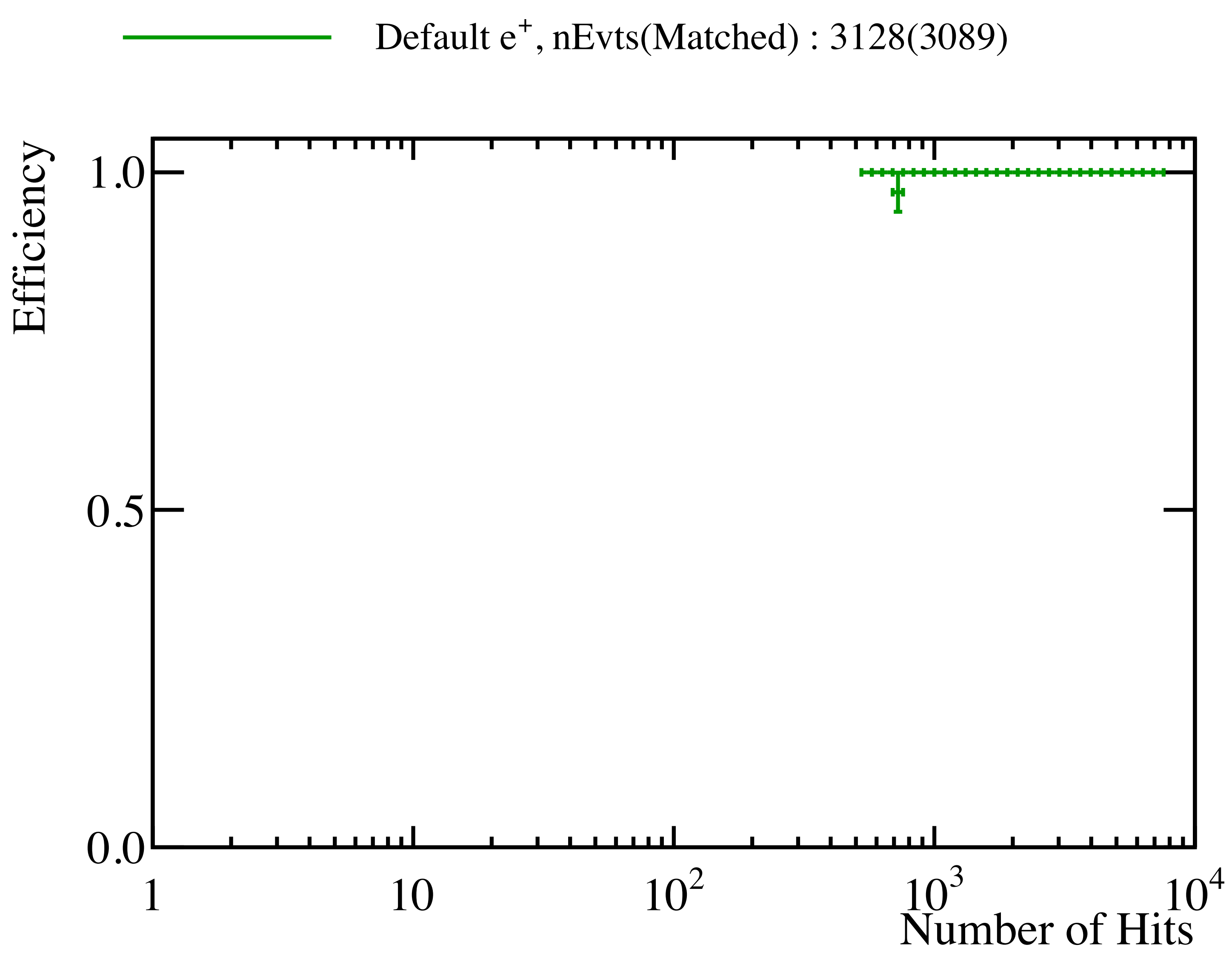
- Electron/Positron showers are significantly easier to reconstruct well as they're independent (i.e. primary particles) rather than belonging to a complex particle hierarchy.

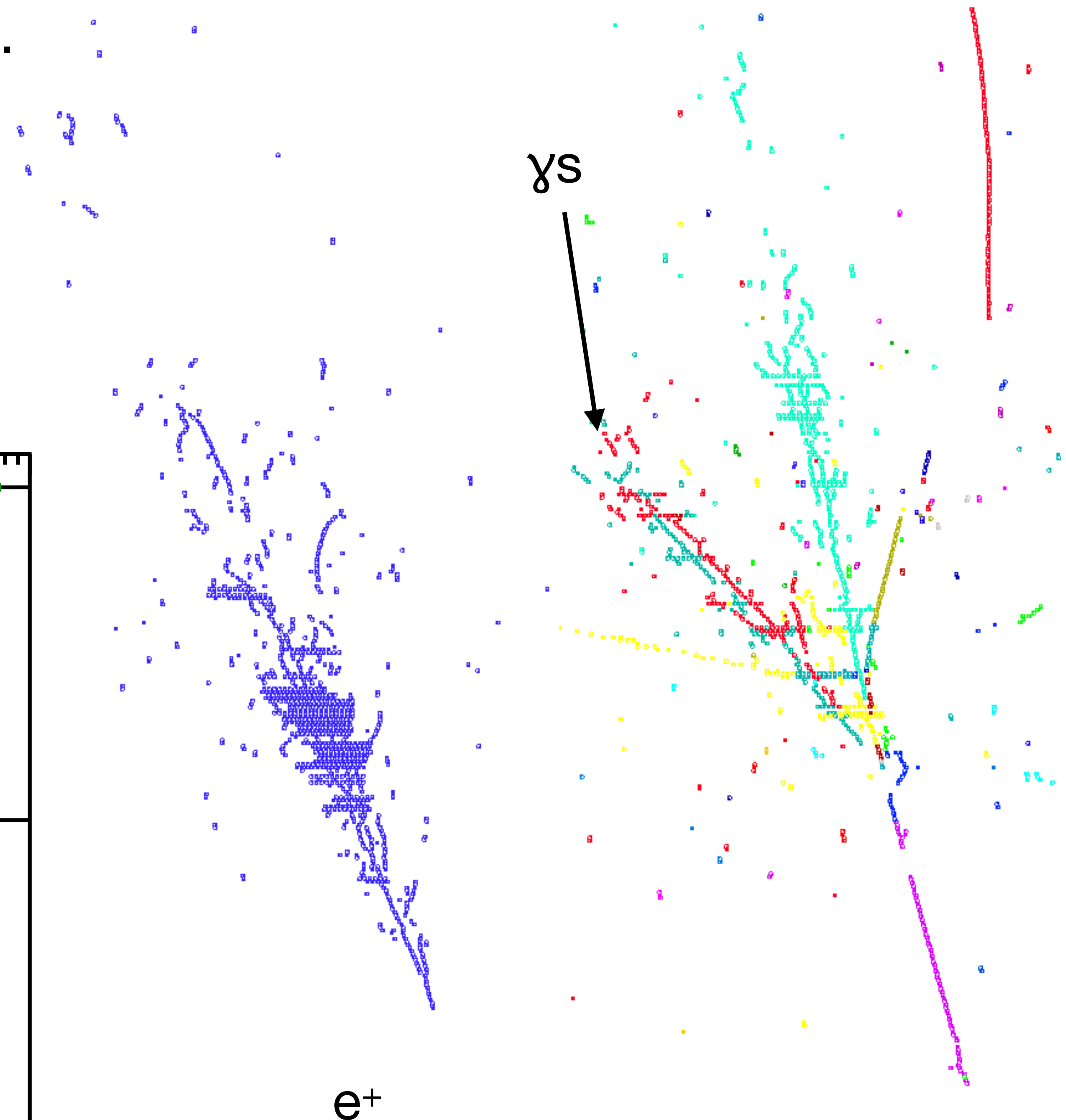






- Electron/Positron showers are significantly easier to reconstruct well as they're independent (i.e. primary particles) rather than belonging to a complex particle hierarchy.

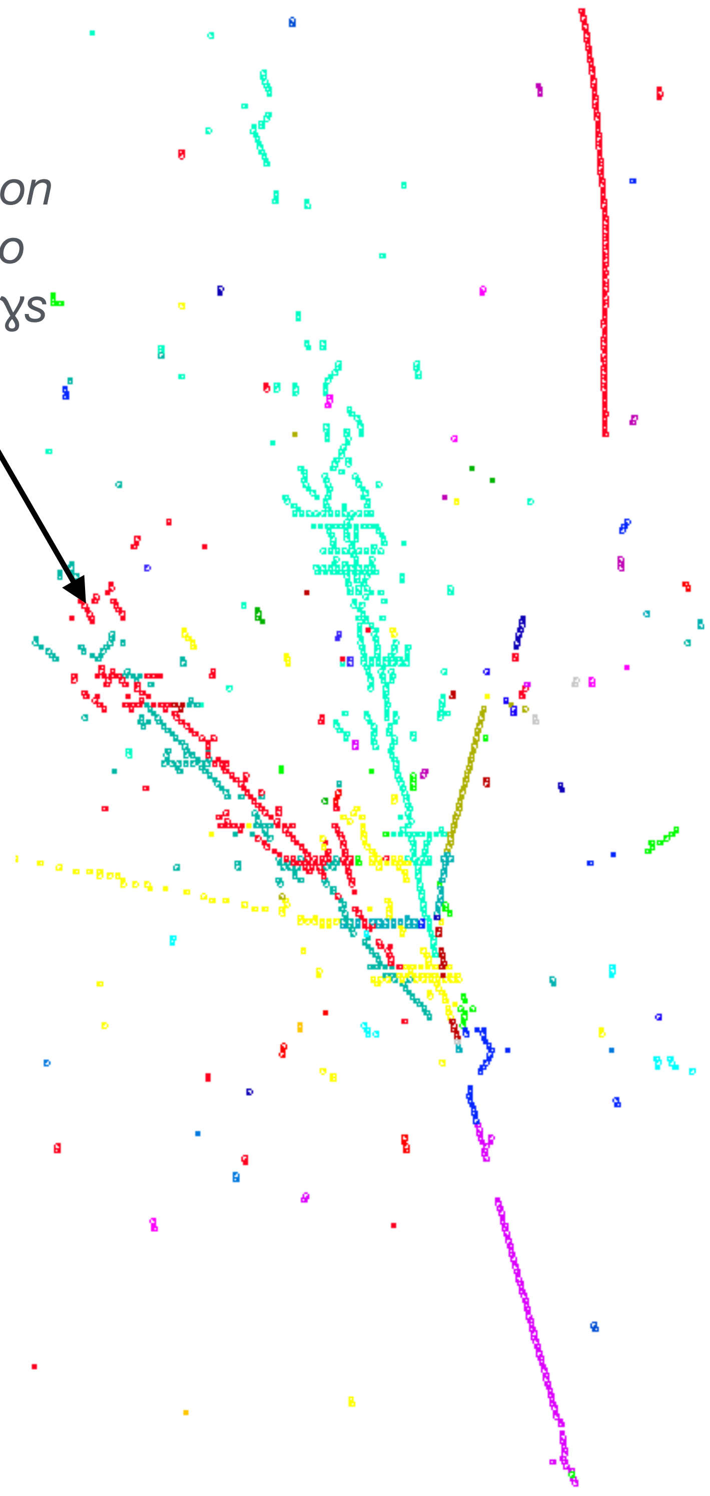




Test Beam Hierarchy : Summary

- We now have a lot of information available about the quality of the reconstructed particle hierarchy.
- From these new metrics it is clear that ɣ reconstruction is one of the biggest challenges facing the reconstruction.
- In an attempt to improve this we are looking into incorporating deep learning techniques into the Pandora reconstruction.

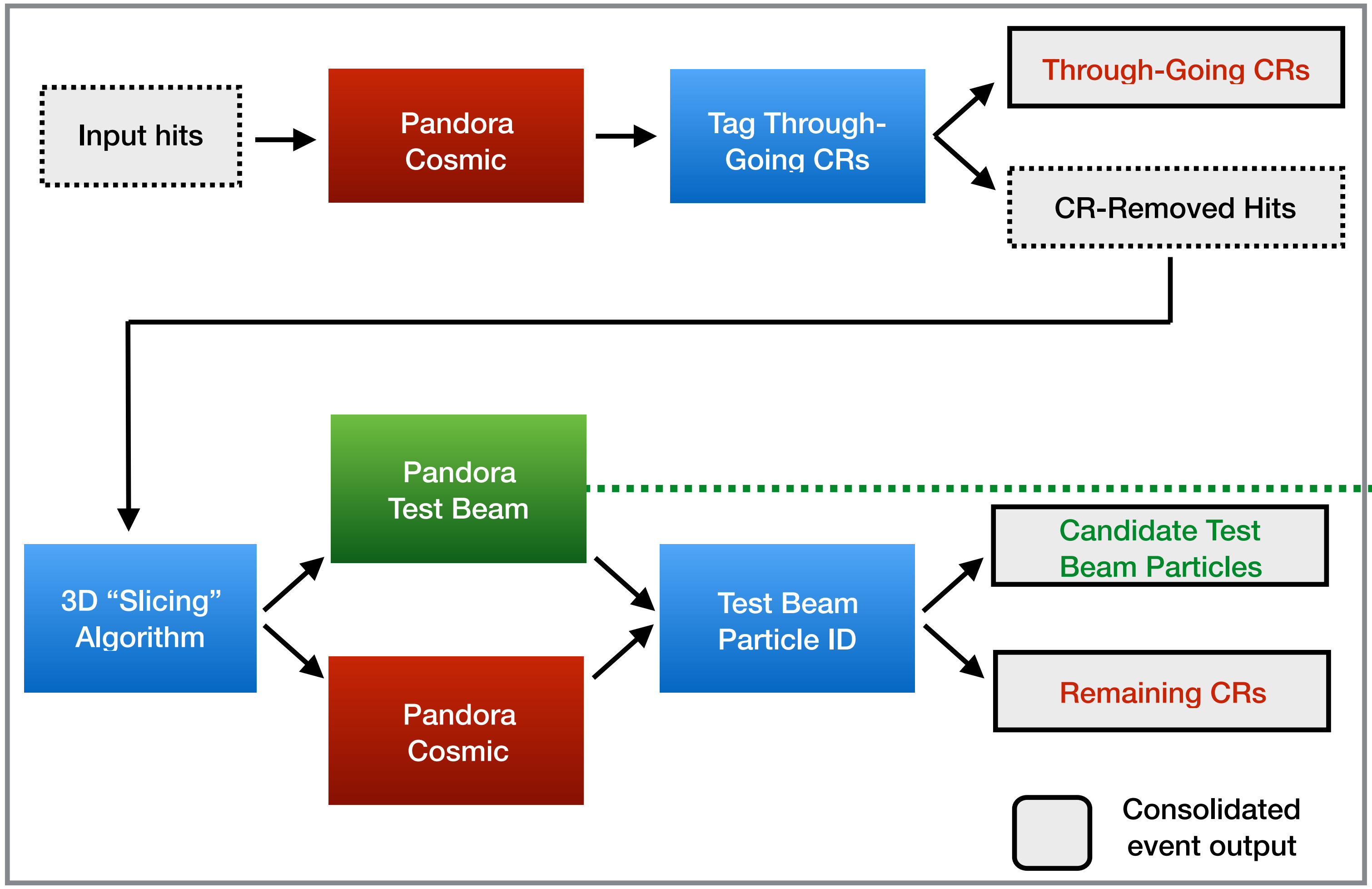
Cheated Reconstruction showing two overlapping ɣs



Semantic Segmentation

- Inside Pandora there are several stages we look to distinguish tracks and showers.

Pandora Consolidated Reconstruction



PandoraCosmic:
Algorithm chain targeting reconstruction of cosmic-rays

PandoraTestBeam:
Algorithm chain targeting reconstruction of test beam interactions

Slicing:
Dividing up of the whole event into regions (slices) containing hits originating from a single parent particle.

Pandora Test Beam Reconstruction Chain

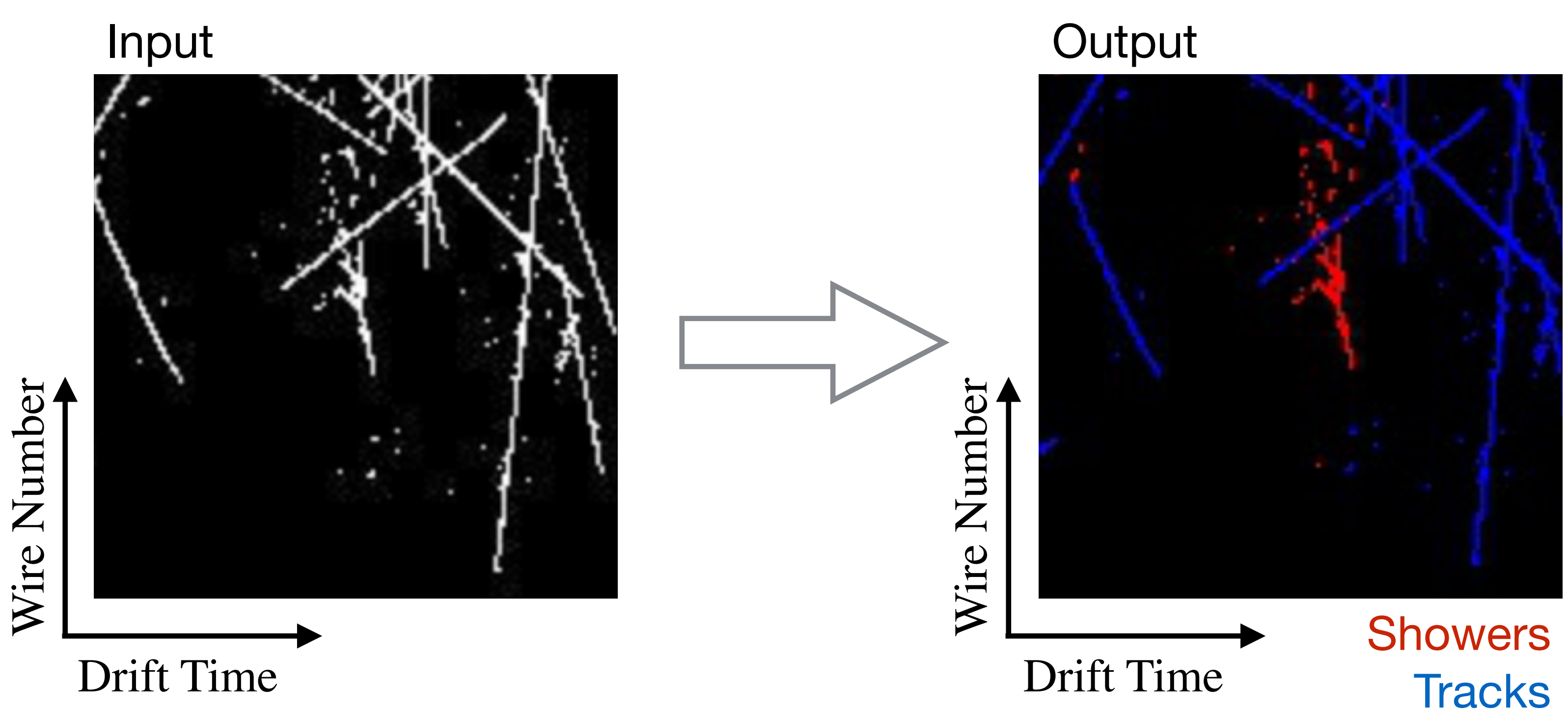
- 2D Reconstruction
- Vertex Finding
- 3D Track Matching
- 2D Shower Growing & 3D Shower Matching
- Repeat 3D Track Matching
- 2D & 3D Recovery Algorithms
- 3D Hit Creation
- Hierarchy Building

The 3D track matching is repeated as, at the second stage, the shower like clusters have been removed.

Semantic Segmentation

- In ProtoDUNE we currently distinguish track and shower like clusters by placing cuts on some basic properties of those clusters (e.g. length, curvature).
- Now we want to see if deep learning can be used in place of these cuts.

Aim:



Pandora Test Beam Reconstruction Chain

2D Reconstruction

Vertex Finding

3D Track Matching

2D Shower Growing & 3D Shower Matching

Repeat 3D Track Matching

2D & 3D Recovery Algorithms

3D Hit Creation

Hierarchy Building

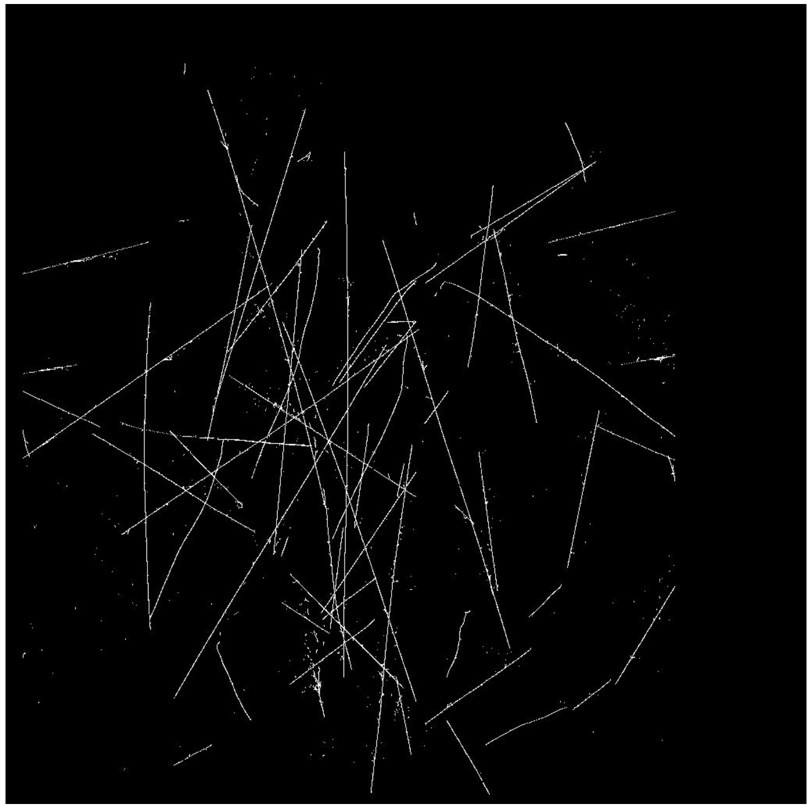
The 3D track matching is repeated as, at the second stage, the shower like clusters have been removed.

- For a first attempt at semantic segmentation we will use a UNet architecture implemented in PyTorch.

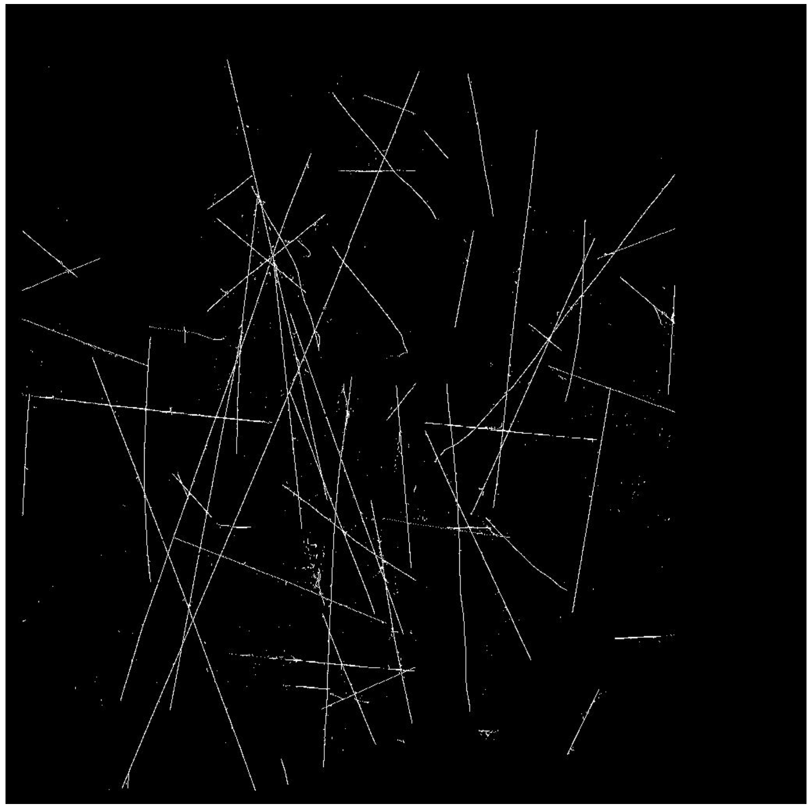
Inputs To Model:

- 1024x1024 image showing the hit positions in the U, V and W views. - Image too big
- The same model is used to classify each view. - Separate models required
- The range of each image (in drift position and wire number) is 980 cm* and is centred such that all hits in the event. - Dynamic image creation needed

U View



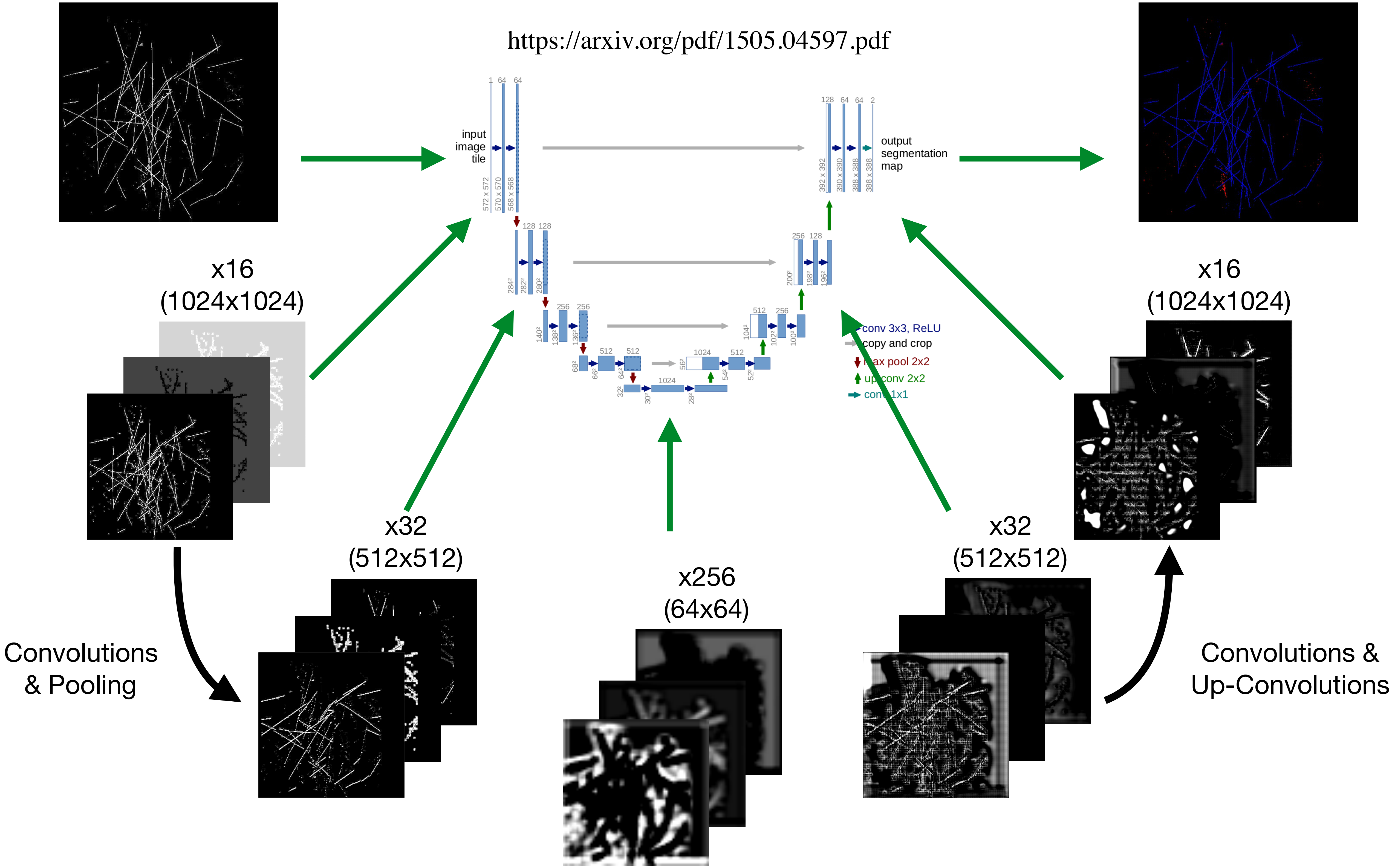
V View



W View

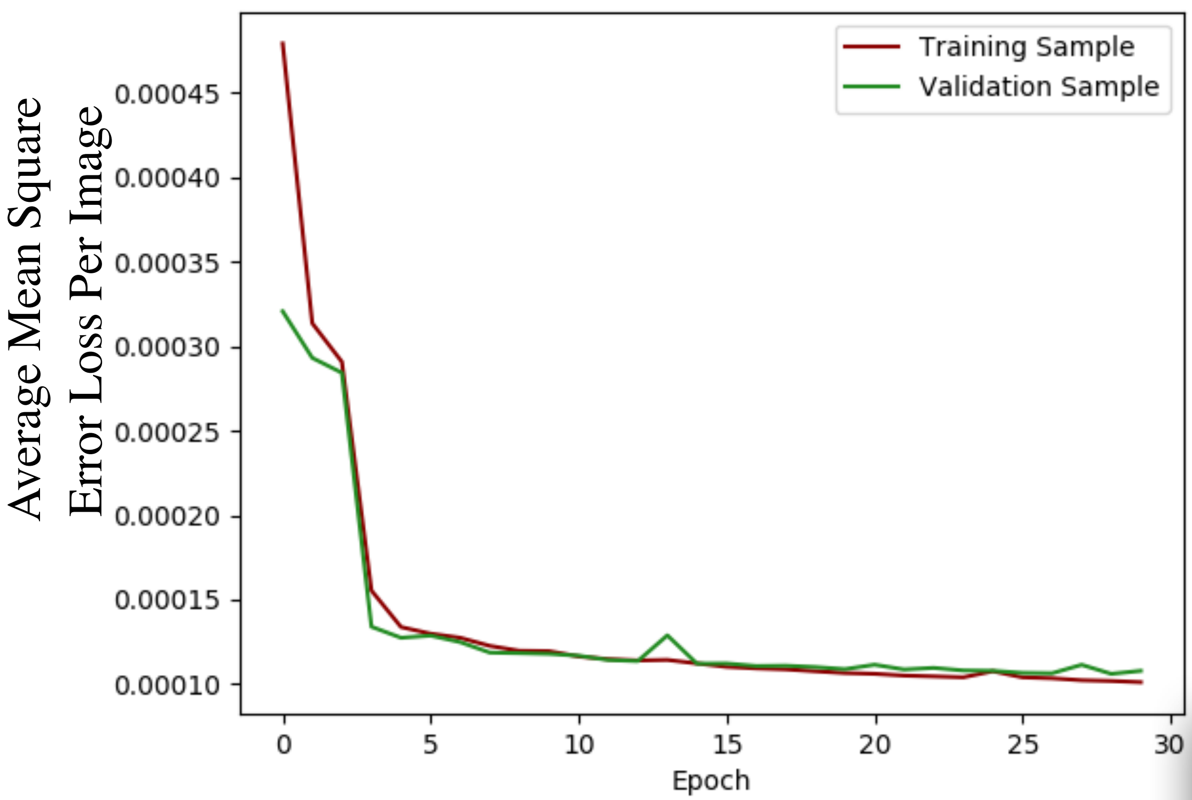
Semantic Segmentation

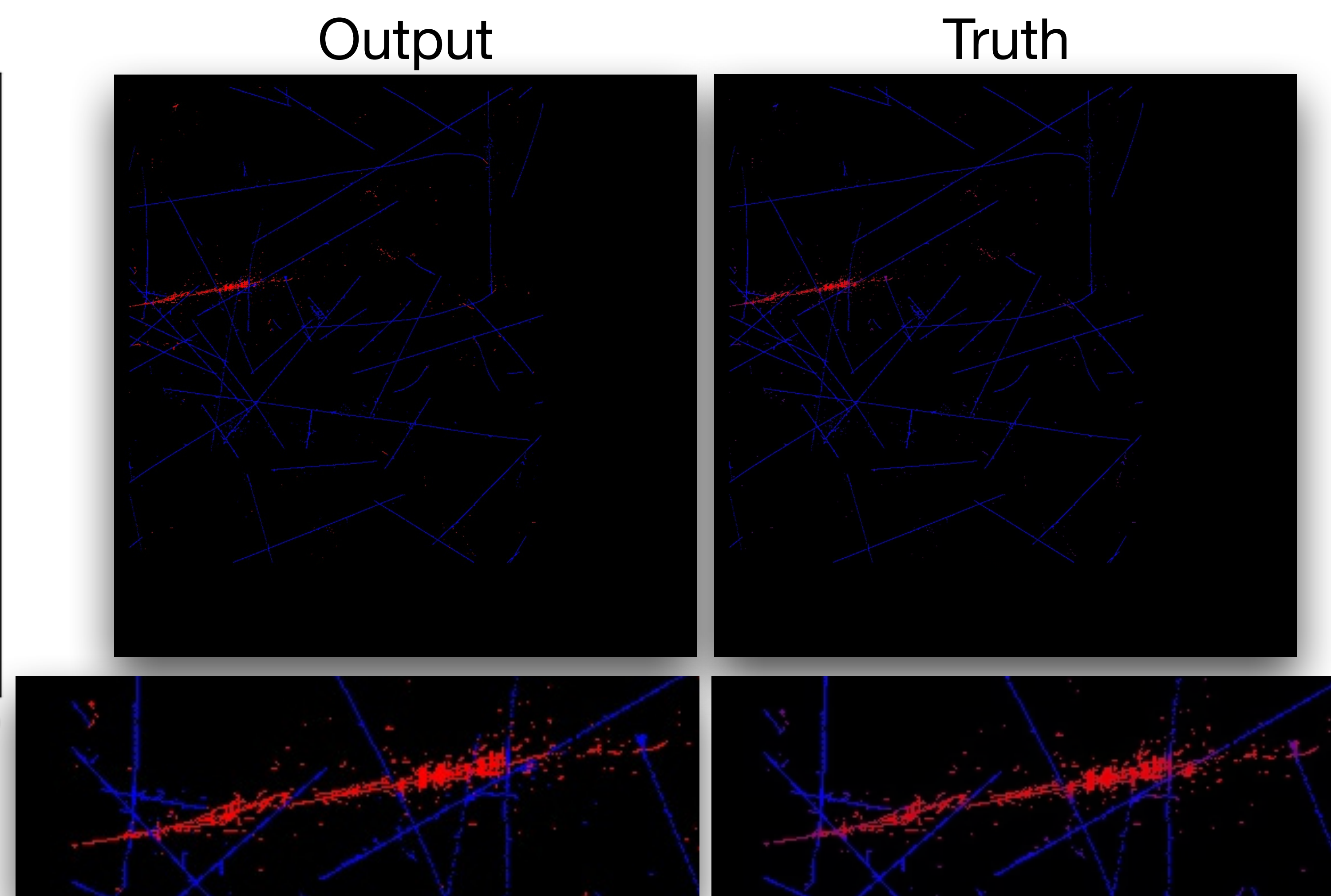
- Brief architecture overview:



Semantic Segmentation

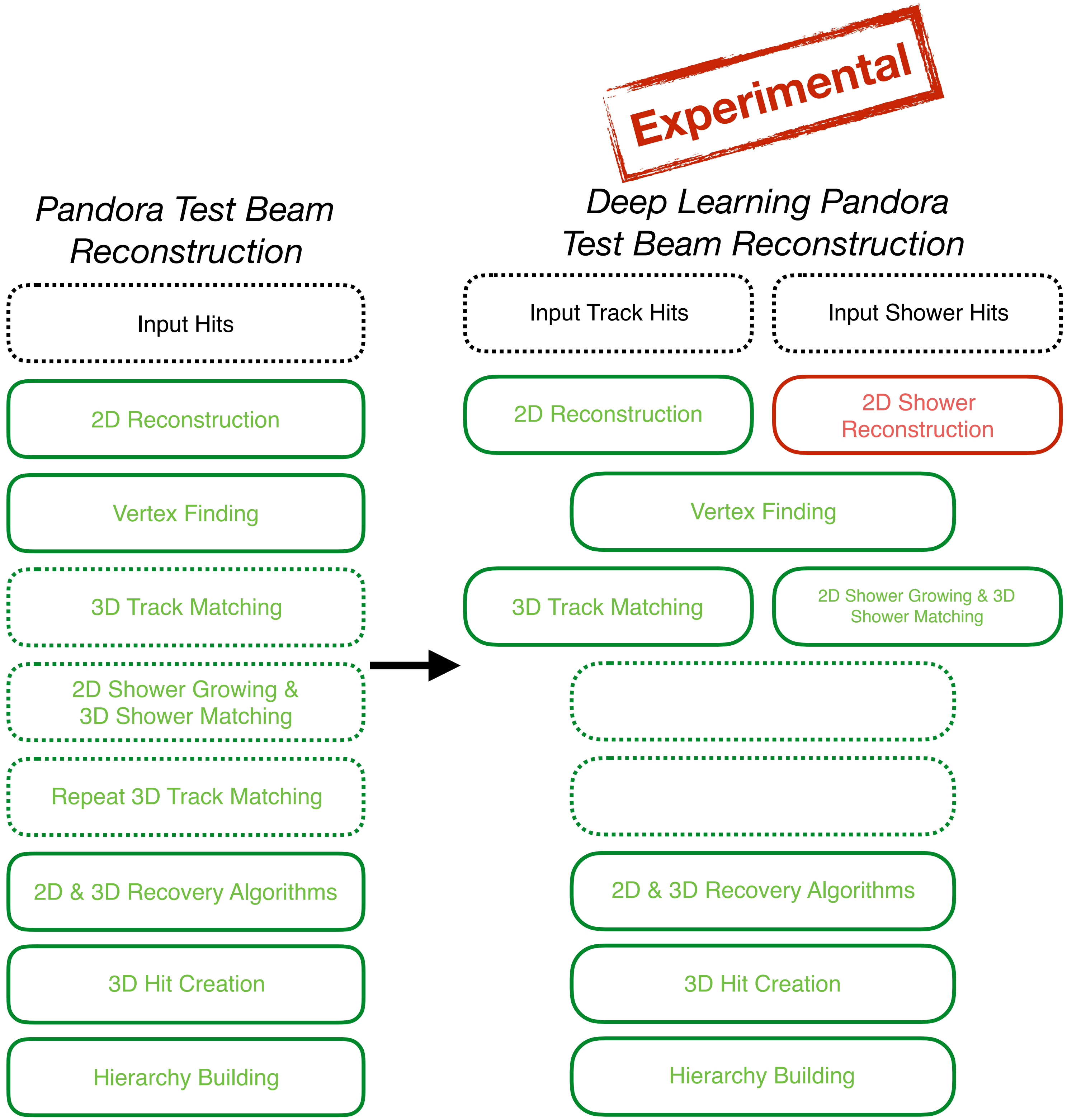
- Trained on MCC11, No Space Charge, (1, 3, 5 & 7 GeV).
- Each image rotated through 90, 180 and 270 degrees to help model learn rotational invariance: ~18,000 images in total.
- 90% of images used for training, 10% for validation.





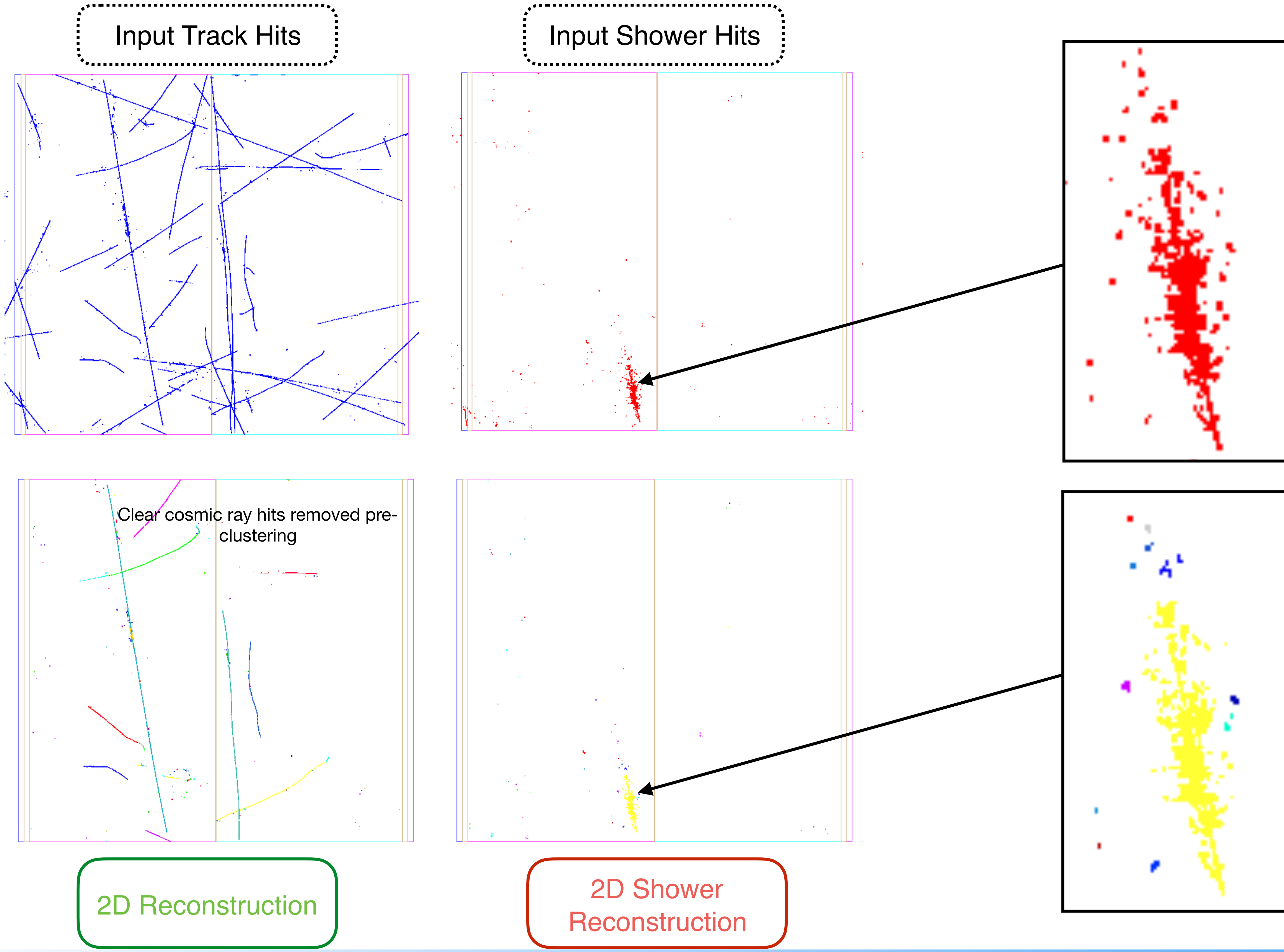
- Model looks promising; it misses some fine details, but picks up prominent features.

- Now we have this model, we will restructure the Pandora test beam reconstruction*:
  - We separately cluster 2D hits for showers and tracks.
  - The initial 2D reconstruction is strongly track orientated, while the new 2D shower reconstruction is proximity based (merge hits together if within 5cm).
  - We no longer repeat the 3D track matching as track and shower clusters are already separate.



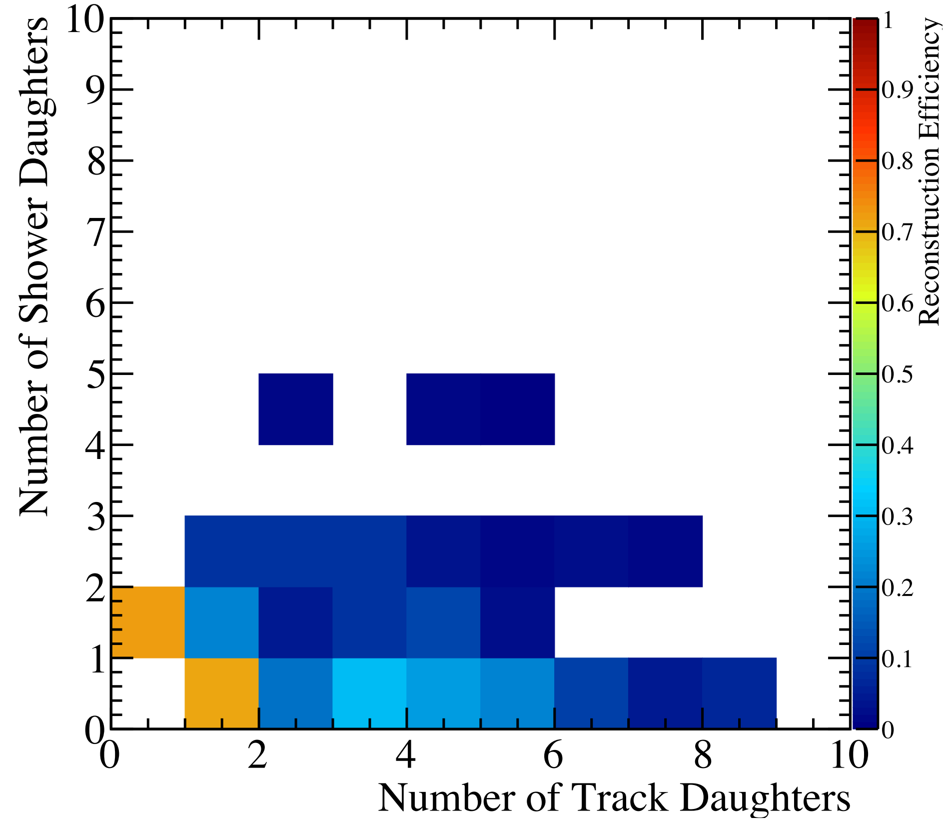
And the 3D slicing, which effectively mirrors the structure shown here.

2D Semantic Segmentation Reconstruction

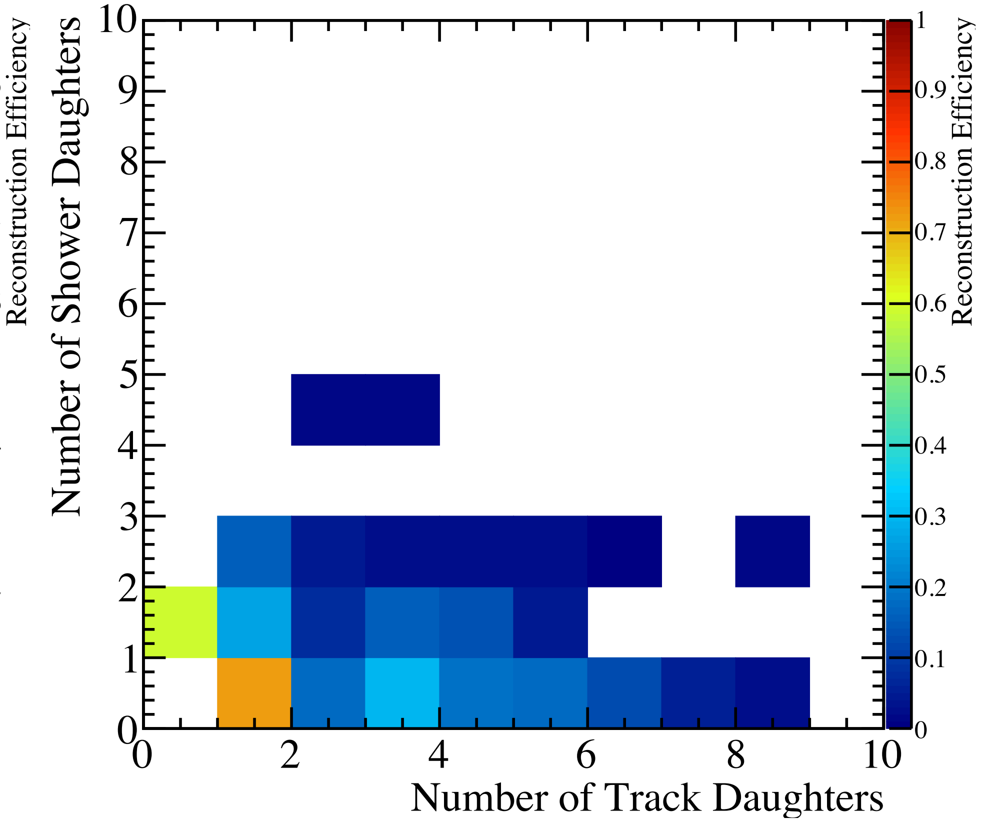


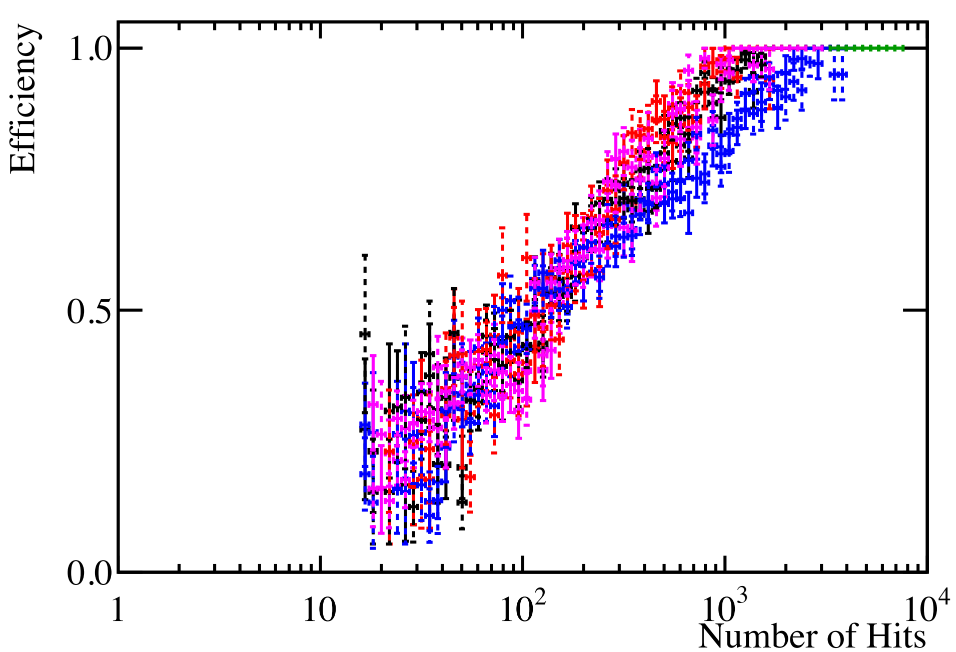
- Using this new model we can examine the hierarchy reconstruction metrics:
- The performance is similar, but there is a drop off in the reconstruction efficiency for single showers i.e. e^+ events.

Standard Pandora



Semantic Segmentation In Pandora

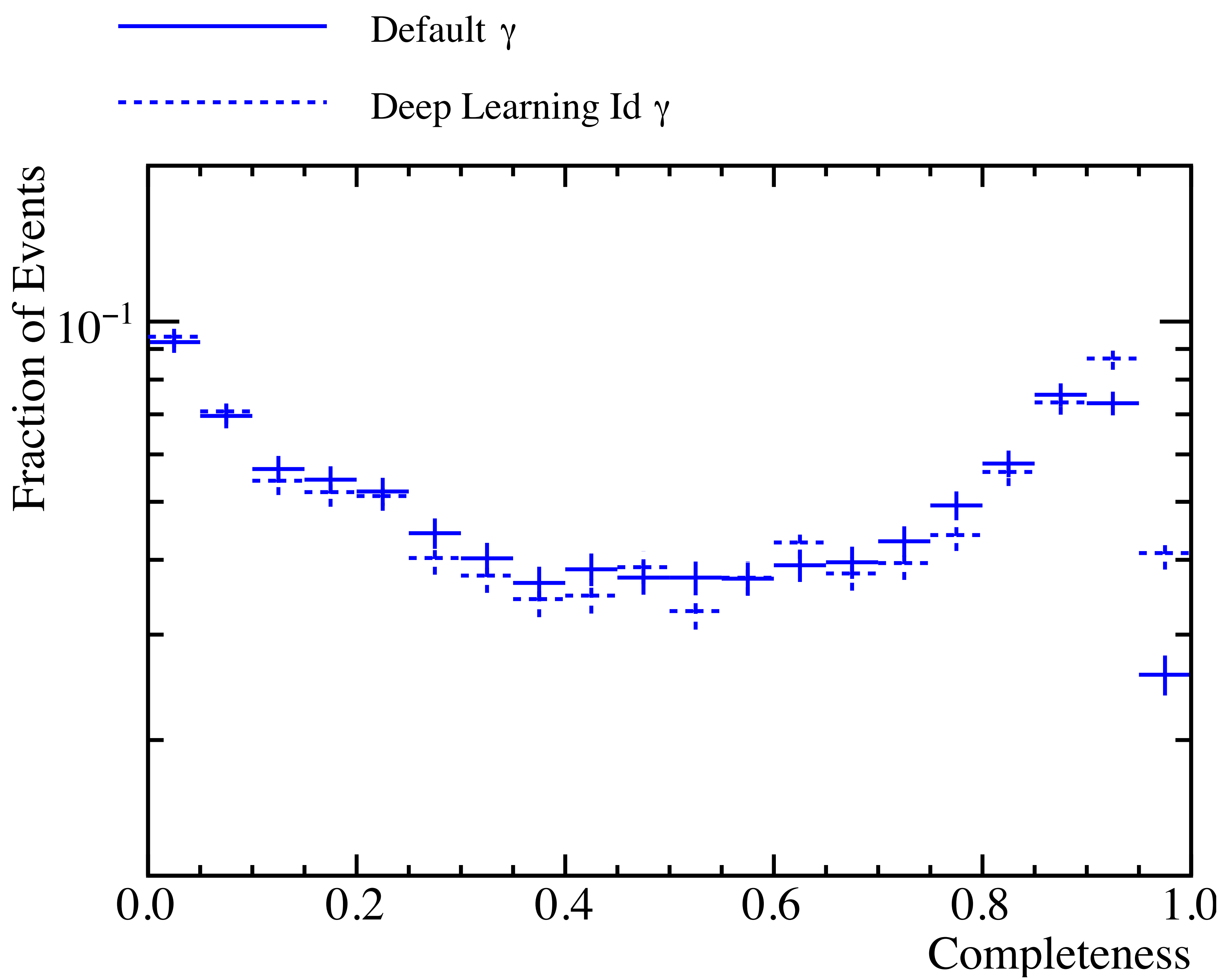




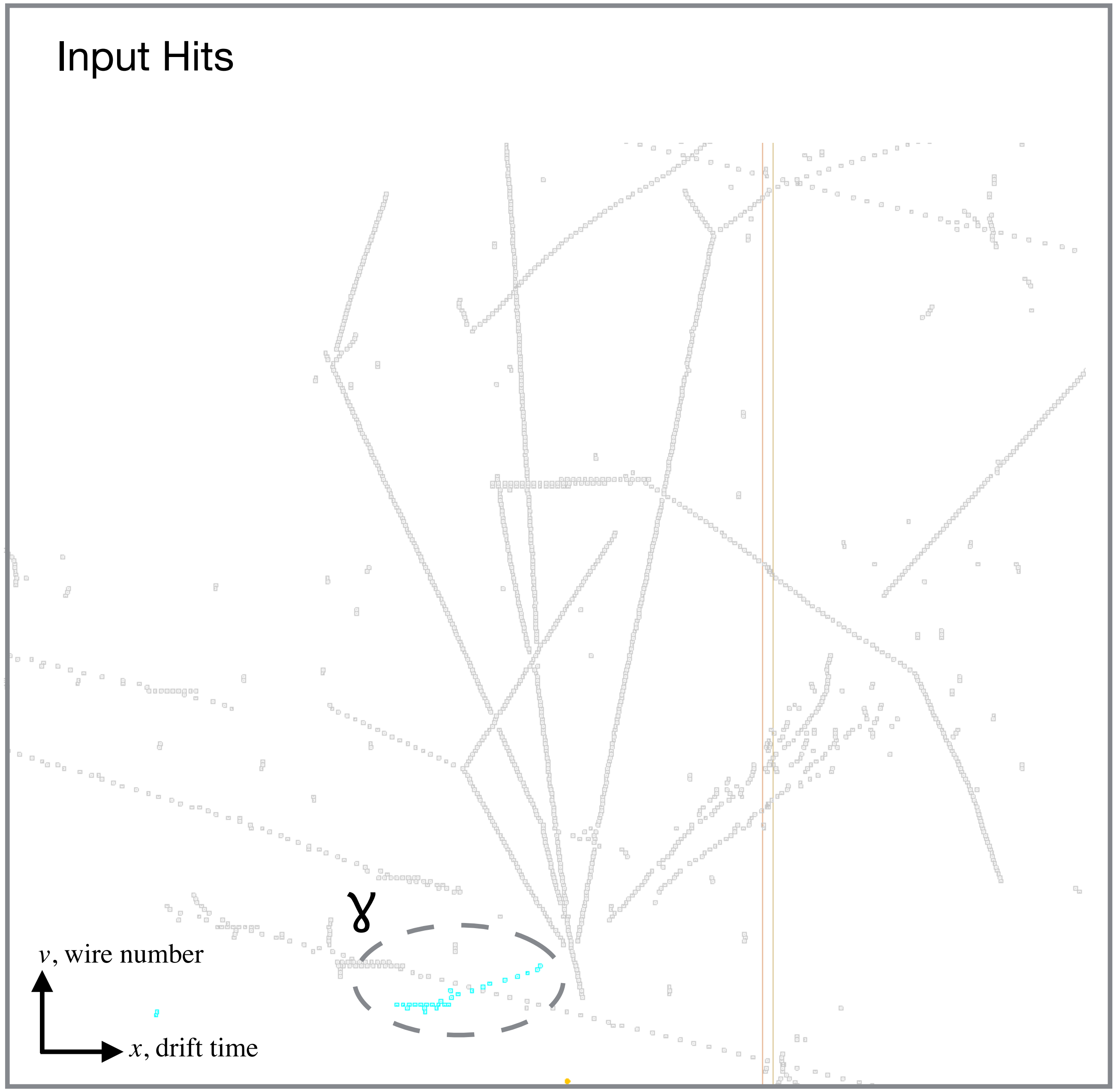
Standard Pandora - Solid Line	*Semantic Segmentation - Dashed Line*
π^+, nEvts(Matched) : 4747(3208)	π^+, nEvts(Matched) : 4747(3260)
π^-, nEvts(Matched) : 2024(1362)	π^-, nEvts(Matched) : 2024(1373)
γ, nEvts(Matched) : 5475(3519)	γ, nEvts(Matched) : 5475(3521)
p, nEvts(Matched) : 3401(1872)	p, nEvts(Matched) : 3401(1937)
e^+, nEvts(Matched) : 717(704)	e^+, nEvts(Matched) : 719(709)

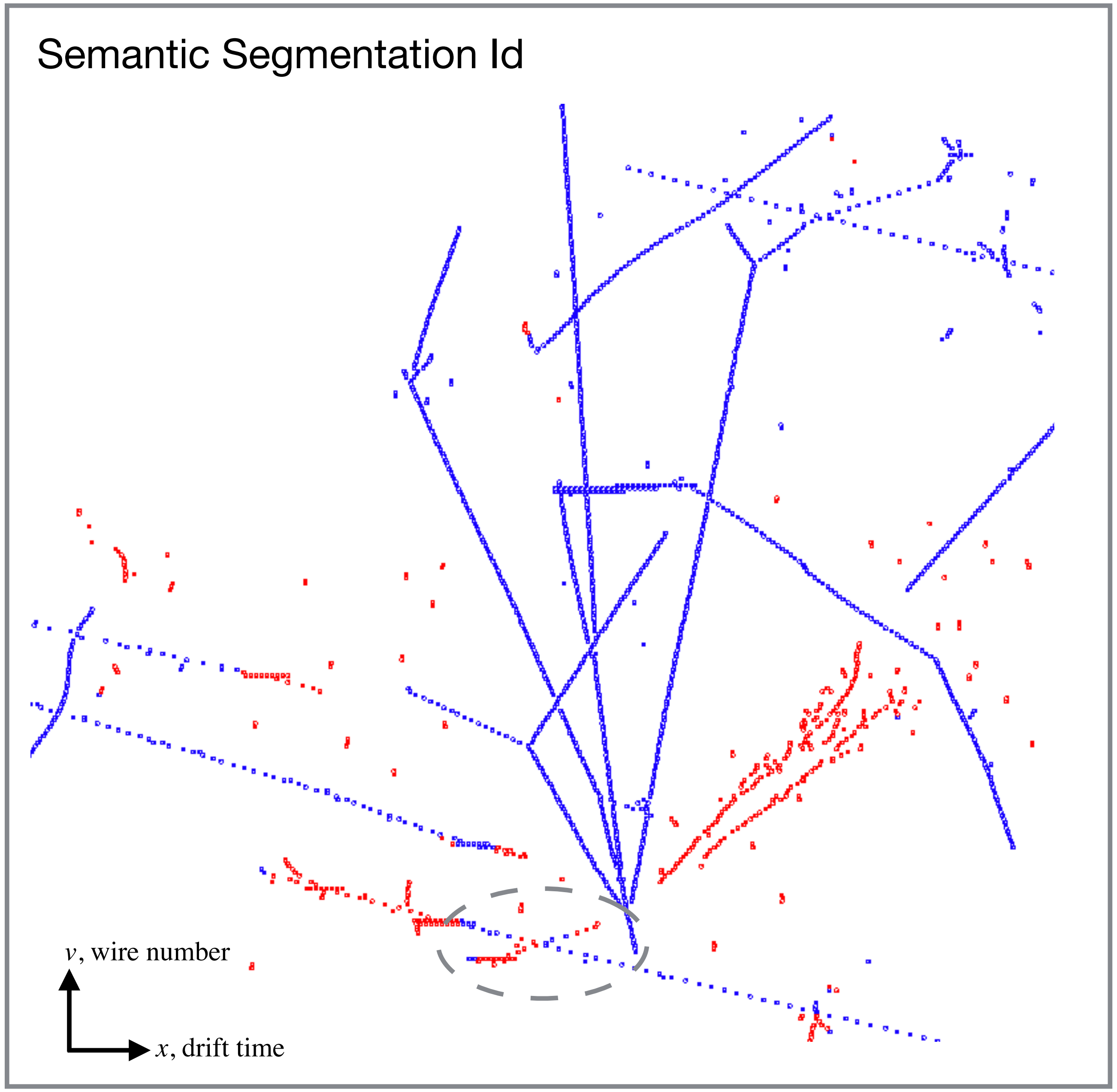
Semantic Segmentation Performance

- But there is an improvement in the photon completeness.
- So there is promise in this approach, but additional work is needed to produce significant benefits to analysers.
- The use of deep learning techniques out of the box will not dramatically improve the current Pandora reconstruction.

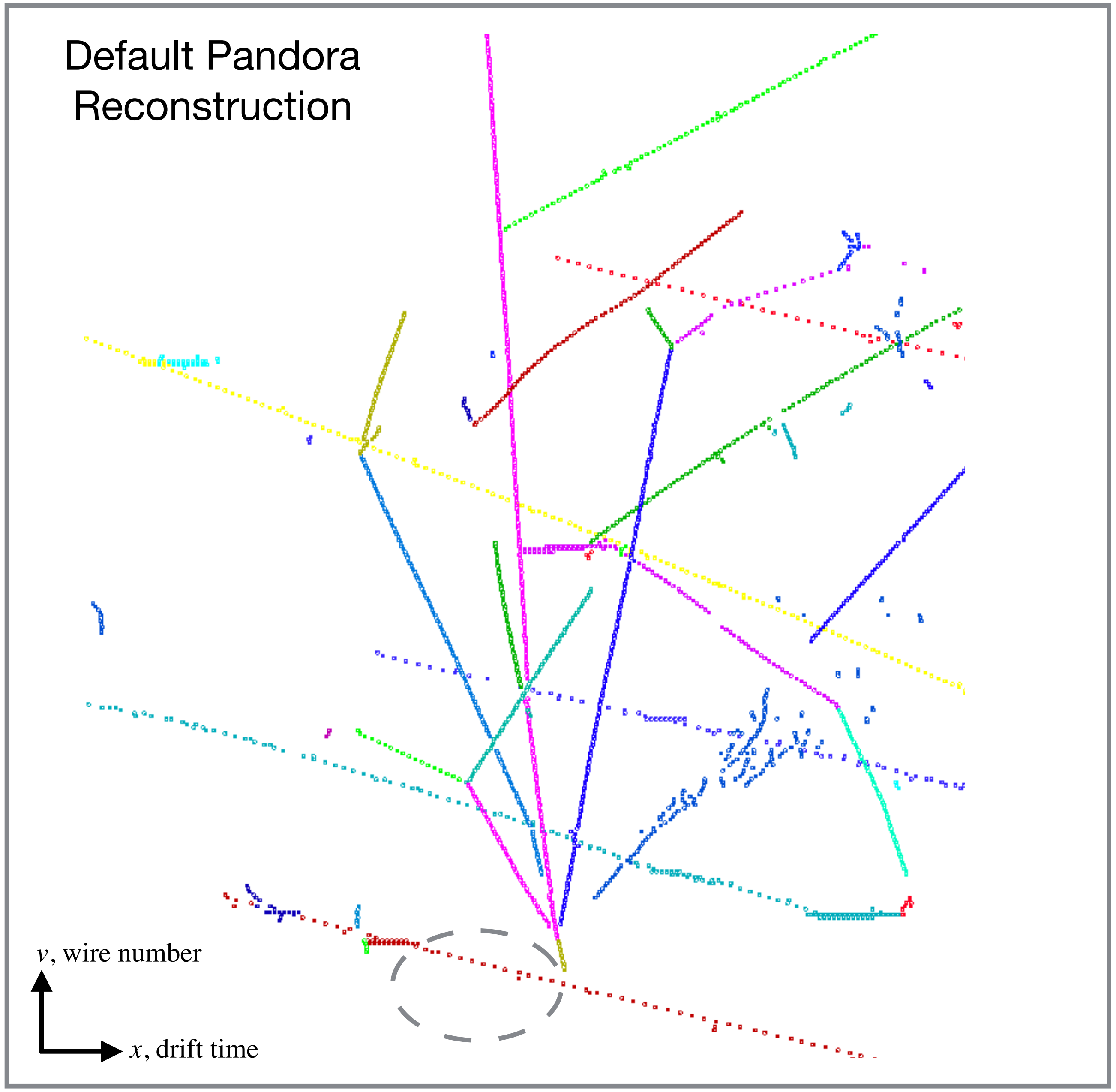


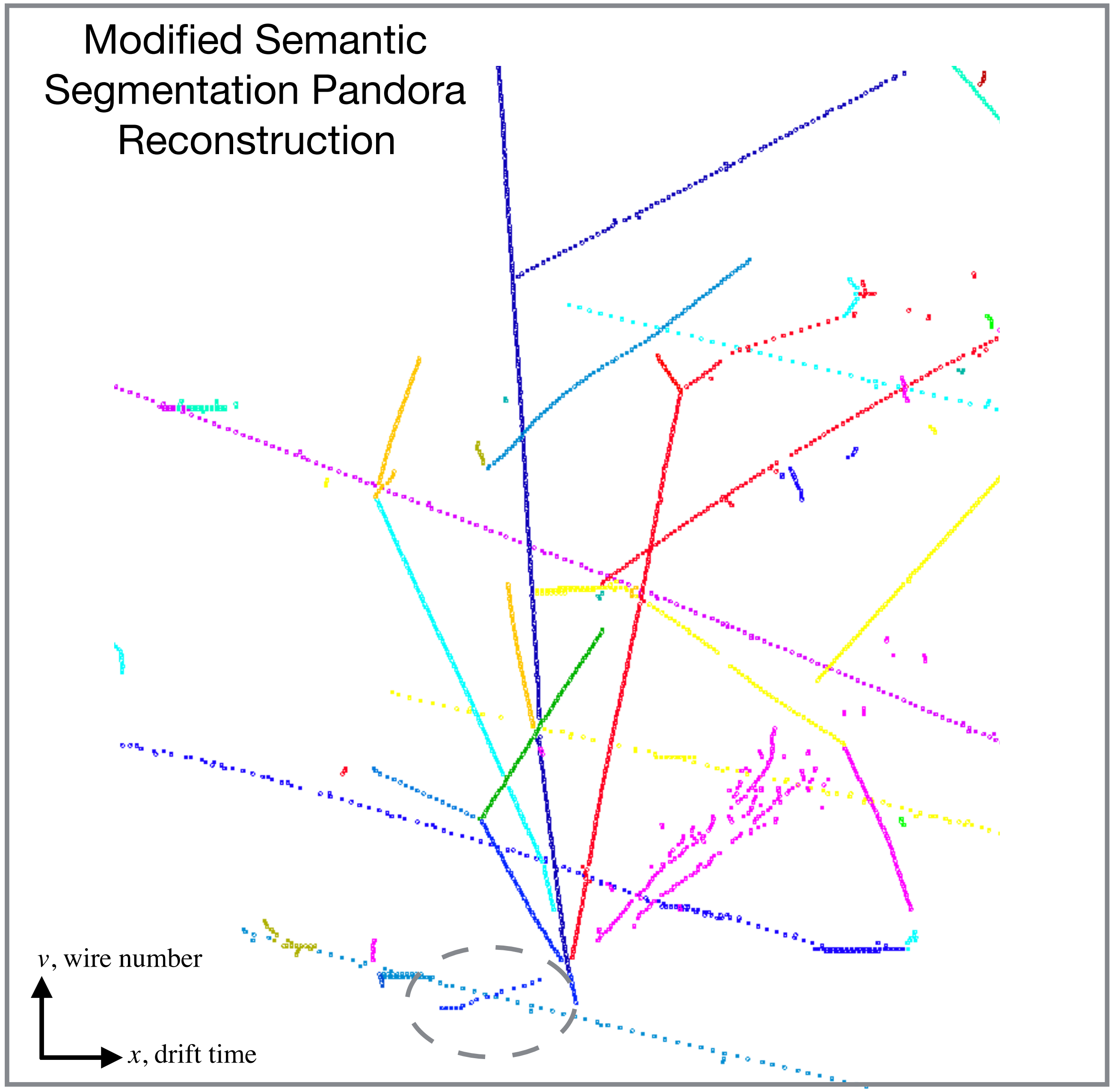
- Example: Complex 7 GeV π^+ interaction where we have highlights a specific primary ɣ daughter.
- The semantic segmentation identifies the majority of the ɣ hits as being shower like.





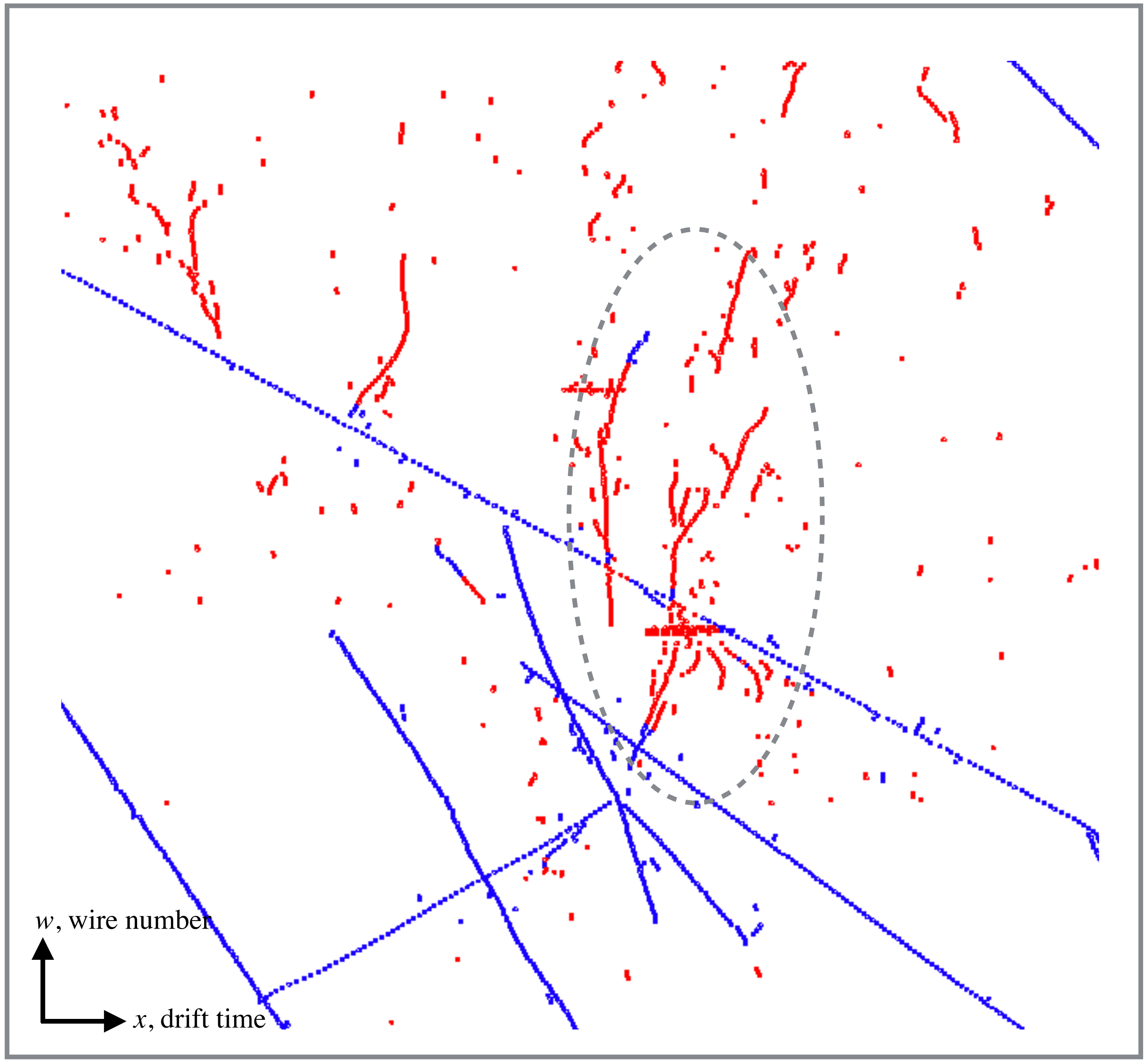
- By reconstructing the shower and track hits separately, this ɣ is reconstructed while in the original reconstruction it is lost.

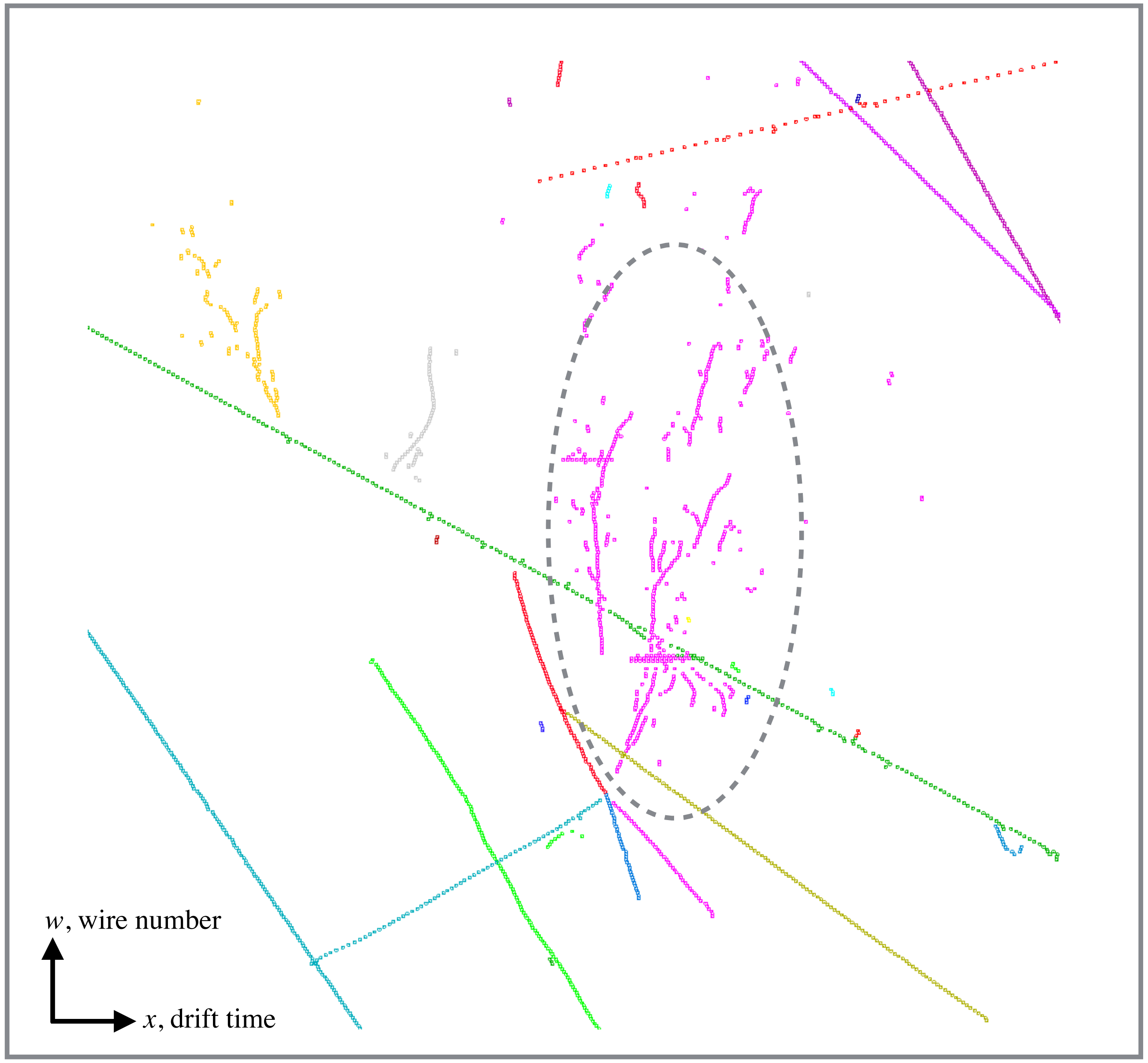




Semantic Segmentation Examples

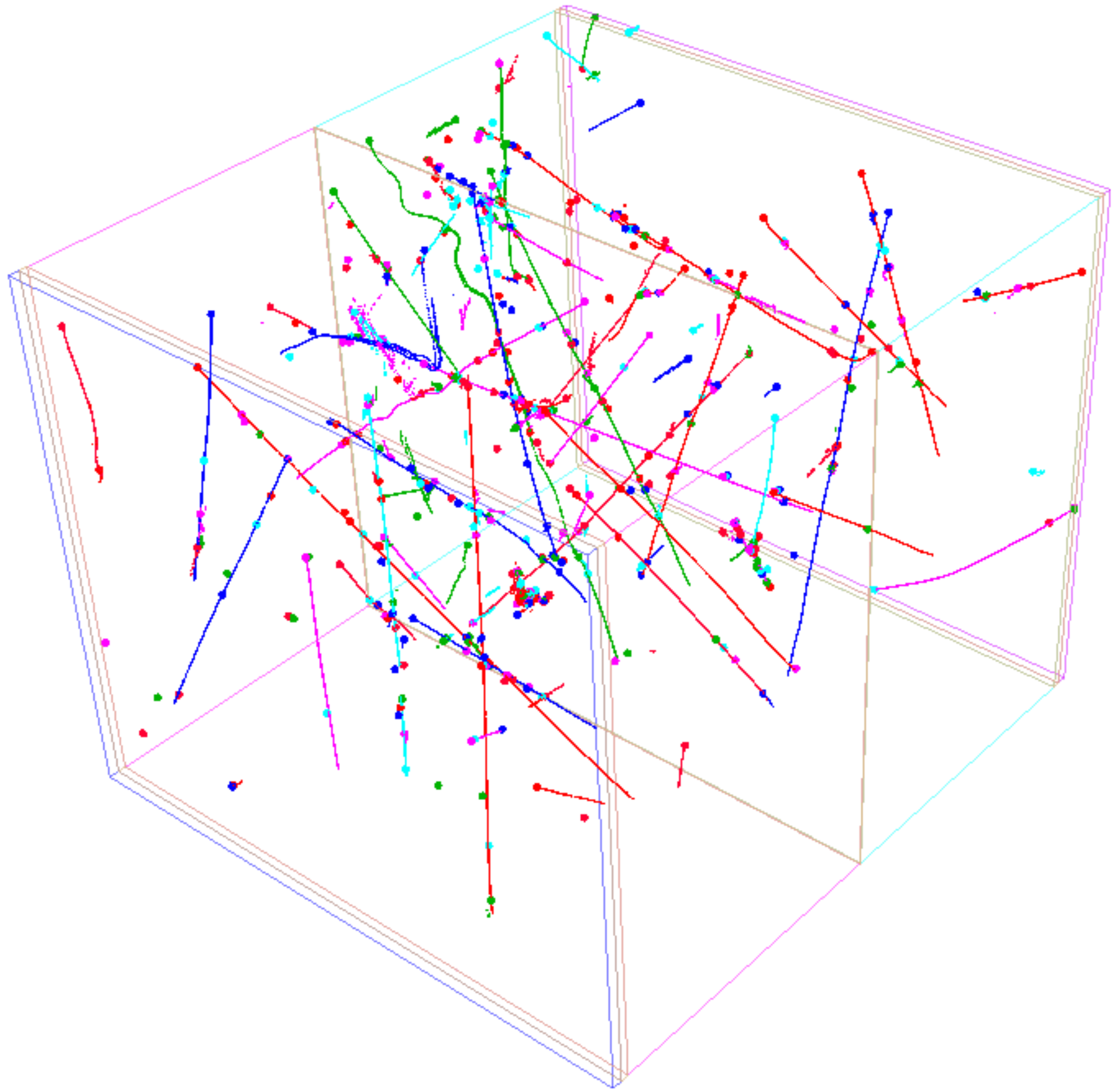
- More work is needed to refine this approach, because merging based on proximity alone is not good enough.
- For example in this complex K^+ interaction two clear showers have been merged together:





Conclusions

- Detailed performance metrics describing the current reconstruction hierarchy have been presented.
- A hit based track shower ID based on semantic segmentation has been implemented and incorporated into the Pandora ProtoDUNE reconstruction.
- While additional work is required to fully exploit the benefits of this new approach the initial results show promise.



Thank you for your attention!

Questions?

Pandora is an open project and new contributors would be extremely welcome. We'd love to hear from you and we will always try to answer your questions.

Pandora SDK Development	John Marshall (John.Marshall@warwick.ac.uk) Mark Thomson (thomson@hep.phy.cam.ac.uk)
LAr TPC algorithm development	John Marshall (John.Marshall@warwick.ac.uk) Andy Blake (a.blake@lancaster.ac.uk)
DUNE FD Integration	Lorena Escudero (escudero@hep.phy.cam.ac.uk)
ProtoDUNE Integration	Steven Green (sg568@hep.phy.cam.ac.uk)
MicroBooNE Integration	Andy Smith (asmith@hep.phy.cam.ac.uk)
Graduate Students	MicroBooNE : Jack Anthony ProtoDUNE : Stefano Vergani DUNE : Jhanzeb Ahmed, Mousam Rai, Ryan Cross

https://github.com/PandoraPFA

https://pandorapfa.slack.com

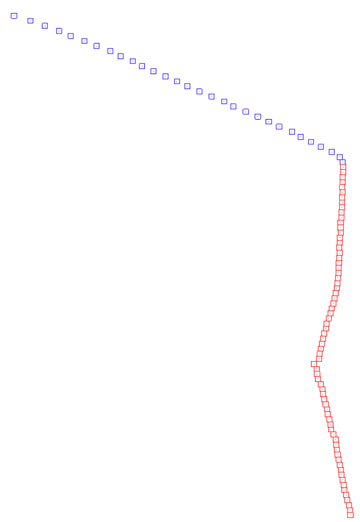
UNIVERSITY OF CAMBRIDGE

Lancaster University

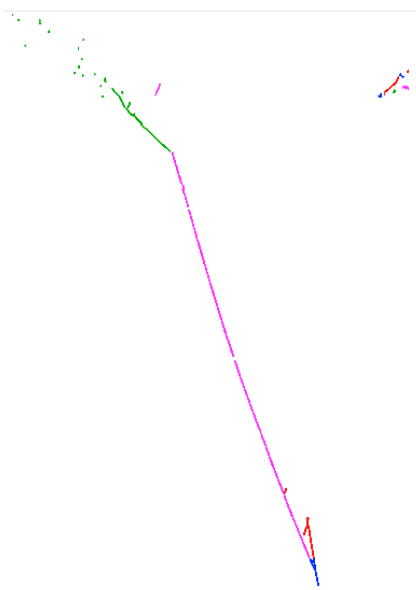
WARWICK
THE UNIVERSITY OF WARWICK

Back Up

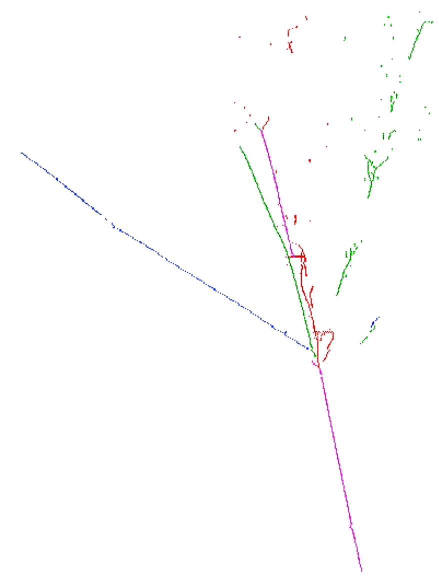
Hierarchy Complexity



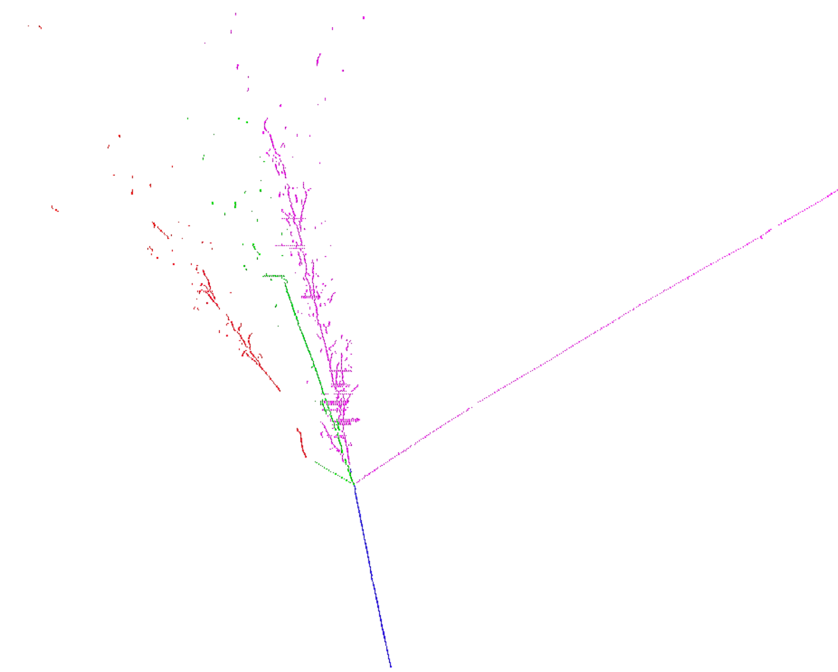
1 GeV



3 GeV



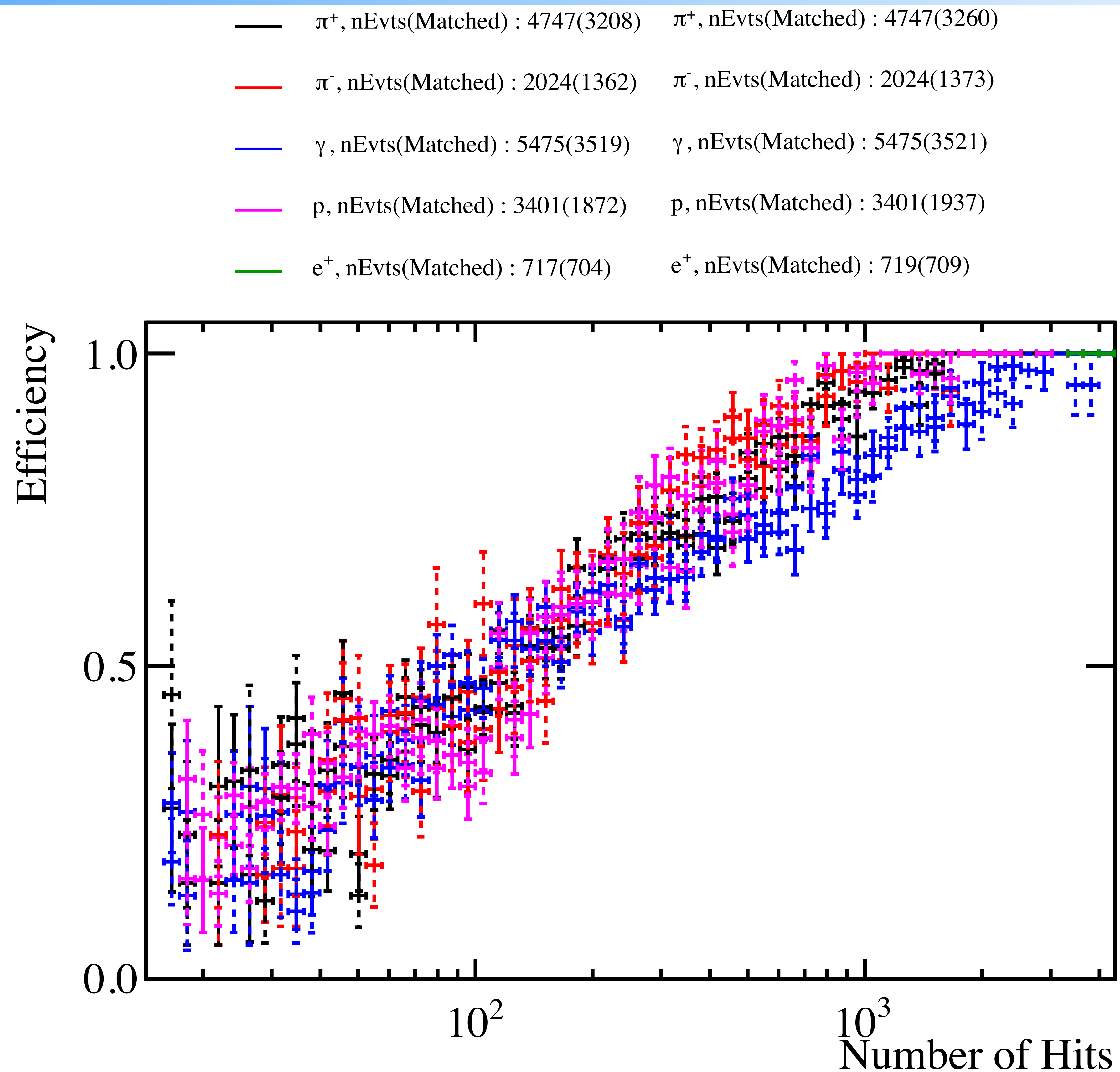
5 GeV



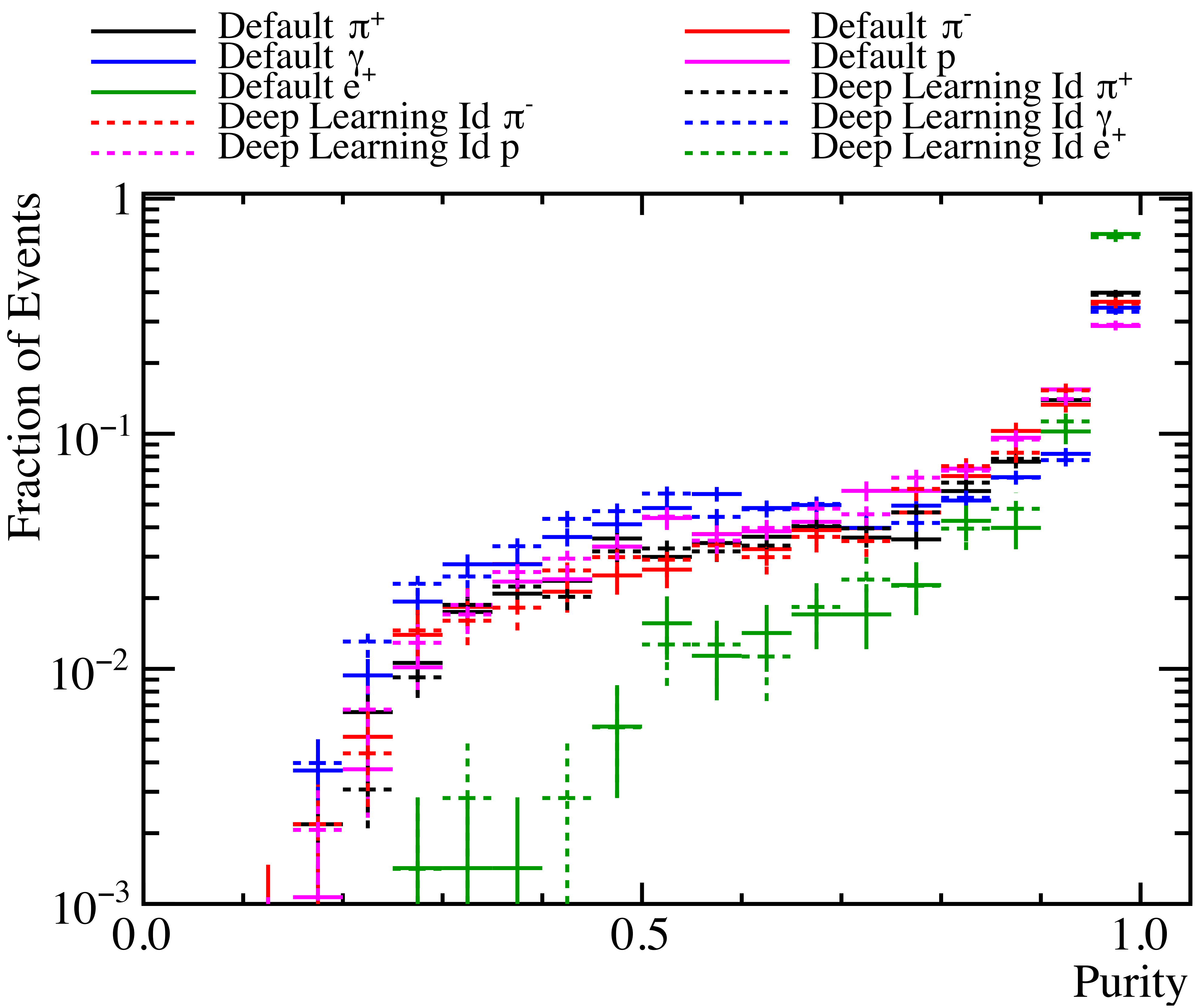
7 GeV

Standard Pandora - Solid Line

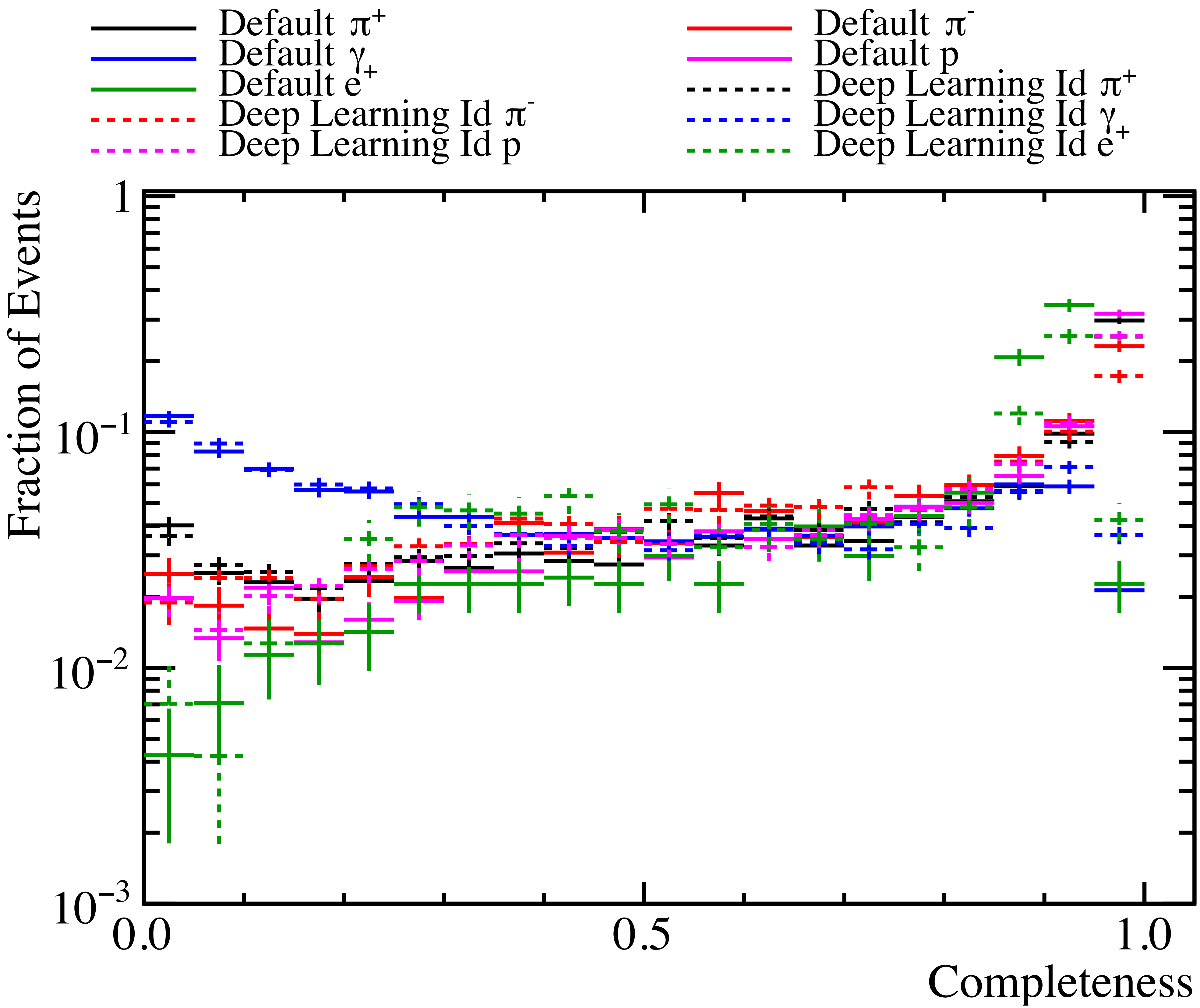
Semantic Segmentation - Dashed Line



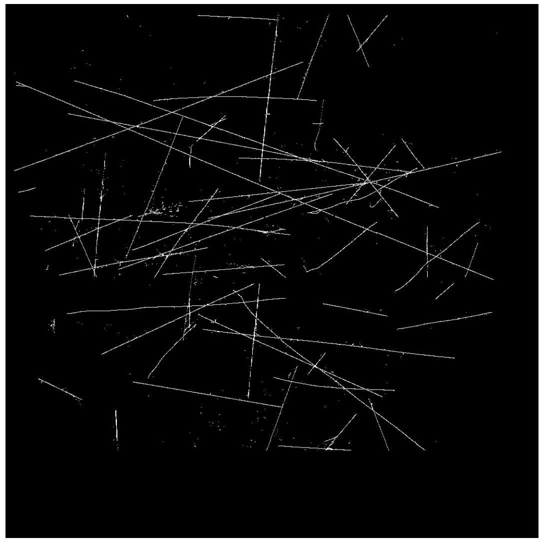
DUNE

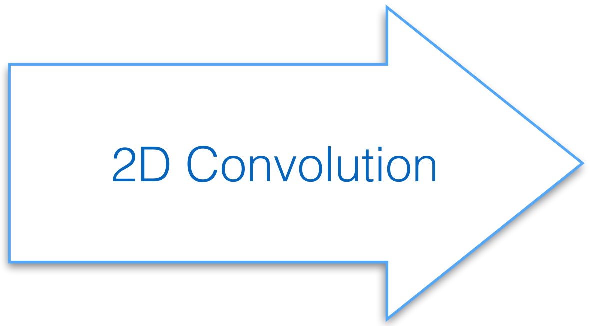


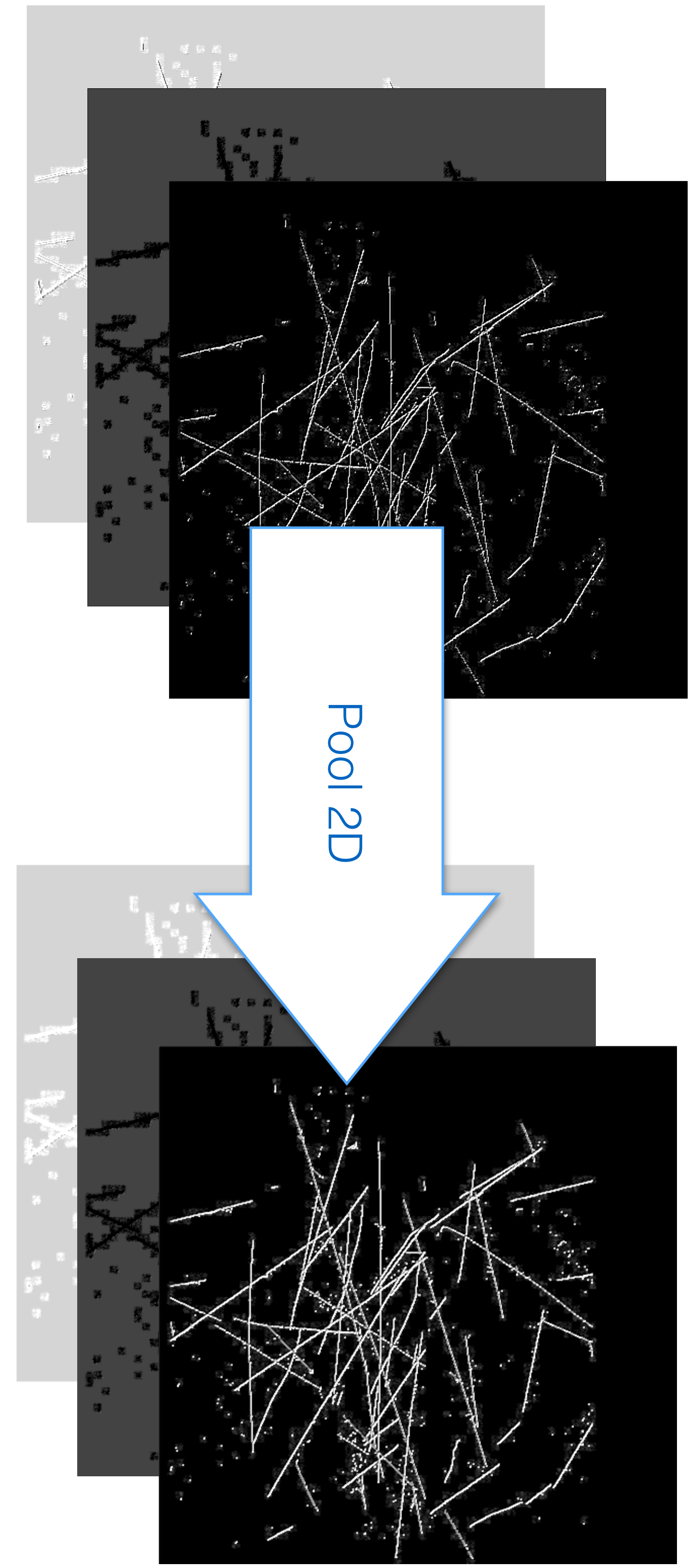
Default π^+
Default π^-
Default γ
Default p
Default e^+
Deep Learning Id π^+
Deep Learning Id π^-
Deep Learning Id γ
Deep Learning Id p
Deep Learning Id e^+
Fraction of Events
1
10^{-1}
10^{-2}
10^{-3}
0.0
0.5
1.0
Purity

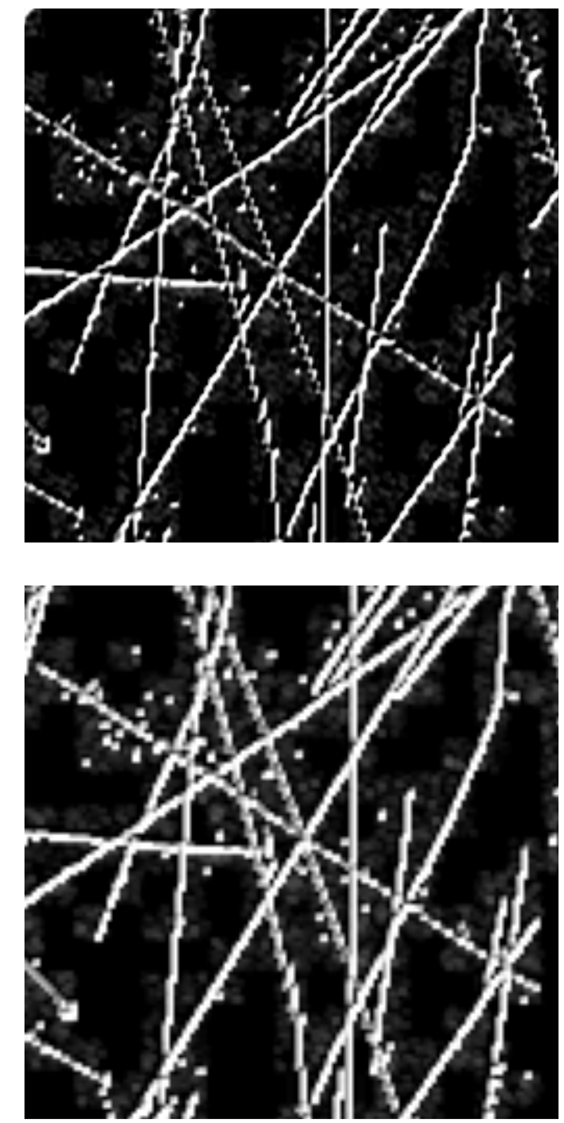

Default π+
Default π-
Default γ
Default p
Default e+
Deep Learning Id π+
Deep Learning Id π-
Deep Learning Id γ
Deep Learning Id p
Deep Learning Id e+
Fraction of Events
1
10−1
10−2
10−3
0.0
0.5
1.0
Completeness

UNet Architecture

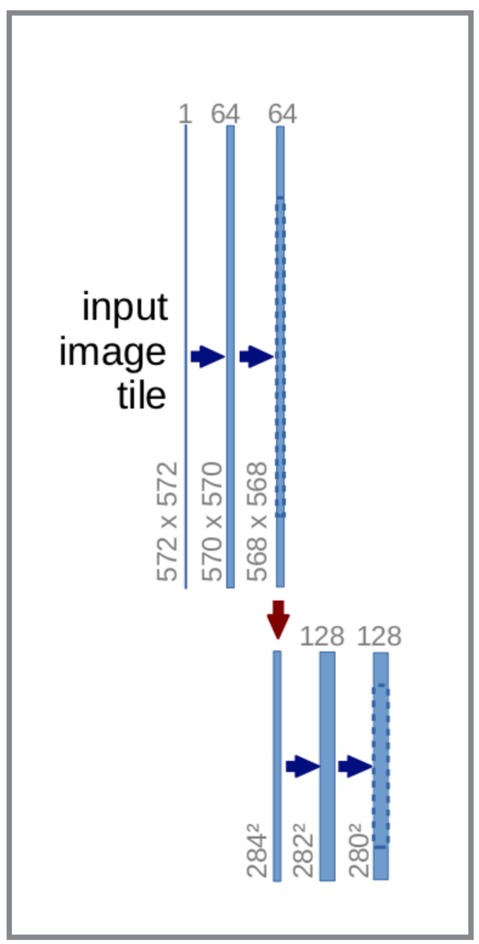




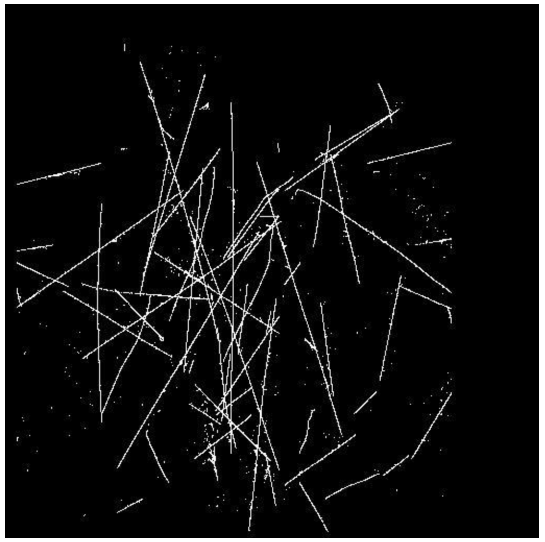




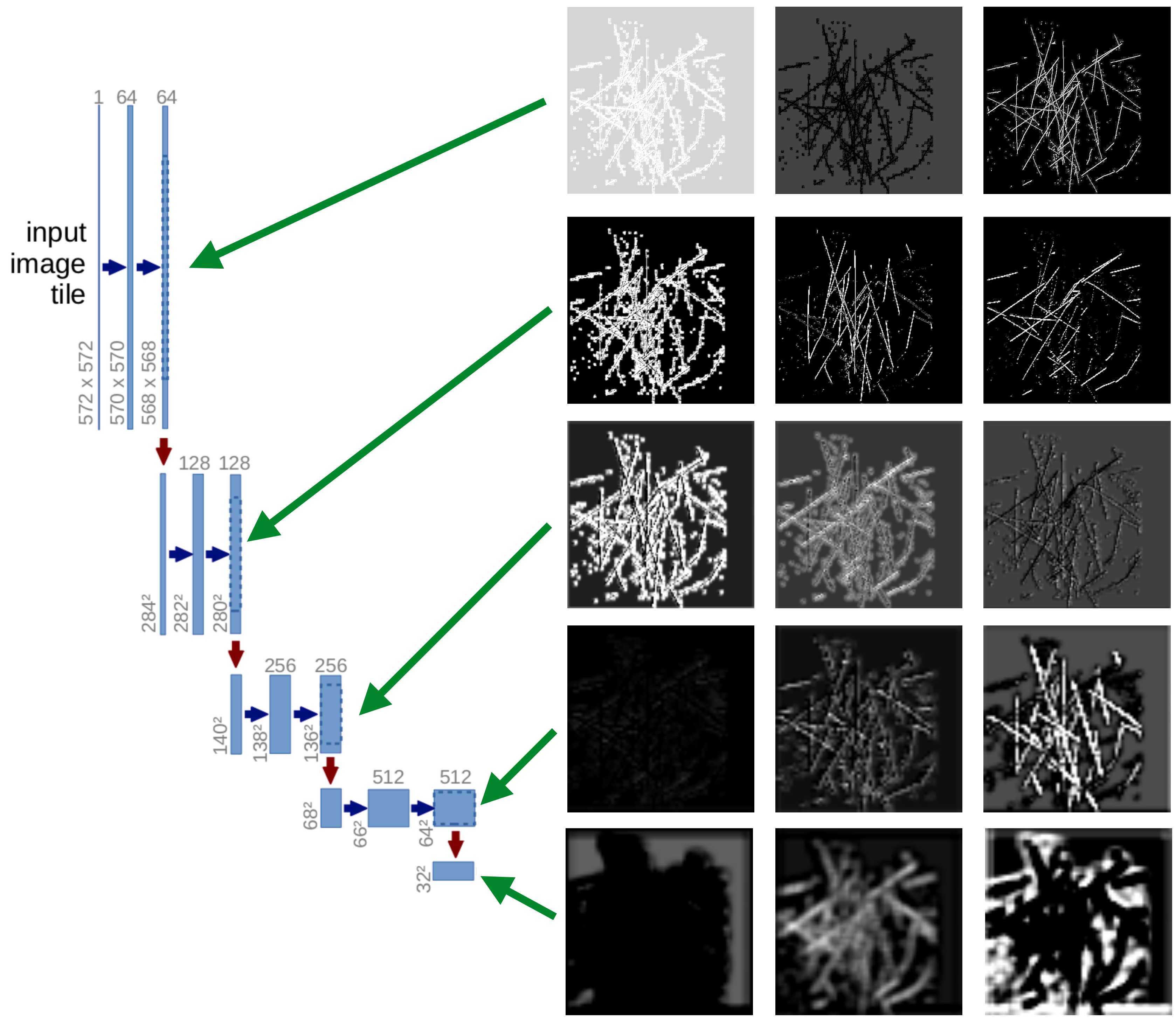
Building Block:



Downsampling:



Input
x3
(1024x1024)



x16
(1024x1024)

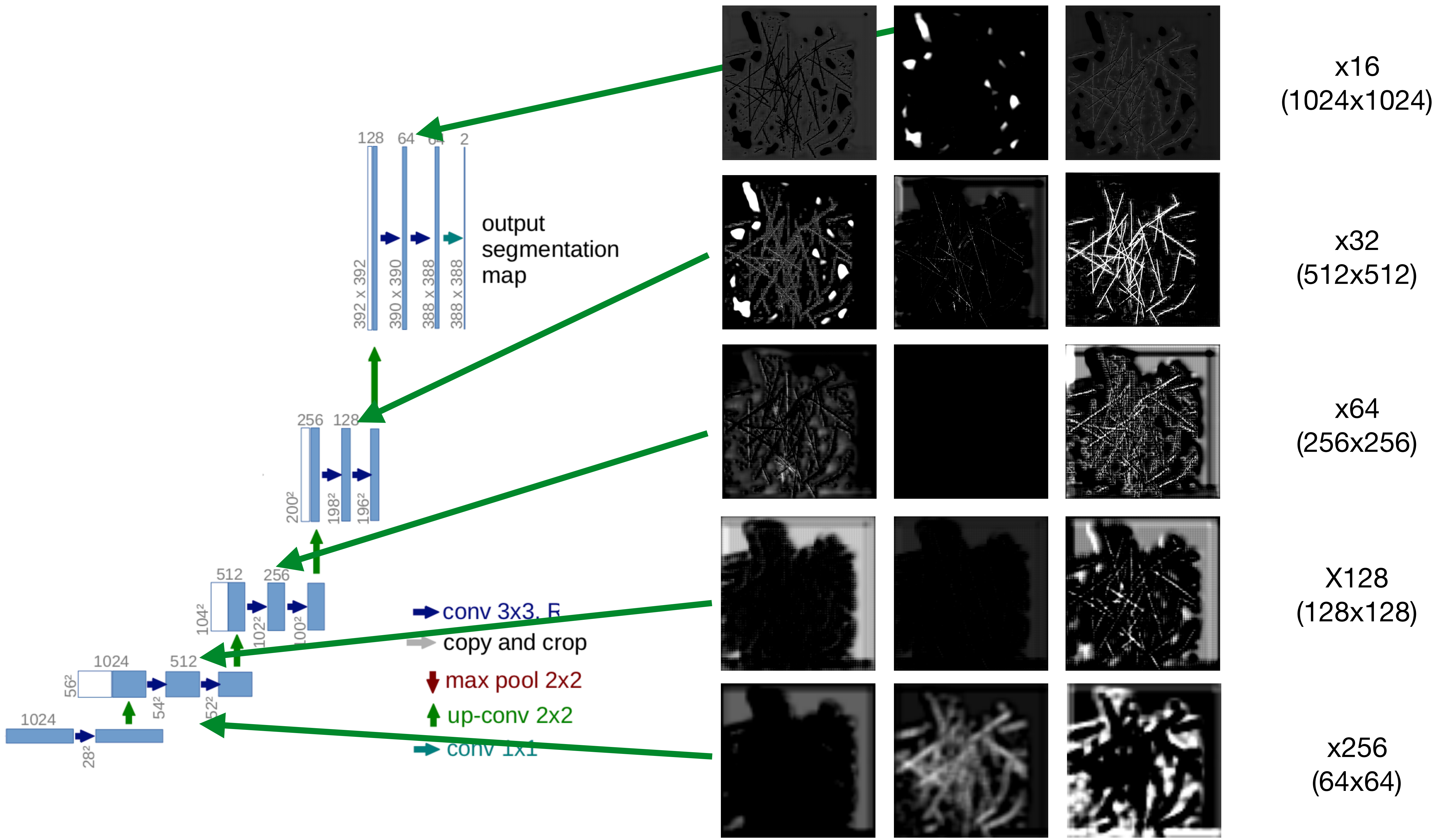
x32
(512x512)

x64
(256x256)

X128
(128x128)

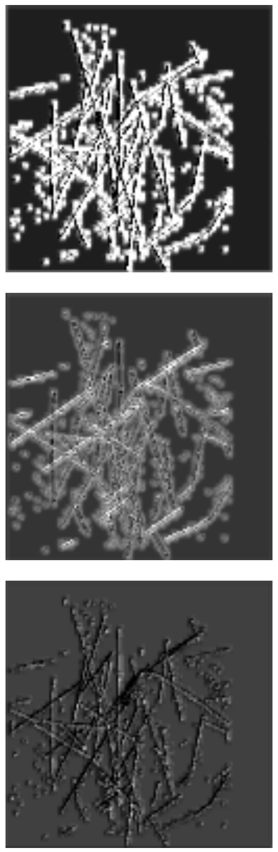
x256
(64x64)

Upsampling:

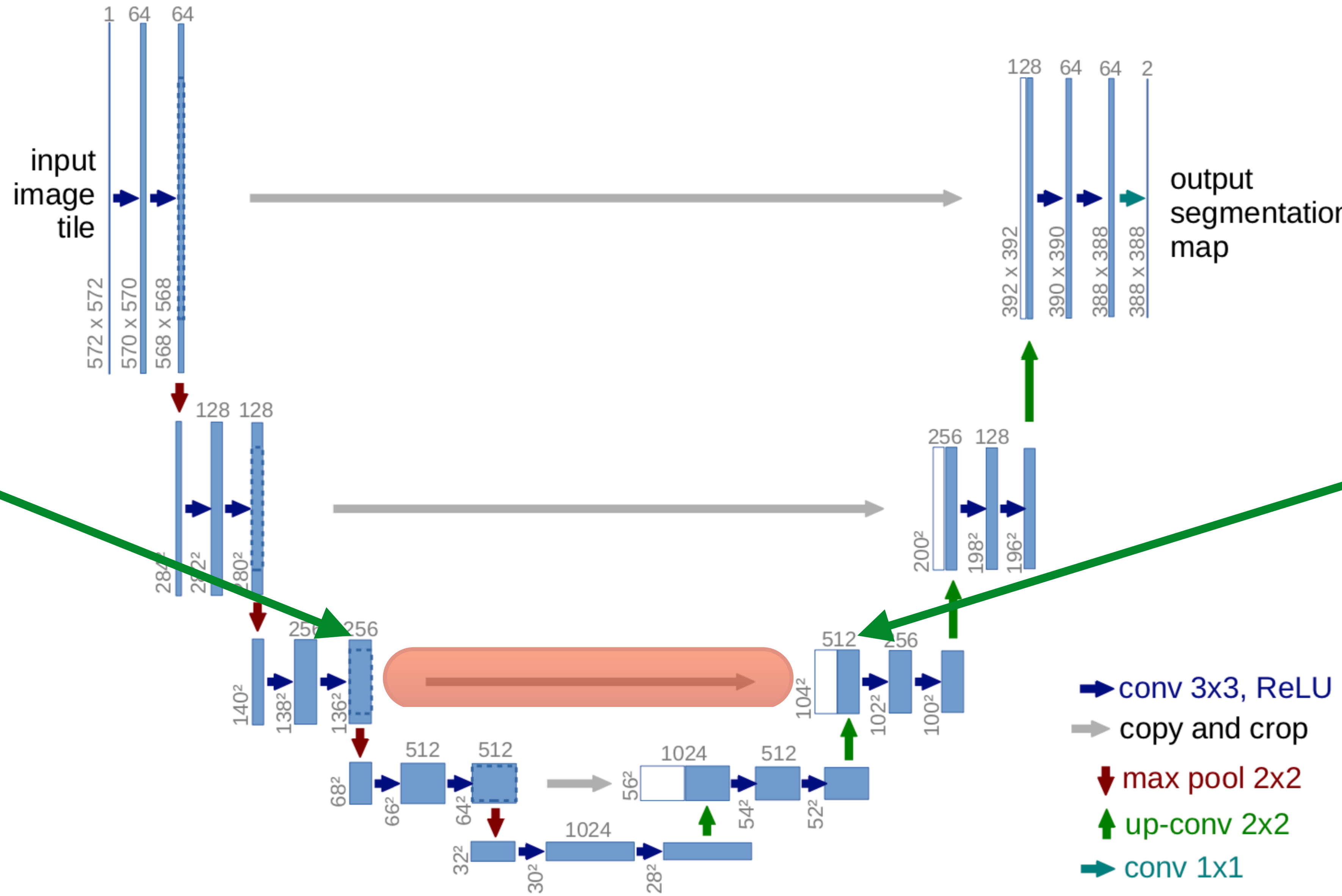


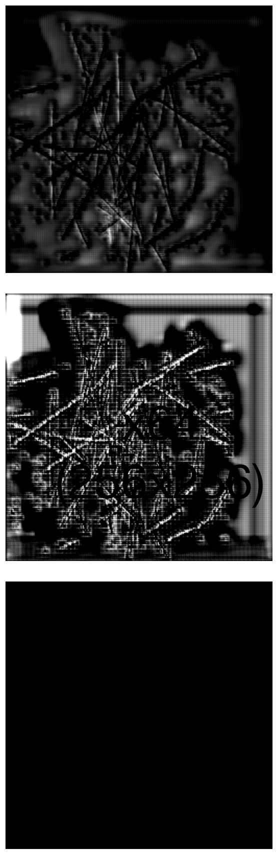
Note: The upsampling uses trained weights, they are not derived from the downsampling convolutions.

Connections: The upsampling combines the relevant images directly from the downsampling part of the architecture.



x64
(256x256)





x64
(256x256)

Conv2d

CLASS `torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros')` [SOURCE]

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{\text{out}}, H_{\text{out}}, W_{\text{out}})$ can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

where $\star$ is the valid 2D cross-correlation operator, N is a batch size, C denotes a number of channels, H is a height of input planes in pixels, and W is width in pixels.

MaxPool2d

CLASS `torch.nn.MaxPool2d(kernel_size, stride=None, padding=0, dilation=1, return_indices=False, ceil_mode=False)` [SOURCE]

Applies a 2D max pooling over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C, H, W), output (N, C, H_{out}, W_{out}) and `kernel_size` (kH, kW) can be precisely described as:

$$
\begin{aligned}
out(N_i, C_j, h, w) = & \max_{m=0,\ldots,kH-1} \max_{n=0,\ldots,kW-1} \\
& \text{input}(N_i, C_j, \text{stride}[0] \times h + m, \text{stride}[1] \times w + n)
\end{aligned}
$$

If `padding` is non-zero, then the input is implicitly zero-padded on both sides for `padding` number of points. `dilation` controls the spacing between the kernel points. It is harder to describe, but this link has a nice visualization of what `dilation` does.