e-/gamma separation in the SFGD

# Intro to a gamma rejection algorithm using a Variational Auto-Encoder (VAE)

Anaëlle Chalumeau & Saúl Alonso-Monsalve

LPNHE neutrino group meeting – 11/09/2024

# Overview

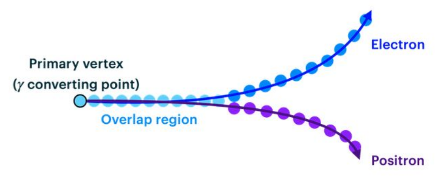
- Motivation: current e-/gamma separation
- What is a VAE
- How to use it for e-/gamma separation
- Simulation data generated for the training
- First look at some training results (very preliminary; just to show it runs)

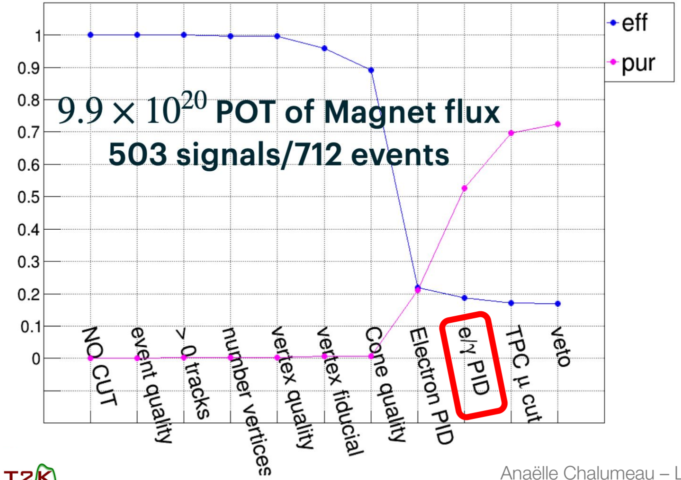
# Motivation: current e-/gamma separation

*from **H. Kobayashi-san** master thesis and last talk at CCNuE/EM meeting*

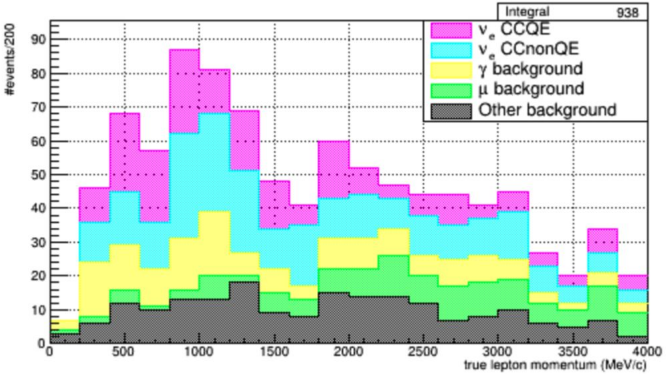
**BDT** using: Energy deposit at first 15 nodes,
Distance and average dE/dx between the primary and second vertex,
Number of connected tracks to the primary track

Idea: dE/dx around vertex of gamma conversion to $e^+e^-$
≃ 2x( dE/dx of a single electron )



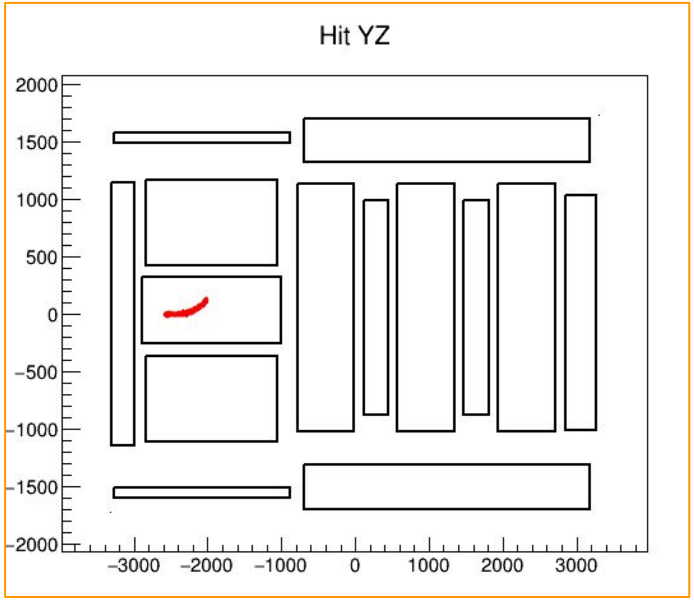




after E/Gamma PID cut:

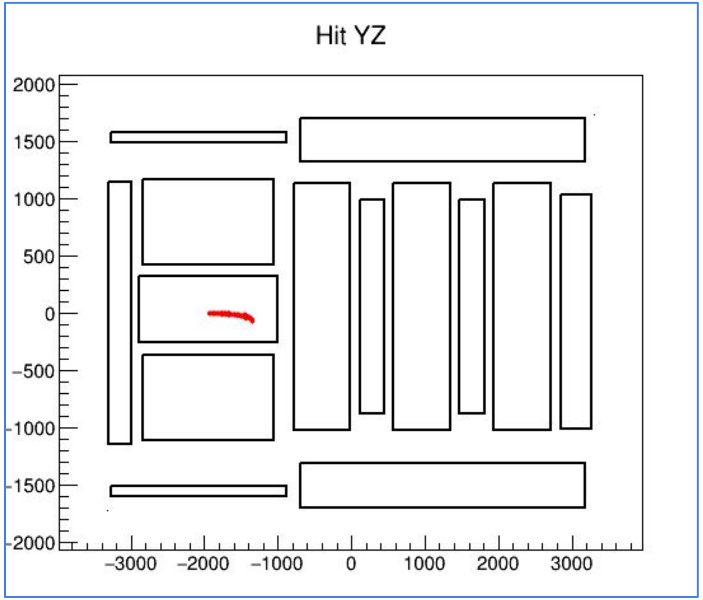


# Motivation: current e-/gamma separation

Also in some cases, only one electron is reconstructed from $\gamma \rightarrow e^+e^-$




*gamma conversion event where only one electron is reconstructed*



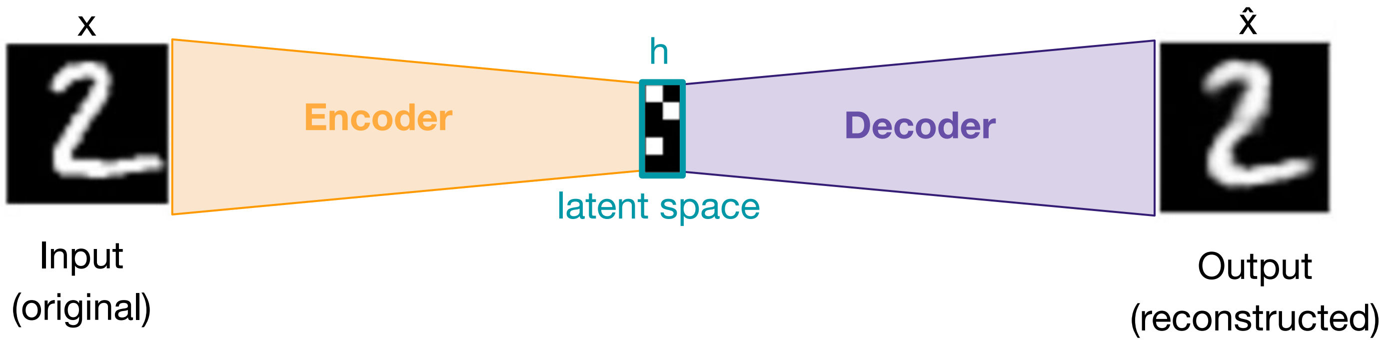

*electron event with no shower*

# What is a VAE

**auto-encoder**

= a type of generative neural network architecture

Idea: encode and decode the incoming information

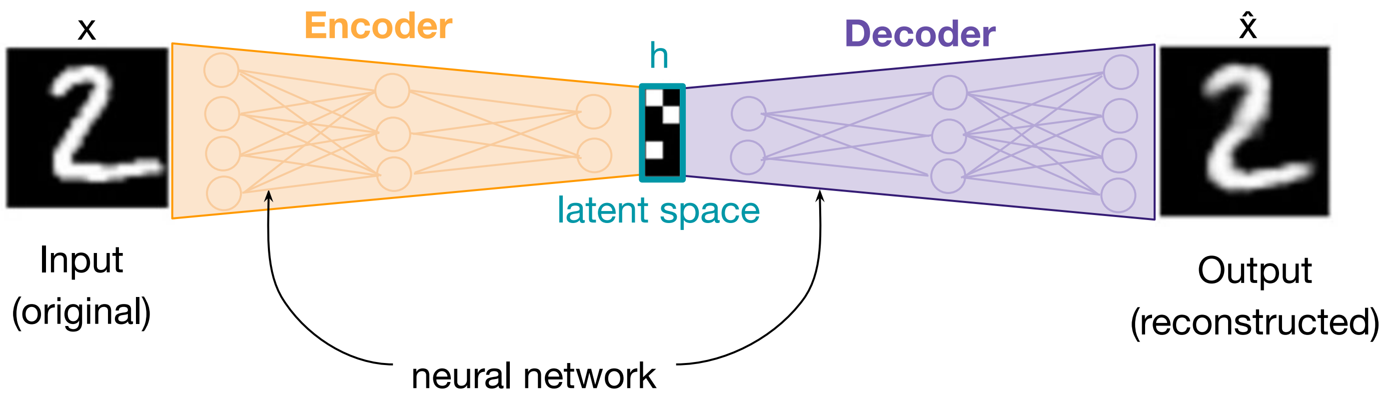


# What is a VAE

**auto-encoder**

= a type of generative neural network architecture

Idea: encode and decode the incoming information

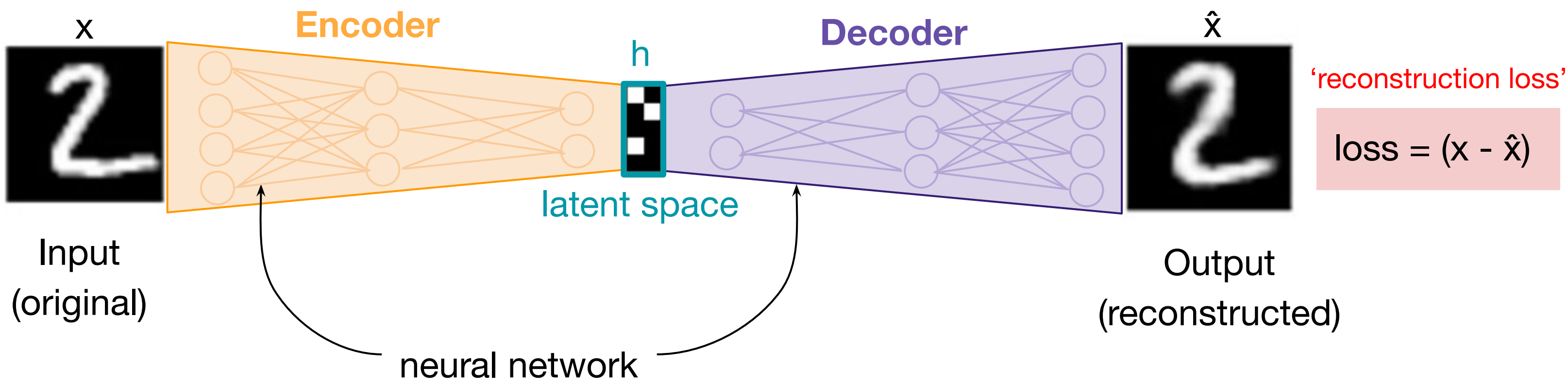


# What is a VAE

**auto-encoder**

= a type of generative neural network architecture

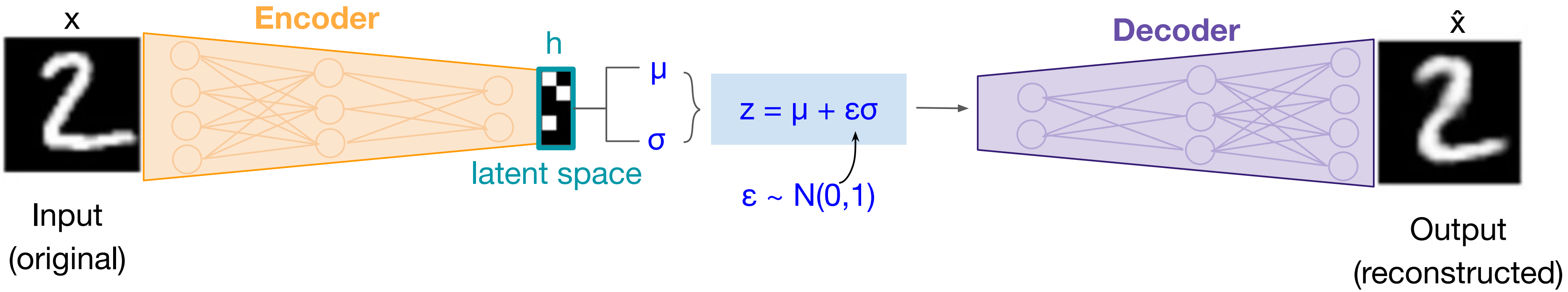
Idea: encode and decode the incoming information



# What is a VAE

**variational auto-encoder**

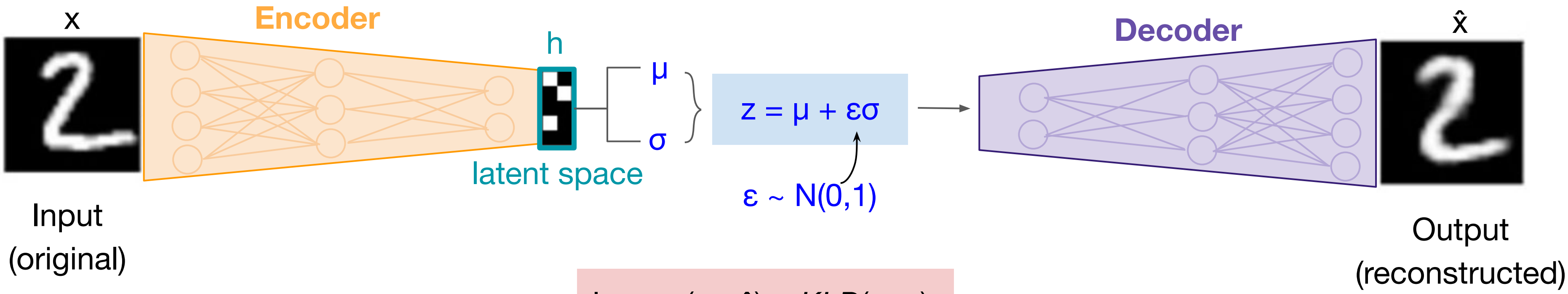
=> learns the distribution of the data instead of a simple compression



# What is a VAE

**variational auto-encoder**

=> learns the distribution of the data instead of a simple compression




$$\text{loss} = \underbrace{(x - \hat{x})}_{\text{'reconstruction loss'}} + \underbrace{KLD(\mu, \sigma)}_{\text{'similarity loss'}}$$

'reconstruction loss' 'similarity loss'

(difference between z and N(0,1))

# How to use it for e-/gamma separation

Principle: **anomaly detection**

1. Train the VAE with many gamma conversion in the SFGD: it will learn to reconstruct the gamma events
2. Test it on e- events: the reconstruction should not be as good → anomaly in the loss function

*More details on the VAE:*

- since we deal with 3-D images, use of a sparse network for efficiency reason (MinkowskiEngine)

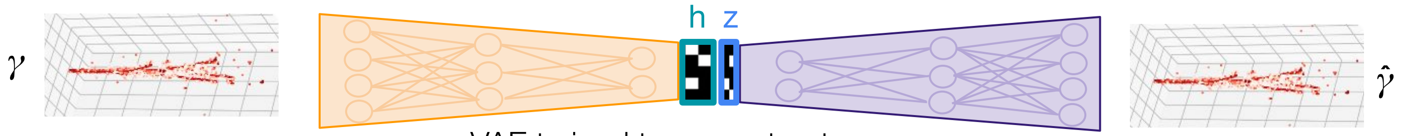
# How to use it for e-/gamma separation

Principle: **anomaly detection**

1 Train the VAE with many gamma conversion in the SFGD: it will learn to reconstruct the gamma events

2 Test it on e- events: the reconstruction should not be as good → anomaly in the loss function

1




VAE trained to reconstruct gammas

*More details on the VAE:*

- since we deal with 3-D images, use of a sparse network for efficiency reason (MinkowskiEngine)

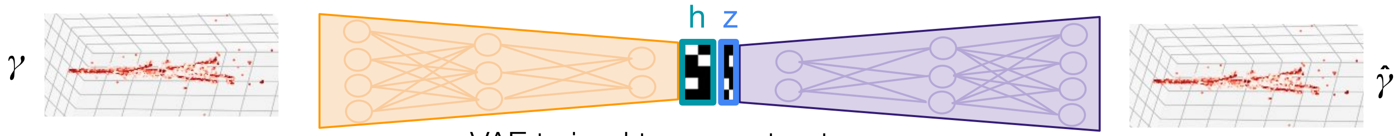
# How to use it for e-/gamma separation

Principle: **anomaly detection**

1 Train the VAE with many gamma conversion in the SFGD: it will learn to reconstruct the gamma events

2 Test it on e- events: the reconstruction should not be as good → anomaly in the loss function

1




VAE trained to reconstruct gammas

loss ~ 0

*More details on the VAE:*

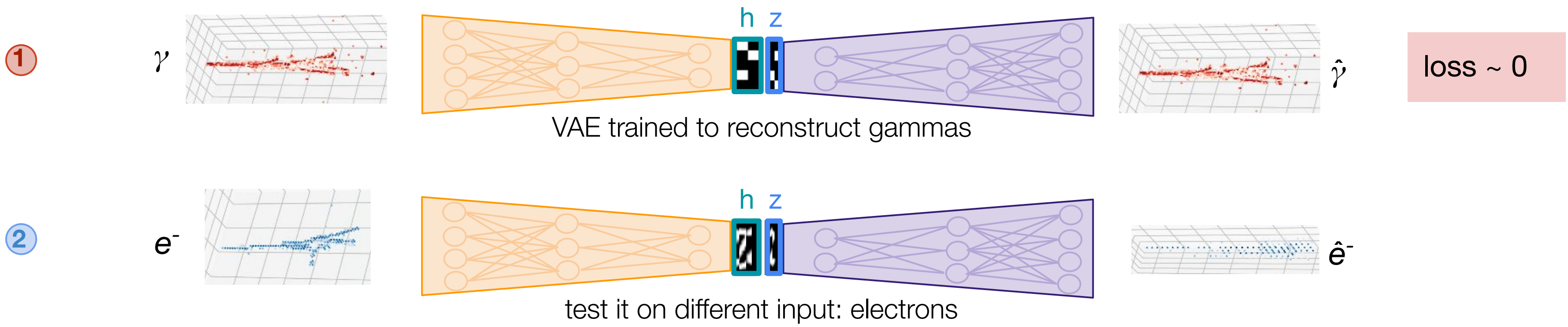
- since we deal with 3-D images, use of a sparse network for efficiency reason (MinkowskiEngine)

# How to use it for e-/gamma separation

Principle: **anomaly detection**

1 Train the VAE with many gamma conversion in the SFGD: it will learn to reconstruct the gamma events

2 Test it on e- events: the reconstruction should not be as good → anomaly in the loss function




*More details on the VAE:*

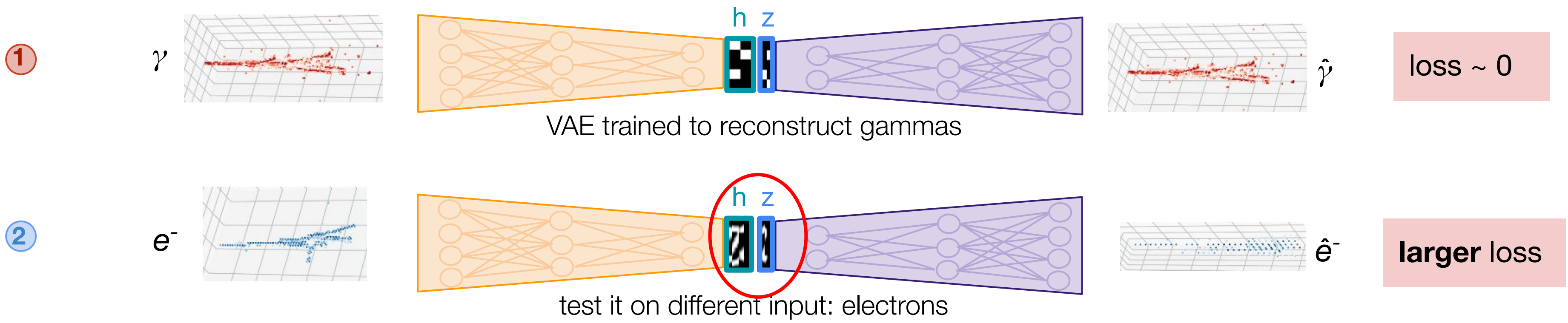
- since we deal with 3-D images, use of a sparse network for efficiency reason (MinkowskiEngine)

# How to use it for e-/gamma separation

Principle: **anomaly detection**

1 Train the VAE with many gamma conversion in the SFGD: it will learn to reconstruct the gamma events

2 Test it on e- events: the reconstruction should not be as good → anomaly in the loss function




*More details on the VAE:*

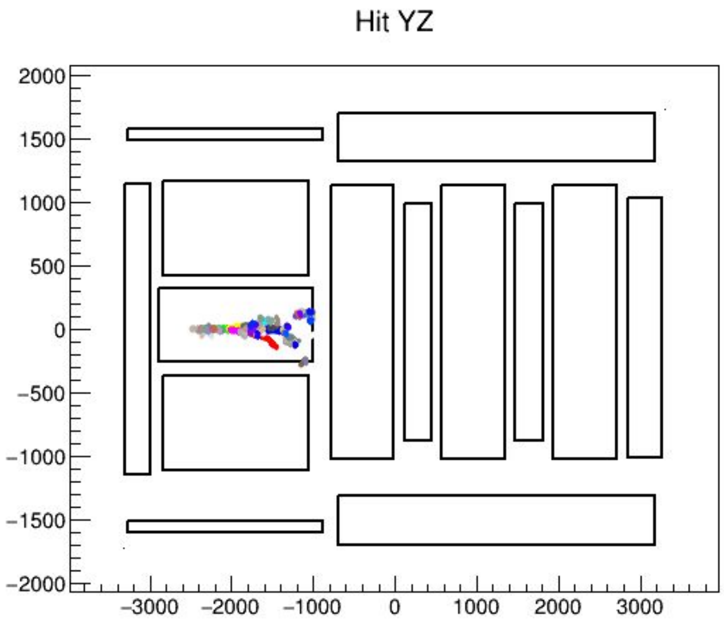
- since we deal with 3-D images, use of a sparse network for efficiency reason (MinkowskiEngine)

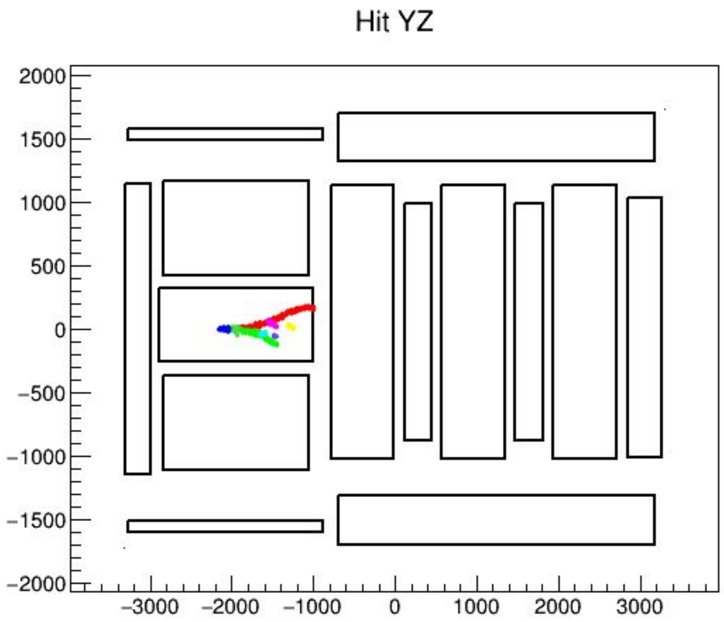
# Simulation data generated for the training

**gamma**

simulation: 283 550 events with initial energy in [100, 2000] MeV and sent along +z

run: g4sim → detResponseSim → eventRecon → eventAnalysis

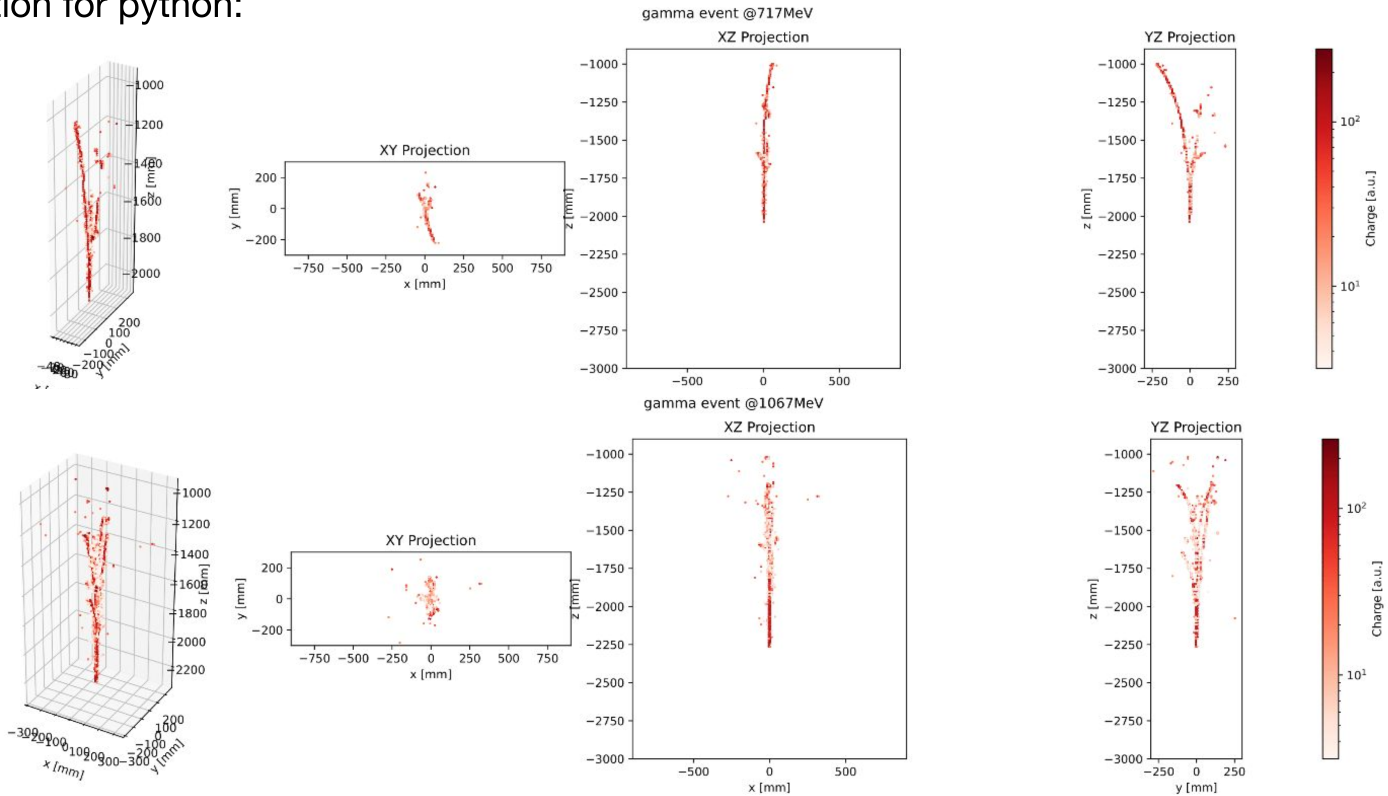


# Simulation data generated for the training

**gamma**

convertion for python:

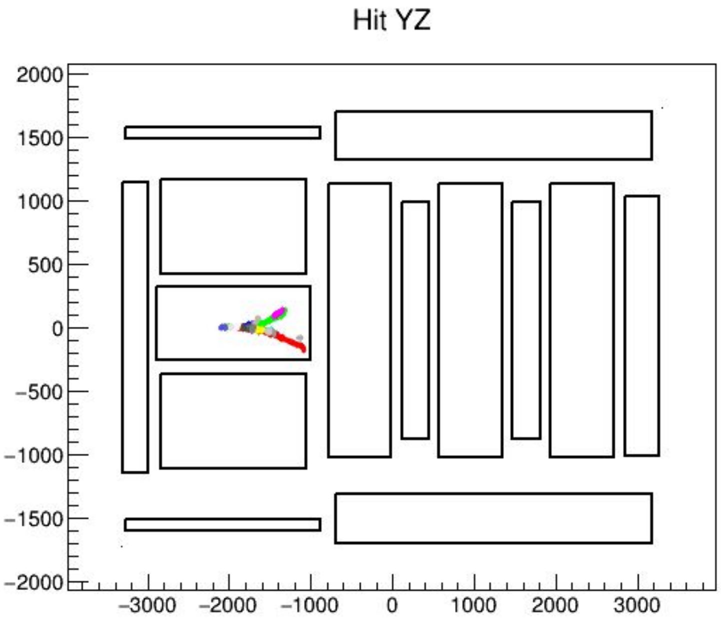


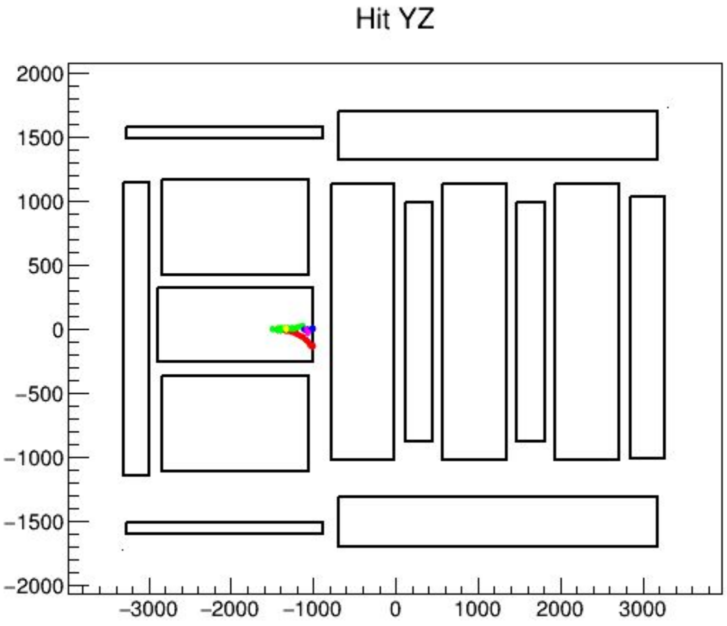
# Simulation data generated for the training

**e-**

(& initial pos ~ gamma conversion proba so that the 2 datasets have same % of tracks with different length)

simulation: 48 368 events with initial energy in [100, 1000] MeV and sent along +z
run: g4sim → detResponseSim → eventRecon → eventAnalysis

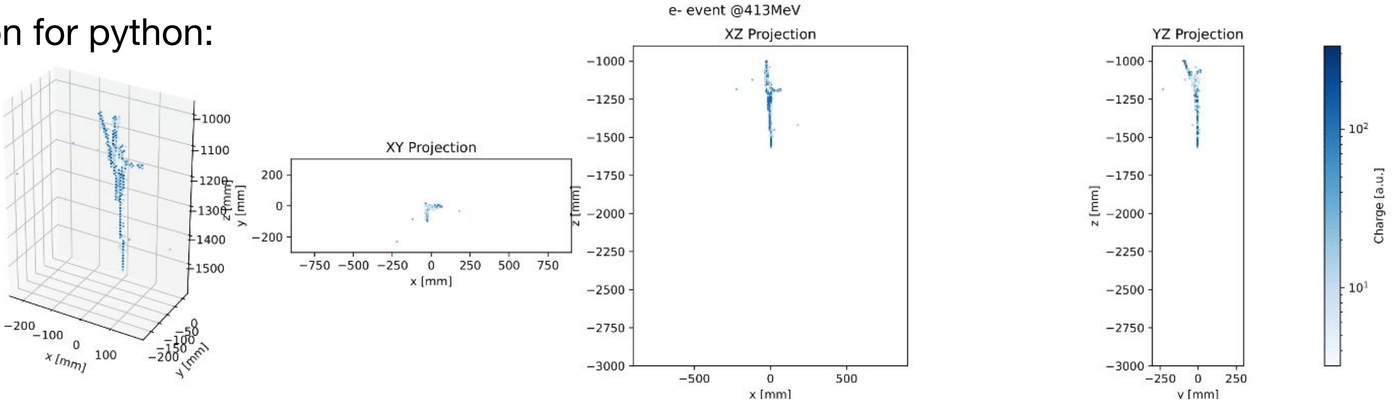


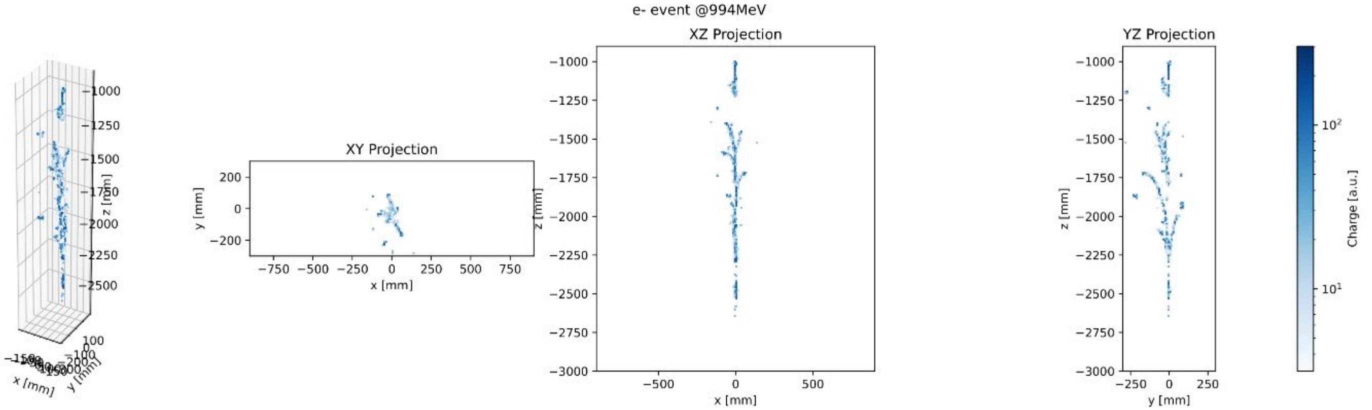



# Simulation data generated for the training

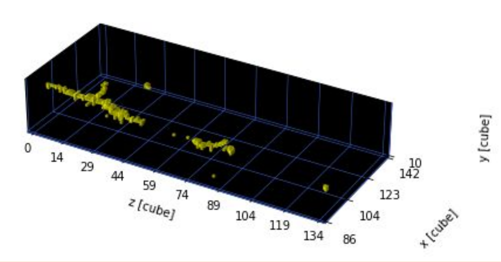
**e-**

convertion for python:



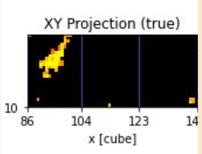



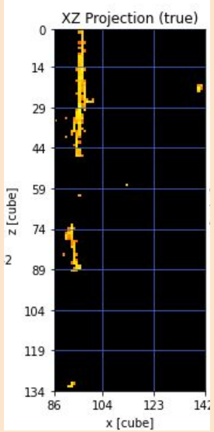
# First look at some training results (very preliminary; just to show it runs)

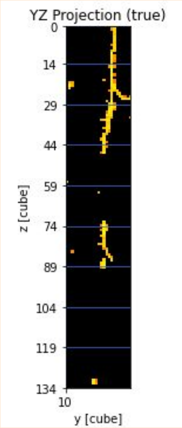


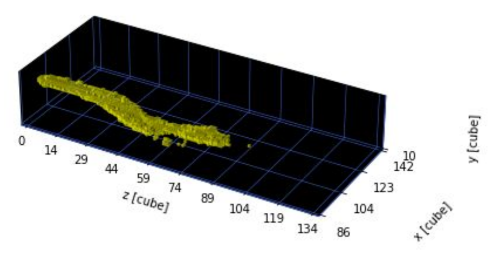

**True $\gamma$**

(VAE input)



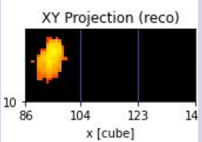



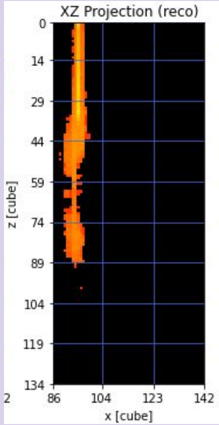



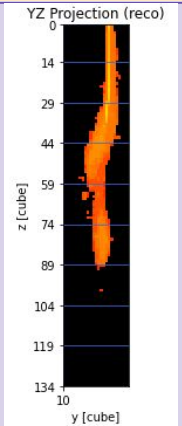



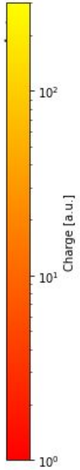

**Reconstructed $\hat{\gamma}$**

(VAE output)



# Conclusion

- We want to reject the gamma background using a VAE to perform anomaly detection
- Simulation data was generated for training (gamma) and testing (e-)
- The training of the VAE runs
- Distinction metrics: the VAE reconstruction error or the latent space itself (expect different distribution for e- than for gamma)

**Next step is to analyse those 2 distinction metrics!**