Hadron Identification at the dRICH Detector Using Deep Learning

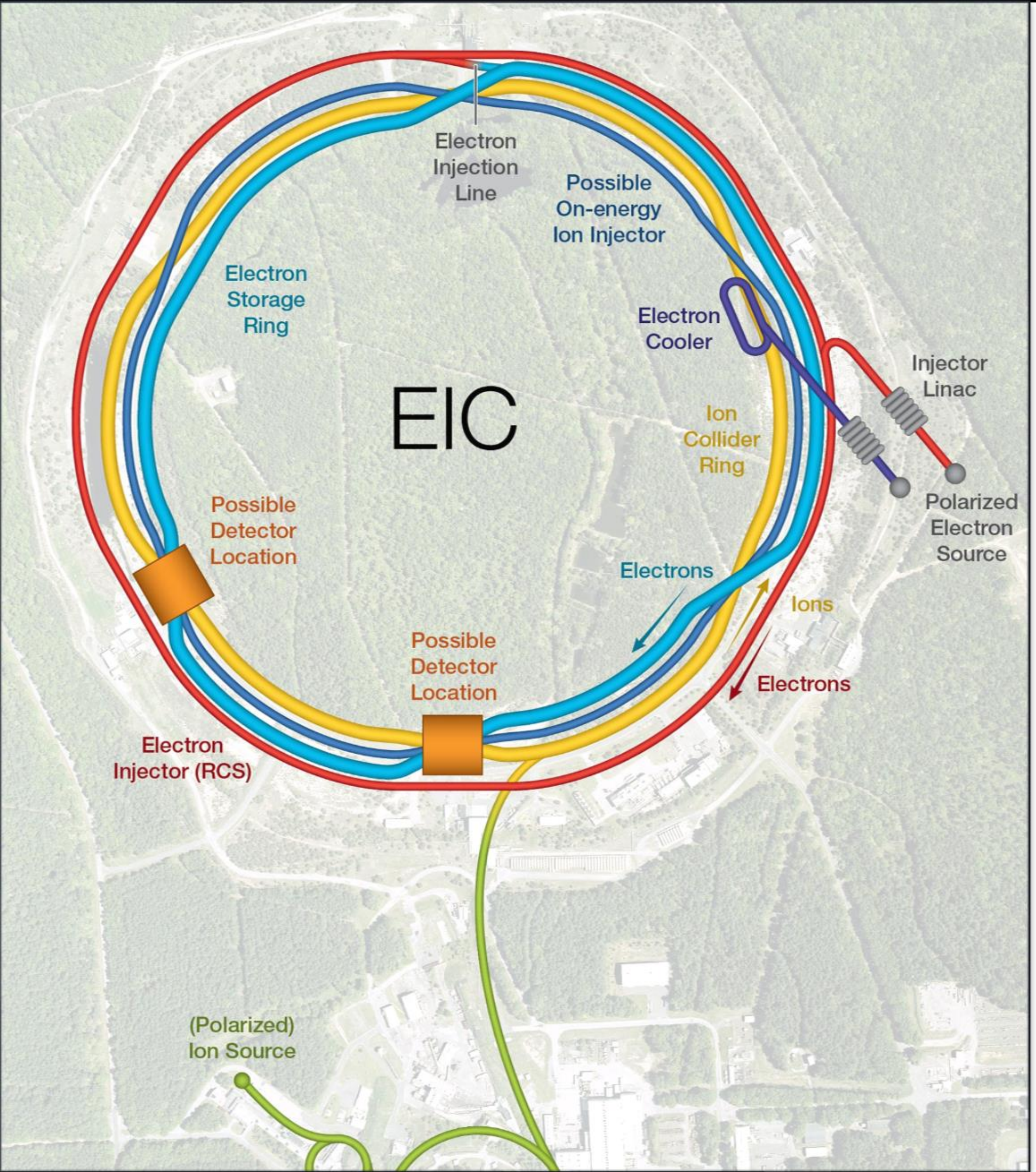
By Omar Hassan (hassano1@uvic.ca)

Supervisors: Dr. Wouter Deconinck (UManitoba)

Dr. Zhiyang Zhou (UManitoba)

27/07/2023

Outline



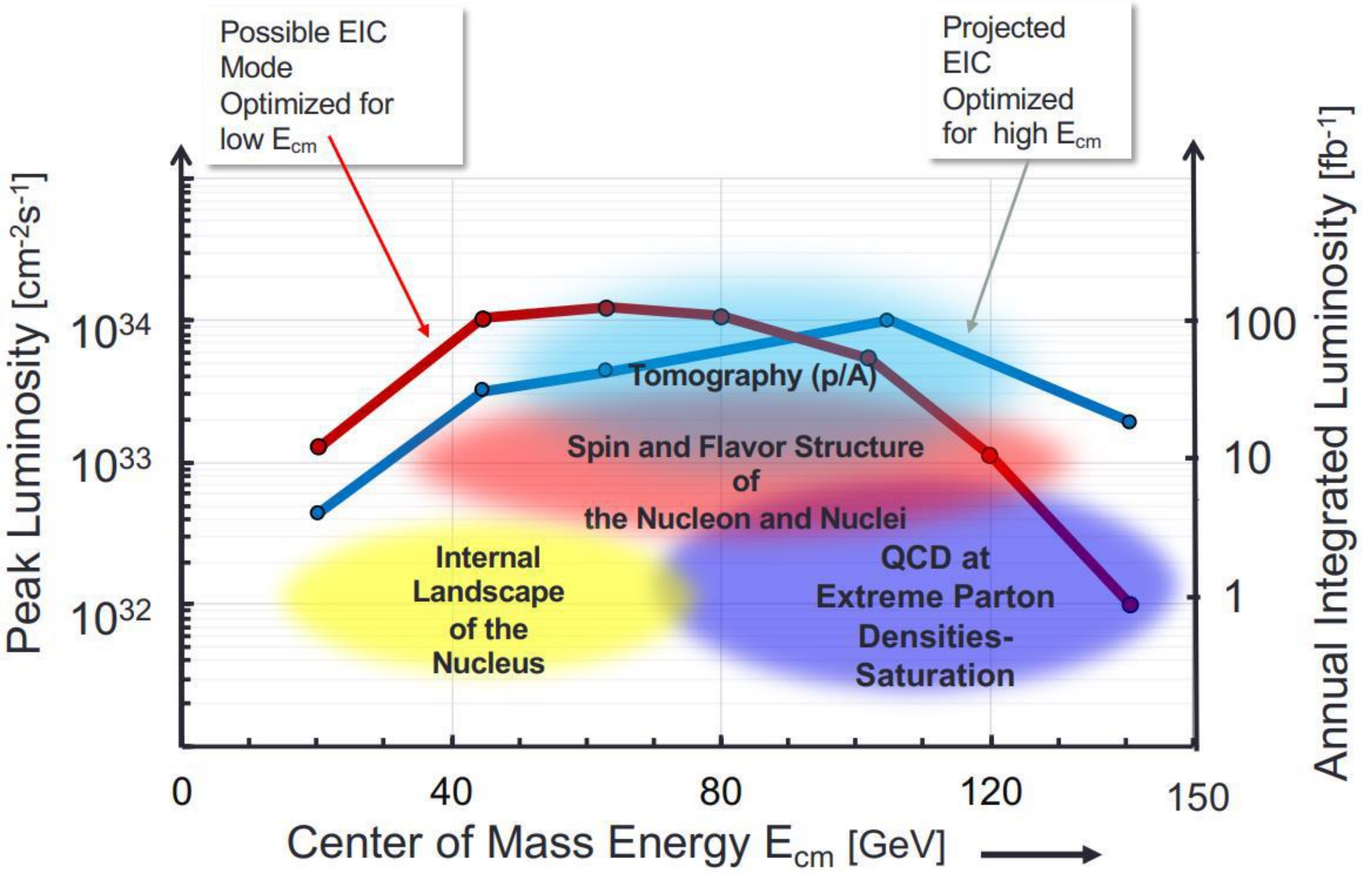
1. Review:
   a) Background
   b) Motivation
   c) Data Cleaning
   d) Convolutional Neural Networks
   e) Original working model

2. Sparsification:
   a) Sparse Tensors
   b) Minkowski Engine
   c) Hyperparameter Optimization
   d) New model

3. Results & Discussion:
   a) Model performance
   b) Limitations
   c) Future work
   d) Q&A

Background

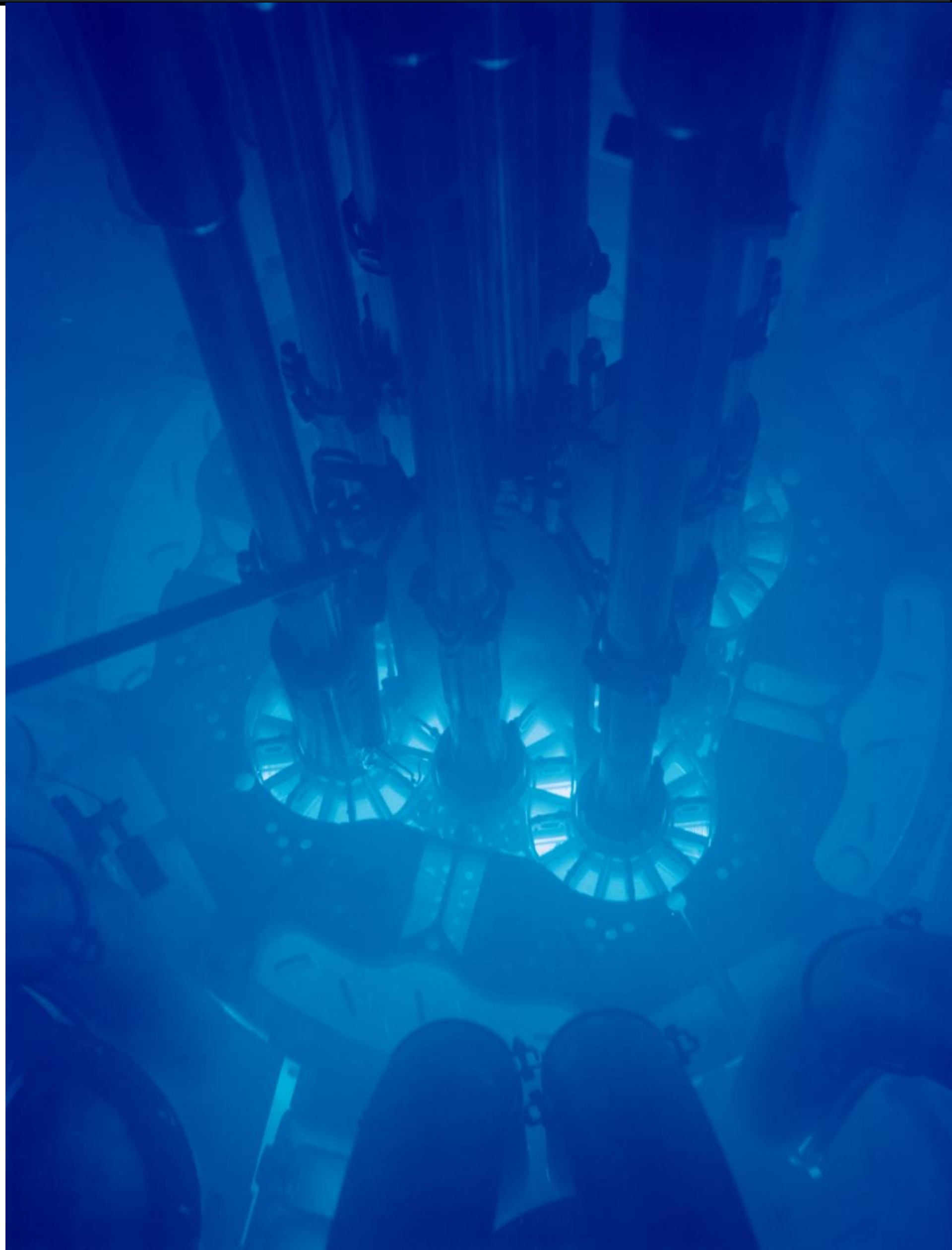
- Brookhaven National Laboratory proposed the Electron-Ion Collider which is scheduled to be built in the 2020s.
- The EIC facility is a high polarization and high luminosity collider.
- Home to the dRICH detector, the focus of our studies.



Rahman, S. Deconinck, W. The Electron-Ion Collider (EIC) poster. EIC-Canada collaboration.

Cherenkov Radiation

- Speed of light is slower in a medium (phase velocity of light)
- Particles can move at higher speed than the phase velocity of light
- When this happens, a characteristic glow is produced. i.e., nuclear reactors
- Analogous to the sonic boom produced at supersonic speeds.



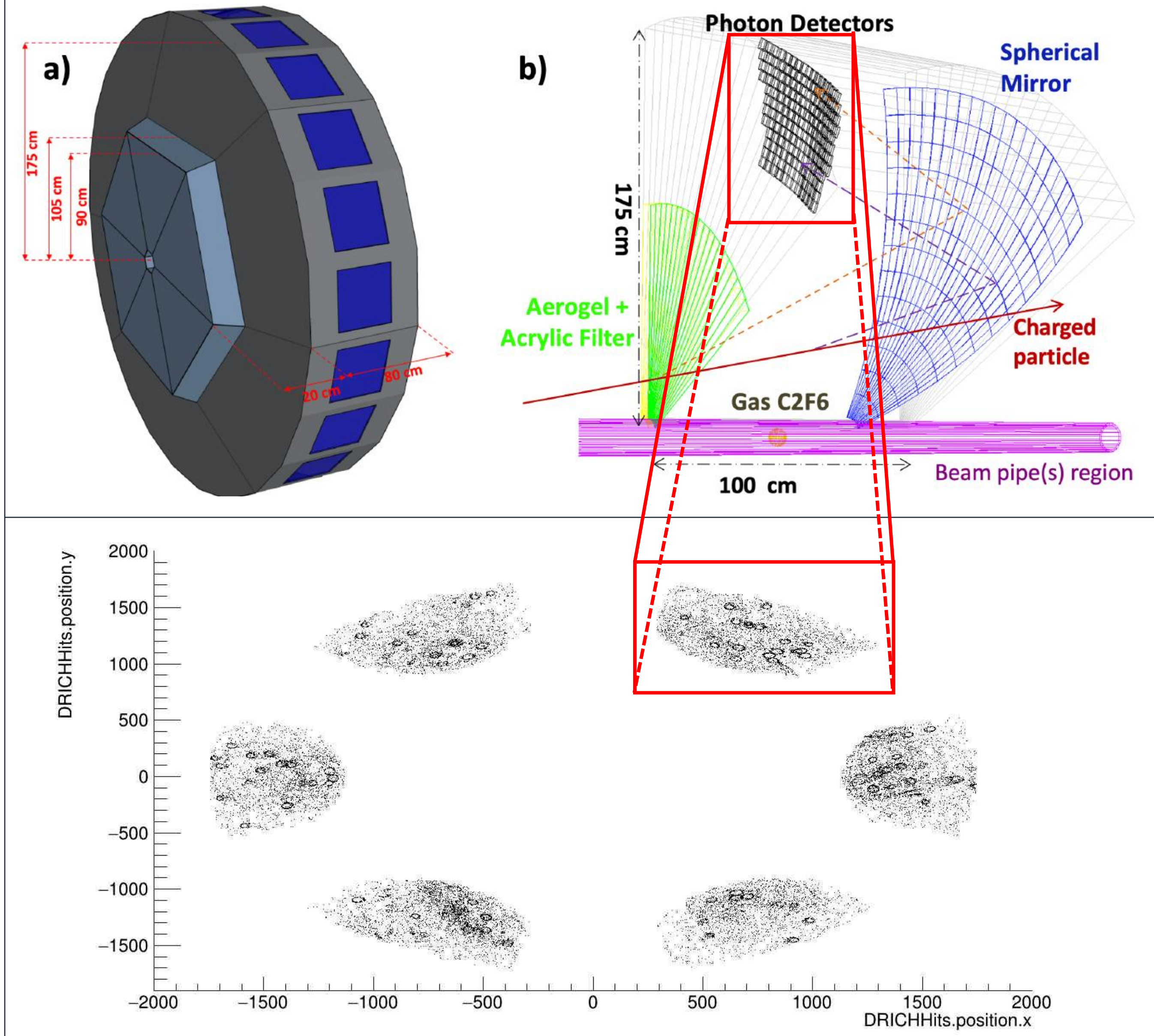
Argonne National Laboratory. (2009, April 8) Advanced Test Reactor core, Idaho National Laboratory. https://www.flickr.com/photos/35734278@N05/3954062594/

EIC dRICH Detector

- The dual-radiator Ring Imaging Cherenkov detector is a photoionization (PID) detector with powerful hadron identification properties (pions, kaons, & protons).

- Composed of 6 sectors, each sector containing aerogel and gaseous layer.

- The principle behind RICH detectors is to use Cherenkov radiation to produce rings of different sizes (depending on particle type).



Khalek, R. et al. (2021, October 27). *Science requirements and detector concepts for the electron-ion collider: EIC yellow report*. https://arxiv.org/abs/2103.05419

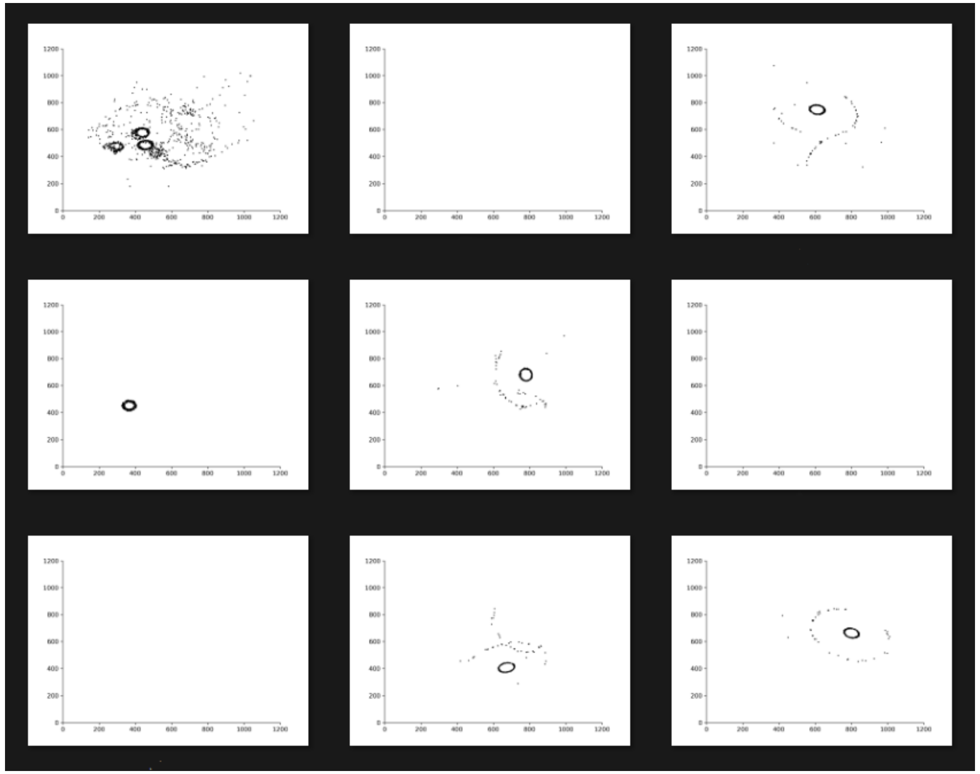
Bock, Friederike. et al. (2021, December 1) *ECCE Particle Identification*. ecce-note-det-2021-04v1.0

Motivation

- Our primary motivation is to provide a continuous method of particle identification with high accuracy.
- Data generated using DD4hep by CERN. ROOT file created containing the specified number of events.
- Sparsification of our model.
- 80/10/10 split for training/validation/test datasets.

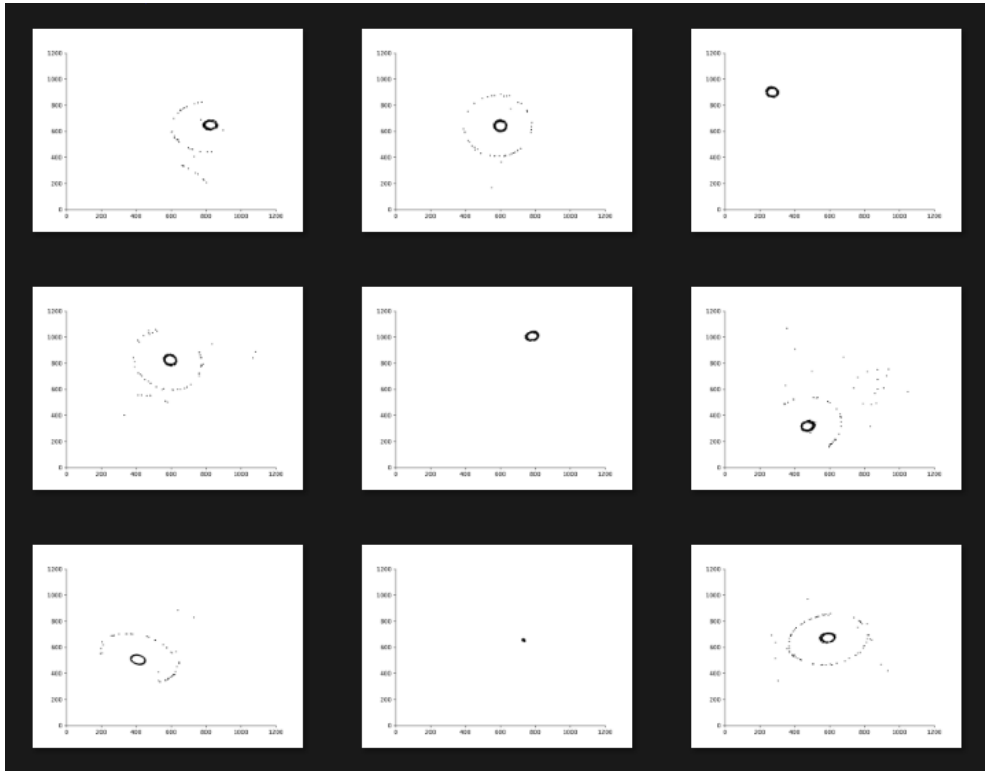
Data Cleaning

Unfiltered Electron Data



Removing Empty Events & events with too many or too little points

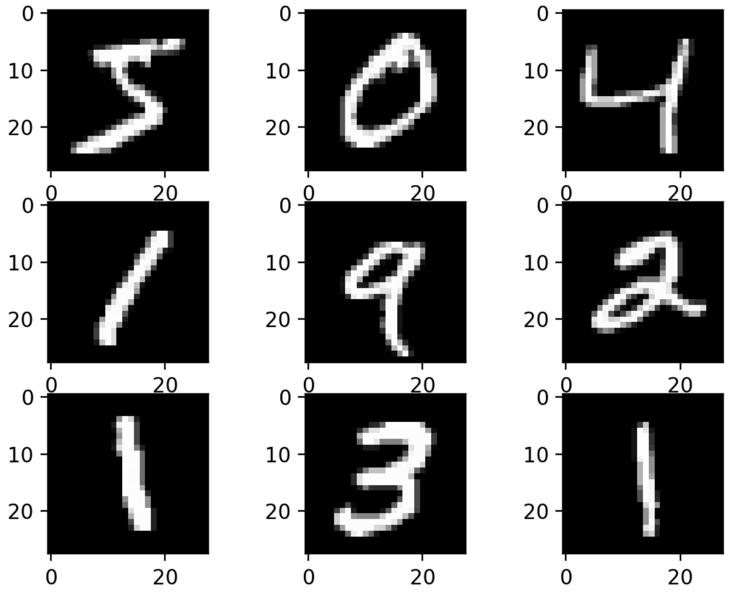
Filtered Electron Data



- Lots of data lost (~ 60% of events are empty)
- Filtered data is not perfect but good enough.

Convolutional Neural Networks (1/3)

- CNNs have remarkable image recognition properties, which make them very suitable for this problem.
- MNIST is an example of a dataset where CNNs perform very well, with an error rate of 0.09%



Brownlee, J. (2021, November 14). *How to develop a CNN for mnist handwritten digit classification*. MachineLearningMastery.com. Retrieved December 12, 2022, from https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/

Convolution Layer

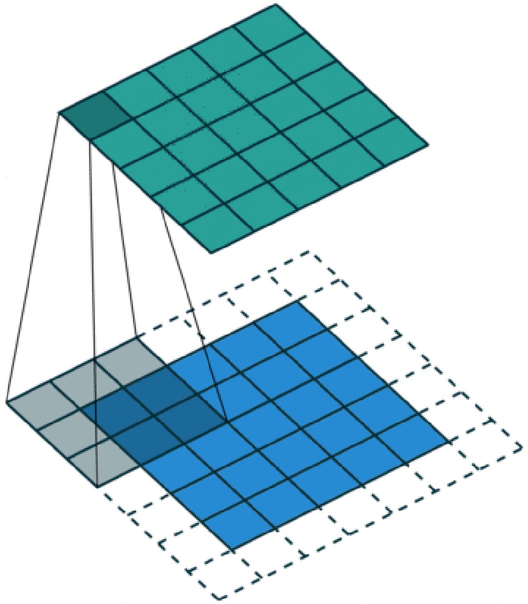
For:
n = input image size
f = filter size
p = padding size
s = stride size

We get:

$$n_{out} = \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1$$

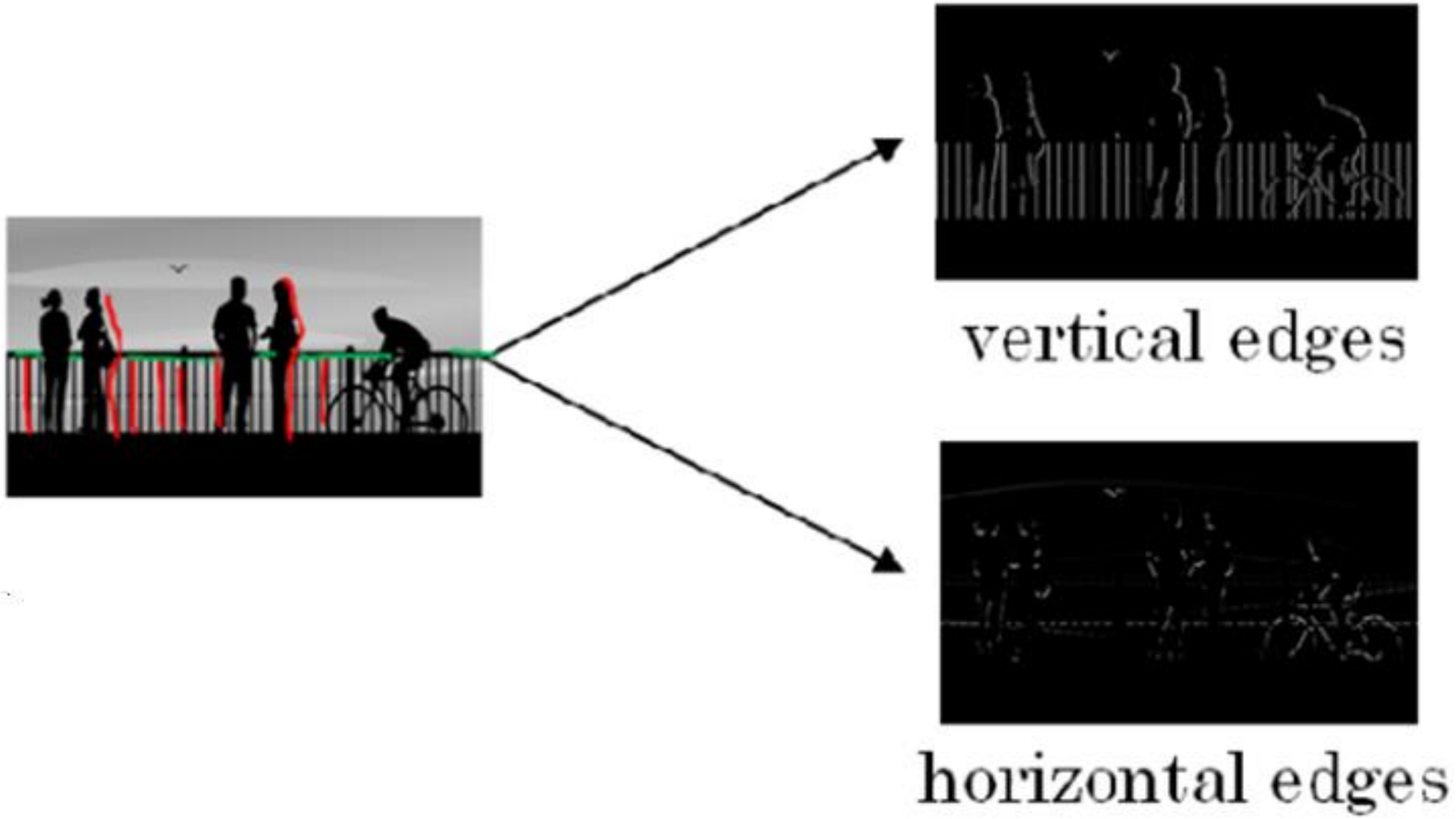
As an example:

$$n_{out} = \left\lfloor \frac{5 + 2(1) - 3}{1} \right\rfloor + 1 = 5$$



P, Pröve. (2018, February 7). *An introduction to different types of convolutions in deep learning. Towards Data Science.* https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

Convolutional Neural Networks (2/3)

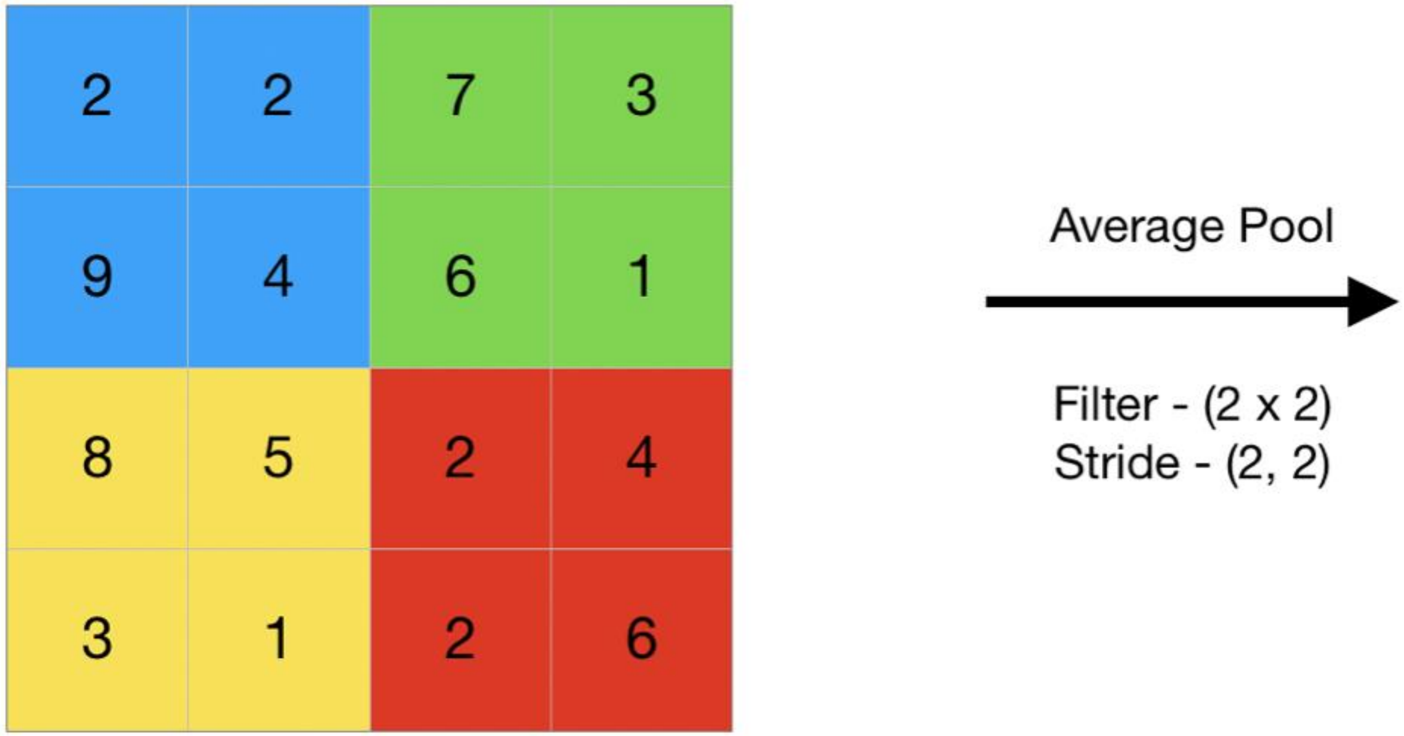


Pant, P. (2019, November 21). *CNN: Understanding edge detection with an example.* Retrieved December 13, 2022, from https://pradeeppant.com/2019/11/21/cnn-understanding-edge-detection-with-an-example.html

Convolutional Neural Networks (Pooling)

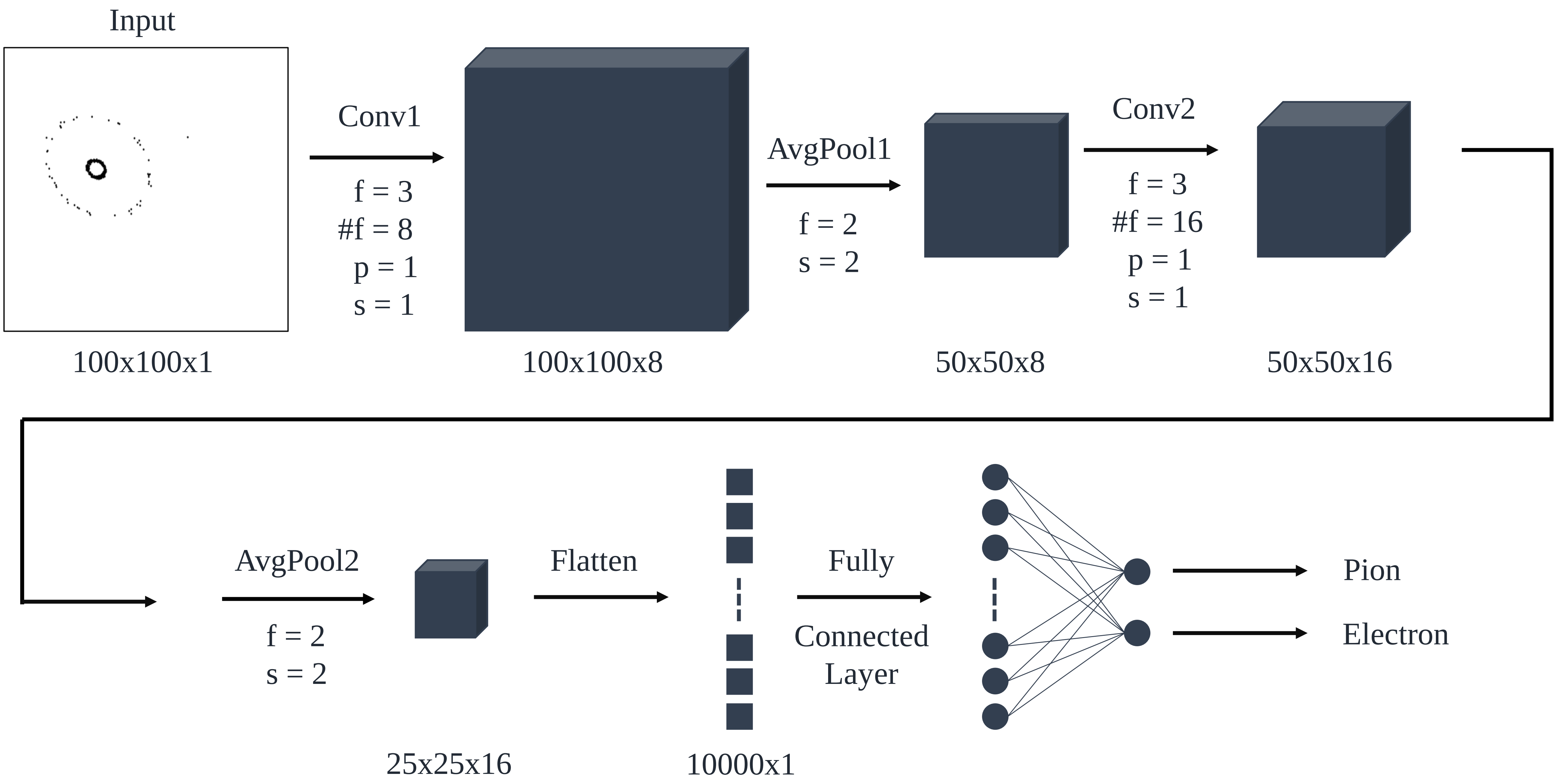
- Pooling layers are used for dimension reduction: retain features of the image while decreasing image size.
- Multiple types of pooling, such as max pooling and average pooling.

Pooling Layer



Khosla , S. (2022, August 24). *CNN: Introduction to pooling layer*. GeeksforGeeks. Retrieved December 12, 2022, from https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

Original Working Model



Sparse Tensors

- Dense tensors become too computationally intensive for a larger number of events.
- A more memory efficient and faster way to store our data is needed.
- Sparsifying our data allows us to do this without a loss in data.
- The amount of memory saved relies on how sparse the data is, but there is at least a 200-fold memory efficiency in sparsification [1].

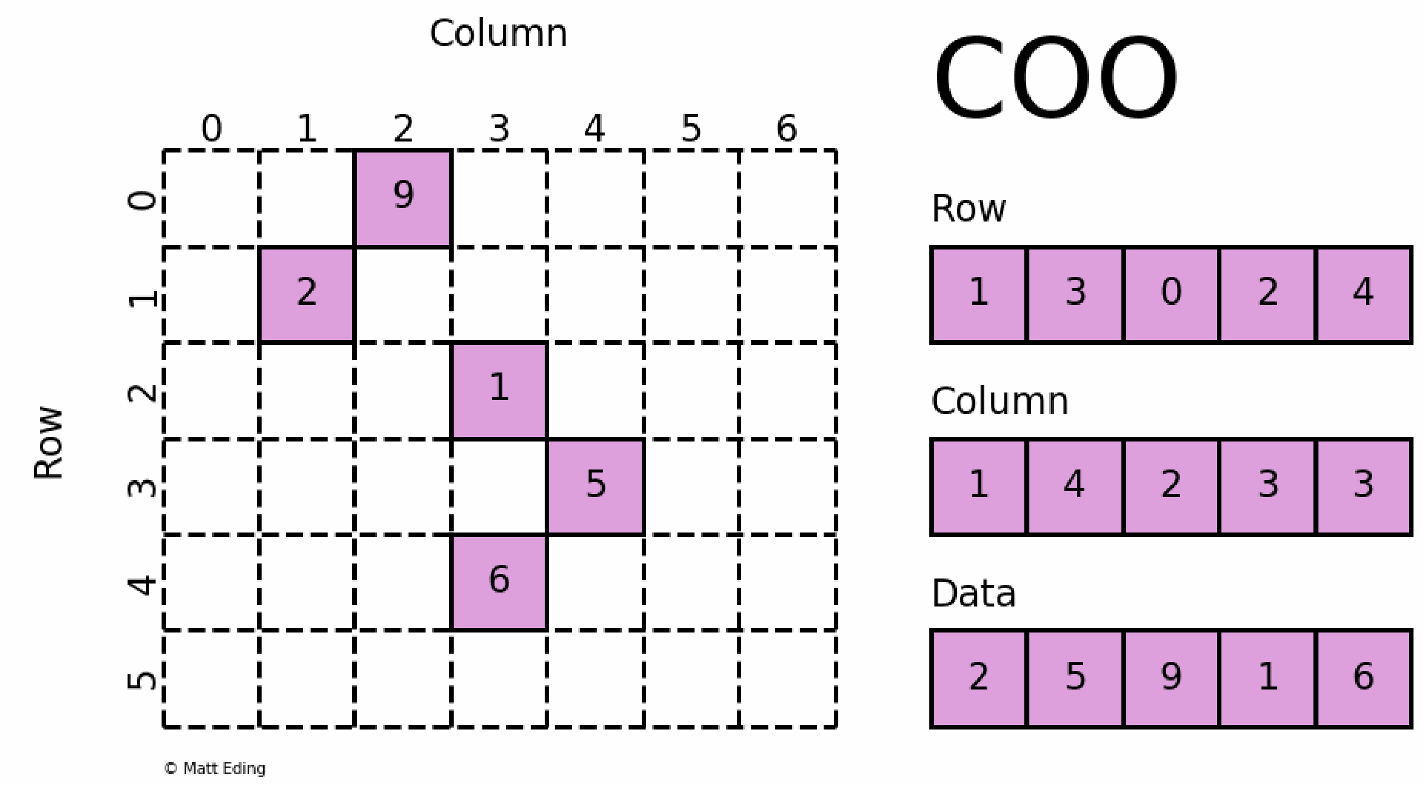
[1] *Torch.sparse*. torch.sparse - PyTorch 2.0 documentation. (n.d.). Retrieved April 10, 2023, from https://pytorch.org/docs/stable/sparse.html#sparse-coo-docs

The Sparse COO Format

- There are a number of formats to store sparse data, for example the coordinate format (COO) and the CSR/CSC format.

- The COO format is used by Minkowski Engine so we will focus on it.

- The COO format stores both the row and column coordinates, alongside the value at that coordinate.



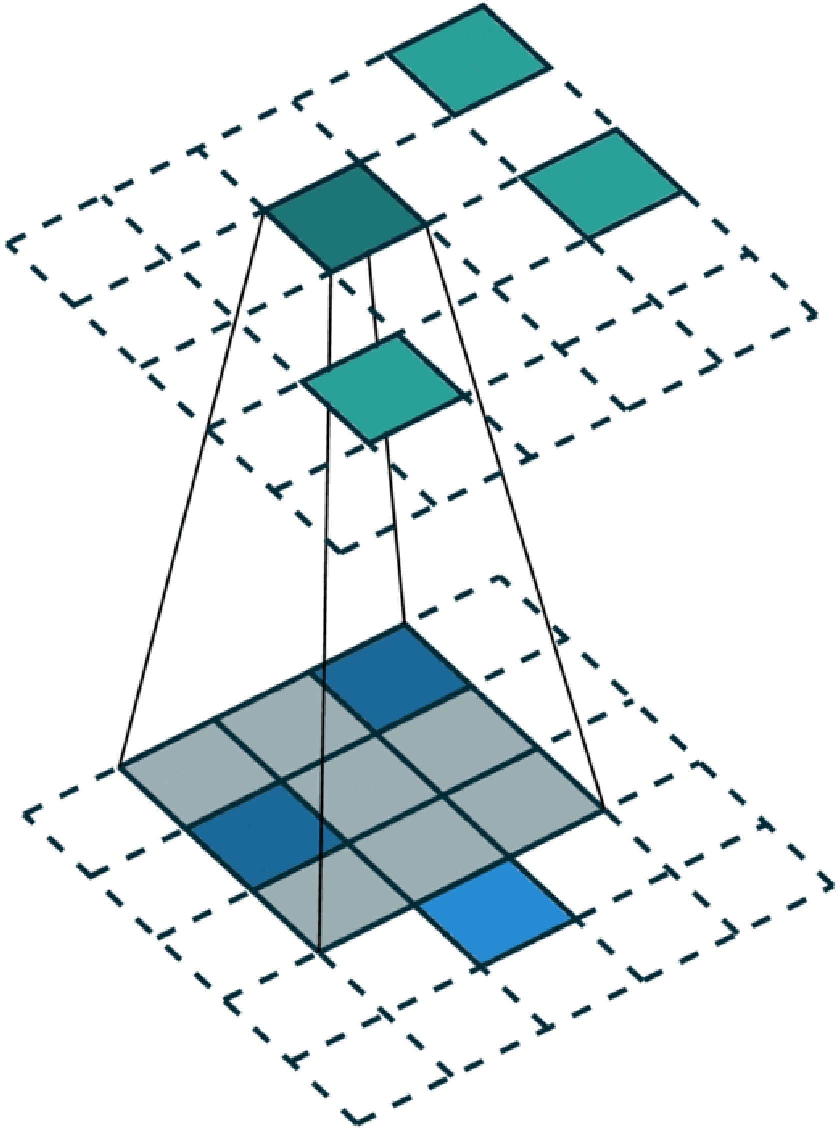
Eding, M. (2019, April 25). *Sparse matrices*. Sparse Matrices · Matt Eding. Retrieved April 7, 2023, from https://matteding.github.io/2019/04/25/sparse-matrices/#coordinate-matrix

Minkowski Engine

- The major frameworks for machine learning lack native support for convolutions with sparse data.

- This leads to using an external library, we opted for Nvidia's Minkowski Engine.

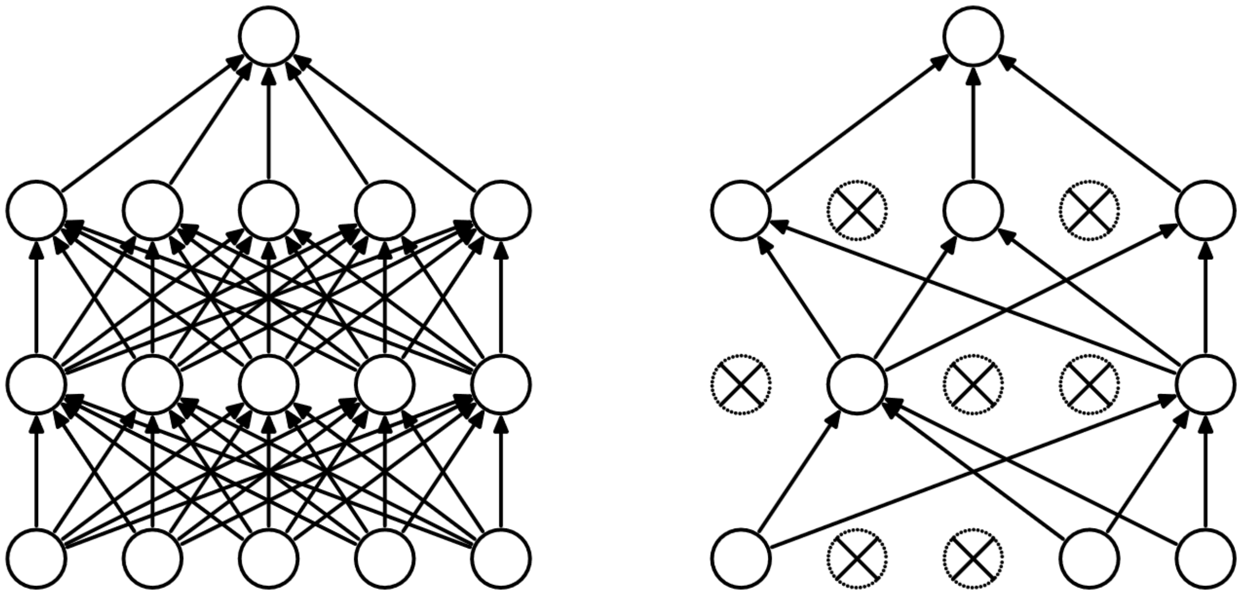
- Data is quantized.



Choy, C. et al. (2019). *4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks*.
https://arxiv.org/abs/1904.08755#

Overfitting & Dropout

- Overfitting is a common issue where the model learns the training data too well. This causes the model to fail with new data.
- Validation dataset allows us to test the model on new data during the training process.
- Dropout layers address this issue by deactivating some neurons during the training process.



(a) Standard Neural Net (b) After applying dropout.

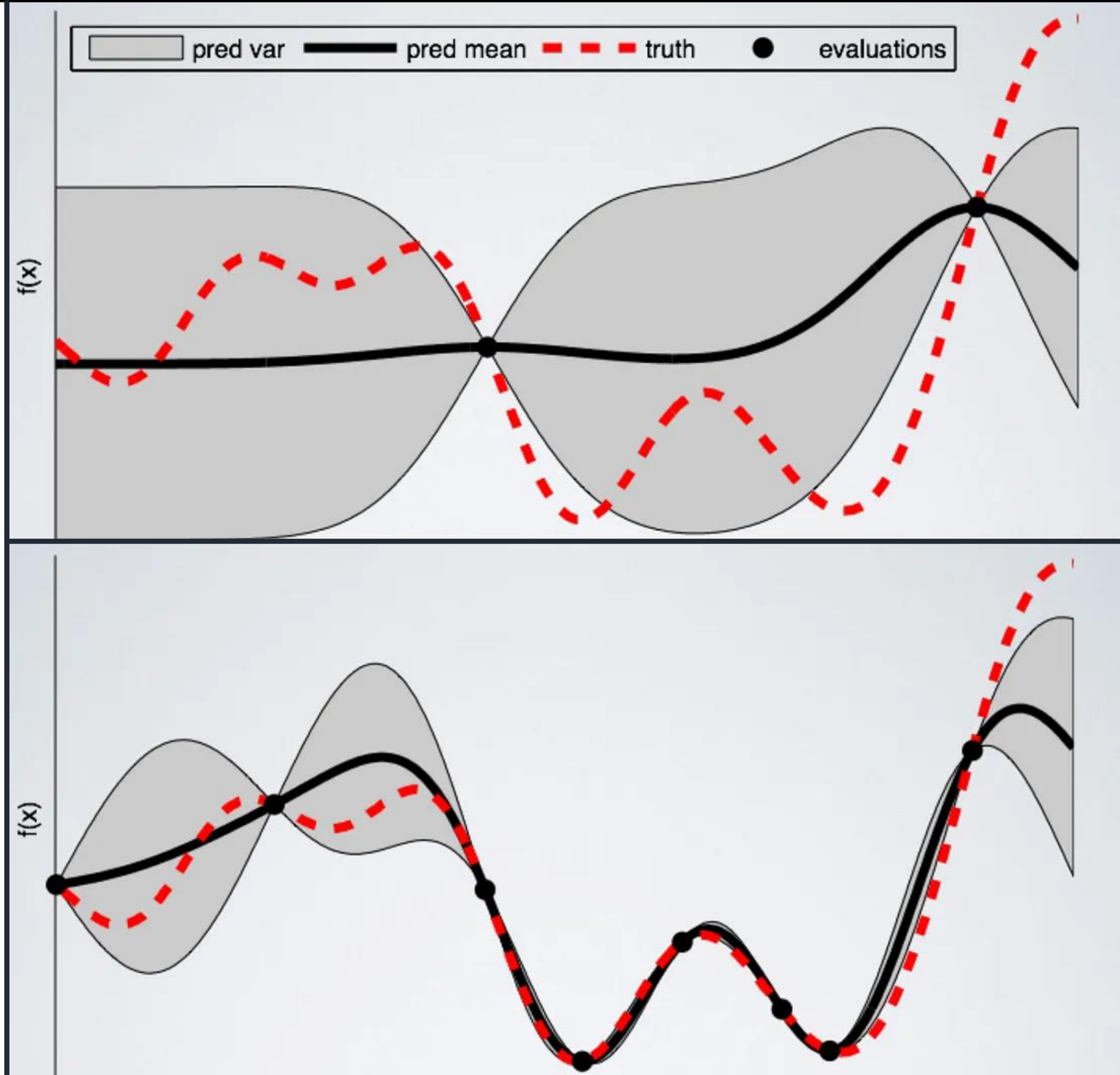
Srivastava, N. et al. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

Hyperparameter Optimization

- Manually adjusting hyperparameters is a tedious process and unlikely to give the best hyperparameters to train our model.

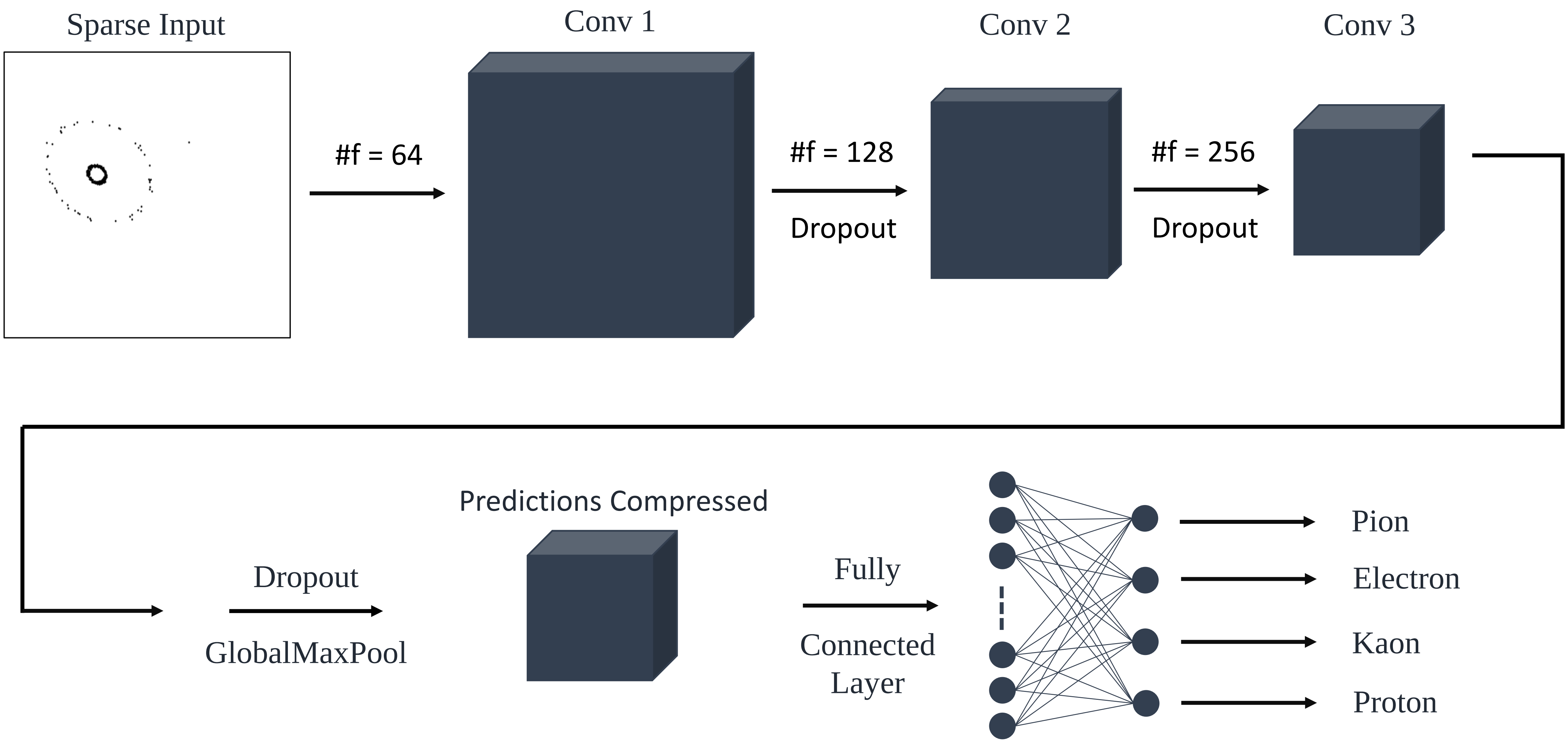
- Hyperparameter optimization methods offer an automated and methodical way by which we can find the best hyperparameters to minimize our loss.

- We use Bayesian hyperparameter optimization, a method which uses prior results to better predict the parameters.



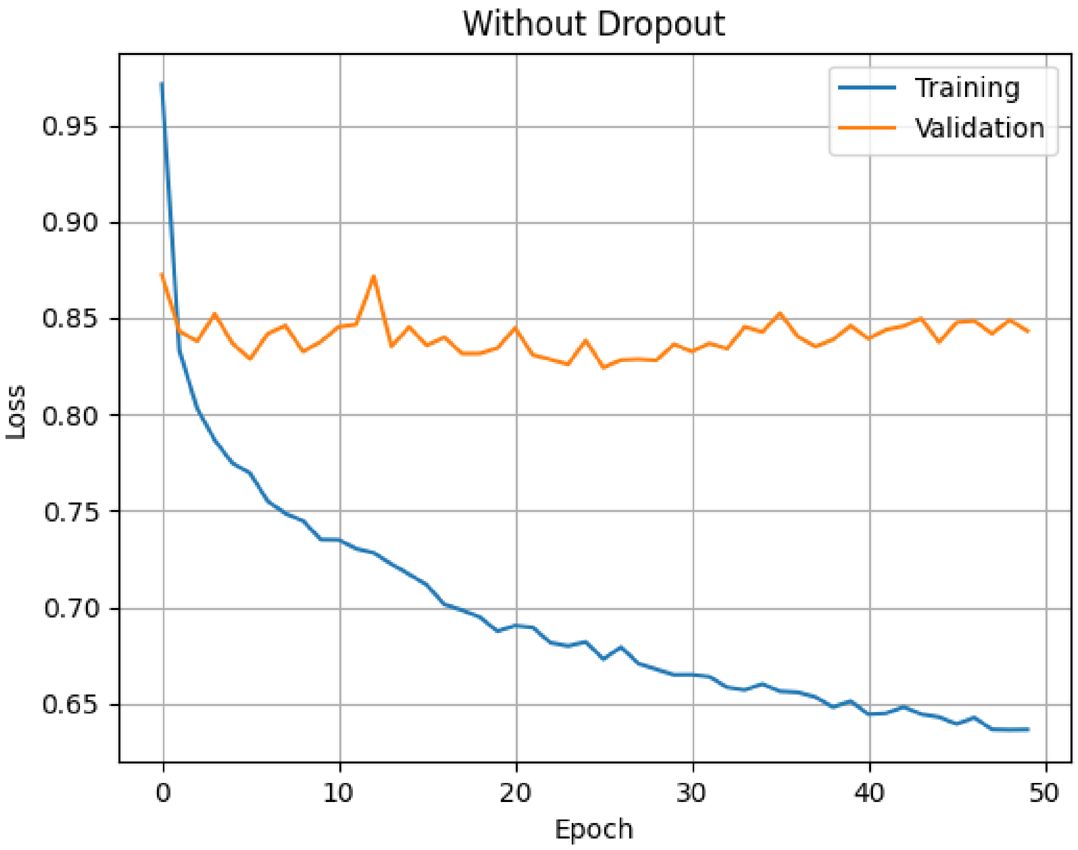
Koehrsen, W. (2018, July 2). *A conceptual explanation of Bayesian hyperparameter optimization for Machine Learning*. Medium. Retrieved April 10, 2023, from https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f

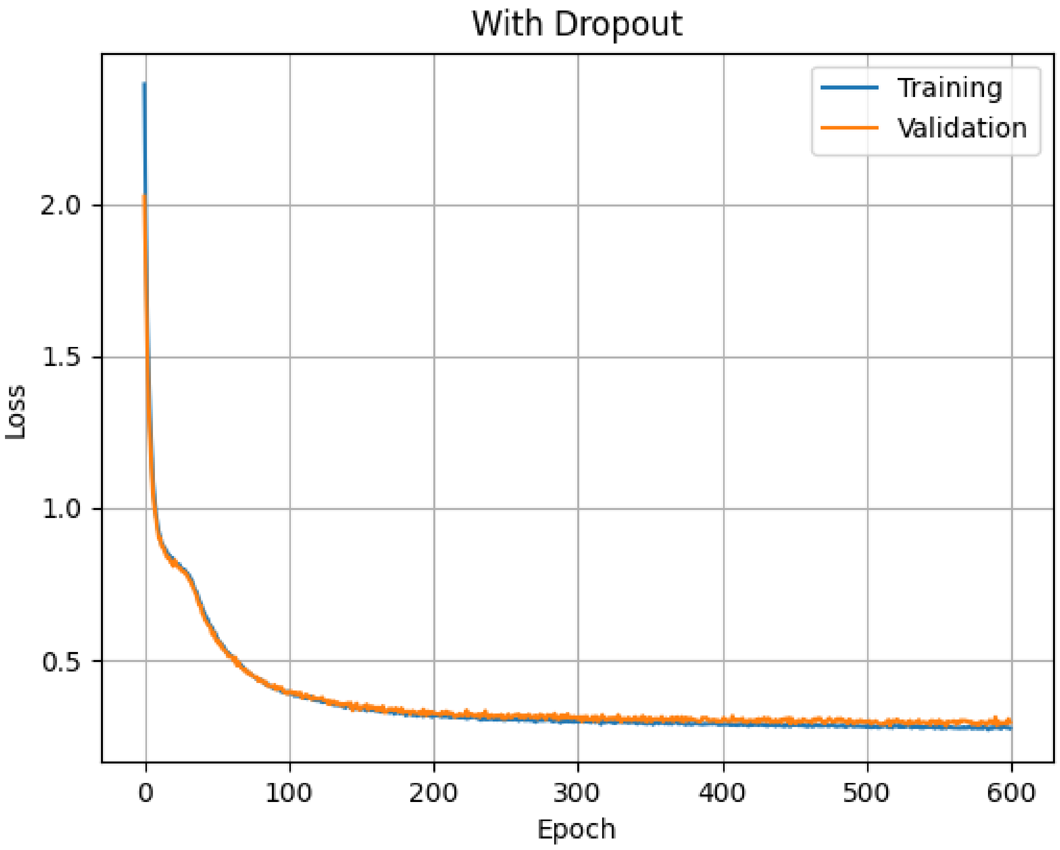
New Model


Sparse Input
Conv 1
Conv 2
Conv 3
#f = 64
#f = 128
Dropout
#f = 256
Dropout
Dropout
GlobalMaxPool
Predictions Compressed
Fully
Connected
Layer
Pion
Electron
Kaon
Proton

Model Performance (1/2)

- Adding dropout layers between the hidden layers addressed our overfitting problem.



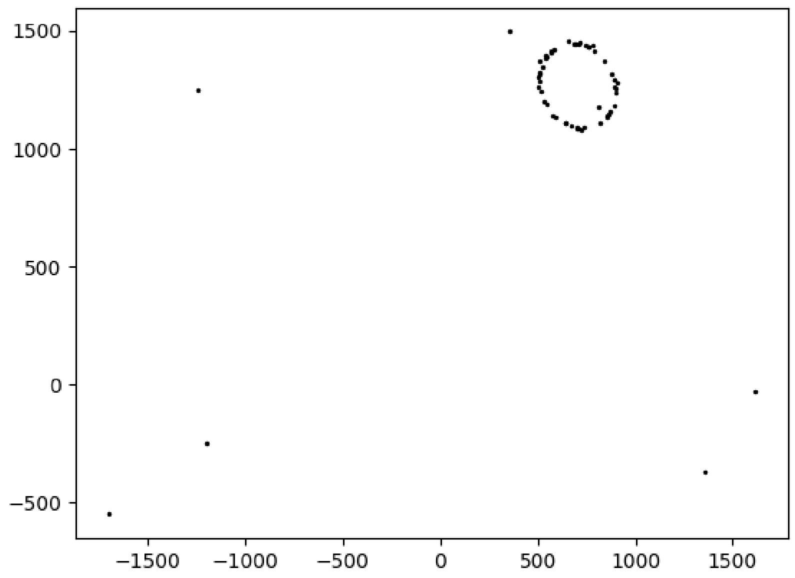


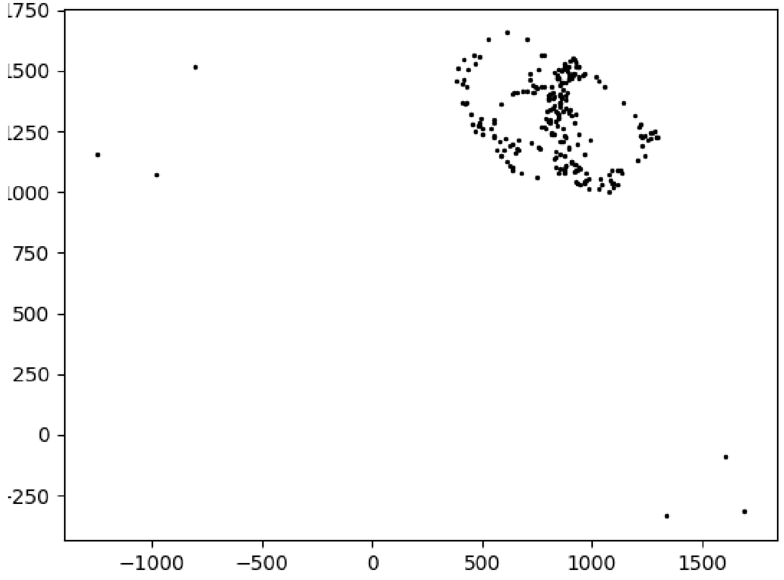
Model Performance (2/2)

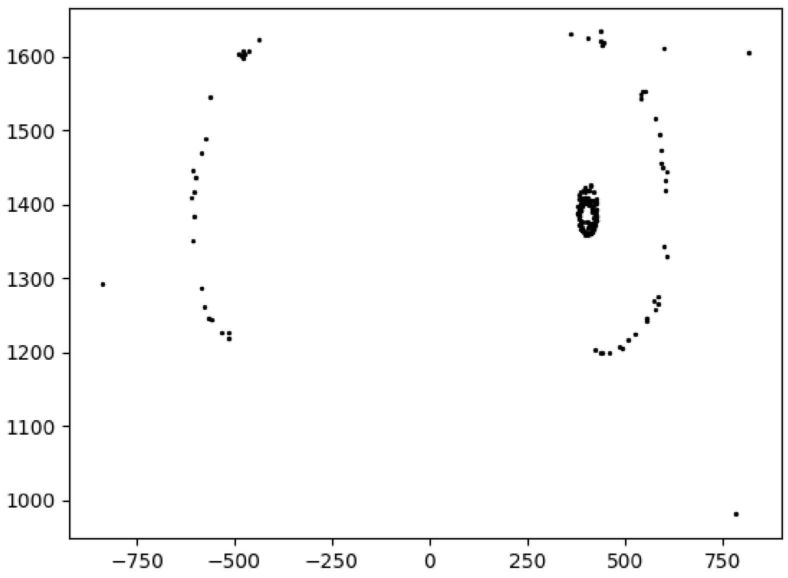
- With the magnetic field disabled, we get an accuracy of 96.75% for the test dataset.
- With the magnetic field enabled, we have a 93.82% accuracy for the test dataset.
- As expected, the magnetic field introduces more noise which decreases the accuracy of our model.

Particle	Events	Accuracy
Pion	9677	95.44%
Electron	8339	95.57%
Kaon	2635	76.81%
Proton	215	4.55%

Kaon Performance

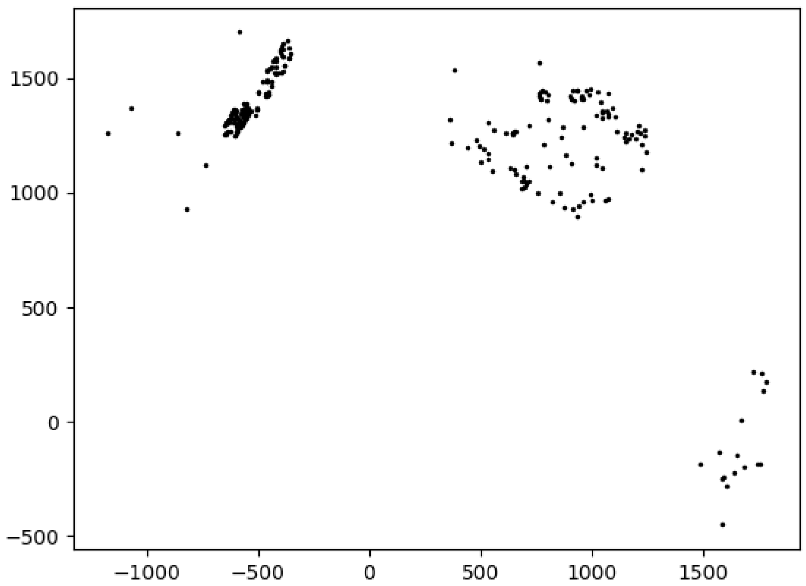


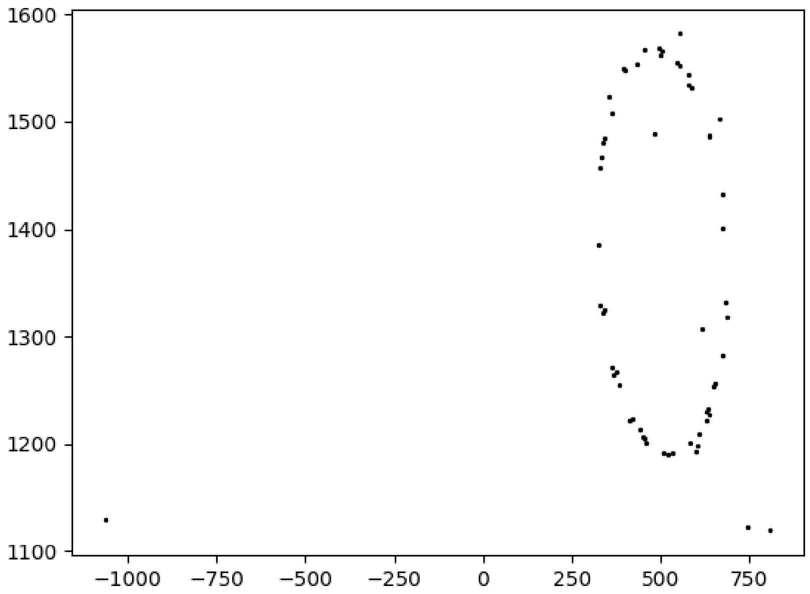


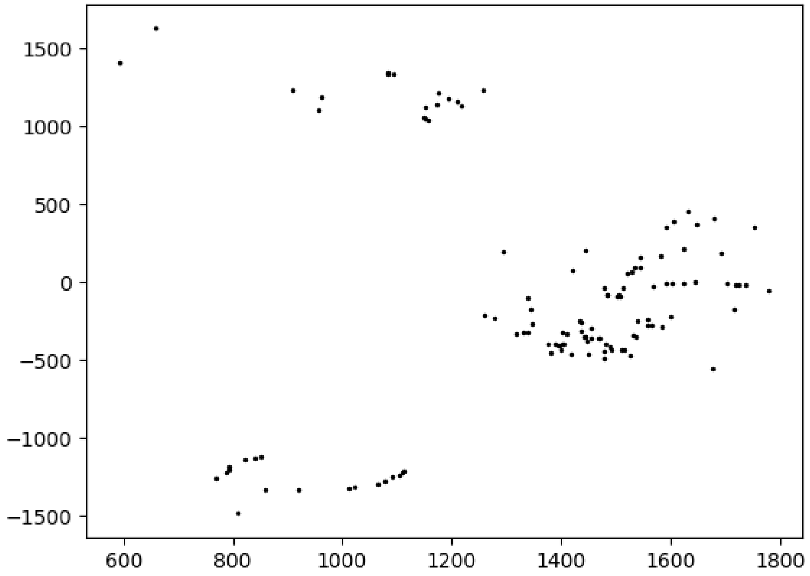


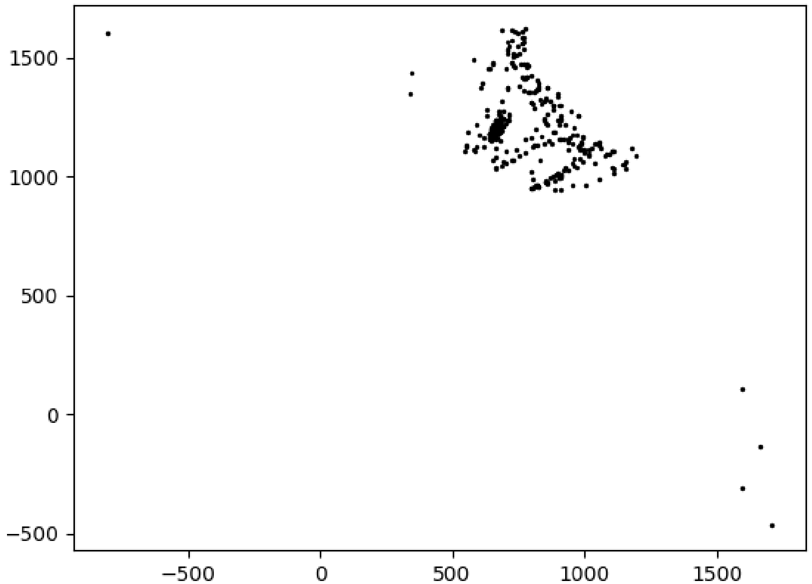
Proton Performance

- The proton events are highly irregular and too noisy. Independent of magnetic fields.
- The low number of events leads to less training data and therefore a much worse accuracy.
- A few good events stand out but it is overwhelmingly comprised of non-rings.









Limitations

- An octree-based framework like O-CNN outperforms Minkowski Engine in both memory efficiency and speed.

- Datasets are not evenly balanced, protons had very few events remaining after cleaning relative to the rest of the particles.

- The number of points per event will be affected once the quantum efficiency of the detectors, dead areas between pixels, and safety factor are considered. This is because we don’t have an exact pixel count since it isn’t a flat grid.

Future Work

- Make sure that the model has equal data for each particle type.
- Replace Minkowski Engine with O-CNN for memory and speed benefits.
- Implement momentum as another factor during training, alongside coordinates.
- Deploy model in C++ interface for speed boost.
- Improve the accuracy even further and deploy the model in the facility.

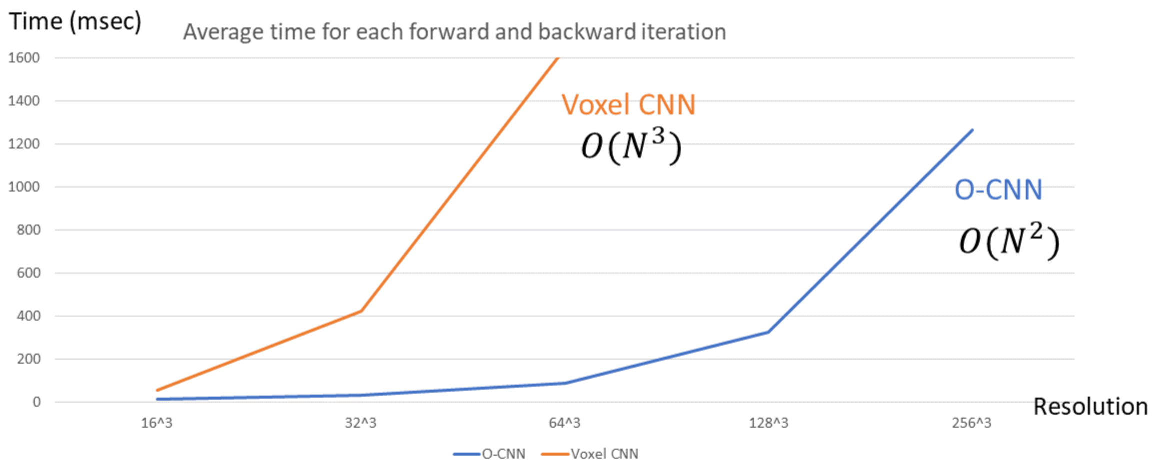
Acknowledgements & Resources

- I’d like to acknowledge Dr. Wouter Deconinck and Dr. Zhiyang Zhou’s help during my thesis.
- I’d also like to thank Sakib Rahman & Max Fatouros for their help with regards to hyperparameter optimization and dropout layers respectively.
- My code and thesis can be found at: https://github.com/ohassn/Particle-Identication-Using-CNNs

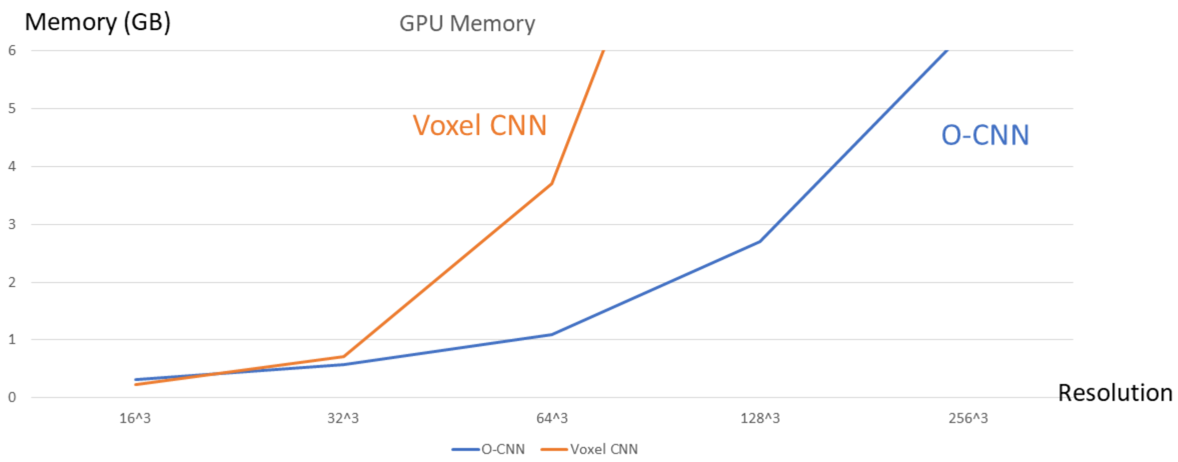
Q&A

Appendix (1/4)

Computational Efficiency

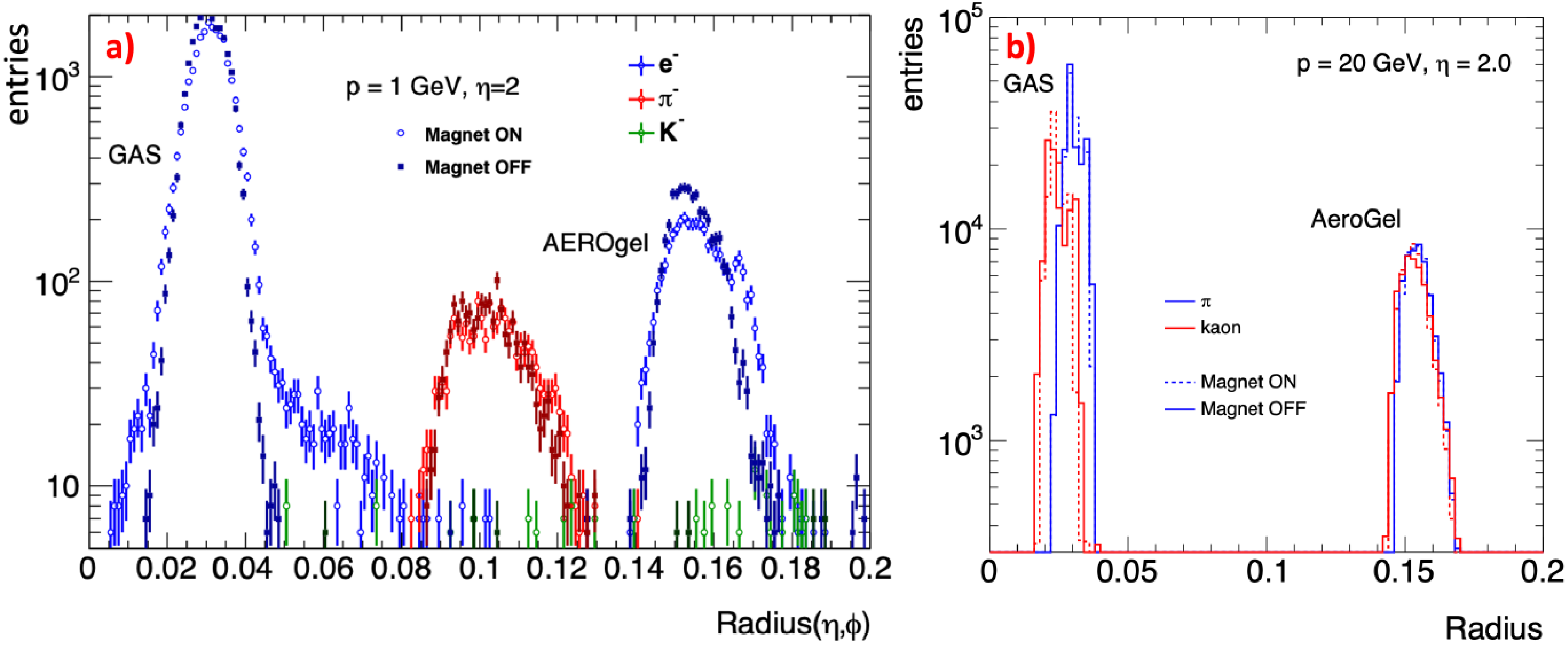


Memory Efficiency



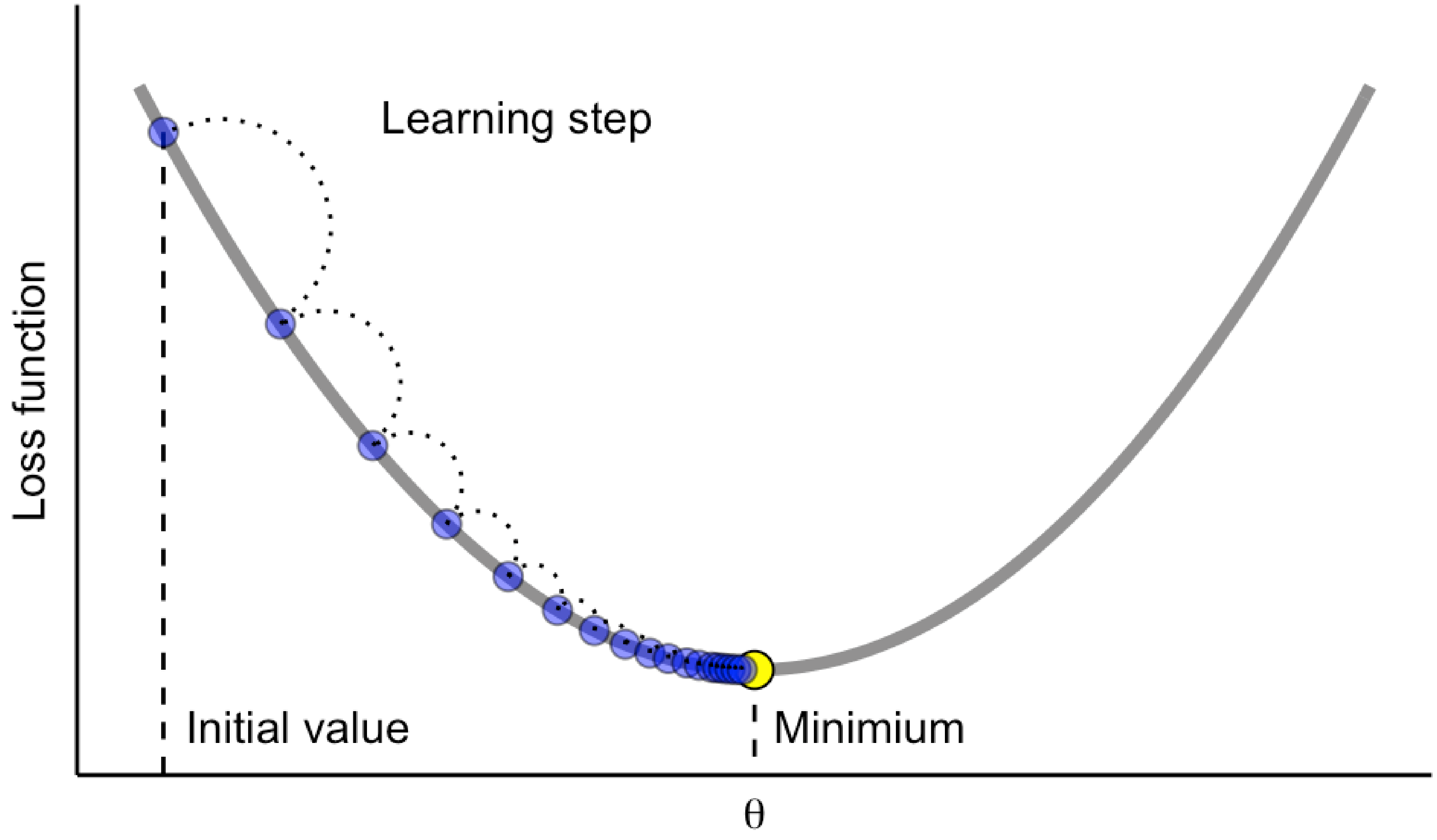
Wang, P. (2017). O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. [PowerPoint slides]. https://onedrive.live.com/view.aspx?resid=97002CD884825EBF!2092&ithint=file\%2cpptx&authkey=!ADgc5oQjg_dQbaQ

Appendix (2/4)



Bock, Friederike. et al. (2021, December 1) *ECCE Particle Identification*. ecce-note-det-2021-04v1.0

Appendix (3/4)



Keshtkar, K. (2021). Convolutional neural networks in computer-aided diagnosis of colorectal polyps and cancer: A Review. https://doi.org/10.20944/preprints202110.0135.v1

Appendix (4/4)

- Dropout layer percentage is unknown, as it is unstated in the documentation
- 600 epochs, 0.0001 learning rate.
- Average and standard deviation of multiple runs for accuracy is preferable to accuracy based on one run.