Cherenkov ring generator

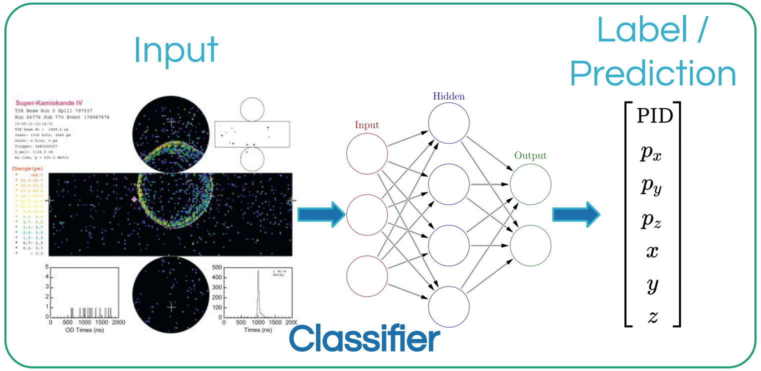
WatChMaL meeting

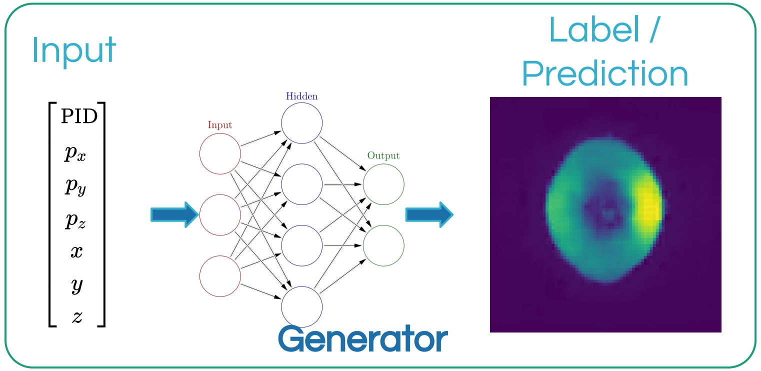
July 8 2019

Cristóvão Vilela

Introduction

- We are exploring an alternative approach to the more traditional "end-to-end" CNN event classification for reconstruction of water Cherenkov events.





- A CNN is trained to predict the hit charges and times for a given set of track parameters.
- This Cherenkov ring generator is incorporated into a maximum-likelihood estimation framework to form an event reconstruction algorithm.
  - This method is analogous to FiTQun reconstruction: the CNN replaces the parameterized charge and time pdf prediction.
- While I don't necessarily expect this method to outperform the end-to-end CNN classifier's accuracy, it has potential advantages in the context of physics analyses.

Why this might be interesting

- Single-ring predictions can be combined to predict arbitrary event hypotheses.
    - E.g.: in FiTQun mean predicted charges at each PMT are added up and time pdfs are combined, weighted by charge.
- Neural network can be trained on single-particle MC:
    - A priori not relying on problematic neutrino interaction and secondary interaction models.
    - Avoid multi-particle final states combinatorics.
- “Interesting” event topologies do not need to be defined at training stage.
    - Analyzers have flexibility to produce very specific event hypotheses out of single-ring predictions without having to retrain the neural network.
        - E.g.: proton decay to kaon and neutrino analysis with FiTQun specifies event with single de-excitation gamma followed (12 ns) by mono-energetic muon.
- This reconstruction approach would be a drop-in replacement for FiTQun.
    - Could be used with current analysis and systematic uncertainty estimation techniques.
    - Could be a useful first step in the move towards end-to-end ML reconstruction.

Generating images with CNNs

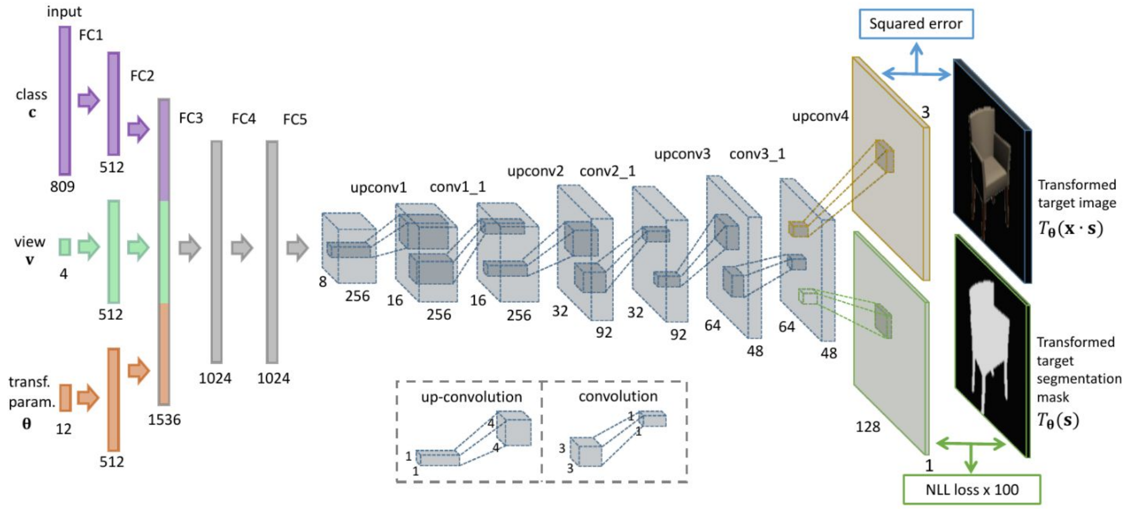
- First iteration of a Cherenkov ring generator neural network follows approach in

IEEE Trans. Pattern Anal. Mach. Intell. 39(4): 692-705, Apr 2017 (arXiv:1411.5928 [cs.CV]).

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 1

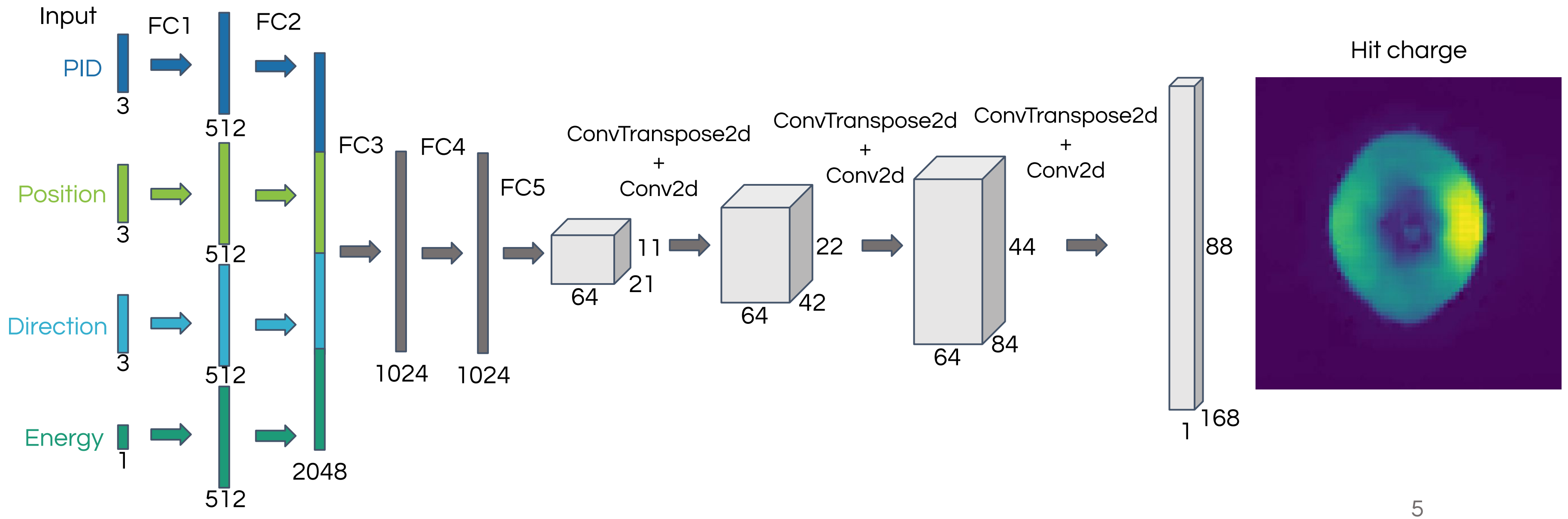
Learning to Generate Chairs, Tables and Cars with Convolutional Networks

Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, Thomas Brox



Generating rings with CNNs

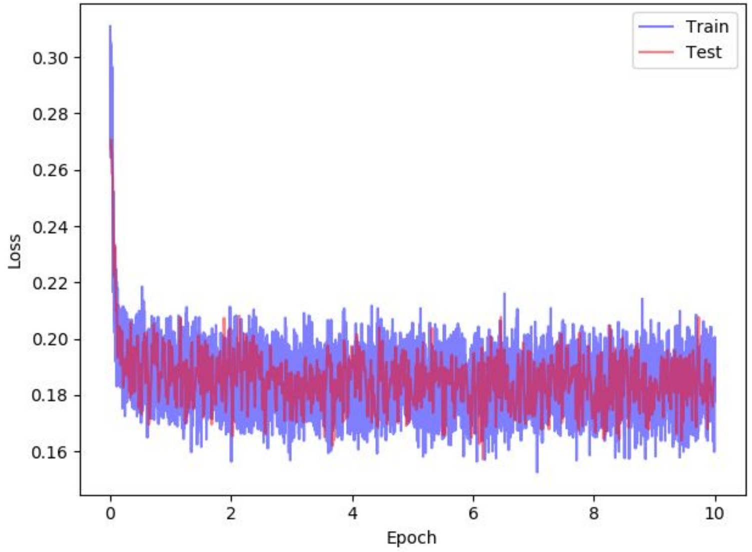
- Follow network architecture described in the paper as close as possible, output is the observed (mean) charge at each PMT in the barrel.
  - Almost certainly not optimal, just want to see if it works.
  - Implemented in PyTorch, based on Kazu's examples from the workshop.



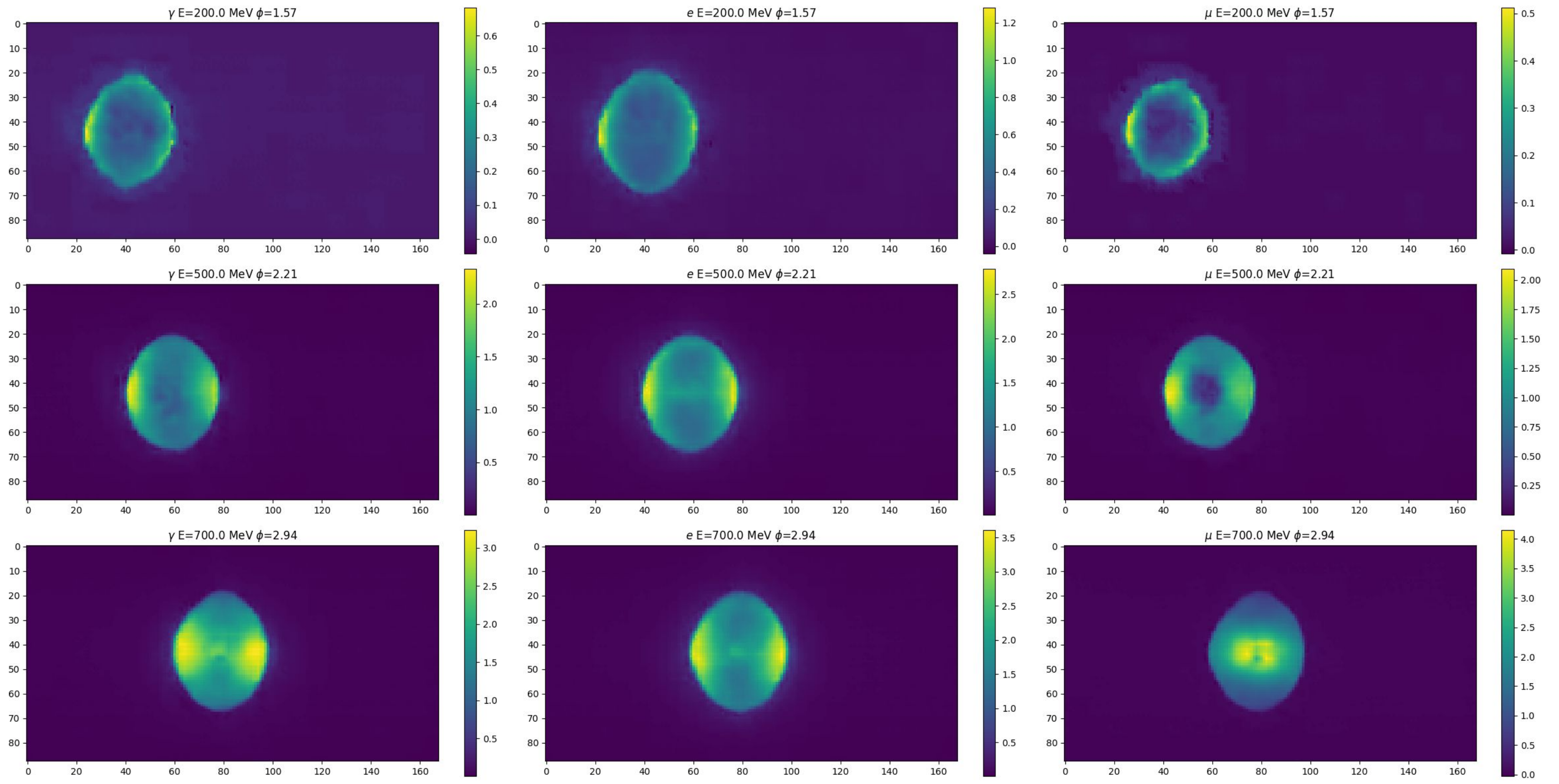
Training

- Used training sample prepared by Nick for the workshop:
  - 1M of each: electrons, gammas and muons.
  - Batch size: 200, train for 10 epochs (~day using PC w/ GPU)
  - SmoothL1Loss (Huber)
  - Adam optimizer

- Hitting loss "floor" due to variation in the samples themselves?
  - Move from predicting mean charge to predicting the charge pdf?
    - E.g.: Output of the network is a Gaussian mixture model?
  - Capture event-by-event variation with additional "latent" input parameters? Something along the lines of a VAE? Not sure if feasible…
- Maybe just terrible network architecture...



CNN-generated rings



Plans

- Ramp up on this work at Stony Brook over the coming weeks.
  - One or two students + myself, Mike Wilking and Chiaki Yanagisawa.
  - Focus will likely be toward T2K and Super-K reconstruction
- Short/medium term tasks:
  - Generate large training Super-K training sample using SKDETSIM.
    - Happy to share with this group (as long as it’s for Hyper-K use).
  - Investigate neural network architecture further:
    - Effects of layer size and number.
    - Try to implement pdf output, rather than mean charges.
    - Look into introducing latent features on the input?
    - Add hit times to the output - will need some kind of pdf output first, I don’t think mean hit times will work…
  - Run basic checks of how this would look like as a reconstruction tool.
    - Start with simple likelihood scans, using FiTQun to pre-process events.