Generating Cherenkov rings with convolutional neural networks

Initial proof-of-concept studies

T2K-SK Pre-meeting

July 24 2019

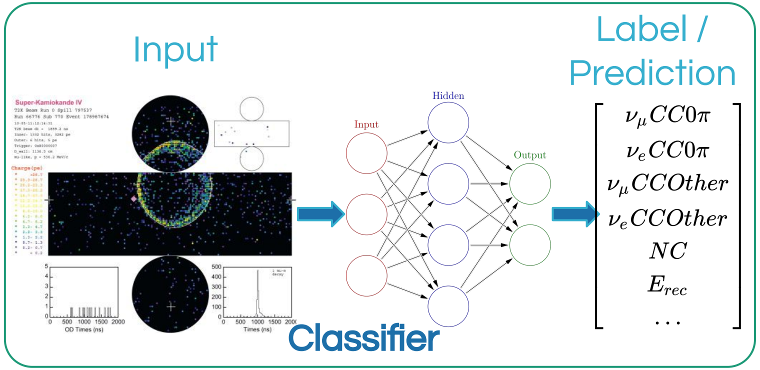
Cristóvão Vilela

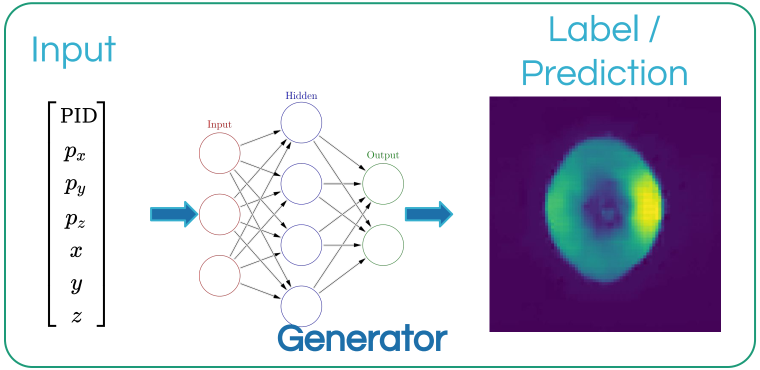
Introduction

- There is an effort to investigate using deep-learning techniques for water Cherenkov event reconstruction.
  - Mostly based in Canada/TRIUMF and focused on Hyper-K.
  - Kick-off workshop held at UVic in April.
- At Stony Brook, we've been thinking of ways to improve the FiTQun likelihood function for a while, including using machine learning techniques.
  - E.g.: looking into replacing 6D look-up table with boosted decision tree in order to further increase dimensionality.
- After attending the workshop, I've been looking into developing a maximum-likelihood reconstruction algorithm (i.e., like FiTQun) using convolutional neural networks to capture the likelihood function.

Reconstruction with CNNs

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





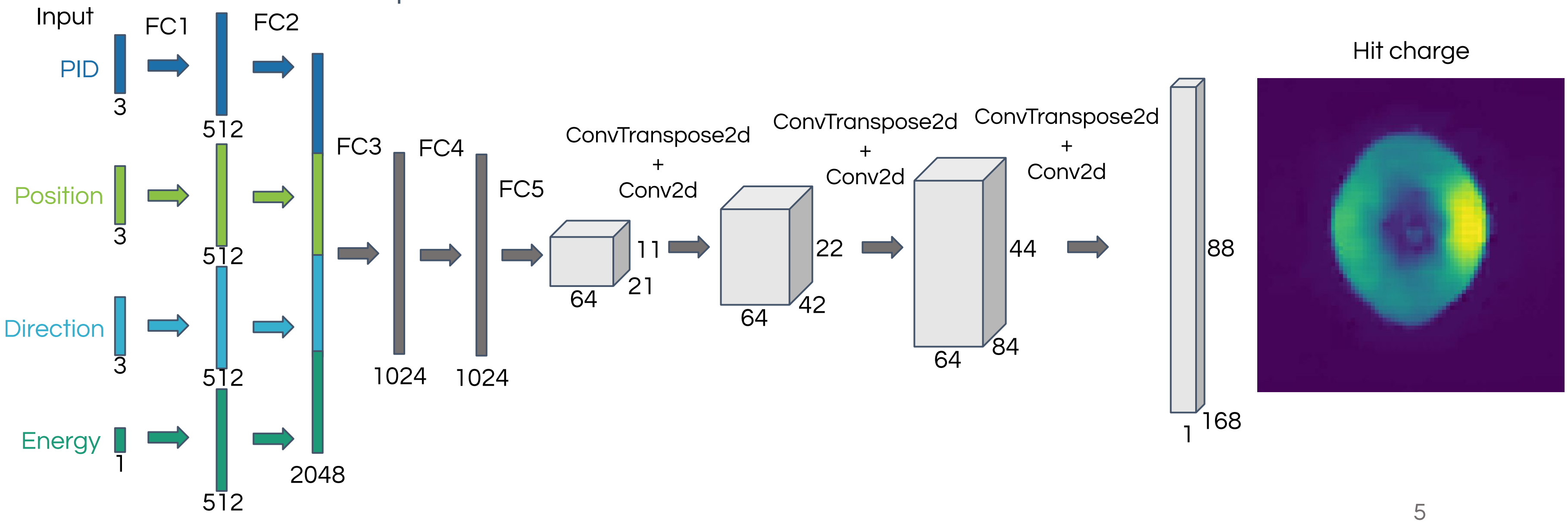
- A CNN is trained to predict the hit charges and times for a given set of track parameters.
- This Cherenkov ring generator is incorporated into a maximum-likelihood estimation framework to form an event reconstruction algorithm.
  - This method is analogous to FiTQun reconstruction: the CNN replaces 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 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.

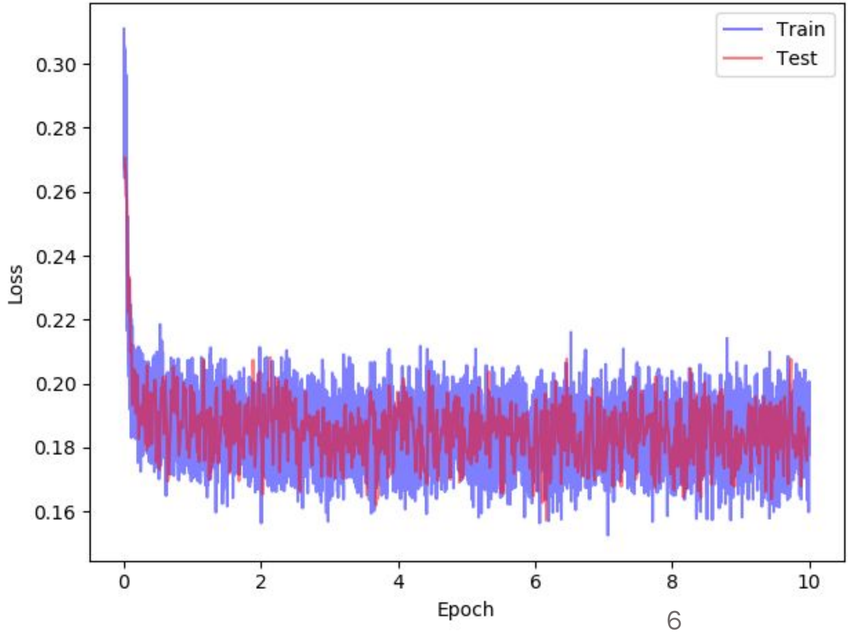
Generating rings with CNNs

- Follow network architecture described in arXiv:1411.5928 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 Terao's examples from the WatChMaL workshop.

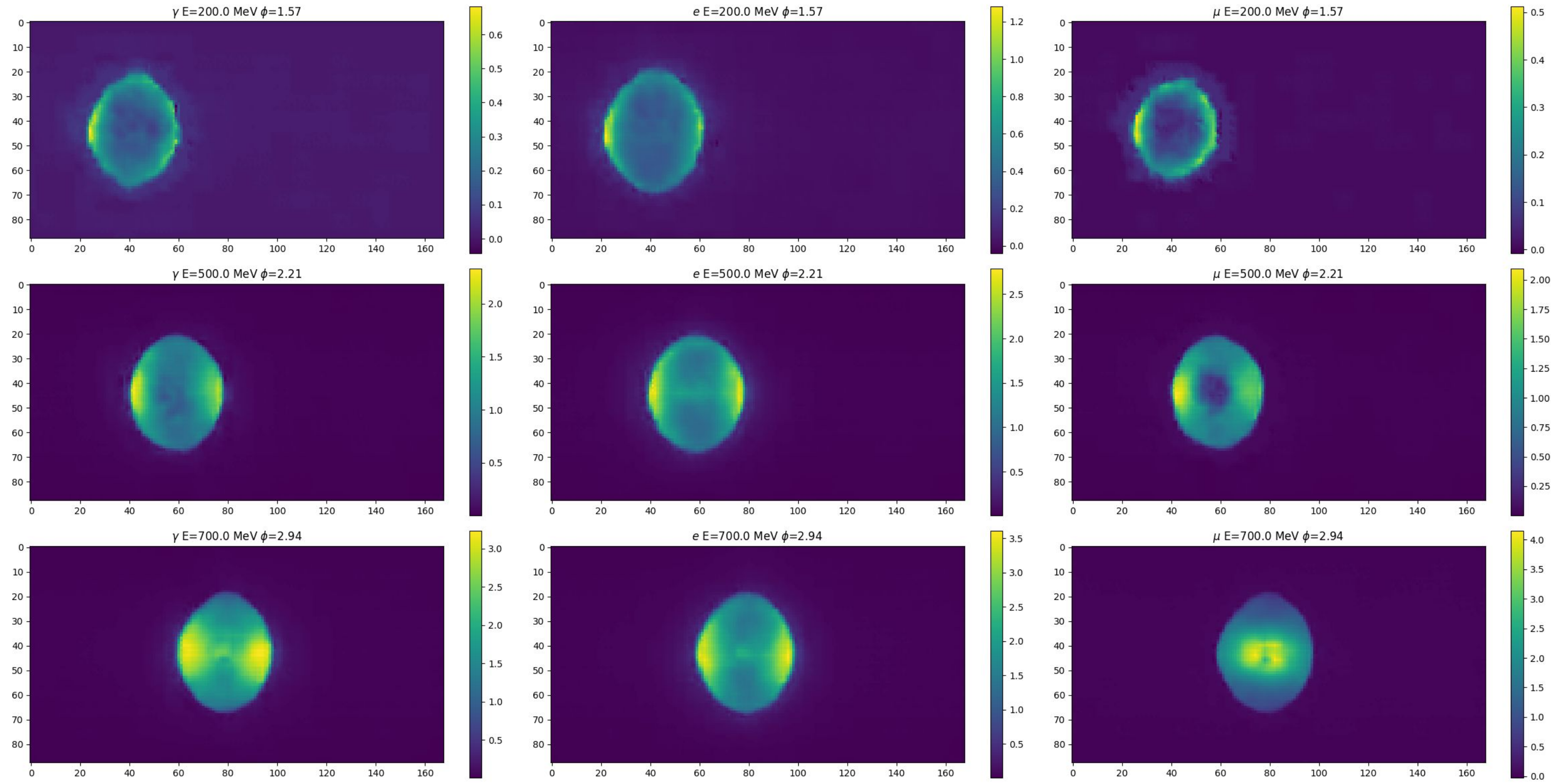


Training

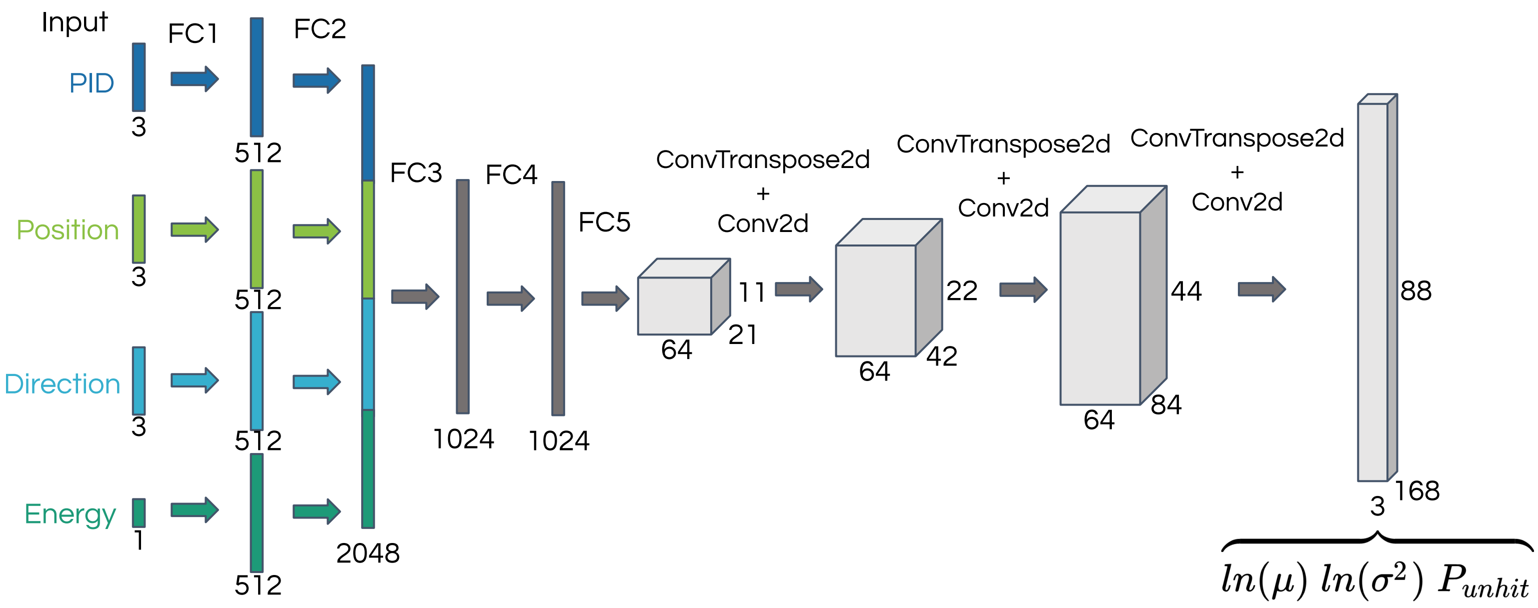
- Used training sample prepared by Nick Prouse for the workshop:
  - 1M of each: electrons, gammas and muons in NuPRISM tank with 88x168 PMTs in the barrel.
  - Batch size: 200, train for 10 epochs (~day using PC w/ GPU)
  - SmoothL1Loss (Huber)
    - $(\text{observed - predicted})^2$ near 0
    - $|\text{observed - predicted}|$ away from 0
  - Adam optimizer (arXiv:1412.6980)



CNN-generated rings

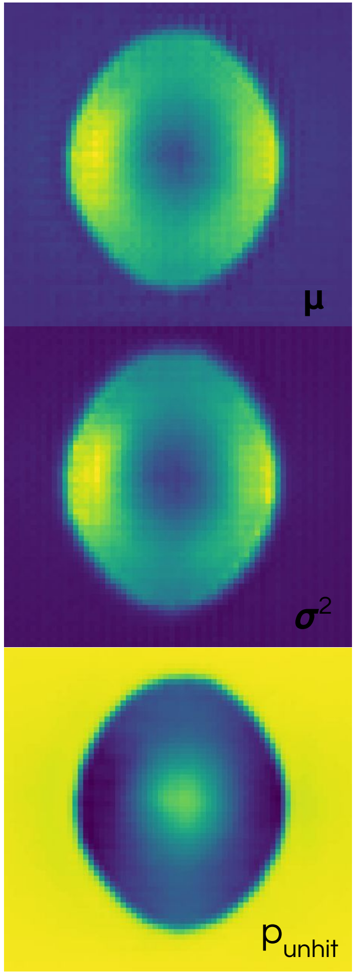


Predicting pdfs



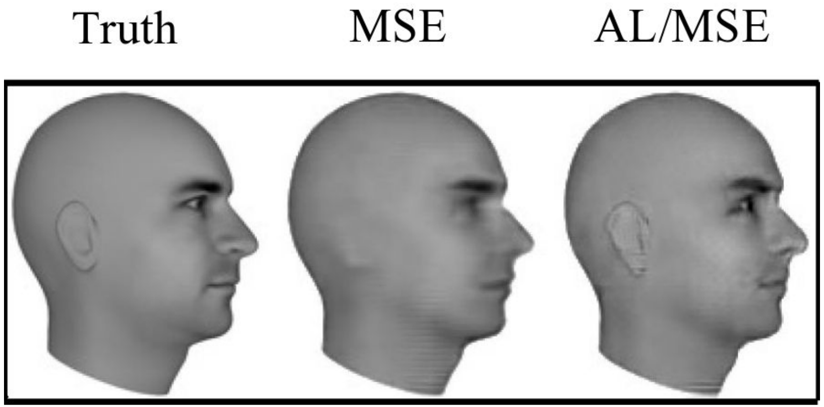
$$\text{Loss} = -ln(\mathcal{L}) = -\sum_{unhit} ln(P_{unhit}) - \sum_{hit} ln(1 - P_{unhit}) - \sum_{hit} \frac{1}{2}\left[ln(2\pi\sigma^2) + \frac{(q_{obs} - \mu)^2}{\sigma^2}\right]$$

- Prediction is a (Gaussian) charge pdf and hit probability for each PMT.
- Basic building block for FiTQun-like MLE reconstruction!



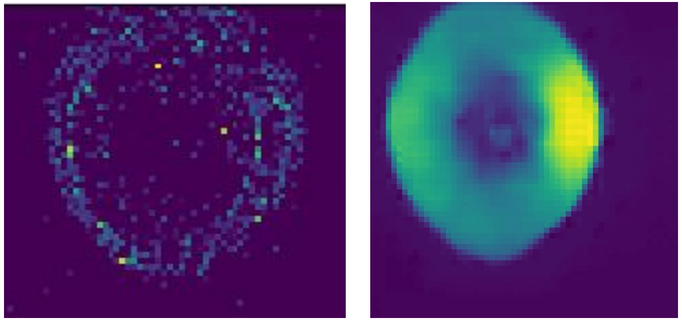
Can we go further?

- What I've shown so far gives one-to-one relation between track parameters and ring predictions.
- But there is event-by-event variation in ring shapes, for example, due to multiple scattering.
- The relation between track parameters and ring prediction is one-to-many.
- Can we use unsupervised machine learning techniques to capture the variations seen in our MC?



arXiv:1511.06380

Monte Carlo MSE Adversarial



?

Conditional GAN example

arXiv:**1605.05396**

Generative Adversarial Text to Image Synthesis

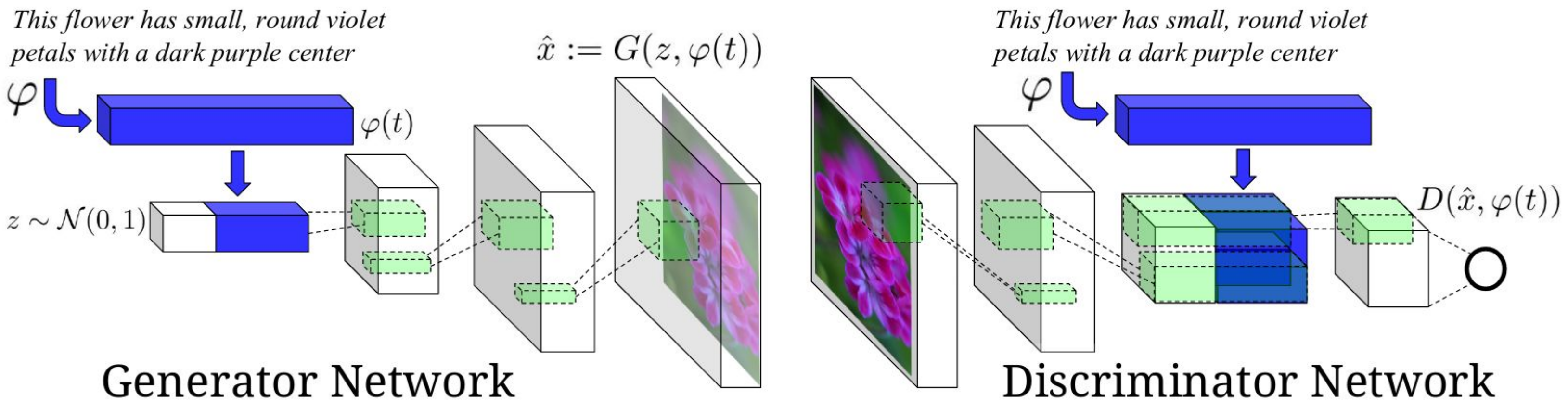
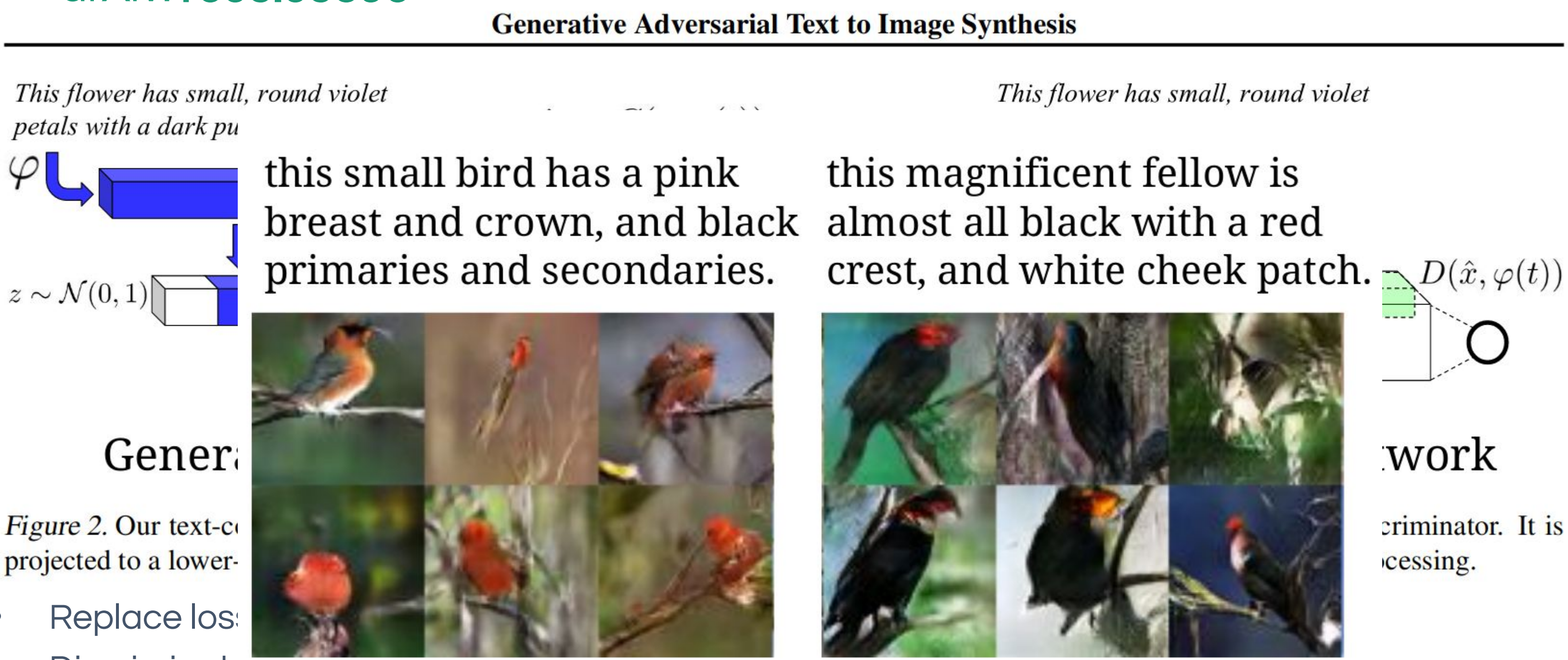


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

- Replace loss function with discriminator neural network.
- Discriminator encourages generator to produce rings that both:
  - Look realistic.
  - Correspond to “MC truth” labels.
- Additional (unsupervised) input parameters on the generator control ring shape variations.

Conditional GAN example

arXiv:**1605.05396**



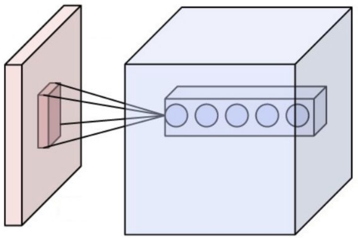
- Replace los
- Discriminator encourages generator to produce rings that both:
  - Look realistic.
  - Correspond to “MC truth” labels.
- Additional (unsupervised) input parameters on the generator control ring shape variations.

Summary and plans

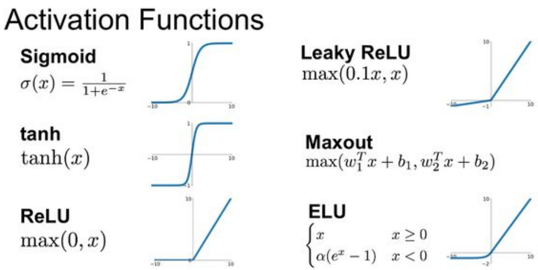
- Initial look into using convolutional neural networks as generative models to be used in water Cherenkov MLE reconstruction shows promising results.
- Ramp up on this work at Stony Brook over the coming weeks/months.
    - Focus will likely be toward T2K and Super-K reconstruction
- Short/medium term tasks:
    - Generate large training Super-K training sample using SKDETSIM.
    - Investigate neural network architecture further:
        - Effects of layer size and number.
        - Look into introducing latent features on the input.
            - Conditional generative adversarial network?
        - Add hit times to the output.
    - Run basic checks of how this would look like as a reconstruction tool.
        - Start with simple likelihood scans, using FiTQun to pre-process events.

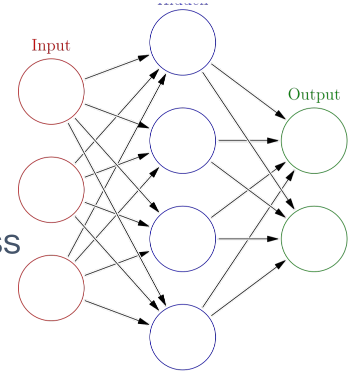
Extra

Neural networks, in < nutshell



- Machine learning technique inspired by biological brains.
- Development exploded in recent years (~2012) mainly due to success in overcoming “trainability” issues.
- Network made of layers of “nodes”.
  - Layers connected to each other:
    - Fully connected.
    - Convolutional (space-aware).
    - ...
  - “Connections” are linear transformations of preceding layers outputs followed by an “activation function”.
    - Non-linear!
- “Training” updates linear transformation parameters in order to minimize a loss function.
  - Choice of loss depends on the problem to be solved.
  - Work backward from the loss using “backpropagation”, i.e. the chain rule.





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](https://arxiv.org/abs/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