A Generative Neural Network for Water Cherenkov Reconstruction

Initial proof-of-concept studies

Neutrino Physics and Machine Learning Workshop

July 24 2020

K. Kumar, L. Mackey, **C. Vilela**, C. Yanagisawa and K. Yang

With support from the WatChMaL group

Current state-of-the-art: FiTQun

Use information from sensors both with and without registered hits in the event

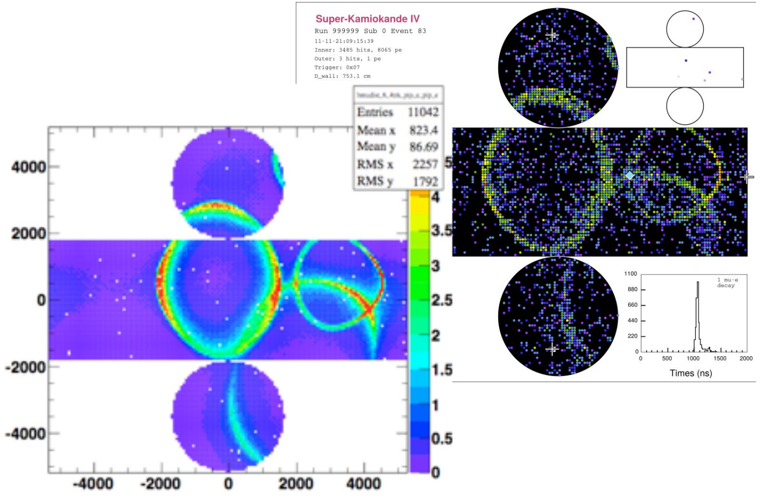
$$\boxed{L(\mathbf{x})} = \boxed{\prod_{i_{unhit}} \mathrm{P}\left(i_{unhit}|\mathbf{x}\right) \prod_{i_{hit}} \mathrm{P}\left(i_{hit}|\mathbf{x}\right)} \boxed{f_q\left(q_i|\mathbf{x}\right)} \boxed{f_t\left(t_i|\mathbf{x}\right)}$$

Build likelihood function for event hypothesis **x**

For hit photosensors:

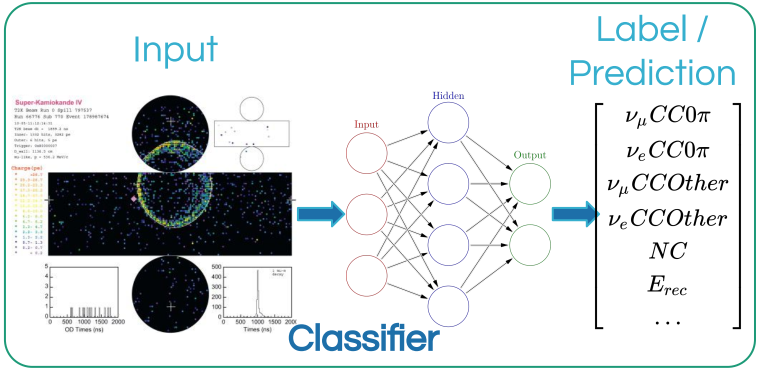
Compare observed charge to prediction and Compare hit time to prediction

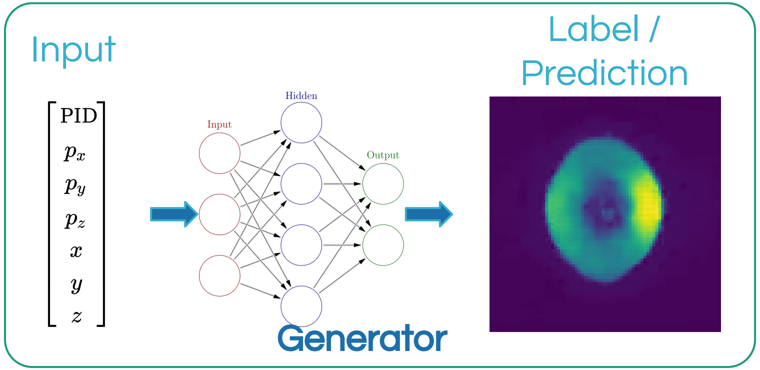
- Make a **prediction** for the **probability distribution functions** of **observables** at each photosensor.
  - Hit/no **hit**; hit **charge** and **time.**
- Maximize the **likelihood** using MINUIT to **reconstruct** events.
  - Use **likelihood ratios** for PID.
- **Combine** several rings for multi-ring hypotheses.



Reconstruction with Generative CNN

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



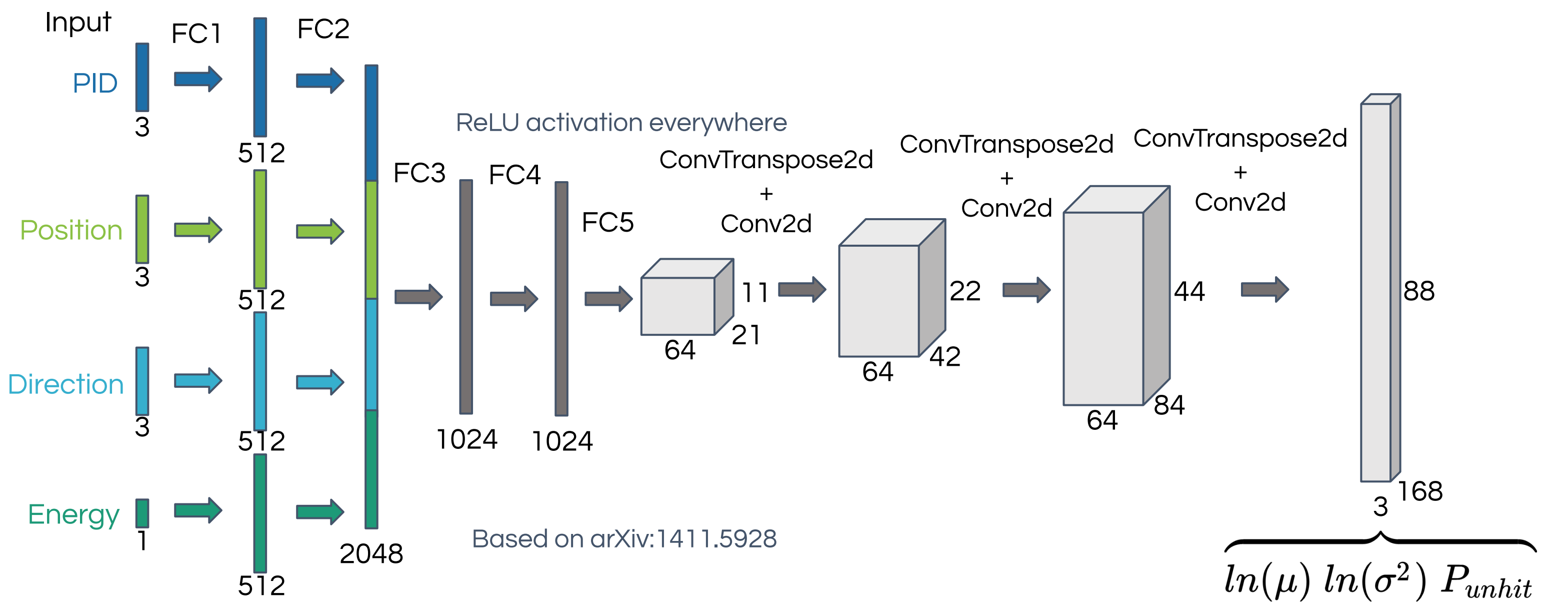


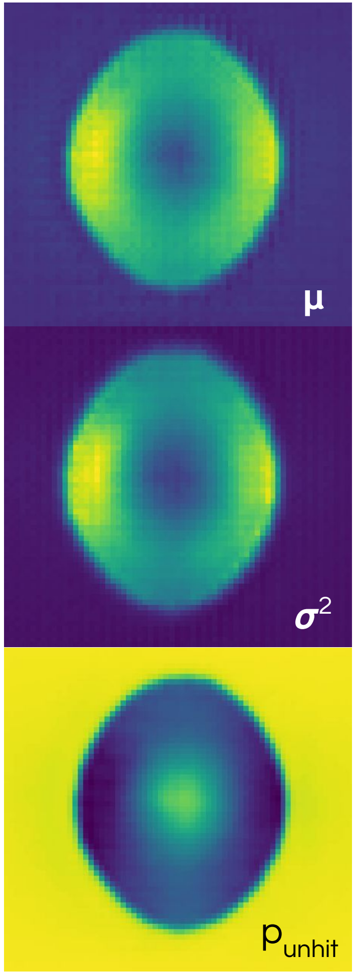
- A CNN is trained to **predict** the hit **charges** and **times** for a given set of track parameters.
- This Cherenkov ring generator can be incorporated into a **maximum-likelihood estimation** framework to form an **event reconstruction algorithm**.
  - This method is **analogous** to **FiTQun** 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, for example in the T2K and Super-Kamiokande experiments.
    - Could be a useful first step in the move towards end-to-end ML reconstruction.

Generating pdfs

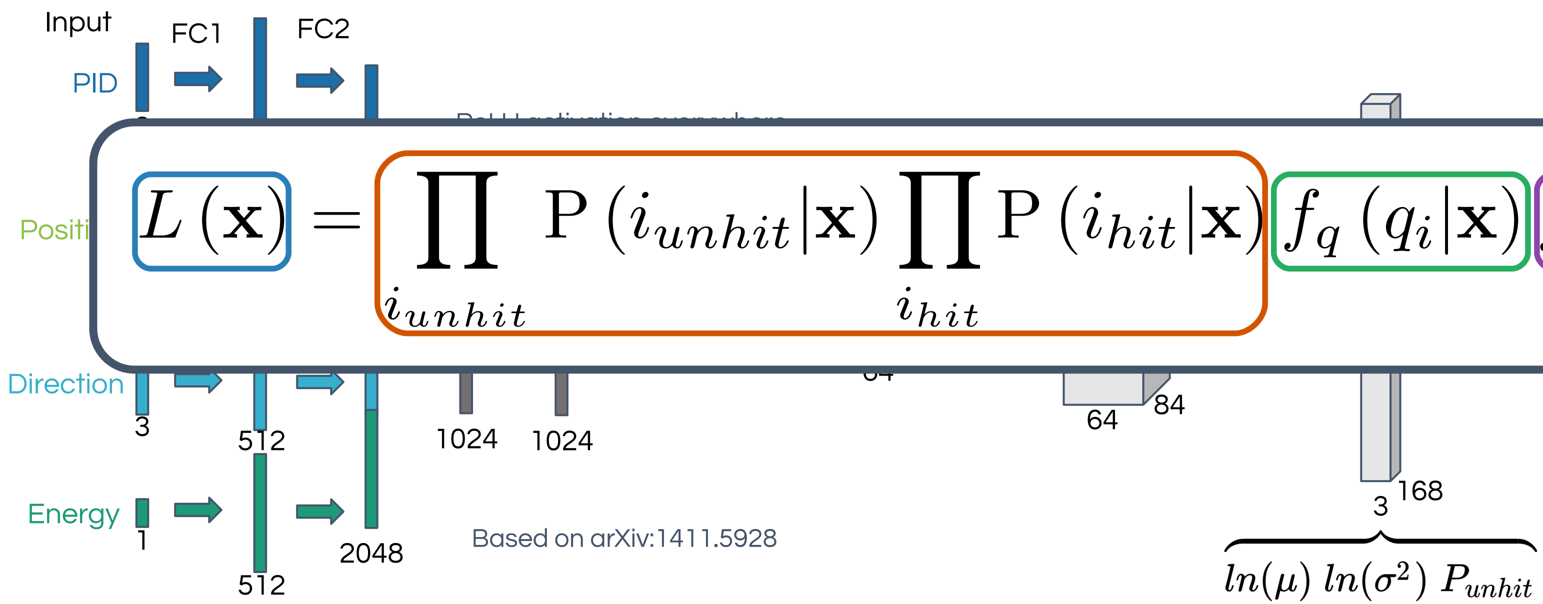




$$\text{Loss} = -ln(\mathcal{L}) = -\sum_{unhit} ln(P_{unhit}) - \sum_{hit} ln(1 - P_{unhit}) - \sum_{hit} \frac{1}{2}\left[ln(2\pi\sigma^2) + \frac{(q_{obs}-\mu)^2}{\sigma^2}\right]$$

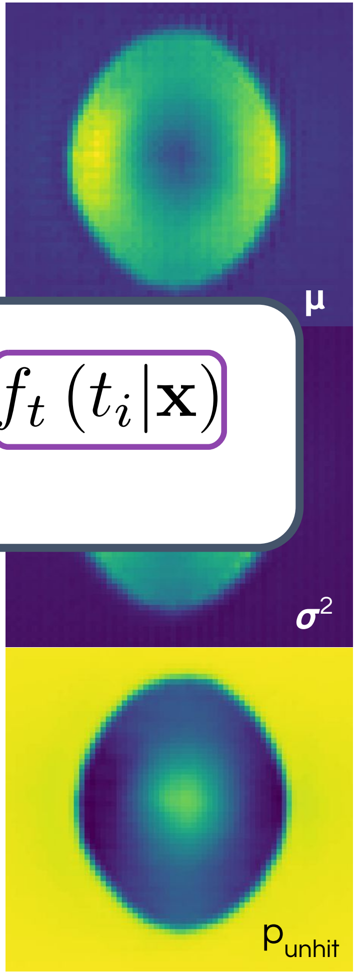
- Output is a (Gaussian) **charge pdf** and **hit probability** for **each PMT**.

Generating pdfs



$$\boxed{L(\mathbf{x})} = \boxed{\prod_{i_{unhit}} \mathrm{P}\left(i_{unhit}|\mathbf{x}\right) \prod_{i_{hit}} \mathrm{P}\left(i_{hit}|\mathbf{x}\right)} \boxed{f_q\left(q_i|\mathbf{x}\right)} \boxed{f_t\left(t_i|\mathbf{x}\right)}$$

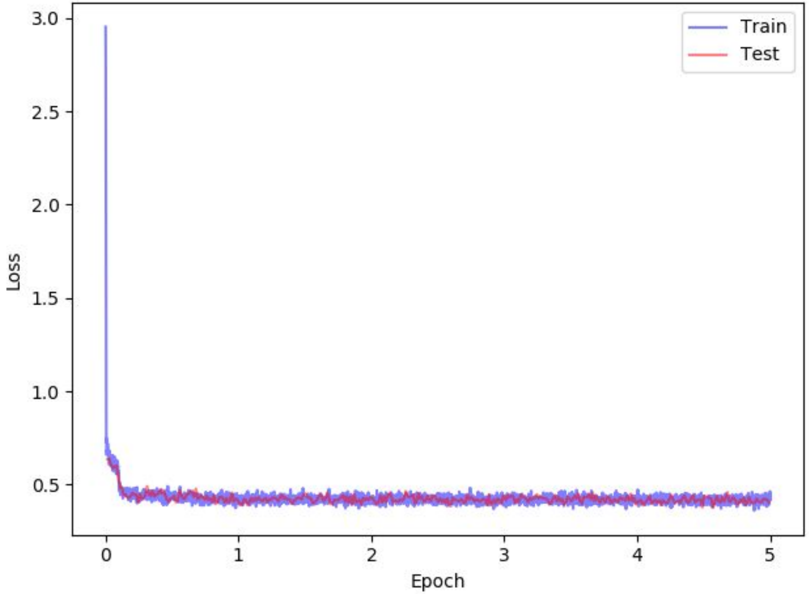
$$\text{Loss} = \boxed{-ln(\mathcal{L})} = \boxed{-\sum_{unhit} ln(P_{unhit}) - \sum_{hit} ln(1 - P_{unhit})} - \boxed{\sum_{hit} \frac{1}{2}\left[ln(2\pi\sigma^2) + \frac{(q_{obs}-\mu)^2}{\sigma^2}\right]}$$



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

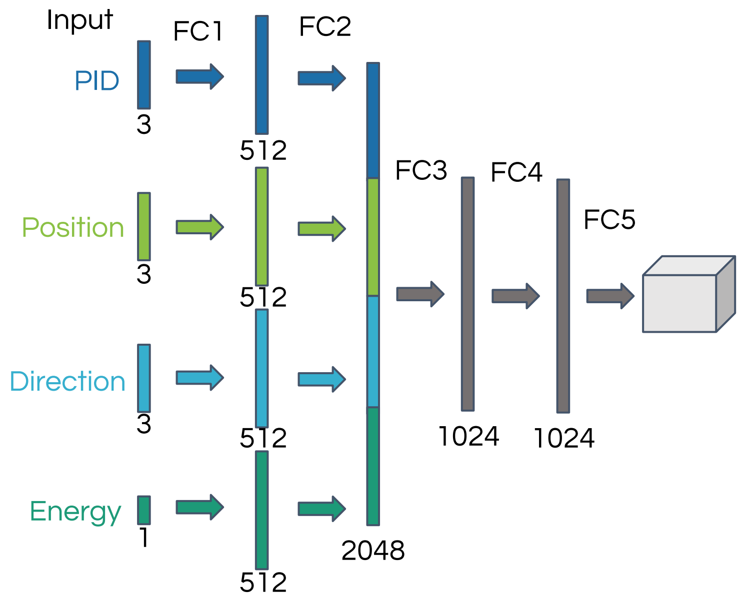
Training

- Training sample produced by WatChMaL group.
  - 1M electrons, muons and gammas.
  - IWCD tank with 88x168 3” PMTs.
  - Uniform, isotropic distribution of events in the tank.
  - Energy up to 2 GeV.
- Model implemented in PyTorch.
  - Trained on GPU with batch size of 200 events.
  - ~10 epochs / day



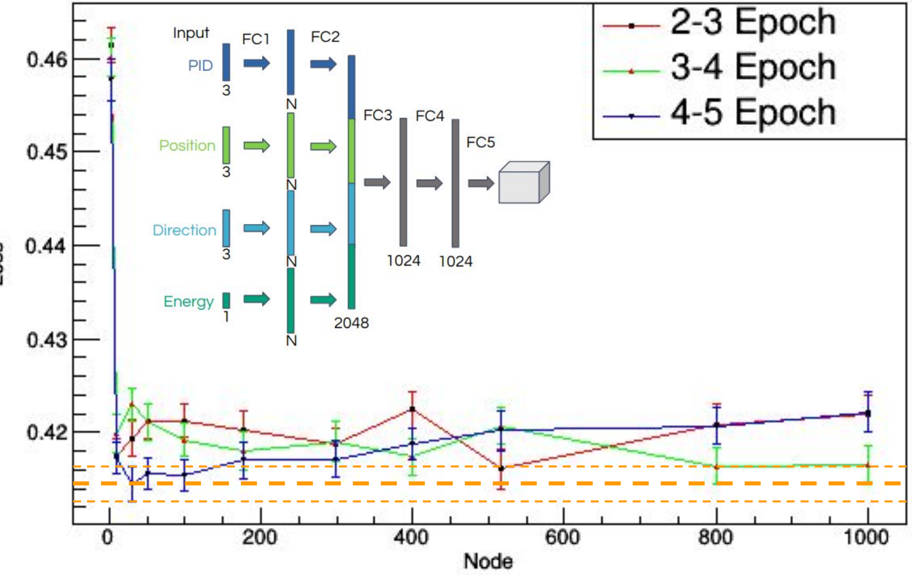
Network optimization studies

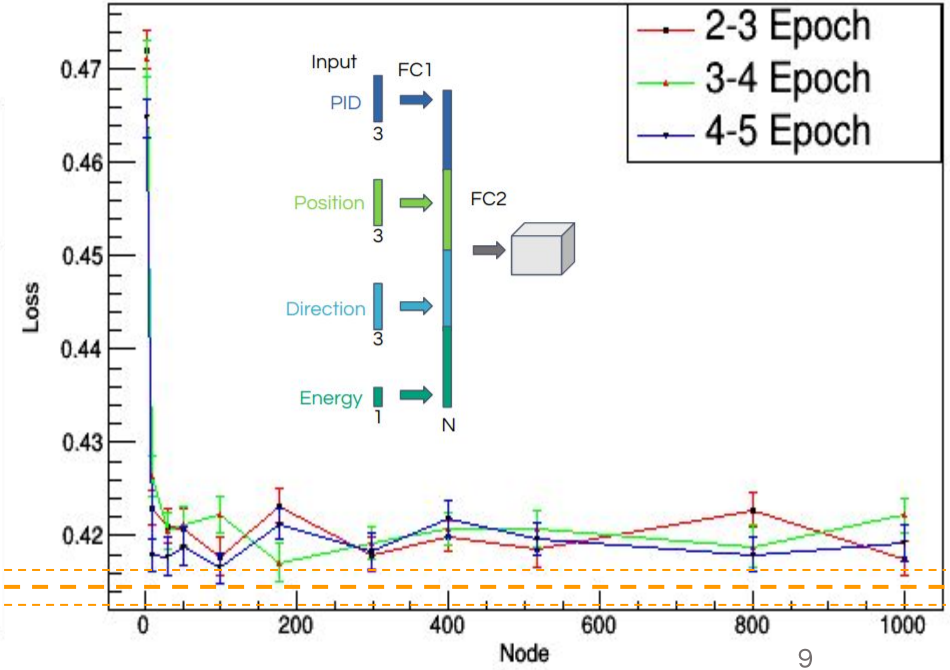
- Current architecture based on paper with quite different objectives.
  - Most likely not optimal for our purposes.
- **Start** by investigating network performance for variations of the fully connected part of the network.
- Change **FC1** from 4 x 512 to **4** x **N**
- **Replace** FC1 to FC4 by **single** fully connected layer with **N** nodes.
  - DCGAN paper (arXiv:1511.06434) recommends no FC layers on generator.



Network optimization studies

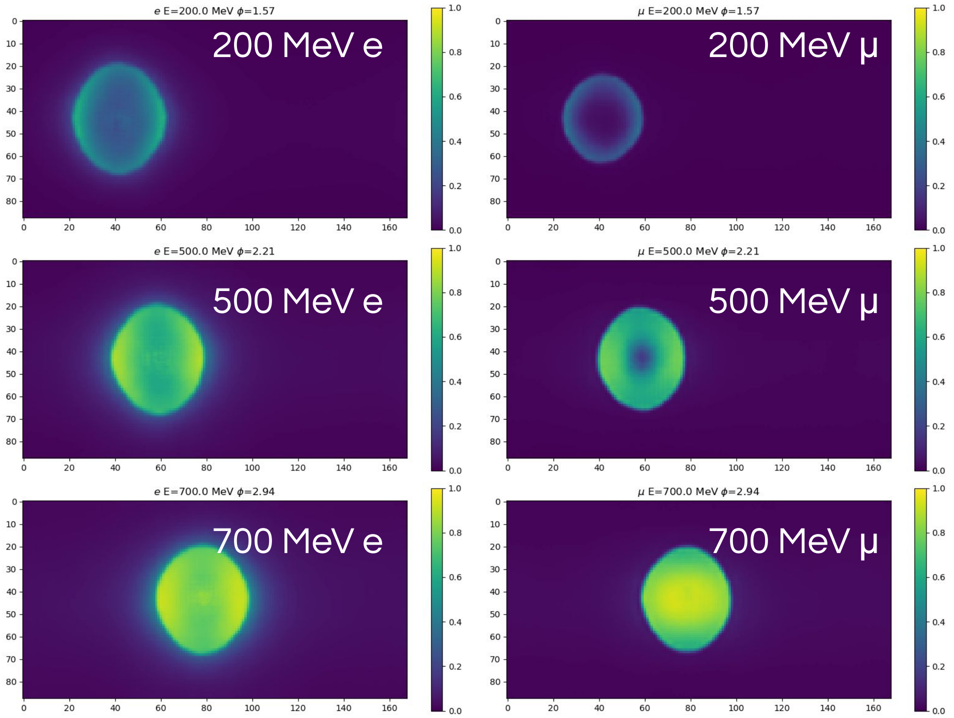
- Best loss achieved with 4 x **30** nodes and keeping the 5 FC layers.
- Performance only two FC layers is comparable despite significantly fewer parameters.



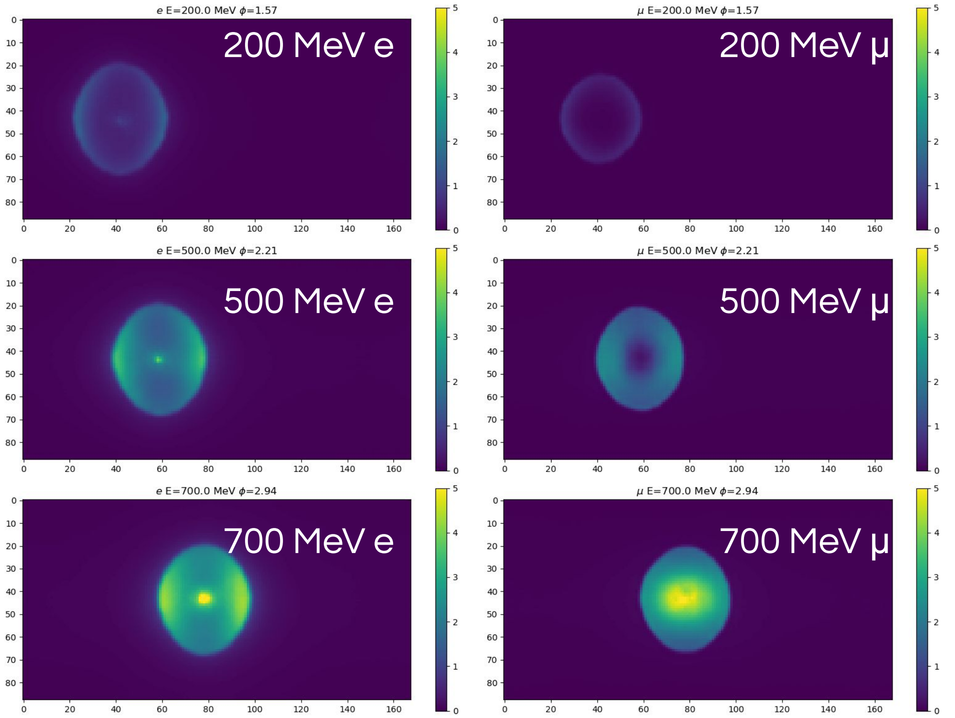


Generated ring examples

Hit probability

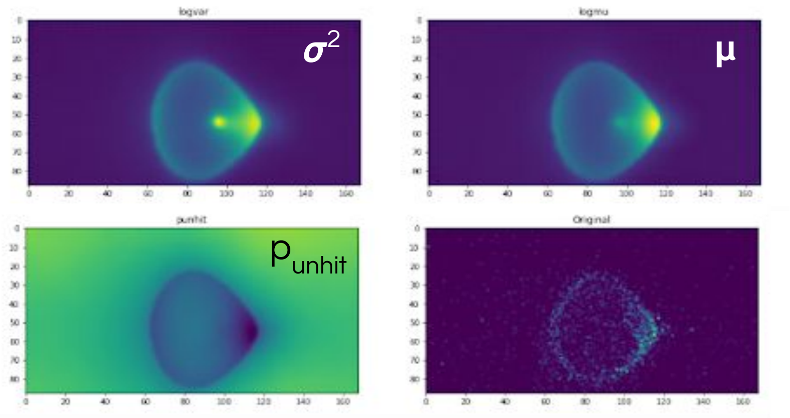


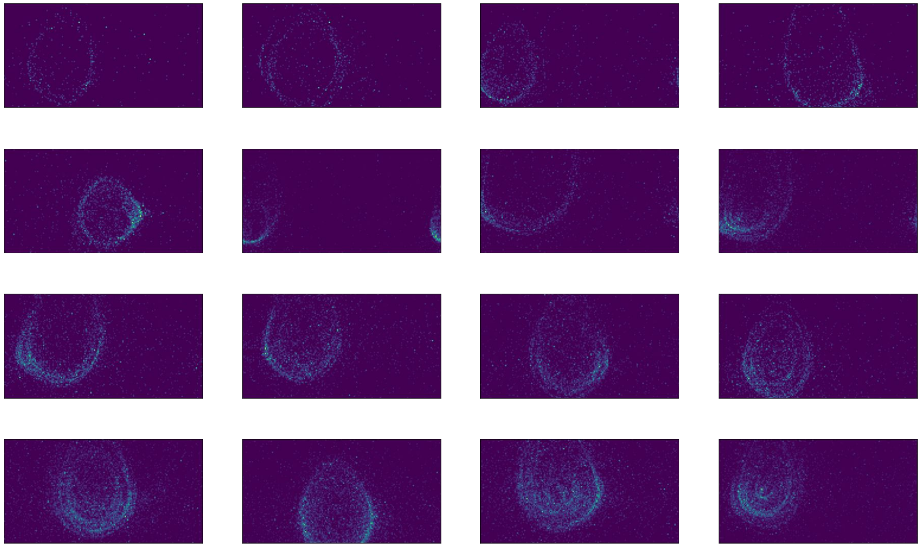
Hit probability x Mean charge



Likelihood scans

- For maximum-likelihood reconstruction need loss function to be **smooth** and **unbiased**.
- Investigate by **evaluating** loss function with position, direction and PID fixed at their true values and scanning the loss as a function of **energy**.

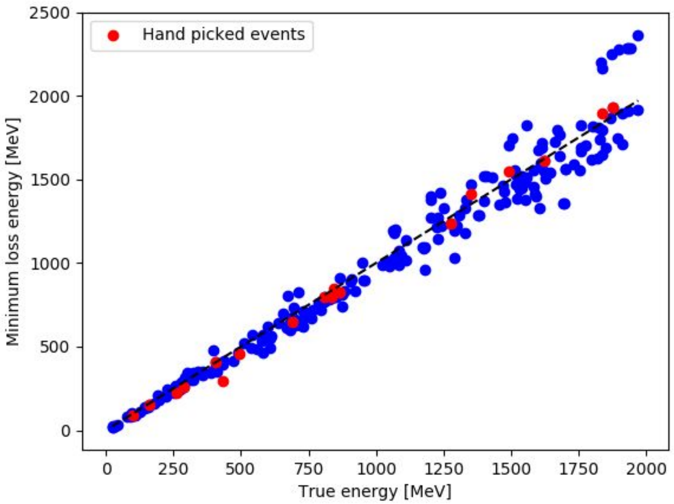


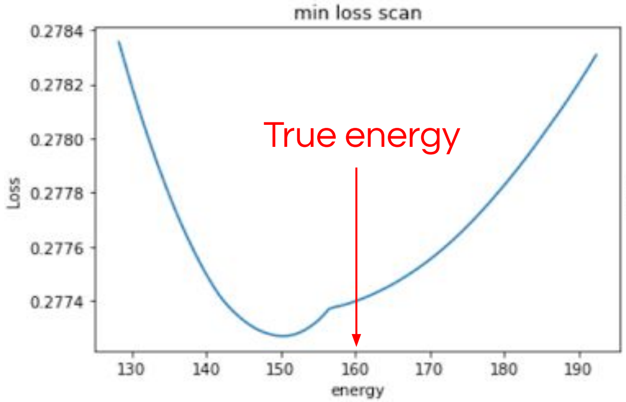


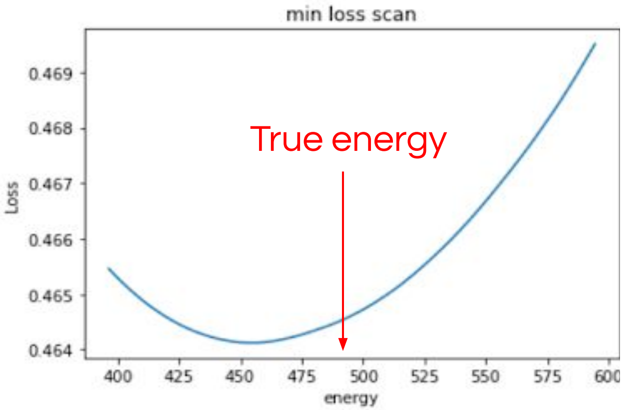
- Choose a set of 16 “easy” events by hand.
  - Ensure ring is mostly in the barrel.
  - Position + direction criterion to select 200 more events

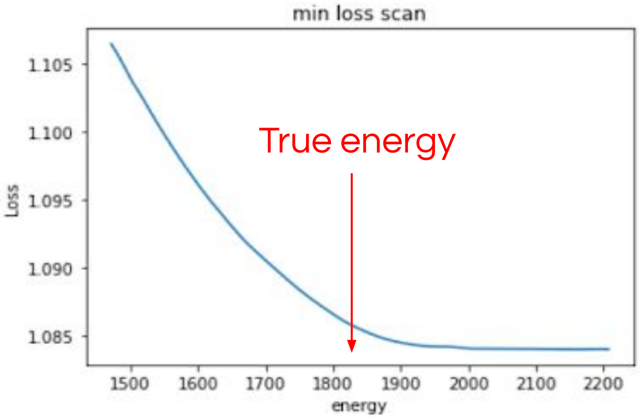
Likelihood scans

- Likelihood surface is generally smooth.
- Minimum tends to be near true value, especially for “hand picked” events.
- Encouraging first results!



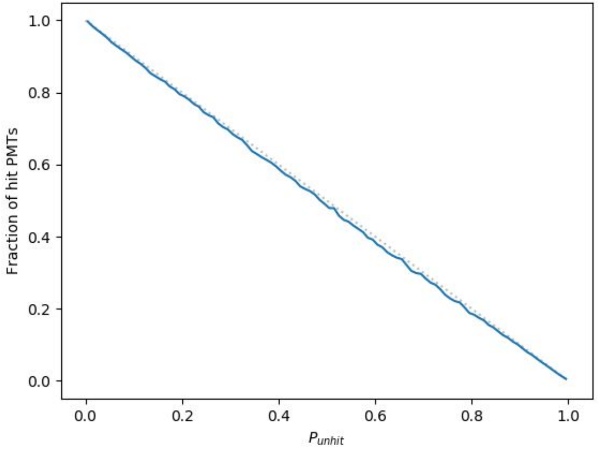


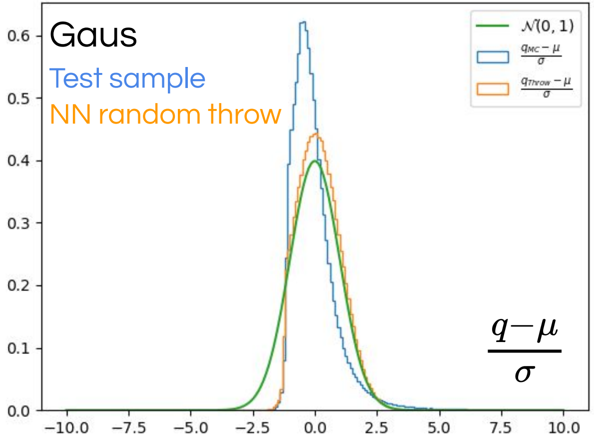


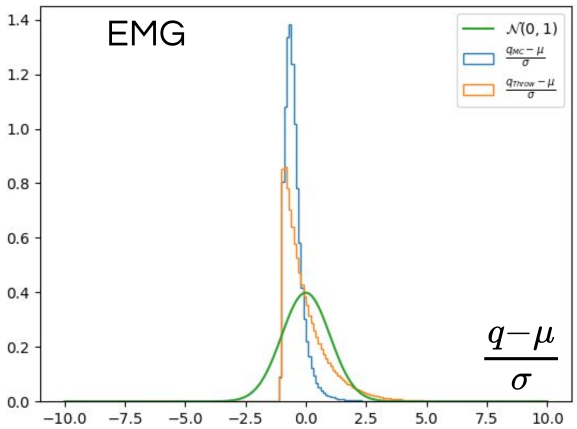


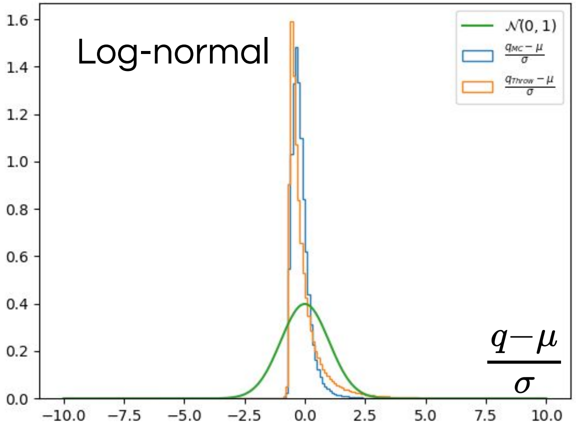
Loss function studies

- Hit probability component of the loss function works well.
- However, charge distribution is not gaussian.
  - Might get some improvement by using a more “taily” distribution.
  - Currently investigating mixture of Gaussians.



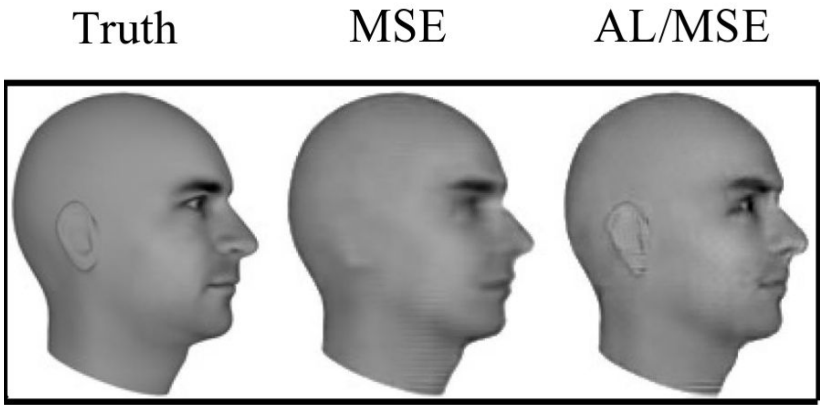




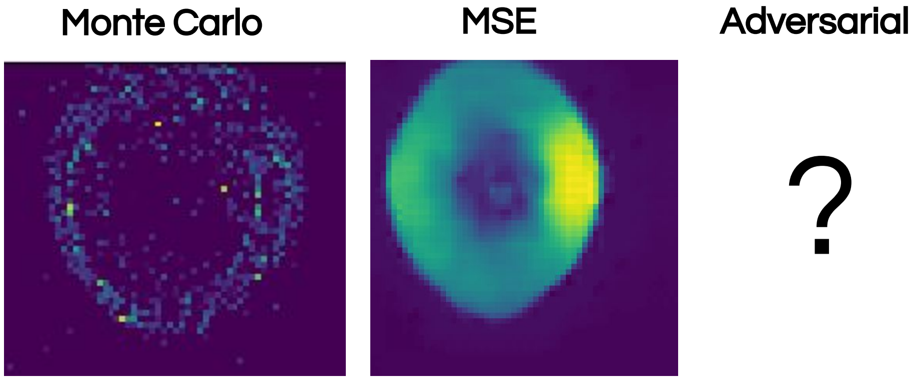


Adversarial training

- 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.
- Plan to investigate training as a conditional adversarial generative network.



arXiv:1511.06380



Summary

- We are exploring an **alternative** use of CNNs in water Cherenkov reconstruction by **generating pdfs** at each PMT for **maximum-likelihood estimation.**
- Initial studies are **encouraging** and demonstrate **smooth** and **unbiased** likelihoods can be achieved.
- On-going studies focus on **improvement** of the network **architecture** and loss function.
- **Near**-term goals include investigating **adversarial** training approaches.
- **Mid**-term goal is to incorporate network in **gradient-descent algorithm.**

Supplementary slides

Generating images with CNNs

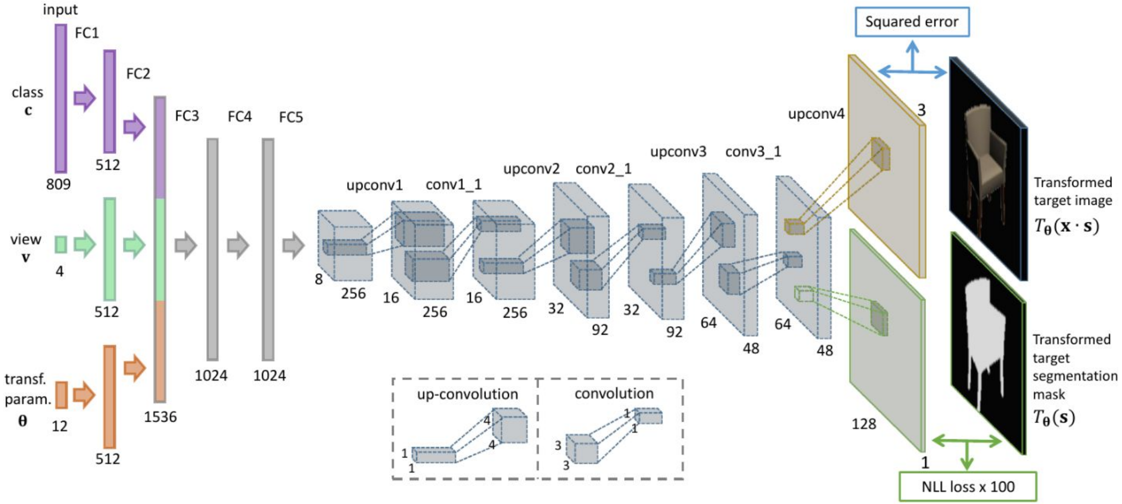
- First iteration of a Cherenkov ring generator neural network follows approach in:

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



Conditional GAN example

arXiv:**1605.05396**

Generative Adversarial Text to Image Synthesis

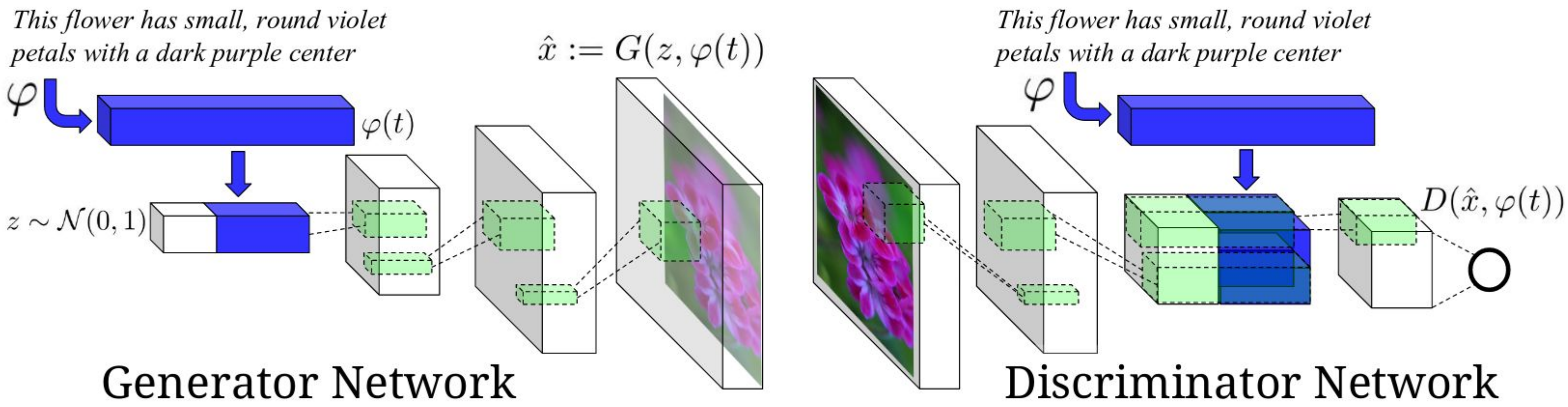
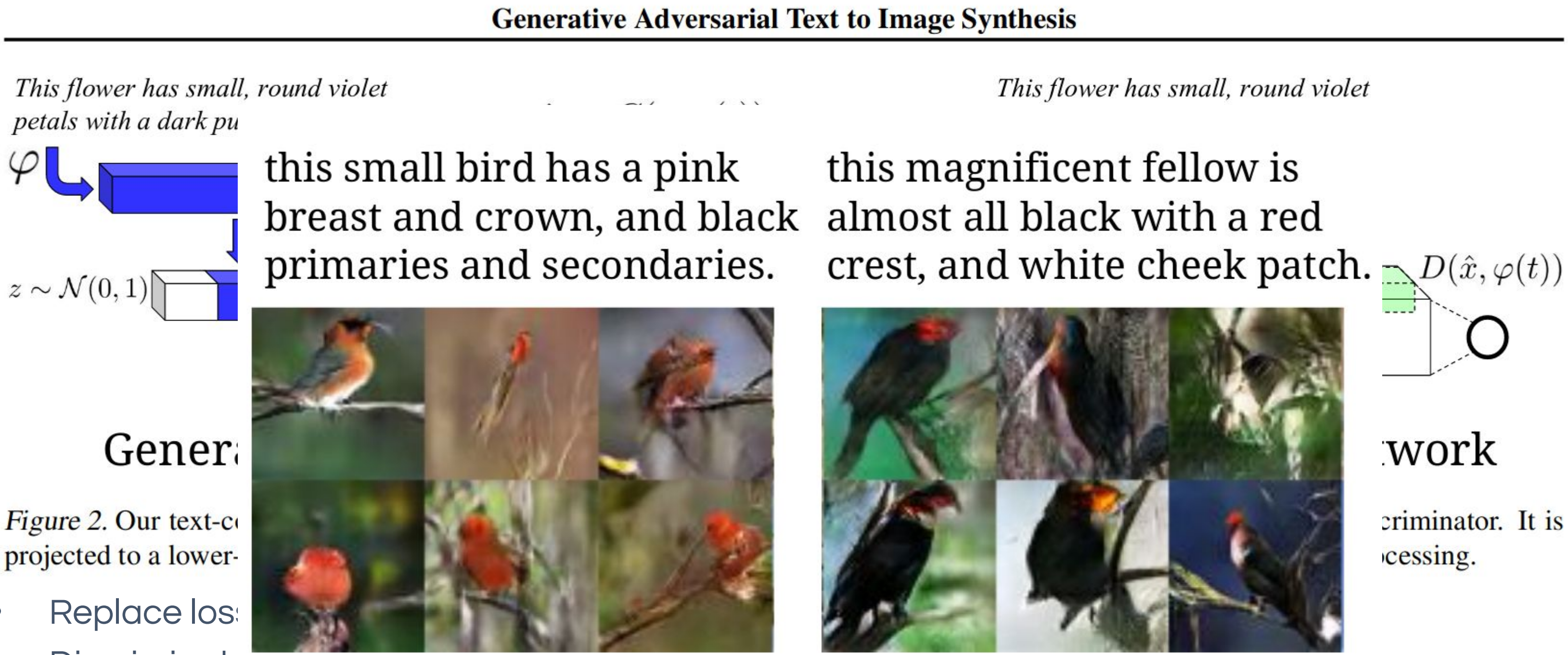


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

- 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.