Emulating Neutrino Oscillation Experiments with a Generative Adversarial Network

Imperial College London

Matthew Legg and Sachin Ranjan | Supervisor: Dr Abbey Waldron | High Energy Physics Group | Imperial College London

Project Aim

To use a Generative Adversarial Network (GAN) to emulate simulation data of observed electron neutrinos from the DUNE experiment.

Motivation: Neutrino Oscillations

- The Universe is matter dominated. As matter and antimatter are created and destroyed together this implies the physics do not apply equally.
- The Deep Underground Neutrino Experiment (DUNE) is a long baseline experiment whose primary aim is to investigate the origin of matter by probing the properties neutrinos – a neutral, small nonzero mass particle with 3 flavours (e, μ, τ) [1].
- Neutrino oscillation is the natural process of neutrinos changing their flavour. Neutrinos and antineutrinos do not oscillate at the same rate which is quantified by the charge-parity violating phase (δ_{CP}) [1].
- DUNE proposes to compare oscillations of muon neutrinos to muon antineutrinos to find a value of δ_{CP} as well as other oscillation parameters (θ_{13}, θ_{12}, θ_{23}) [1].

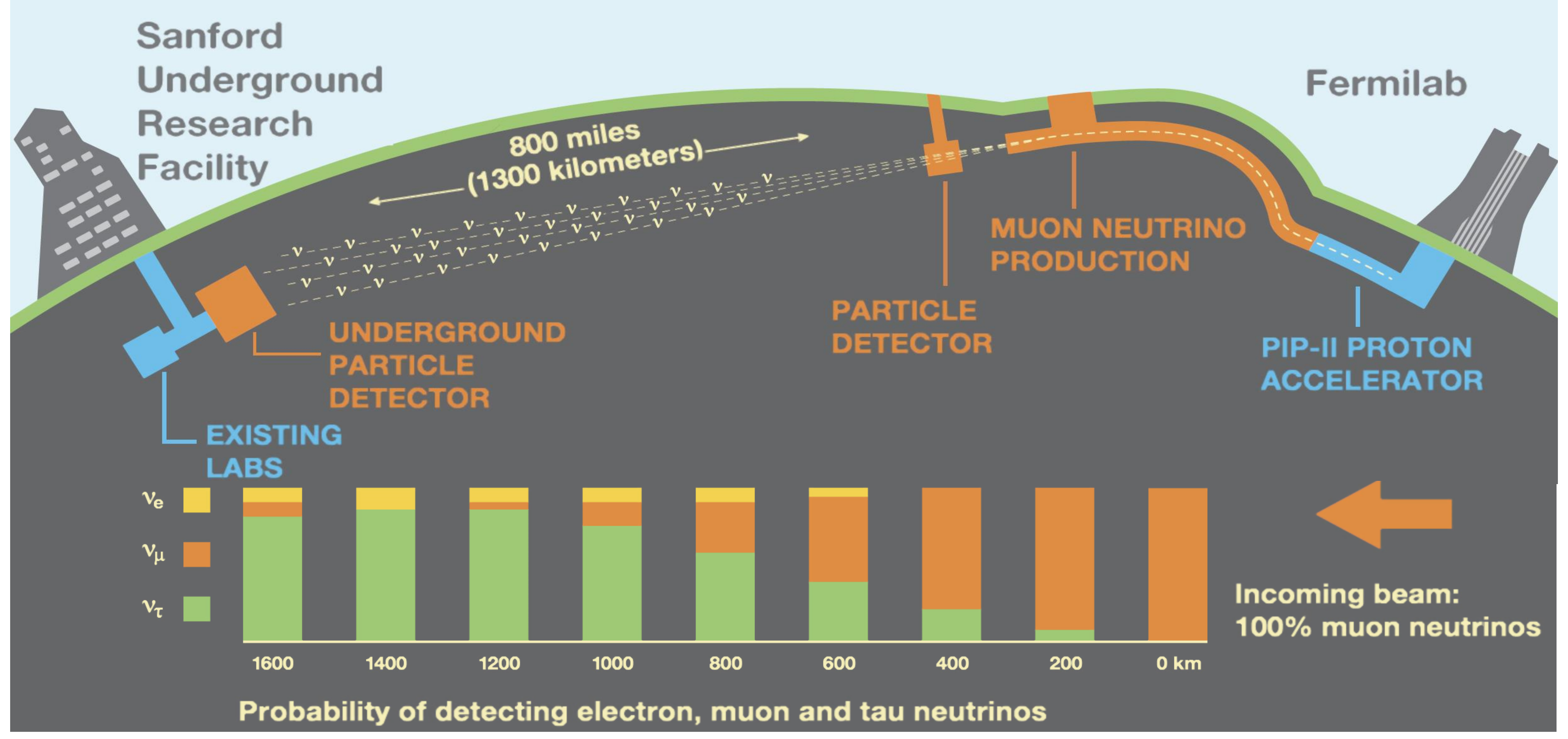


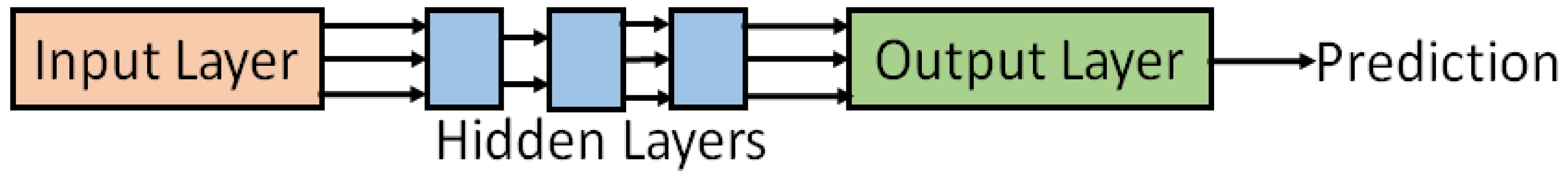
Figure 1: DUNE setup from the Fermilab in Illinois to SURF in South Dakota [2]

Motivation: Machine Learning

- DUNE simulations are comprised of multiple stages with complex systematic errors to be considered at each stage.
- Machine learning allows one to input the oscillation parameters and systematics uncertainties, and output results in a single computationally inexpensive process.
- The Generator Neural Network is used to generate realistic histograms of the appearance of electron neutrinos vs energy from the parameters.
- This is paired with a Discriminator Network which can check for a range of physically impossible or incorrect outputs.
- Training these two networks against each other forms a Generative Adversarial Network (GAN).

Deep Neural Networks

Figure 2: Outline of a basic Deep Neural Network



- **Input Layer**: Contains the values of inputted data.
- **Hidden Layers**: Each layer performs a multiplication of a weight vector to the previous layer under various rules. Dense, Convolutional and Deconvolutional layers have been used. The result of which is fed through a nonlinear function e.g. tanh, ReLu or Leaky ReLu.
- **Output Layer**: Predicts the values of certain events or the probability of a certain outcome.

The weights are initialised randomly and are then optimised in training using the Adam Optimiser technique to reduce the Loss Function – a function describing the performance of the network.

Our Networks

- The Generator inputs are the experimental parameters, which are put through 1D deconvolutional layers and output histogram bin values.
- The Discriminator inputs are the histogram bin values. It puts it through 1D Convolution layers which were pre-trained to predict the parameters.
- It then inputs the initial input parameters for comparison and is sent through two dense layers. The output is a probability that these histograms are real simulation data.
- Loss is calculated via binary cross-entropy, giving a measure of how accurate this probability was to their actual True/False labels.

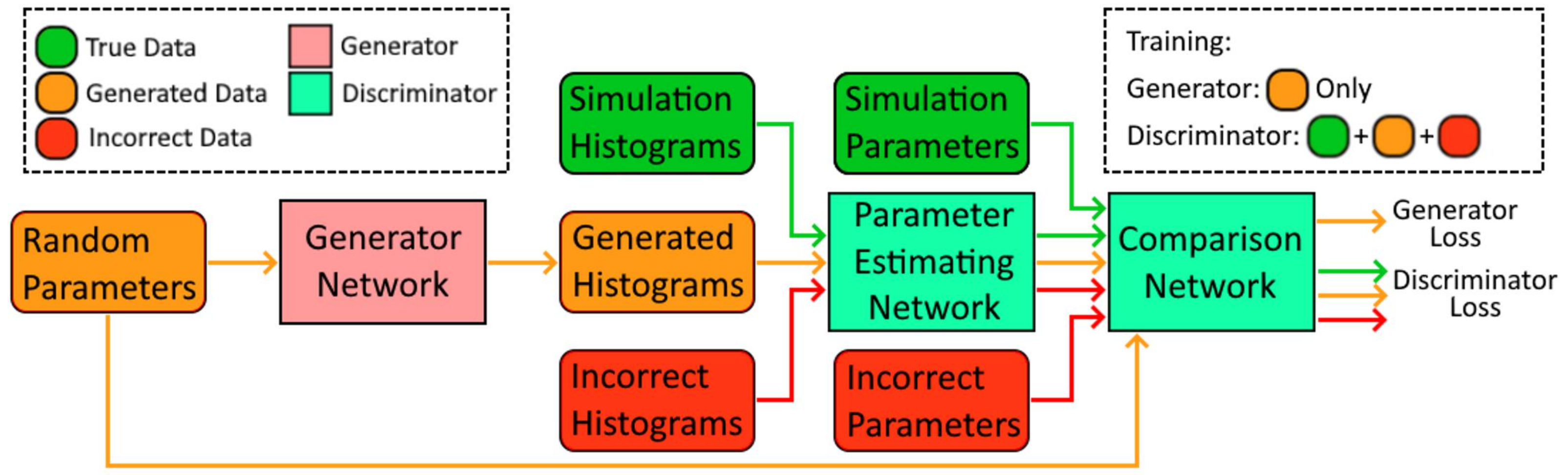


Figure 3: A diagram of our GAN and the inputs it is trained on

Pre-training

- Generator was pretrained to match the generated histograms with the simulated histograms using the simulation parameters.
- The parameter estimator was trained to give the correct parameters from the simulated histograms.
- The output layer of the trained parameter estimator was removed, and the rest was joined to the untrained comparison network to form the Discriminator.
- The Discriminator was pretrained to recognise the simulated data as True and random noise data as False.

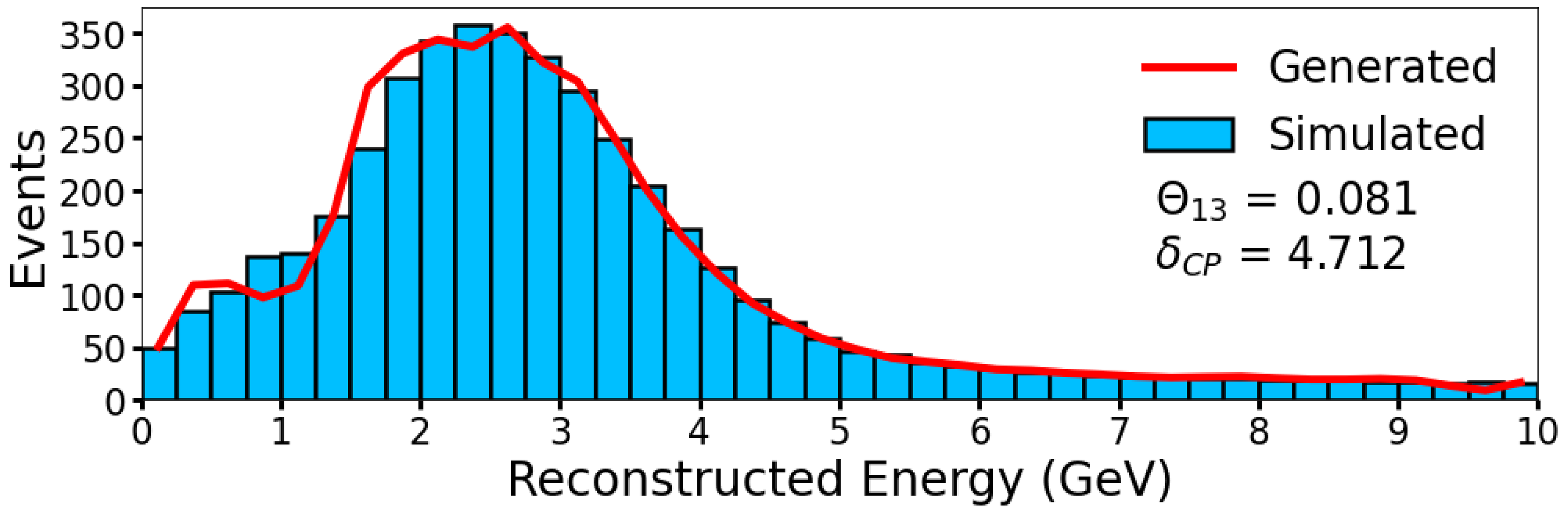
Training

1. GAN Training Part 1: The Discriminator was trained to recognise the simulated data as true and generated data as false.
2. GAN Training Part 2: The Generator was trained to generate data that the Discriminator will label as True.

These adversarial training processes were iterated over many times.

Results

Figure 4: Generated data compared to the simulated data with the same parameters



- Generated histograms match well particularly in the tail for the 2 input parameters shown (θ_{13}, δ_{CP}).
- Conducting further training to improve results with loss < $O(10^{-4})$ and increasing to more parameters.

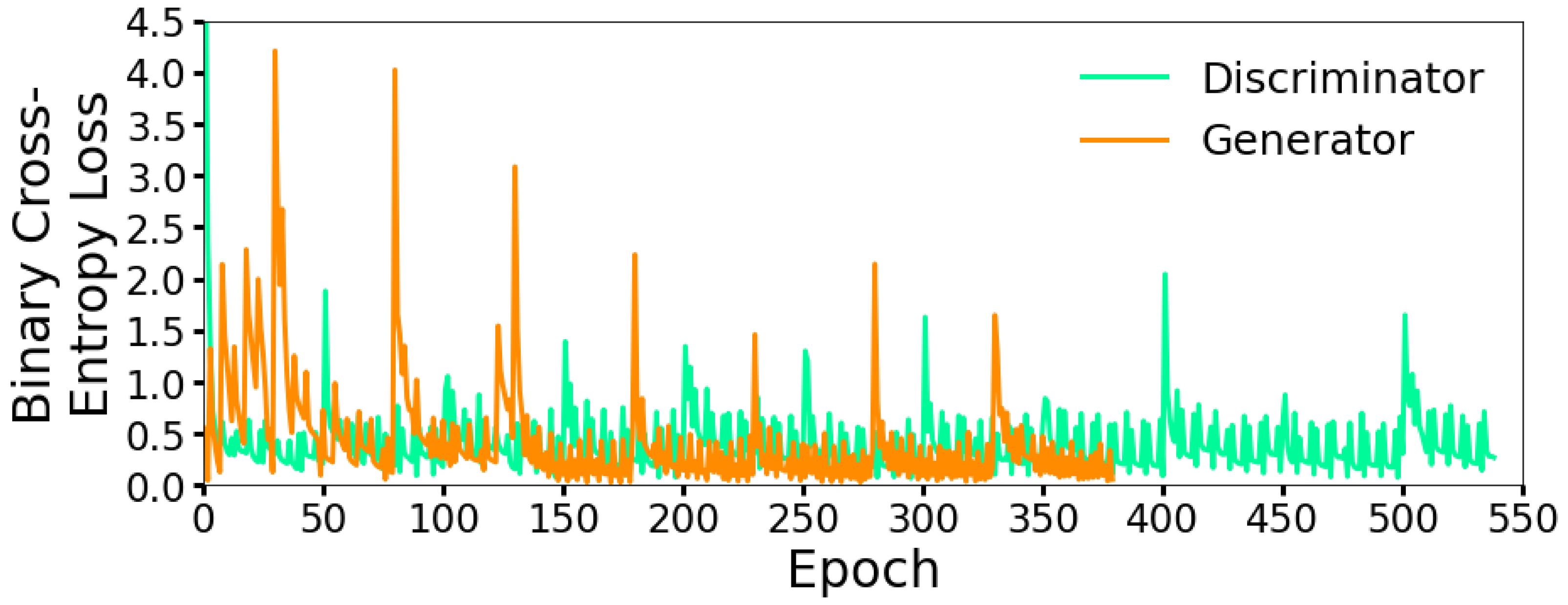


Figure 5: Graph showing the Loss Function of each network after each epoch of training in the GAN training cycle

Future Implications

- Fast production of data without the need for a simulation.
- Use the Generator to create a multidimensional likelihood surface as Neural Networks are differentiable.
- Likelihood surface can analyse DUNE data and predict the values of parameters, systematic uncertainties, and their confidence intervals.

References

[1] The DUNE Colab., "Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) Conceptual Design Report Volume 2: The Physics Program for DUNE at LBNF", DUNE, 2016.

[2] *Institute of Physics of the Czech academy of Science, DUNE Experiment. [image]: <https://www.fzu.cz/en/research/research-topics/deep-underground-neutrino-experiment-dune> [Accessed 3/3/2021].*