Machine learning and high-performance computing for neutrino oscillations

Saúl Alonso-Monsalve
ETH Zurich

Fall Seminar Series
National HPC Competence Centre
The Cyprus Institute
18 October 2022

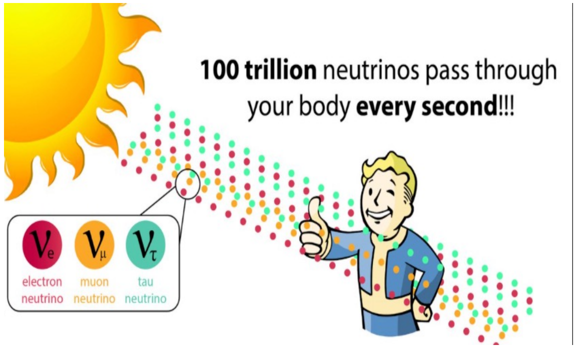
Overview

- Introduction to neutrinos.
- Deep learning in neutrino experiments:
  - Deep Underground Neutrino Experiment (DUNE).
  - Tokai to Kamioka (T2K).
- Study of deep-learning workloads.
- Summary.

Overview

- **Introduction to neutrinos.**
- Deep learning in neutrino experiments:
  - Deep Underground Neutrino Experiment (DUNE).
  - Tokai to Kamioka (T2K).
- Study of deep-learning workloads.
- Summary.

Neutrinos



- **Neutrinos** are **light subatomic particles**.
  - They are present since the **origin of the Universe**.
  - They are the **second most abundant particle in the Universe**, after photons.
- There are **three** types of neutrinos (and their corresponding antineutrinos), known as **flavours**.
  - **Electron neutrino (ν_e)**, **muon neutrino (ν_μ)**, and **tau neutrino (ν_τ)**.
  - They differ in the way they interact with other particles.
- **Neutrinos oscillate***, meaning that hey can change their flavour.
  - A neutrino generated with a specific flavour can later be measured to have a different flavour.

2015 Nobel Prize in Physics. Takaaki Kajita, Art McDonald: "For the discovery of* ***neutrino oscillations**, which shows that neutrinos have mass."*

Mystery of neutrinos

- Neutrinos are **elementary particles** belonging to the **Standard Model (SM) of particle physics**.

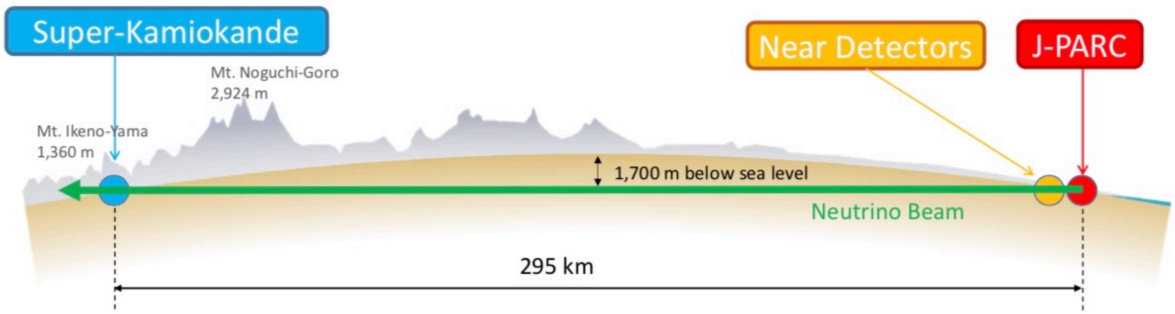
- The SM is one of the **most successful theories in physics**.
  - It can be used to explain most of the experimental observations.
  - However, it **cannot explain the phenomenon of neutrino oscillations**.

- Neutrinos can be the **key to discover physics beyond the SM**.
  - Current measurements do not explain why the Universe is matter-dominated.
  - The difference in how matter and antimatter particles interact is known as *CP*-violation.
  - It is possible that neutrinos and antineutrinos oscillate differently, and a **discovery of *CP*-violation in neutrino oscillations could be the catalyst to understanding the matter-antimatter asymmetry of the Universe**.

Neutrino oscillation experiments

- **Long-baseline neutrino oscillation experiments** use two detectors to characterise a beam of (anti)neutrinos.
  - A **near detector**, located a few hundred metres away from the target that determines the original beam composition.
  - A **far detector**, located several hundred kilometres away, that measures neutrinos flavour oscillations.

- Example: the T2K experiment in Japan.



Source: https://www.t2k-experiment.org/t2k/

Some open challenges in neutrino physics

- **Maximise the *CP*-violation sensitivity**: efficiently identify the signal interactions and have a powerful rejection of background events.
  - **Precise algorithms** are needed to achieve very high signal efficiency and background rejection for event classification.

- **Reconstruct particle tracks** that are detectable in fine-grained detectors.
  - It is necessary to develop **mechanisms to fit and categorise** the different 3D hits, so most of the ambiguities can be identified and rejected.

- **Reduce the gap between simulated and experimental data**.
  - The detector design and optimisation are always guided by accurate and computationally-expensive simulations of the detector behaviour.
  - **Ensuring the robustness of algorithms** against systematic uncertainties becomes a fundamental requirement.

Overview

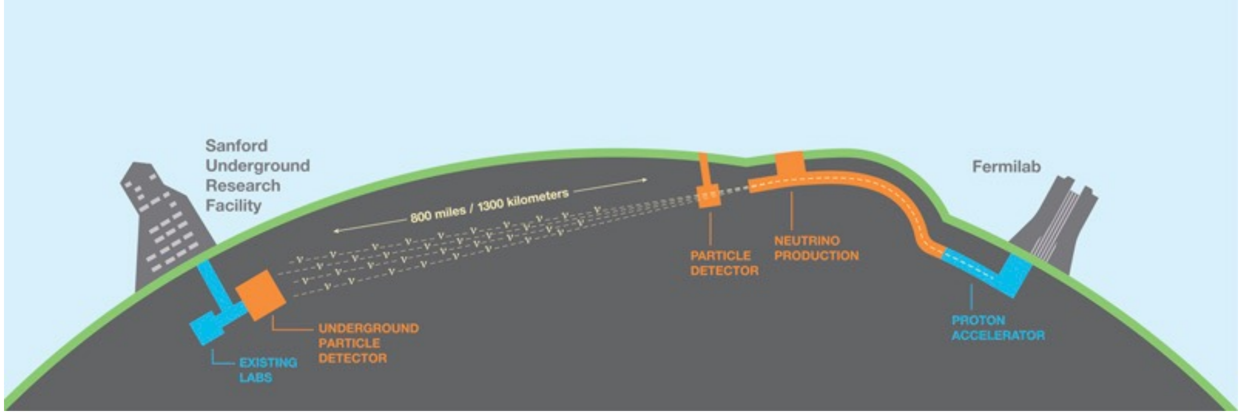
- Introduction to neutrinos.
- **Deep learning in neutrino experiments:**
  - Deep Underground Neutrino Experiment (DUNE).
  - Tokai to Kamioka (T2K).
- Study of deep-learning workloads.
- Summary.

Overview

- Introduction to neutrinos.
- **Deep learning in neutrino experiments:**
  - **Deep Underground Neutrino Experiment (DUNE).**
  - Tokai to Kamioka (T2K).
- Study of deep-learning workloads.
- Summary.

The DUNE experiment

- The Deep Underground Neutrino Experiment (**DUNE**) is a **next-generation neutrino oscillation experiment**.



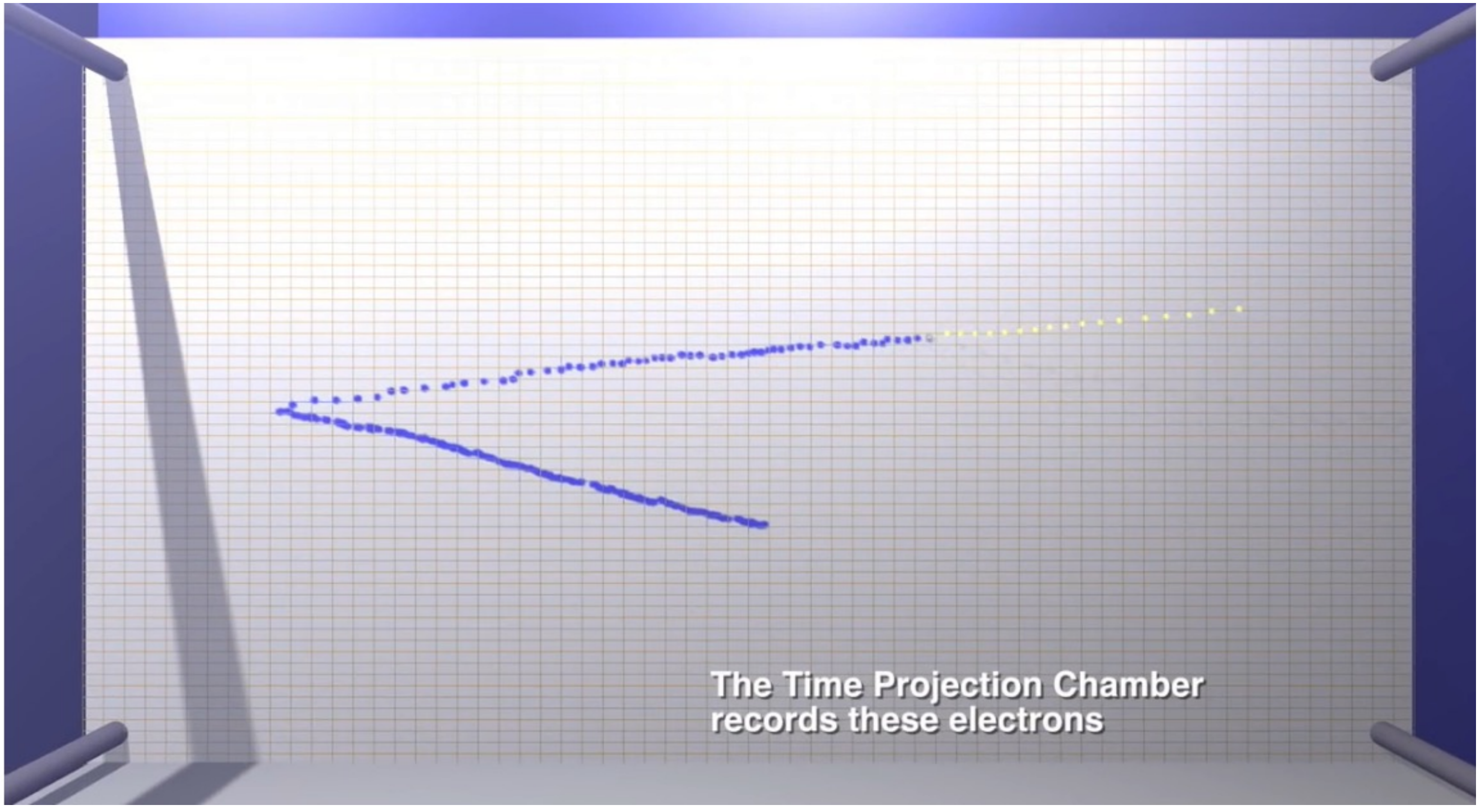
Source: https://www.dunescience.org/

- The far detector is 1300 kilometres from the neutrino beam source.
  - It will consist of four 10 kt **LArTPC detectors**.
- Look for the appearance of electron (anti)neutrinos at the far detector.
  - **Measure *CP*-violation**.

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

LArTPC

- Liquid-Argon Time Projection Chamber (LArTPC).
  - This provides **“images” of each neutrino interaction**.

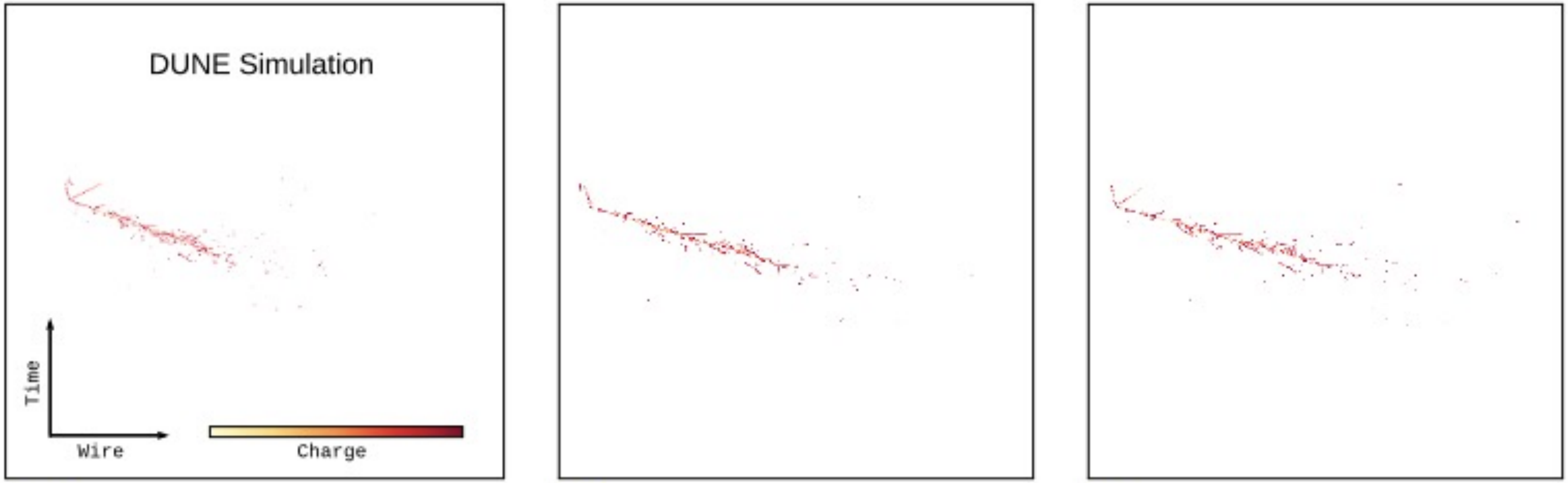


Source: https://www.youtube.com/c/fermilab

Far detector data

- The Far Detectors contain three wire readout planes.
  - This provides three “images” of each neutrino interaction.

- Official simulated electron neutrino interaction (signal).

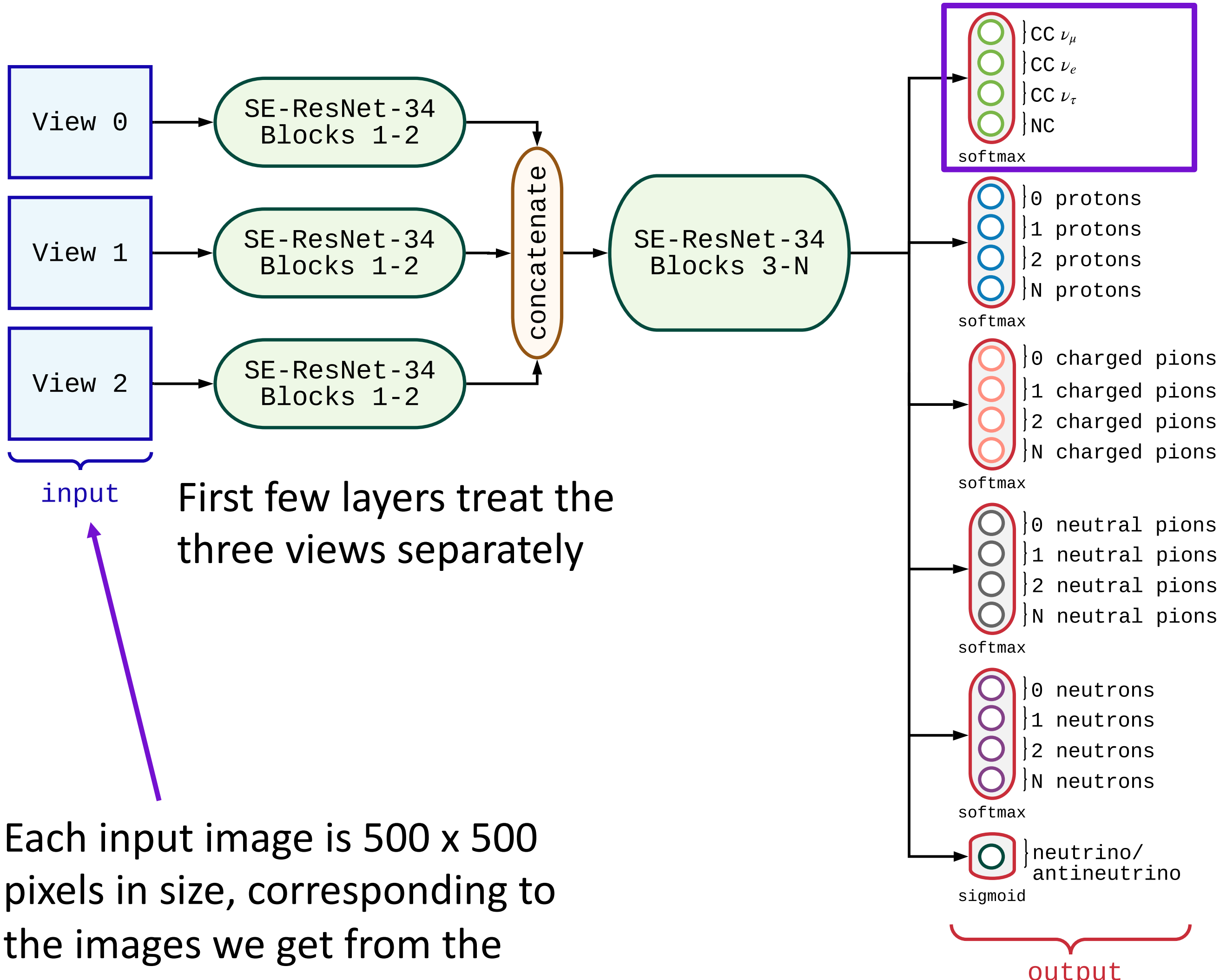


(a) View 0: induction plane (U)

(b) View 1: induction plane (V)

(c) View 2: collection plane (Y)

DUNE CVN overview (2018)

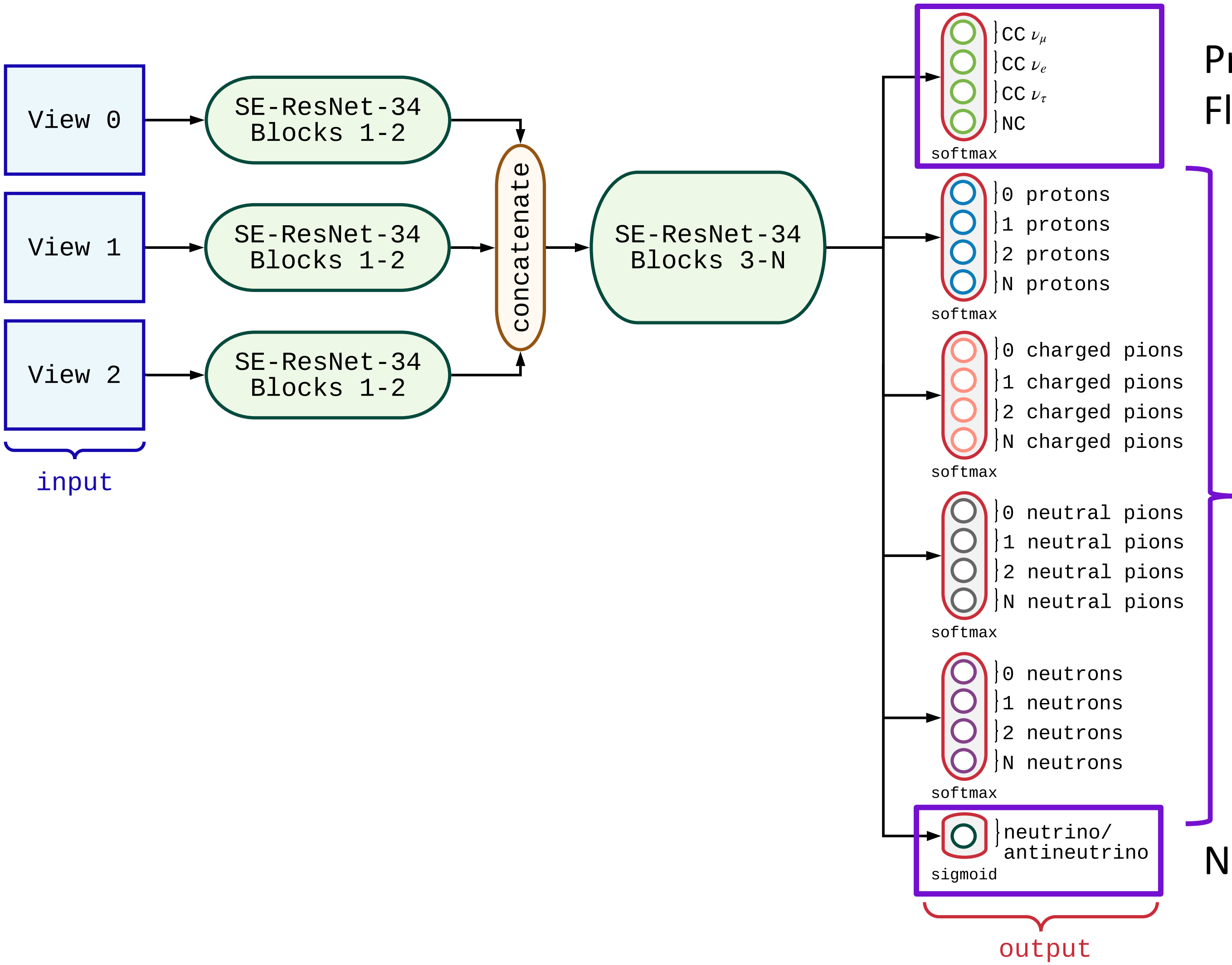


Primary output: Flavour identification

First few layers treat the three views separately

Each input image is 500 x 500 pixels in size, corresponding to the images we get from the three wire readout planes.

DUNE CVN overview (2018)



Primary output:
Flavour identification

Secondary outputs:
Particle counting for exclusive final states

Neutrino / antineutrino

Training and using the CVN

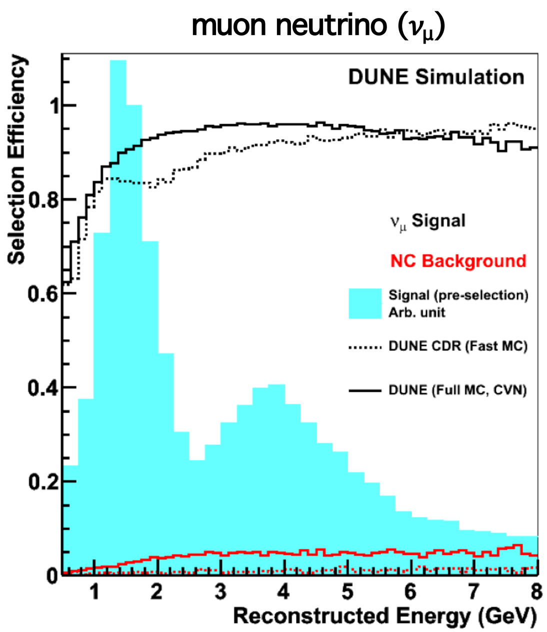
- Training details:
  - Use ~10M images of simulated neutrino interactions.
    - Tested on a fully independent sample (also ~10M images).
  - Trained for 15 epochs on 8 NVIDIA Tesla V100 GPUs, using Keras on top of TensorFlow (recently moved to TF2.0).
    - SGD as optimiser; mini-batch size of 64 events, learning rate of 0.1, weight decay of 0.0001, and momentum of 0.9.
  - Small data release of the code is available at https://github.com/DUNE/dune-cvn.

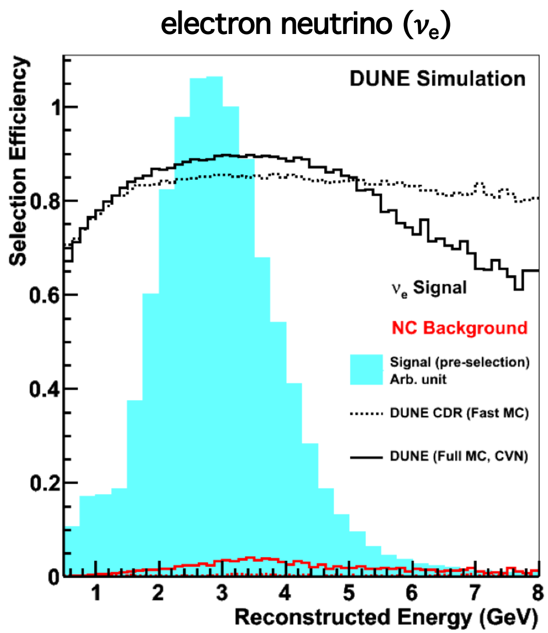
- Publication: ***B. Abi et al. (DUNE Collaboration), ``Neutrino interaction classification with a convolutional neural network in the DUNE far detector'', ISSN: 2470-0029***.
  - https://doi.org/10.1103/PhysRevD.102.092003.

- The **primary output results** (flavour) were **used in the official DUNE neutrino oscillation sensitivity analyses**.
  - DUNE Technical Design Report (TDR): arXiv:2002.03005.
  - DUNE Long-baseline (LBL) analysis: https://doi.org/EPJC/S10052-020-08456-Z.

Efficiencies

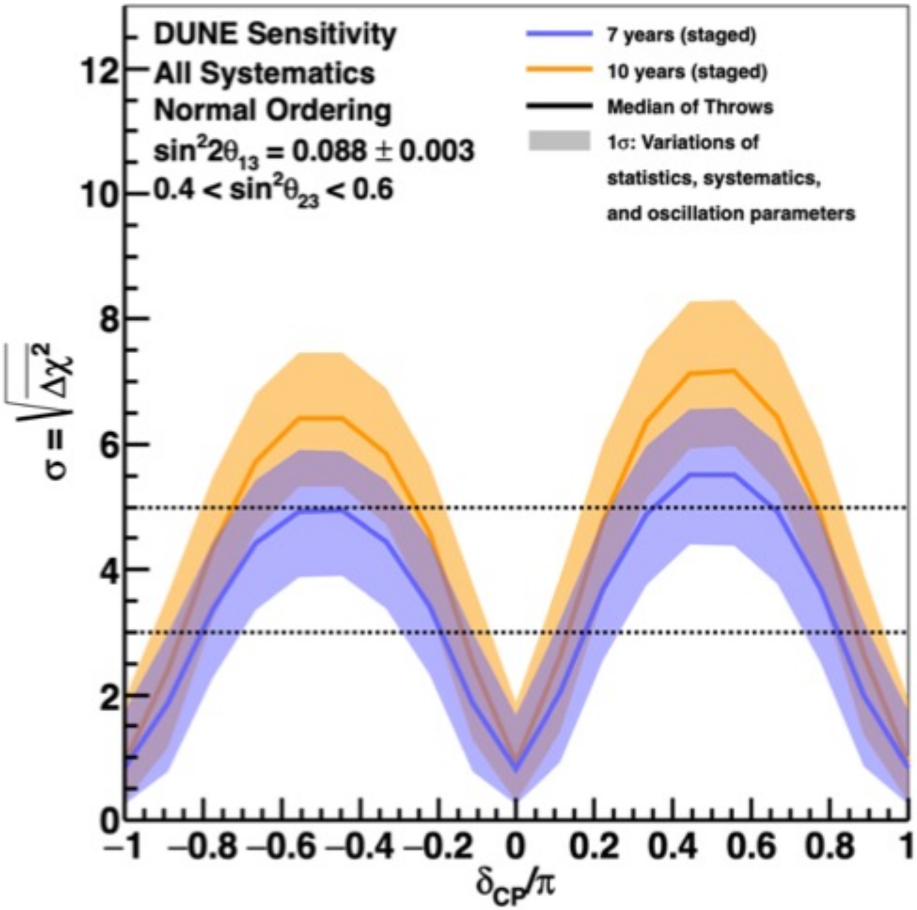
- Muon neutrinos:
  - Select all events that are more than 50% likely to be muon neutrinos.
  - **Over 90% selection efficiency** in the flux peak.
- Electron neutrinos:
  - Select all events that are more than 85% likely to be electron neutrinos.
  - **Over 90% selection efficiency** in the flux peak.





DUNE *CP*-violation sensitivity

- Same selection criteria:
  - ν_e selection: $P(\nu_e) > 85\%$.
  - ν_μ selection: $P(\nu_\mu) > 50\%$.
- The solid lines show the median sensitivity.
- **Results available at DUNE Long-baseline analysis article: https://doi.org/10.1140/epjc/s10052-020-08456-z**
- **Milestone for the experiment!**



Light simulation using GANs

- Accurate simulations are critical to HEP experiments.
  - They are typically computationally expensive.
  - There is great interest in fast simulations.
- In the current **DUNE photon detector simulation**, the entire geometry is stored in memory.
  - The idea is to have higher resolution and cover a larger volume, both of which will make it impossibly large.
- The approach is to try the fast-simulation segment from our **Model-Assisted GAN** (MAGAN) to speed things up.
  - Modification of a Generative adversarial network (GAN); details in backup.
  - S. Alonso-Monsalve and L. H. Whitehead, "Image-Based Model Parameter Optimization Using Model-Assisted Generative Adversarial Networks," in *IEEE Transactions on Neural Networks and Learning Systems*, 2020. DOI: https://doi.org/10.1109/TNNLS.2020.2969327.

Generative adversarial networks

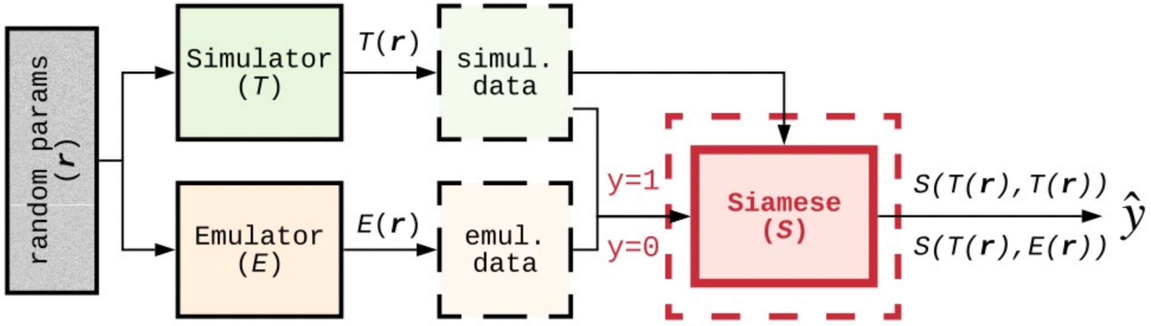
- Generative adversarial networks (GANs) have been shown to produce fake images indistinguishable from true images.

arXiv:1812.04948

arXiv:1809.11096

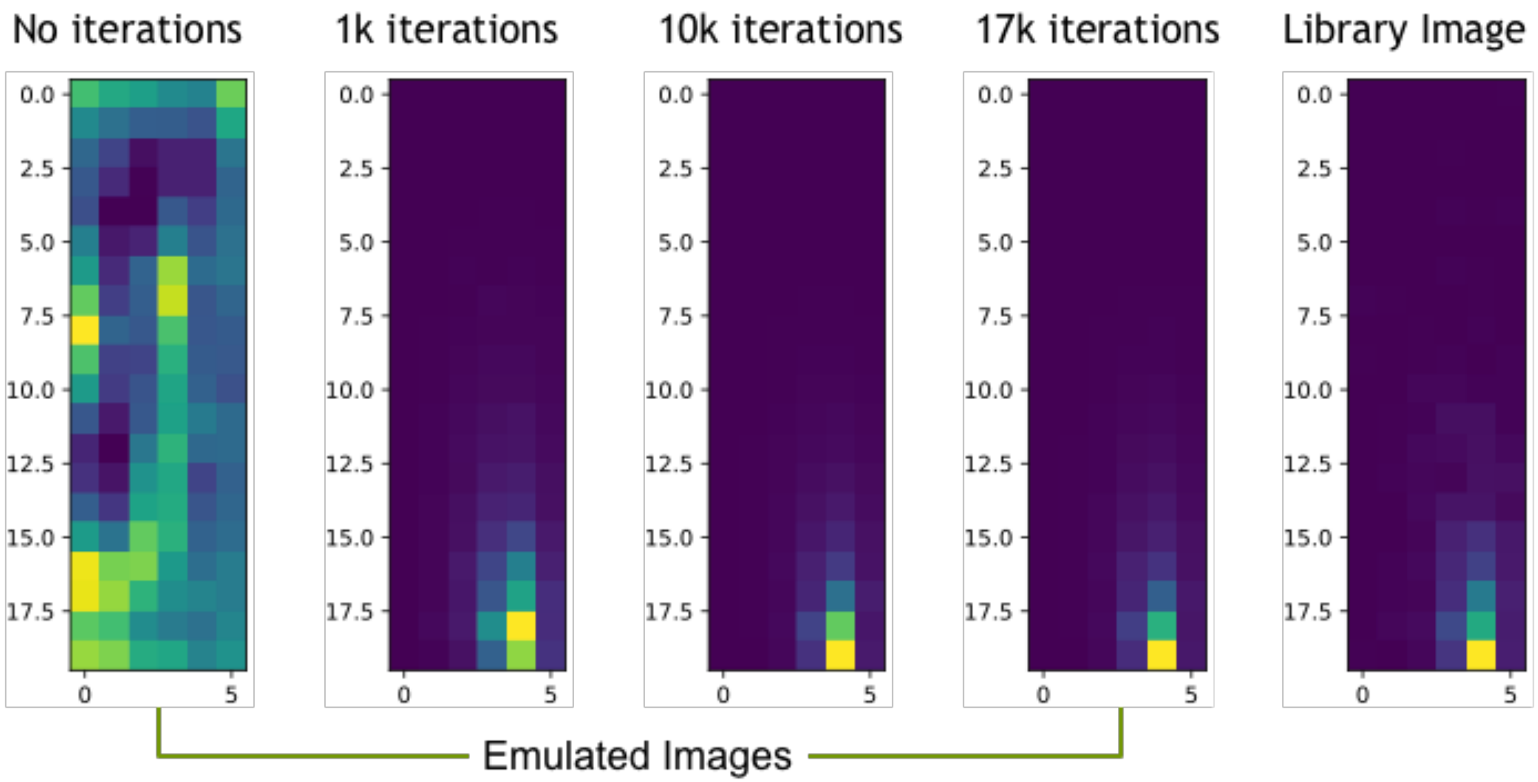
Application to the DUNE photon detector simulation (2019)

- The **goal is to learn the whole simulation using a GAN**.
- The model parameters are just (*x*,*y*,*z*).
  - Output: photon detector system as a 20x6 pixel image, where each pixel gives the visibility of one photon detector.
- Trained on 3M images.
- Our implementation is similar to a conditional-GAN.
  - However, instead of using a standard discriminator, we use a Siamese network in order to make sure the true (simulated) and the fake (emulated) images are the same for the same input parameters.



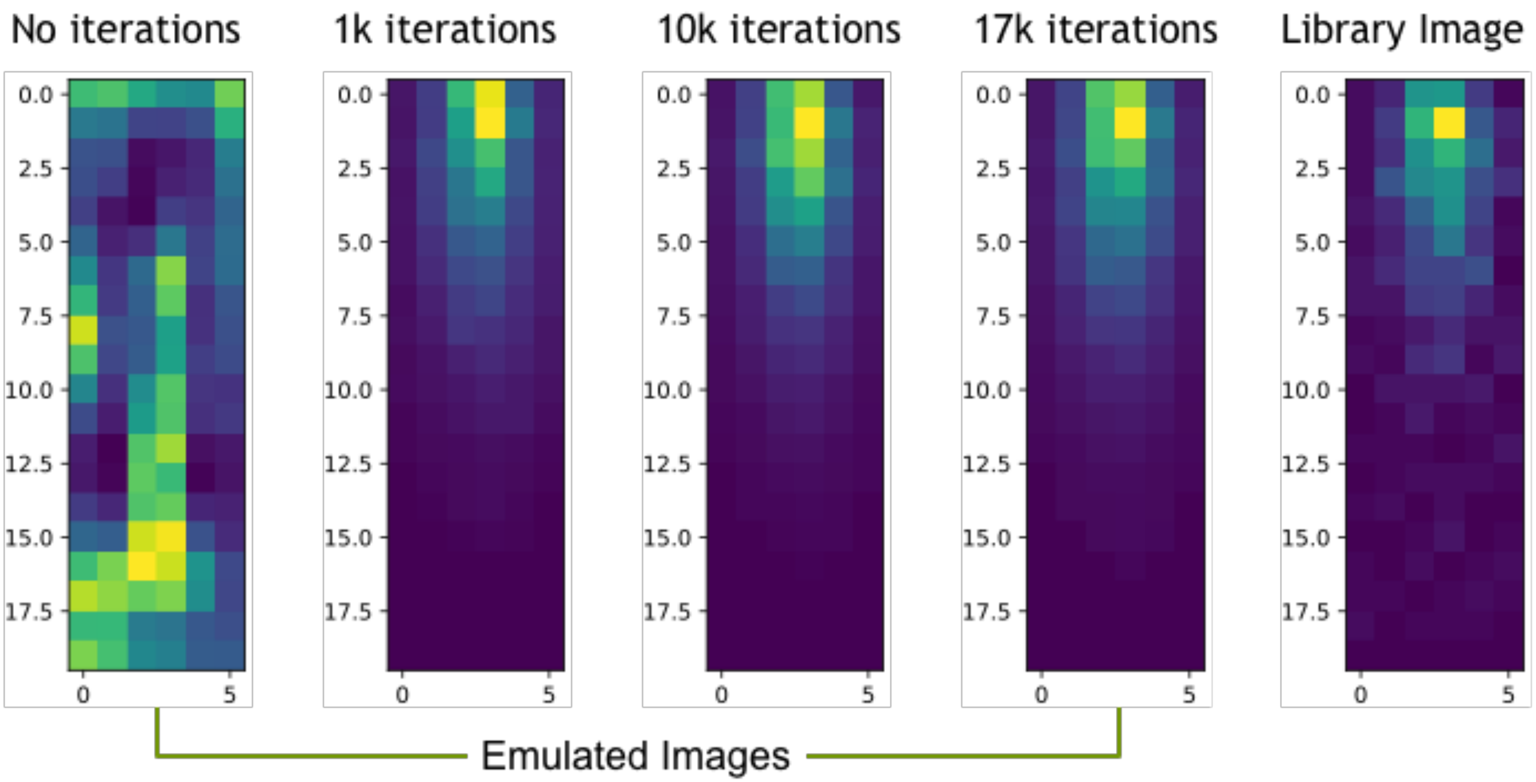
Example Image I

- We trained for roughly 17k iterations.



Example Image II

- We trained for roughly 17k iterations.



- The simulation takes ~1 week to produce 1M images, while **the GAN takes less than two minutes to produce the same number of images on a V100 GPU**.

Overview

- Introduction to neutrinos.
- **Deep learning in neutrino experiments:**
  - Deep Underground Neutrino Experiment (DUNE).
  - **Tokai to Kamioka (T2K).**
- Study of deep-learning workloads.
- Summary.

T2K

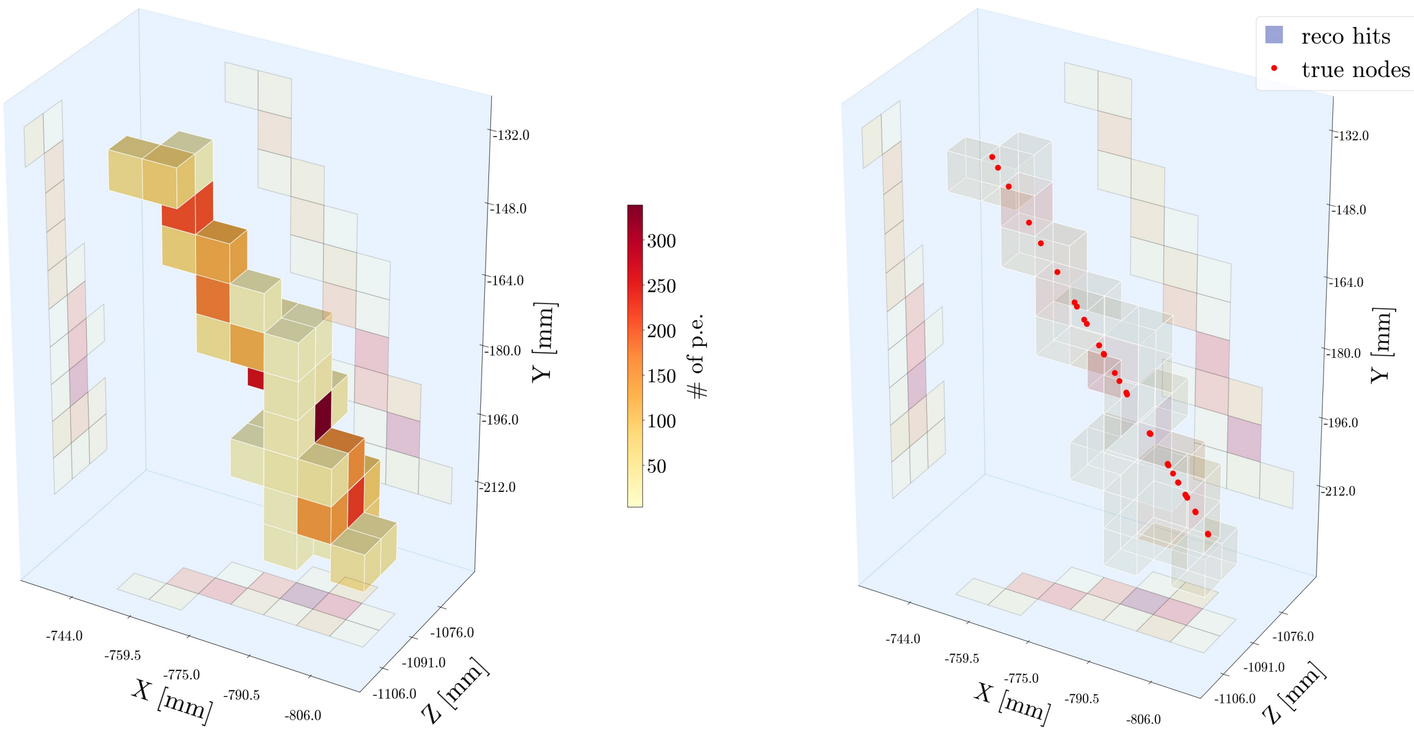
- **Tokai to Kamioka (T2K)** is a **long-baseline neutrino experiment in Japan**, and is studying neutrino oscillations.

- Super Kamiokande (far detector): very large cylinder of ultra-pure water, detects muon neutrino after oscillating.

- ND280 (near detector): measures the number of muon neutrinos in the beam before any oscillations occur and characterizes the physical properties of the beam.
  - In the near future, **an upgrade of the ND280 is planned**.

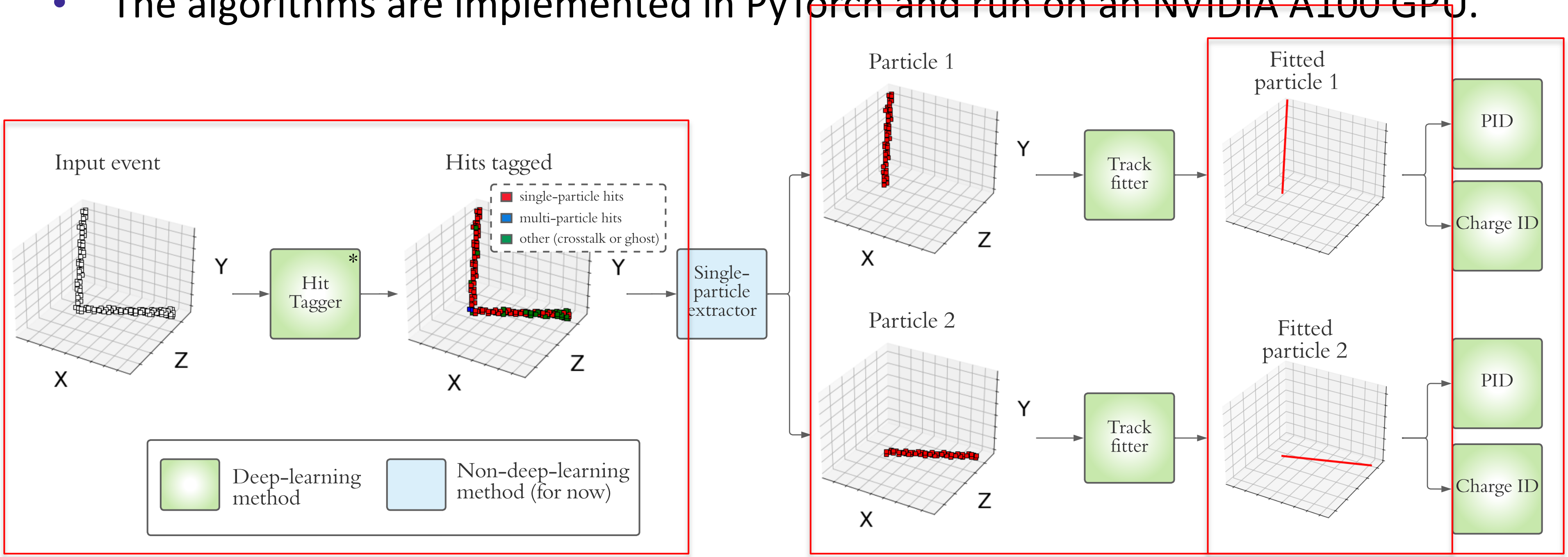
T2K's ND 280 upgrade: SuperFGD detector

- Full active fine-grained detector (FGD) with three views: SuperFGD.
  - Optically independent cubes: spatial localization of scintillation light.
  - Lower momentum threshold: 1 single hit gives immediately XYZ.
  - Example of a simulated muon neutrino:



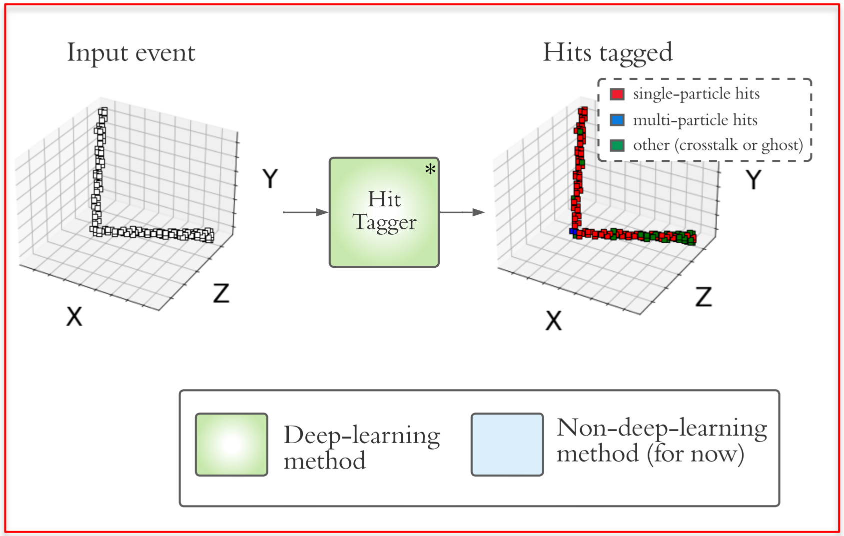
Reconstruction-chain using deep learning

- Most steps of the **reconstruction** in the SFGD can be **done using deep learning**:
  - **Method 1**: Hit tagging (identify different kinds of hits).
  - **Method 2**: Track fitting (adjust the particle trajectory)
  - **Method 3**: Identify the particle and the charge.
- The algorithms are implemented in PyTorch and run on an NVIDIA A100 GPU.

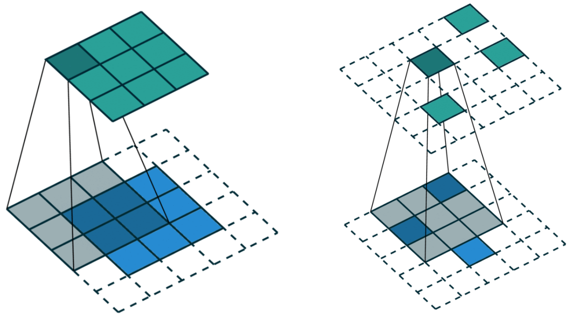


Method 1: Hit tagging (2020)

- Classify each individual hit as:
  - **Single-particle hit**: only one particle passes through the hit cube and no other tracks pass through its adjacent cubes
  - **Multiple-particle hit**: at least two different particles pass through the hit cube and its adjacent cubes.
  - **Other**: mainly crosstalk.
- Using a sparse U-Net-based neural network architecture.
  - Neutrino detector data is inherently sparse, in contrast to "real world" images (i.e., photos).
    - **Standard CNNs are very inefficient when applied to sparse data.**



Dense convolution **Sparse convolution**



"Dense" image

https://www.britannica.com/

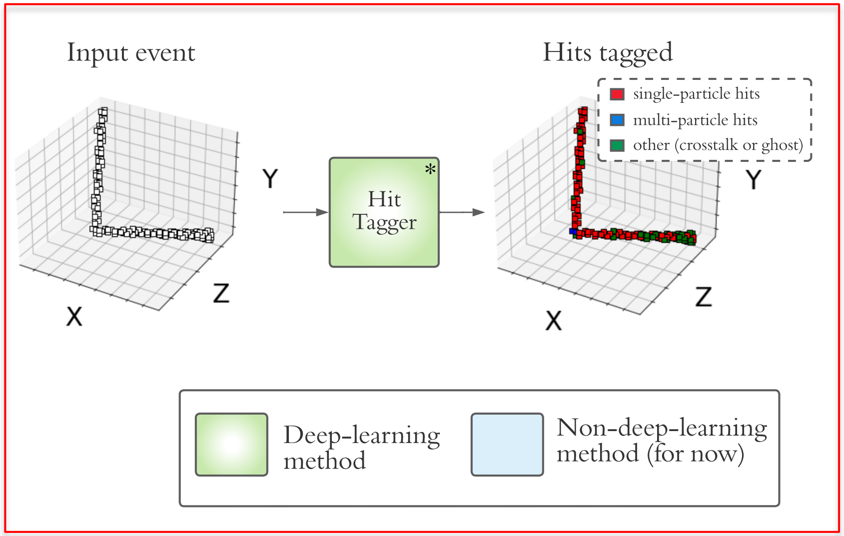
"Sparse" image

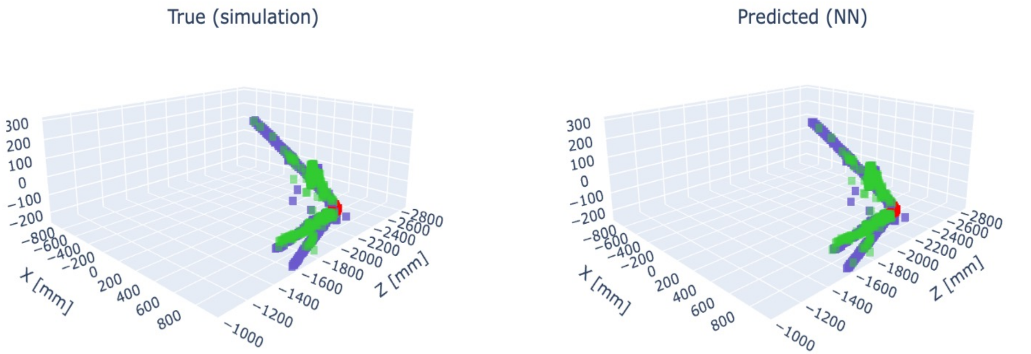
https://link.aps.org/doi/10.1103/PhysRevD.102.092003

Method 1: Hit tagging (2020)

- Classify each individual hit as:
  - **Single-particle hit**: only one particle passes through the hit cube and no other tracks pass through its adjacent cubes
  - **Multiple-particle hit**: at least two different particles pass through the hit cube and its adjacent cubes.
  - **Other**: mainly crosstalk.
- Using a sparse U-Net-based neural network architecture.
  - Neutrino detector data is inherently sparse, in contrast to "real world" images (i.e., photos).
    - **Standard CNNs are very inefficient when applied to sparse data.**

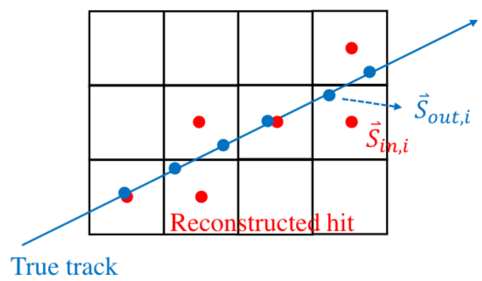
	True multiple-particle hit	True single-particle hit	True other
Pred. multiple-particle hit	**83.48%**	10.70%	5.83%
Pred. single-particle hit	0.68%	**97.52%**	1.80%
Pred. other	1.24%	6.88%	**91.87%**





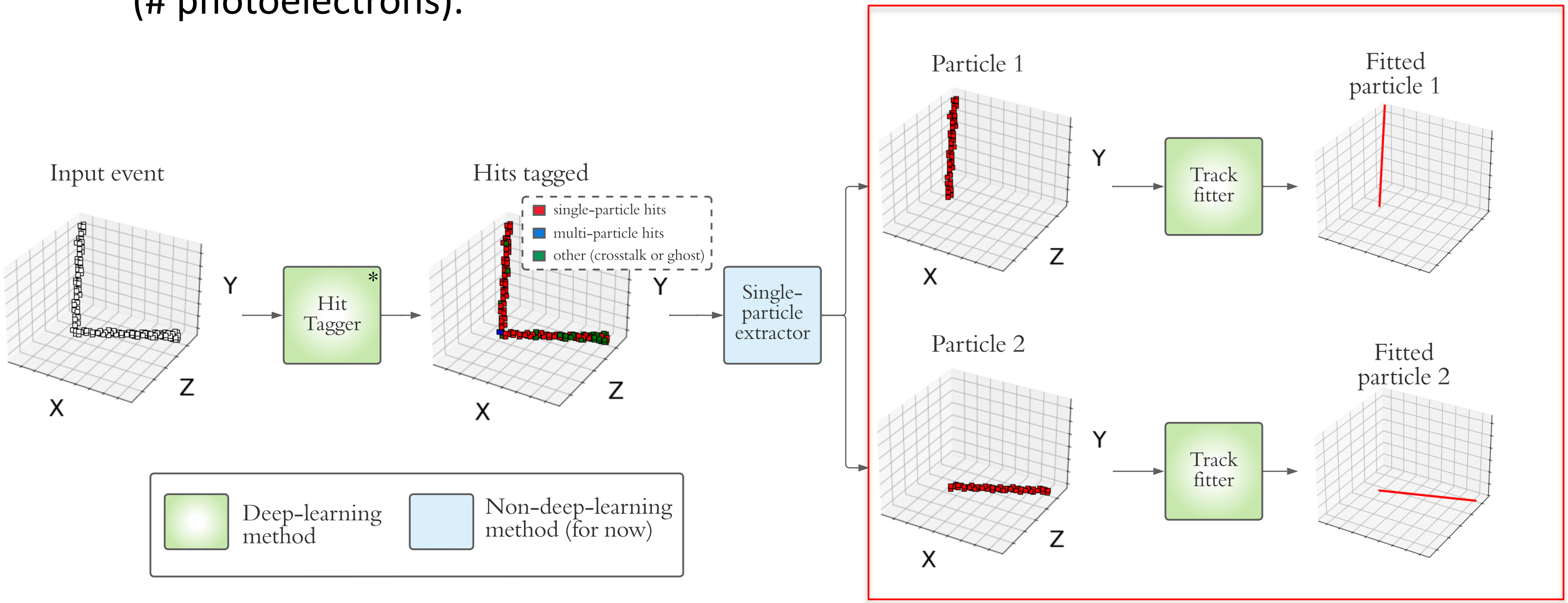
Method 2: Track fitter (2022)

- Based on track hits information, we want to use **neural networks to predict node states along the track (particle trajectory points)**.
- For each state we consider 3D position and energy deposition (# photoelectrons).



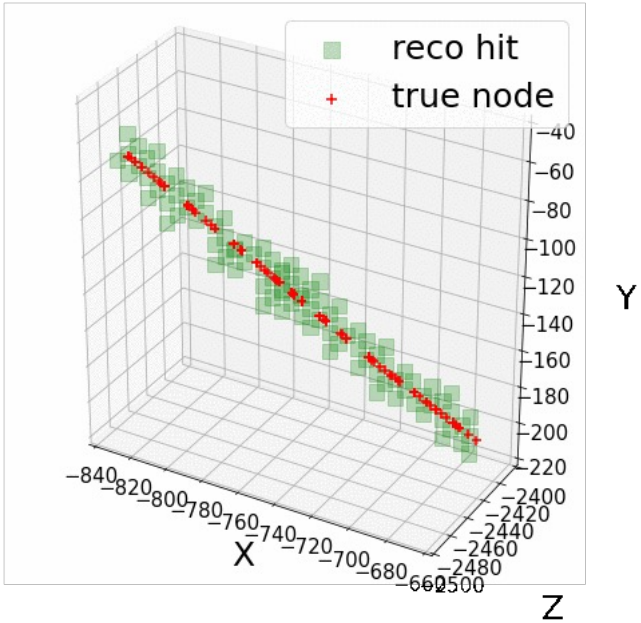
- Input hit state: $\vec{S}_{in,i} = (x_i, y_i, z_i, E_i), i = 1, \cdots, N.$
- Output node state: $\vec{S}_{out,i} = (x_i, y_i, z_i), i = 1, \cdots, N.$
- Use neural network to construct the map:

$$\{\vec{S}_{in,i}\} \rightarrow \{\vec{S}_{out,i}\}$$



Sequential-importance-resampling particle filter (SIR-PF) implemented

- Method:
  - Use the training set to fill a histogram with the following variations of consecutive true nodes:
    - Δx, Δy, Δz, $\Delta\theta$ (in spherical coordinates), Δpe (photoelectrons).
  - Use the first hit as prior (particle gun).
  - In each step, the particles are propagated (resampled) along the track direction.
  - For each particle, the algorithm calculates the variation in x, y, z, θ, and pe, and assigns a likelihood based on the value of the corresponding bin in the previously filled histogram.
    - The next fitted node is the weighted average (using the likelihood) of the positions of the different particles.
  - Weighted average of forward and backward fitting.



- **Ran twice:**
  - **On all the hits** (direct comparison with NNs).
  - **On track-hits only** (unrealistic best-posible scenario).

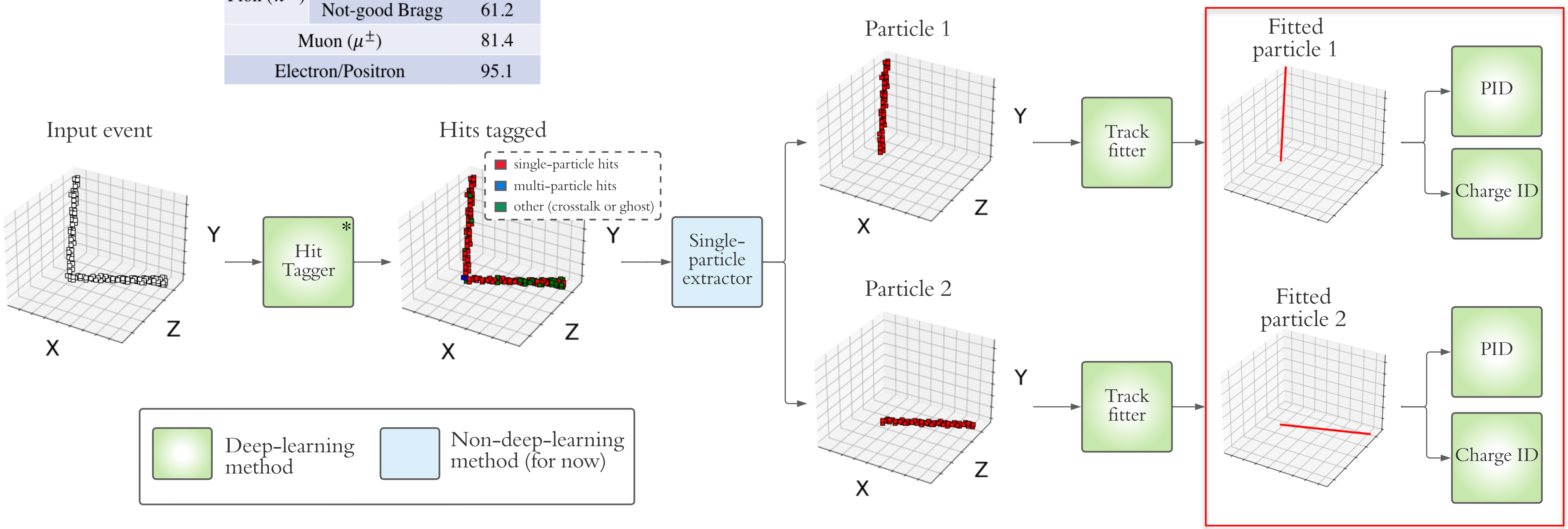
Method 3: PID and Charge ID (2020)

- Approach:
  - **Train two sparse neural networks for particle and charge identification** (PID and charge ID).
  - PID results (left) and charge ID (right) using NNs **outperform any other method used by the experiment**.

SFGD contained tracks:

Particle Type		SCNN
		Efficiency
Proton	Good Bragg	93.5
	Not-good Bragg	63.2
Pion ($\pi^{\pm}$)	Good Bragg	77.3
	Not-good Bragg	61.2
Muon ($\mu^{\pm}$)		81.4
Electron/Positron		95.1

Method	Efficiency
CNN	96.5%
Shower CoM	84.8%
Primary track	81.7%



Overview

- Introduction to neutrinos.
- Deep learning in neutrino experiments:
  - Deep Underground Neutrino Experiment (DUNE).
  - Tokai to Kamioka (T2K).
- **Study of deep-learning workloads.**
- Summary.

Performance study of deep-learning workloads

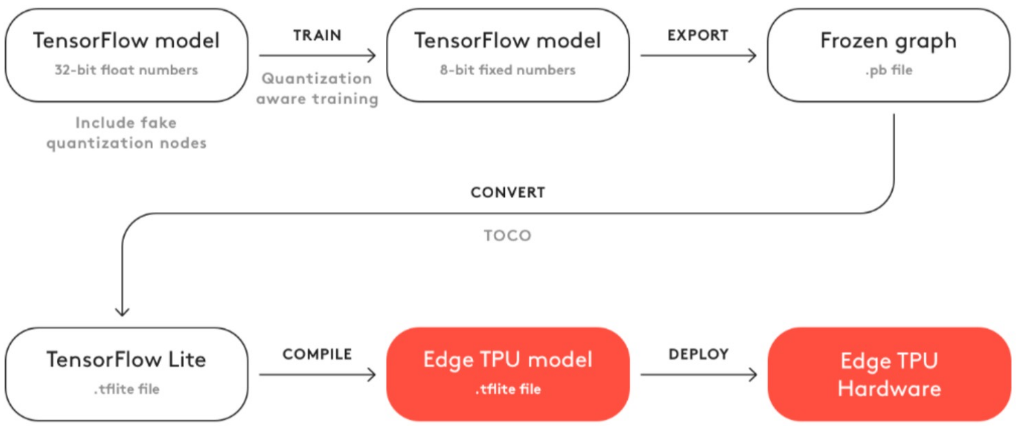
- Being able to **run computationally efficient deep-learning workloads is becoming key for both science and industry**.
  - In the case of the neutrino world, it would allow us to save time and money.
- For training, scaling the computation of deep-learning models the most reasonable option.
  - Many options: parallelise the computation, understand your GPU(s), avoid bottlenecks in the data I/O by having multiple processes preparing the inputs, etc.
- For inference, **a possible approach is to run trained neural networks on deep-learning accelerator boards**
  - In DUNE, we are exploring Google TPUs or FPGAs designed for running deep-learning workloads.

Fermilab - Google Collaboration

- Specifications:

	CPU	GPU	Edge TPU
Model	Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz	NVIDIA Tesla K80 (from Google Colab)	Coral Edge TPU
TDP*	65 w (16 w per core)	300 w	2 w
Price (USD)	200	5,000	80

- Generating the right model:



*Thermal Design Power (TDP) represents the average power, in watts, the processor dissipates when operating at Base Frequency with all cores active under an Intel-defined, high-complexity workload.

Results

- Tested using ResNet-50 on MNIST dataset:

	CPU (Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz)	GPU (NVIDIA Tesla K80)	Coral Edge TPU
Categorical accuracy	97%	97%	95%
Total inference time (10k images)	142 s	14.7 s	356 s
Inference per image	14 ms	1.5 ms	35 ms

- Tested using the DUNE CVN for neutrino identification (50 test images):

	CPU (Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz)	GPU (NVIDIA Tesla K80)	Coral Edge TPU
Categorical accuracy	88%	86%	88%
Total inference time (10k images)	22 s	1 s	5 s
Inference per image	431 ms	20 ms	100 ms

- Costs: $cost/inference = time/inference \times TDP \times cost\ of\ energy = K \times cost\ of\ energy$

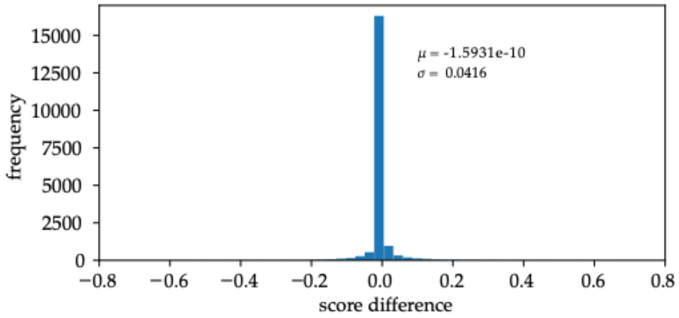
	CPU (Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz)	GPU (NVIDIA Tesla K80)	Coral Edge TPU
K factor (ResNet-50 on MNIST 56x56 images)	0.21	0.45	0.07
K factor (DUNE 500x500 images)	6.9	6	0.2

- GPU appears to be by far the fastest piece of hardware.
- Edge TPU performs better with bigger images
- Edge TPU showed the smallest cost per inference and CPU showed the biggest cost per inference.

CERN Openlab - Micron Collaboration

- Hardware: SB-852.
  - FPGA-based unit from Micron.
  - Designed for running neural networks.
  - 64GB DDR4 SODIMM.
  - High-bandwidth / low-latency.
- Workflow:
  - Convert the network into ONNX.
  - Compile it using the Micron Framework.
  - Deploy into the inference engine.
- Future plans:
  - Measure time and energy.
  - Integrate the FPGA in the protoDUNE-SP DAQ.
  - Test how far we can go in the data selection or even in fast online reconstruction.

- Already ran the DUNE CVN on the FPGA.
  - **Same results in GPU and FPGA.**



- **~x2.6 time speedup with respect to the hardware we use in DUNE for inference.**

Processor	Average time (ms)	STD	Min	Max
SB852	103.6074	0.5505	102.4658	105.0381
CPU (i7-8750H)	264.8545	0.8653	262.1692	267.2548

Overview

- Introduction to neutrinos.
- Deep learning in neutrino experiments:
  - Deep Underground Neutrino Experiment (DUNE).
  - Tokai to Kamioka (T2K).
- Study of deep-learning workloads.
- **Summary.**

Summary

- **Deep learning algorithms provide many powerful mechanisms for processing input data from many different fields**, including high-energy physics and neutrino experiments in particular.

- Several schemes using deep learning in neutrino experiments:
  - **Standard CNNs** for favour identification.
  - **GANs** for fast simulations.
  - **Sparse CNNs** for hit tagging, particle and charge identification.
  - **Particle filters** for particle tracking.

- Inference via edge computing: two current projects.
  - Using **Google TPUs**.
  - Using **Micron FPGAs**.

- Next steps: approach to computing systematic uncertainties (**need to test the methods extensively to avoid biases**):
  - Test on different statistically independent samples (also, samples from different generators).
  - Understand what the networks are learning (e.g., occlusion tests).

Machine learning and high-performance computing for neutrino oscillations

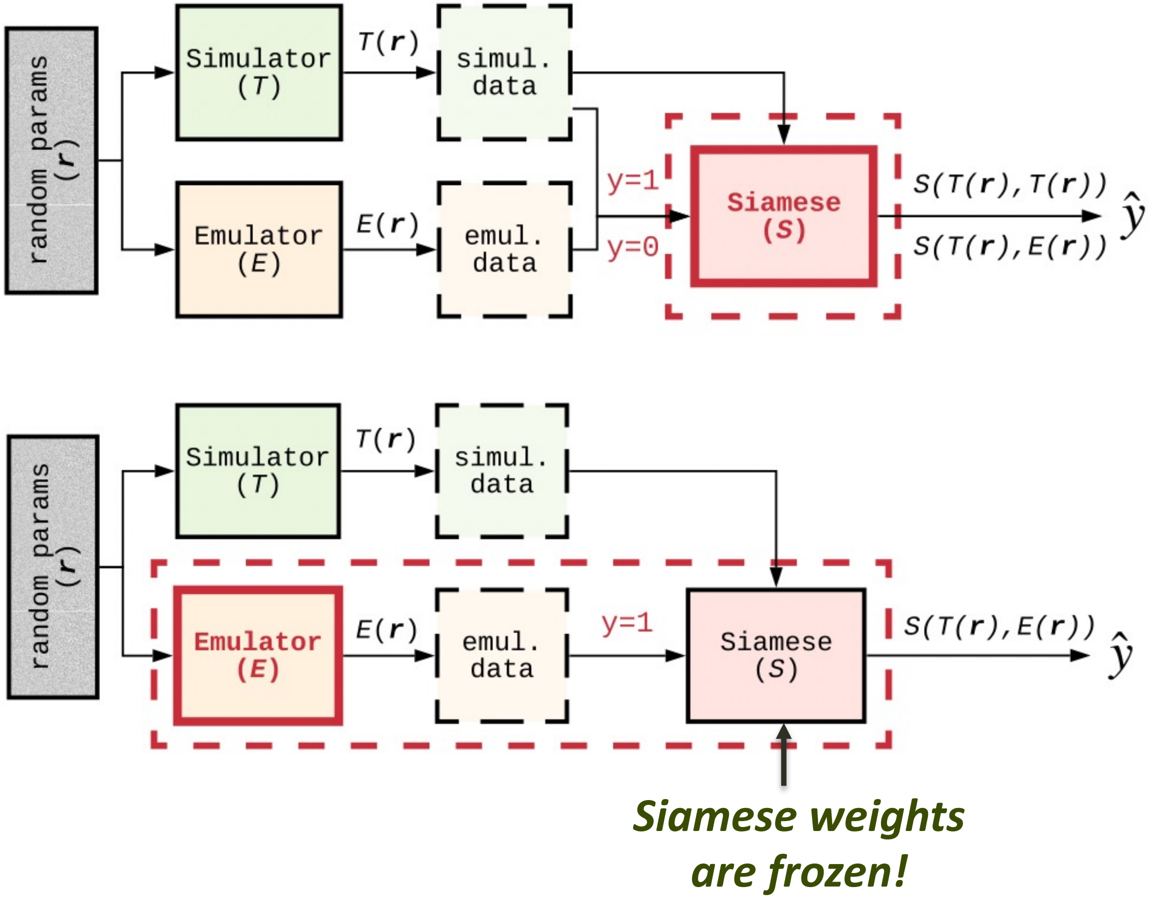
Saúl Alonso-Monsalve
ETH Zurich

Fall Seminar Series
National HPC Competence Centre
The Cyprus Institute
18 October 2022

Backup Slides

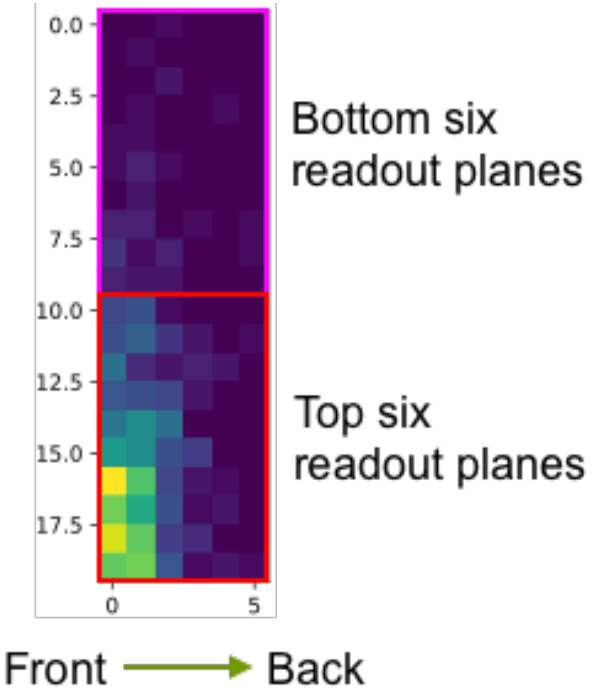
Model-Assisted GAN

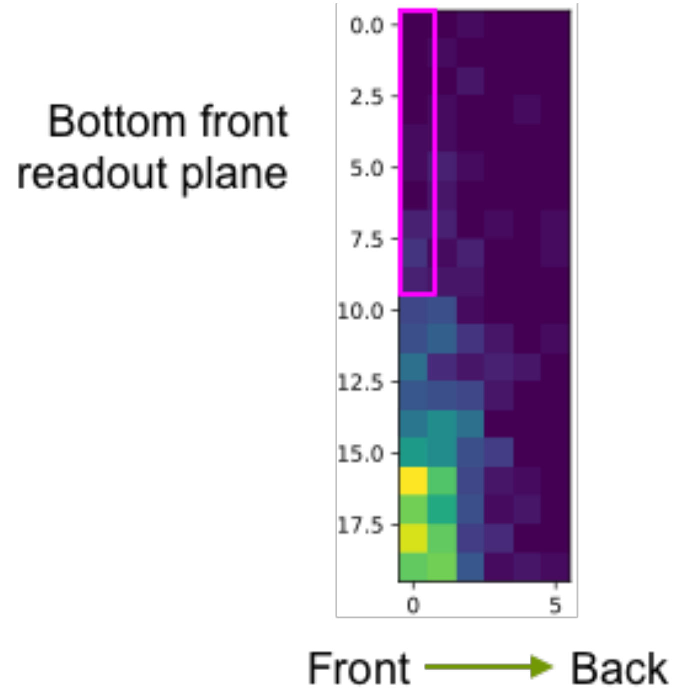
- The **Siamese network** *S* is trained to learn the similarity of the simulated and emulated images.
- The **emulator** *E* is trained to learn to create emulated images that mimic simulated images, so that *E* and the **simulator** *T* generate an identical image from all possible parameter sets.



DUNE photon detector system: Image format

- The images are 20 x 6 pixels.
  - Two readout planes high, and six readout planes wide.





CVN occlusion tests

- Prove the robustness of the CVN by hiding portions of the input events.
    - I.e., changing a small patch of pixels to zeros.

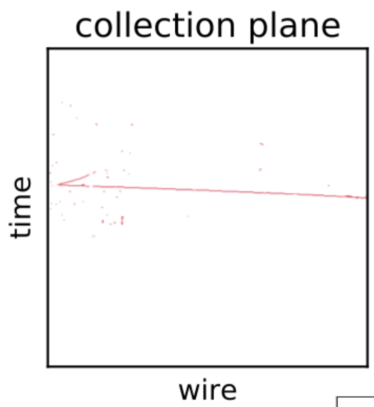
- Use collection plane view only.
    - It is not a perfect test, but it gives us a good idea of what the CVN is using for classification.

- Compare the CVN scores before and after withholding a small patch of an input event from the network.
    - If the scores remain the same (or very close) means the CVN is robust against small image variations.
    - The score difference is placed into a separate map at the pixel corresponding to the centre of the patch.

- Repeat this procedure across the entire input image.

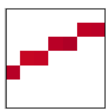
CVN occlusion tests: example

- Input (500x500 pixel image):

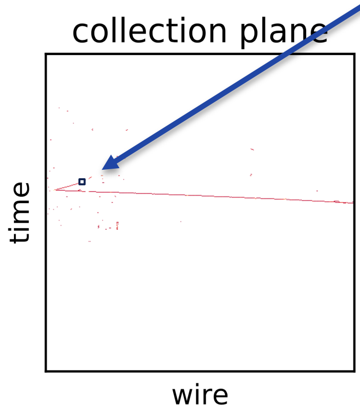


CVN

CC ν_μ score: 0.968
CC ν_e score: 0.000
CC ν_τ score: 0.032
NC score: 0.000



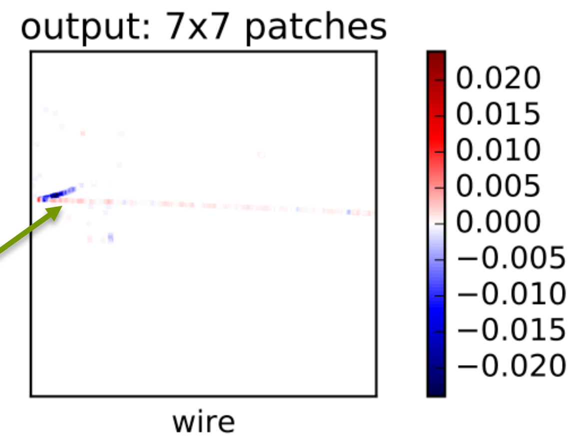
Small 7x7 patch is withheld from the network



CVN

CC ν_μ score: 0.953
CC ν_e score: 0.002
CC ν_τ score: 0.041
NC score: 0.004

0.953 - 0.968 = -0.015

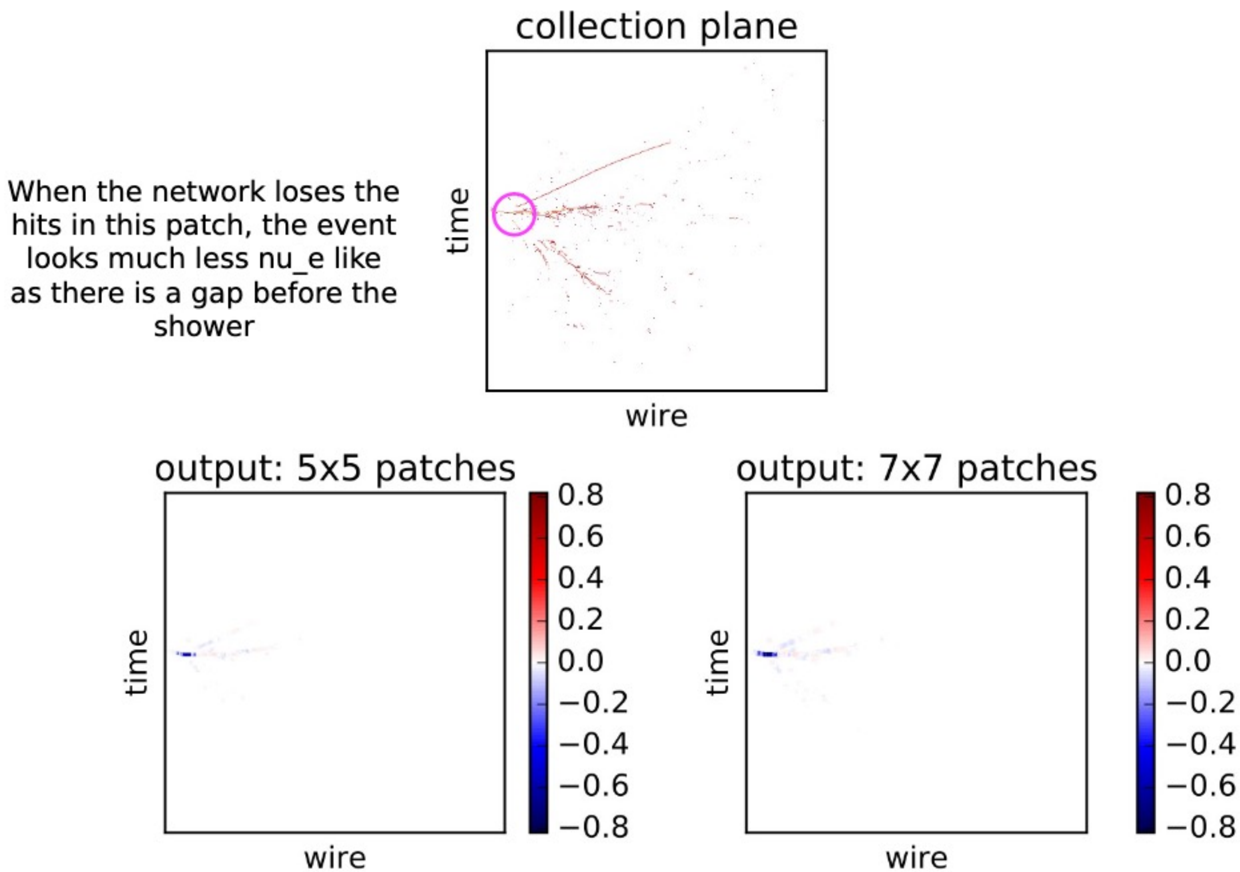


We repeat the same procedure across the entire input image (stride of 1).

CVN occlusion tests

- We ran tests on a small sample (100 events).
- 5x5 pixel patches, and 7x7 pixel patches.
  - Applied to collection plane view only.
- Tests incredibly slow.
  - Not performing tests on patches that are already blank, but still needed to run the CVN hundreds (or event thousands) of times per event.
  - ~10 hours to run the tests on a NVIDIA V100 GPU.

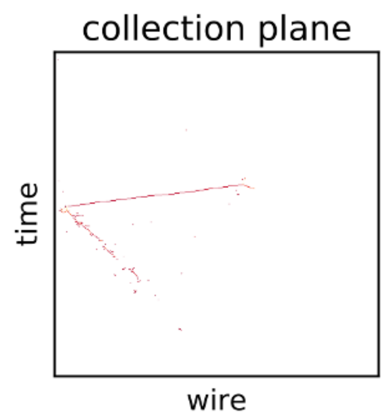
CVN occlusion tests: event gallery (I)

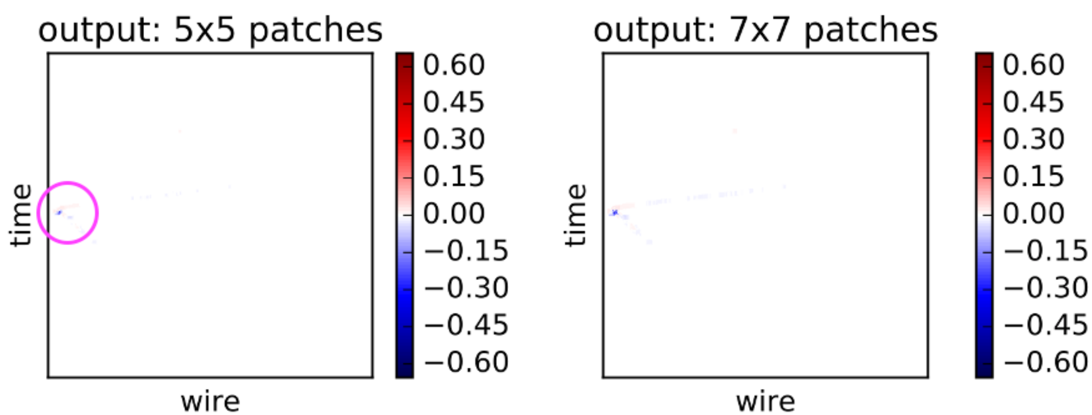


- True label: CC ν_e
- CVN original scores:
  - CC ν_μ score: 0.0009
  - CC ν_e score: 0.9184
  - CC ν_τ score: 0.0090
  - NC score: 0.0717
- CVN scores (largest 5x5 difference):
  - CC ν_μ score: 0.0015
  - CC ν_e score: 0.1003
  - CC ν_τ score: 0.0098
  - NC score: 0.8884
- CVN scores (largest 7x7 difference):
  - CC ν_μ score: 0.0028
  - CC ν_e score: 0.1872
  - CC ν_τ score: 0.0128
  - NC score: 0.7972

CVN occlusion tests: event gallery (II)

Removing the start of the electron shower reduces the CC nue score - as we'd hope

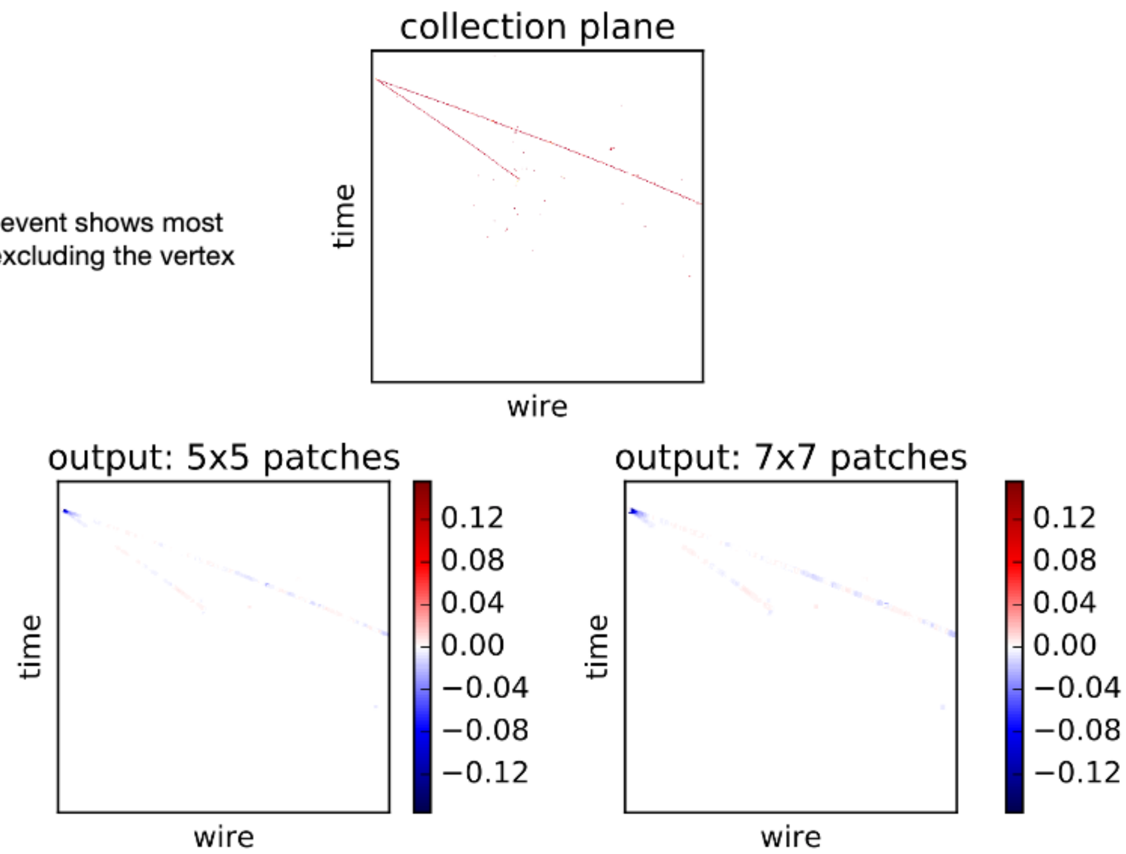




- True label: CC ν_e
- CVN original scores:
  - CC ν_μ score: 0.0007
  - CC ν_e score: 0.9560
  - CC ν_τ score: 0.0185
  - NC score: 0.0248
- CVN scores (largest 5x5 difference):
  - CC ν_μ score: 0.0026
  - CC ν_e score: 0.3013
  - CC ν_τ score: 0.0234
  - NC score: 0.6727
- CVN scores (largest 7x7 difference):
  - CC ν_μ score: 0.0027
  - CC ν_e score: 0.4975
  - CC ν_τ score: 0.0358
  - NC score: 0.4640

CVN occlusion tests: event gallery (III)

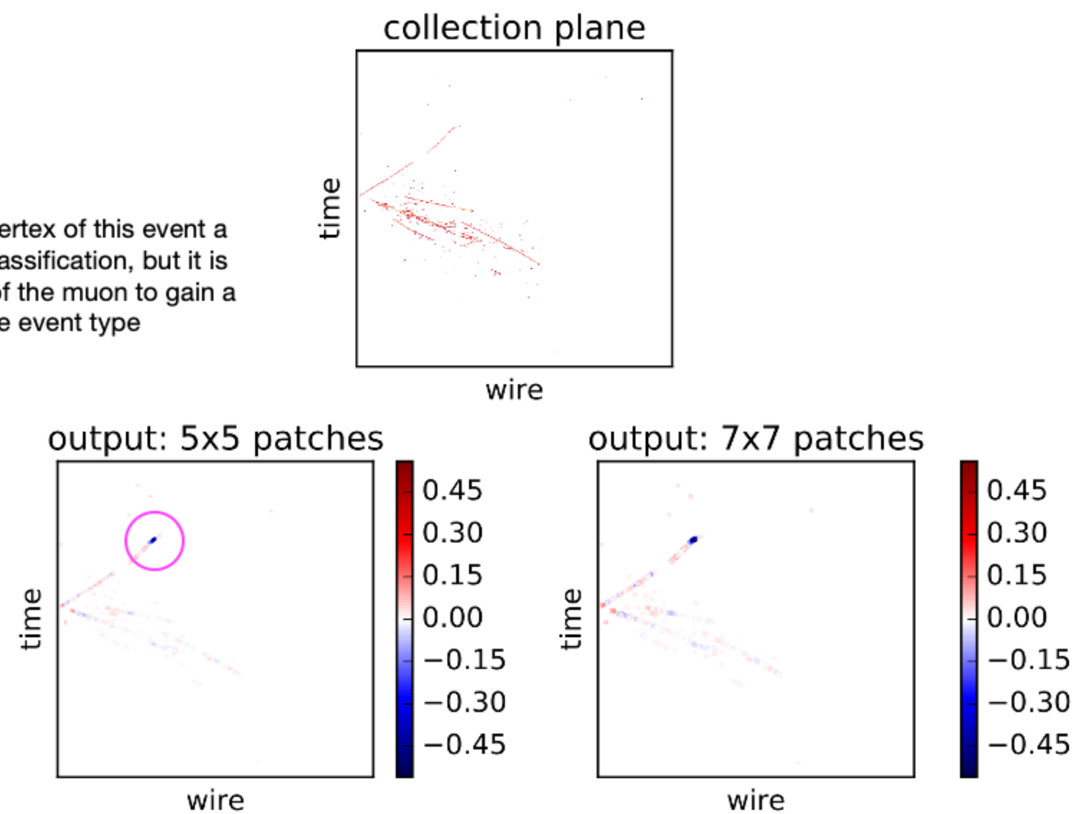
Simpler CC Numu event shows most degradation when excluding the vertex



- True label: CC ν_μ
- CVN original scores:
  - CC ν_μ score: 0.9672
  - CC ν_e score: 0.0002
  - CC ν_τ score: 0.0258
  - NC score: 0.0068
- CVN scores (largest 5x5 difference):
  - CC ν_μ score: 0.8112
  - CC ν_e score: 0.0002
  - CC ν_τ score: 0.0953
  - NC score: 0.0933
- CVN scores (largest 7x7 difference):
  - CC ν_μ score: 0.8112
  - CC ν_e score: 0.0002
  - CC ν_τ score: 0.0953
  - NC score: 0.0933

CVN occlusion tests: event gallery (IV)

The CVN finds the vertex of this event a bit ambiguous for classification, but it is using the end point of the muon to gain a handle on the event type

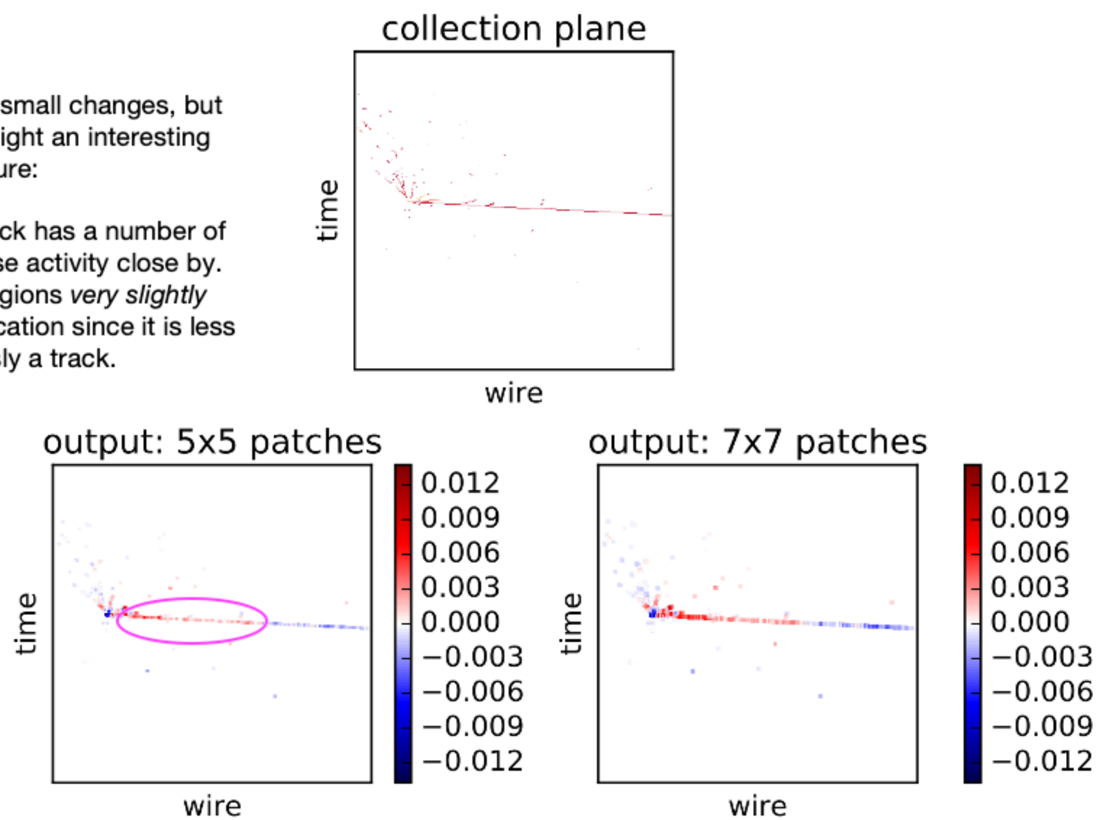


- True label: CC ν_μ
- CVN original scores:
  - CC ν_μ score: 0.7142
  - CC ν_e score: 0.0007
  - CC ν_τ score: 0.0750
  - NC score: 0.2101
- CVN scores (largest 5x5 difference):
  - CC ν_μ score: 0.1551
  - CC ν_e score: 0.0011
  - CC ν_τ score: 0.1552
  - NC score: 0.6886
- CVN scores (largest 7x7 difference):
  - CC ν_μ score: 0.1854
  - CC ν_e score: 0.0011
  - CC ν_τ score: 0.1550
  - NC score: 0.6585

CVN occlusion tests: event gallery (V)

This event has very small changes, but we wanted to highlight an interesting feature:

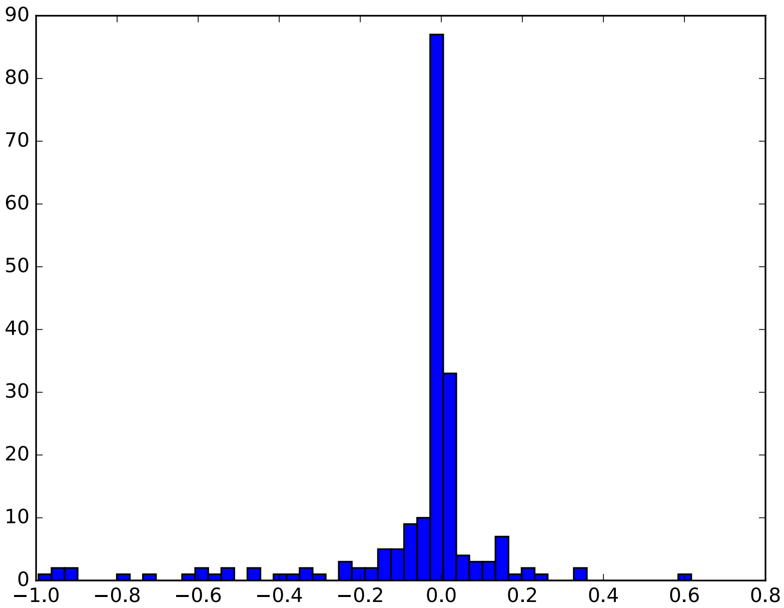
Part of the muon track has a number of delta rays and diffuse activity close by. Excluding these regions *very slightly* improves the classification since it is less ambiguously a track.



- True label: CC ν_μ
- CVN original scores:
  - CC ν_μ score: 0.9614
  - CC ν_e score: 0.0002
  - CC ν_τ score: 0.0372
  - NC score: 0.0012
- CVN scores (largest 5x5 difference):
  - CC ν_μ score: 0.9477
  - CC ν_e score: 0.0001
  - CC ν_τ score: 0.0511
  - NC score: 0.0011
- CVN scores (largest 7x7 difference):
  - CC ν_μ score: 0.9478
  - CC ν_e score: 0.0002
  - CC ν_τ score: 0.0510
  - NC score: 0.0010

CVN occlusion tests: histograms

- Largest score difference distribution (5x5 patches):



- Largest score difference distribution (7x7 patches):