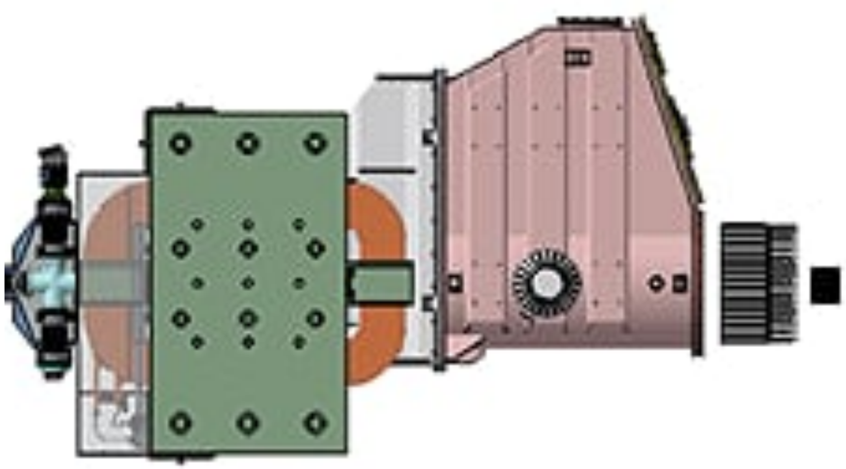
SAPIENZA
UNIVERSITÀ DI ROMA

Simulation and Processing of Signals in the Vetoes of the PADME Experiment

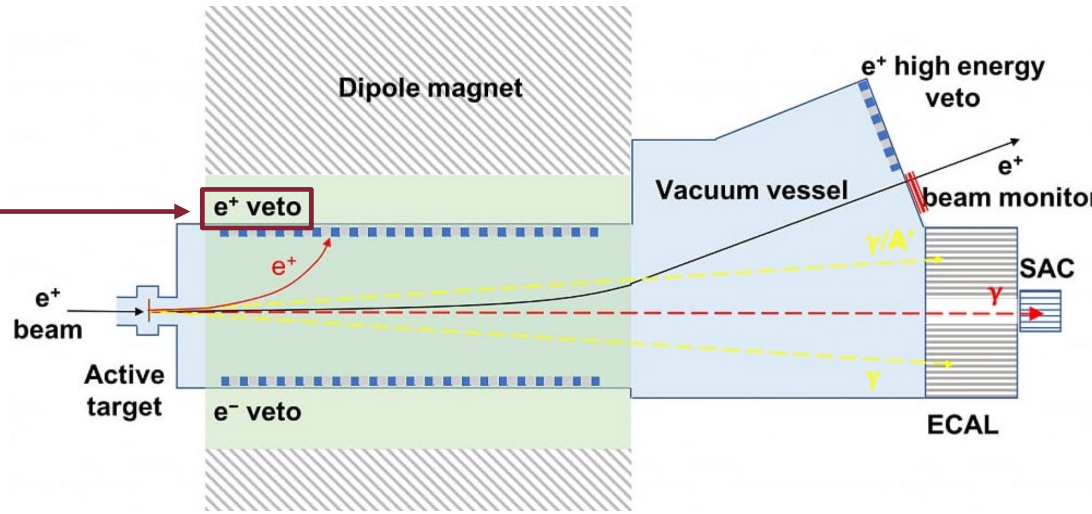
106 Congresso Nazionale SIF 2020

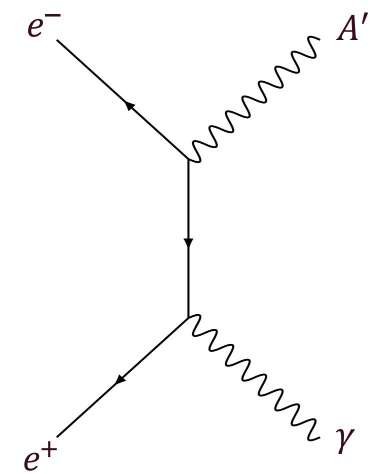


Elizabeth Long - Sapienza Università di Roma

The PADME Experiment

- The Positron Annihilation to Dark Matter Experiment searches for a dark photon A' in the process $e^+e^- \rightarrow A'\,\gamma$.
- The principle background is Bremsstahlung from positrons on target (in red in the diagram below).
- To detect positrons which have undergone Bremsstrahlung, we use a Positron Veto (PVeto) made of plastic scintillators in a magnetic field.



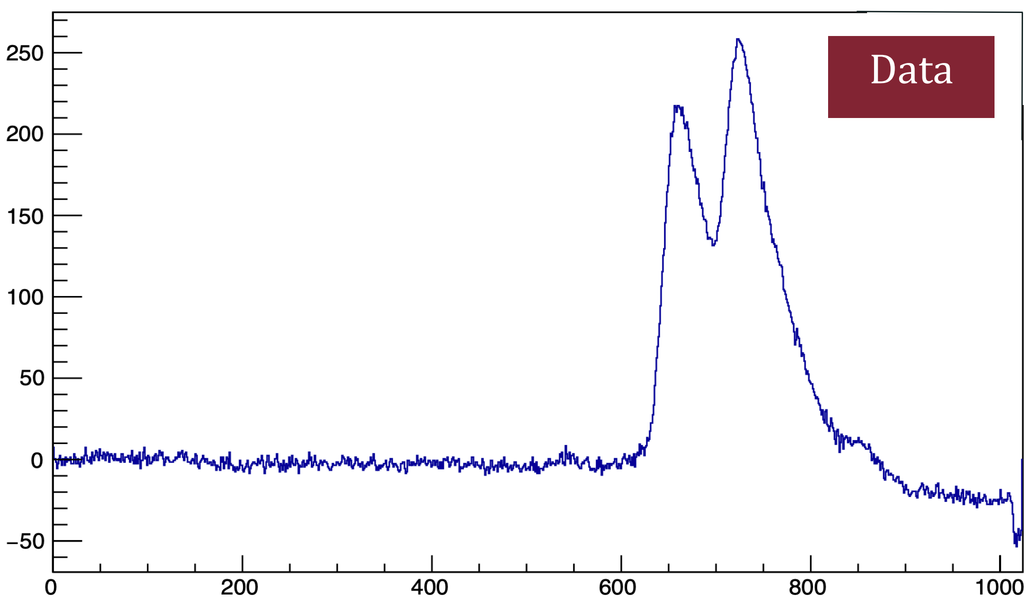


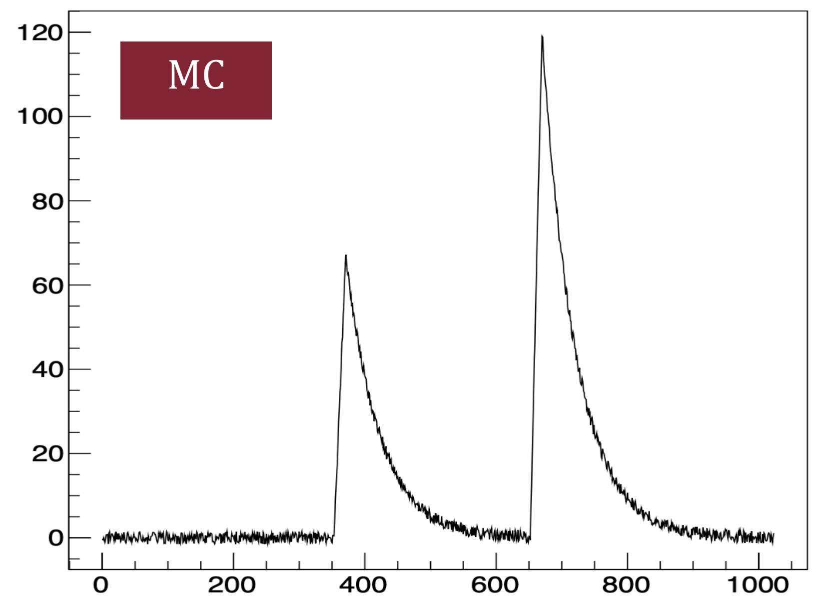
Studying Signals

- The high rate of particles entering the vetoes makes distinguishing between particles difficult.
- To study the performance of different approaches to separating particle hits, we used a toy MC that creates signals.

Veto Signals

- The simulated signals have a linear rise time of 7ns and exponential fall with τ of 20ns.
- The number of hits per event ~ Poisson(3), arrival time ~Uniform[80ns,280ns], amplitude ~ Landau(50,12)



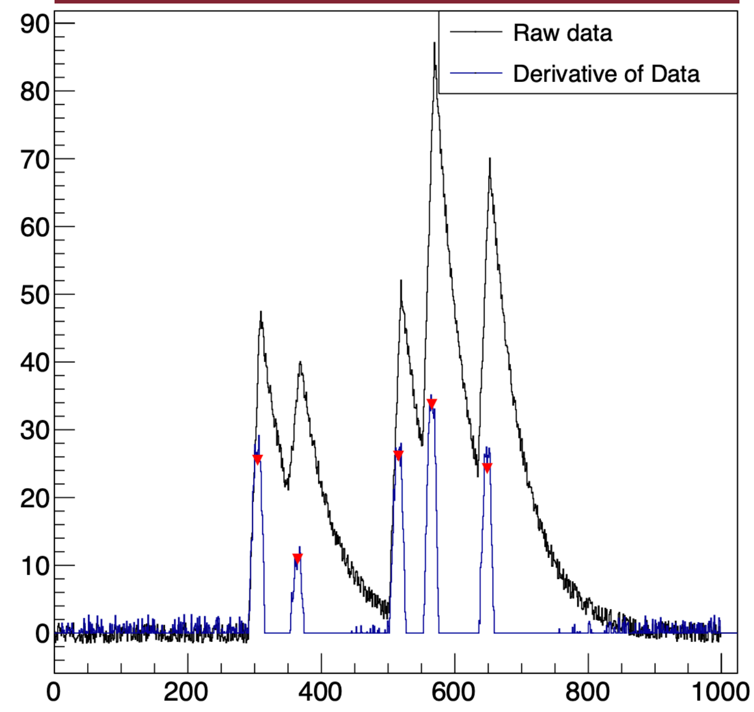


1024 digitizer samples = 400ns

Raw Data + Derivative + TSpectrum

- The signal derivative is approximated by $V_{Deriv}(t) = V(t) - V(t-4ns)$, where V($t$) is the height of the signal at time t (ns).
- If the derivative is negative, it is set to 0.
- The derivative is passed to TSpectrum - a built-in Gaussian peak finder in ROOT.
- TSpectrum returns the number and position of the peaks it finds.

Red markers indicate the position of peaks found by TSpectrum.



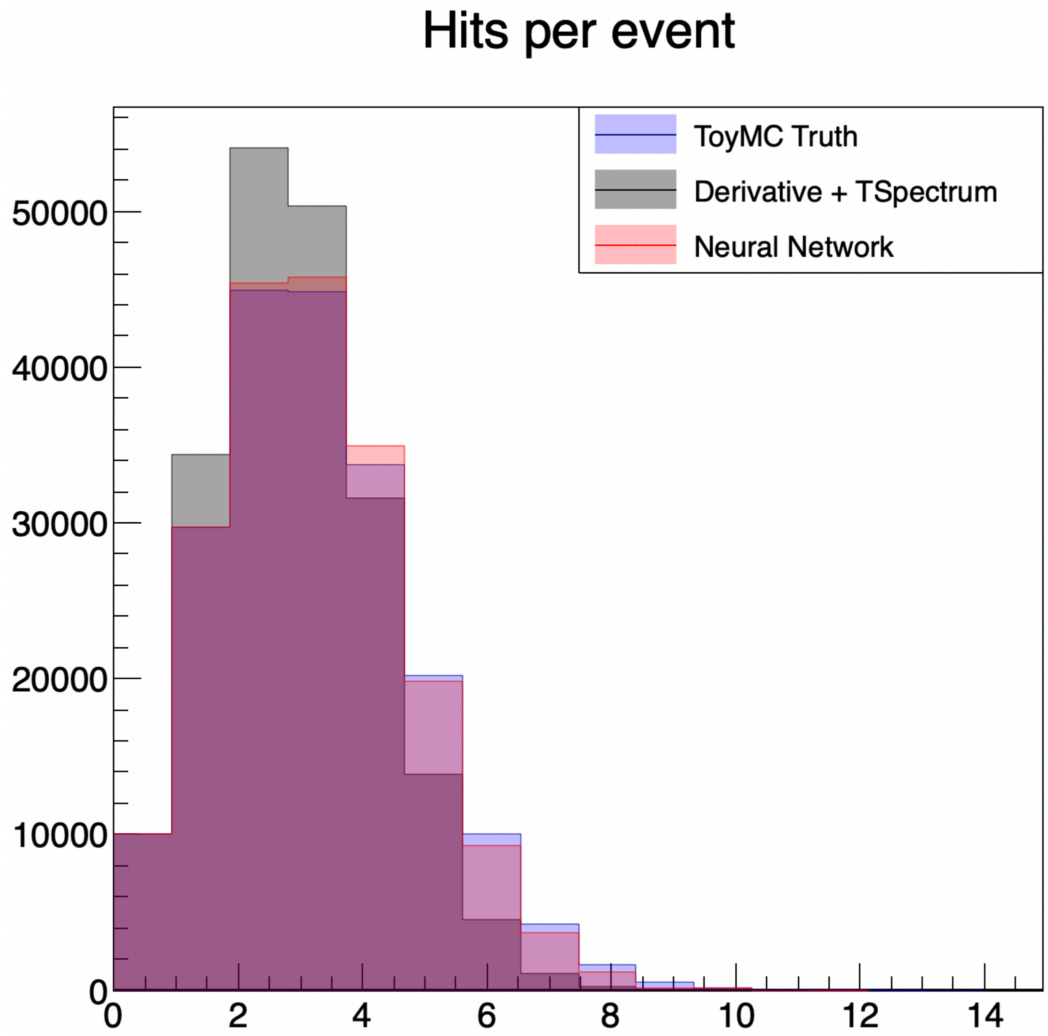
Convolutional Neural Network (CNN)

- A simple CNN was constructed to count the number of hits present in a waveform.
- Input: raw (unprocessed) waveforms
- Output: textfile of number of hits reconstructed per event
- Trained on 200,000 training and 100,000 validation events.
- Tested on 200,000 events.

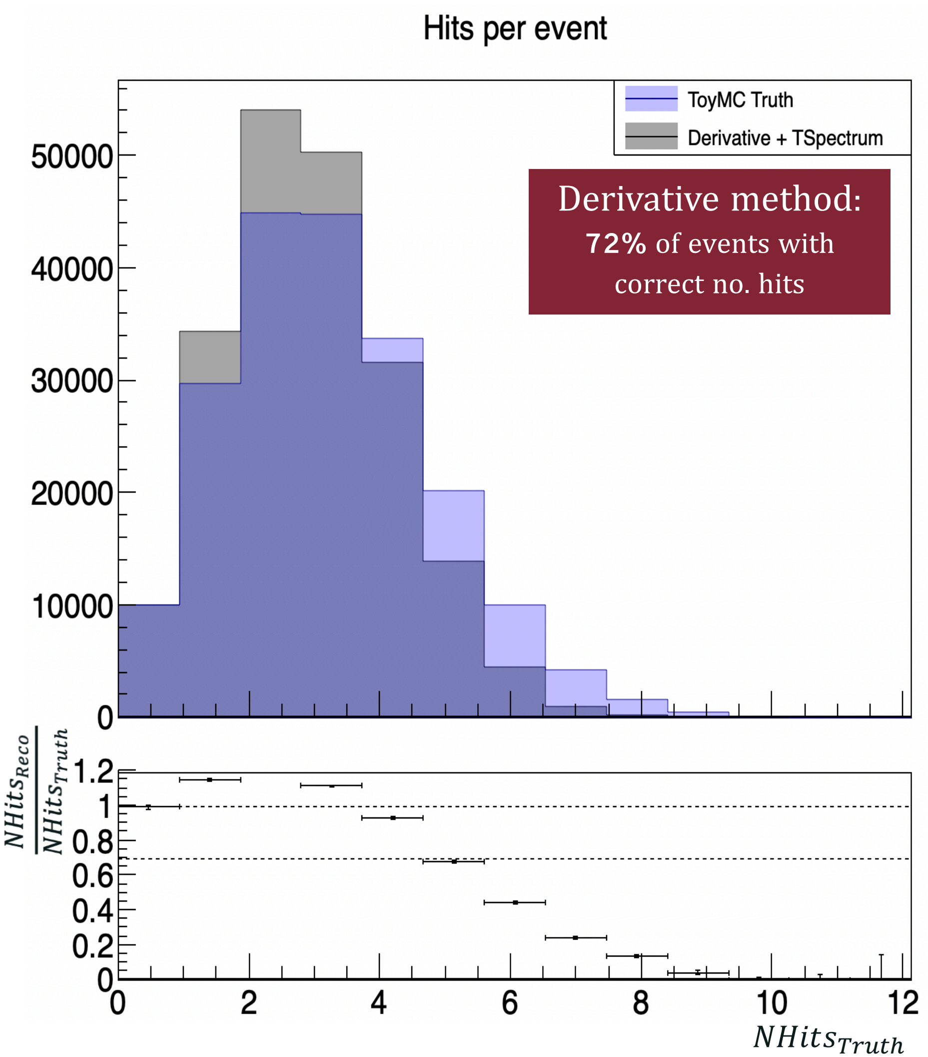
Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 1024, 1)]	0
Conv_1 (Conv1D)	(None, 1022, 16)	64
leaky_re_lu_6 (LeakyReLU)	(None, 1022, 16)	0
MaxPool_1 (MaxPooling1D)	(None, 511, 16)	0
Conv_2 (Conv1D)	(None, 509, 16)	784
leaky_re_lu_7 (LeakyReLU)	(None, 509, 16)	0
MaxPool_2 (MaxPooling1D)	(None, 254, 16)	0
Conv_3 (Conv1D)	(None, 252, 32)	1568
leaky_re_lu_8 (LeakyReLU)	(None, 252, 32)	0
MaxPool_3 (MaxPooling1D)	(None, 126, 32)	0
Flatten (Flatten)	(None, 4032)	0
Dense_1 (Dense)	(None, 64)	258112
ReLU_dense_1 (ReLU)	(None, 64)	0
dropout_2 (Dropout)	(None, 64)	0
Dense_2 (Dense)	(None, 64)	4160
ReLU_dense_2 (ReLU)	(None, 64)	0
Output (Dense)	(None, 14)	910

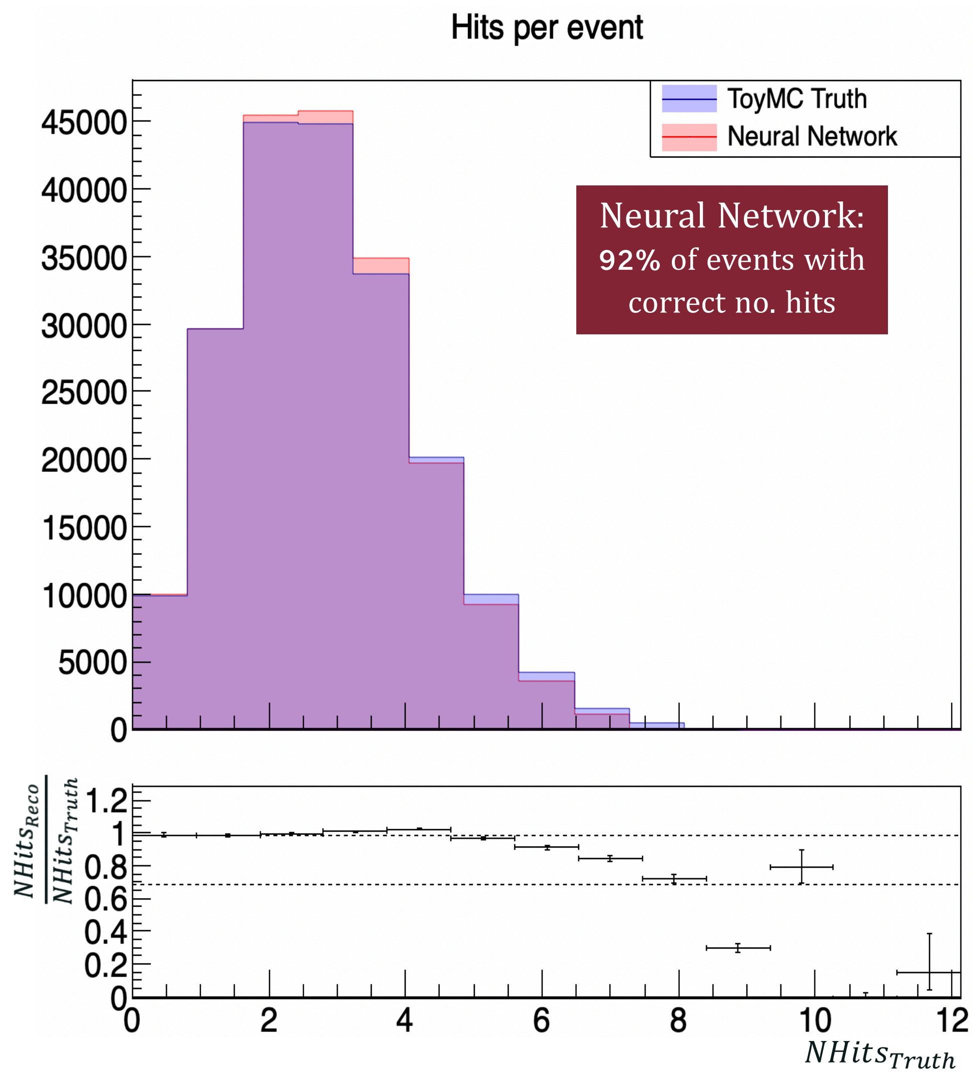
Total params: 265,598
Trainable params: 265,598
Non-trainable params: 0

Results: Number of Hits Per Event



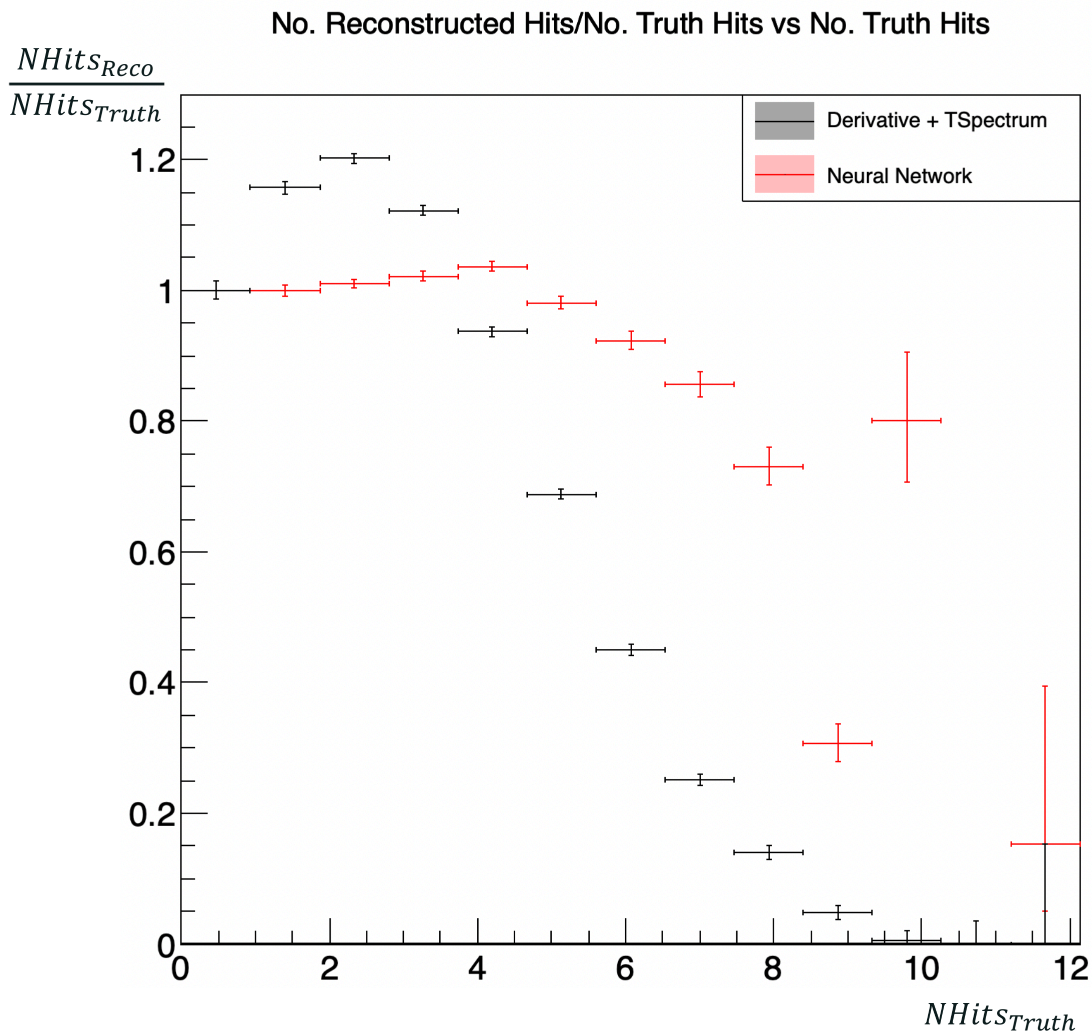
Results: Number of Hits Per Event



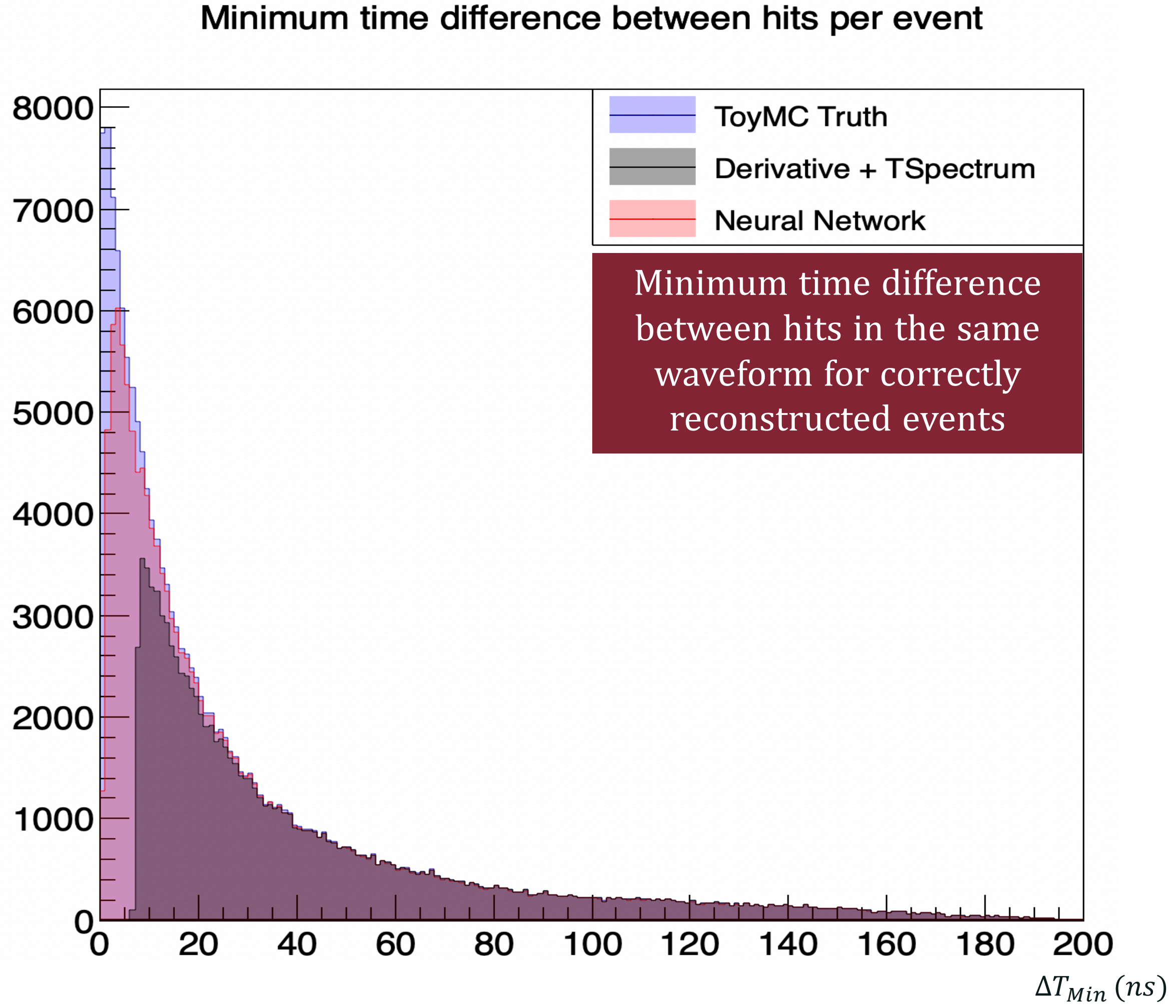


Results: Reconstruction Efficiency Ratios

- For the CNN, efficiency remains ~1 for $NHits_{True} \leq 5$
- Using the derivative + TSpectrum:
  - Number of events with $NHits_{True} \leq 3$ is overestimated
  - Number of events with $NHits_{True} > 3$ is underestimated

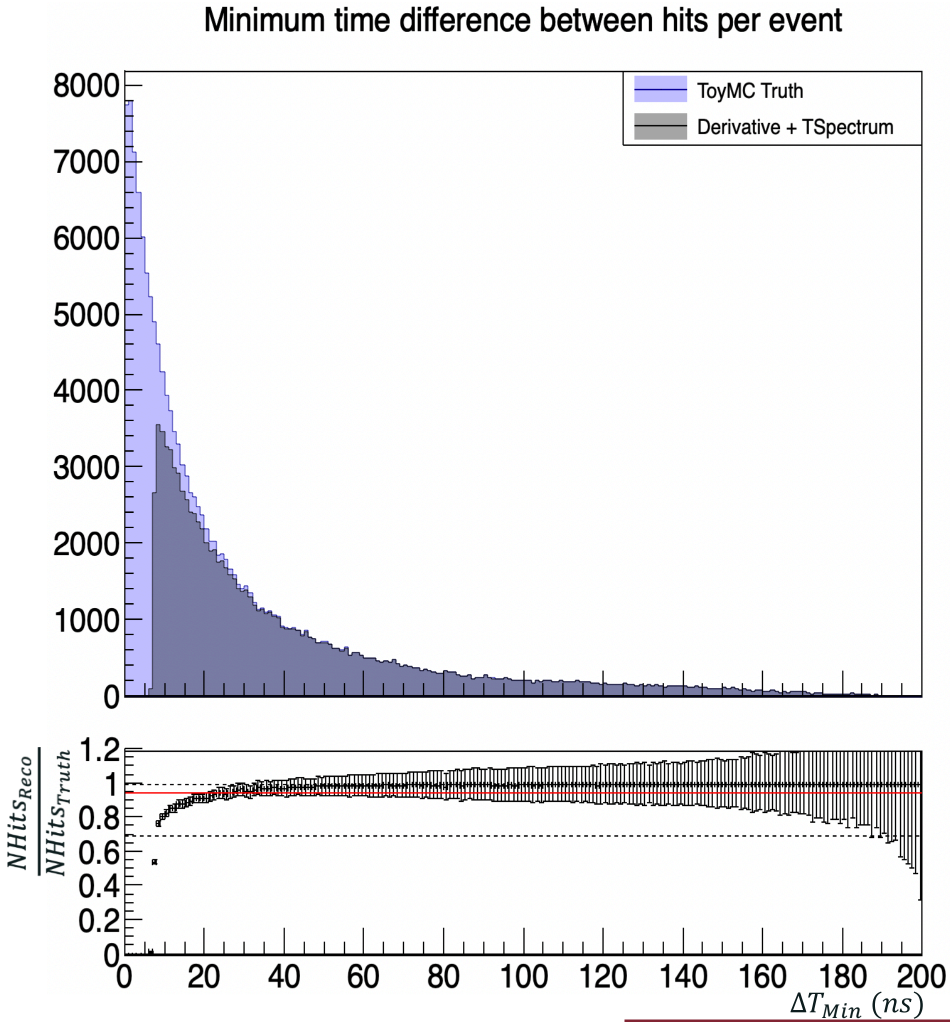


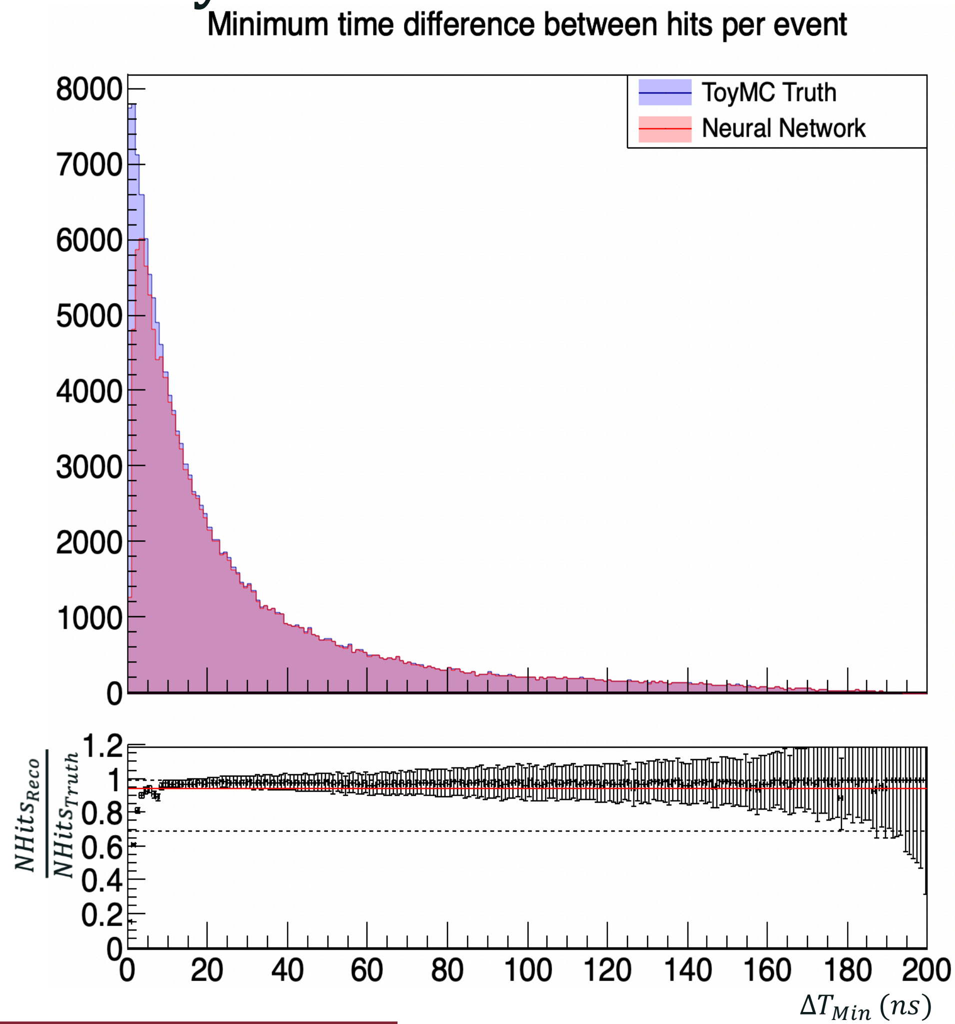
Results: Reconstruction Efficiency vs Time



Minimum time difference between hits in the same waveform for correctly reconstructed events

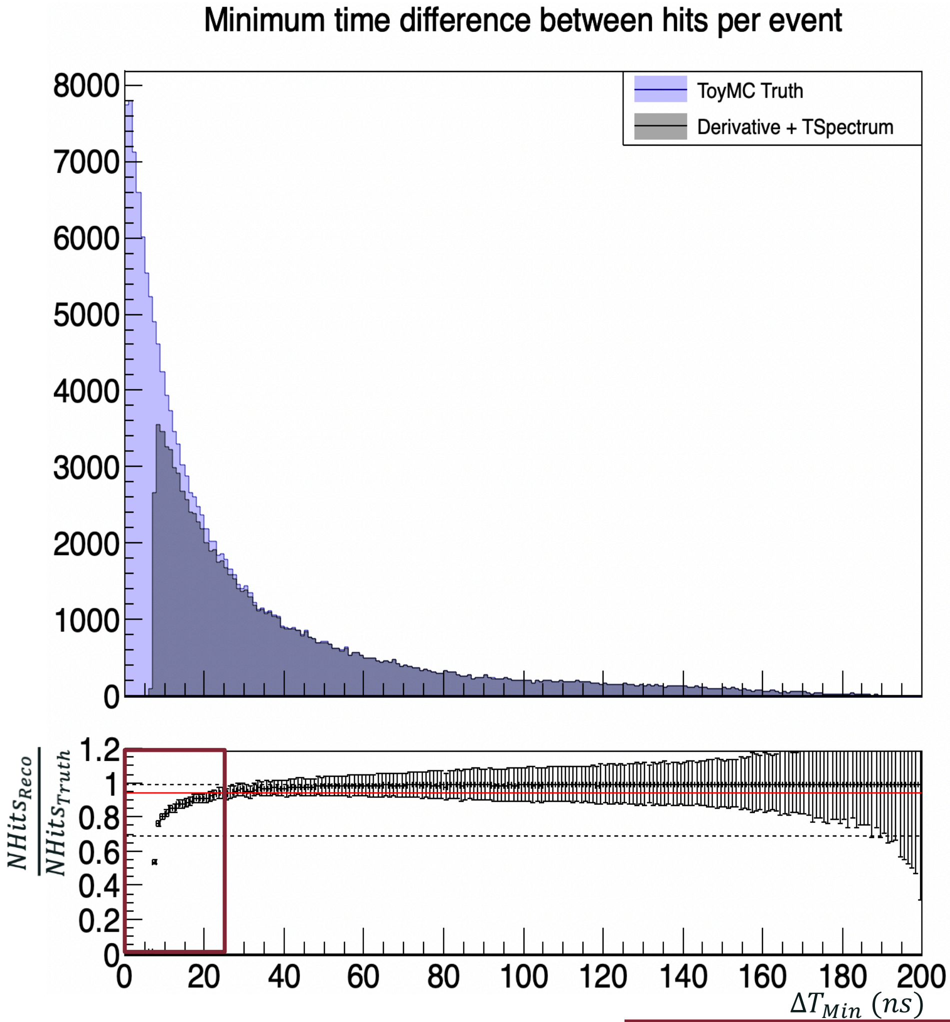
Results: Reconstruction Efficiency vs Time

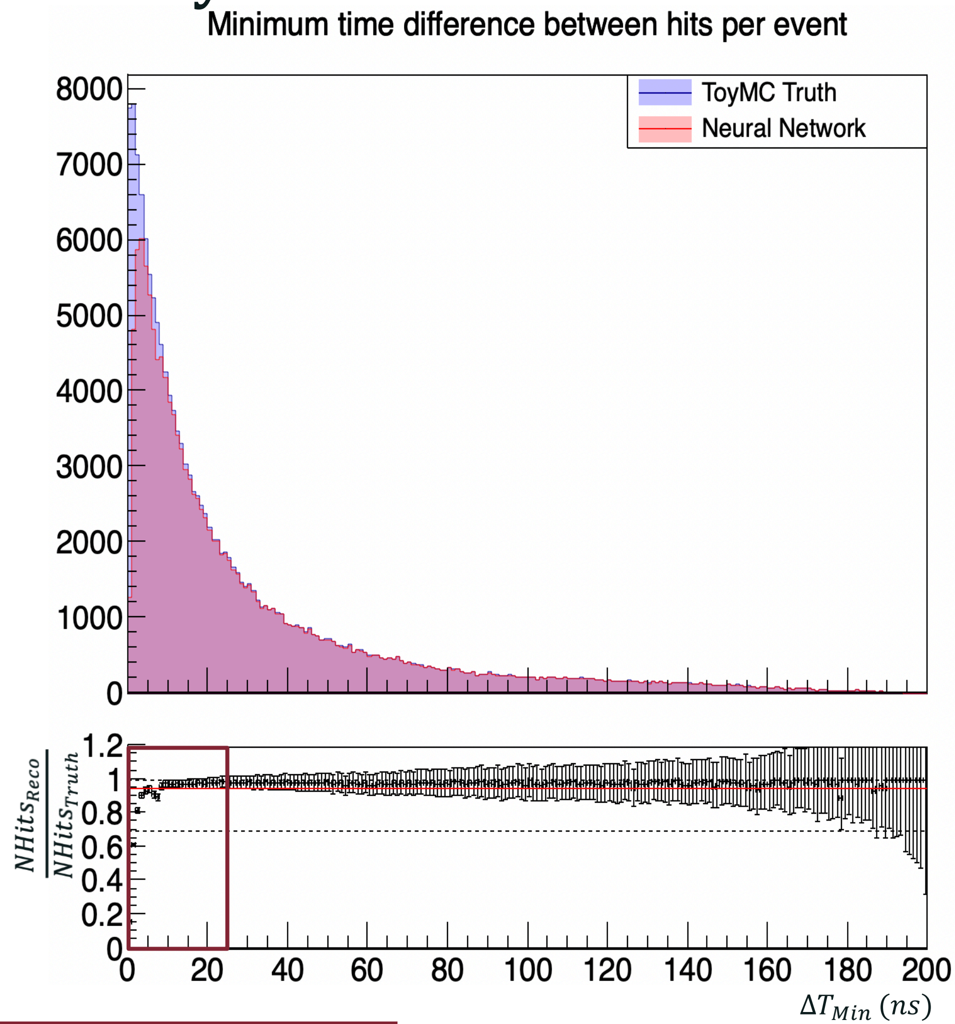




Red line (95%) = separability threshold:
Minimum ΔT at which hits are considered separable

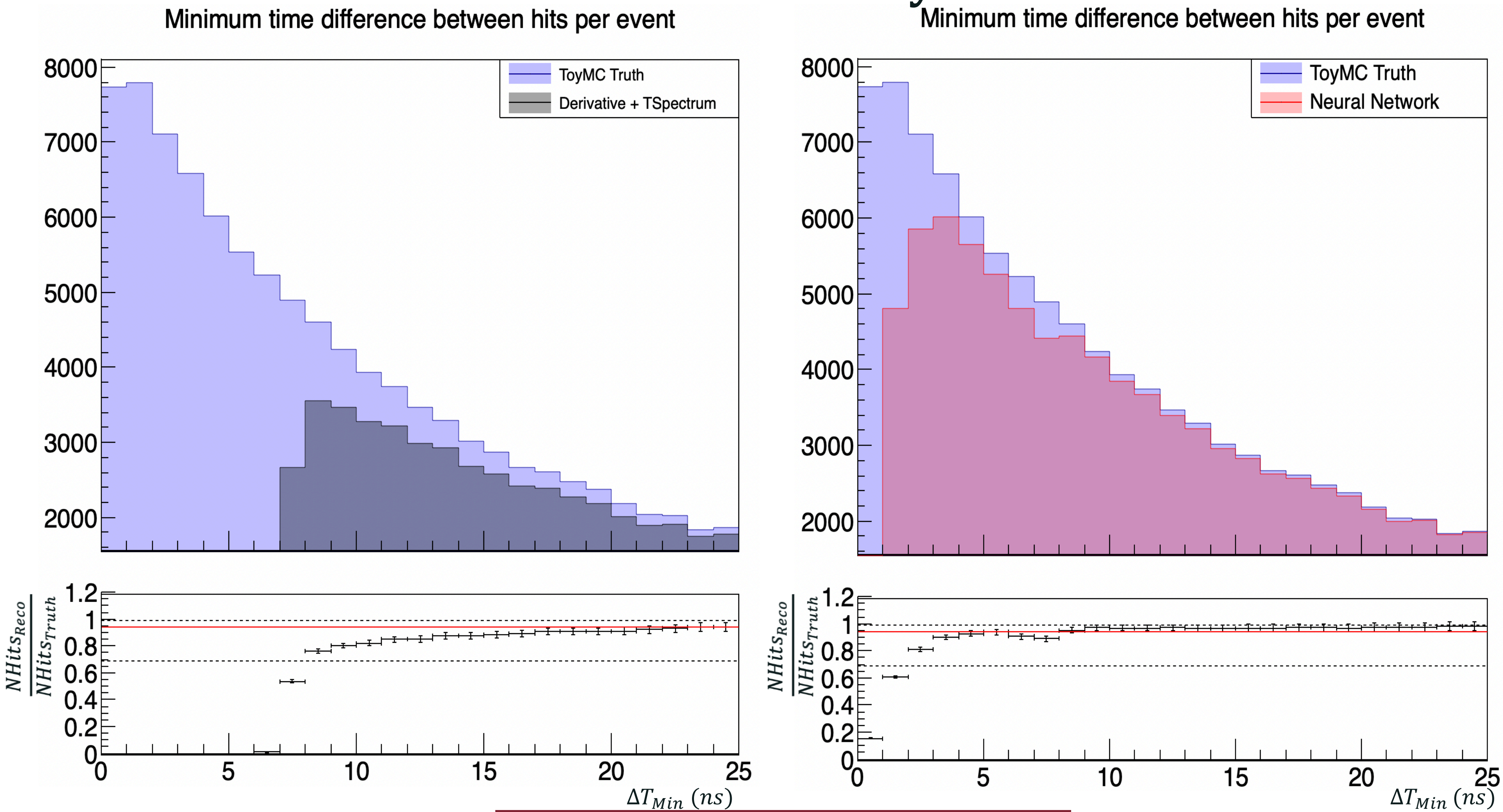
Results: Reconstruction Efficiency vs Time





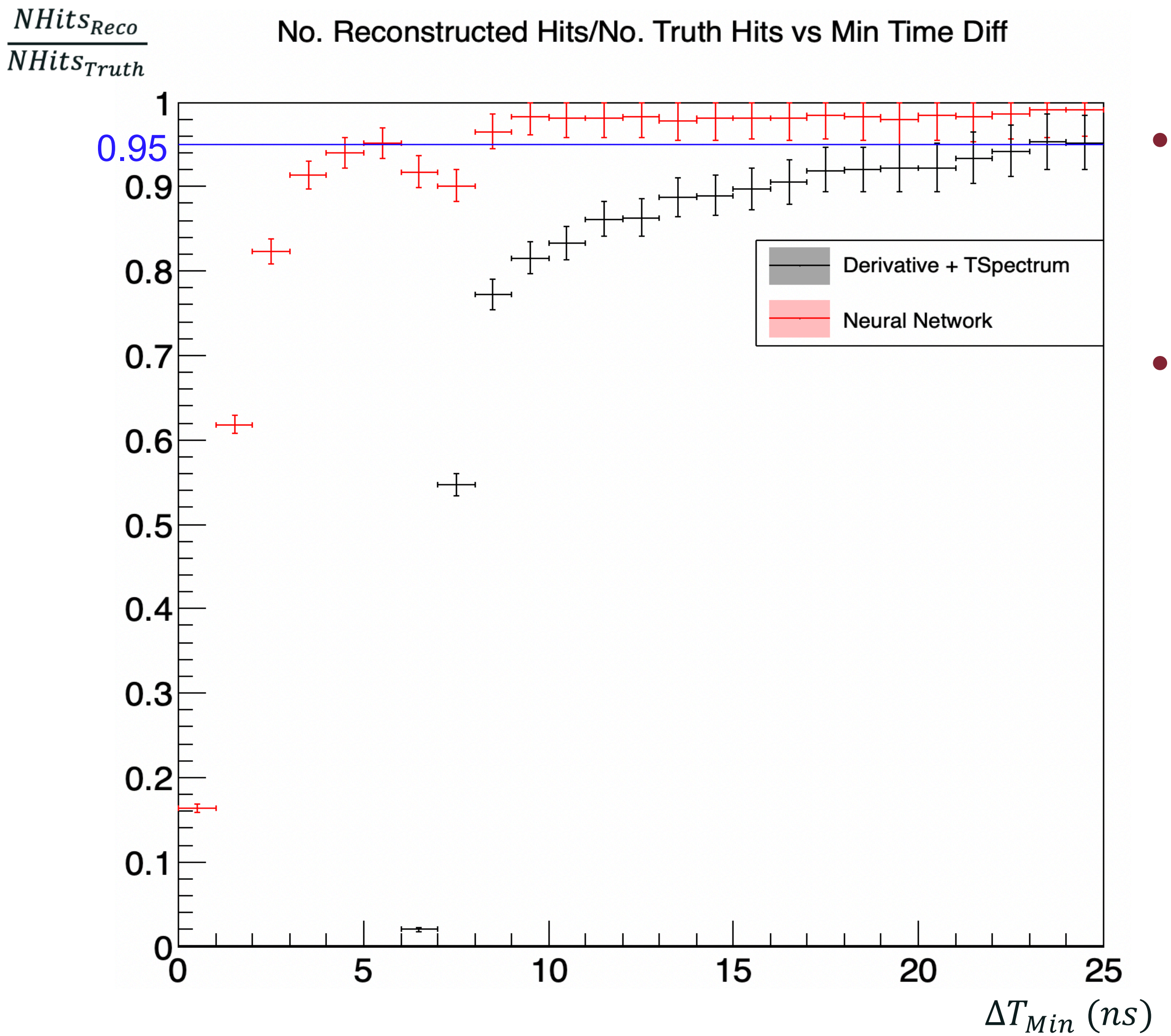
Red line (95%) = separability threshold:
Minimum ΔT at which hits are considered separable

Results: Reconstruction Efficiency vs Time



Red line (95%) = separability threshold:
Minimum ΔT at which hits are considered separable

Results: Reconstruction Efficiency Ratios



- The CNN reaches separability (95% efficiency) at 8ns
- With the derivative + TSpectrum separability is reached at ~20ns

Summary

- The global efficiency of the CNN at reconstructing the number of hits per event is 92%, while the derivative + TSpectrum method is only 72% efficient.
- 95% efficiency in hit separation is reached at:
  - $\Delta T_{Min} \geq 8$ns for CNN
  - $\Delta T_{Min} \geq 20$ns for derivative + TSpectrum
- This shows that using a CNN is a good way of reconstructing the number of hits that have produced a simulated waveform.
- Next steps are to study how well this works on data and how to recover hit times.

Thank you for your attention.