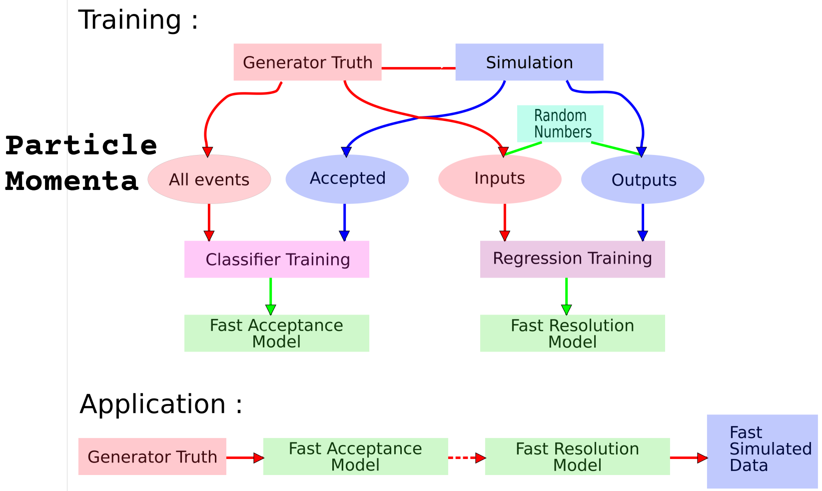
Machine Learned Simulations (Derek Glazier, Glasgow)

Fast Simulation Scheme



Classifier Training

Create acceptance map for each particle species

Deep Neural Net with BoostedDecisionTree corrections

$A(x) = P(x)/(1-P(x))$

with P(x)=classifier output

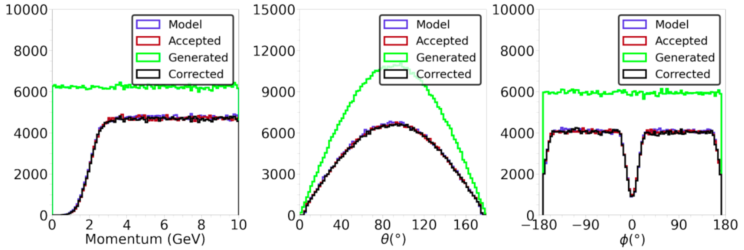


Figure 11: Results of applying a neural network with a Gaussian transform for acceptance modelling with a BDT correction. The BDT used 100 weak learners with a maximum depth of 10 and a learning rate of 0.1. The network used is the higher capacity model with 4 hidden layers of 512, 256, 128, and 16 neurons respectively. The improvement in the 3-vector component distributions is smaller than in the case of the low capacity network.

Regression Training

Create resolution simulator

Decision Tree/kNN with randomisation

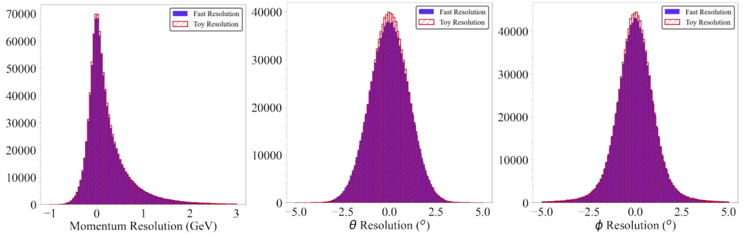
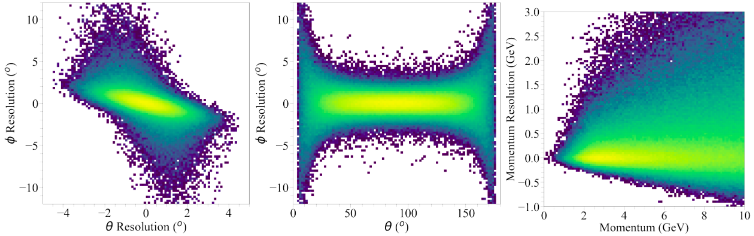


Figure 20: The Fast (blue) and Toy (red) momentum, θ, ϕ resolutions.



macparticles

Machine Learned Particle Simulations

Authors : Richard Tyson, Derek Glazier, Darius Darulis;

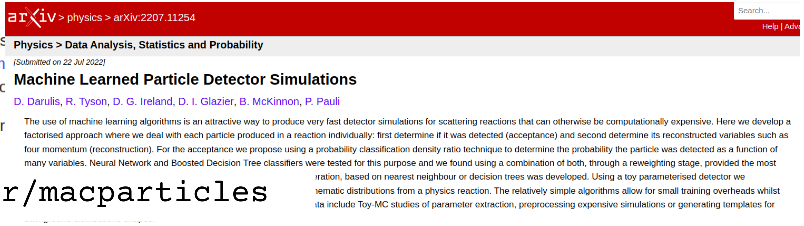
Affliation : Nuclear and Hadron Physics, School of Physics and Astronomy, University of Glasgow

Aim: Produce very fast simulation results via Neural Networks and Decision Trees.

Users supply their own truth and reconstructed events from full simulations to train the machine learning algorithms to produce replicable outputs as a fast alternative for full simulations.

This interface is based around CERN ROOT libraries and interactive sess
running training and predictions in the python ecosystem; the RDataFram
between ROOT and Python sessions, while providing fast filtering and plo

We provide some C++ classes for automating the training and Python scr
using tensorflow, keras and scikit-learn.



arXiv > physics > arXiv:2207.11254

Search...

Help | Adva

Physics > Data Analysis, Statistics and Probability

[Submitted on 22 Jul 2022]

Machine Learned Particle Detector Simulations

D. Darulis, R. Tyson, D. G. Ireland, D. I. Glazier, B. McKinnon, P. Pauli

The use of machine learning algorithms is an attractive way to produce very fast detector simulations for scattering reactions that can otherwise be computationally expensive. Here we develop a factorised approach where we deal with each particle produced in a reaction individually: first determine if it was detected (acceptance) and second determine its reconstructed variables such as four momentum (reconstruction). For the acceptance we propose using a probability classification density ratio technique to determine the probability the particle was detected as a function of many variables. Neural Network and Boosted Decision Tree classifiers were tested for this purpose and we found using a combination of both, through a reweighting stage, provided the most
...eration, based on nearest neighbour or decision trees was developed. Using a toy parameterised detector we
...nematic distributions from a physics reaction. The relatively simple algorithms allow for small training overheads whilst
...ata include Toy-MC studies of parameter extraction, preprocessing expensive simulations or generating templates for

https://github.com/dglazier/macparticles

Results/Status

CLAS12 GEMC $ep \rightarrow e'\pi^+K^+K^-n$

Toy Detector $\gamma p \rightarrow \pi^+\pi^-p$

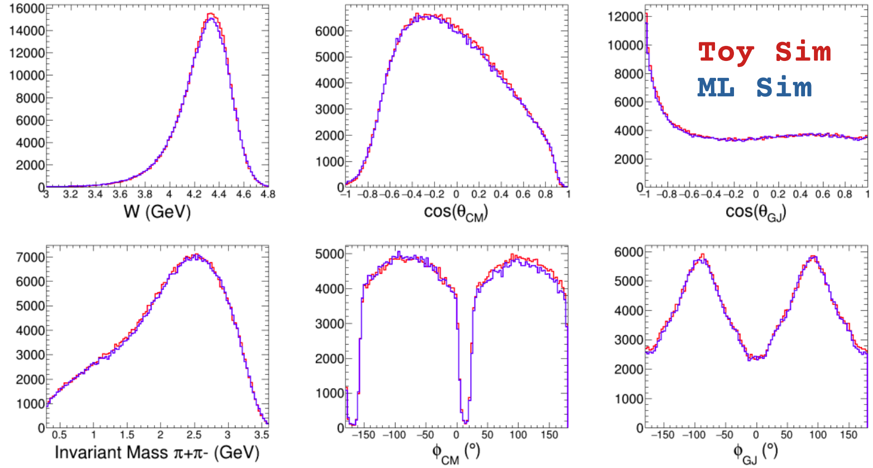
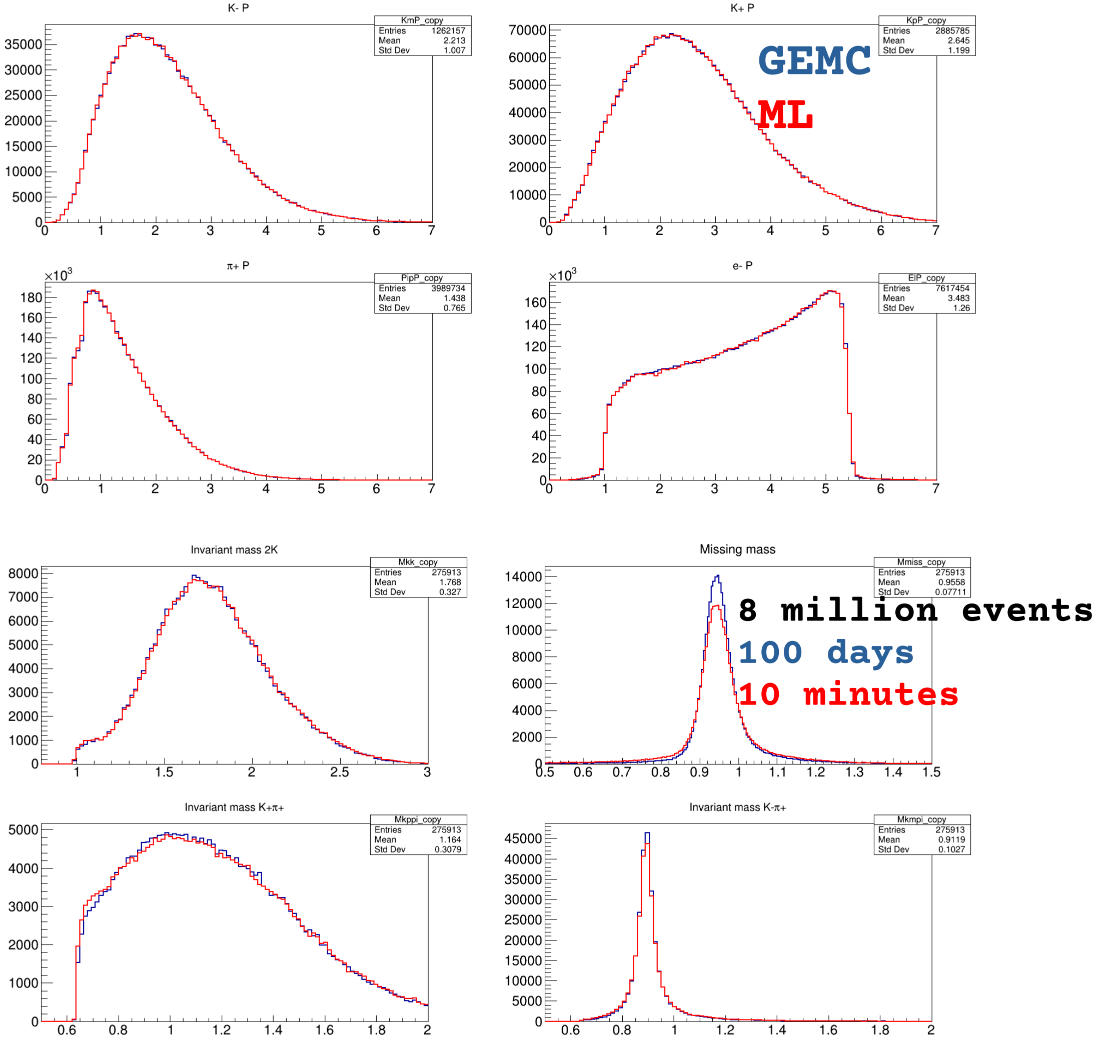


Figure 26: Accepted and reconstructed physics variables for the Fast (blue) and Toy (red) simulations of the 2 pion photoproduction reaction. The distributions show: the invariant mass of the three final state particles, W; the invariant mass of the two pions, $M(2\pi)$; the production angles in the centre-of-mass system ($\cos(\theta_{CM}), \phi_{CM}$); and the decay angles of the two pions.



Approximate ML simulations are ideal for Physics scoping, background simulations,…

Requires full single particle simulations
ML Training is simple and fast

Full reaction simulations may require corrections due to correlations (trigger, PID…)

Scope for switching resolutions to GANs