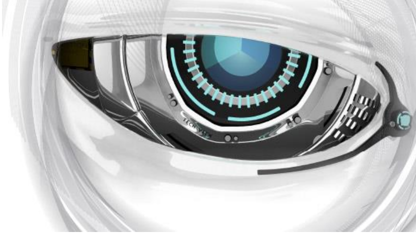
Distributed Training of Generative Adversarial Networks for Fast Simulation

HPC and AI in High Energy Physics

G. Khattak, F. Carminati, S. Vallecorsa, D. Podareanu, V. Codreanu, V. Saletore, H. Pabst
S. Choi, M. Cai
24/9/2019

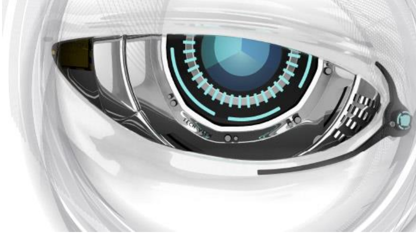


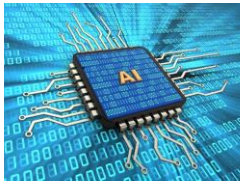
Overview

- Introduction
  - HPC and AI in HEP
  - Fast simulation
  - 3DGAN
- Distributed Training
  - Distributed training initial optimization
  - Scaling up to 256 nodes
  - Inference time
- Summary

HPC and AI

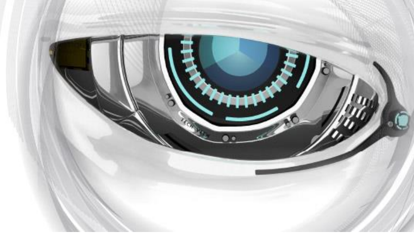
Main driving forces………...





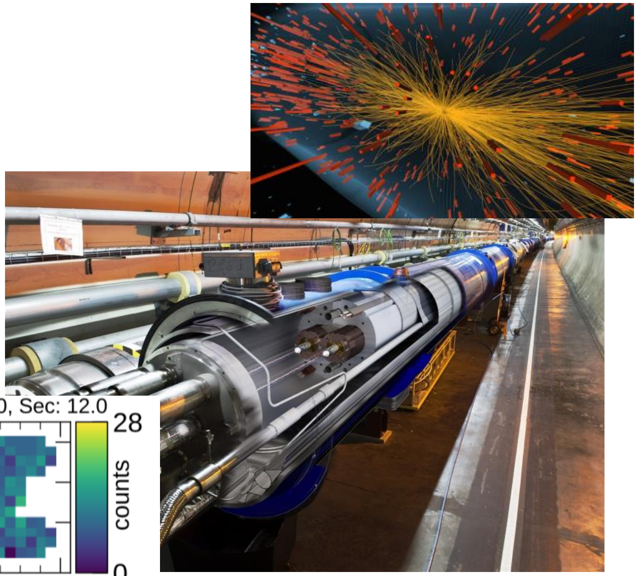
- Deeper models
  - Deep Neural Networks often have millions of parameters
- Big data
  - More complex problems require more data
- Faster
  - Training speedup
  - Inference speedup
- Parallelizable processes
  - Parallelism can be implemented at different levels

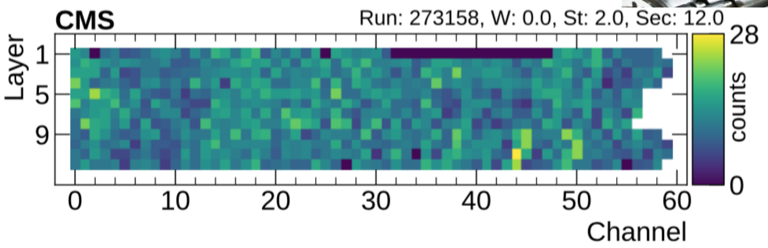
High Energy Physics



AI applications in HEP

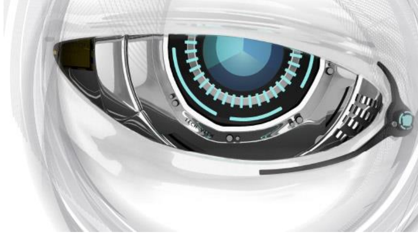
- All venues of science are benefitting from AI for problems where..
    - Underlying processes are difficult to model
    - Require high computational sources
    - Time consuming
    - Noisy data
- High Energy Physics
    - Applications
        - Reconstruction and Analysis
        - Trigger optimization
        - Simulation
- AI crucial for HEP experiments
    - HPC hardware
        - Maximize performance
        - Fast time-to-model





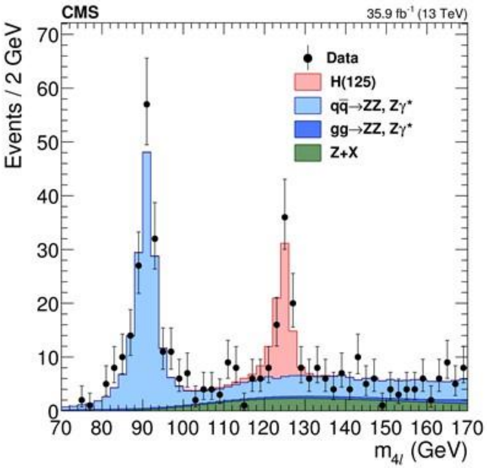
CERN openlab

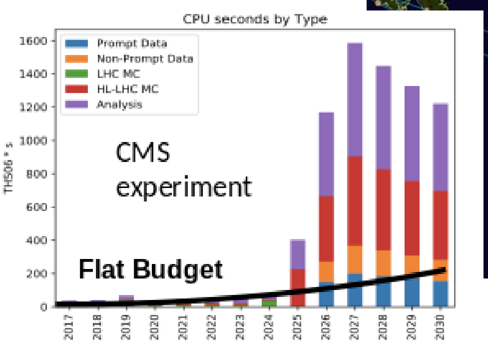
HEP Simulation

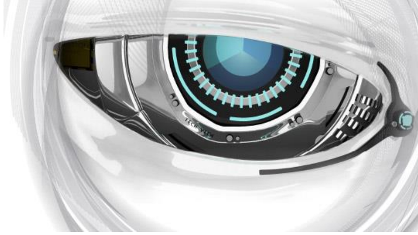


Essential for data analysis & detector design

- Understand how detector design affects measurements and physics
  - Correct for inefficiencies, inaccuracies, unknowns
  - Compare theory models to data
- Complex physics and geometry modeling
  - >50% of Worldwide LHC Computing Grid (WLCG) power today
  - Increase by 100x by 2025!



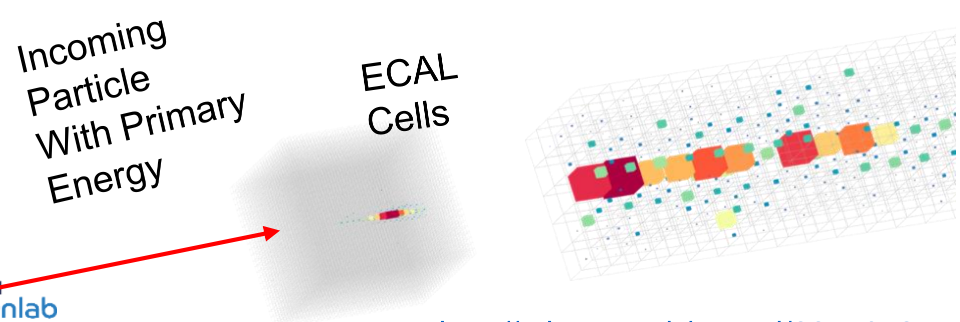


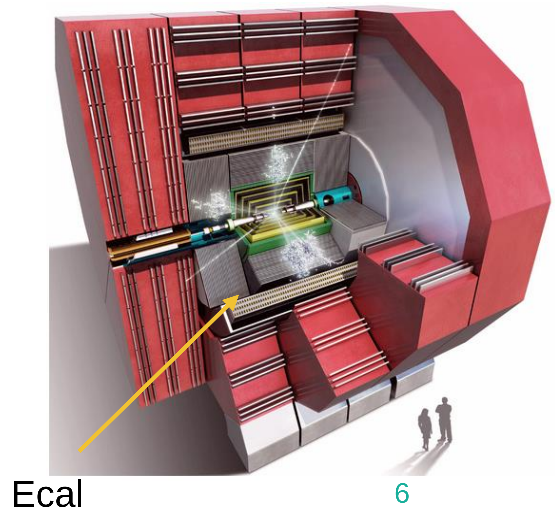


Data set

Compact Linear Collider CLIC

- **Proposed** linear particle accelerator
- **Calorimeter data set** developed for ML applications
- Events as selected cells around the barycenter of particle showers simulated using Geant4
- Primary particle energy 10-500 GeV (electrons)
  - Event → 25 x 25 x 25 image → 15, 625 cells
    - 200,000 events
- Detector response as 3D images
  - Highly segmented (pixelized)
    - critical for particle identification and energy determination
  - Highly sparse
    - only ~20% cells with energy deposition
  - Large dynamic range
    - > seven orders of magnitude

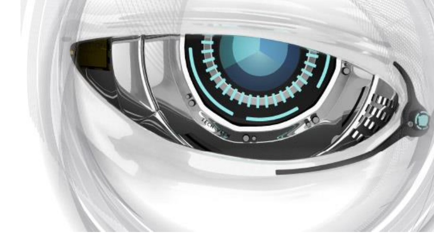




http://cds.cern.ch/record/2254048

http://clicdp.web.cern.ch/

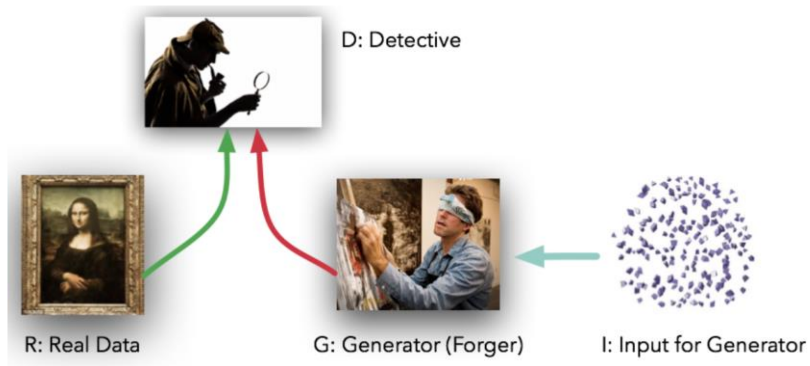
CERN openlab

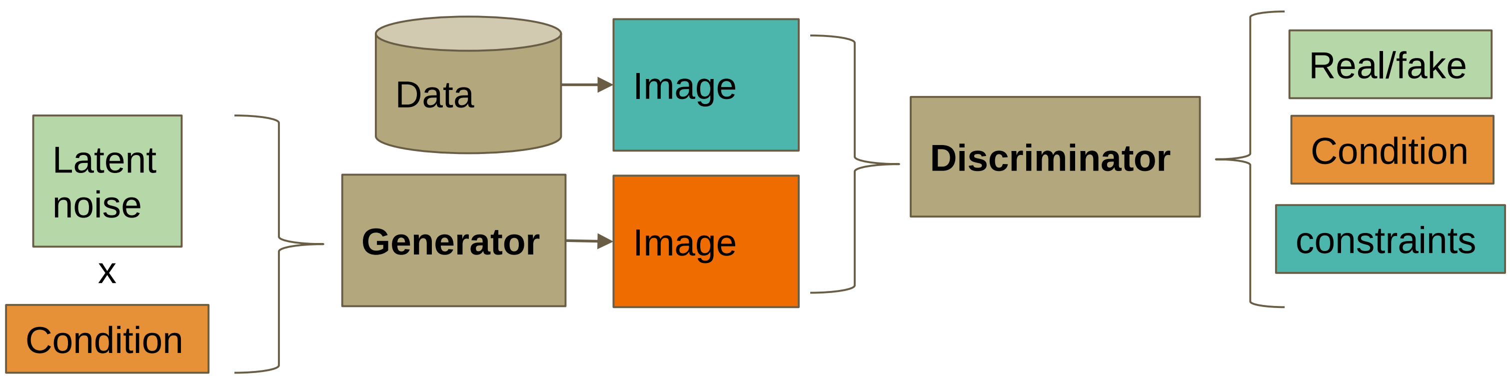


3DGAN

Generative Adversarial Network

- Simultaneously train two networks that compete and cooperate with each other
  - Discriminator
  - Generator

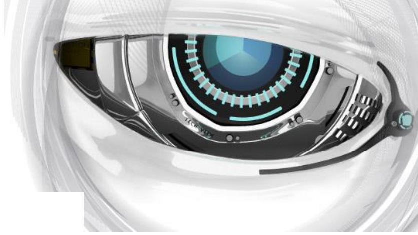


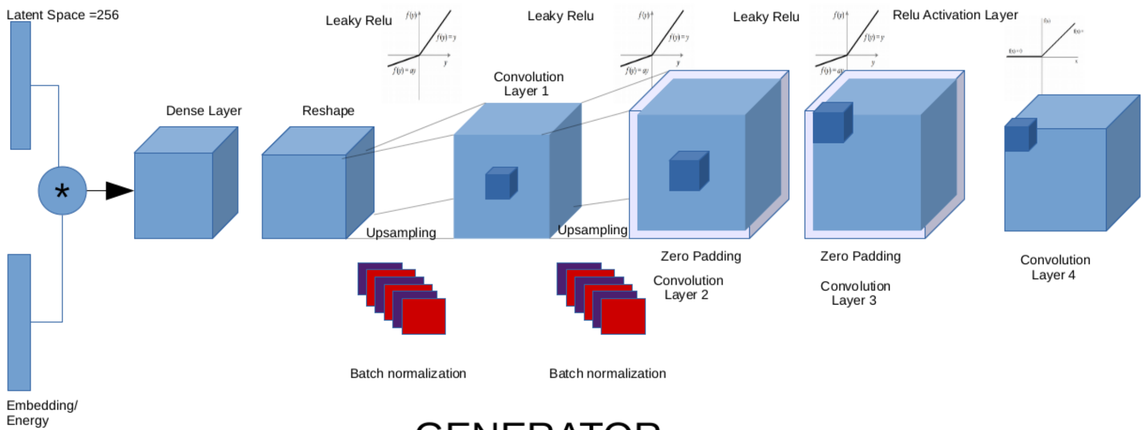


A generalized view of 3DGAN

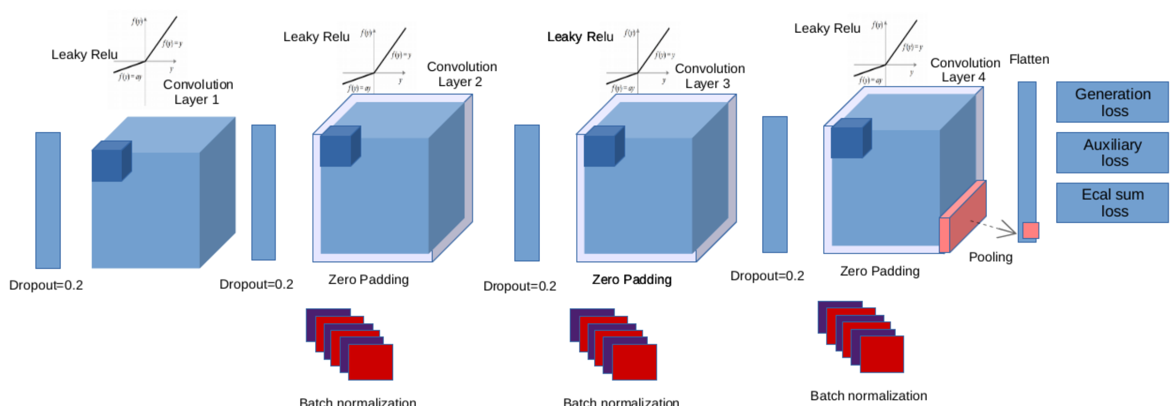
CERN openlab

3DGAN Architecture





GENERATOR



DISCRIMINATOR

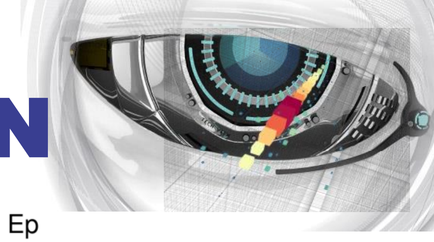
~1 M parameters
Total model Size 3.8MB

Evaluating the performance by agreement to labels and Physics related constraints

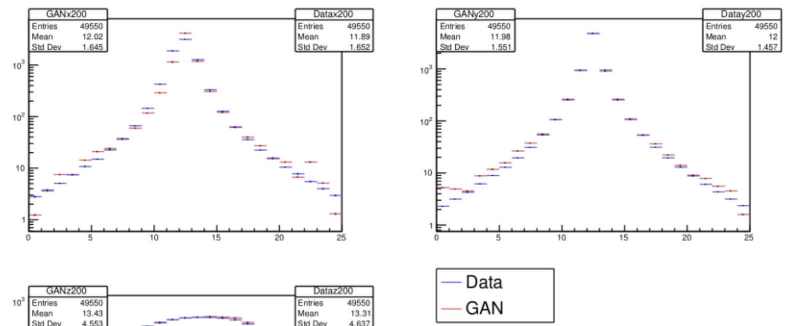
"Caltech ibanks GPU cluster thanks to Prof M. Spiropulu"

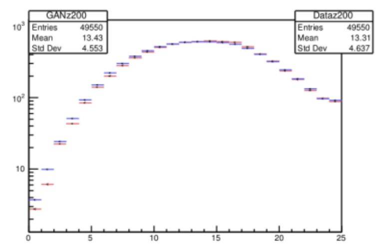
CERN openlab

Physics Simulation with 3DGAN

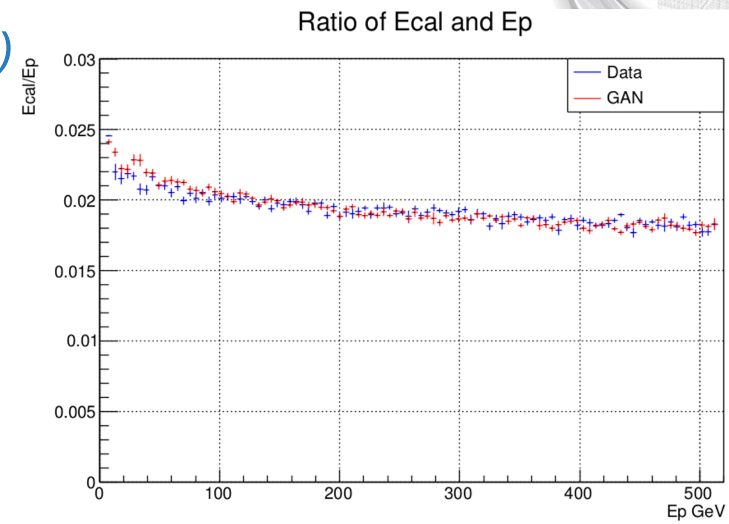


Comparison to Monte Carlo (>300 plots)

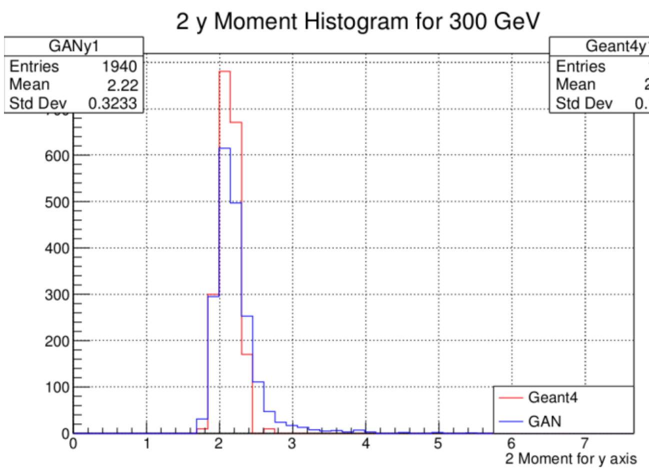




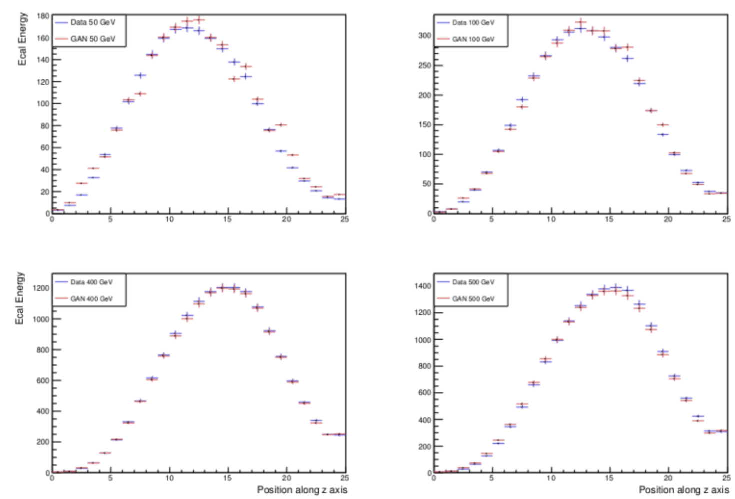
Energy Deposition along x, y, z axis for 200 Gev



Sampling Fraction (Ecal sum/Ep)

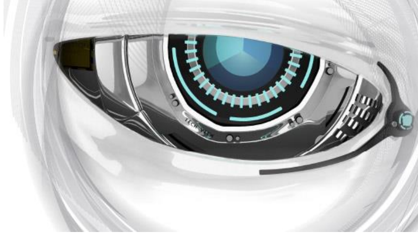


Y moment (width)



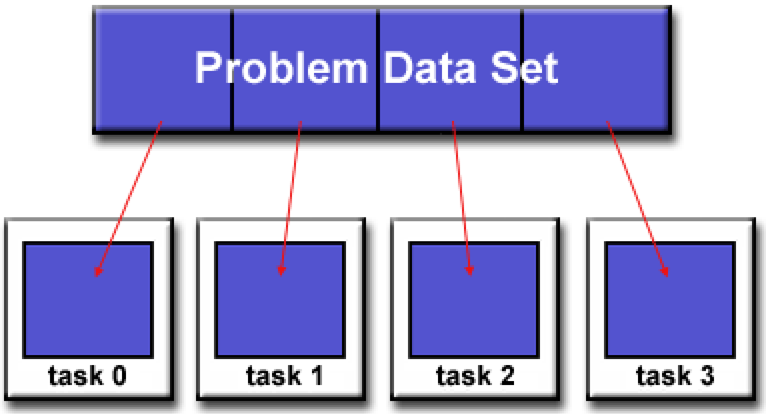
Energy deposition in transverse direction for 50, 100, 300 and 400 GeV

Distributed training



3DGAN

- Training time ~ **1hour/epoch** on GeForce GTX 1080
- 30 to 50 epochs for complete training taking **days**
- Reducing training time is essential for:
  - Hyper parameter scans
  - Detector design studies
- Distributed training with Horovod
  - Data parallelism
  - Synchronous update

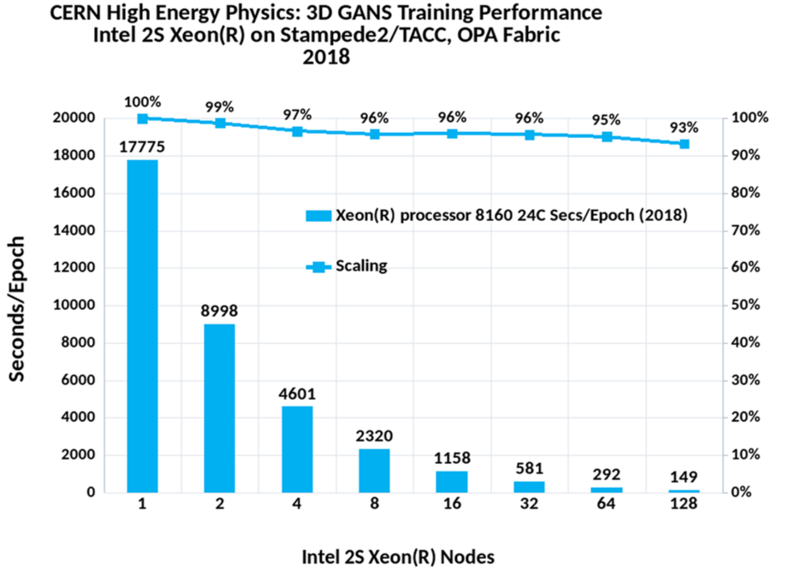


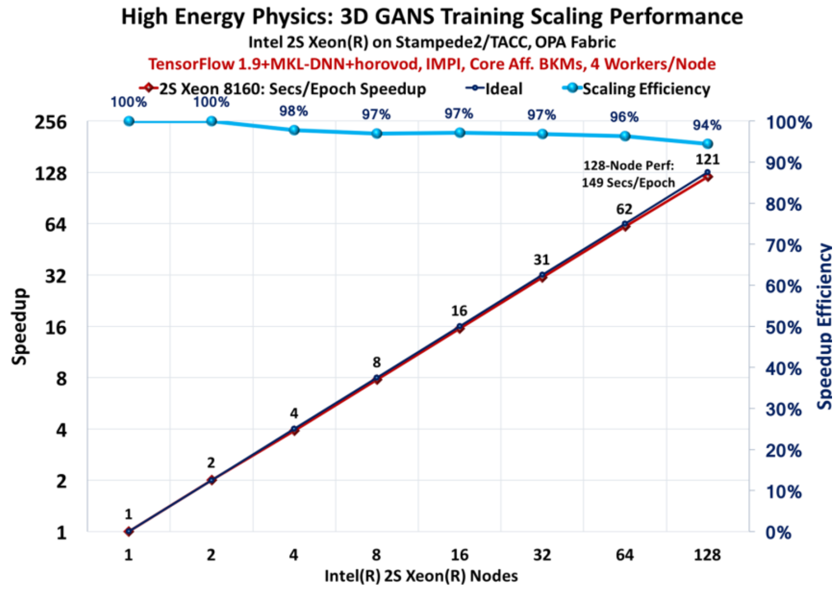
Distributed Training initial optimization

TACC Stampede 2 (2018)

- Stampede 2 cluster
  - Dual socket Intel® Xeon® **8160**
  - 2x 24 cores per node, 192 GB RAM
  - Intel® Omni-Path Architecture
- Software
  - **Tensorflow 1.9** (Intel optimized)
  - Keras 2.13
  - Horovod 0.13.4
- Single Node Optimization:
  - Replace Eigen with **MKL-DNN**
  - Optimize number of **convolution filters**
- Parallelize:
  - 4 workers/node

IXPUG 2018



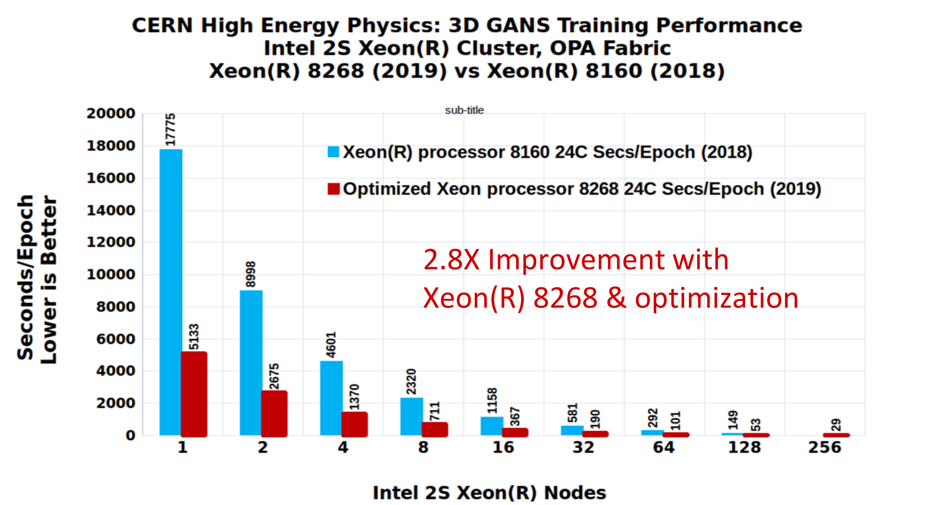


CERN openlab

Scaling up to 256 nodes

Xeon 8268 (2019)

- Intel Endeavour cluster:
  - **NASA** Advanced Supercomputing Division (NAD)
  - Named after spaceship Endeavour
  - Xeon® **8268 Cascade Lake**
  - 2 Sockets /node
  - 24 cores per socket
  - Intel® Omni-Path Architecture
- Software:
  - **Tensorflow 1.14** (Intel optimized)
  - MKL-DNN 0.18
  - Horovod 0.16.4
  - Keras 2.2.4



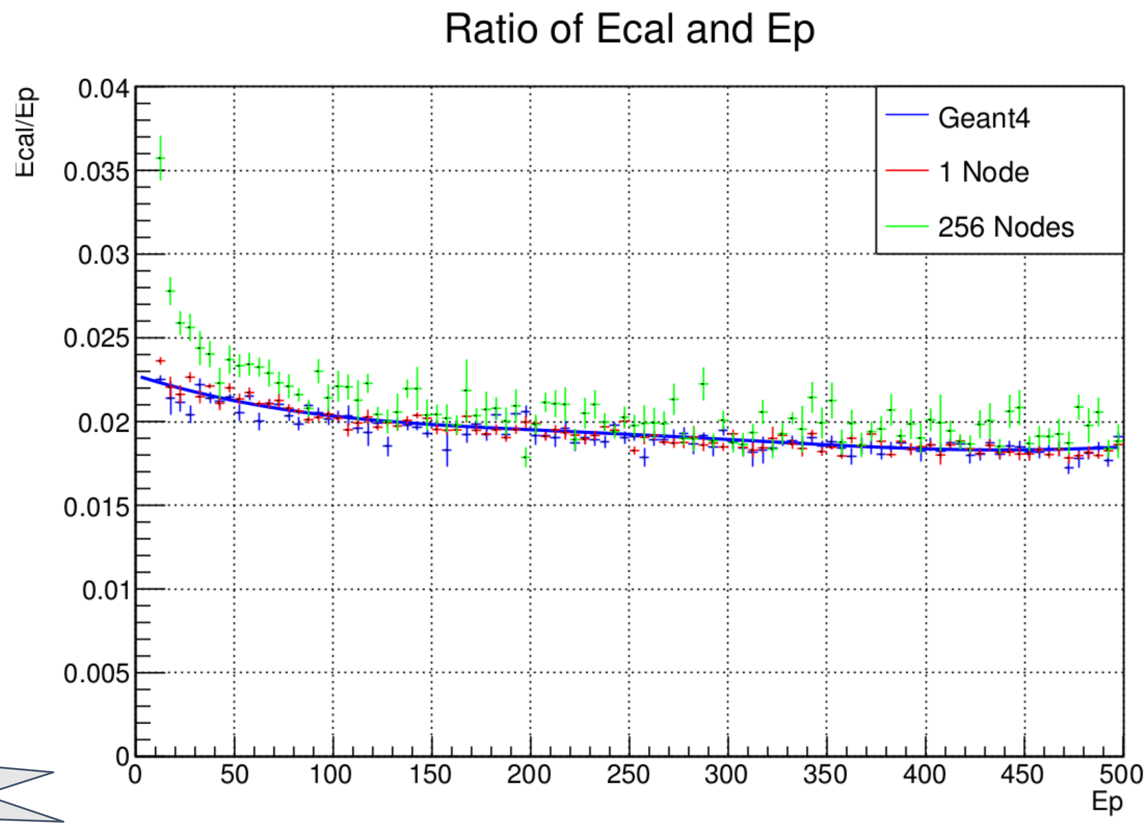
- For 128 2CPU Xeon Nodes
  - 2018: < **2.5 Mins**/Epoch Xeon 8160 (Skylake CPUs)
  - 2019: < **1 Min**/Epoch Xeon 8268 (Cascade Lake CPUs) – **2.5X**
    - Time to Train to Accuracy: **14.4 minutes** on 256 Nodes

2019

CERN openlab

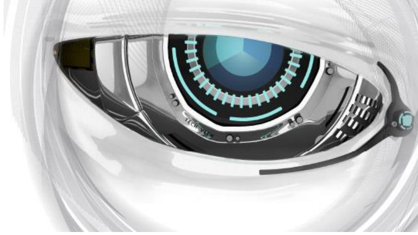
Physics Performance

Sampling Fraction



2019

Inference time



Tensorflow 1.9

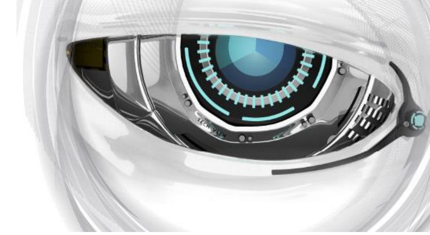
Method	Platform	Time/Shower (ms)	Speedup
Classical Monte Carlo (Geant4)	2S Intel Xeon Platinum 8180	17000	1.0
3DGAN (BS=128) 1-stream		16	2500

Baseline (TF 1.4)

Method	Platform	Time/Shower (ms)	Speedup
Classical Monte Carlo (Geant4)	2S Intel Xeon Platinum 8180	17000	1.0
3DGAN (BS=128) 1-stream	2S Intel Xeon Platinum 8160	1.25	13600
3DGAN (BS=128) 2-stream		0.93	18279
3DGAN (BS=128) 4-stream		0.85	20000

TF 1.9 (optimized)

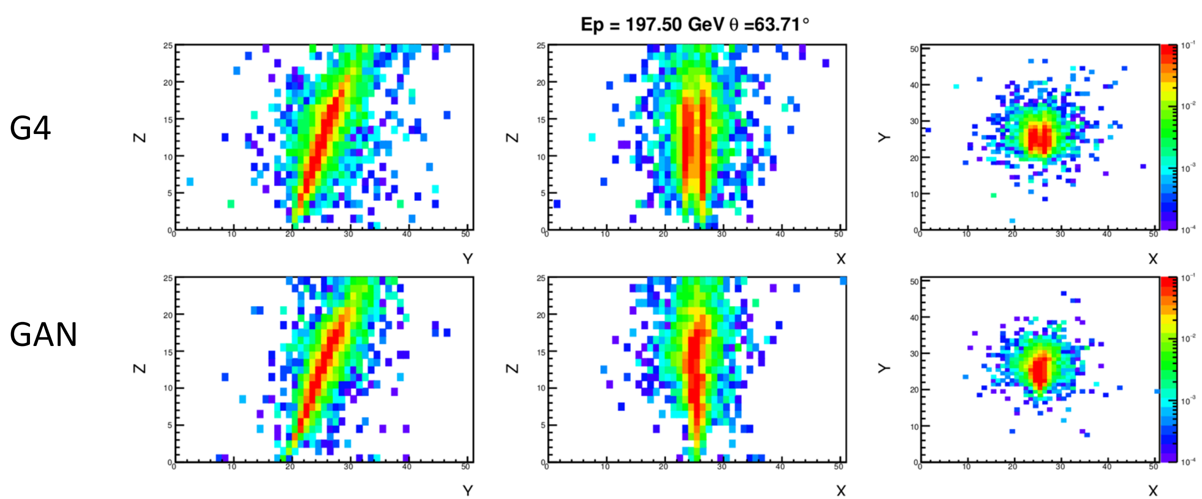
TF 1.14(WIP)



More complex 3DGAN

Larger images with incident angle 60° to 120°

- A more **realistic scenario** where image is generated condition on both:
  - Primary particle energy
  - Incident angle
- Variable angle data (electrons)
  - Event → **51 x 51 x 25 image → 65, 025 cells**
    - **400,000** events from 2 to 500 GeV
- Event size is more than **4x** larger
- Thus training data size is also larger
- Network is deeper (**~1.2 M parameters**)

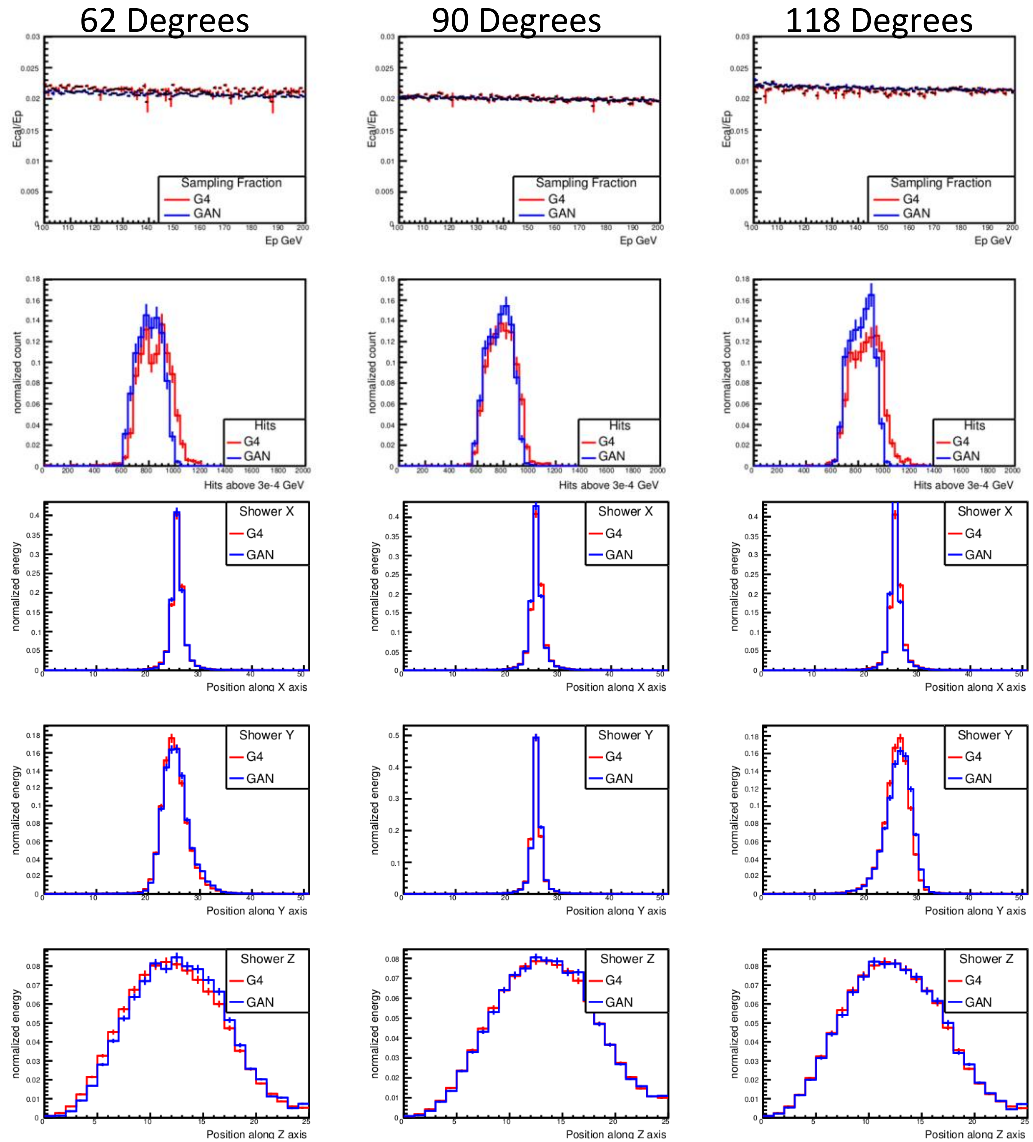


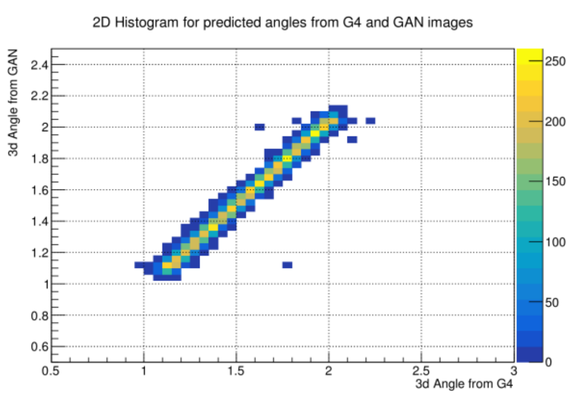
CERN openlab

Physics performance

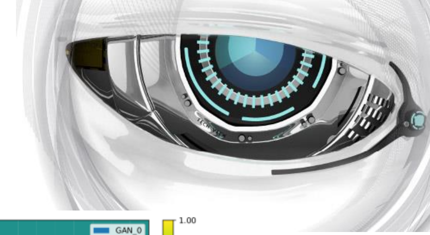
For primary particle energy 100-200 GeV and angle in bins around 62, 90 and 118 Degrees

- Sampling Fraction
- Hits
- Shower Shapes:
  - Energy deposited along x, y and z axis
- Measured Angle

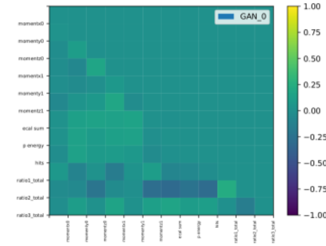


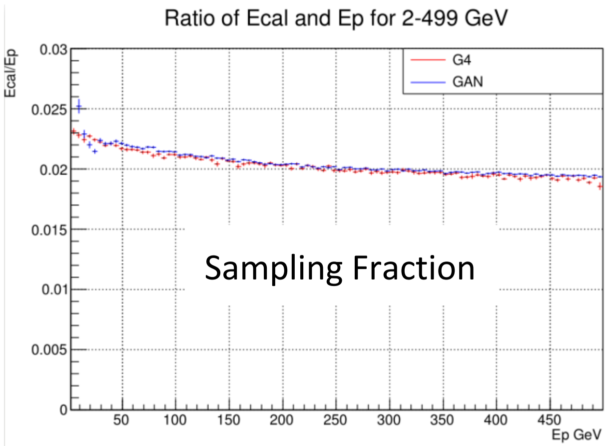


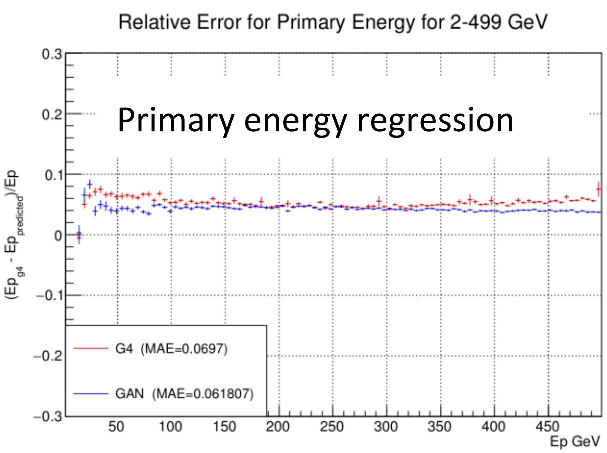
Transfer Learning



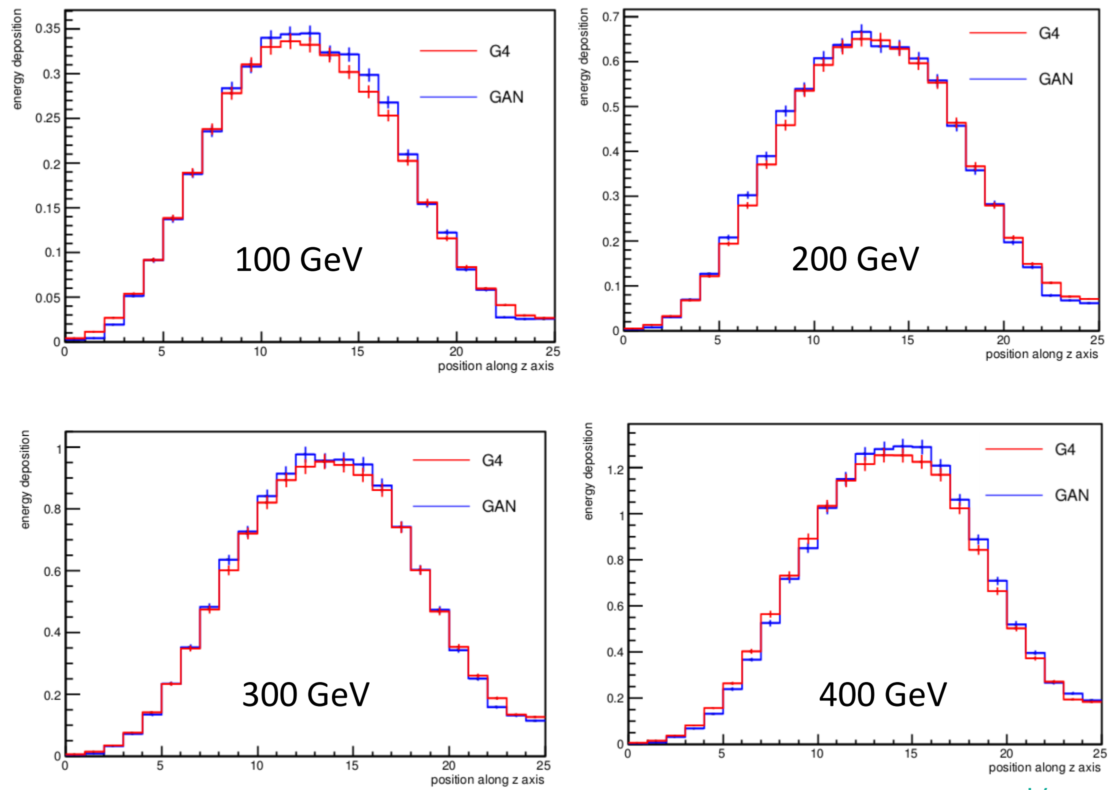
- Training for 2-500 GeV spectrum
  - Starting from pretrained weights (trained for 100-200 GeV)



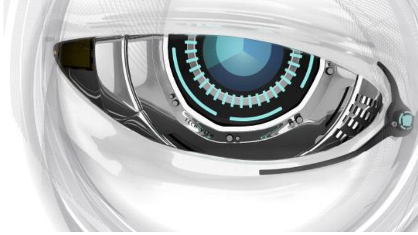




Shower shapes in the longitudinal direction for Different Primary energies



Generalisation



Training and architecture hyper-parameters optimisation

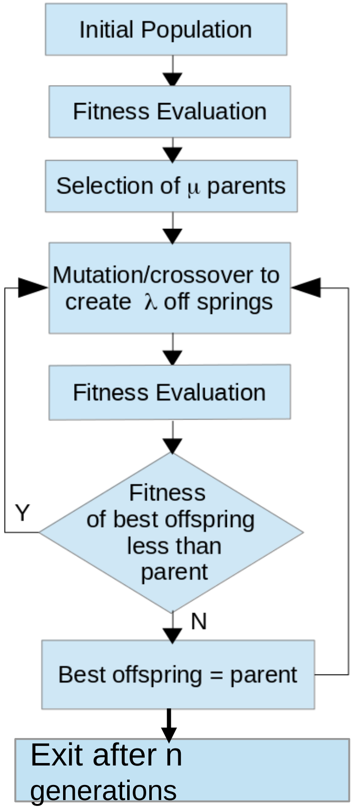
- Different geometries, read-out patterns, energy scales
  - Tuning the right architecture cannot be done by hand
  - Full parameter scan is resource/time consuming.
- Test different optimisation approaches:
  - Sequential Model-Based Optimization
    - Optimize intial architecture candidate, defining a finite set of states to explore
  - Reinforcement Learning
    - Network accuracy is the **reward function**. Architecture or hyper-parameter modification are **actions**
  - Evolutionary Algorithms
    - Can allow simultaneous weights training and architecture optimisation

CERN openlab

Evolutionary Approach

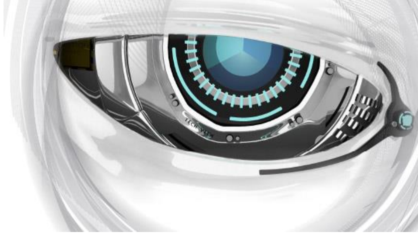
Genetic Algorithm to train and optimize neural networks simultaneously

- **GA** can train Neural Network
  - **Global** instead of local minima
  - **Complex and indirect cost** functions are possible
  - **Highly Scalable**
- Currently used in hyper-parameter scans
  - Architectures are encoded as a chromosome
  - Weights are trained by gradient descent for evaluation of each individual
  - Time and resource intensive



Challenges

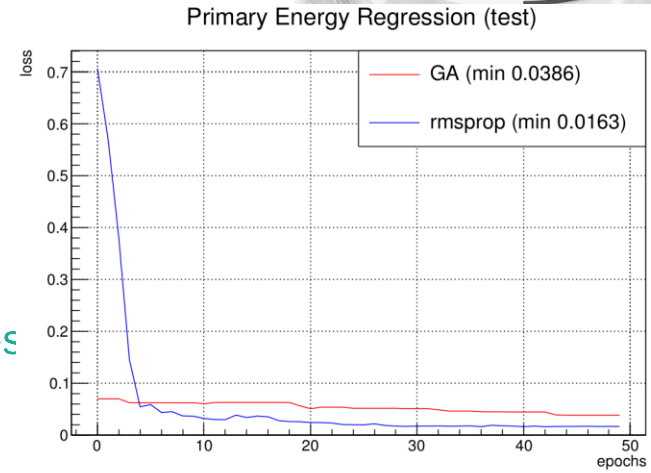
GA for GAN

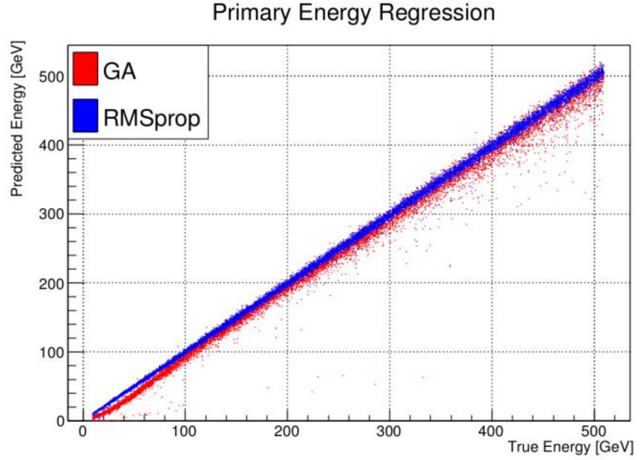


- Network Size:
  - Trainable parameters in **millions** for 3DGAN model
- Large Computing resources?
- Architecture Optimization:
  - Flexible and stable
- Adversarial Training:
  - Simultaneous training of **two networks**
- Inexact solution:
  - A **hybrid approach** can incorporate SGD as a callback

Initial Implementation

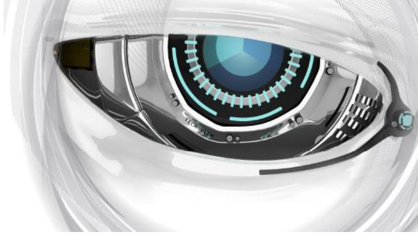
GA with pytorch





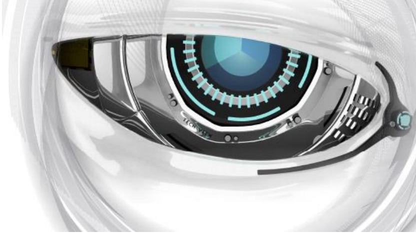
- Reduced complexity:
  - Simplified discriminator regression on 2D images
  - Weight update
- Implement evolutionary algorithm
  - Testing batch training with GA
    - Smaller batches resulted in better training
  - Random updates vs. random weights
    - Random weights allow faster convergence
  - 6-8 optimal number of offsprings
  - No Bias
  - Comparison to RMSprop (lr=0.01)
    - Faster convergence
    - Lower accuracy
- Update weights and architecture at the same time is faster than GA based hyper-parameter scan
- Improve accuracy by adding gradient descent steps

Summary & Plans



NN training will be a new workflow for large HEP experiments

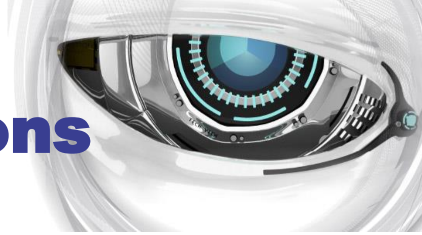
- Distributed training and HPC optimization is critical
    - Enables **architecture optimization** and **generalization**
    - Increase the size of the problems we can solve
- Results on 3DGAN optimisation are very promising
    - Reduced training time by **8x** on single node
    - Linear scaling brings down training time to **< 0.5 min/epoch** on **256** modes of 2CPU Xeon 8268
    - Inference time is **x20000** faster than Monte Carlo approach
- Implement GA based combined architecture and training search Estimate performance and needs in terms of computing resources
    - Several techniques can be explored (indirect weight encoding and asynchronous update)
- Increase complexity (network, dataset, …)



Thank you

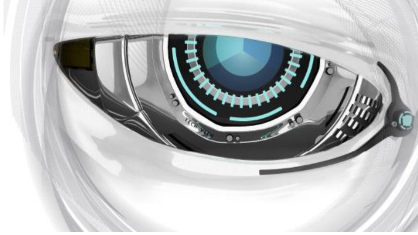
Questions ?

Architecture, Dataset & Runtime Options



- Optimise filter sizes
  - Conv Filters: Multiple of 16 (MKL-DNN optimizations)
- Dataset: 200000 electrons
  - Training Samples: 180000 & Validation: 20000
- Batch Size: 8/Worker, # Workers/Node=4/Node (Mapped to NUMA domains)
- TF tuning: inter_op: 2 & Intra_op: 11 (Xeon® 8160 is 24C/CPU); AVX512 –FMA support
- Learning Rate: 0.001, Optimizer: RMSprop
- Warmup Epochs: 5 (Facebook Methodology), Training Epochs: 25

Stampede2/TACC



Configuration Details

- Compute Nodes:
  - 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.
- TensorFlow 1.6:
  - Built & Installed from source: https://www.tensorflow.org/install/install_sources
- Model:
  - CERN 3D GANS from https://github.com/sara-nl/3Dgan/tree/tf
- Dataset:
  - CERN 3D GANS from https://github.com/sara-nl/3Dgan/tree/tf
- Performance (256 Nodes):
  - OMP_NUM_THREADS=24 HOROVOD_FUSION_THRESHOLD=134217728 export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2 mpirun -np 512 -ppn 2 python resnet_main.py --train_batch_size 8 --num_intra_threads 24 --num_inter_threads 2 --mkl=True --data_dir=/path/to/gans_script.py --kmp_blocktime 1
  - https://portal.tacc.utexas.edu/user-guides/stampede2

Training time