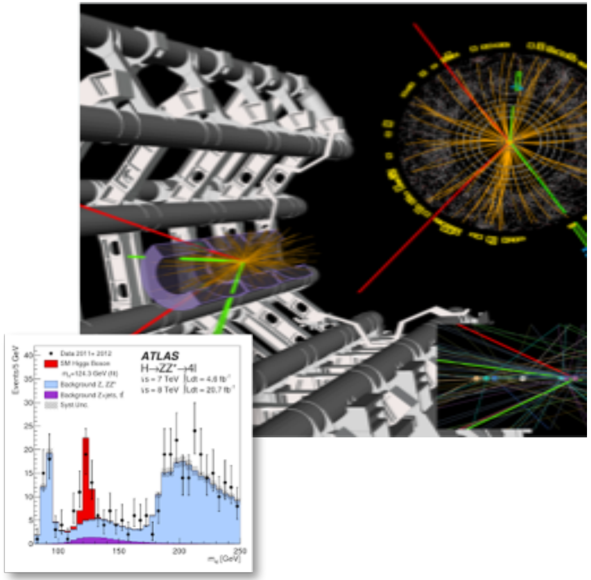
Machine Learning for (fast) simulation

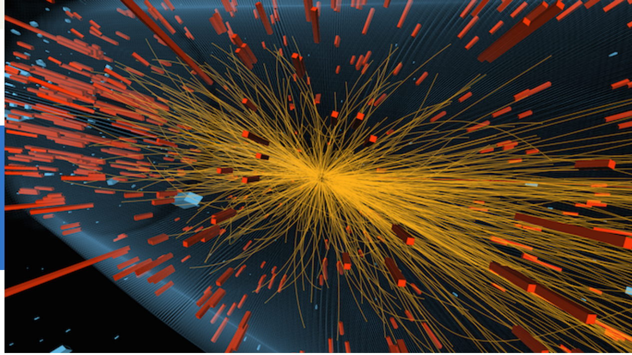
Sofia Vallecorsa for the GeantV team

Monte Carlo Simulation: Why

- **Detailed simulation of subatomic particles is essential for data analysis, detector design**
  - Understand how detector design affect measurements and physics
  - Use simulation to correct for inefficiencies, inaccuracies, unknowns.
  - The theory models to compare data against.



A good simulation demonstrates that we understand the detectors and the physics we are studying



The problem

- Complex physics and geometry modeling
  - Some physics process are extremely rare!
- Heavy computation requirements, massively CPU-bound
- Already now more than 50% of WLCG power is used for simulations

WLCG
Worldwide LHC Computing Grid

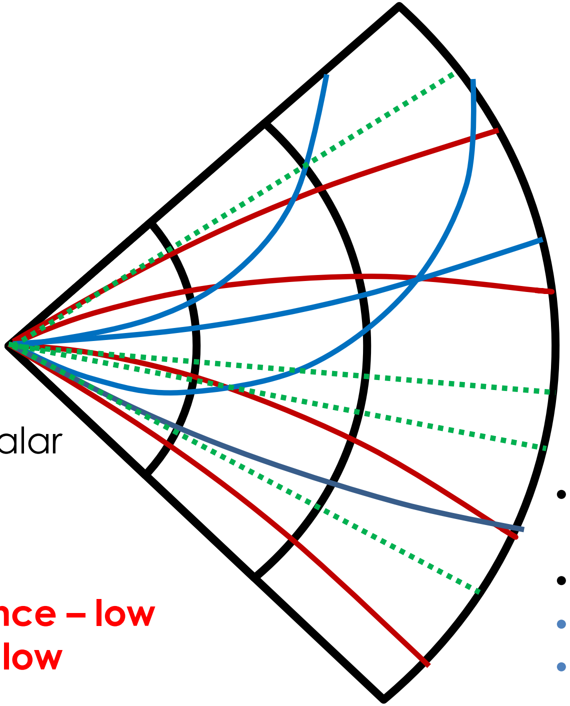
200 Computing centers in 20 countries: > 600k cores

@CERN (20% WLCG): 65k processor cores ; 30PB disk + >35PB tape storage

By 2025 with the High Luminosity LHC run we will have to run simulation 100x faster!

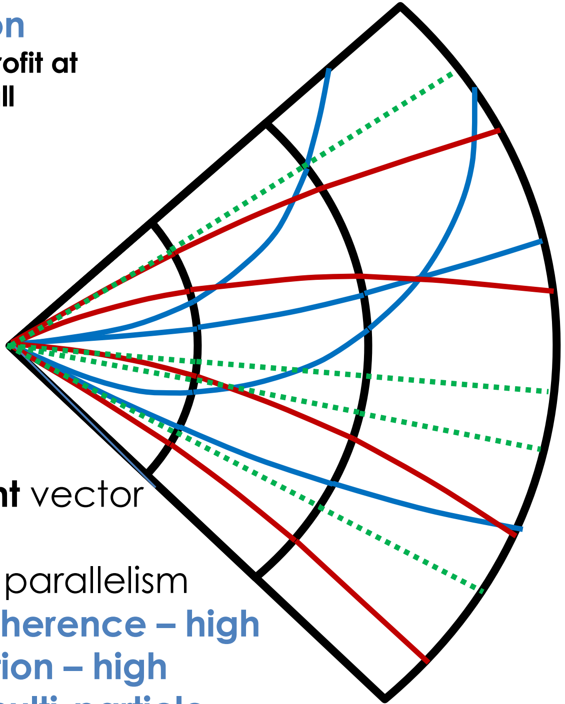
GeantV: Adapting simulation to modern hardware

Classical simulation
hard to approach the full machine potential



- **Single event** scalar transport
- Embarrassing parallelism
- **Cache coherence – low**
- **Vectorization – low (scalar auto-vectorization)**

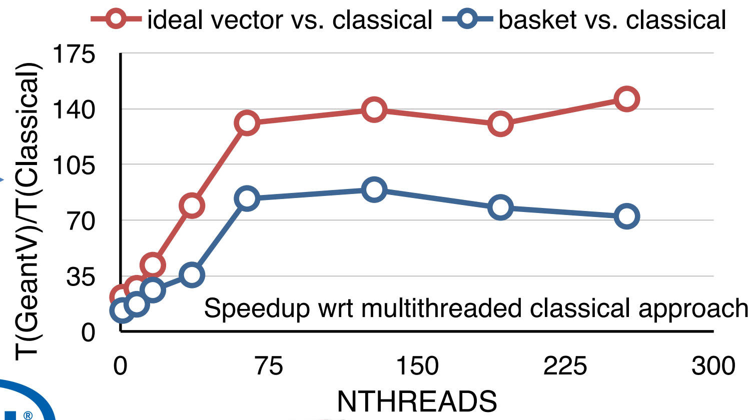
GeantV simulation
needs to profit at best from all processing pipelines

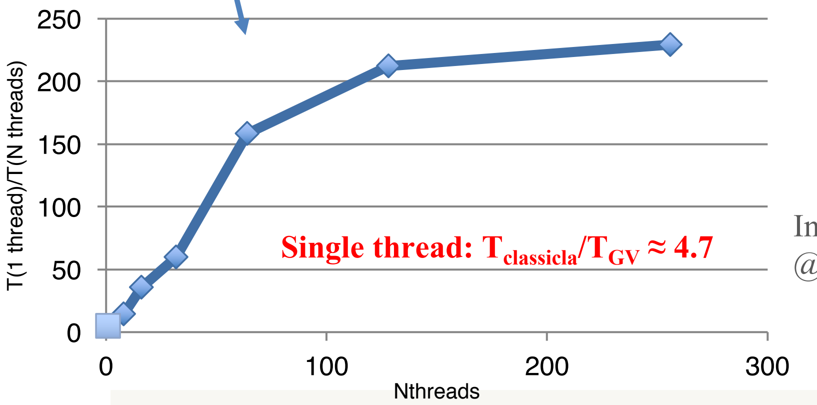


- **Multi-event** vector transport
- Fine grain parallelism
- **Cache coherence – high**
- **Vectorization – high (explicit multi-particle interfaces)**

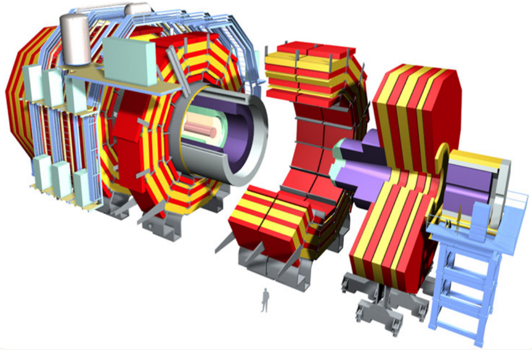
Some benchmarks on Intel Xeon Phi

- GeantV delivers already a part of the expected performance
- Testing geometry navigation performance wrt classical
- Full detector simulation (LHC CMS)



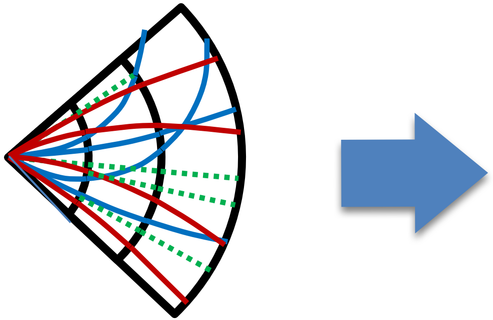


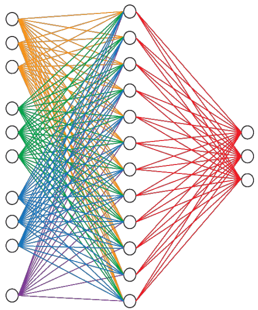
Intel Xeon Phi 7210
@1.30 Hz – 64 cores



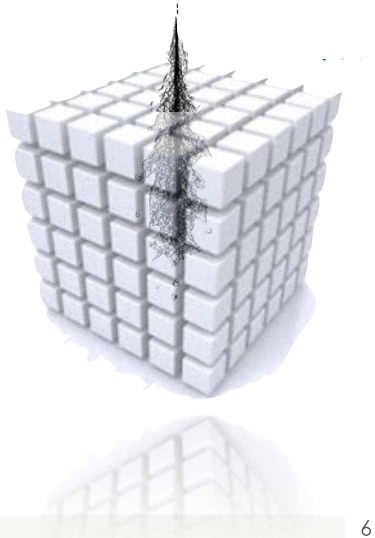
Going beyond 10x: fast simulation

- In the best case scenario GeantV will give 10x speedup → not enough
- Improved, efficient and accurate fast simulation based on Deep Learning techniques





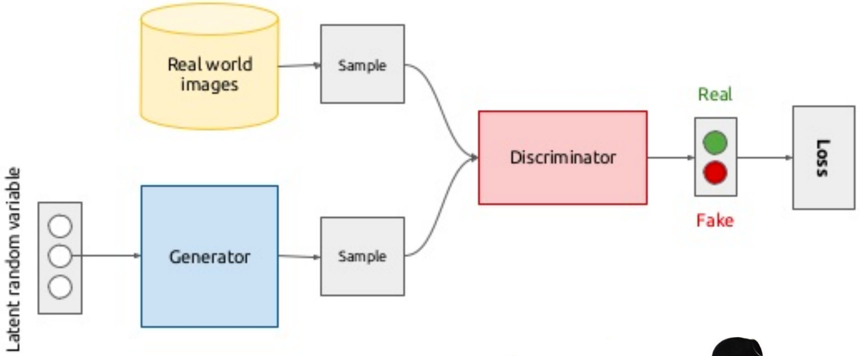
Test on most time consuming detectors: calorimeters



Generative Adversarial Networks

arXiv:1406.2661v1

- Mostly used for computer vision (Goodfellow et al, 2014)
- Simultaneously train two models:
  - Generative model G to capture the data distribution
  - Discriminative model D to distinguish real data from G data (“**catch G**”)
- The training procedure for G is to maximize the probability of D making a mistake



Training GANs is a many steps process:

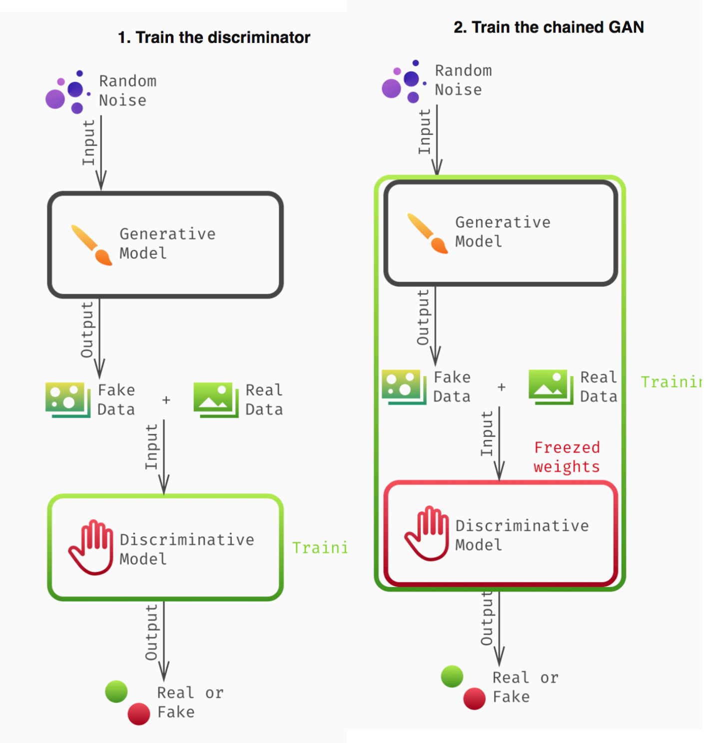
1. Sample noise and generate images with the Generator.

2. Tthe Discriminator to recognize Generator data from Real data.

3. Push the chained Generator and Discriminator to tell you that it is Real data.

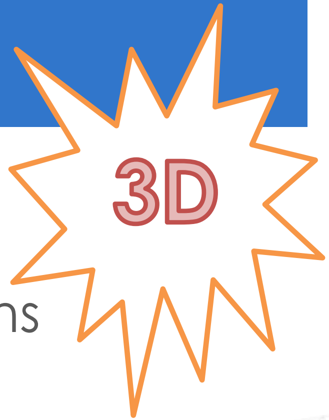
   I. Discriminator weights are frozen.

4. Back feed to Discriminator and repeat for as many epochs as needed

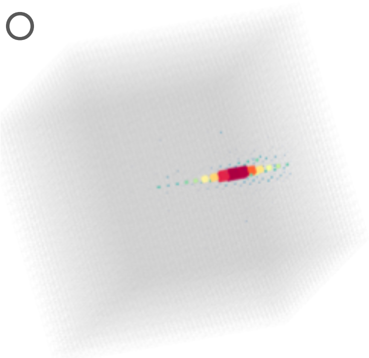


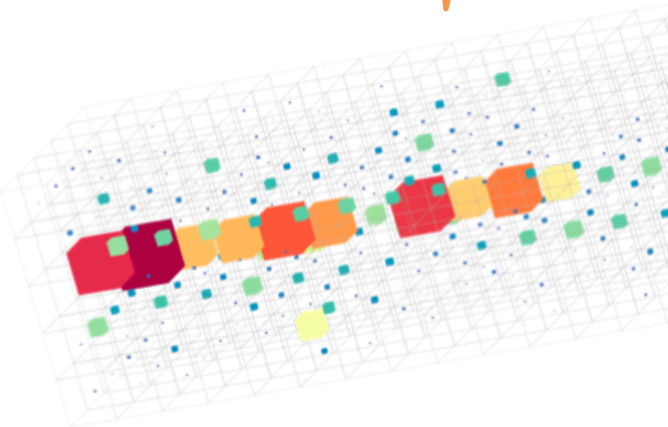
http://www.rricard.me/machine/learning/generative/adversarial/networks/2017/04/05/gans-part1.html

3D GAN for particle detectors



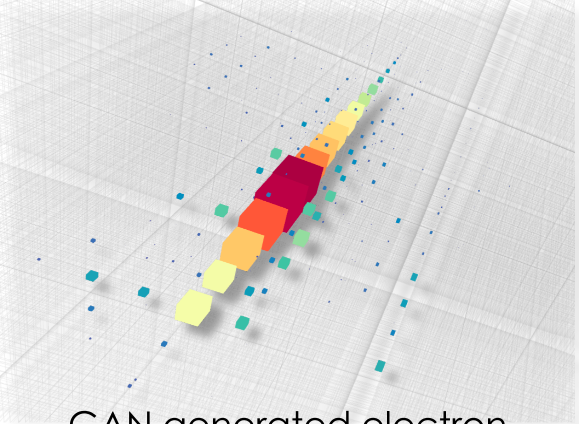
- Treat energy deposits in cells as 3D image
  - Generator and Discriminator based on 3D convolutions
- Explored several "tips&tricks"
  - No batch normalisation in the last step, LeakyRelu, no hidden dense layers 😀, Adam optimiser ☹️
  - Batch training
  - Combined cross entropy



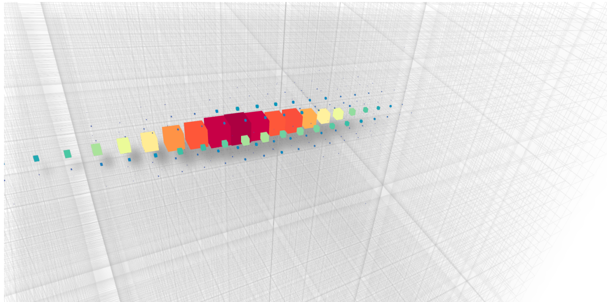


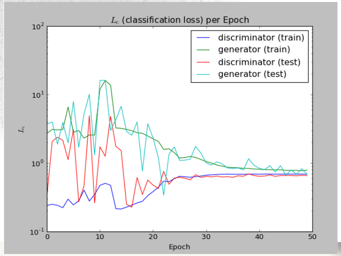
Some generated images

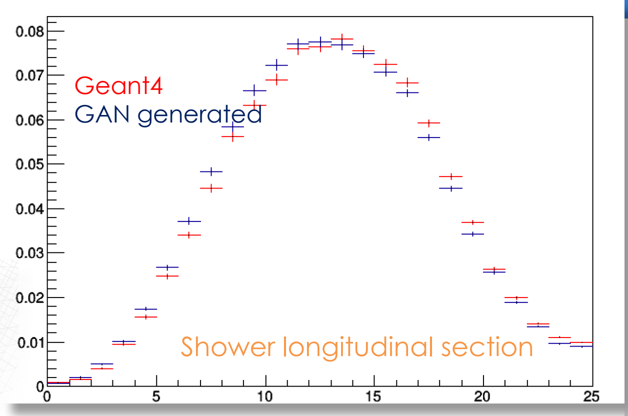
- First results look very promising!
- Qualitative results show no collapse problem



GAN generated electron







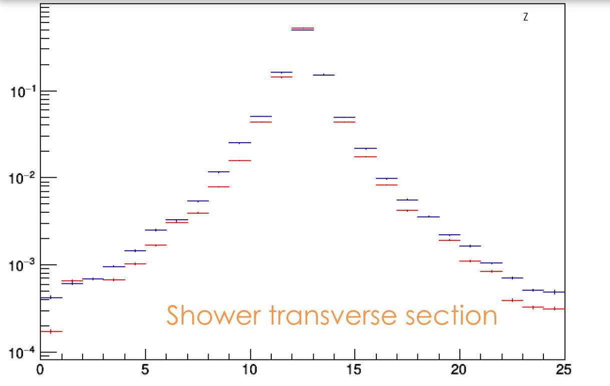
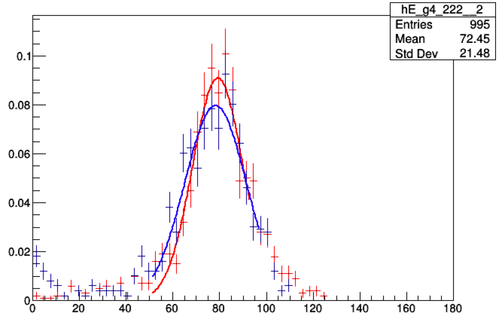
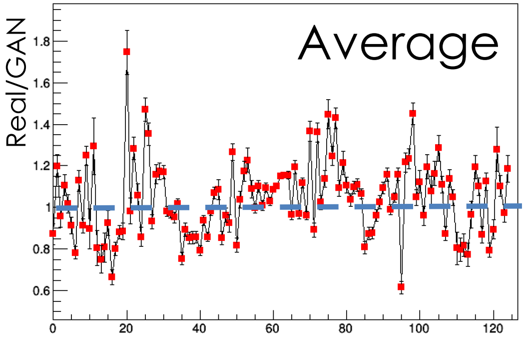
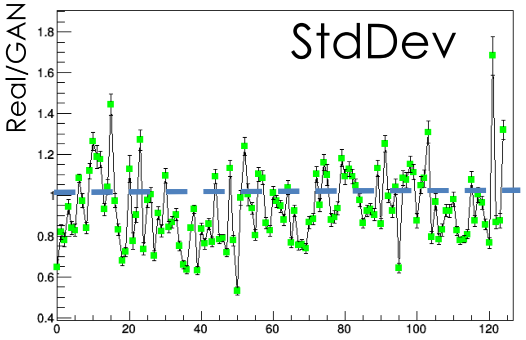


Image validation

- Energy distribution in single cells







- Cell energy standard deviation is underestimated by GAN
- Set up higher level criteria for image validation

Training time ?

- Using DL techniques for fast simulation is profitable if training time is not a bottleneck
  - Depending on the use case retraining might be necessary
  - Hyper-parameters scan and meta-optimization
  - 3D generative adversarial networks are not "out-of-the-box"
    - Complex training process
    - Our model is currently based on keras + tensorflow (no MPI!)

Prototype on multi-nodes

- Thanks to a collaboration with the CINECA center, Italy and Intel, we have access to a cluster of Xeon Phi interconnected with Intel Omni-Path
- Implement model in Intel optimized Caffe* and link to Intel MLSL and Intel MKL-DNN
  - Needs fixes in Intel Caffe*
- Measure scaling and hotspots on single Xeon Phi and clusters

Summary & Plan

- First results are very promising!
- Detailed assessment of current performance & optimisation
- Generalisation to different detectors
- Comparison to other DL techniques (recurrent networks)
- **Looking forward to test upcoming Intel software & hardware solutions!**
  - Switch to Neon as soon as v3.0 is available
  - Next-generation Intel® Xeon® processor family "Skylake" and next generation of Intel Xeon Phi processors "Knights Mill"
  - Test inference dedicated hardware (integrated FPGA solution) Intel DLIA

Thank you