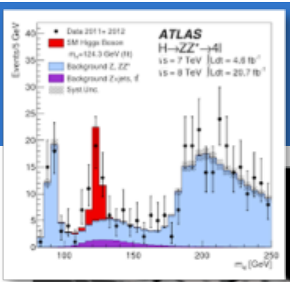
GANs for (fast) simulation

Sofia Vallecorsa for the GeantV team

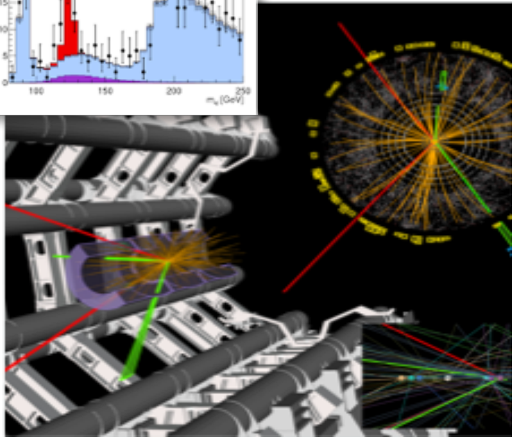
Outline

- Introduction
- A generic framework for fast simulation
- GANs for calorimeter showers
- Summary & Plans

Simulation in HEP



- **Detailed simulation of subatomic particles is essential for data analysis, detector design**
  - Complex physics and geometry modeling
  - Heavy computation requirements, massively CPU-bound



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

@HL-LHC needs x100 speed-up in simulation

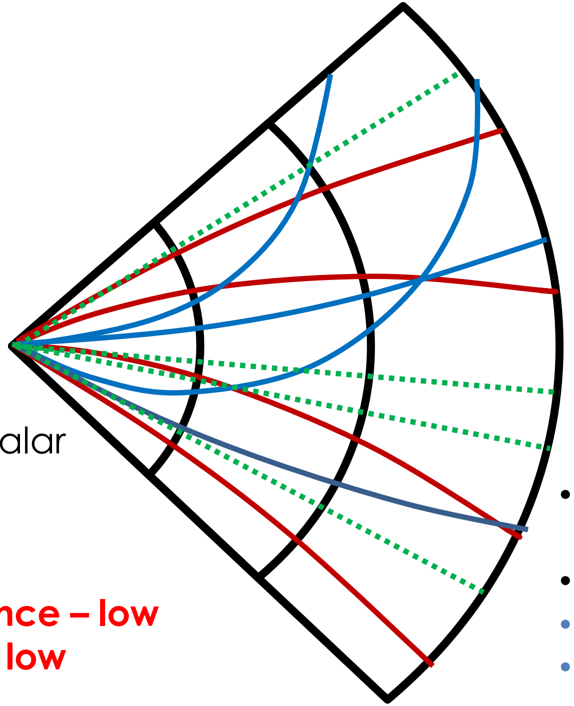
Speeding up simulation

- State of the art software is Geant4
  - All particle MonteCarlo transport program
  - C++ open source simulation toolkit
  - Capable of handling extremely complex geometries
  - Large spectrum of applications
  - Massive and extensive validation
- Event level multi-threading

GeantV: Adapting simulation to modern hardware

Classical simulation

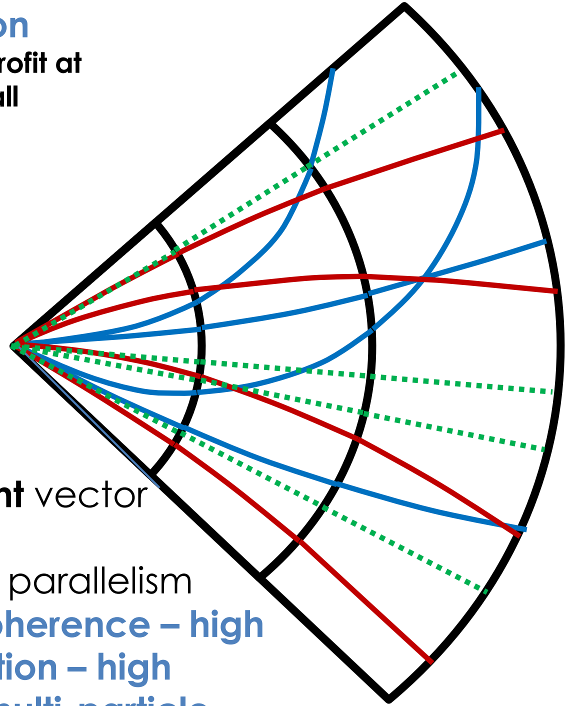
hard to approach the full machine potential



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

GeantV simulation

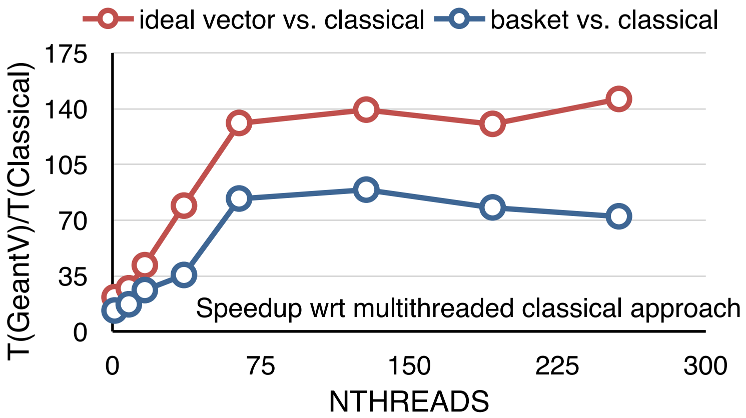
needs to profit at best from all processing pipelines

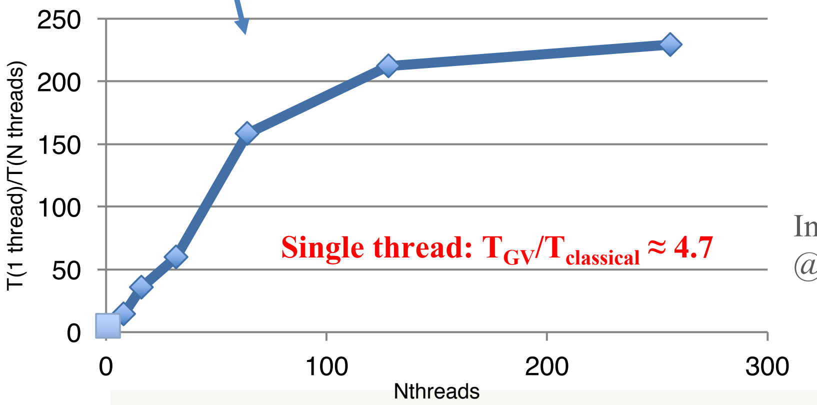


- **Multi-event** vector transport
- Fine grain parallelism
- **Cache coherence – high**
- **Vectorization – high (explicit multi-particle interfaces)**

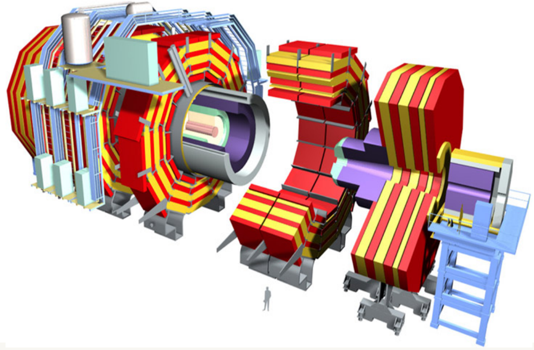
Some benchmarks on multi-cores

- GeantV delivers already a part of the expected performance
- Testing new geometry navigation performance wrt classical (ROOT)
- CMS detector simulation (tabulated physics)





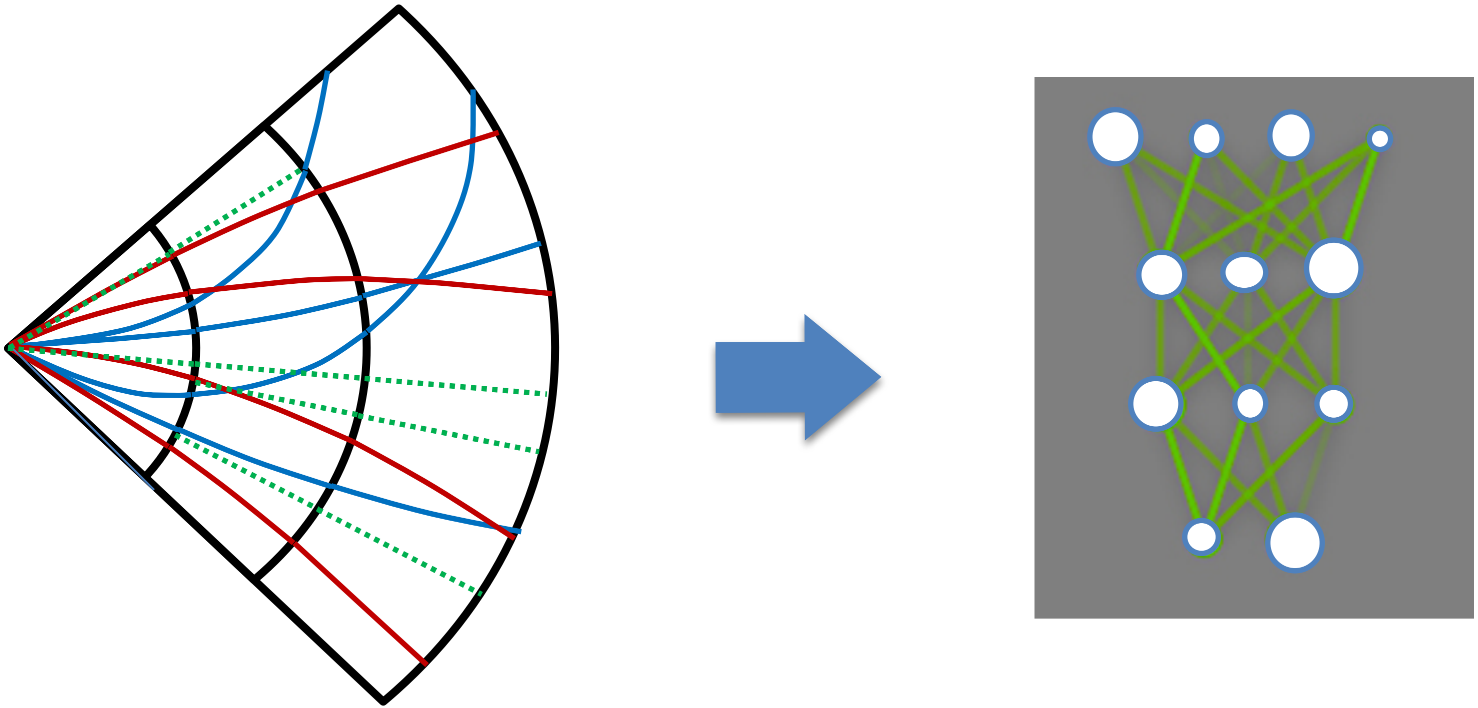
Intel Xeon Phi 7210
@1.30 Hz – 64 cores



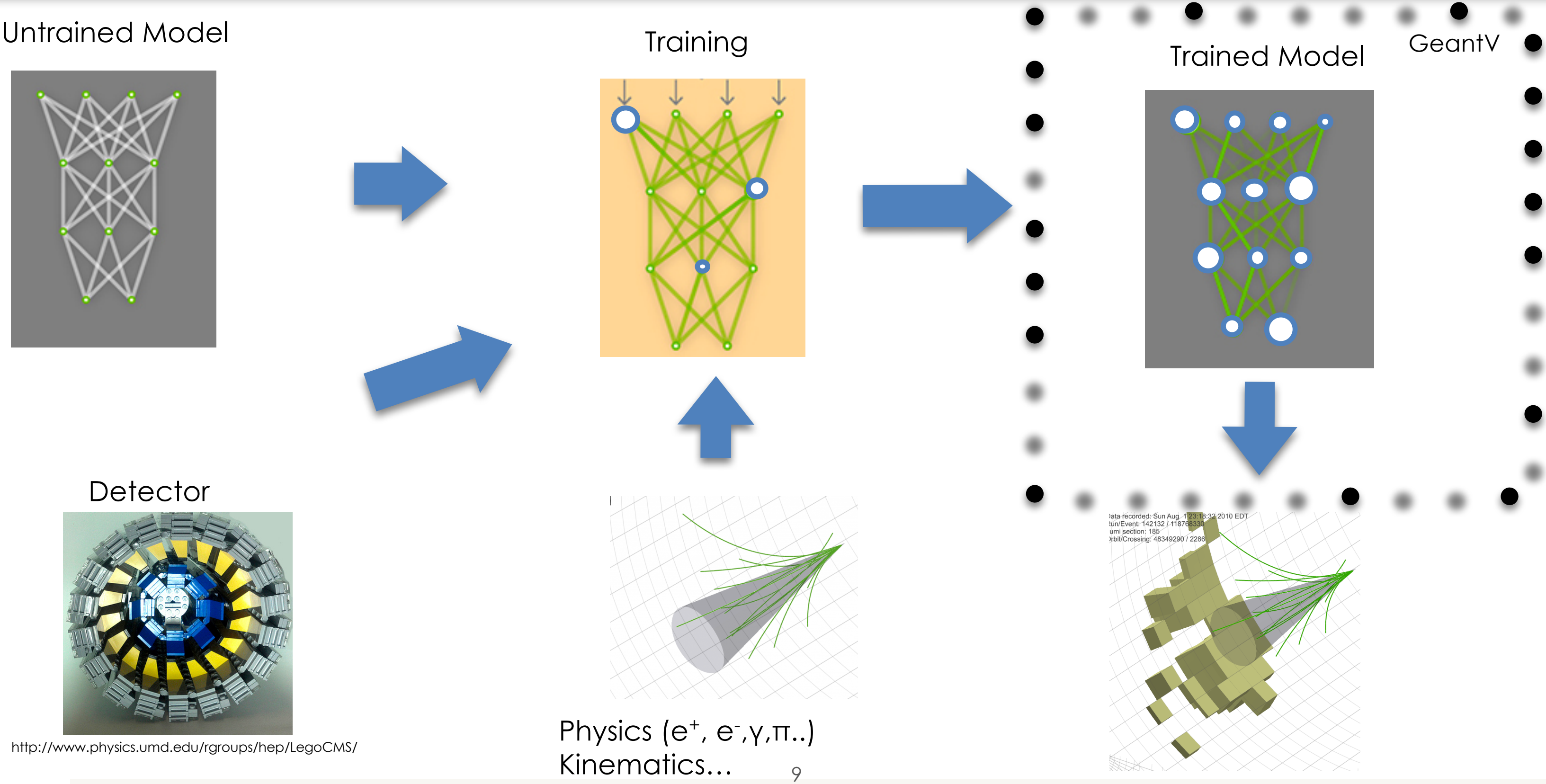
Going beyond x10: fast simulation

- In the best case scenario GeantV will give O(10) speedup
  - Not enough to cope with HL-LHC expected needs
- Improved, efficient and accurate fast simulation
  - Currently available solutions are detector dependent
- A general fast simulation tool based on Machine Learning techniques
  - Fully configurable interface embedded in GeantV

Going beyond x10: fast simulation

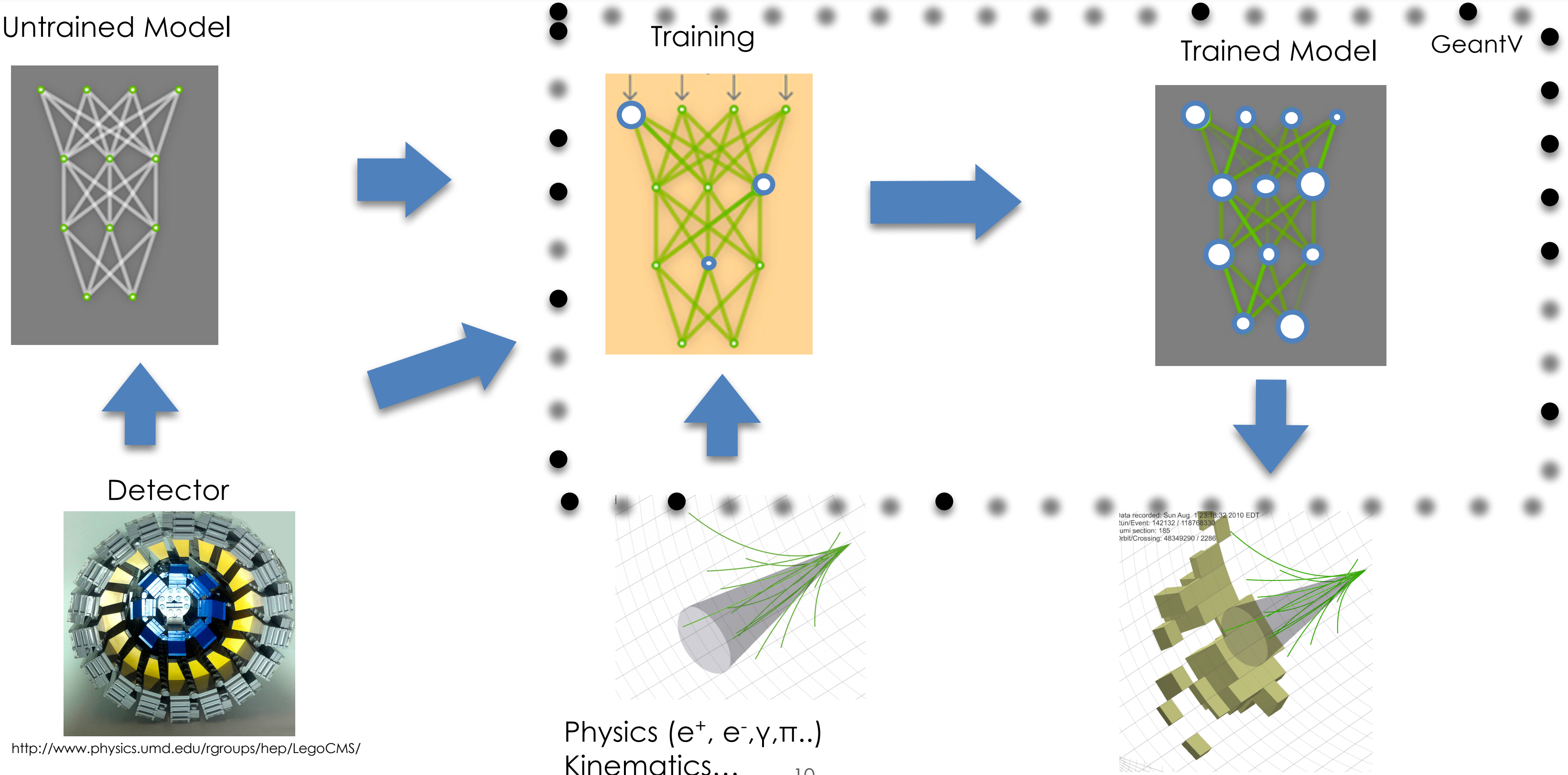


ML engine for fast simulation



http://www.quantumdiaries.org/wp-content/uploads/2011/06/JetConeWithTracksAndECAL.png

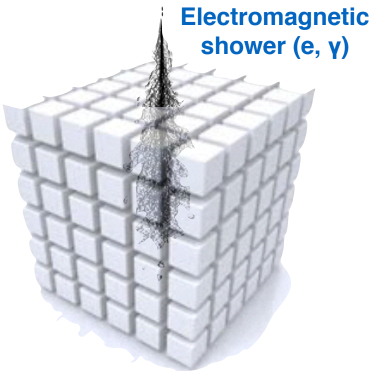
ML engine for fast simulation

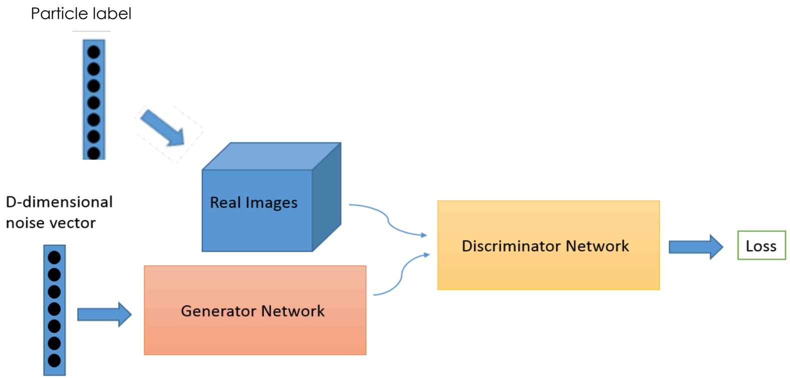


http://www.quantumdiaries.org/wp-content/uploads/2011/06/JetConeWithTracksAndECAL.png

Testing GANs for calorimeter images

- Calorimeters simulation is time consuming
- Treat energy deposits in cells as 3D image
  - Use LCD ECAL dataset(1)



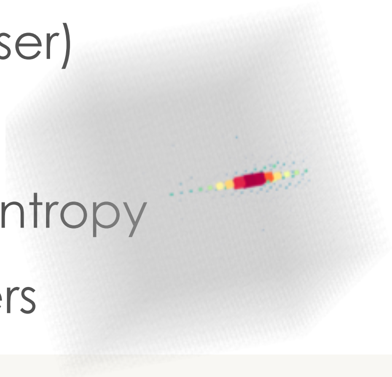


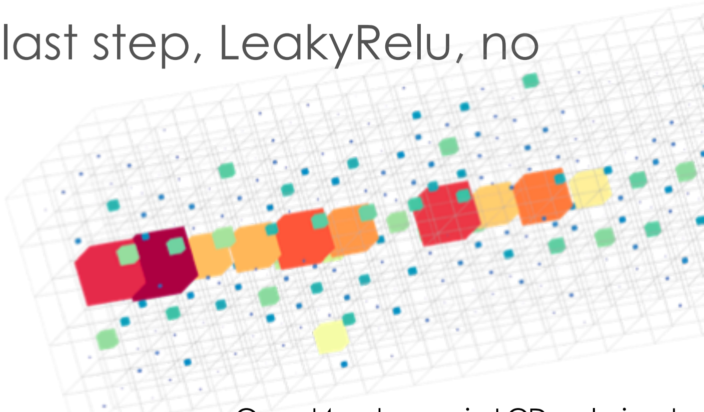
- Use particle flags to condition the training
- 3D convolutions using Keras + Tensorflow

(1) See Amir's talk and tutorials

Testing GANs for calorimeter images

- Similar discriminator and generator models
  - 3D conv layers with different x,y,z filter sizes
- Implemented several tips&tricks found in literature
  - Some helpful (no batch normalisation in the last step, LeakyRelu, no hidden dense layers)
  - Some not (Adam optimiser)
- Batch training
  - Loss is combined cross entropy
  - Tested different optimisers





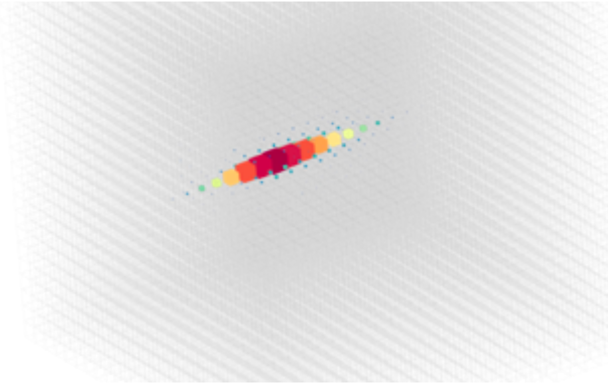
Geant4 π shower in LCD calorimeter

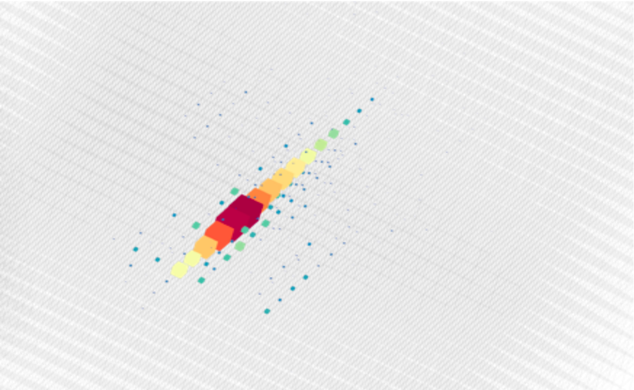
https://github.com/tpmccauley/ispy-hepml

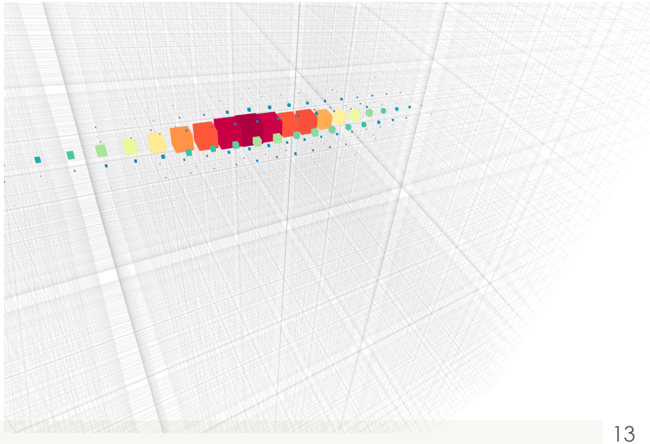
Preliminary

Some images

- Slice energy spectrum
- Start with photons & electrons

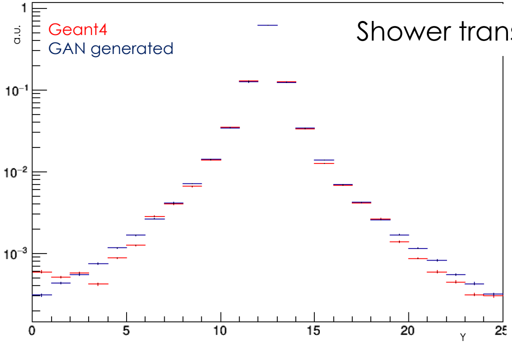


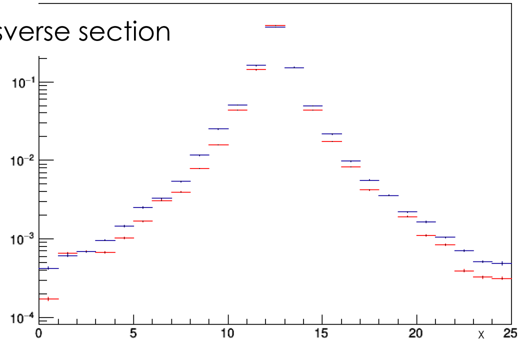


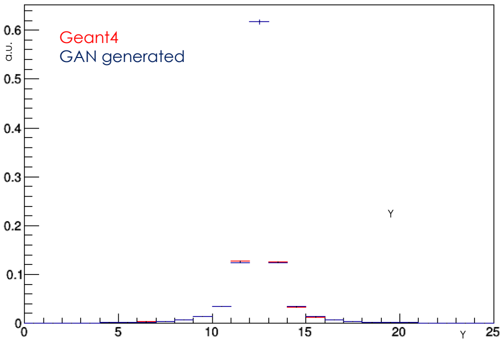


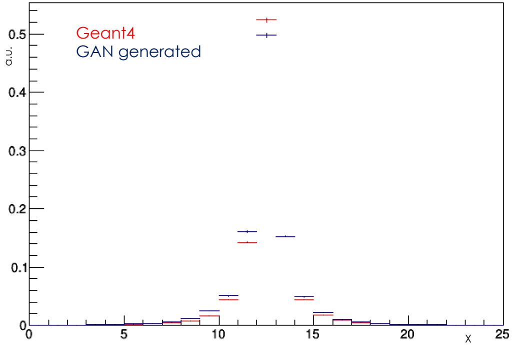
GAN generated electrons

Shower transverse section

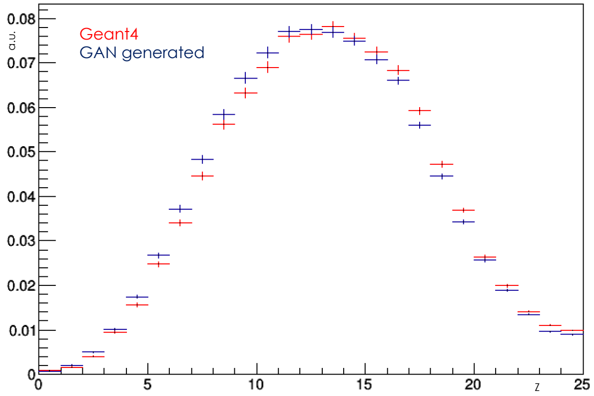






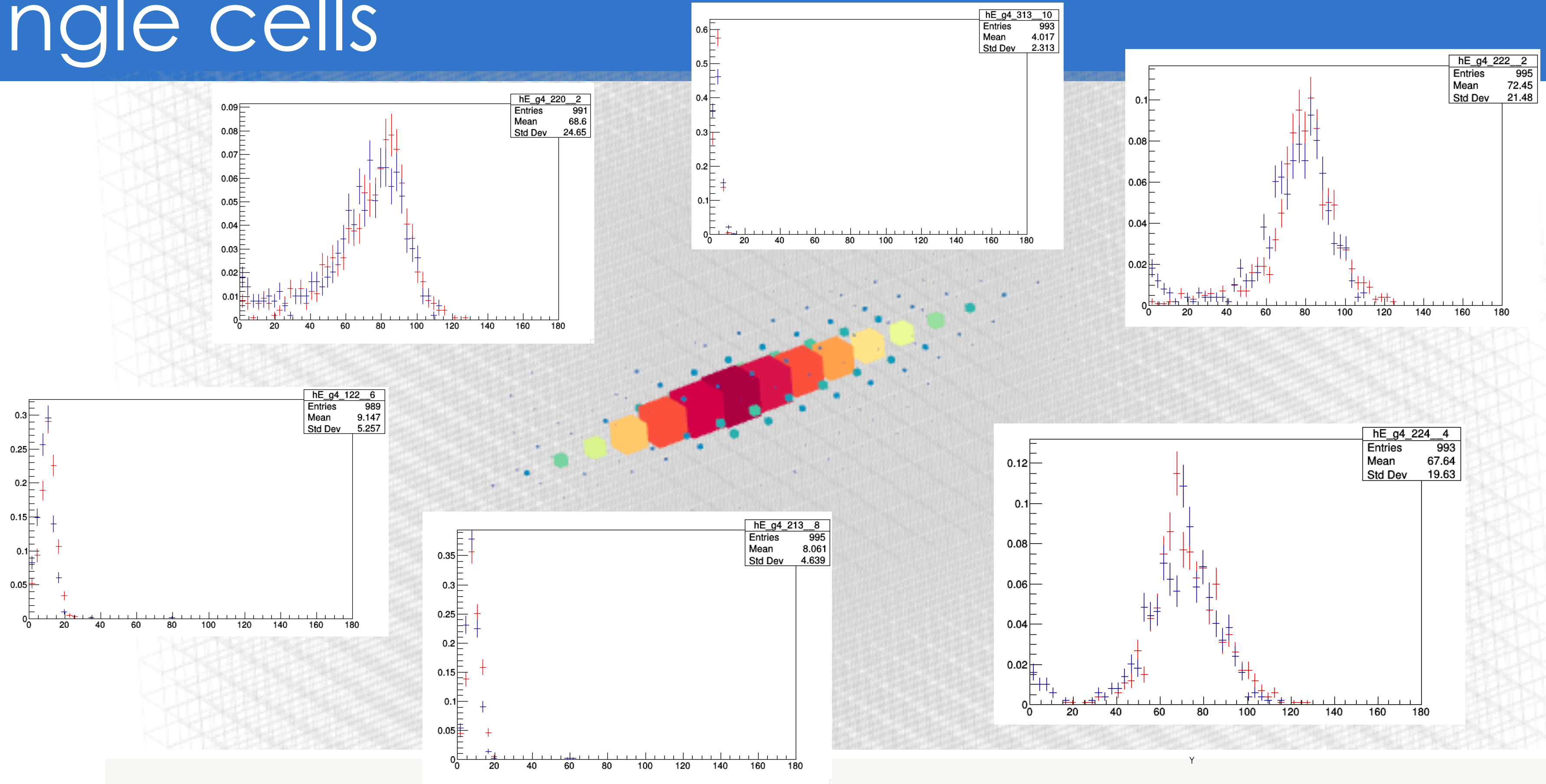


Shower longitudinal section



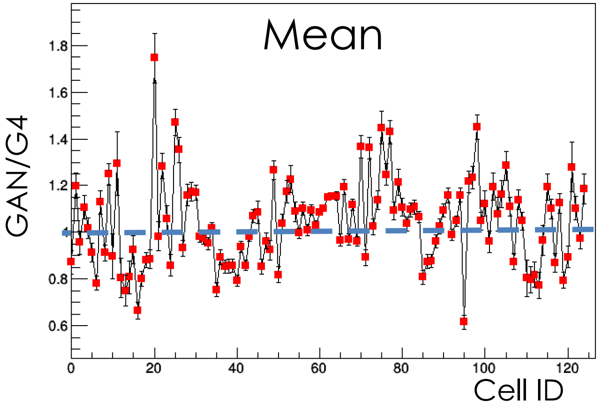
Preliminary

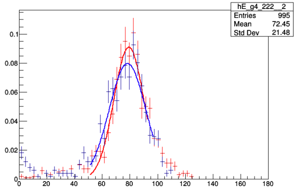
Single cells

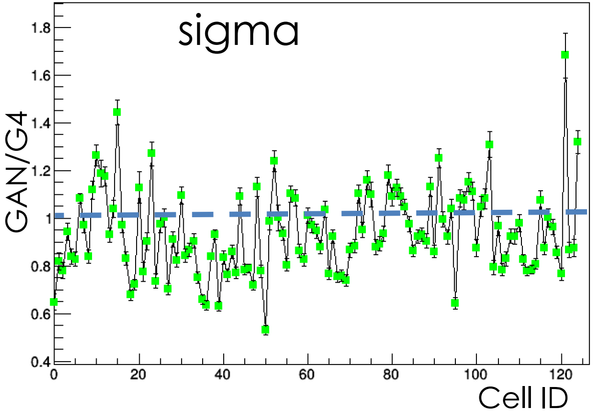


Single cells

- Compare energy distribution mean and sigma per cell







- Cell energy sigma is underestimated by GAN
- Set up higher level criteria for image validation (reconstructed variables)

Next steps

- Detailed study of calorimeter response and comparison to full sim and different fast sim tool
  - Testing different models to improve physics performance
  - Include energy info
- Use information available in the LCD dataset to compare to different techniques (i.e. MO regression)
  - Test different frameworks
- Test training on real data

Training time ?

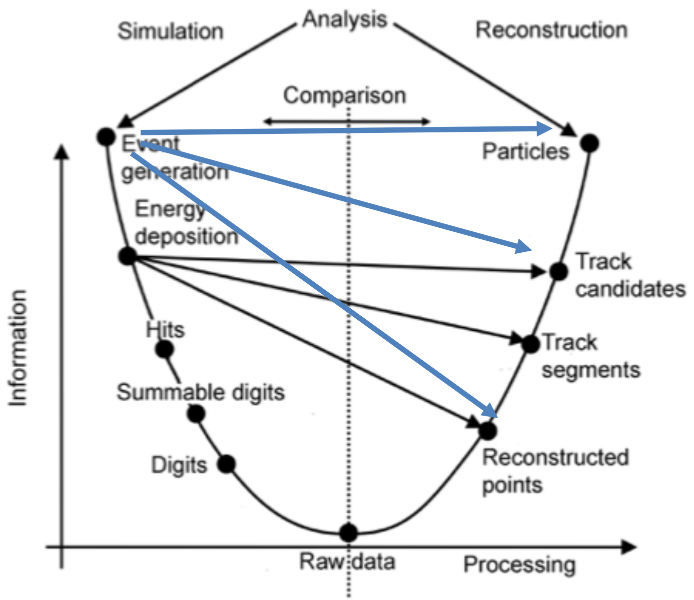
- Currently adversarial training takes a few hours on NVIDIA GTX1080
    - Work on the training algorithm
- Using DL techniques for fast simulation is profitable if training time is not a bottleneck
    - Depending on the final use case retraining the networks might be necessary
- Test different hardware & multi-node scaling

Longer term…

- We want to provide a generic fully configurable tool
  - Optimal network design depends on the problem to solve
    - Embedded algorithms for hyper-parameters tuning and meta-optimization
    - Large hyper-parameter scans
  - Study parallelization on large clusters
    - Evaluate existing libraries
    - Optimize training strategy by reducing communication overhead

Summary

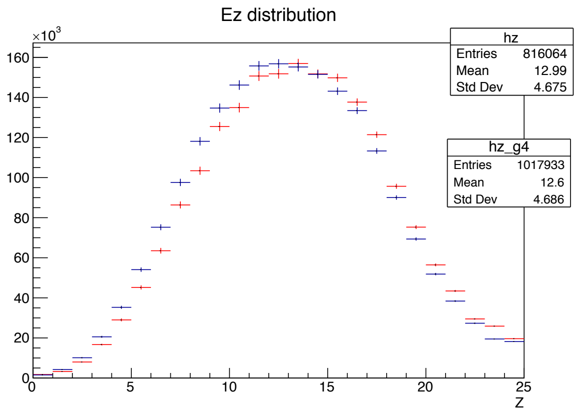
- First images using GANs look very promising
  - Keep working on understanding and improving performance
- Insure computing efficiency and optimal performance on modern hardware
- Initial step of a wider plan to do ML based fast simulation with GeantV
- Even larger speedup gained by replacing digitization and reconstruction steps



Thank you

..many thanks to M. Pierini and J. Vlimant !

Z filter size = 8



Z filter size = 9