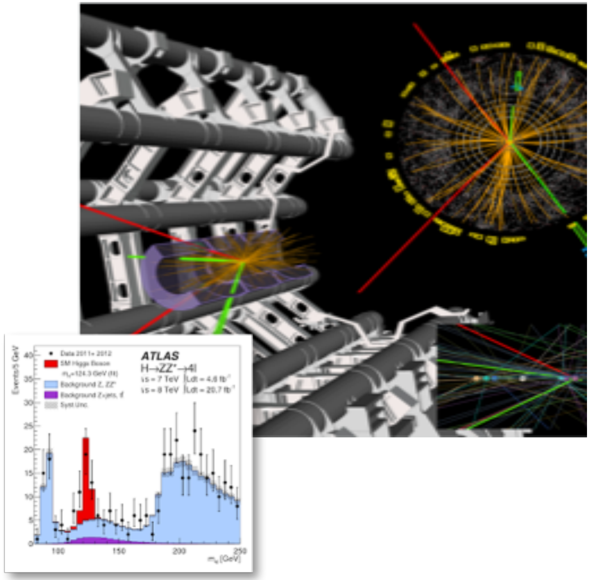
Machine Learning for (fast) simulation

Sofia Vallecorsa for the GeantV team

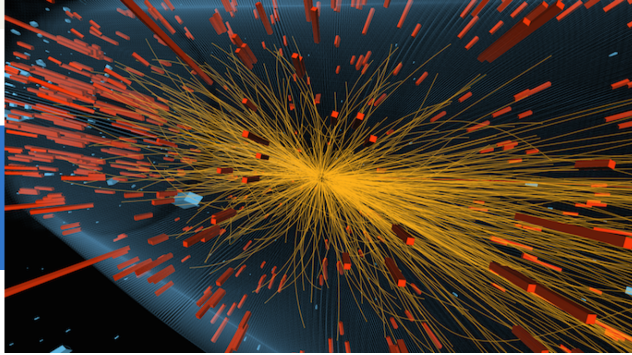
Monte Carlo Simulation: Why

- **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.

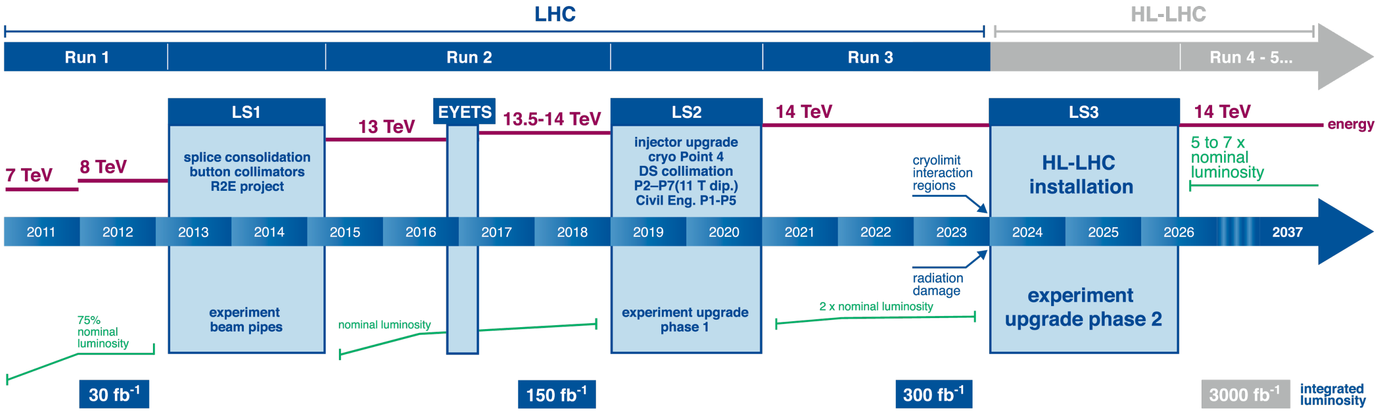


A good simulation demonstrates that we understand the detectors and the physics we are studying

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

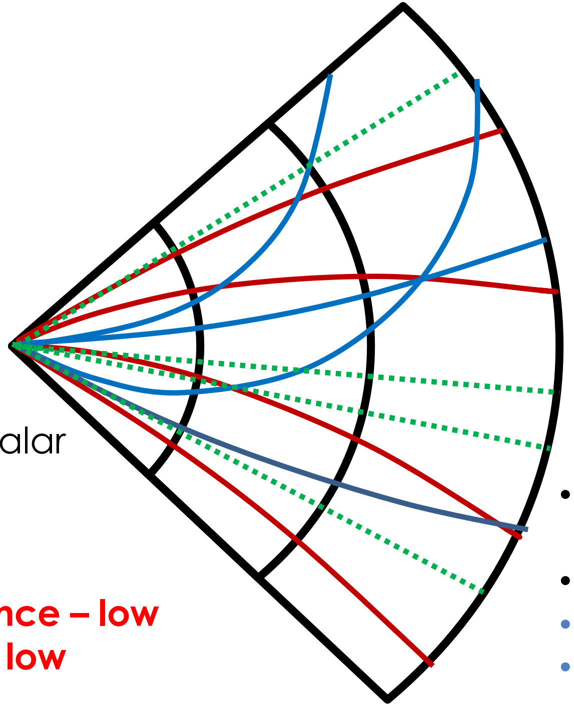


@HL LHC we will need to simulate

- More data
- More complex events
- Faster!

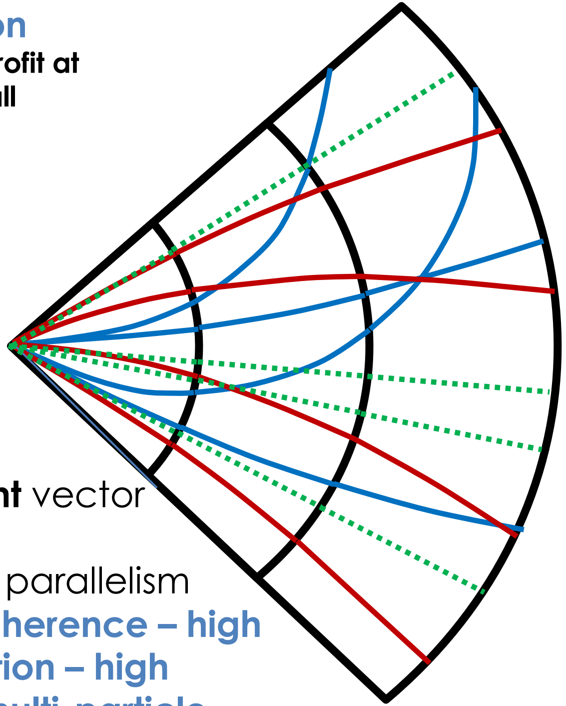
GeantV: Adapting simulation to modern hardware

Classical simulation
hard to approach the full machine potential



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

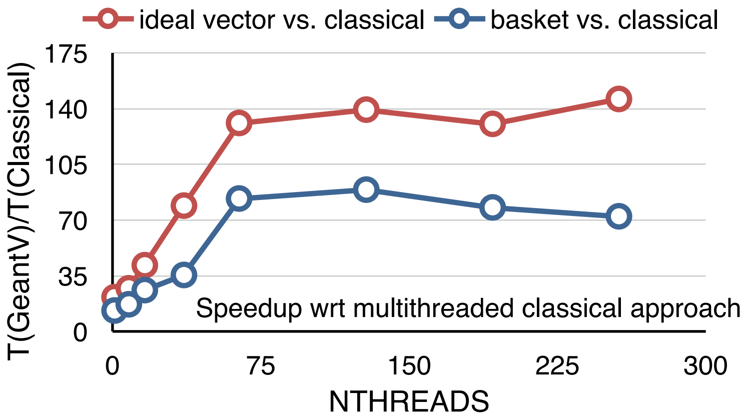
GeantV simulation
needs to profit at best from all processing pipelines

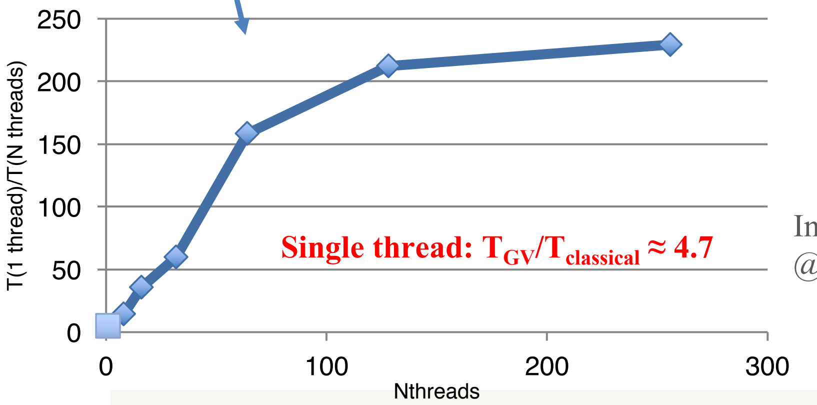


- **Multi-event** vector transport
- Fine grain parallelism
- **Cache coherence – high**
- **Vectorization – high (explicit multi-particle interfaces)**

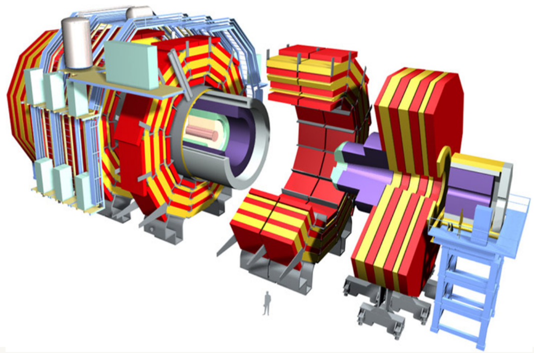
Some benchmarks on Intel Xeon Phi

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





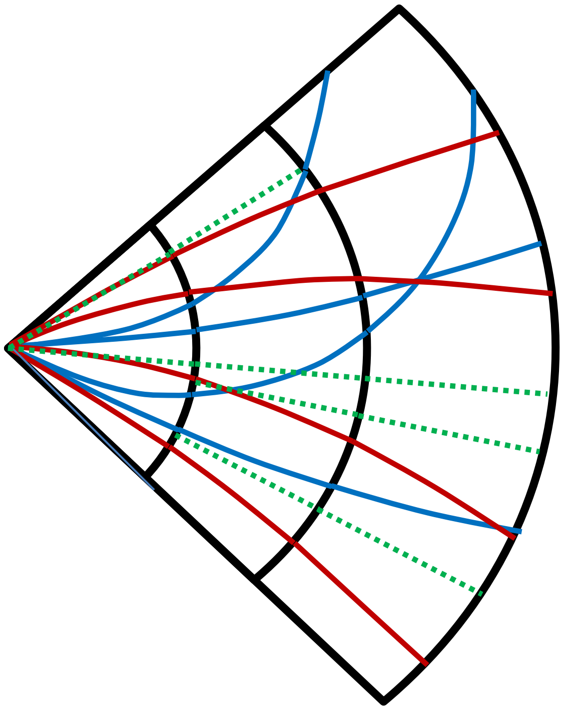
Intel Xeon Phi 7210
@1.30 Hz – 64 cores

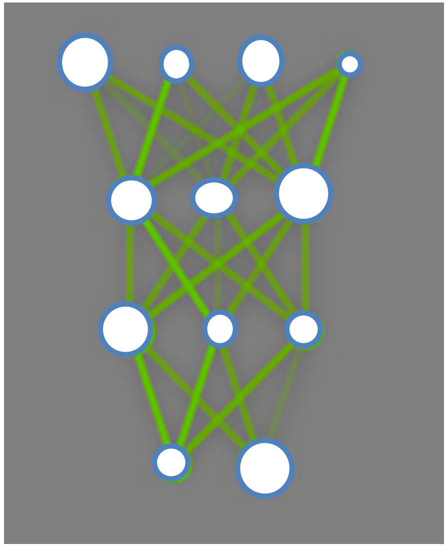


Going beyond x10: fast simulation

- In the best case scenario GeantV will give O(10) speedup
  - It likely won't be enough to cope with HL-LHC expected needs
- Improved, efficient and accurate fast simulation
  - Currently available solutions are detector dependent
  - Looking for a generic approach + user API
- A general fast simulation tool based on Machine Learning techniques
  - ML techniques are more and more performant in different HEP fields
  - Optimizing training time becomes crucial

Going beyond x10: fast simulation



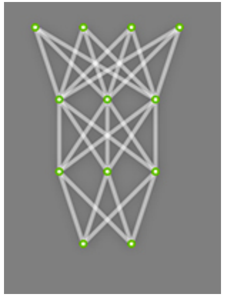


GeantV fast simulation

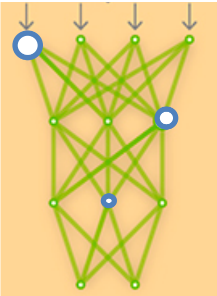
- A project in two steps:
  - Phase1: Proof of concept and generic fastsim interface in GeantV
  - Phase2: Networks design and training optimisation on HPC

ML engine for fast simulation

Untrained Model

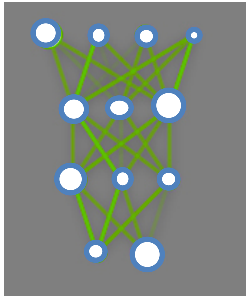


Training

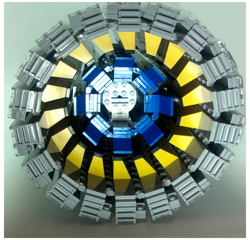


GeantV

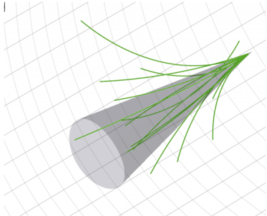
Trained Model



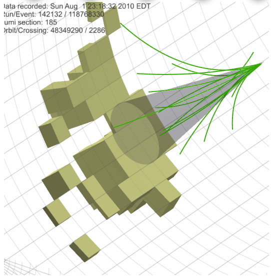
Detector



http://www.physics.umd.edu/rgroups/hep/LegoCMS/

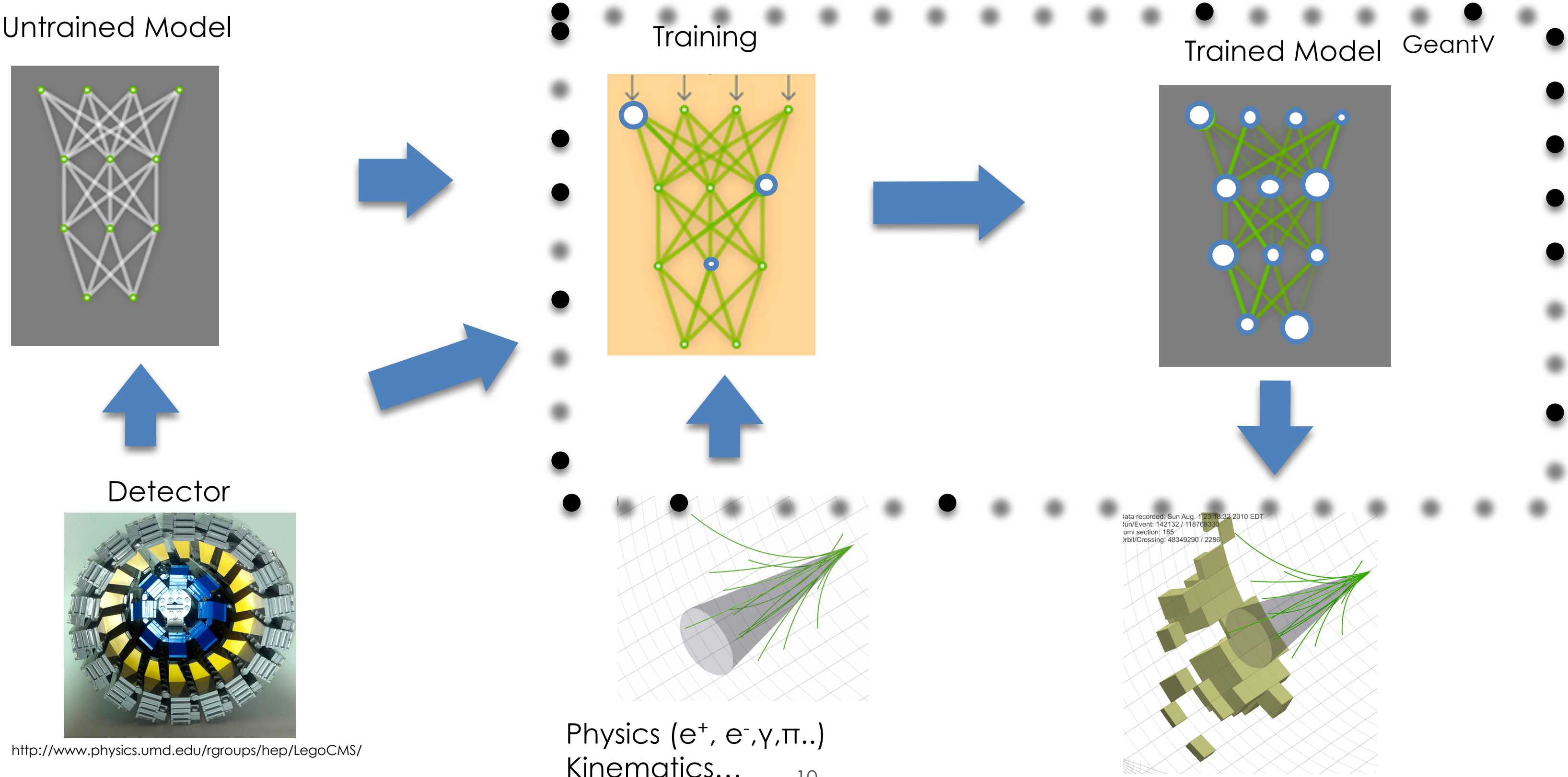


Physics (e^+, e^-,γ,π..)
Kinematics…



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

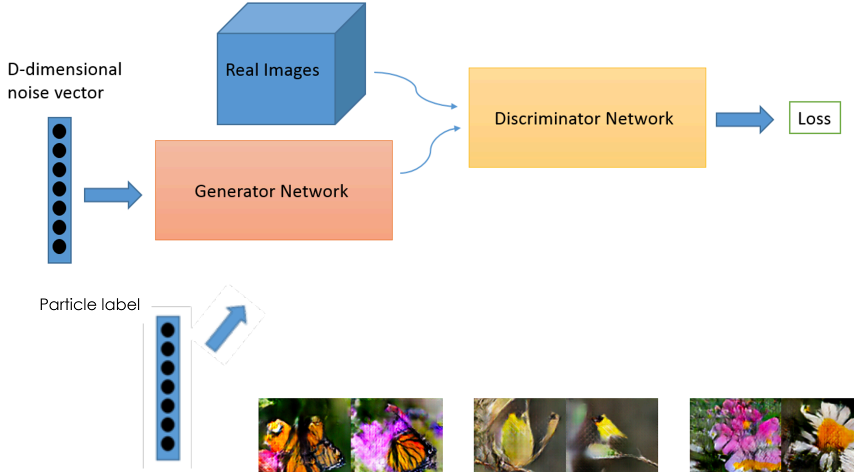
Phase 1: Proof of concept and interface in GeantV

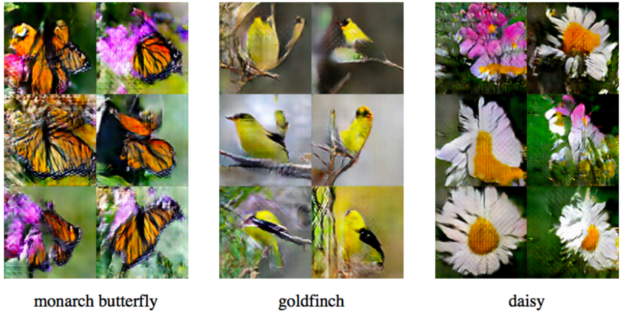
- Identify significant variables (PCA analysis, variable reduction)
- Test different ML and DL techniques
  - Generative adversarial networks
  - PCT and MP for MO regression
- Focus on most time consuming detectors
  - Initially reproduce calorimeter showers
- Train networks on full simulation
  - Eventually test possibility of training on real data
- Integrate a generic interface in GeantV
- Automatic tool for training

Ex: testing GANs for calorimeter showers

- Simultaneously train two models:
- Generative model G to capture the data distribution
- Discriminative model D to estimate the probability that a sample came from training data rather than G
- The training procedure for G is to maximize the probability of D making a mistake

arXiv:1406.2661v1

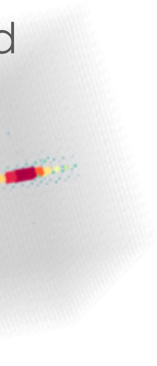




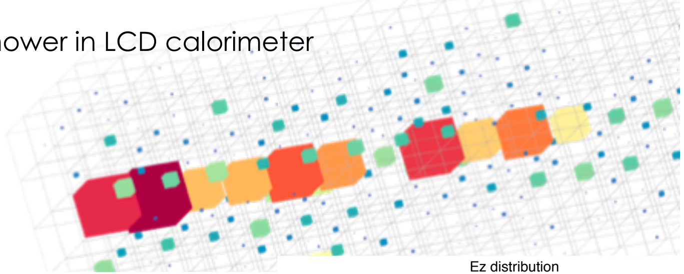
Conditional: arXiv:1610.09585v3

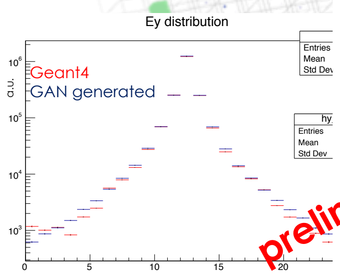
Ex: testing GANs for calorimeter showers

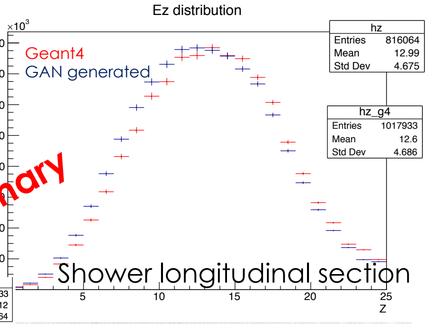
- High granularity LCD calorimeter single particle benchmark datasets simulated with Geant4[1]
- 3D convolutional models implemented using Keras + Tensorflow
- Batch training
- Test different optimisers (SGD, Adam, RMSprop)
- Working on model optimisation to improve physics description



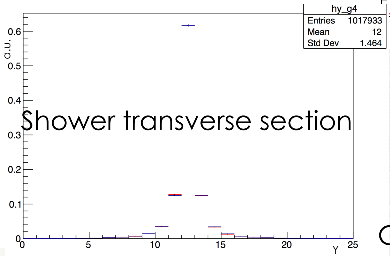
Geant4 π shower in LCD calorimeter



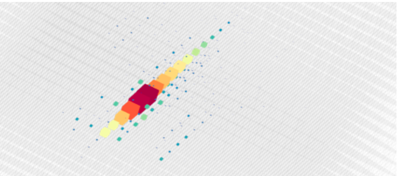




Shower longitudinal section



Shower transverse section



GAN generated electron shower

(1) https://indico.cern.ch/event/575212/contributions/2361407/attachments/1386217/2109575/MLchallengesForHEP_OpenLabDec2016.pdf

Optimising training time

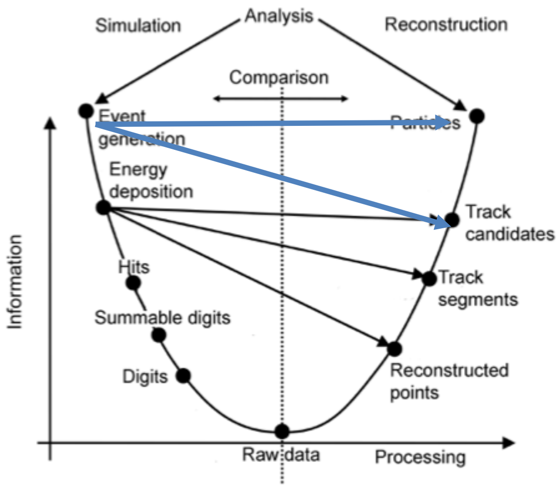
- Using DL techniques for fast simulation is profitable if training time is not a bottleneck
  - Currently adversarial training of the generative models takes a few hours on NVIDIA GTX1080 (Pascal)
  - Study and optimise algorithm
- Test different hardware
  - Test on a single KNL node and measure multi-threading speedup, memory footprint …
- Test multi-node scaling
  - Thanks to a collaboration with CINECA and Intel, we have access to a cluster of KNL

Phase 2: Optimization and training on clusters

- We want to provide a generic, fully configurable tool
  - Optimal network design depends on the problem to solve
    - Need embedded algorithms to perform hyper-parameters tuning and meta-optimization
    - Scan large hyper-parameter space
  - Need to improve training time by parallelization on large clusters
    - Evaluate existing libraries and improve scaling of training process on distributed systems
    - Optimize training strategy by reducing communication overheads

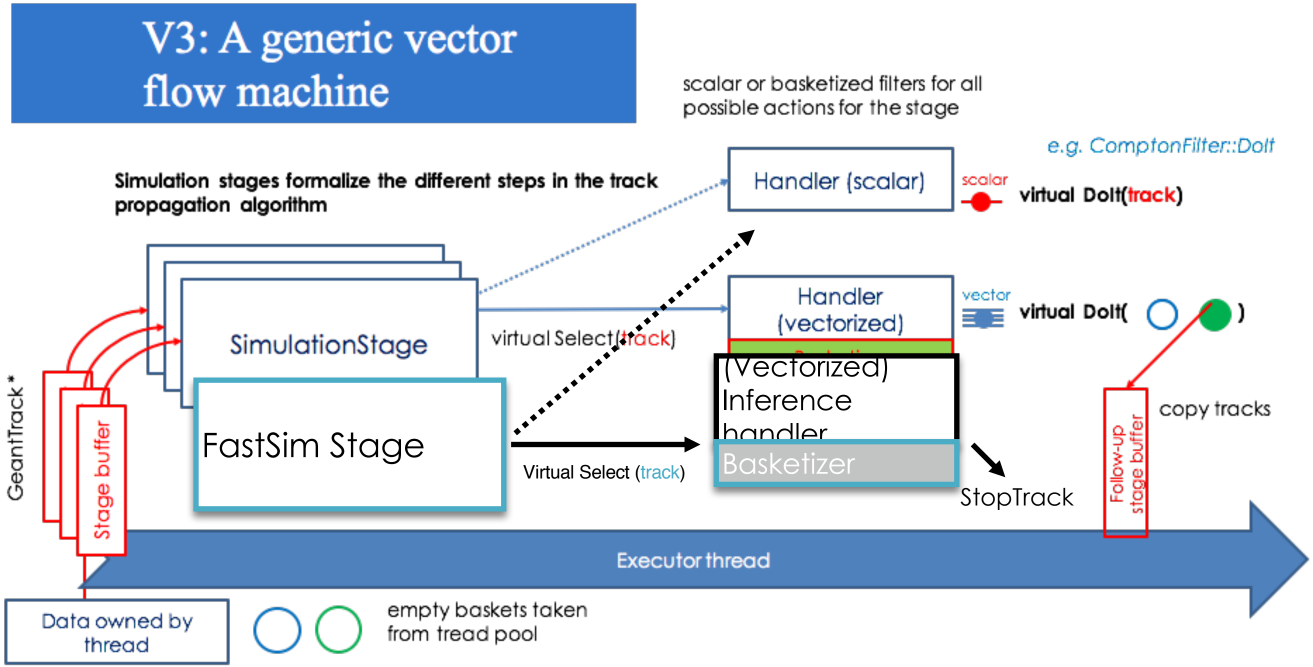
Summary

- **Ambition: to have the first ML prototype engine for fast simulation ready and fully integrated in GeantV by the end of 2018 (GeantV beta)**
  - We are testing different models and techniques in order to achieve the best possible physics results
  - We also keep in mind computing efficiency and insure optimal performance on modern hardware
  - Test inference step on dedicated hardware
- Even larger speedup gained by replacing digitization and reconstruction steps



Thank you

GeantV framework



GeantV for HPC environments

- GeantV can run in many-nodes and multi-sockets modes
- NUMA aware
- Standard 1 process per node or multi-event server mode for better work balancing available (MPI based)