Fast Simulation with GANs

Details and Benefit for Simulation Activities in ATLAS

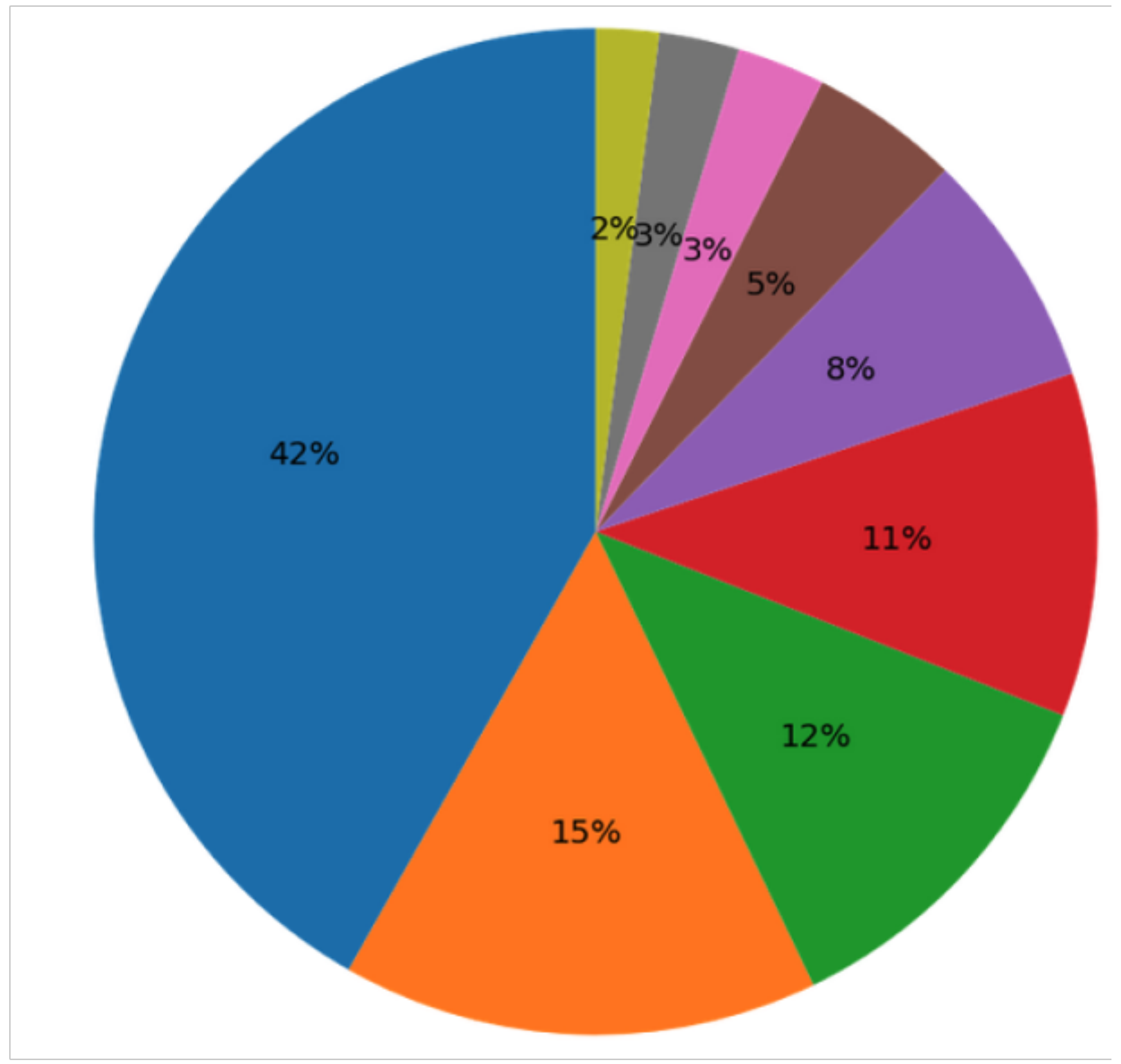
Federico A.G. Corchia
Alma Mater Studiorum
University of Bologna
INFN Bologna
The ATLAS Collaboration

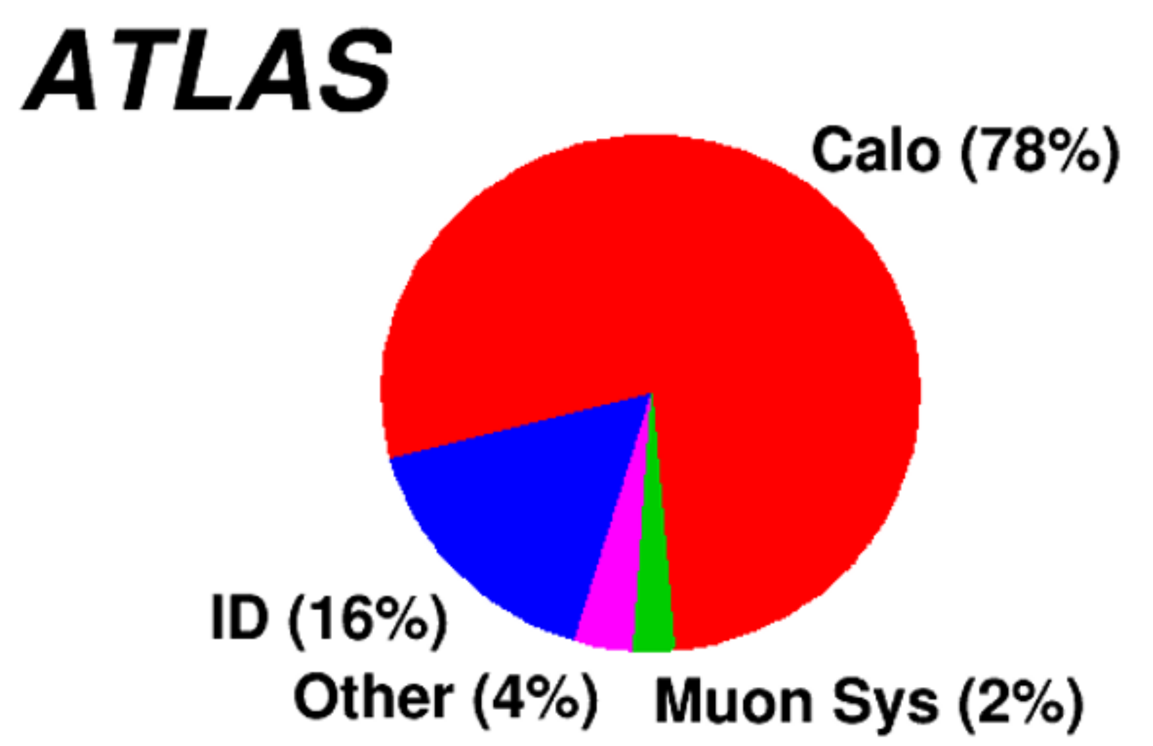
Genoa, 29 November 2023

General Context

Simulation in ATLAS: Full Vs Fast

- Detector simulation is a fundamental part of ATLAS activities.
- It's very heavy though, taking **~half of the total workload** on computing resources!
  - of which **simulation of calorimeters** takes ~80%;
  - only expected to increase in the future!
  - when aiming at better computing performance, calorimeter simulation becomes the logical **burden to attack!**
- **Fast simulation** tools perform calorimeter simulation **much faster** than full (i.e. "as usual") ones.
  - good accuracy must be kept;
  - may employ ML techniques (e.g. Generative Adversarial Nets, GANs).





General Context

Fast Simulation in ATLAS: ATLFAST3

- ATLFAST3 is the ATLAS detector fast simulation system created for Run 3. **It wraps various tools** to run fast simulation: depending on the particle type and energy, it chooses one of these tools, i.e. the one performing best for that case, to run simulation.

- **Tools**:

  - **FastCaloGAN** (GAN-based);

  - FastCaloSim (parametrisation-based);

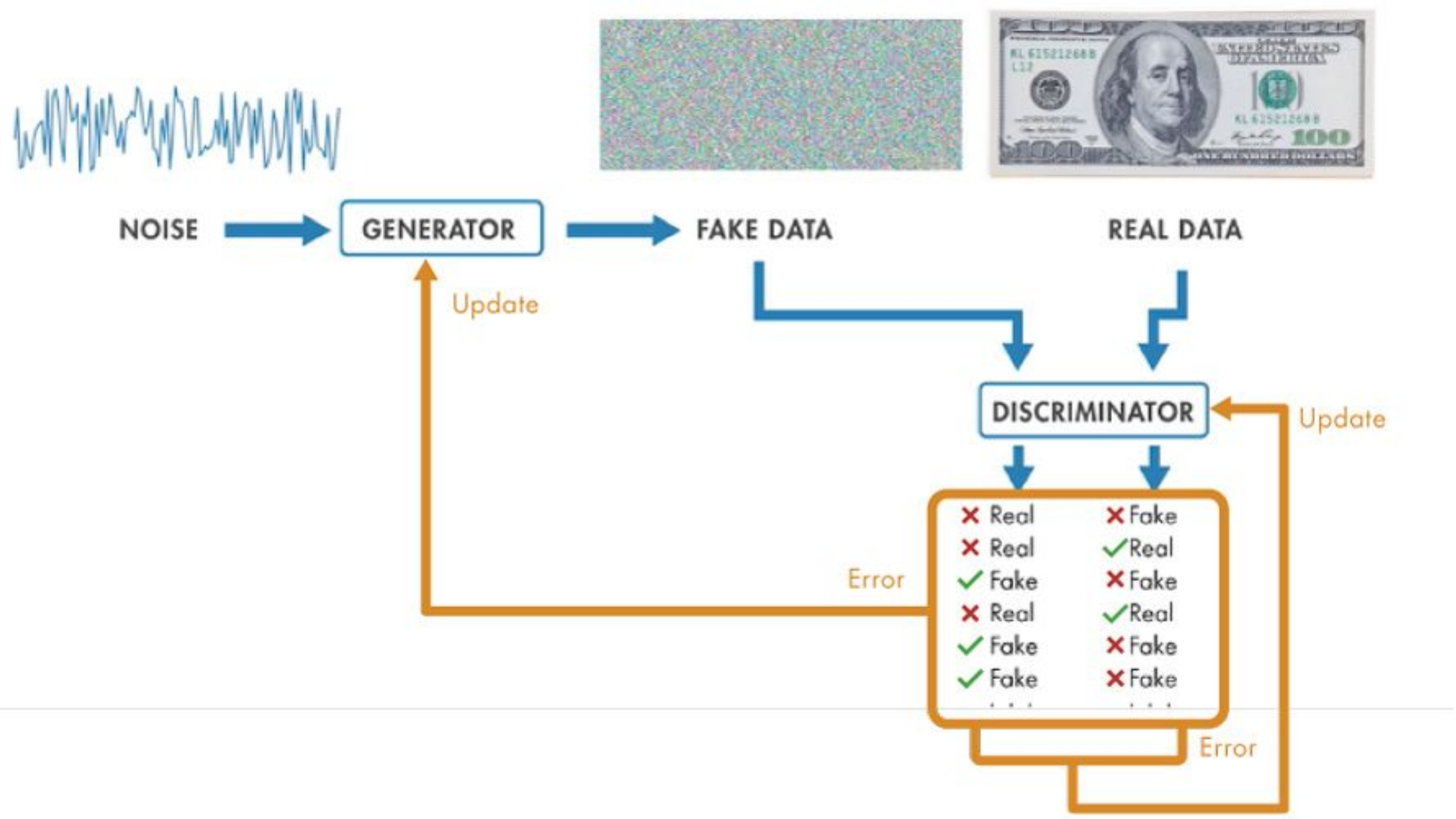
  - Geant4 (full simulation tool, only limited to specific cases e.g. muons and the rest of the detector).

Basics on GANs

- Introduced in Goodfellow, I.J. et al., "Generative Adversarial Nets", Advances in Neural Information Processing Systems (2014).

- **Training samples** + **generator** aiming at generating samples replicating the distribution of training data + **discriminator** getting samples and determining where they come from (generator or training data?).

- Simultaneous training of generator and discriminator. The generator must maximise the probability for the discriminator to believe a sample from the generator is real data, the discriminator must make this hard for the generator.



Fast Simulation with GANs: FastCaloGAN

- FastCaloGAN is the GAN-based tool for fast simulation in ATLAS.

- Training data = Geant4 data. The generator must learn how to produce Geant4-like data, the discriminator must distinguish generator-produced data from actual Geant4 data. When finished training, **FastCaloGAN simulates calorimeter response as similarly as possible to Geant4** (in the limit of this being a fast simulation and not a full one), but **much faster**!

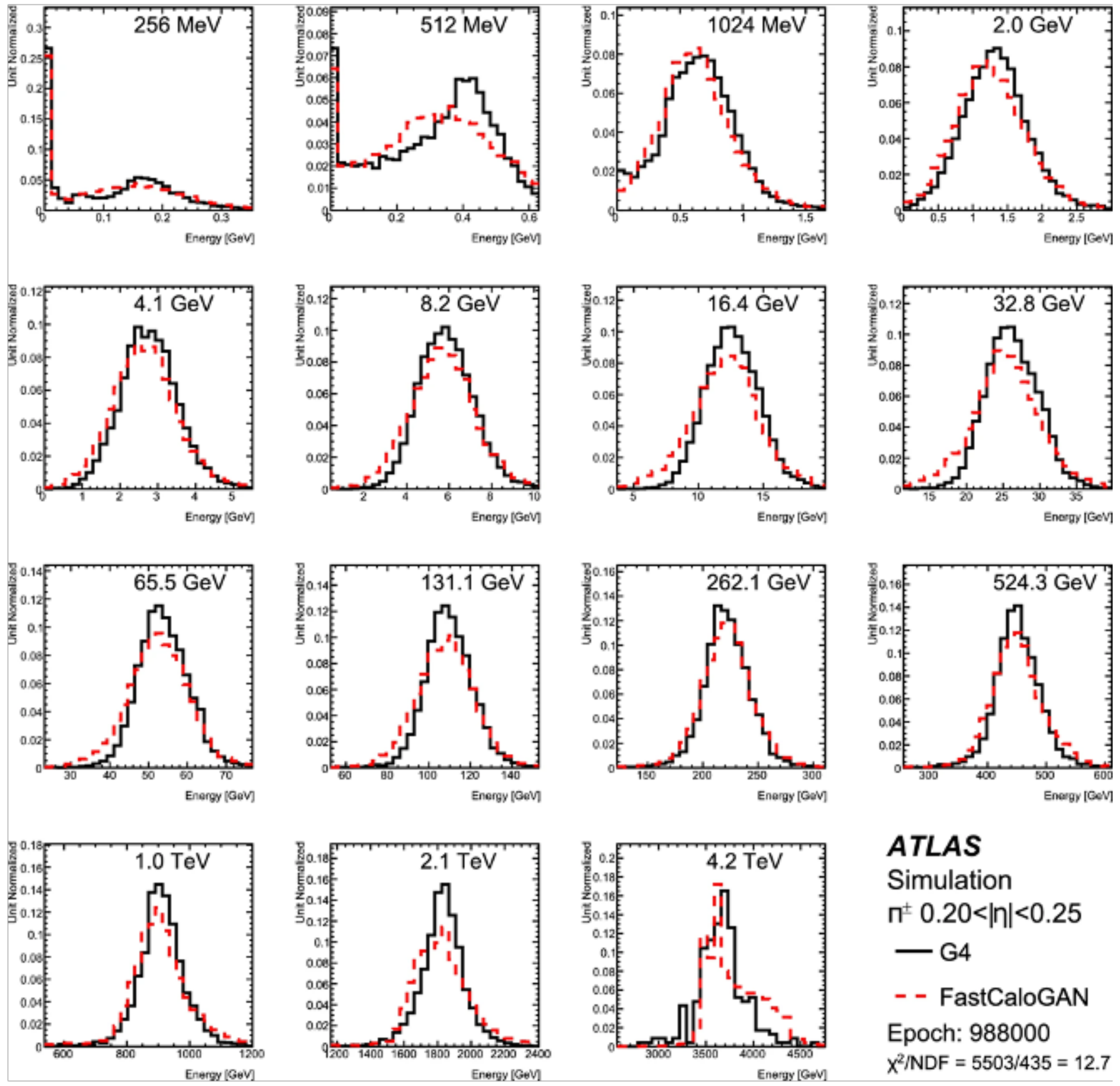
- Currently able to simulate calorimeter showers for photons, electrons, pions and protons between 256 MeV and 4 TeV over full detector acceptance (protons only at $-0.25 \leq \eta \leq 0.25$).

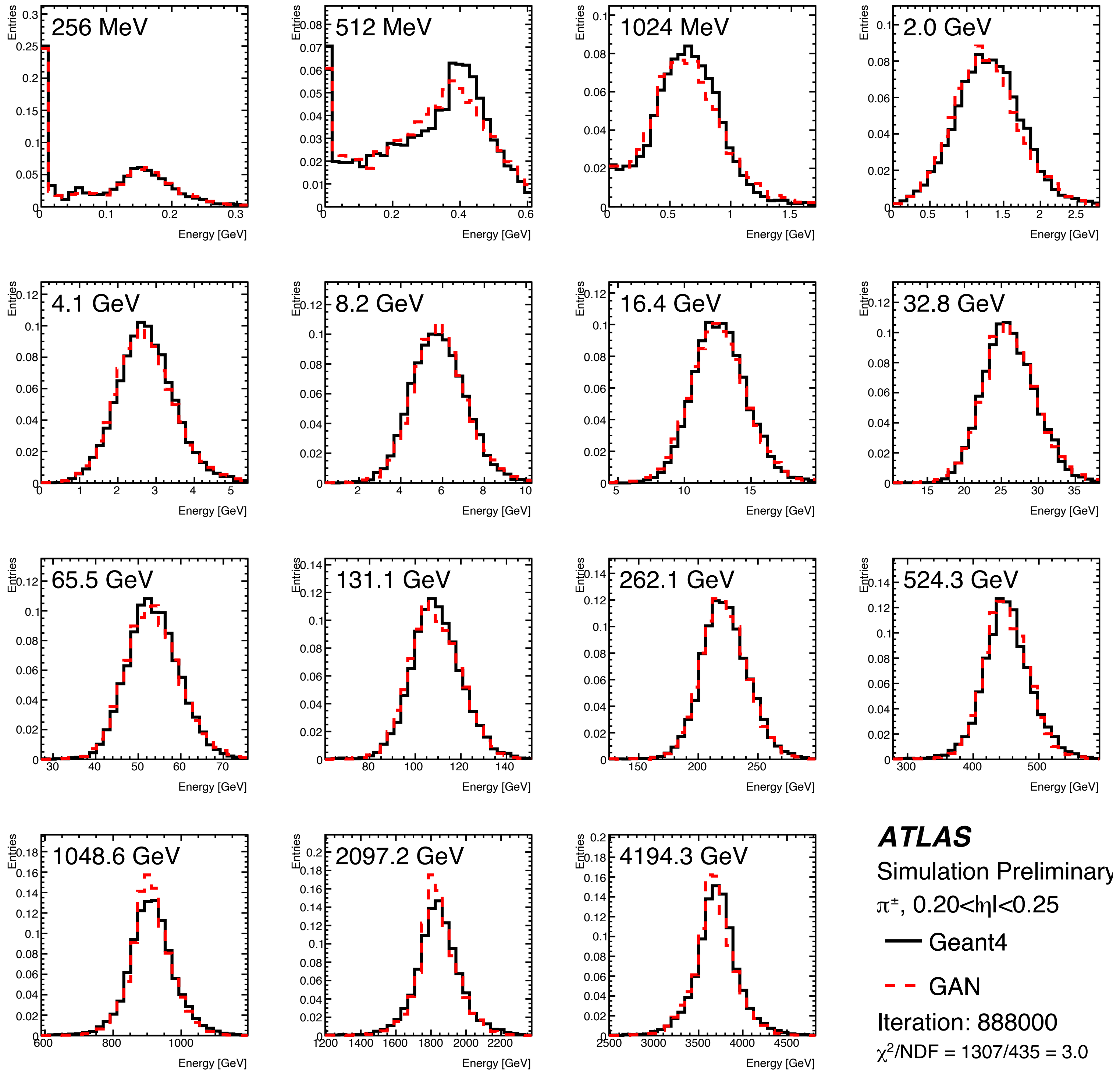
FastCaloGAN

Validation For Run 3

- FastCaloGAN simulations are validated by comparing its simulated data to the ones of Geant4, which acts as a reference.

- Geant4-FastCaloGAN comparison for pion simulation, with the old version of FastCaloGAN (left) and the latest one (right). Remarkable improvement - also observed for photons and electrons.





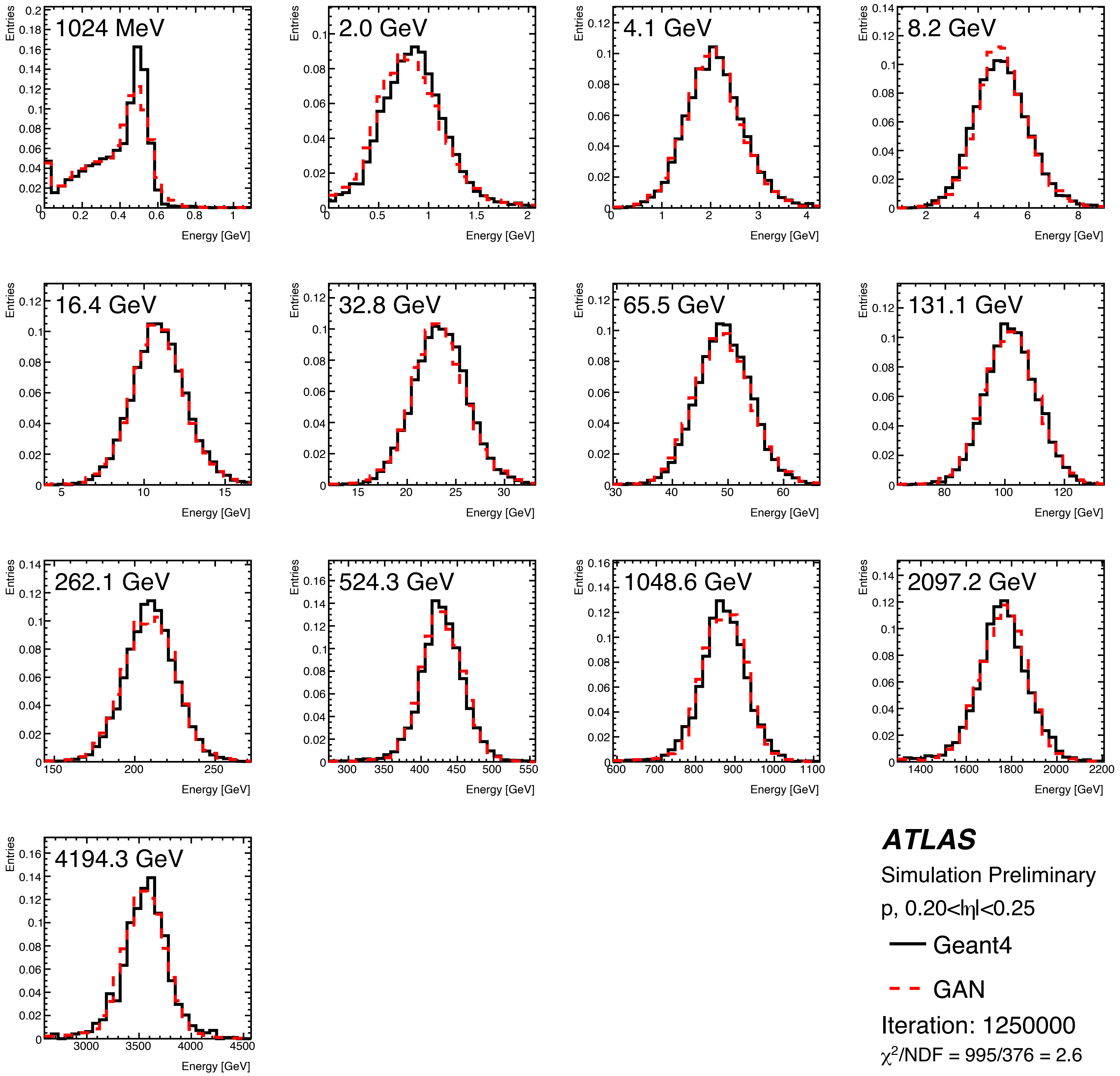
- Plot References
  Left: The ATLAS Collaboration, "AtlFast3: The Next Generation of Fast Simulation in ATLAS", Comput Softw Big Sci 6, 7 (2022)
  Right: Public plots at atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2022-001/

FastCaloGAN

Validation For Run 3 (cont’d)

- Geant4-FastCaloGAN comparison for proton simulation (proton handling has been introduced in the latest version). **Already remarkable results!**

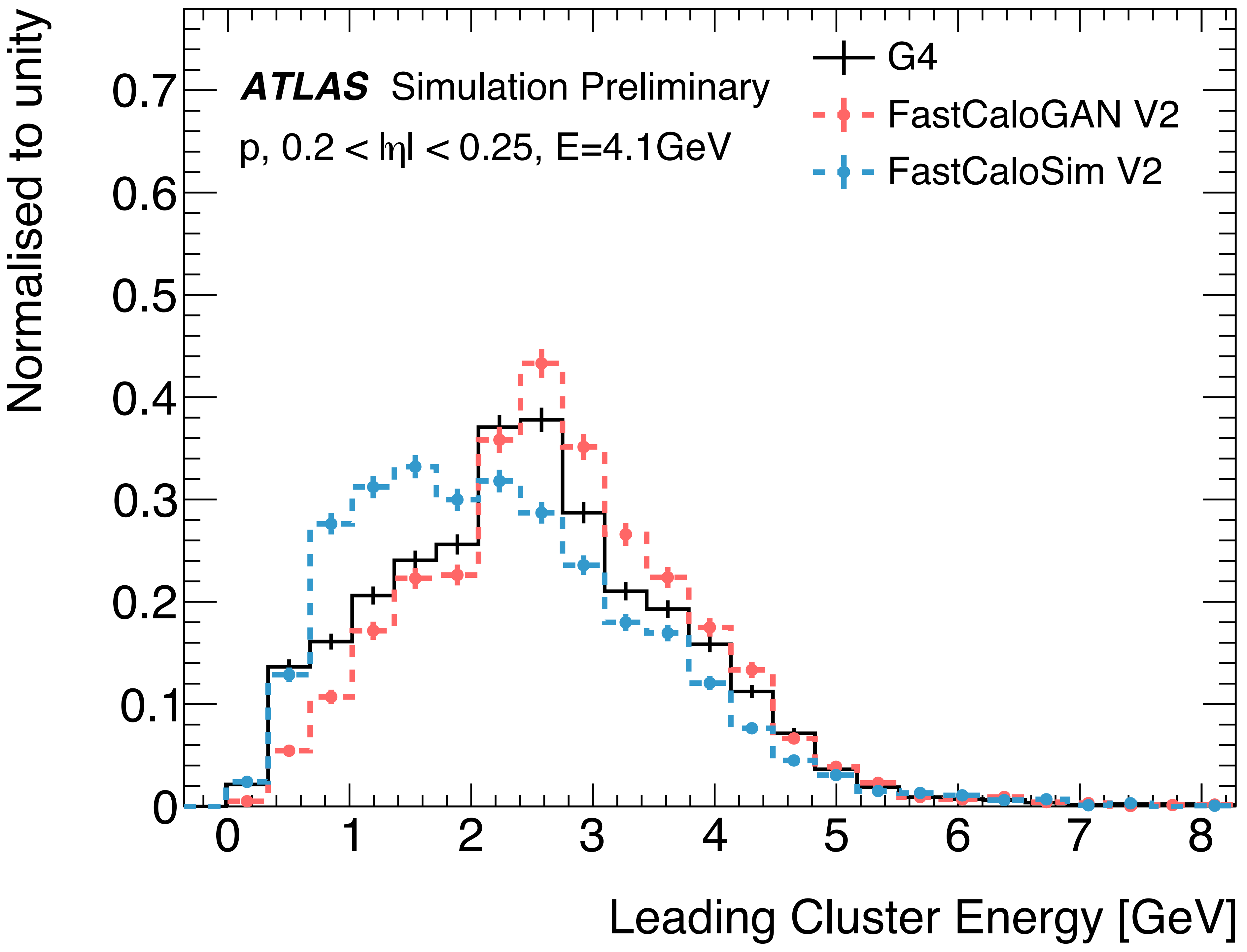
- Plot Reference
  Public plots at atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2023-004/



FastCaloGAN

Validation For Run 3 (cont'd)

- Geant4-FastCaloGAN-FastCaloSim V2 comparison for proton simulation. **FastCaloGAN produces better results** than FastCaloSim V2 (the ATLFAST3 parametrisation-based tool), also correcting the large discrepancy at low energy.

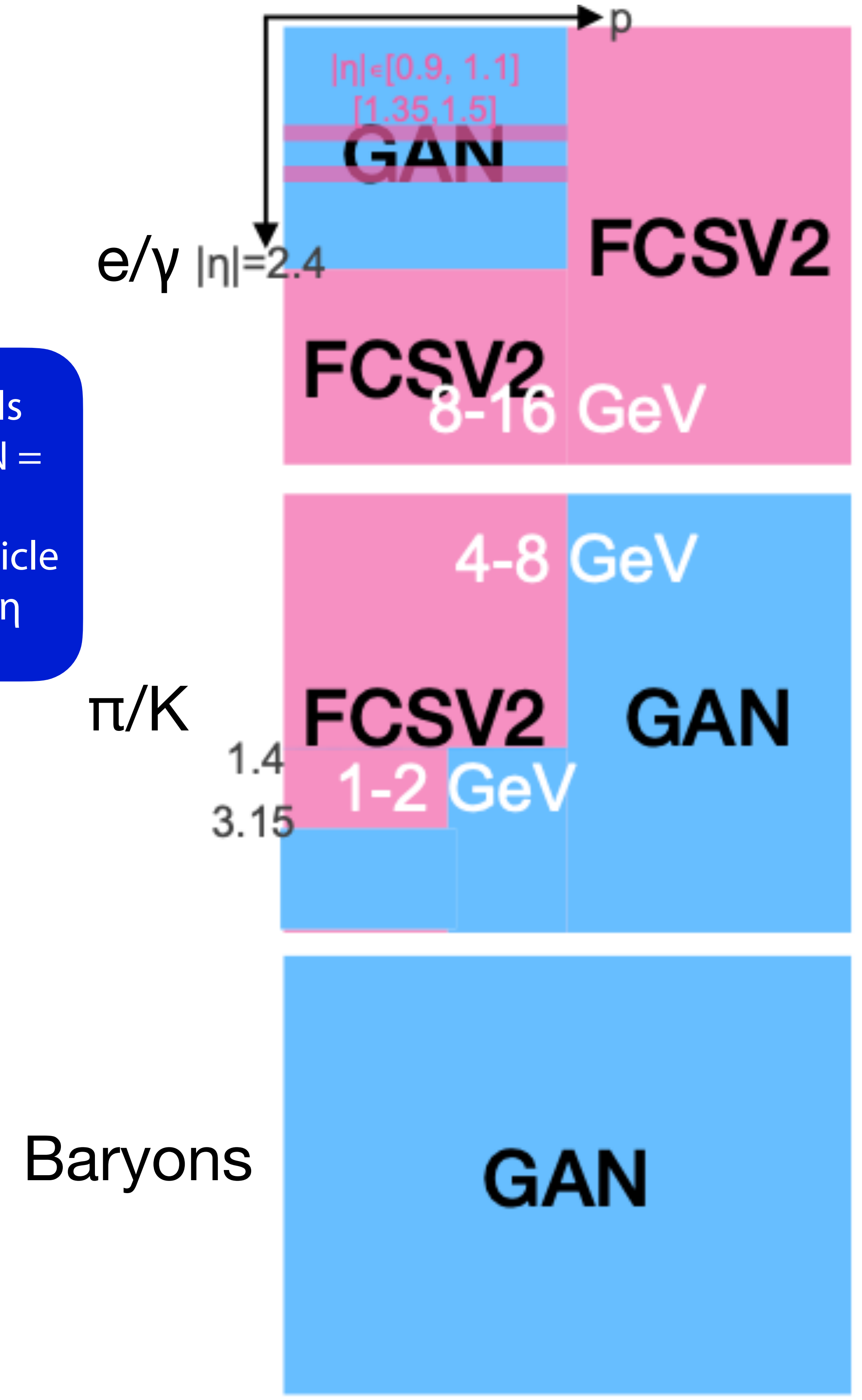


- Plot Reference
  Public plots at atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2023-004/

FastCaloGAN
Validation For Run 3 (cont'd)

- **ATLFAST3, including FastCaloGAN, has been accepted into ATLAS Run 3 simulation chain!**
- Right: AtlFast3, depending on particle type and energy, employs FastCaloSim or FastCaloGAN depending on which one yields the best results. Baryons are fully modelled by FastCaloGAN and also the other particles are for a good part!

Which one of the two tools (FCSV2 = FastCaloSim, GAN = FastCaloGAN) is used for simulation of the given particle type at given energy and η



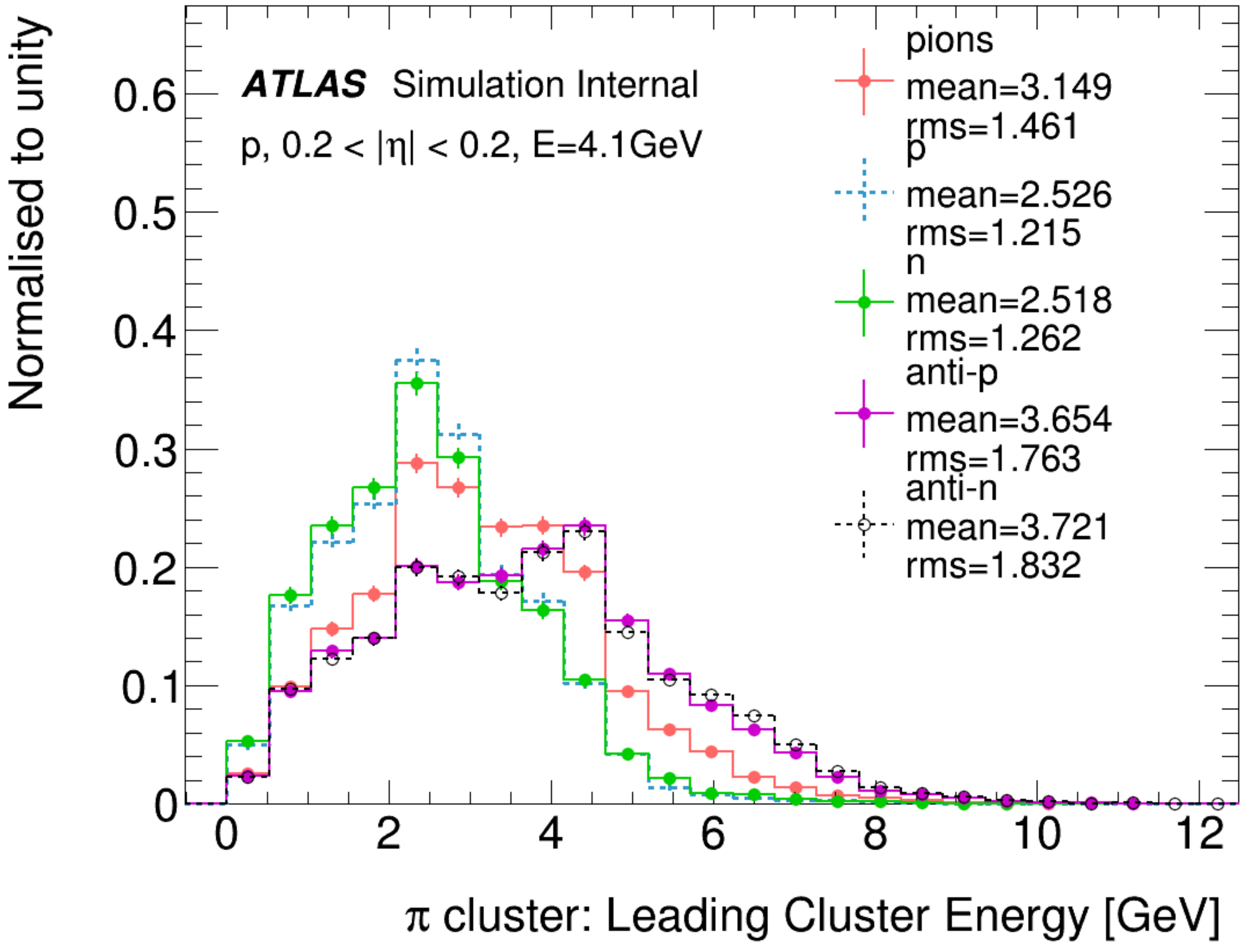
Further Development Ideas

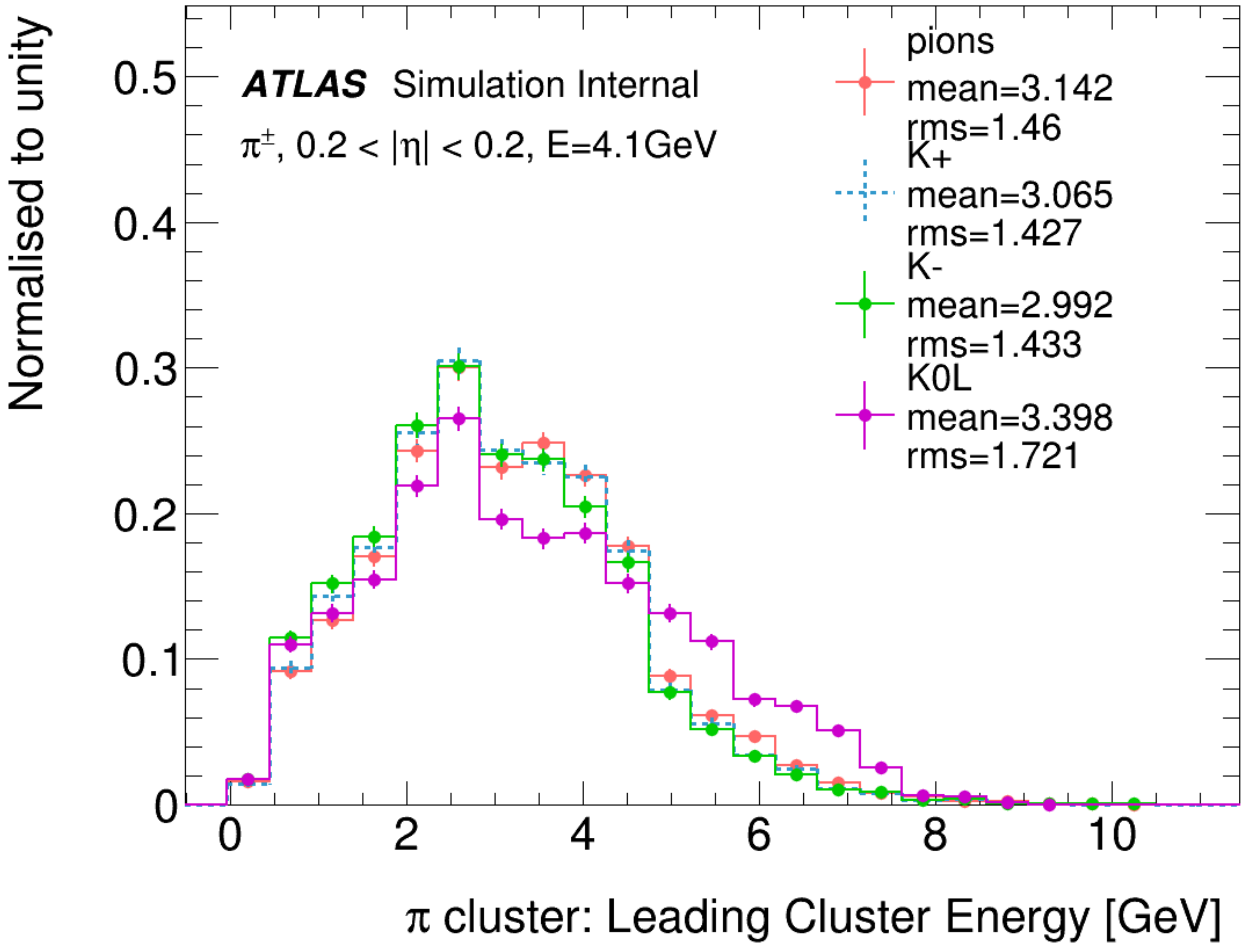
- After inclusion of FastCaloGAN/ATLFAST3 into ATLAS Run 3 simulation chain, there's obviously still room for improvement!
    - New particle types and generally use cases to add, better voxelisation, better GAN hyperparameters…
    - New additions are expected to be deployed by the end of Run 3 (~2025), when reprocessing Run 3 data/MC.
    - Shall also be the basis for fast simulation in Run 4 (ATLFAST4).

FastCaloGAN

Further Development Ideas (cont'd)

- One of the ideas for further development is **extending to further hadrons** (n, anti-p, anti-n, K^+, K^-, K^0_L).
- Before attempting GAN training, it's reasonable to have a little inspection on the current situation, i.e. on how these hadrons are currently modelled with full simulation, to collect ideas on what to care of when preparing fast simulation.
- Right: comparison of Geant4 simulations of the new hadrons of interest (10k events per particle type at 4.096 GeV).
  - With this comparison we can see if it's reasonable to try using GANs trained on a given particle to simulate another one (e.g. π GANs to simulate K). It is when their Geant4's are similar. This check has also been done all throughout FastCaloGAN development for Run 3.





FastCaloGAN on HPC

- Further improvement and relief on current resources may come not only from software requiring less resources, like fast simulation tools, but also from running them on **other clusters than LXPLUS!**

- FastCaloGAN is designed to run on CERN resources:

  - CentOS7 (Intel x86_64 cores) with V100 GPUs;

  - Managed by HTCondor scheduler.

- To have it run on other more powerful clusters, including general usage (i.e. non-HEP-specific) supercomputers, it must be ported → **containers**!

FastCaloGANtainer

- We're making a **container** prototype, FastCaloGANtainer, so that FastCaloGAN and so its analysis may run on other clusters.
- Apptainer (no need to be a privileged user) container replicating LXPLUS environment (CentOS7+ CVMFS + CUDA driver/libs):
  - Built from CentOS7;
  - Local FS within container. Data directory bind-mounted at runtime, FastCaloGAN code directory installed inside the container;
  - Installation of CUDA, cvmfs-exec (unprivileged, also including runscript for cvmfs-exec start if CVMFS is not present on host node), other needed sw utils.
- Container image able to run interactively or within batch scheduler (SLURM or HTCondor).

FastCaloGANtainer Testing

- FastCaloGANtainer successfully run on 3 testbeds.
- Computing performance on GPUs (INFN-CNAF) comparable with CERN machines (~6 hours).
- Much slower on single CPU machines: ~12 hours (UniBO OPH).

Resource	Details	Testing Situation
INFN-Bologna Cluster	INFN Bologna computing cluster. (Old) nodes w/ CentOS7, CVMFS, HTCondor, no GPUs	Done and successful
OPH Cluster	Cluster of the Open Physics Hub Project of the University of Bologna. Several nodes w/ Rocky Linux 8, SLURM, no CVMFS, no GPUs	Done and successful
INFN-CNAF	INFN-CNAF HPC cluster (close to WLCG INFN-T1) CentOS7 nodes, no CVMFS, SLURM, V100 GPUs	Done and successful
LEONARDO	The 4th most powerful supercomputer in the world, at CINECA in Bologna	To do. Agreement being established between INFN and CINECA: production access to resources expected shortly
Cloud Resources	(e.g. AWS, Google) Only recommended when resource request peaks	To do

Further Development Ideas

- Improve integration with FastCaloGAN (i.e. tree structure of the code and parameter handling) and parallelism (multi-CPU, multi-GPU);
- Test on more resources (possibly with GPUs) and architectures (i.e. ARM, very challenging);
- Run FastCaloGANtainer without CVMFS dependencies (Root, Python) for isolated nodes;
- Make FastCaloGANtainer publicly available (Docker Hub) with documentation.

Acknowledgements

- FastCaloGAN Creation, Training and Validation:
  - Michele Faucci Giannelli (University and INFN, Rome Tor Vergata)
  - Rui Zhang (University of Wisconsin Madison)
  - Joshua Falco Beirer (Georg August Universität Göttingen)
  - Michael Dührssen-Debling (CERN)
  - Christopher Young (CERN)
  - Henry Ann Day-Hall (Czech Technical University in Prague)
  - Dalila Salamani (CERN)
  - Josef Sorenson (University of New Mexico)
  - Marilena Bandieramonte (University of Pittsburgh)
  - Tommaso Lari (Università degli Studi and INFN, Milan)
- FastCaloGANtainer Creation and Testing:
  - Lorenzo Rinaldi (University and INFN, Bologna)