

# Deployment of ATLAS Calorimeter Fast Simulation Training Through Container Technology

Federico Andrea Guillaume Corchia<sup>1,2</sup>, Lorenzo Rinaldi<sup>1,2</sup>,  
Joshua Falco Beirer<sup>3</sup>, Rui Zhang<sup>4</sup> and Michele Faucci  
Giannelli<sup>5,6</sup>

<sup>1</sup>Department of Physics and Astronomy, University of Bologna, Bologna, Italy

<sup>2</sup>INFN Bologna, Bologna, Italy

<sup>3</sup>European Organization for Nuclear Research (CERN), Geneva, Switzerland

<sup>4</sup>University of Wisconsin-Madison, Madison, United States

<sup>5</sup>INFN Rome Tor Vergata, Rome, Italy

<sup>6</sup>University of Rome Tor Vergata, Rome, Italy

E-mail: corchia@bo.infn.it

**Abstract.** Detector simulation uses the largest fraction of computational resources of the ATLAS experiment at the LHC, with about 80% of that only used for simulation of calorimeters. Interventions to ease this burden are necessary for HL-LHC. A solution, already in production, involves the usage of Machine Learning-based fast simulation tools, capable of simulating calorimeter response faster than traditional tools but preserving good accuracy. However, these systems require large resources in the training phase, therefore a further strategy involves the deployment of fast simulation training to resources external to CERN, with the opportunity to have an additional boost in computing performance. This work introduces BoloGANtainer, the containerized version of the training part of the Machine Learning-based fast simulation tool used in the ATLAS Collaboration. This paper contains a description of how the tool can be trained on more powerful devices like High Performance Computing clusters and how it reduces software dependencies on local or distributed file systems. The testing methodology is also presented, together with results obtained on different resources with different operating systems and installed software, with or without GPUs.

## 1 Introduction

The simulation of the detector response is a major computing challenge at the ATLAS experiment [1] at the LHC, as shown in Fig. 1 [2]. In particular, simulation of calorimeters is the most demanding part, as it alone takes about 80% of the resources used by detector simulation [3] and is expected to increase with the forthcoming start of the High Luminosity LHC (HL-LHC). Therefore, solutions to ease this burden are particularly needed.

A solution already developed involves the usage of *fast simulation* tools mainly based on Machine Learning, able to simulate calorimeter response faster than the *full simulation* software Geant4 [4, 5, 6], preserving at the same time good accuracy. In particular, the ATLAS Collaboration has developed AtlFast3 [3] as its fast simulation tool, which combines a classic parametric approach with one based on Generative Adversarial Networks (GANs) [7, 8]. This tool is already in production for Run 3 of the LHC.

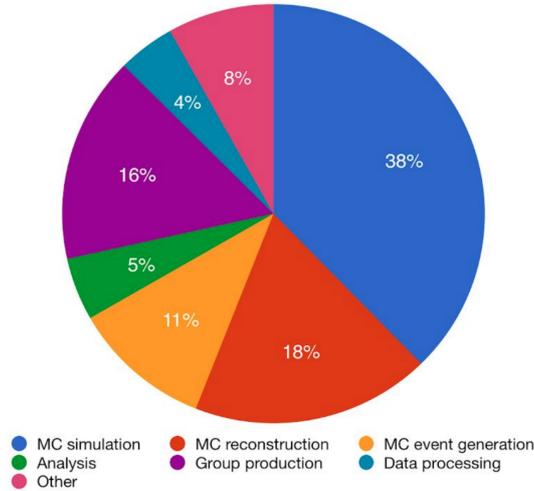


Figure 1: CPU-hours usage by ATLAS activities [2].

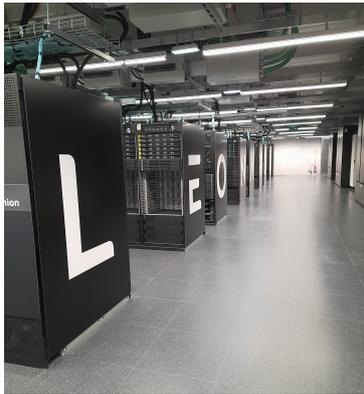


Figure 2: The Leonardo supercomputer at CINECA in Bologna, Italy [9].

However, GANs require a large amount of resources to be trained. As a consequence, a further idea involves distributing this training on other resources than the ones commonly used at CERN (the CERN batch system LXBATCH and the Worldwide LHC Computing Grid [10]), in order to obtain more space for training and, if cutting-edge resources such as supercomputers are used, a remarkable performance boost. For this reason, *BoloGANtainer* has been developed.

## 2 Technical Details on BoloGANtainer

The BoloGANtainer system containerizes AtIFast3 GAN training so that it may be deployed on other machines than usual ones. BoloGANtainer is based on a container developed with Apptainer [11] and extends the official ATLAS CentOS 7 image, which replicates the (current until end of June 2024) operating system and software of the CERN batch system. It is independent from CVMFS, but requires CUDA 11 [12] and cuDNN [13] for GPU usage. The rest of the required software is directly installed inside the container and not linked via CVMFS: in this way training can also run on nodes without CVMFS or Internet connection. The container recipe is shown in Appendix A.

## 3 Performance

BoloGANtainer has been deployed on the clusters shown in Table 1. Training on Leonardo (Fig. 2) runs in about half the time needed on LXBATCH: remarkable speedup is observed thanks to NVIDIA A100 GPUs, showing how the usage of supercomputers brings great advantages.

Training performance is evaluated by means of a reduced  $\chi^2$  obtained from the comparison between

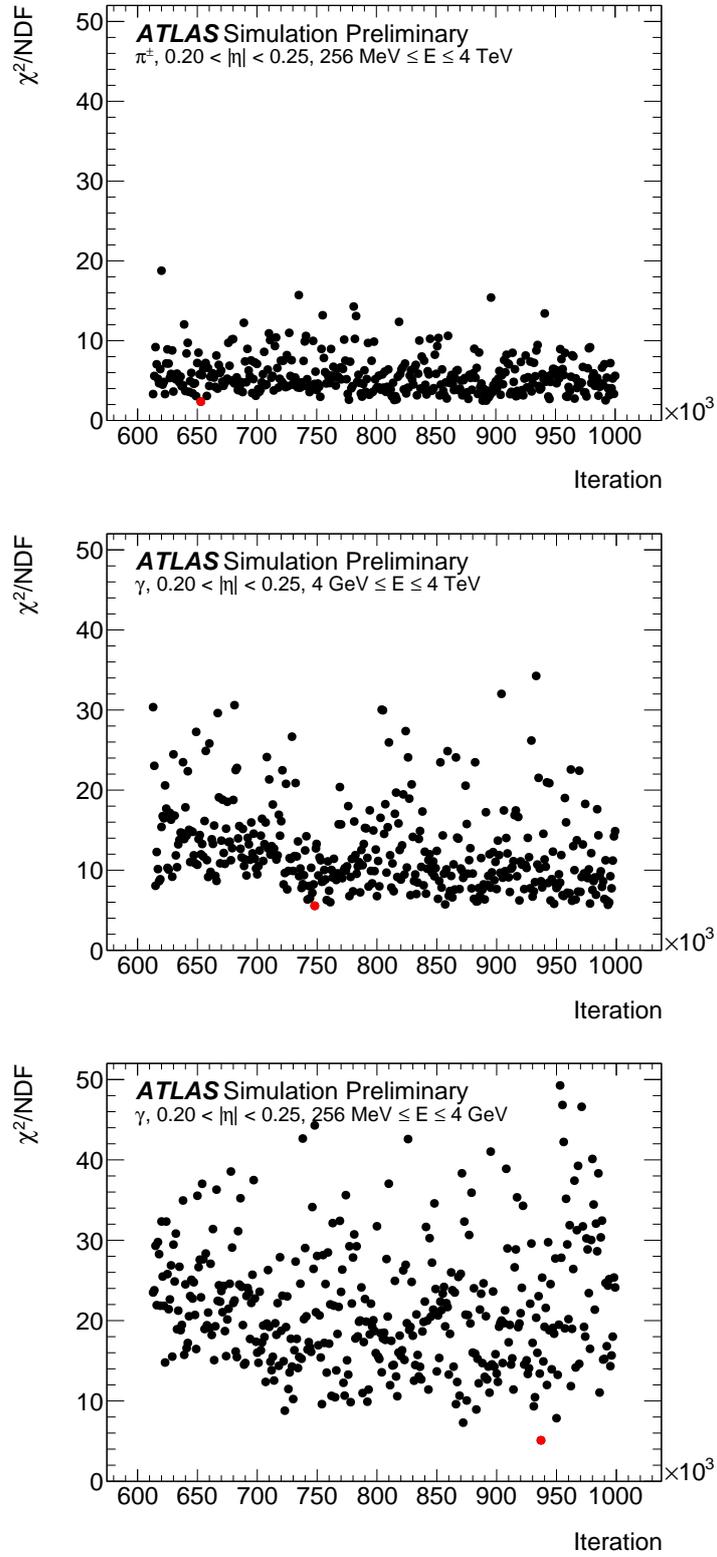


Figure 3: Reduced  $\chi^2$  per iteration of the GAN training in the pseudorapidity region  $0.20 < |\eta| < 0.25$  for (respectively, from top) pions, high energy photons (i.e. with energy above 4 GeV) and low energy photons (with energy up to 4 GeV) [14]. Training ran on the Leonardo supercomputer. The best iteration that minimizes the reduced  $\chi^2$  is shown in red.

Table 1: BoloGANtainer performance on tested clusters. For pions one single GAN was trained for all incident energy values, while for photons two GANs were, one for energies below or equal to 4 GeV and the other for energies above. NDF is the number of degrees of freedom.

Resource	Type and Owner	Hardware and Software	Pion Results	Photon Results
LXBATCH	CERN batch system, reference cluster	CentOS 7 (for nodes used), CVMFS, HTCondor, NVIDIA V100 GPUs	Runtime: 12 h $\chi^2/\text{NDF} \sim 2$	Runtime: 30-31 h $\chi^2/\text{NDF} \sim 5$
Leonardo [15]	The 10 <sup>th</sup> most powerful cluster in the TOP500 ranking [16] at CINECA in Bologna, Italy	RHEL 8.7, no CVMFS, SLURM, NVIDIA A100 GPUs, isolated nodes	Runtime: 6-7 h $\chi^2/\text{NDF} \sim 2$	Runtime: 10-11 h $\chi^2/\text{NDF} \sim 5$

reference histograms produced through Geant4 and the ones produced with the GANs. The plots in Fig. 3 show the reduced  $\chi^2$  per GAN training iteration on Leonardo. The best training iteration is the one that returns simulations as similar as possible to the ones of Geant4, thus to assess this the reduced  $\chi^2$  between the fast simulation sample and the full simulation one is calculated for the total energy distribution, and the iteration with the lowest reduced  $\chi^2$  is chosen as the best one and used to perform simulation activities. The plots show that training is stable for pions and high energy photons, while it is rather unstable for low energy photons. Analogous results are found with LXBATCH.

#### 4 Further Development

This work will be further developed using other types of resources than the ones on which BoloGANtainer has already been tested (including cloud resources); other architectures (ARM); and for more particle types. Optimization of the code shall be also studied, intended both as a general improvement and to employ multi-core and multi-GPU nodes in a more efficient way.

#### 5 Conclusion

Detector simulation takes the largest load of the computing resources of the ATLAS experiment at the LHC, with a major fraction taken by calorimeter simulation. Since this burden is relevant and will increase with the HL-LHC, solutions have been developed, the first one being Machine Learning-based fast simulation systems. As their training requires significant resources, a further solution involves the distribution of the training on various external resources. This work has presented BoloGANtainer as an effective tool to deploy the training of ATLAS fast simulation on external resources, with good performance and accuracy and a significant speedup if cutting-edge supercomputers are employed.

Copyright 2025 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license.

#### References

- [1] ATLAS Collaboration 2008 The ATLAS experiment at the CERN Large Hadron Collider *JINST* **3** S08003
- [2] ATLAS Collaboration 2020 ATLAS HL-LHC computing conceptual design report CERN-LHCC-2020-015 [cds.cern.ch/record/2729668](https://cds.cern.ch/record/2729668)
- [3] ATLAS Collaboration 2022 AtIFast3: the next generation of fast simulation in ATLAS *Comput. Softw. Big Sci.* **6** 7
- [4] Allison J *et al.* 2016 Recent developments in Geant4 *Nucl. Instrum. Meth. A* **835** pp 186-225
- [5] Allison J *et al.* 2006 Geant4 developments and applications *IEEE Trans. Nucl. Sci.* **53** pp 270-278
- [6] Agostinelli S *et al.* 2003 Geant4 - a simulation toolkit *Nucl. Instrum. Meth. A* **506** pp 250-303

- [7] ATLAS Collaboration 2020 Fast simulation of the ATLAS calorimeter system with generative adversarial networks ATL-SOFT-PUB-2020-006 [cds.cern.ch/record/2746032](https://cds.cern.ch/record/2746032)
- [8] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks *Advances in Neural Information Processing Systems* **3** 11
- [9] INGV 2022 LEONARDO | Il supercomputer al servizio delle geoscienze [ingv.it/it/stampa-e-urp/stampa/comunicati-stampa/5269-leonardo-il-supercomputer-al-servizio-delle-geoscienze](https://ingv.it/it/stampa-e-urp/stampa/comunicati-stampa/5269-leonardo-il-supercomputer-al-servizio-delle-geoscienze) (photograph by INGV - Istituto Nazionale di Geofisica e Vulcanologia)
- [10] The Worldwide LHC Computing Grid [wlcg.web.cern.ch](https://wlcg.web.cern.ch)
- [11] Apptainer [apptainer.org](https://apptainer.org)
- [12] NVIDIA CUDA [developer.nvidia.com/cuda-zone](https://developer.nvidia.com/cuda-zone)
- [13] NVIDIA cuDNN [developer.nvidia.com/cudnn](https://developer.nvidia.com/cudnn)
- [14] ATLAS Collaboration 2024 Public plots [atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2024-004](https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2024-004)
- [15] The Leonardo supercomputer [leonardo-supercomputer.cineca.eu](https://leonardo-supercomputer.cineca.eu)
- [16] TOP500 Ranking - June 2025 [top500.org/lists/top500/2025/06](https://top500.org/lists/top500/2025/06)

## A BoloGANtainer Recipe

Below is the Apptainer recipe for BoloGANtainer. After GAN training there is an evaluation part where the best GAN training iteration is chosen (see Section 3).

BootStrap: docker

From: atlas/centos7-atlasos

```
%setup
  mkdir ${APPTAINER_ROOTFS}/data

%files
  < copies GAN training code into the container >

%post
  rpm -Uvh https://packages.microsoft.com/config/centos/7/packages-microsoft-prod.rpm
  yum install -y dotnet-sdk-7.0 man man-pages libXpm gcc-c++ git root python3-root
  pip3 install --upgrade pip setuptools
  pip3 install tensorflow pandas

%runscript
  if [ "$1" = "init" ] ; then
    < prepares directory for output and logfiles >
  elif [ "$1" = "train" ] ; then
    if [ "$2" != "pions" ] && [ "$2" != "photons" ] ; then
      < warns about invalid option and exits >
    else
      < launches training script >
    fi
  elif [ "$1" = "bestiter" ] ; then
    if [ "$2" != "pions" ] && [ "$2" != "photons" ] ; then
      < warns about invalid option and exits >
    else
      < launches iteration evaluation script >
    fi
  else
    < warns about invalid option and exits >
  fi
```