PROCEEDINGS OF SCIENCE

Fast simulation with generative models at the LHC

Liza Mijović[a,1,*]

[a] *University of Edinburgh*

E-mail: liza.mijovic@cern.ch

The increasing integrated luminosity of the data collected at the major Large Hadron Collider experiments – ALICE, ATLAS, CMS and LHCb – necessitates increasingly large simulated samples. Given that the computational resources won't grow proportionally to the integrated luminosity, how can the experiments produce these large samples? A key technique the experiments use to address this challenge is replacing traditional detector simulation with generative machine learning models. These generative models achieve $O(10 - 1000)$ times improvements in computational efficiency while maintaining high accuracy. Specifically, I discuss four solutions: ALICE's simulation of Zero Degree Calorimeter with a Variational Autoencoder, ATLAS's use of Generative Adversarial Networks for calorimeter simulation, CMS's end-to-end FlashSim simulation based on Normalising Flows, and LHCb's Lamarr pipe-line employing Generative Adversarial Networks. The speed-up and physics performance achieved by these solutions cements the status of generative models as a viable, faster alternative to the established simulation techniques, which is an important step towards addressing the computational demands of the current and future LHC data analyses.

12th Large Hadron Collider Physics Conference (LHCP2024)
3-7 June 2024
Boston, USA

[1] For the ALICE, ATLAS, CMS and LHCb collaborations.
[*] Speaker

1. Why do the LHC experiments need fast simulation?

The detectors of the major LHC experiments: ALICE [1] ATLAS [2], CMS [3] and LHCb [4], are undergoing - or have undergone - upgrades targeting an order of magnitude higher luminosities compared to Run 2 (2015–2018). The analysis of these larger data-sets requires correspondingly larger simulated samples, which in turn require larger computing resources. The problem is that this large increase in computing resources, especially the processing power, is not financially viable; Figure 1 shows typical CPU usage of an LHC experiment; this is dominated by the detailed (also called full) simulation with Geant4 [5], followed by fast simulation, which alone saturates the pledged resources. To solve the processing power problem, the experiments therefore need to:

a) use fast simulation rather than the detailed simulation for a large fraction of events,
b) substantially speed-up fast simulation.

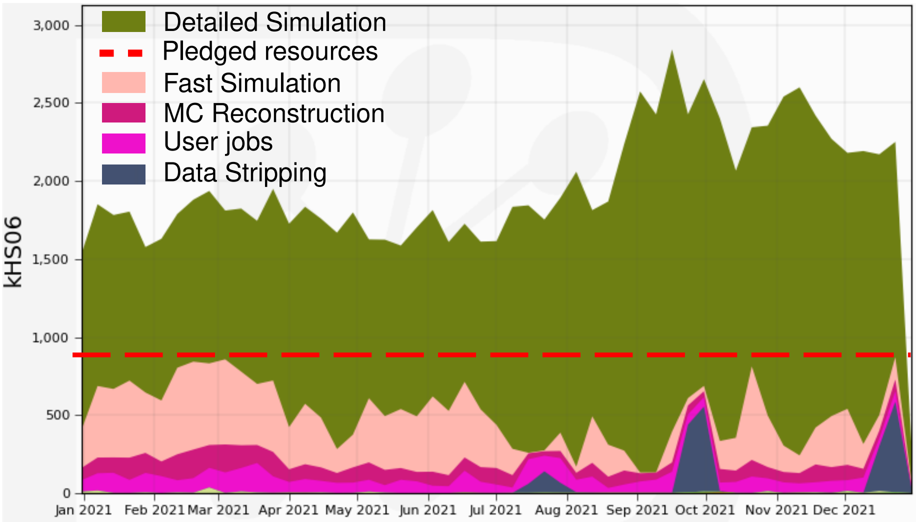


Figure 1: Usage of LHCb CPU power at Tier0/1s during 2021. Adapted from [6].

Both steps involve a trade-off between the speed and accuracy. In terms of these, two approaches have emerged:

- **fast simulation** of individual detector components in which the full simulation of the slowest detector component (typically calorimeter) is replaced by a tailored fast simulation.
- **ultra-fast simulation** which simultaneously replaces multiple simulation and reconstruction steps with fast approximations (a prototype of such approach is DELPHES [7]).

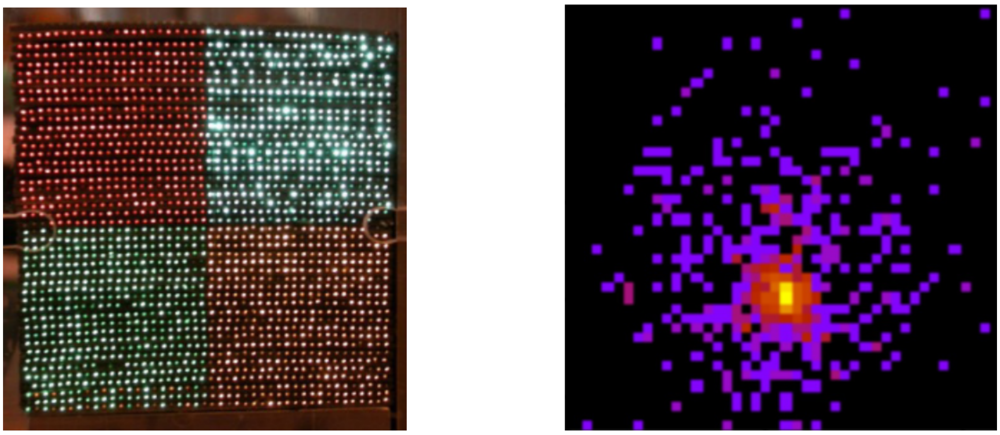
The fast simulation provides high accuracy and a limited speed-up, because the next slowest component of the production chain becomes the bottle-neck. On the other hand the ultra-fast simulation sacrifices some flexibility to deploy new object reconstruction, as it emulates the reconstruction at the time of developing the simulation model. All the experiments have deployed or developed both approaches. I limit the discussion to the work published by the experiments, because a collaboration endorsement means the approach is contributing to resolving the experiment's computing resources problem. I focus on the most recent incarnations which harness generative machine learning (ML) models - a technique of choice to simultaneously ensure high speed and accuracy.

2. Fast Simulation

I first discuss fast simulation, which aims to replace the simulation of the most CPU consuming detector with a generative model, whereas detailed simulation is used for the rest of the detector. For each of ALICE, ATLAS, CMS and LHCb, the simulation of the calorimeter was the bottle-neck, and all four experiments have presented a generative ML version of their calorimeter simulation. Typically, this results in $O(100)$ faster calorimeter simulation while retaining high accuracy, and a $O(10)$ speed-up of the detector simulation chain. I discuss the solutions by ALICE and ATLAS.

2.1 Fast Simulation of ALICE Zero Degree Calorimeter

ALICE has developed a fast simulation of the Zero Degree Calorimeter (ZDC), which measures energy of spectators – particles that did not directly participate in collision. The ZDC is a system of five sampling calorimeters placed about 120 meters upstream of the ALICE time projection chamber. The ZDC read-out consists of 44 × 44 fibers (Figure 2a), and the corresponding images are used as fast simulation inputs (Figure 2b).



(a) (b)

Figure 2: (a) ALICE ZDC read-out and (b) corresponding images used for fast simulation. Source: [8]

The best performing method is based on a Variational AutoEncoder (VAE), which consists of two networks: an encoder compressing the data to a latent space, and a decoder trying to reconstruct the original input data from this latent space. To improve the accuracy of the ZDC simulation, a **CorrVAE** architecture with two latent spaces is used, one for user-defined properties, and one for the rest [9]. In any detector simulation, the resulting images need to correspond to the images produced by the input particles. For the ZDC simulation, this conditioning is introduced by a third latent space encoding particle properties. The ZDC simulation based on CorrVAE is found to over-perform simulations based on VAEs and generative adversarial networks. A prototype has been integrated in ALICE's production chain, and delivers a 100-times speed-up of the ZDC simulation.

2.2 Fast Simulation of ATLAS Calorimeter

At ATLAS, the calorimeter takes up about 80% of the detailed simulation time [10]. The fast calorimeter simulation (AtlFast3) speeds it up through a hybrid of two approaches (Figure 3):

- FastCaloSim: a parametrised model, and

- FastCaloGAN: using generative adversarial networks **(GANs)**, a system of two networks in which the discriminator network is trained to distinguish the detailed simulation from the fast simulation samples produced by the generator network.

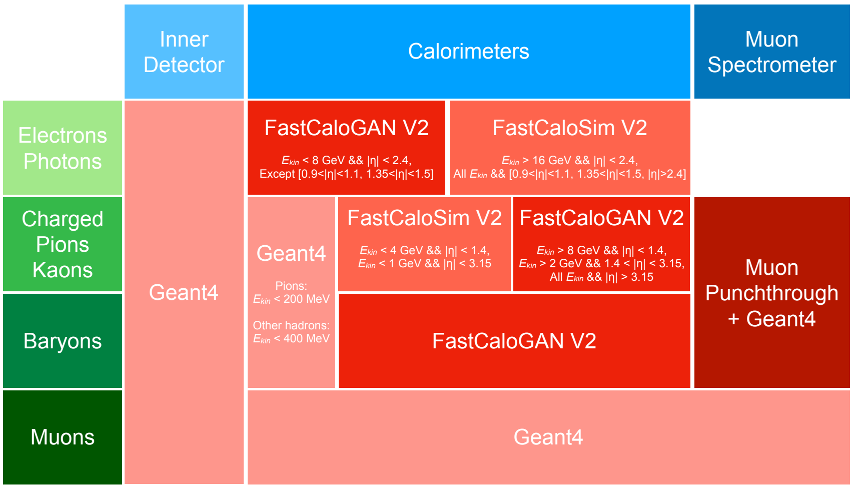
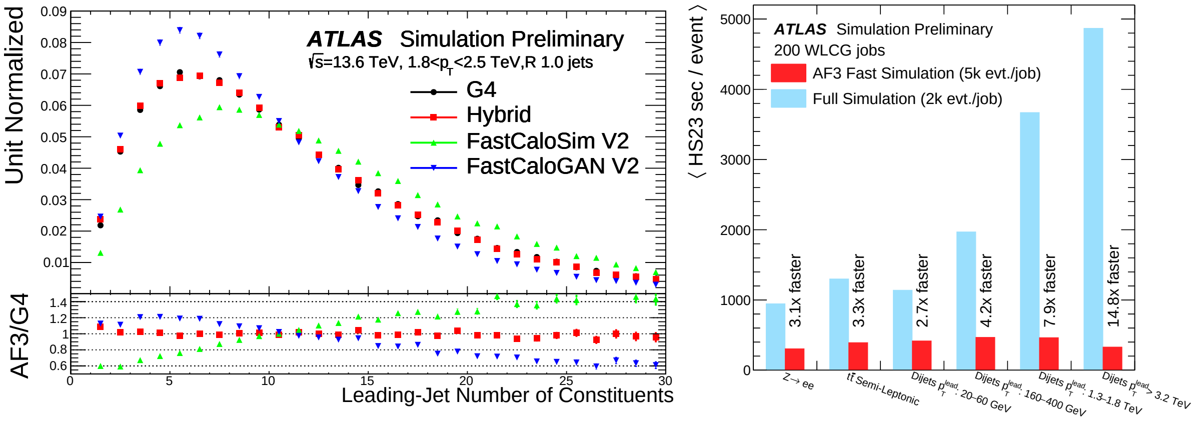

Inner Detector
Calorimeters
Muon Spectrometer
Electrons Photons
Charged Pions Kaons
Baryons
Muons
Geant4
FastCaloGAN V2
E_{kin} < 8 GeV && |η| < 2.4, Except [0.9<|η|<1.1, 1.35<|η|<1.5]
FastCaloSim V2
E_{kin} > 16 GeV && |η| < 2.4, All E_{kin} && [0.9<|η|<1.1, 1.35<|η|<1.5, |η|>2.4]
Geant4
Pions: E_{kin} < 200 MeV
Other hadrons: E_{kin} < 400 MeV
FastCaloSim V2
E_{kin} < 4 GeV && |η| < 1.4, E_{kin} < 1 GeV && |η| < 3.15
FastCaloGAN V2
E_{kin} > 8 GeV && |η| < 1.4, E_{kin} > 2 GeV && 1.4 < |η| < 3.15, All E_{kin} && |η| > 3.15
FastCaloGAN V2
Muon Punchthrough + Geant4
Geant4

Figure 3: Simulation tools for AtlFast3, depending on the detector region, particle type and energy [11].

One benefit of such hybrid approach is shown in Figure 4a; neither FastCaloSim nor FastCaloGAN is able to reproduce the number of large-radius jet constituents, whereas the hybrid approach can. Depending on the physics process, AtlFast3 speeds up the ATLAS detector simulation chain by 3–15 times (Figure 4b), and is used for about 50% of the simulated events.



(a) **(b)**

Figure 4: ATLAS fast calorimeter simulation: a) Number of constituents in large-radius jets in samples simulated with detailed simulation (circles), FastCaloSim & FastCaloGAN (triangles) and their hybrid (squares) [12]. b) Simulation time per event, showing speed-up of fast (AF3) over detailed (full) simulation [13].

3. Ultra-Fast Simulation

Next, I discuss ultra-fast simulation which replaces multiple simulation and reconstruction steps with fast approximations, as sketched in Figure 5. This targets $O(100 - 1000+)$ speed-up of the full production chains, and emulates higher level algorithms, such as track reconstruction, in addition to the detector simulation. While delivering very high speed-ups, such algorithms need to capture very large sets of high level observables using a limited number of input features, and need to be retrained in case reconstruction changes. I discuss the approaches by the CMS and LHCb.

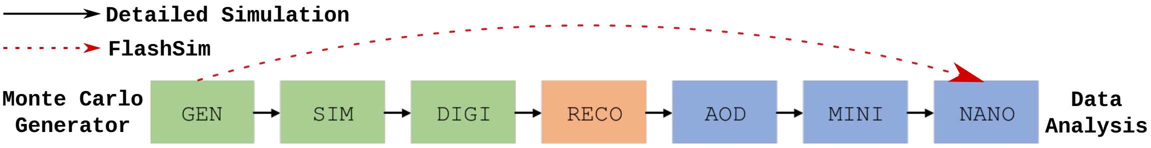


Figure 5: Comparison of production flows of FlashSim and the detailed simulation. Adapted from [14].

3.1 CMS FlashSim

CMS' ultra-fast simulation, FlashSim [14], takes event generator events as input, and outputs events with emulated simulation and reconstruction at a NANOAOD [15] level, used directly by the analyses (Figure 5). The FlashSim is analysis-agnostic, and employs a **normalising flow (NF)** model, in which complex distributions (output) are obtained from simple distributions (input) through invertible, smooth function transforms. FlashSim deploys one model per physics object (jet, b-jet, electron...), ran in a chain to capture correlations between the objects. Early results in Fig. 6 show good modelling of these, and FlashSim achieves up to kHz event generation.

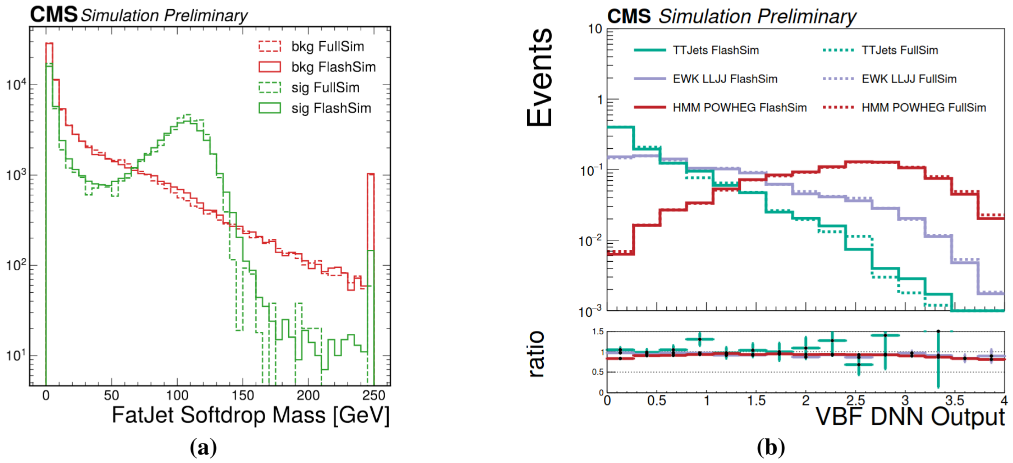


Figure 6: CMS FlashSim performance for a) large radius jets and b) deep neural network discriminant between VBF $H \rightarrow \mu\mu$ signal and backgrounds. FlashSim in full, detailed simulation in dotted lines [14].

3.2 LHCb's Lamarr

The LHCb's ultrafast simulation [16] is split into charged particles (top branch of Figure 8) and neutral particles (bottom branch). A set of **GANs** is used to emulate aspects key to LHCb's analyses: tracking resolution and charged particle identification. The performance of these GANs is shown in Figure 8, for Λ_b^0 production with $\Lambda_b^0 \rightarrow \Lambda_c^+ \mu^- \overline{\nu}_\mu$, and $\Lambda_c^+ \rightarrow pK^-\pi^+$. The good modelling on the reconstructed mass of the Λ_c^+ (a) requires good modelling of the tracking resolution for the p, K^-, π^+ decay products. The proton reconstruction efficiency as a function of p_T (b) requires good particle identification. Lamarr is fully integrated into the LHCb's production workflow and achieves $O(100)$ speed-up over the detailed simulation.

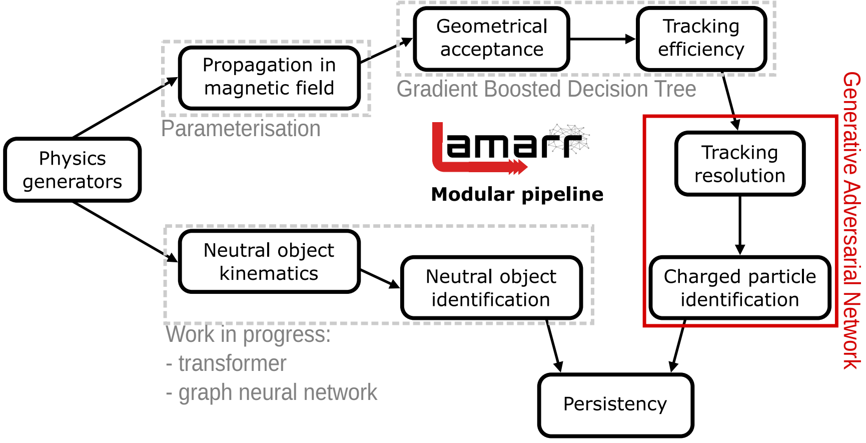


Figure 7: Components of LHCb's ultra-fast simulation. Adapted from [16].

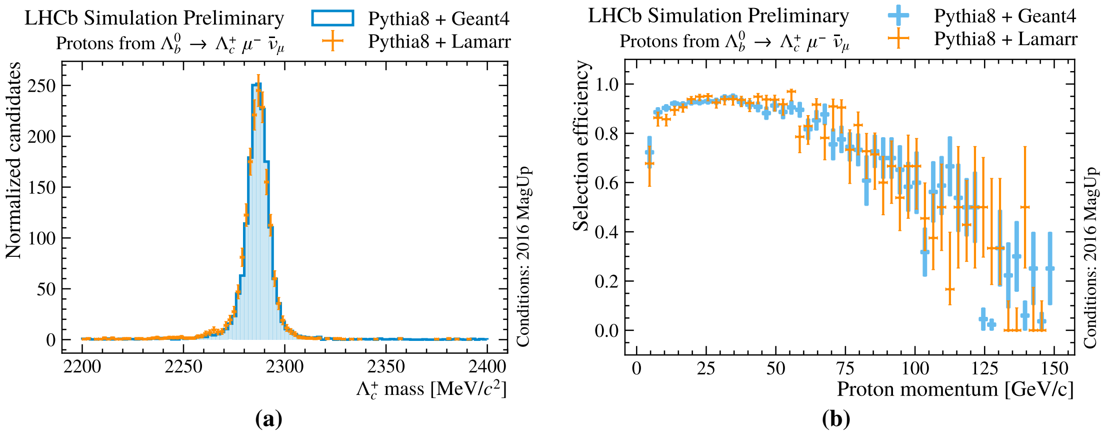


Figure 8: Performance of LHCb's ultra-fast simulation in $\Lambda_b^0 \to \Lambda_c^+ \mu^- \nu_\mu$, $\Lambda_c^+ \to pK^-\pi^+$ decays. a) Reconstructed mass of the Λ_c^+, b) proton selection efficiency [16].

4. Summary and Outlook

To address computing resource constraints, the LHC experiments harness generative fast simulation approaches, including: ALICE's ZDC simulation (CorrVAE), ATLAS's calorimeter simulation (GANs), and ultra-fast approaches: CMS's FlashSim (NFs) and LHCb's Lamarr (GANs). These achieve $\mathcal{O}(10-1000)$ times speed-ups over the traditional production, and excellent physics performance establishes the generative fast simulation as a viable alternatives to the detailed simulation.

An open question is: how to balance the required simulation speed-ups with the demands of the LHC physics program, including ‰-level Higgs boson measurements and developments of next-generation object reconstruction? Steps in this direction include: (1) developing fast simulation of more detector components, such as inner trackers; (2) advancements in generative models, such as diffusion models (DALL-E 2 [17], Imagen [18], StableDiffusion [19]) preferred in industry over CorrVAEs, NFs and GANs; (3) Experiment-independent fast simulation tools, where LHC experiments have made large progress, including ATLAS's tool `pygeosimplify` [20] for building simplified geometry and LHCb's VAE calorimeter simulation based on the CaloChallenge [21].

References

[1] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, JINST **3** S08002 (2008).

[2] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3** S08003 (2008).

[3] CMS Collaboration, *The CMS experiment at the CERN LHC*, JINST **3** S08004 (2008).

[4] LHCb Collaboration, *The LHCb Detector at the LHC*, JINST **3** S08005 (2008).

[5] S. Agostinelli et al, *Geant4 - A Simulation Toolkit*, NIM A **506** (2003).

[6] LHCb Collaboration, *LHCb Computing Resource usage in 2021*, LHCb-PUB-2022-011.

[7] S. Ovyn et al, *DELPHES, framework for fast simulation of a generic collider experiment*, arXiv:0903.2225.

[8] J. Dubinski et al, *Machine Learning methods for simulating particle response in the Zero Degree Calorimeter at the ALICE experiment, CERN*, arXiv:2306.13606.

[9] J. Dubinski et al, *DL-simulation with control over generated data properties*, arXiv:2405.14049.

[10] ATLAS Collaboration, *AtlFast3: the next generation of fast simulation in ATLAS*, Comput Softw Big Sci **6** 7 (2022), arXiv:2109.02551.

[11] ATLAS Collaboration, *AtlFast3 Plots for ACAT* 2024, PLOT-SIMU-2024-03.

[12] ATLAS Collaboration, *Performances of AtlFast3 for Run 3*, PLOT-SIMU-2023-004.

[13] ATLAS Collaboration, *CPU performance of AtlFast3 in Run 3*, PLOT-SIMU-2023-005.

[14] CMS Collaboration, *FlashSim prototype: end-to-end fast simulation using Normalizing Flow*, EPJ Web of Conf. **295** 09020 (2024) and CERN-CMS-NOTE-2023-003.

[15] CMS Collaboration, *A further reduction in CMS event data for analysis: NANOAOD format*, EPJ Web Conf. **214** 06021 (2019).

[16] LHCb Collaboration, *The LHCb ultra-fast simulation option, Lamarr*, EPJ Web of Conf. **295** 03040 (2024), arXiv:2309.13213 and LHCB-FIGURE-2022-014.

[17] A. Ramesh et al, *Hierarchical Text-Conditional Image Generation with CLIP Latents*, arxiv:2204.06125.

[18] C. Saharia et al, *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*, arXiv:2205.11487.

[19] R. Rombach et al, *High-Resolution Image Synthesis with Latent Diffusion Models*, arXiv:2112.10752.

[20] https://github.com/jbeirer/pygeosimplify

[21] M. Mazurek, *Performance of the Gaussino CaloChallenge-compatible infrastucture for ML-based fast simulation in the LHCb Experiment*, presented at ACAT2024.