Towards an experiment-independent toolkit for fast calorimeter simulation

From ATLAS to future HEP detectors

Joshua Falco Beirer[1,*], on behalf of the ATLAS Collaboration

[1]CERN

Abstract. The production of a sufficiently large number of simulated Monte Carlo events is anticipated to be one of the most significant bottlenecks for many future high-energy physics (HEP) experiments. The simulation of the calorimeter response, in particular, represents a major computational challenge. While substantial efforts have been made by the HEP community to develop machine-learning based fast simulation models, integrating these into realistic experimental setups remains a significant hurdle.
Building on the fast simulation tools developed by the ATLAS Collaboration at the LHC, this paper presents recent efforts to create a fully experiment-independent library for fast calorimeter simulation. The library aims to provide a universal interface for both the lateral and longitudinal parameterization of calorimeter showers, as well as for machine-learning based approaches to shower generation.

1 Introduction

The production of simulated Monte Carlo (MC) events is a critical component in modern high-energy physics (HEP) experiments. As experiments collect increasingly large datasets, the computational resources required to simulate MC events are becoming a major bottleneck.

The simulation of the calorimeter response, traditionally performed with Geant4 [1], is particularly demanding as incident particles generate showers containing thousands of secondaries that need to be individually tracked throughout the detector to obtain an accurate simulation of the detector response.

To speed-up the simulation of MC events, HEP experiments have been working towards building *fast simulation* models that typically parametrize the detector response depending on the energy and position of the incident showering particles. Most notably, the ATLAS experiment [2] has been deploying its state-of-the-art fast simulation tool AtlFast3 [3, 4] since the end of Run 2 of the Large Hadron Collider (LHC). To simulate energy deposits, AtlFast3 uses a combination of histogram-based sampling methods and Generative Adversarial Networks [5], tuned to maximize the simulation accuracy across a wide range of physics process.

*e-mail: jbeirer@cern.ch

Driven by the recent rapid developments in fundamental research on generative artificial intelligence (GenAI), the HEP community has shown great interest in exploring various architectures deemed promising for fast calorimeter simulation. Today, the results of the community-driven *CaloChallenge* [6] provide the most complete and comprehensive survey of cutting-edge generative architectures for fast calorimeter simulation. Flow-based generative models [7] and diffusion models [8] have been shown to be particularly promising, albeit the latter can be too slow for practical applications. However, recent advances, such as knowledge distillation and step-reduction techniques, offer promising solutions to improve their efficiency while maintaining generative quality [9].

Despite the impressive progress made in developing and benchmarking cutting-edge GenAI models for fast calorimeter simulation, their implementation into realistic production environments for HEP experiments remains a significant hurdle. In practice, the irregular and complex detector geometries make it difficult to condition a single model to accurately simulate all regions. As a result, multiple models often need to be trained and deployed to account for the variations in detector geometry. Moreover, models are typically trained to reproduce energy deposits in artificially constructed spatial regions known as *voxels*, which do not correspond to the cell sizes of the calorimeter system. During simulation, the predicted voxel energies must be correctly assigned to individual cells by uniformly sampling hits from the voxel surfaces and assigning the generated hits to cells in the detector geometry. Lastly, an accurate simulation also requires accounting for any effects that might arise from the presence of magnetic fields.

Bridging the gap between cutting-edge GenAI architectures and their implementation in realistic HEP environments is crucial to rapidly profit from advancements in the field. The fast simulation tools developed by the ATLAS Collaboration already address most of the technical challenges that arise in implementing ML models for fast calorimeter simulation in realistic settings.

This paper presents the initial steps taken to detach the ATLAS fast calorimeter simulation from its experiment-specific implementation, laying the groundwork to evolve it into an experiment-agnostic library. By generalizing and broadening its functionality, the resulting toolkit aims to support the wider HEP community, particularly the next generation of detectors envisioned for facilities like the Future Circular Collider [10].

2 The ATLAS Fast Simulation Infrastructure

To meet the stringent requirements for physics accuracy in the production of fast simulation MC samples, the ATLAS experiment deploys a suite of advanced simulation tools collectively referred to as AtlFast3. AtlFast3 is designed to significantly reduce the CPU time required per event while maintaining a high level of agreement with Geant4, ensuring minimal degradation in physics accuracy.

While the Inner Detector simulation continues to rely on Geant4, the calorimeter shower simulation is predominantly replaced by two dedicated tools: FastCaloGANV2, which employs generative adversarial networks to generate calorimeter showers, and FastCaloSimV2, which uses histogram-based sampling methods to simulate the detector response efficiently and accurately. Both tools are components of what in the following will be referred to as the FastCaloSim library. The choice of which tool to use during simulation depends on several factors, including the particle type, the region where it enters the calorimeter system, and its kinematic properties, with parameters tuned on an ad-hoc basis to maximize simulation accuracy.

To integrate the FastCaloSim library with other simulators, such as Geant4, ATLAS employs its proprietary Integrated Simulation Framework (ISF) [11]. Originally developed over a decade ago to handle complex use cases that ultimately proved unnecessary, ISF provides a flexible framework for combining full and fast simulation tools by implementing advanced particle routing algorithms. These algorithms apply user-defined simulation selection rules that may vary for each simulated hard scattering event. However, as new simulators and their corresponding detector regions are added, the framework's complexity has grown substantially, making it increasingly difficult to maintain and adapt to evolving needs.

3 FastCaloSim as Experiment-Independent Library

Transforming FastCaloSim into an experiment-independent library represents a critical step toward enabling future HEP experiments to simplify the integration of new models for fast calorimeter simulation into realistic detector setups. Although FastCaloSim was not inherently experiment-dependent, several steps needed to be taken to enable its usage beyond ATLAS.

Notably, dependencies on ISF were removed and replaced with detector-agnostic tools and implementations. Furthermore, the library was decoupled from the ATLAS core software and restructured as a standalone entity hosted in its own dedicated GitHub repository [12].

Geant4 already provides a robust infrastructure for combining full and fast simulation tools through the use of its built-in fast simulation model interface [13]. The new implementation of FastCaloSim is specifically designed to function as a Geant4 fast simulation model, completely replacing ISF's functionality as a particle stack dispatcher and eliminating any reliance on proprietary simulation infrastructure.

Figure 1 shows the comparison between the current implementation of FastCaloSim within ISF in ATLAS' Athena core software and its envisioned integration as Geant4 fast simulation model for future HEP experiments. While the legacy implementation is tied to ISF and the ATLAS core software, the new library can be used as a simple hook in a Geant4 fast simulation model. By enabling a direct and streamlined integration within Geant4, the new implementation eliminates the complexities and potential overhead associated with ISF's intricate particle routing algorithms, all while retaining the full required functionality.

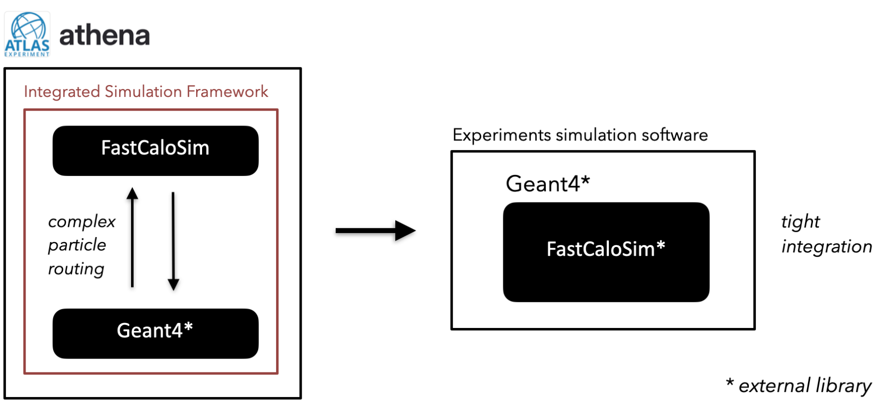


Figure 1: Sketch of the current implementation of the FastCaloSim library within ISF in the ATLAS core software Athena and its future experiment-independent implementation as a Geant4 fast simulation model.

The following outlines some of the recent key developments that are essential toward enabling the use of FastCaloSim for generic detectors.

3.1 Track Transport in Simplified Geometry

To accurately reproduce the lateral shapes of particle showers, FastCaloSim relies on the calculation of the shower centre position within each calorimeter layer during simulation. This involves propagating the incident particle through the calorimeter system, starting at the boundary with the tracking detector, while accounting for magnetic field effects [14]. In fact, the propagation of particles is the key bottleneck in FastCaloSim's simulation.

The current deployment of FastCaloSim within ISF relies on the propagation with customary tracking software in ATLAS' Athena software. For broader applicability beyond ATLAS, alternative experiment-agnostic solutions needed to be found. In principle, Geant4 offers all required functionality within its built-in `G4PropagatorInField` class, which can be attached to the global `G4FieldManager` instance. However, a propagation of particles in the full calorimeter geometry implies `G4Steps` at each cell boundary, which proved computationally prohibitive.

A precise simulation of the detector response with FastCaloSim does not necessitate a highly accurate representation of the track throughout the calorimeter system. Instead, the simulation relies solely on the position of the track at the entrance and exit of each calorimeter layer. By simplifying the calorimeter geometry for particle navigation—avoiding cell-by-cell steps—the propagation speed can be dramatically increased without compromising accuracy.

To simplify the generation of such geometries for future HEP experiments, a detector-agnostic Python package named pyGeoSimplify [15] was developed. Given cell positions and dimensions as input, pyGeoSimplify can automatically generate a clash-free simplified detector in the `GDML` file format [16]. The file can be easily read into Geant4 and a new `G4CaloTransportTool` in FastCaloSim automatically manages the propagation in the simplified geometry, fully independent of the main navigation of the Geant4 simulation. Figure 2 shows a visualization created with pyGeoSimplify of the ATLAS calorimeter cells used as inputs to generate the simplified geometry.

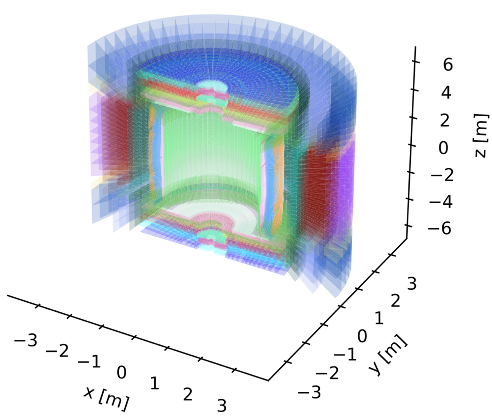
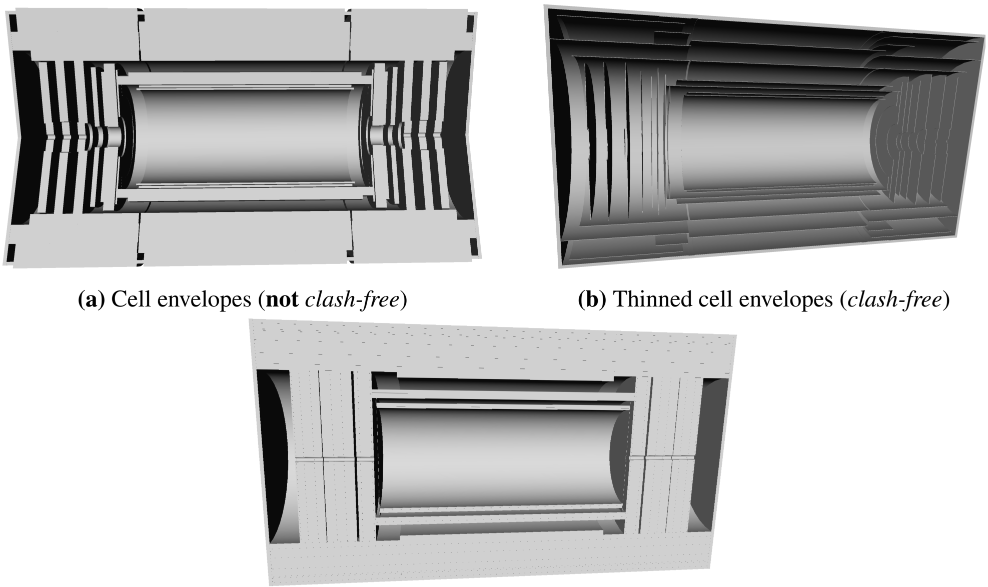


Figure 2: Display of the calorimeter cells of the ATLAS detector produced by pyGeoSimplify. The various colours encode the different layers of the calorimeter.

To build a simplified geometry representation, `pyGeoSimplify` aims at representing each calorimeter layer as a cylindrical structure, approximating the real entry and exit surfaces of the original geometry. The surfaces of these cylinders are defined to match the actual boundaries through which particles enter and exit the calorimeter layers, with the constraint of generating a final geometry free from any internal overlaps, ensuring an unambiguous navigation of particles through the simplified model.

To achieve this clash-free representation, `pyGeoSimplify` uses a multi-step approach, visualized in Figure 3 for the ATLAS calorimeter. Each calorimeter layer is initially mod-



(a) Cell envelopes (**not** *clash-free*)

(b) Thinned cell envelopes (*clash-free*)

(c) Processed thinned cell envelopes (*clash-free*)

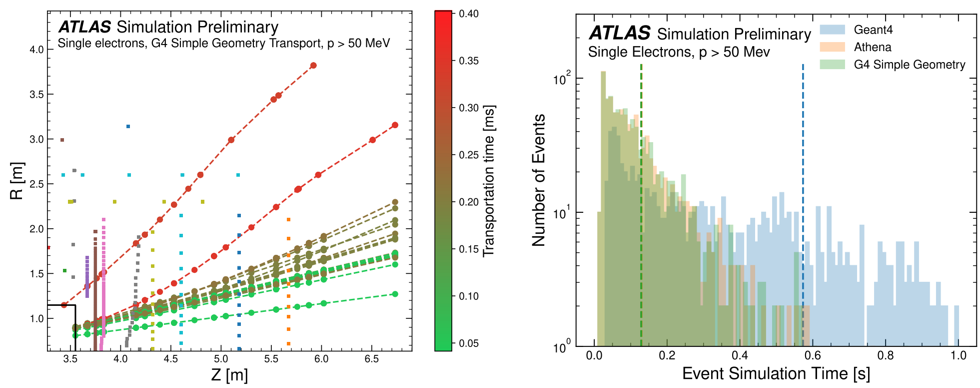
Figure 3: Rendering of the intermediate steps taken by pyGeoSimplify to generate a simplified clash-free detector geometry, in this case for the ATLAS calorimeter system.

elled as a cylindrical envelope determined from the maximum geometric extension of its constituent cells. These cylindrical hulls approximate the original lateral and longitudinal extent of the detector layers. To avoid overlaps among the cylindrical envelopes, a thinning process is applied. For barrel layers, the hull — initially spanning inner and outer radii — is thinned down to a single midpoint radius. For endcap layers, the extent in the longitudinal direction is similarly reduced to a single midpoint surface. After the initial thinning, the geometry is grown back as much as possible toward the original envelope sizes, while avoiding the reintroduction of overlaps. Through this controlled restoration step, the simplified geometry is optimized to closely match the actual detector boundaries while ensuring a clash-free configuration.

For ATLAS, the track transport previously handled by the proprietary tracking software has now been successfully replaced by the Geant4-based propagation in the simplified geometry, with no noticeable changes in physics observables. Figure 4a shows the propagation of a single electron event through the simplified ATLAS calorimeter in the FastCaloSim implementation as a Geant4 fast simulation model. Figure 4b compares the simulation time for single-electron events using the original ATLAS tracking software and the Geant4-based approach in both the full and simplified geometry. While the full geometry significantly increases the CPU time, the simplified geometry restores performance to a level comparable with the original ATLAS tracking tools.

3.2 Hit-to-cell Matching

In order to distribute the simulated energy within each calorimeter layer, FastCaloSim samples a series of hit positions that must be matched to the appropriate cells of the detector



(a) **(b)**

Figure 4: (a) Transport of a single electron event through the simplified calorimeter geometry in the GEANT4 fast simulation model. The dots indicate individual GEANT4 steps taken at each boundary, while the colours of the individual tracks encode the time required to transport the track. (b) Event simulation time of single electron events using the GEANT4 transport in the full calorimeter geometry (blue), with the ATHENA tracking tools (orange), and with the GEANT4 transport in a simplified geometry (green). The vertical lines indicate the mean track transport time in both cases. [17]

geometry. The current approach relies on custom, complex ATLAS-dependent geometry maps to achieve a rapid hit-to-cell assignment.

To enable the usage for other experiments, the matching was generalised by organizing cell information into R-trees [18]. R-trees are spatial indexing data structures that do not necessitate detector-specific information for the lookup and minimize the number of comparisons needed to identify the correct cells. Figure 5 compares the time required to match hits to cells using custom geometry maps and the new R-tree approach. Although the R-tree cell

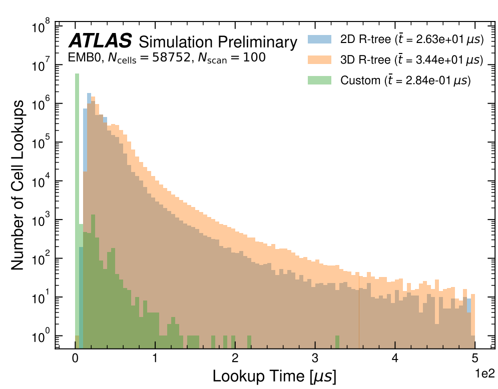


Figure 5: Time required to match positions in the EMB0 layer of the ATLAS calorimeter to their corresponding cells using 2D R-trees (blue), 3D R-trees (orange) and custom, experiment-dependent geometry maps (green). 100 positions are sampled randomly within each cell of the calorimeter layer and matched to its respective cell. [19]

lookup is observed to be up to a factor 100 slower, this level of performance is still expected to be sufficiently fast.

It is important to highlight that the default geometry treatment in FastCaloSim assumes that calorimeter cells can be approximated as rectangular cuboids in some coordinate system. In scenarios where the cell shapes deviate substantially from this approximation, the R-tree-based lookup may not provide the required accuracy. For example, the highly irregular cells of the ATLAS forward calorimeter [20] cannot be reliably handled with a generic approach. To accommodate such cases, FastCaloSim allows developers to implement experiment-specific geometry handlers on a per-layer basis, ensuring flexibility and accuracy for complex detector layouts.

3.3 Open Source Distribution

FastCaloSim is now distributed as an open-source shared library, with its source code hosted on GitHub [12]. The development process embraces modern best practices: a `CMake`-based build system for flexible integration, unit tests with GoogleTest, and static analysis with `clang-tidy` and `cppcheck` for continuous code quality. Consistent coding style is enforced through formatting and linting tools such as `clang-format`, while reproducible development environments are provided via Docker images. The project is continuously tested on multiple platforms, including AlmaLinux, Ubuntu, and LCG [21] releases, ensuring a robust and portable solution for the HEP community. Figure 6 offers a visual example of the simulation output for two photon showers, produced by one of FastCaloSim's unit tests. Such outputs are continuously monitored to detect any unintended changes or regressions in the library's core components.

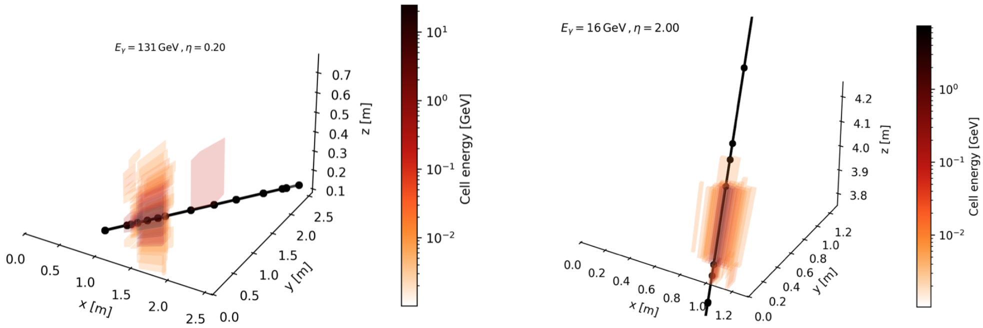


Figure 6: Visualization of simulated photon showers produced by FastCaloSim unit tests.

4 Conclusion

Significant progress has been made in decoupling FastCaloSim from ATLAS-specific infrastructure, enabling its use as an experiment-agnostic toolkit for fast calorimeter simulation. By reimplementing core functionalities with detector-agnostic tools and introducing open-source practices, FastCaloSim now provides a robust foundation for integrating advanced fast simulation models into diverse experimental setups.

Future work will focus on demonstrating its capabilities with the Open Data Detector [22], providing a concrete example of how to integrate FastCaloSim into current and future experiments, including those extending beyond the LHC.

References

[1] S. Agostinelli et al., Geant4 — a simulation toolkit, Nucl. Instrum. Methods Phys. Res. A **506**, 250 (2003). 10.1016/S0168-9002(03)01368-8

[2] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST **3**, S08003 (2008). 10.1088/1748-0221/3/08/S08003

[3] ATLAS Collaboration, AtlFast3: The Next Generation of Fast Simulation in ATLAS, Comput. Softw. Big Sci. **6**, 7 (2022). 10.1007/s41781-021-00079-7

[4] ATLAS Collaboration, Software and computing for Run 3 of the ATLAS experiment at the LHC (2024). 10.48550/arXiv.2404.06335

[5] I. Goodfellow et al., Generative adversarial networks, Commun. ACM **63**, 139–144 (2020). 10.1145/3422622

[6] C. Krause et al., Calochallenge 2022: A community challenge for fast calorimeter simulation (2024). 10.48550/arXiv.2410.21611

[7] I. Kobyzev et al., Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2020). 10.1109/TPAMI.2020.2992934

[8] J. Ho et al., Denoising diffusion probabilistic models (2020), NIPS '20. 10.48550/arXiv.2006.11239

[9] T. Salimans et al., Progressive distillation for fast sampling of diffusion models (2022), ICLR '22. 10.48550/arXiv.2202.00512

[10] FCC Collaboration, FCC Physics Opportunities: FCC Conceptual Design Report Volume 1, Eur. Phys. J. C **79**, 474 (2019). 10.1140/epjc/s10052-019-6904-3

[11] W. Lukas, Fast Simulation for ATLAS: Atlfast-II and ISF, J. Phys. Conf. Ser. **396**, 022031 (2012). 10.1088/1742-6596/396/2/022031

[12] J.F. Beirer et al., fcs-proj/FastCaloSim: v0.2.0-alpha (2024), 10.5281/zenodo.14308465

[13] A. Zaborowska, Geant4 fast and full simulation for Future Circular Collider studies, J. Phys. Conf. Ser. **898**, 042053 (2017). 10.1088/1742-6596/898/4/042053

[14] J.F. Beirer, Novel Approaches to the Fast Simulation of the ATLAS Calorimeter and Performance Studies of Track-Assisted Reclustered Jets for Searches for Resonant $X \to SH \to b\bar{b}WW^*$ Production with the ATLAS Detector, PhD Thesis, Georg-August-Universität Göttingen (2023). 10.53846/goediss-9900

[15] J.F. Beirer, jbeirer/pygeosimplify: v0.0.11 (2024), 10.5281/zenodo.14308340

[16] R. Chytracek et al., Geometry description markup language for physics simulation and analysis applications., IEEE Trans. Nucl. Sci. **53**, 2892 (2006). 10.1109/TNS.2006.881062

[17] ATLAS Collaboration, Public PLOTS: SIM-2024-004, accessed 09-01-2025

[18] A. Guttman, R-trees: a dynamic index structure for spatial searching, SIGMOD Rec. **14**, 47–57 (1984). 10.1145/971697.602266

[19] ATLAS Collaboration, Public PLOTS: SIMU-2024-09, accessed 09-01-2025

[20] ATLAS Collaboration, The ATLAS liquid argon calorimeters: Construction, integration, commissioning and performance from selected beam-test results, Nucl. Instrum. Methods Phys. Res. **598**, 253 (2009). 10.1016/j.nima.2008.08.121

[21] S. Roiser, Servicing HEP experiments with a complete set of ready integreated and configured common software components, J. Phys. Conf. Ser. **219**, 042022 (2010). 10.1088/1742-6596/219/4/042022

[22] C. Allaire et al., Open Data Detector (2022), 10.5281/zenodo.6445359