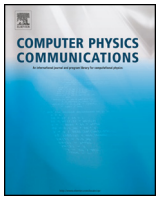
Computer Physics Communications 286 (2023) 108671

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Lorenzetti Showers - A general-purpose framework for supporting signal reconstruction and triggering with calorimeters ☆,☆☆

M.V. Araújo [a], M. Begalli [b], W.S. Freund [a], G.I. Gonçalves [b], M. Khandoga [c], B. Laforge [c], A. Leopold [d], J.L. Marin [e], B.S-M. Peralva [b], J.V.F. Pinto [a], M.S. Santos [e], J.M. Seixas [a], E.F. Simas Filho [e,*], E.E.P. Souza [c]

[a] Signal Processing Laboratory - COPPE/Poli - Federal University of Rio de Janeiro (UFRJ), Brazil
[b] Rio de Janeiro State University, Brazil
[c] LPNHE - CNRS/IN2P3 - Sorbonne Université, Paris, France
[d] Royal Institute of Technology (KTH), Stockholm, Sweden
[e] Digital Systems Laboratory - PPGEE - Federal University of Bahia, Brazil

ARTICLE INFO

Article history:
Received 27 July 2022
Received in revised form 25 December 2022
Accepted 14 January 2023
Available online 25 January 2023

Keywords:
High-energy physics simulation frameworks
Calorimeters
Event reconstruction
Triggering systems
Signal processing

ABSTRACT

Calorimeters play an important role in high-energy physics experiments. Their design includes electronic instrumentation, signal processing chain, computing infrastructure, and also a good understanding of their response to particle showers produced by the interaction of incoming particles. This is usually supported by full simulation frameworks developed for specific experiments so that their access is restricted to the collaboration members only. Such restrictions limit the general-purpose developments that aim to propose innovative approaches to signal processing, which may include machine learning and advanced stochastic signal processing models. This work presents the Lorenzetti Showers, a general-purpose framework that mainly targets supporting novel signal reconstruction and triggering strategies using segmented calorimeter information. This framework fully incorporates developments down to the signal processing chain level (signal shaping, energy estimation, and noise mitigation techniques) to allow advanced signal processing approaches in modern calorimetry and triggering systems. The developed framework is flexible enough to be extended in different directions. For instance, it can become a tool for the phenomenology community to go beyond the usual detector design and physics process generation approaches.

Program summary
Program Title: Lorenzetti Showers
CPC Library link to program files: https://doi.org/10.17632/sy64367452.1
Developer's repository link: https://github.com/lorenzetti-hep/lorenzetti
Licensing provisions: GPLv3
Programming language: Python, C++.
Nature of problem: In experimental high-energy physics, simulation is essential for experiment preparation, design and interpretations of ongoing acquisitions. Especially for calorimeters, an accurate simulation that can describe detector geometry, behavior to different physics processes and signal generation close to the readout electronics and data acquisition levels is required to properly develop signal processing and computational methods. Such detectors may face very challenging demands arising from the new designs, such as pileup mitigation and noise reduction tasks under unprecedented levels. In this sense, simulation requirements continuously increase in complexity and performance, because new physics searches require large datasets and accurate modeling to experimental effects.
Solution method: The Lorenzetti Showers is an integrated software framework that provides complete calorimeter information close enough to the electronic readout chain. Thus, the proposed framework allows users to access cell readout values, configurable sensor pulse-shapes, crosstalk modeling, and different energy estimation methods. It aims at supporting designs that target low or high pileup

☆ The review of this paper was arranged by Prof. Z. Was.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: eduardo.simas@ufba.br (E.F. Simas Filho).

https://doi.org/10.1016/j.cpc.2023.108671

operation conditions in an easy-to-use modular structure. The developed framework is based on Pythia 8 (particle generation) and Geant4 (interactions with the calorimeter technique under analysis). An efficient data recording structure was used to allow full access to the Lorenzetti Showers outputs. In summary, the Lorenzetti Showers tool provides to the scientific community a user-friendly, flexible, user-oriented, and low-level calorimeter simulation framework.

Additional comments including restrictions and unusual features: The framework current version provides the implementation of a generic segmented calorimeter (electromagnetic and hadronic sections), which may be modified by the user, if desired. It allows the generation of particles interactions using Pythia 8 (native) or any generator compatible with the HepMC format (which may be integrated using an external input file) and propagation through a user-configurable calorimeter using Geant4.

1. Introduction

High-energy calorimeters provide accurate and extremely fast response, which allow effective online triggering and offline analysis, essential aspects for particle detector operation [1–3]. In high-energy physics (HEP) experiments, the calorimeter system is usually a multi-layer, highly segmented detector providing energy deposition profiles that allow particle identification, and high-resolution energy and position measurements [3].

Different filtering approaches for particle selection, including electrons and photons [4], jets [5], taus [6] and muons [7] rely on calorimeter information for proper characterization [8,9]. Calorimeters are also fundamental for direct observation of high-energy cosmic rays, for which, high-performance calorimeters with high energy resolution are required [10–14]. Detection and characterization in neutrino physics also depend on calorimeter information [15,16]. Calorimeters are as well used for monitoring and characterization of nuclear power plants [17]. Different technologies allow fast response and different signal reconstruction techniques are used to properly deal with calorimeter energy deposition profiles [3,8]. Traditionally, high-level variables (shower-shapes) are computed from the particle shower development in calorimeters to allow a good discrimination between different types of particles (e-γ, π, τ), and jets [3].

Calorimeters may either be homogeneous or present a sampling structure in which different materials perform passive particle absorption and active signal generation. For homogenous calorimeters, the entire material volume is active [8]. Sampling calorimeters are often employed in high-energy particle colliders, and it is also usual to occur the generation of (scintillation or Cherenkov, maybe even both) light or electric charge [18,19].

The shaping circuit [20] receives the signals produced by the active calorimeter material and generates a typical pulse shape in its output. Among the benefits of such an operation are to increase the signal-to-noise ratio and have a standard pulse (with controlled width, amplitude, and shape) to better cope with the analog-to-digital converter (ADC) specifications (sampling frequency, input signal range, etc.) [21]. After signal acquisition and digitization, at the trigger or analysis level, the digital information from the calorimeter cells may be used to recover the sampled energy amplitude and estimate the particle time-of-flight by using specific energy estimation strategies [22].

Some undesired effects such as signal pileup and crosstalk may introduce energy and time estimation errors. The pileup effect occurs when a pulse appears in the measurement channel before the tail of the preceding pulse has decayed to zero [23]. It affects the pulse amplitude and can produce incorrect energy and time estimation results. The crosstalk is observed whenever the signal generated in a given sensor affects the signal readout of neighboring sensors [24]. Such interference also leads to estimation errors.

Future experiments, such as in the High-Luminosity Large Hadron Collider (HL-LHC) [25] and the Future Circular Collider (FCC) [26], are also planning to use calorimeter information intensively [27,28]. Those experiments will face extremely stringent conditions. For instance, in the FCC-pp, the center-of-mass energy is expected to be as high as 100 TeV, with a peak luminosity of 5×10^{35} cm^{-2} s^{-1}, generating an enormous number of simultaneous collisions, which is expected to reach an average of 1000 particle collisions per bunch-crossing ($<\mu>=1000$), considerably increasing the pileup effect [26]. For dealing with this challenging scenario, novel information processing strategies are under investigation, such as applying image processing techniques to calorimeter cells (low-level information) as an alternative for shower-shapes analysis [29]. Alternatively, a fast and fully-parallelized density-based clustering algorithm optimized for high occupancy scenarios is proposed in [30].

Advanced calorimeter designs are also underway to better deal with energy and time estimation in very high luminosity scenarios [31–33]. To fully explore the information produced in particle collisions, calorimeters are also being designed for the very-forward region along the beam axis [34,35].

Machine learning and especially deep-learning methods [36] have been applied to solve different problems in high-energy particle physics [37,38], neutrino physics [39] and especially in the LHC experiments, for which machine learning applications have experienced considerable growth in the last years [40,41]. Especially, signal processing and machine learning methods have been applied to calorimeter information for optimal particle identification [42,43]. Additionally, embedded systems using ASICS or Field Programmable Gate Arrays (FPGAs) have been designed for machine learning solutions operating at online trigger level to cope with the stringent time requirements of particle physics experiments, which have only few μs at level-1 trigger stage to decide whether an event is to be kept or not for offline analysis [44,45]. There are also some efforts on developing trigger solutions using the multi-core architecture of graphics processing units (GPUs), which can operate at high data rates and perform a large variety of pattern recognition tasks [46]. In the context of an increasing interest for machine learning solutions in HEP, high-quality, reliable, well-characterized, and application-oriented databases are required for system training, validation, and testing [36,47]. Events simulations play an important role in providing valuable data for such purposes.

In experimental HEP, simulation is essential for preparing the experiment, designing the sub-detectors, interpreting the results of ongoing acquisitions, and also for guiding overall upgrades [48]. An accurate simulation that can describe detector geometry, physical processes and signal generation close to the electronics data acquisition level is also required to properly develop signal processing and computational methods that may face very challenging demands arising from the new designs, such as pileup mitigation and noise reduction tasks under unprecedented levels. The aim is to achieve optimal performance both online and offline [49]. Often, Monte Carlo simulation of HEP experiments involves the use of

Geant4 [50] and Pythia 8 [51] within a given simulation framework, which is typically developed considering particular detector descriptions. This is the case of the state-of-the-art general-purpose experiments in the LHC, e.g. CMS [52] and ATLAS [53].

In calorimetry, fast, but approximate, simulation approaches have been used as an alternative to full detector simulation [54–56]. In Delphes [57], a framework was proposed for fast simulation of a generic collider experiment. It applies a smearing approach and includes all typical subdetectors (tracking, calorimeter, and muon systems). Unfortunately, this approach uses a very simplified model with respect to actual experimental effects, as observed in LHC experiments. This makes such simulation very approximate when it is used to recast LHC results or to derive new observations. This approach usually lacks flexibility for user configuration of low-level calorimeter properties.

Recently, machine learning techniques have also been proposed for fast simulation, aiming at learning from data how to generate calorimeter information [42,55,58,59]. One main disadvantage of such data-driven information generation is that it is only feasible for existing calorimeters with pre-configured specifications (as it learns from experimental data) and is not suitable for supporting the development of novel detector designs, nor data processing techniques.

Furthermore, simulation requirements continuously increase in complexity and performance, because new physics searches require large datasets and precise experimental effects modeling. For instance, the calorimeter upgrades from the main LHC experiments [60–62] and future collider detectors now in preparation [63,34] will experience stringent conditions in their measurement systems, as the increasing collider luminosity will produce extreme signal pileup levels [23]. In such situations, the precise simulation of calorimeter response to different particles is of significant importance, as the novel signal processing and machine learning methods under development for pileup effect mitigation require intense data analysis [64]. Therefore, an accurate and fast pileup simulation is among the main concerns for modern experiments [23,65,66].

Unfortunately, the use of event simulation and reconstruction environments is often restricted to the collaboration that develops a given experiment. This limits broader contributions from experimental HEP, signal processing, and machine learning research communities to solve typical calorimeter challenges. Complex analyses involving calorimetry often require specific processing modules, which may be configured in different ways and plugged to the main simulation chain. The aim is to provide effects such as different noise sources, pileup, crosstalk, miscalibration, electronic pulse deformation, and phase delays. It is also relevant to investigate diverse calorimeter designs by varying layer segmentation, cell granularity, geometry, absorber and active materials, and other parameters [3,8].

Addressing advanced simulation support for general-purpose calorimeter designs, this paper presents the Lorenzetti[1] Showers, an integrated software framework that provides complete calorimeter information, which is close enough to the electronic readout chain. Thus, the proposed framework allows users to access cell readout values, configurable sensor pulse-shapes, crosstalk modeling, and different energy estimation methods. It aims at supporting designs that target low or high pileup operation conditions in an easy-to-use modular structure, which is based on Pythia 8 (particle generation) and Geant4 (interactions with the calorimeter technique under analysis). An efficient data recording structure was used to allow full access to the Lorenzetti Showers outputs. In summary, the Lorenzetti Showers tool provides to the scientific community a user-friendly, flexible, user-oriented, and low-level calorimeter simulation framework. The proposed framework will also develop further in the near future to provide a tool to the phenomenology community to better include experimental effects in LHC results recast when data need to be confronted to new models [67–69].

The paper is organized as follows. In Section 2, the Lorenzetti Showers simulator is fully described, presenting its main features and sub-modules. Instructions for installation and running are provided in an online repository. In Section 3, the features of the proposed simulation environment are highlighted by developing a general-purpose application in modern segmented sampling calorimetry. The results from a case-study application are presented in Section 4, together with computer performance evaluation in terms of typical simulation processing times. Event displays and the analysis of typical calorimeter discriminant variables are used for exploiting the environment flexibility. Then, the conclusions are derived in Section 5. Finally, Appendix A provides information about each of the data formats produced by the Lorenzetti framework and Appendix B presents the installation procedure.

2. Lorenzetti Showers framework

The Lorenzetti Showers simulation framework is implemented using Pythia 8 and Geant4 cores and also special modules to deal with pileup, crosstalk, and energy estimation modeling. The proposed framework may also cope with events generated from user-configurable software, replacing the Pythia 8 generator, which is vital for achieving a high level of user-configurable physics process modeling. The proposed software also includes optimized data formats to deal with the complex processing chain required to produce the final simulation output. Among the main features of the Lorenzetti Showers framework is that it incorporates different pulse shapes for the calorimeter readout chain and the corresponding event reconstruction. Additionally, pileup and crosstalk effects modeling is available. Crosstalk usually appears in adjacent sensors of highly segmented calorimeters due to electrical coupling effects [70]. The details of the Lorenzetti Showers simulation environment are presented in the following subsections.

2.1. General description

The Lorenzetti Showers software was developed based on the state-of-the-art technologies used in the reconstruction and simulation frameworks of modern particle collision experiments, which may comprise operation in high-luminosity conditions (such as ATLAS and CMS, at the Large Hadron Collider). In terms of the software structure, the Lorenzetti Showers framework allows a non-expert user to set up the main features of a calorimeter simulation process. As illustrated in Fig. 1, the physics process (particle type, pileup level) is configurable, and the detector specifications (e.g., geometry, number of layers, used materials, pulse shape, and electronics noise) as well. Thus, the proposed framework comprises:

- An event generation module, which is responsible for all interfaces and tools required for particles generation. The default version of the framework uses Pythia 8 to provide an abstraction to the user. Different particle generation software may be used to replace Pythia 8 within the Lorenzetti Showers, allowing a detailed and configurable physics process description. An interface for accepting external input files in the HepMC format is also available in the framework, in order to provide

[1] In Brazil, Lorenzetti is very popular, as there is a full line of water showers with this name. Actually, it has become a general name for water shower equipment as they are recognized for their ease of installation, comprising different models and specifications. In this sense the authors named the proposed simulation framework as an abstraction for shower, trying to highlight the fact that the presented framework is intended to be widely available and easy to use.

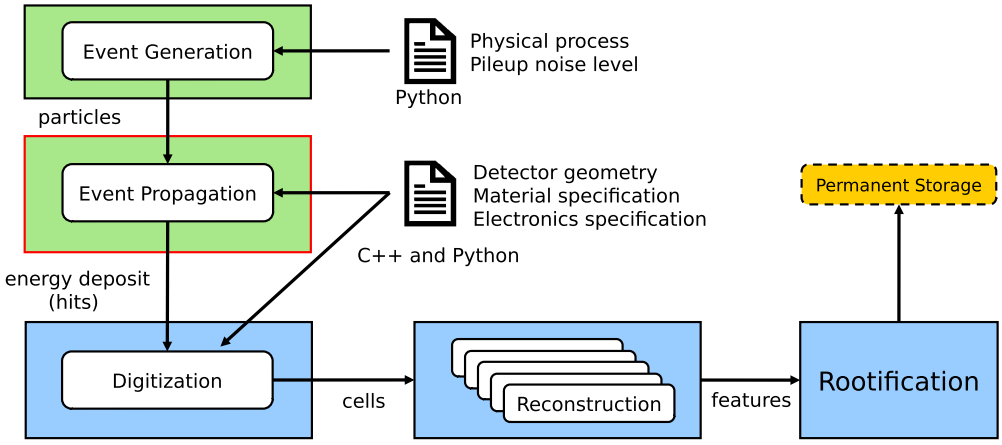


Fig. 1. Diagram of the Lorenzetti Showers framework.

flexibility to users who might have a more convenient event generator;

- An event propagation module based on Geant4, which is widely adopted in areas like high-energy, nuclear and accelerator physics, medical and space science as the primary tool for simulating the particles' interaction with matter using Monte Carlo methods. Geant4 use is not trivial for non-expert physicists; therefore, this module is responsible for abstracting all the construction logic and simulation steps at the user level. In this sense, a user who wants to test a new reconstruction or energy estimation algorithm, for example, does not need to interact directly with Geant4 layers;
- A digitization module was used to emulate the analog-to-digital conversion (ADC) in front-end instrumentation systems. In this stage, the calorimeter cells information generated by Geant4 is quantized to a limited number of values (2^{υ}) according to the defined number (υ) of bits in the ADC.
- A reconstruction module inspired by Gaudi [71] for providing the primary interfaces and services for user-level reconstruction algorithms.

The simulation process generally starts with the user's definition of the target physics processes. The event generation step is pre-configured to allow the simulation of p-p collisions and further production of single particles such as e^{-}, γ and also more complex decays such as jets[2] and $Z \rightarrow e^{-}e^{+}$. The generated events are then propagated thought the detector and finally reconstructed. To simulate the propagation of events through a detector of choice, the user provides to the framework a set of files[3] specifying material, geometry, and dimensions of the calorimeters. Such information will be used by Geant4, to propagate the particles of each event.

The Lorenzetti Showers framework event generation module allows the user to simulate both in-time and out-of-time pileup conditions from multiple particle interactions [23]. The generation of the in-time pileup effect is defined as the realization of numerous ordinary interactions around the main event (seed) in the same bunch crossing. Actually, it is known that modern particle physics experiments collide packets of protons intermittently with a defined time spacing. For instance, in the case of the LHC, the nominal time-spacing between two particle packets crossing (bunch crossing) is 25 ns [72]. Furthermore, the particle interaction with the calorimeter may extend for many bunch crossings due to the time required to produce the full calorimeter response, yielding out-of-time pileup effects when collider luminosity is sufficiently high, increasing calorimeter cell occupancy from particles produced in two subsequent bunch crossings [73].

After the event propagation using Geant4, other experimental effects such as crosstalk and electronic noise are simulated and added to the expected electronic signal in the digitization step. Those signals are then passed to the pulse-shaping module to produce standard readout pulses and sampled at a given frequency to emulate the ADC operation. The resulting samples are used as input to energy and time reconstruction algorithms. In this way, the whole electronic and signal processing chain usually present in real calorimeters is modeled in the proposed simulation software.

The proposed event reconstruction workflow is described in Fig. 2. To enable parallelized processing, several Geant4 instances are interconnected with equivalent instances[4] to begin, execute and end the event processing. In this module, there is a user layer in which it is possible to set up the main features of the simulator. After processing all the events, the information generated by the reconstruction algorithms is stored in persistent format (HIT). To switch among operating modes, four file formats have been defined:

- **EVT (*Event*)**: used to store the information produced by the particle generator. The initial particle states produced by simulating a collision for each bunch-crossing are stored;
- **HIT (*Raw*)**: used to store the information produced by the generator in terms of particle initial states produced during the Geant4 simulation step. This file stores the energy information of all hits[5];
- **ESD (*Event Summary Data*)**: used after the digitization step, usually performed without requiring the execution of the Geant4 module (in standalone mode). This file contains cell information (truth energy, $\eta \times \phi$ position, cell pulse);
- **AOD (*Analysis Object Data*)**: File format containing high-level event information (energy, cluster and others). In general, this format is produced after the cell digitization step.

[2] The simulation of jets comprises the production of QCD jets and prompt photons (gamma + jet and gamma + gamma) after proton-proton collisions.

[3] The set of files with the detector specifications should be written to cope with Geant4; for this, the current version uses C++ files. A simpler way to declare such specifications is under development and will become operational in future versions of the proposed framework.

[4] The parallel computation is used in the event propagation step and is managed by the Geant4 and the *RunReconstruction* instance.

[5] Hits are regions (a volume) in the detector that contain the energy information, deposited by the simulation, concerning each of the simulated bunch crossings.

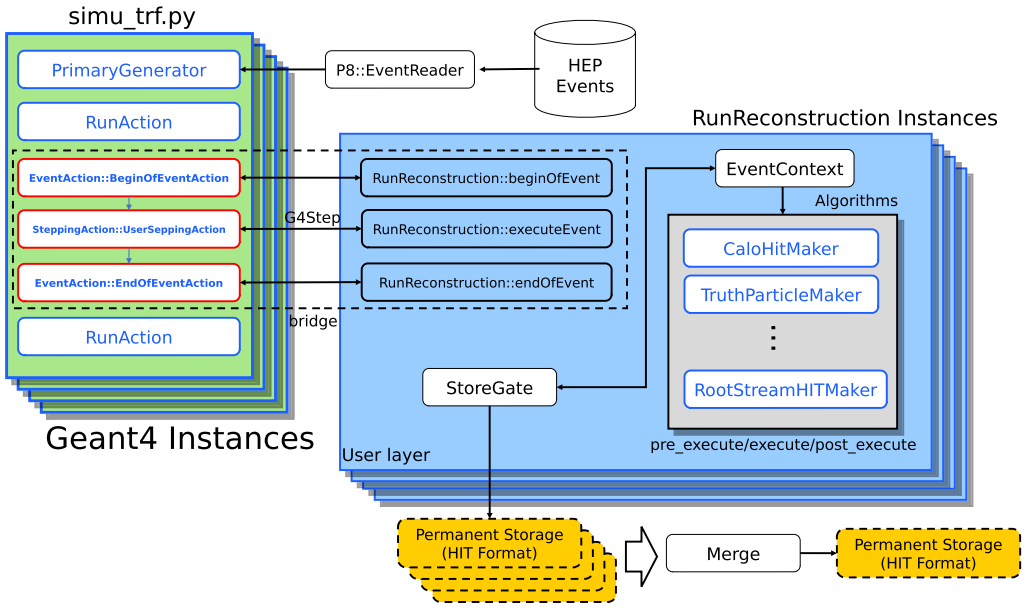


Fig. 2. Diagram of the event reconstruction framework in Geant4's execution mode. The instances in green represent the main steps for running the Geant4 simulation. The instances in blue represent the main components executed by the event manager (*RunReconstruction*) during the execution of the steps highlighted in red by Geant4. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

2.2. Simulation chain

The *Lorenzetti Showers* simulation chain comprises the following main sequence: (1) event generation, (2) propagation, (3) digitization and energy estimation and (4) event reconstruction and analysis. Additionally, pileup merge and crosstalk blocks can be added to the simulation step. These steps are detailed in the following.

2.2.1. Generation

Currently, the *Lorenzetti Showers* framework allows the user to generate certain physics processes from proton-proton (pp) collisions using preconfigured Pythia 8 code, which are called the main events. The following processes are available:

- $pp \rightarrow Z \rightarrow e^{-}e^{+}$: Production of an electron-positron pair from the decay of the Z boson;
- $pp \rightarrow$Jets with a user-configurable minimum energy X GeV (JFX): Production of hadronic particles (jets[6]) with an energy cluster of at least X GeV;
- Single particles: Production of stable particles such as electrons, photons or pions in a position (η, ϕ) defined by the user;
- $pp \rightarrow$ Minimum Bias: Production of multiple physics processes along the entire detector. In general, this production is used to generate the pileup noise events. Such events are processed separately due to their computational cost, and may be added later to the main events.

At the end of the process, each event will have all particles information for a given simulated bunch crossing saved. The information produced in this step is stored in a persistent format of the EVT type.

[6] Pythia 8 configuration for the jets generation uses a set of $2 \rightarrow 2$ processes at the hard scattering level. Pythia 8 can provide more than two jets in the final states because of the parton showering process used inside it to go from the partonic final state to the stable particle final state.

2.2.2. Event propagation

To perform a simulation in the Geant4 environment, it is necessary to use C++ to describe several features, such as the geometry and materials used in the detector, the particles of interest, the physics processes, the initial state of the particles and the main function that manages the execution of the simulation. However, here all these steps are abstracted at the user level and managed by the Lorenzetti Showers framework.

For each simulation step, particle interaction position (η, ϕ) is used to identify the hit corresponding to the simulated event. A numeric identifier is generated and passed to the calorimeter hits map that returns the corresponding object. Finally, the event's total energy deposited in the calorimeter is accumulated in a time-domain position, corresponding to the simulated bunch crossing. At the end of the process, each hit will have the deposited energy information (in MeV) for a given simulated bunch crossing. The information produced in this step is stored in a persistent format of the HIT type.

2.3. Digitization

After the event propagation phase, the digitization step begins with analog electronic pulse and data acquisition (analog-to-digital conversion) emulation. For this, different pulse shapes may be configured by the user to represent the sensor's response and its front-end electronics properly. There are available unipolar and bipolar pulse-shapes in the current Lorenzetti Showers version. This module allows an accurate description of the calorimeter cells and may be used to test different possible sensor and electronics configurations. After running all substeps described below, the entire hits information provided by the HIT file will be converted to cells and stored in a persistent format of the ESD type.

2.3.1. Pileup merge

The merge procedure adds pileup to the main event from a pre-simulated set of randomly drawn minimum-bias events. To perform such operation, the user must inform the average number of pileup events ($< \mu >$) that is to be added. The pileup effect

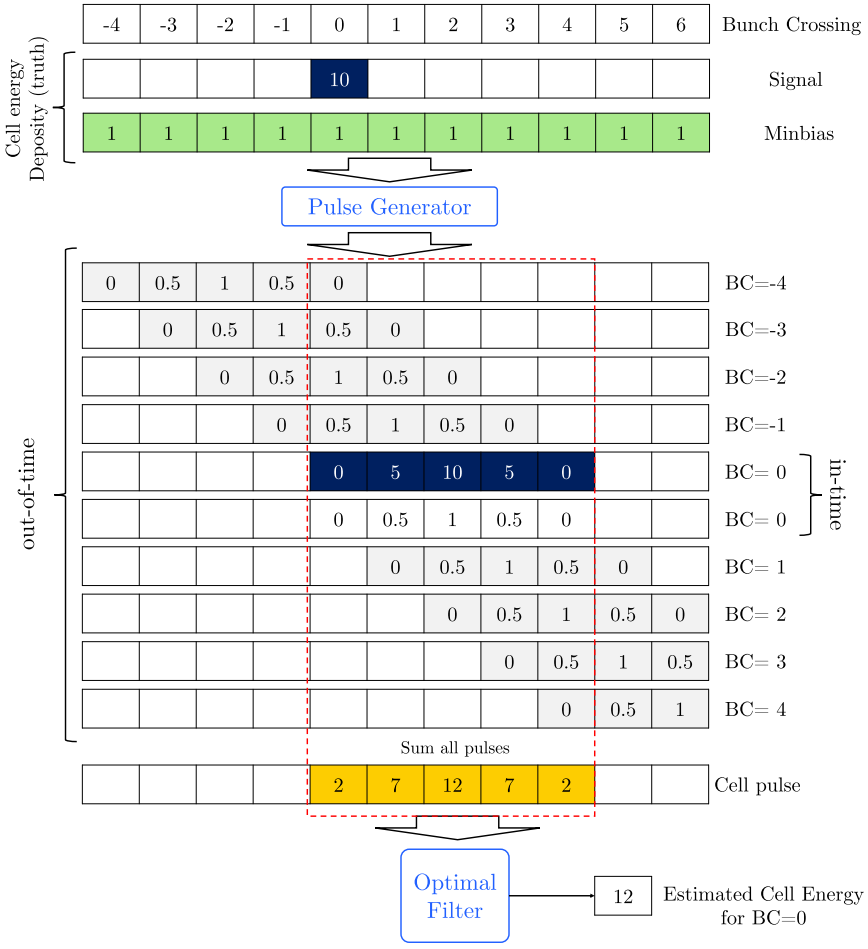


Fig. 3. Illustration of the in-time and out-of-time pileup simulation procedures in a given calorimeter cell. The values and units used are arbitrary.

simulation is illustrated in Fig. 3. It starts with the generation of the electronic pulse for all the calorimeter cells. Taking an electromagnetic calorimeter cell as an example, the energy deposited (hit) in a bunch crossing is used to determine the amplitude (peak) of the generated pulse. The generated samples may span through multiple bunch crossings. For example, the pulse generated by the readout electronics chain spans five bunch crossings (5 samples). As the collisions happen at each bunch crossing (25 ns), there is an overlapping of the electronic signals generated along each bunch crossing by different interactions (signal and noise), characterized as an out-of-time pileup effect. The main event (signal) is allocated at bunch crossing zero (reference).

2.3.2. Crosstalk simulation module

The crosstalk represents interference in a measurement channel coming from other channels in its proximity [24]. In this case, a signal from another channel is replicated and distorts the target information. This phenomenon may appear from three possible couplings: resistive, capacitive, and inductive. The influence of crosstalk is prone to distort the calorimeter shower-shapes, generating possible errors in particle characterization [24]. Considering this, to proper describe the possible effects of such distortions, which are inherent to the calorimeter channels, an accurate simulation of crosstalk is required.

For the Lorenzetti Showers, the crosstalk module allows producing realistic simulations and, in its standard operation, considers that the main contributions are from inductive and capacitive sources. Considering a calorimeter cell, the capacitive crosstalk signal in Laplace transform [74] domain ($XT(s)$) may be expressed in terms of the cell voltage, V(s), using [70]:

$$XT(s) = I_d \frac{sR_{in}C_x}{(1 + sR_{in}C_t)} V(s), \tag{1}$$

where I_d is the current in a cell of interest, R_{in} is the cell input resistance, C_x the coupling capacitance and C_t the total (cell + coupling) capacitance. Similar modeling was used for the inductive crosstalk. The crosstalk module also introduces a stochastic time shift in the pulse-shapes. During bunch crossings at state-of-the-art particle colliders (such as the LHC), interactions can occur on the region with a typical RMS length of a few centimeters around the nominal interaction point. This effect provides vertices that are not at the $(0, 0, 0)$ position. The z displacement provides some time shift for the particles coming from this vertex on a single event basis. A second effect also contributes to this time shift related to the fact that the t=0 reference at the colliders corresponds to the exact time when the maximum density points of the two packets overlay, but early collisions can occur as soon as the two packets start to overlay, while late collisions can still happen when the last proton of each bunch meet each other. In the current implementation, the time shifts were randomly drawn from a zero-mean Gaussian distribution with $\sigma = 0.1$ ns. Beyond such modeling, it is possible to use different crosstalk descriptions by providing additional configuration files.

For the simulations produced in this work, the crosstalk module complete features are still under integration to become native in the Lorenzetti framework, considering this, the presented results

were achieved by accessing the simulated detector hits and applying externally the crosstalk modeling function.

2.3.3. Energy estimation

The energy estimation module allows different methods to estimate the particle energy from the simulated digitized pulses.

A classic approach for energy estimation in calorimeters uses the optimal filtering (OF) method [75–77]. The OF approach models the N time-samples $x[k]$ from a given calorimeter cell as:

$$x[k] = Ag[k-\tau] + n[k] + ped, \quad k = 0, \ldots, N-1, \tag{2}$$

where A is the pulse amplitude, $g[k]$ contains the normalized signal pulse shape information at the time step k, $n[k]$ is the noise, τ is the phase deviation and ped corresponds to the pulse baseline added before the ADC conversion (pedestal). The signal amplitude is estimated through a weighted sum given by:

$$\hat{A}_{OF} = \sum_{k=0}^{N-1} x[k]w[k], \tag{3}$$

where $w[k]$ are the OF weights, which may be computed as described in [78]. The pileup effect is not explicitly modeled in Equation (2) and is expected to be accounted for in the noise term. Such a simplification usually works fine for low pileup scenarios but may be a limitation for high pileup operation.

Additionally, alternative energy estimation methods such as the Matched Filter (MF) [79] and the Constrained Optimal Filter (COF) [80] have been recently proposed and are both available in the Lorenzetti Showers framework. The MF algorithm addresses the problem by developing a hypothesis testing for detecting the signal of interest as a random process [81], envisaging the maximization of the signal-to-noise ratio (SNR) [79]. Such a random process is firstly whitened and, in the sequence, becomes described by the Karhunen-Loeve (KL) series [82], to assess its orthogonal components. Hence, a simple filter is designed for matching each series component. The decision is obtained from random process projections, weighted according to their signal energy description (optimizing the SNR). Envisaging online operation, for which the computational effort is an important issue, a deterministic approximation for the MF design may be developed [83], leading to a procedure similar to Equation (3). The MF output can be calibrated to recover the signal amplitude regardless of the version (deterministic approximation or fully stochastic).

Unlike the OF method, the signal pileup effect is not considered as noise in the COF approach. Instead, it estimates the amplitude of the signals that are present within a given readout time-frame window through a linear signal deconvolution procedure [80]. As a result, the noise accounts only for the electronic noise, commonly modeled as a zero-mean Gaussian process, and the filter design becomes independent of the signal pileup condition.

2.4. Event reconstruction

Lorenzetti Showers software has a standalone mode responsible for accessing the signals produced by the calorimeter cells and reconstructing the high-level information. At this stage, calorimetry-standard shower (shape) variables [3] are computed, including ring sums, which have been introduced as a topological mapping that formats a shower development in terms of concentric rings built around the shower's barycenter [84]. The cells belonging to a given ring have their corresponding energy added to obtain the ring sum values. Such ring sums are employed by the NeuralRinger algorithm [43], which is used for electron triggering in the High-level Trigger of the ATLAS experiment through an ensemble of neural networks fed from ring sums [85].

To build the cluster of calorimeter cells to be analyzed, the position of interaction provided by the simulation determines the center of a searching window, which is set to $\Delta\eta \times \Delta\phi = 0.4 \times 0.4$. Sequentially, the position of the hottest cell in the second electromagnetic calorimeter (ECAL) layer is used as the new search window center in all calorimeter layers. All cells within this window are used to compute the cluster information.

The computed variables were inspired in the ones used in ATLAS experiment [86] and are briefly described in the following[7]:

- E_{ratio} - considering the highest and the second highest energy deposits in the first electromagnetic layer ($E_{1,1\text{st}}$ and $E_{1,2\text{nd}}$, respectively), the energy ratio is computed using Equation (4):

$$E_{\text{ratio}} = \frac{E_{1,1\text{st}} - E_{1,2\text{nd}}}{E_{1,1\text{st}} + E_{1,2\text{nd}}}; \tag{4}$$

- R_η and R_ϕ - the shower extension in η and ϕ are defined in Eqs. (5) and (6), respectively:

$$R_\eta = \frac{E_{3\times7}}{E_{7\times7}}; \tag{5}$$

$$R_\phi = \frac{E_{3\times3}}{E_{3\times7}}; \tag{6}$$

  where $E_{m\times n}$ is the energy deposited in a region of $m \times n$ cells in $\eta \times \phi$ around the hottest cell of the second electromagnetic layer.
- $w_{\eta 2}$ - The lateral width is calculated using a window of 3×5 cells using the energy weighted sum over all cells, which depends on the particle impact point within the cell:

$$w_{\eta 2} = \sqrt{\frac{\sum E_i \times \eta_i^2}{\sum E_i} - \left(\frac{\sum E_i \times \eta_i}{\sum E_i}\right)^2}; \tag{7}$$

- Ring Sums - the assembly process for rings uses the position in η and ϕ of the hottest cell in the second electromagnetic calorimeter layer. The cells adjacent to the first ring (hottest cell) will be added together to form the second ring, and this process will be repeated up to a certain number of rings. For all other layers, the first ring is defined using the same position of the hottest cell in EM2, and, sequentially, the assembly process is repeated.

2.5. Simulation chain example

Fig. 4 represents a summary of $Z \to ee$ event simulation chain with pileup. In (1), the $Z \to ee$ and pileup (minbias) generation are performed separately. In order to save computation time, minimal bias simulation is stored separately and can be merged into the main event. This procedure allows the user to simulate different types of physical processes (main events) and merge them with the simulated pileup files only once. This separate execution also applies in (2) for the simulation steps. In (3), the HITs generated from the main event and the randomly drawn pileup event are merged. In (4), the digitization step is performed. In this step, the energy deposited by the simulation is used to generate the electronic pulses, and perform the crosstalk, and energy estimation. The crosstalk is produced in the ECAL cells, triggering an optional module that may be enabled for specific studies. The high-level event reconstruction from cells information is performed in (5). Finally, in (6), the file produced is submitted to an analysis step.

[7] It is worth noting that the user may specify different variables to be computed in this step, as required for proper characterization of the physics processes of interest.

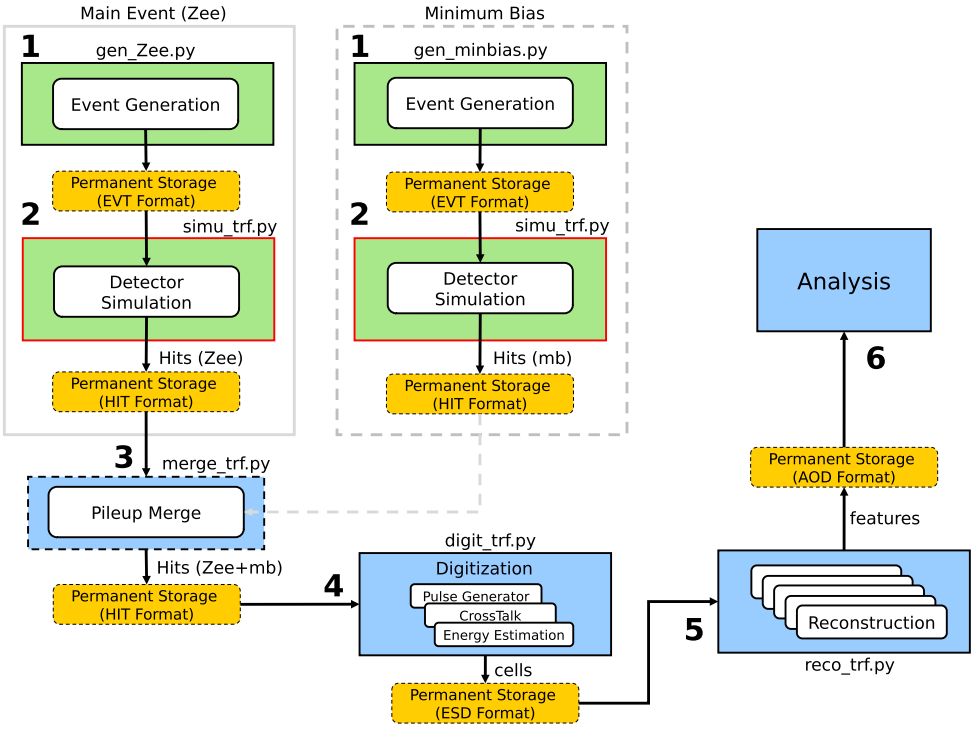


Fig. 4. Summary of the simulation chain for the example of $Z \rightarrow ee$ events with pileup.

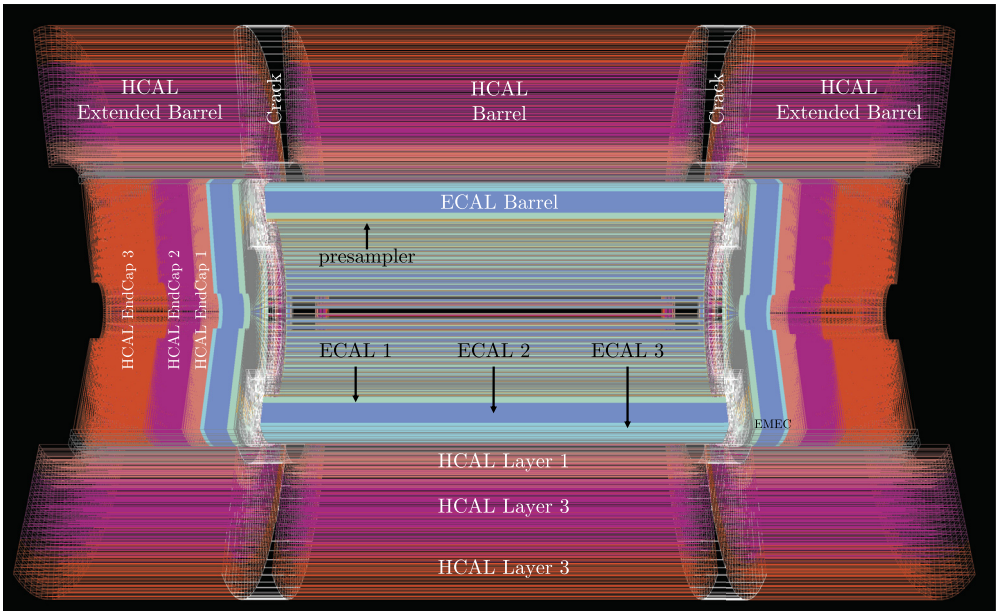


Fig. 5. Cross-sectional view of the general-purpose calorimeter used to illustrate a simulation example with the Lorenzetti Showers framework.

3. General-purpose calorimeter simulation

A general-purpose calorimeter system, presented in Fig. 5, was used to illustrate the functionalities of the Lorenzetti Showers framework. It consists of sampling calorimeters with symmetry and full coverage in ϕ. The Electromagnetic Calorimeter (ECAL), which is the innermost part of the calorimeter, covers a region of $|\eta| < 3.2$. It can be split into barrel, which covers a region of $|\eta| < 1.475$, and end-caps (EMECs), covering $1.375 < |\eta| < 3.2$. The ECAL comprises three superimposed layers.

The Hadronic Calorimeter (HCAL) covers the same pseudo-rapidity region as the ECAL, surrounding it, and is also split into three layers. The HCAL barrel reaches up to $|\eta| < 1.0$, and an extension is added to increase the barrel coverage up to the $0.8 < |\eta| < 1.7$ region. Finally, the HCAL end-caps cover the $1.5 < |\eta| < 3.2$ region. The detailed calorimeter segmentation description is presented in Table 1. Poorly instrumented regions (gaps) are modeled around $|\eta| = 1.5$ to emulate the areas used in real detectors for cable passing.

A 2 Tesla magnetic field is configured to act within the detector volume uniformly axially and parallel to the beam direction (z). If the particle is neutral, or has no charge, its trajectory is a straight line from the point of production, center of the detector,

Table 1
Coverage regions in η, cell granularity and sampling layers used in the simulated calorimeter.

Layer	Sampling	Coverage	Granularity ($\Delta\eta \times \Delta\phi$)
Presampler	Barrel	$0.00 < \|\eta\| < 1.58$	0.025×0.1
	End-Cap	$1.50 < \|\eta\| < 1.80$	0.025×0.1
Electromagnetic Calorimeter			
Layer 1	Barrel	$0.00 < \|\eta\| < 1.55$	0.003×0.1
		$1.37 < \|\eta\| < 1.80$	0.003×0.1
	End-Cap	$1.80 < \|\eta\| < 2.00$	0.025×0.1
		$2.00 < \|\eta\| < 2.37$	0.006×0.1
		$2.37 < \|\eta\| < 3.20$	0.1×0.1
Layer 2	Barrel	$0.00 < \|\eta\| < 1.50$	0.025×0.025
	End-Cap	$1.35 < \|\eta\| < 2.50$	0.025×0.025
		$2.50 < \|\eta\| < 3.20$	0.1×0.1
Layer 3	Barrel	$0.00 < \|\eta\| < 1.58$	0.05×0.1
	End-Cap	$1.35 < \|\eta\| < 2.50$	0.05×0.025
		$2.50 < \|\eta\| < 3.20$	0.1×0.1
Hadronic Calorimeter			
Layer 1	Barrel	$0.00 < \|\eta\| < 1.09$	0.1×0.1
	Extended Barrel	$0.94 < \|\eta\| < 1.77$	0.1×0.1
	End-Cap	$1.50 < \|\eta\| < 2.50$	0.1×0.1
		$2.50 < \|\eta\| < 3.20$	0.2×0.2
Layer 2	Barrel	$0.00 < \|\eta\| < 1.09$	0.1×0.1
	Extended Barrel	$0.85 < \|\eta\| < 1.41$	0.1×0.1
	End-Cap	$1.50 < \|\eta\| < 2.50$	0.1×0.1
		$2.50 < \|\eta\| < 3.20$	0.2×0.2
Layer 3	Barrel	$0.85 < \|\eta\| < 0.72$	0.2×0.1
	Extended Barrel	$0.85 < \|\eta\| < 1.41$	0.2×0.1
	End-Cap	$1.50 < \|\eta\| < 2.50$	0.1×0.1
		$2.50 < \|\eta\| < 3.20$	0.2×0.2

to a calorimeter cell. Otherwise, it follows a helical path until it reaches the calorimeters.

4. Simulation results

In this section, results are shown by the use of the Lorenzetti Showers framework to simulate the propagation of electrons (from $Z \rightarrow ee$ decays) and jets through the general-purpose calorimeter described in Section 3. A total of 10,000 events were produced for each signature (with and without pileup). The events were generated without pileup and with an average pileup of $< \mu >= 60$. The crosstalk effects were also considered.

The event display of a $Z \rightarrow ee$ with pileup is presented in Fig. 6. One may observe hits in specific detector regions (marked in yellow) along with shower propagation. Figs. 7 and 8 show typical layer-level views of the energy deposition profiles for an electron and a jet without pileup, respectively. As expected, after interacting with the sequential calorimeter layers, smaller longitudinal and lateral spreads were observed for the electromagnetic shower, when compared to the hadronic one. It is possible to see that the considered electron presented very low energy in the hadronic layers (lower than 10 MeV, which is far below the simulated electronic noise level, that was set up to ~ 20 MeV for those layers). Oppositely, the hadronic shower reached those layers with more significant energy.

4.1. Pulse-shaper

As shown in Fig. 9, typical calorimeter cell pulse-shapes, bipolar and unipolar, were used for electromagnetic and hadronic layer response validation, respectively. The pulse shapes are generated in calorimeters through the processing of the signals by a pulse-shaping electronic circuit [87]. The pulse shapes are used in the simulation to produce the cells' timing response, which is essential for advanced calorimeter studies concerning energy and arrival time estimation methods. The Lorenzetti Showers framework allows the user to set up different pulse-shape functions and energy estimation algorithms, which can be of interest for the development of future experiments.

4.2. Crosstalk

The crosstalk effects in the pulse-shape of a bipolar pulse-shaped cell are illustrated in Fig. 10. It may be observed in the left plot, the original signal ($g(t)$), the capacitive (XT_C) and inductive (XT_L) crosstalk contributions, the noise term, and the sun (composition) of all previous terms. A zoom is presented in the right plot to highlight the amplitude and time distortion produced by the crosstalk.

Fig. 11 shows the crosstalk effect in terms of the energy deposition along with two rings of neighboring cells around the hottest cell for an electron. It can be seen that a considerable lateral energy spread appears due to crosstalk, together with a distortion of the energy observed in the central (hottest) cell.

4.3. High-level information for shower description

In Fig. 12, some typical calorimeter shower shapes are presented for electrons and jets (no pileup). It is possible to observe that electrons from $Z \rightarrow ee$ decays present R_η and R_ϕ values concentrated around 1 (see Figs. 12a and 12b, respectively). Alternatively, the probability density functions (pdfs) for jets span to values below 0.5, indicating considerable lateral leakage for hadronic showers, as expected.

In a typical calorimeter design, some η regions are poorly instrumented with sensors (cells). The general-purpose calorimeter used here tries to emulate such experimental limitation, which usually degrades particle characterization. It can be seen in Fig. 13 that the E_{ratio} is considerably influenced by the poorly instrumented region around $|\eta| = 1.5$. Such shower variable pdfs for electrons and jets are heavily mixed in this region.

Fig. 14 displays the average ring profiles produced by the Lorenzetti Showers for electrons and jets (no pileup). Such information may generate discriminant profiles for particle identification. The ring profiles highlight the larger lateral and longitudinal spreads generated by hadronic decays.

The effect of pileup merging may be observed in Fig. 15. Comparing such variables when they are computed from a simulation with pileup ($< \mu >= 60$), it is possible to observe, especially for R_η and R_{had} that the electron-jet discrimination power is reduced. In this sense, such distributions become more similar for those two classes of particles.

4.4. Computational performance evaluation

The computational requirements for simulation production were evaluated from producing several events of different physics processes (single electrons, $Z \rightarrow ee$, jets, and minimum bias) and by estimating the average simulation times in a stand-alone computer.[8] The Lorenzetti Showers simulator was executed several times using the setup detailed in Table 2. As illustrated in Fig. 16 (a), it is possible to see that for single electron simulations, the average time per event tends to stabilize for each computational step after the production of 50 events. Similar behavior was observed for the other types of simulated particles. This indicates no cumulative computational overhead effect.

[8] This analysis was performed using the following computational environment: personal computer with Core i5-6200U 2.3 GHz processor, 8 GB RAM, 1 TB HDD, and Ubuntu 20.04.3 LTS 64 bits.

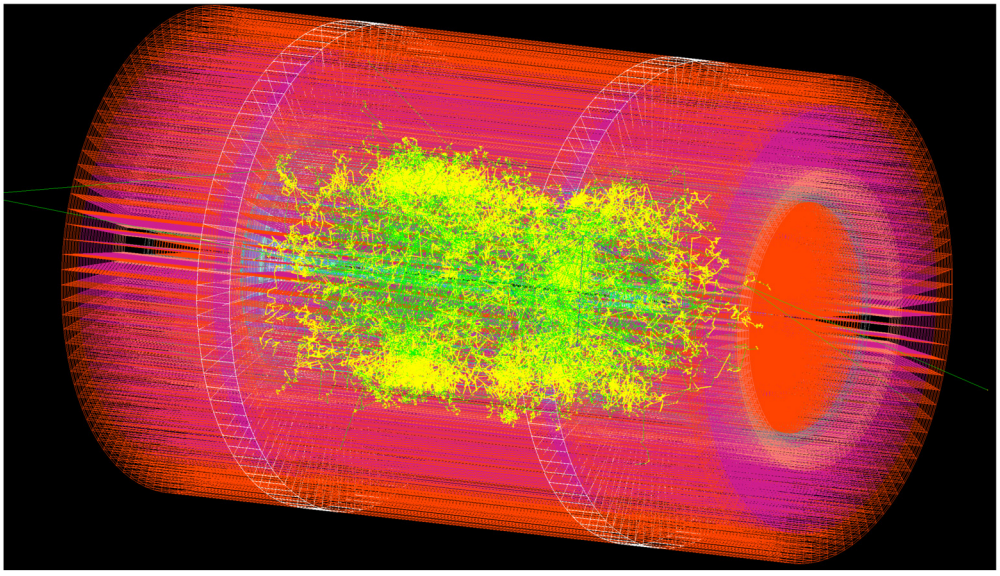


Fig. 6. Front view of the general-purpose calorimeter (G4Hits) after an event propagation.

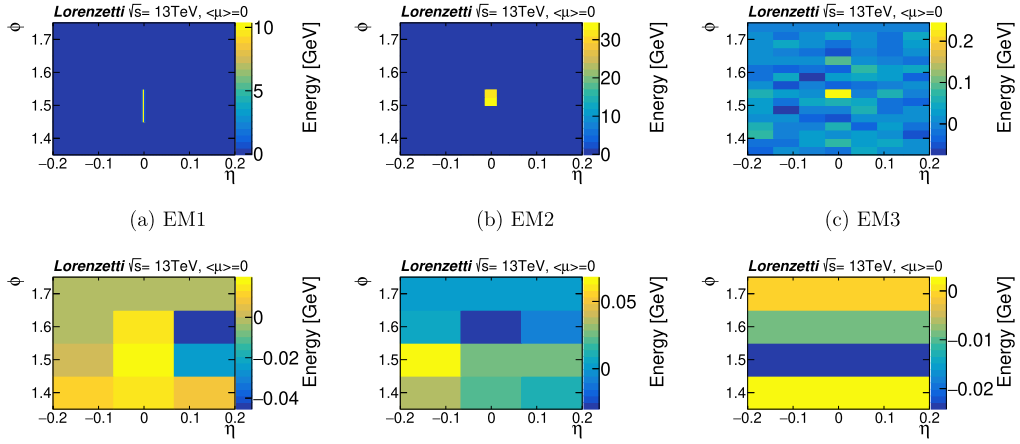


Fig. 7. Typical electron energy deposition profiles of an electron through layers of a general-purpose calorimeter: (a) - (c), electromagnetic and (d) - (f), hadronic.

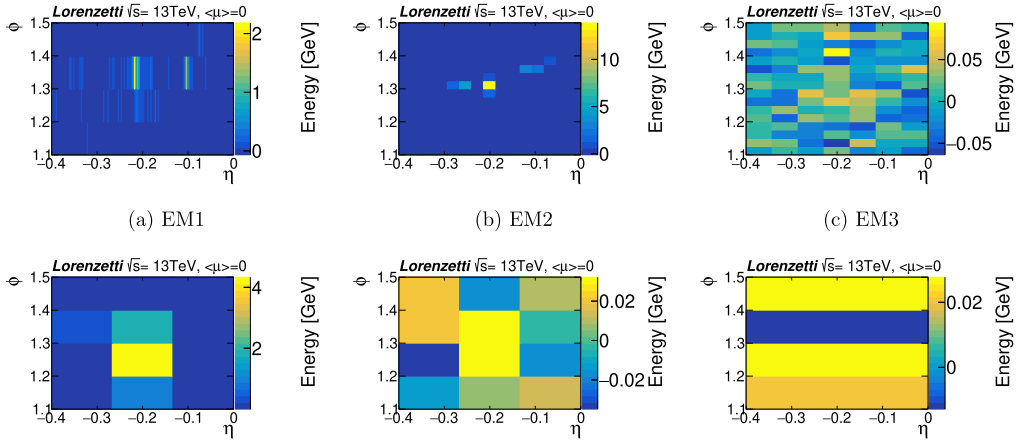


Fig. 8. Typical jet energy deposition profiles through layers of a general-purpose calorimeter: (a) - (c), electromagnetic and (d) - (f), hadronic.

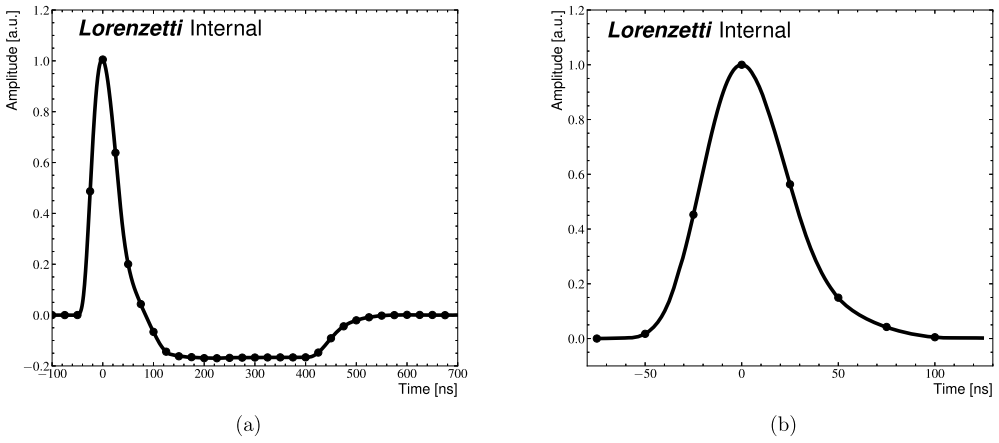


Fig. 9. Pulse shapes for bipolar (a) and unipolar (b) calorimeter responses.

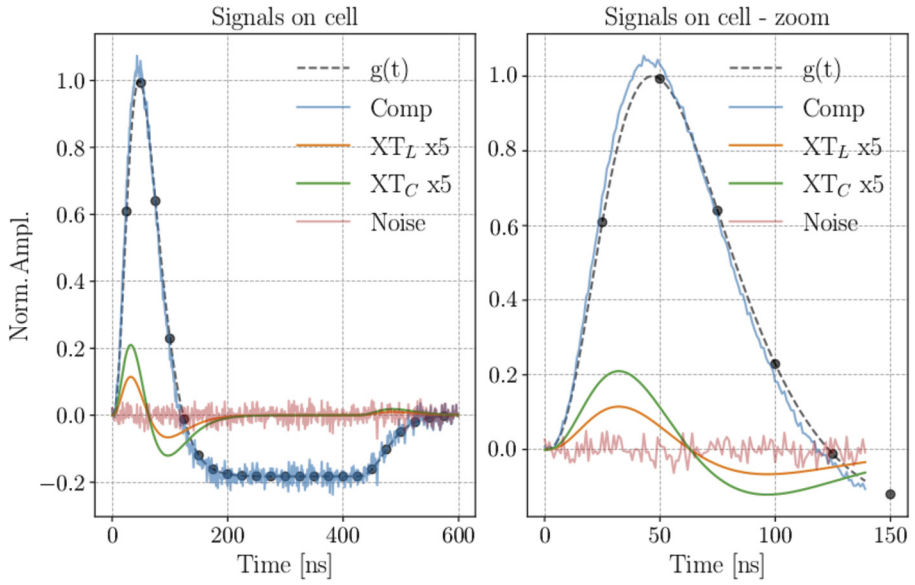


Fig. 10. Simulated signals for a cell of the ECAL. The cell signal is shown by the black dotted line. In green, the XTC signal; in orange, the XTL signal, and, in red, the noise contribution, and the resulting signal (Comp). A detailed zoom (right plot) highlights the amplitude change and peak-time shift.

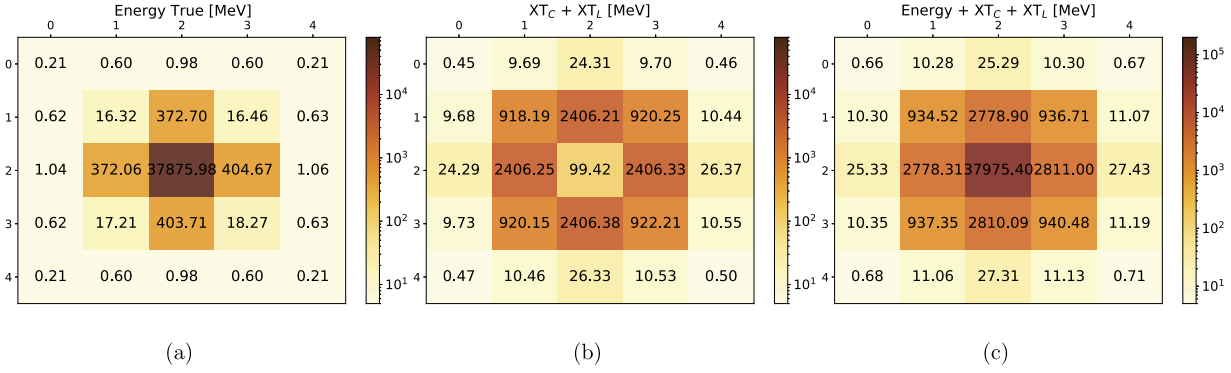


Fig. 11. Cell energy of 5 × 5 clusters: (a) the true energy cell by cell, (b) XT contributions and (c) energy + XT.

From Fig. 16 (b) it is possible to see that the relative amount of time required for each computational step varies for the different main events (electron, $Z \rightarrow ee$, and jets). For example, the simulation step takes a more significant part of the total time for jets generation when compared to electrons. This is probably due to the more complex physics phenomena involved in jets production. The minimum bias (MB) events are simulated separately from the main events, recorded, and used to produce combined (merged) simulations: electron+minimum bias (eMB), $Z \rightarrow ee$+minimum bias (ZeeMB), and jets+minimum bias (JF17MB). After recording the MB events, it is considerably faster to produce the combined events. This feature of the Lorenzetti Showers framework saves computational time for the production of different main event types with pileup, as the MB samples may be reused.

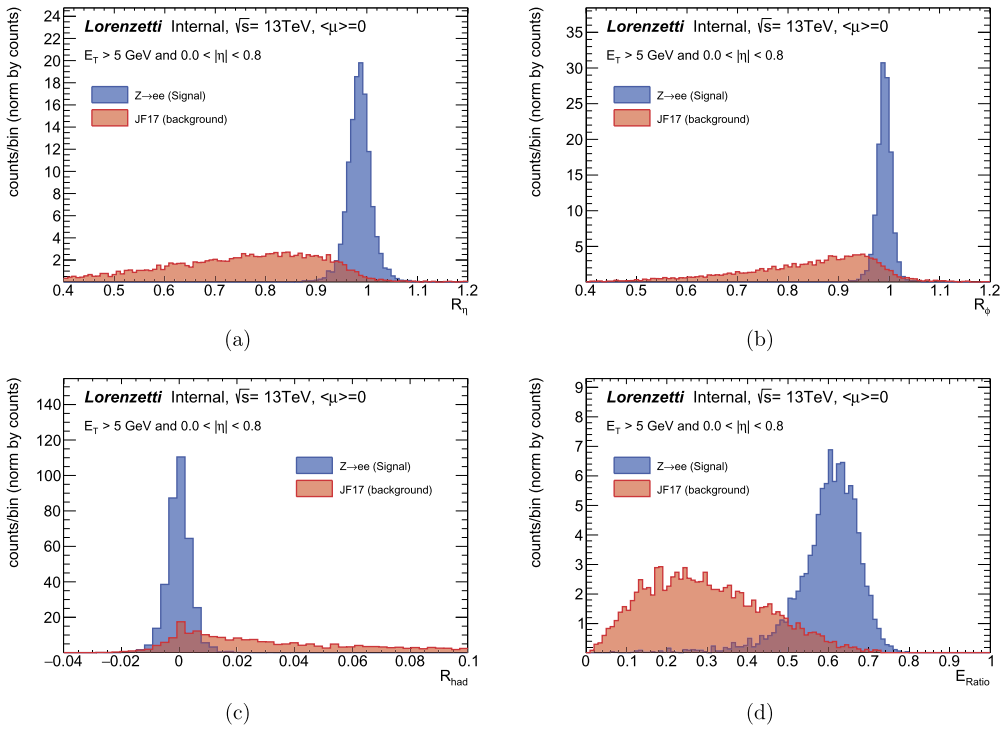


Fig. 12. Shower variables computed for the energy clusters from $Z \rightarrow ee$ (blue) and jets (red) without pileup: (a) R_{η}, (b) R_{ϕ}, (c) $w_{\eta 2}$ and (d) E_{ratio}.

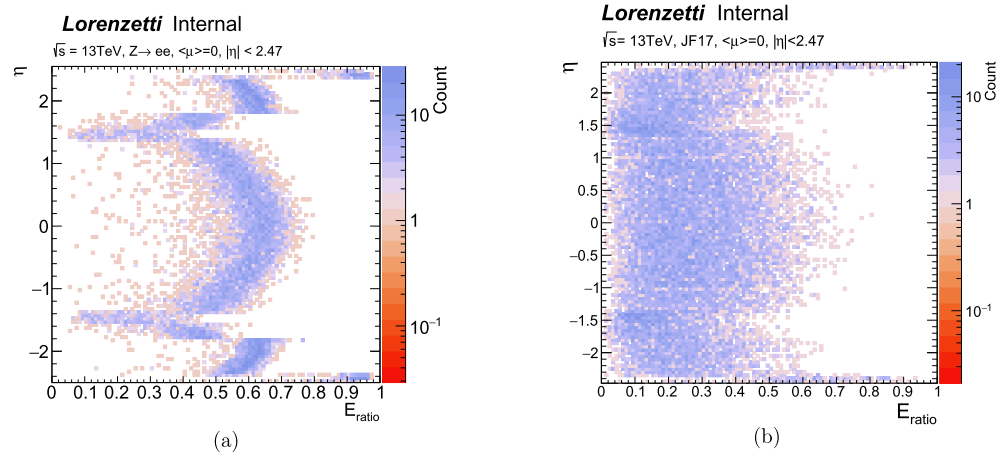


Fig. 13. $E_{ratio} \times \eta$ plots for (a) electrons and (b) jets.

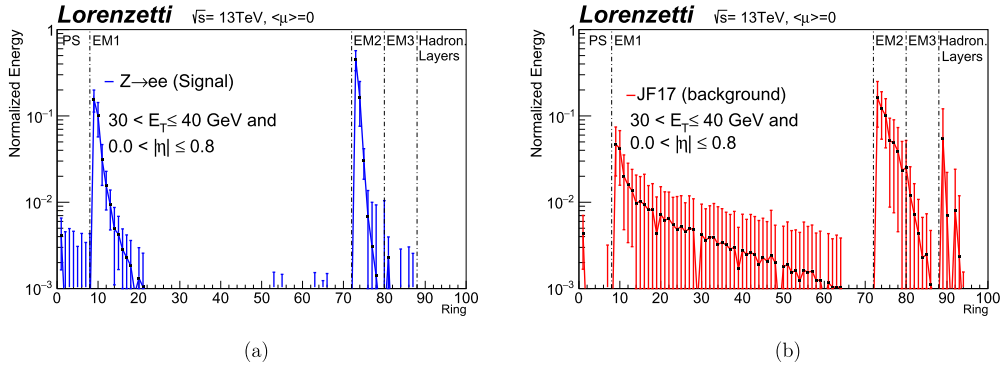


Fig. 14. Discriminant calorimeter variables (rings) generated in the Lorenzetti simulations for typical electrons (a) and jets (b).

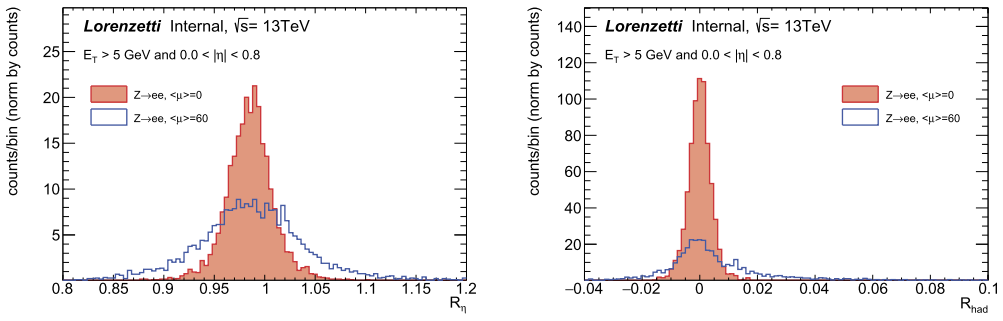


Fig. 15. Shower variables computed for energy clusters from $Z \rightarrow ee$ without pileup (red) and with pileup of $< \mu >= 60$ (blue): (a) R_{η} and (b) R_{Had}.

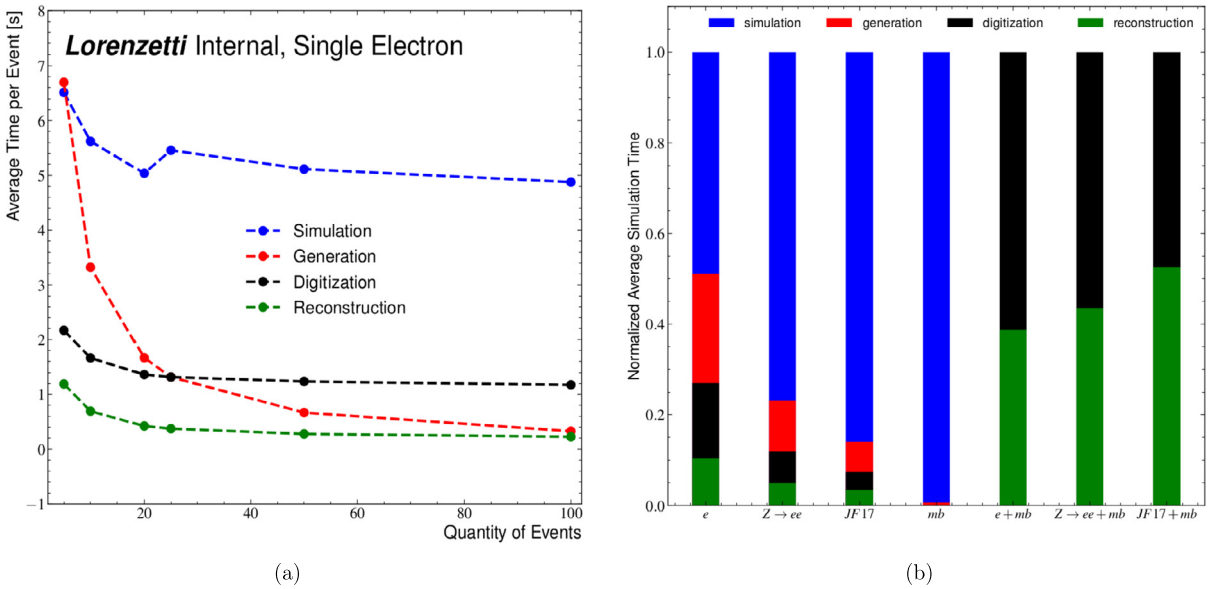


Fig. 16. Simulation time results: (a) average time for electrons along the different computational steps and (b) relative simulation for each computational step considering different types of events.

Table 2
Setup used for the simulation time evaluation.

Simulation Time Evaluation Setup	
Main Event Generation	Electrons, Jets, $Z \rightarrow ee$
Minimum Bias Generation	$< \mu >= 60$
Propagation	General-Purpose Calorimeter (7 layers)
Energy Estimation	Optimal Filter
Reconstruction	Shower Variables + Rings

Table 3 summarizes the average processing times for 10 events for different computational chains. Single electrons require a considerably smaller production time than $Z \rightarrow ee$ and jets. In this sense, the MB production is responsible for the highest processing time, which is more than 150 times larger than for electrons.

5. Conclusions

Modern high-energy experiments rely on calorimeters due to their high energy resolution, fine-grained information, and fast response. To deal with the increasingly stringent experimental conditions arising from upgrade programs of running experiments and future designs, accurate simulation is of major importance. Nowadays, particular emphasis is being placed on incorporating advanced stochastic signal processing and machine learning methods into the online processing chains for calorimeter signals and the corresponding offline analysis.

This paper presents the Lorenzetti Showers framework, an alternative for detailed, accurate, and user-configurable simulation of high-energy calorimeters. The simulation framework already focuses on supporting online signal processing and advanced offline analysis at adjustable luminosity conditions, as there is a trend for high-luminosity colliders in the area. The Lorenzetti Showers framework allows general-purpose calorimeter designs to be described and produces EM and hadronic shower profiles for adjustable pile-up levels, allowing using different pulse-shaping strategies, energy estimation algorithms for handling signal readout in extreme pile-up conditions, and signal crosstalk modeling and mitigating techniques for handling undesired effects on neighboring calorimeter cells. The reconstruction framework particularly supports sophisticated triggering topologies, producing, together with cell information, several particle discriminant calorimeter-based measures, such as shower-shape variables and ring-sums.

The results presented in a case study highlight the flexibility of the Lorenzetti Showers framework. The computational time analysis suggests that the proposed simulator is executable even from stand-alone computers on a feasible time scale. Further improvements are foreseen, which may include the implementation of a tracking system model between the collision point and the

Table 3
Average production times (in seconds) for simulation of 10 events from different physics processes of interest.

Electrons	$Z \rightarrow ee$	Jets	MinBias (MB)	Elect.+MB	$Z \rightarrow ee$+MB	Jets+MB
101.9 ± 5.4	174.2 ± 18.0	260.8 ± 29.8	15311.0 ± 356.0	20.1 ± 0.1	21.3 ± 0.1	23.6 ± 0.1

calorimeter; exploitation of innovative calorimeter structures, especially for the forward region, where new trigger strategies are being planned; accurate particle scattering timing simulation and the generation of different physics processes. It is also planned to provide the implementation of a novel strategy to enhance min-bias statistics, for pileup, by randomly rotating the events before overlaying them with the main physics process.

CRediT authorship contribution statement

J.V.F. Pinto	Software core development, analysis development, supervision, writing - original draft, review and editing
W.S. Freund	Detector development, event generation development, analysis development and supervision
M.V. Araújo	HIT/Cells map module development and note editing
J.L. Marin	Rings reconstruction, analysis development and note editing
G.I. Gonçalves	Energy estimation module development, analysis development and note editing
M.S. Santos	Crosstalk module development, analysis development and note editing
E.E.P. Souza	Rings reconstruction development and note editing
M. Khandoga	Crosstalk module development
B.S-M. Peralva	Optimal Filter development and writing - editing
E.F. Simas Filho	Contributor and writing - review and editing
M. Begalli	Pileup merge simulation design and note editing
A. Leopold	Contributor and note editing
J.M. Seixas	Supervision, analysis and writing - review and editing
B. Laforge	Supervision, analysis and writing - review and editing

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank CNPq (40858720210), FAPESB (96982015), FAPERJ (002362/2020), FAPEMIG (21006820/2020), and RENAFAE (Brazil), and COFECUB (France) for the financial support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001.

Appendix A. Data format description

The data formats generated by Lorenzetti Showers framework are described here. Each simulation step stores (in a specific file format) a different set of variables that are used as inputs to the next step. After the generation step, a EVT file is produced. Table A.4 presents the main attributes stored in such file format. It is possible to observe that the stored variables are related to the simulated physical event. If the user is interested in producing the simulation using a framework different from Pythia 8, a converter script is available (see details in https://github.com/lorenzetti-hep/lorenzetti#readme).

The main information available at the HIT file produced in the Lorenzetti Showers framework are presented in Table A.5. This file may be changed according to the used detector geometry. Descriptions of the contents of the ESD and AOD files are presented in Tables A.6 and A.7, respectively.

Some example files are made available at this address: https://github.com/lorenzetti-hep/lorenzetti/tree/master/examples. There are four files from single electrons simulation (of EVT, HIT, ESD and AOD types), which may be used as templates for users interested in adapting the framework for different physics processes production.

Table A.4
Variables stored in the output file of EVT generation step.

Output File	Attribute	Content Description
EVT	pdg_id	Index related to particle type.
	e	Truth particle energy.
	et	Truth particle transverse energy
	eta, phi	Truth particle coordinate
	px, py, pz	Particle momentum in Cartesian coordinates.
	prod_x, prod_y, prod_z	Production vertex coordinates
	prod_t	Temporal production vertex coordinate

Table A.5
Variables stored in the output file of HIT simulation step.

Output File	Attribute	Content Description
HIT	Sampling	Integer related to detector layer
	Detector	Integer associated to the Lorenzetti region
	eta, phi	Detector-particle interaction coordinate
	Δ_η, Δ_ϕ	Region of interest size.
	bcid_{start}, bcid_{end}, $\text{bc}_{duration}$	Bunch crossing information.
	hash	Cell number identification.
	AvgMu	Pileup information.
	eventNumber	Number of the event. Starts with 0.

Appendix B. Installation and running

To proper run the Lorenzetti framework in a standalone machine, the Singularity [88] application needs to execute a Docker container. All the Lorenzetti prerequisites are available in the container: Geant4, Pythia8, Gaugi, ROOT, etc. To download the Docker image by using the singularity, the following commands are needed:

```
singularity download docker://lorenzetti/lorenzetti
singularity run lorenzetti_base.sif
```

The environment setup, inside the Singularity environment, is made by running a script in the root directory:

```
source setup_envs.sh
```

There is a set of bash scripts in the examples folder to obtain different types of event generation and also for minimum bias data production. Those scripts cover all the Lorenzetti steps: from event generation to final reconstruction.

References

[1] D. Edmunds, P. Laurens, Nucl. Instrum. Methods Phys. Res., Sect. A 598 (2009) 334–339, https://doi.org/10.1016/j.nima.2008.08.036.
[2] ATLAS Collaboration, in: Proc. of 27th International Symposium on Nuclear Electronics & Computing, Becici, Budva, Montenegro, 2019.

Table A.6
Variables stored in the output file of ESD digitization step.

Output File	Attribute	Content Description
ESD	Cell_e	Reconstructed cell energy.
	Cell_eta, Cell_phi	Cell coordinates.
	Cell_deta, Cell_dphi	Cell size (layer dependent).
	Cell_descriptor_link	Integer that connects cell and its hash.
	Cell_sampling	Layer that the cell belongs to.
	Cell_detector	Region that the cell belongs to.
	Cell_rmin, Cell_rmax	Minimum and maximum of cell thickness.
	Cell_edep	Truth cell energy deposition.
	Cell_bcid_start, Cell_bcid_end	Starting and ending bunch crossing index.
	Cell_bc_duration	Duration of each reconstructed cell.
	Cell_pulse	Pulse generated by each interaction (bunch crossing dependent).
	Cell_hash, Cell_link	Integers that connect description to the cells.

Table A.7
Variables stored in the output file of AOD simulation step.

Output File	Attribute	Content Description
AOD	ShowerShapes (Cluster)	Shower variables (high level information)
	Cluster	Cell Cluster around the RoI (0.4×0.4)
	Cluster E_T	Cluster transverse energy
	Cluster η, ϕ	η and ϕ Coordinates of the cluster's barycenter
	Cluster Δ_η	Difference of η cluster coordinates in relation to the hottest cell
	Cluster Δ_ϕ	Difference of ϕ cluster coordinate in relation to the hottest cell
	Ringer Signatures	Concentric rings signals around the hottest cell

[3] R. Wigmans, Calorimetry: Energy Measurement in Particle Physics, 2nd edition, International Series of Monographs on Physics, Oxford University Press, Oxford, 2017.
[4] CMS Collaboration, in: The European Physical Society Conference on High Energy Physics (EPS-HEP2017), in: PoS Proceedings of Science, vol. 314, 2017.
[5] S. Shimizu, et al., J. Phys. Conf. Ser. 513 (2014) 012034, https://doi.org/10.1088/1742-6596/513/1/012034.
[6] M. Dam, et al., J. Phys. Conf. Ser. 219 (2010) 032006, https://doi.org/10.1088/1742-6596/219/3/032006.
[7] T. Ciodaro, J.M. de Seixas, A. Cerqueira, IEEE Trans. Nucl. Sci. 61 (2) (2014) 1047–1055, https://doi.org/10.1109/TNS.2014.2305989.
[8] C.W. Fabjan, D. Fournier, Particle Physics Reference Library - vol. 2: Detectors for Particles and Radiation, Springer, 2020 (Ch. Calorimetry).
[9] W.H. Smith, Triggering and High-Level Data Selection, Springer International Publishing, 2020, pp. 533–554.
[10] P. Cattaneo, et al., J. Instrum. 12 (06) (2017) C06004, https://doi.org/10.1088/1748-0221/12/06/c06004.
[11] K. Pretzl, Nucl. Instrum. Methods Phys. Res., Sect. A 454 (2000) 114–127, https://doi.org/10.1016/S0168-9002(00)00812-3, arXiv:hep-ex/9911028.
[12] S. Alnussirat, et al., in: Proc. of 37th Inter. Cosmic Ray Conference — PoS(ICRC2021), vol. 395, 2021, p. 137.
[13] F. Cadoux, et al., IEEE Trans. Nucl. Sci. 55 (2) (2008) 817–821, https://doi.org/10.1109/TNS.2008.918517.
[14] I. Cojocari, et al., Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 1046 (2023) 167662, https://doi.org/10.1016/j.nima.2022.167662.
[15] A. Giachero, J. Phys. Conf. Ser. 841 (2017), https://doi.org/10.1088/1742-6596/841/1/012027.
[16] S. Magill, J. Phys. Conf. Ser. 404 (2012) 012035, https://doi.org/10.1088/1742-6596/404/1/012035.
[17] A. Volte, et al., IEEE Trans. Nucl. Sci. 69 (4) (2022) 840–848, https://doi.org/10.1109/TNS.2022.3150148.
[18] P. Lecoq, Scintillation Detectors for Charged Particles and Photons, Springer International Publishing, 2020, pp. 45–89.
[19] R. Wigmans, in: 11th Pisa Meeting on Advanced Detectors, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 617 (1) (2010) 129–133, https://doi.org/10.1016/j.nima.2009.09.118.
[20] W.R. Leo, Electronics for Pulse Signal Processing, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, pp. 277–302.
[21] V. Radeka, Signal Processing for Particle Detectors, Springer International Publishing, 2020, pp. 439–484.
[22] R. Forty, O. Ullaland, Particle Identification: Time-of-Flight, Cherenkov and Transition Radiation Detectors, Springer International Publishing, 2020, pp. 281–335.
[23] G. Soyez, Phys. Rep. 803 (2019) 1–158, https://doi.org/10.1016/j.physrep.2019.01.007.
[24] J. Labbé, R. Ishmukhametov, Crosstalk Measurements in the Electromagnetic Calorimeter during ATLAS Final Installation, Tech. Rep. ATL-LARG-INT-2009-004, ATL-COM-LARG-2008-012, CERN, Geneva, Dec. 2008, https://cds.cern.ch/record/1143373.
[25] M. Titov, J. Instrum. 15 (10) (2020) C10023, https://doi.org/10.1088/1748-0221/15/10/c10023.
[26] A. Zaborowska, J. Phys. Conf. Ser. 1162 (2019) 012011, https://doi.org/10.1088/1742-6596/1162/1/012011.
[27] B.S.-M. Peralva, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 1038 (2022) 166951, https://doi.org/10.1016/j.nima.2022.166951.
[28] F. Carrió, IEEE Trans. Nucl. Sci. 69 (4) (2022) 687–695, https://doi.org/10.1109/TNS.2022.3143233.
[29] L. De Oliveira, B. Nachman, M. Paganini, Nucl. Instrum. Methods Phys. Res., Sect. A 951 (2020) 162879, https://doi.org/10.1016/j.nima.2019.162879, arXiv:1806.05667.
[30] M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez, Front. Big Data 3 (2020) 591315, https://cds.cern.ch/record/2709269.
[31] A. Papa, M. De Gerone, S. Dussoni, L. Galli, D. Nicolo, G. Signorelli, Nucl. Phys. B, Proc. Suppl. 248–250 (2014) 115–117, https://doi.org/10.1016/j.nuclphysbps.2014.02.021.
[32] A. Para, H.-J. Wenzel, M. Demarteau, R.-y. Zhu, S. Magill, in: CPAD Instrumentation Frontier Workshop 2021, NordiCHI, 2021, Online event.
[33] T. Peitzmann, et al., Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 1045 (2023) 167539, https://doi.org/10.1016/j.nima.2022.167539.
[34] V. Ghenescu, Y. Benhammou, in: Proceedings of the Vienna Conference on Instrumentation 2016, Nucl. Instrum. Methods Phys. Res., Sect. A 845 (2017) 515–519, https://doi.org/10.1016/j.nima.2016.05.039.
[35] V. Mikhaylov, A. Kugler, V. Kushpil, O. Svoboda, P. Tlustý, M. Golubeva, F. Guber, A. Ivashkin, S. Morozov, V. Klochkov, I. Selyuzhenkov, A. Senger, in: ISHEPP XXIV, EPJ Web Conf. 204 (2010) 11004.
[36] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, Adaptive Computation and Machine Learning Series, MIT Press, 2016, https://books.google.com.br/books?id=Np9SDQAAQBAJ.
[37] A. Radovic, M. Williams, D. Rousseau, et al., Nature 560 (2018) 41–48, https://doi.org/10.1038/s41586-018-0361-2.
[38] D. Bourilkov, Int. J. Mod. Phys. A 34 (2019) 1930019, https://doi.org/10.1142/S0217751X19300199.
[39] F. Psihas, M. Groh, C. Tunnell, K. Warburton, Int. J. Mod. Phys. A 35 (33) (2020) 2043005, https://doi.org/10.1142/S0217751X20430058, arXiv:2008.01242.
[40] D. Guest, K. Cranmer, D. Whiteson, Annu. Rev. Nucl. Part. Sci. 68 (2018) 161–181, https://doi.org/10.1146/annurev-nucl-101917-021019, arXiv:1806.11484.
[41] M. Andrews, M. Paulini, S. Gleyzer, B. Poczos, Comput. Softw. Big Sci. 4 (2018) 6, https://doi.org/10.1007/s41781-020-00038-8, 14 pages, 5 figures; v3: published version, arXiv:1807.11916.
[42] D. Belayneh, F. Carminati, A. Farbin, et al., Eur. Phys. J. C 80 (2020) 165326, https://doi.org/10.1140/epjc/s10052-020-8251-9.
[43] W.S. Freund, et al., J. Phys. Conf. Ser. 1525 (1) (2020) 012076, https://doi.org/10.1088/1742-6596/1525/1/012076.
[44] S. Summers, et al., J. Instrum. 15 (2020) P05026, https://doi.org/10.1088/1748-0221/15/05/P05026, arXiv:2002.02534.

[45] J. Duarte, et al., J. Instrum. 13 (07) (2018) P07027, https://doi.org/10.1088/1748-0221/13/07/p07027.
[46] R. Aaij, et al., Comput. Softw. Big Sci. 4 (7) (2020) 817–821, https://doi.org/10.1007/s41781-020-00039-7.
[47] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th edition, Morgan Kaufmann, Amsterdam, 2017.
[48] J. Apostolakis, Particle Physics Reference Library - vol. 2: Detectors for Particles and Radiation, Springer, 2020 (Ch. Detector Simulation).
[49] Particle Data Group, Prog. Theor. Exp. Phys. 2020 (8) (2020) 083C01, https://doi.org/10.1093/ptep/ptaa104.
[50] S. Agostinelli, et al., Nucl. Instrum. Methods Phys. Res., Sect. A 506 (3) (2003) 250–303, https://doi.org/10.1016/S0168-9002(03)01368-8.
[51] T. Sjöstrand, Comput. Phys. Commun. 246 (2020) 106910, https://doi.org/10.1016/j.cpc.2019.106910.
[52] S. Banerjee, J. Phys. Conf. Ser. 396 (2) (2012) 022003, https://doi.org/10.1088/1742-6596/396/2/022003.
[53] N. Madysa, et al., EPJ Web Conf. 214 (2019) 02006, https://doi.org/10.1051/epjconf/201921402006.
[54] G. Grindhammer, M. Rudowicz, S. Peters, Nucl. Instrum. Methods Phys. Res., Sect. A 290 (2) (1990) 469–488, https://doi.org/10.1016/0168-9002(90)90566-O.
[55] R. Rahmat, R. Kroeger, A. Giammanco, J. Phys. Conf. Ser. 396 (6) (2012) 062016, https://doi.org/10.1088/1742-6596/396/6/062016.
[56] H. Ahmed, in: 25th International Conference on Computing in High Energy and Nuclear Physics, 2021, http://cds.cern.ch/record/2767570.
[57] J. de Favereau, et al., J. High Energy Phys. 57 (2014) 1–25, https://doi.org/10.1007/JHEP02(2014)057.
[58] M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. D 97 (1) (Jan. 2018), https://doi.org/10.1103/physrevd.97.014021.
[59] G.R. Khattak, et al., Eur. Phys. J. C 82 (4) (2022) 386, https://doi.org/10.1140/epjc/s10052-022-10258-4.
[60] B. Marzocchi, Nucl. Instrum. Methods Phys. Res., Sect. A 962 (2020) 160181, https://doi.org/10.1016/j.nima.2017.10.035.
[61] B. Deng, et al., Nucl. Instrum. Methods Phys. Res., Sect. A 981 (2020) 164495, https://doi.org/10.1016/j.nima.2020.164495.
[62] C.A. Beteta, et al., in: Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011), Phys. Proc. 37 (2012) 1744–1749, https://doi.org/10.1016/j.phpro.2012.02.498.
[63] M. Antonello, et al., in: Proceedings of the Vienna Conference on Instrumentation 2019, Nucl. Instrum. Methods Phys. Res., Sect. A 958 (2020) 162063, https://doi.org/10.1016/j.nima.2019.04.017.
[64] M.-R. Mohammadian-Behbahani, S. Saramad, Nucl. Instrum. Methods Phys. Res., Sect. A 951 (2020) 163013, https://doi.org/10.1016/j.nima.2019.163013.
[65] Z. Marshall, the ATLAS Collaboration, J. Phys. Conf. Ser. 513 (2) (2014) 022024, https://doi.org/10.1088/1742-6596/513/2/022024.
[66] M. Preston, et al., Nucl. Instrum. Methods Phys. Res., Sect. A 1011 (2021) 165601, https://doi.org/10.1016/j.nima.2021.165601.
[67] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lönnblad, J. Monk, H. Schulz, F. Siegert, Comput. Phys. Commun. 184 (12) (2013) 2803–2819, https://doi.org/10.1016/j.cpc.2013.05.021.
[68] A. Buckley, D. Kar, K. Nordström, SciPost Phys. 8 (2) (Feb. 2020), https://doi.org/10.21468/scipostphys.8.2.025.
[69] J.Y. Araz, B. Fuks, G. Polykratis, Eur. Phys. J. C 81 (4) (Apr. 2021), https://doi.org/10.1140/epjc/s10052-021-09052-5.
[70] J. Colas, C. de La Taille, R. Lafaye, N. Massol, P. Pralavorio, D. Sauvage, L. Serin, Crosstalk in the ATLAS electromagnetic calorimeter, Tech. Rep. ATL-LARG-2000-004, CERN, Geneva, Oct. 2000, https://cds.cern.ch/record/683952.
[71] G. Barrand, et al., Comput. Phys. Commun. 140 (2001) 45–55, https://doi.org/10.1016/S0010-4655(01)00254-5.
[72] L.R. Evans, P. Bryant, J. Instrum. 3 (2008) S08001, https://doi.org/10.1088/1748-0221/3/08/S08001.
[73] T. Novak, in: CHEP 2018, EPJ Web Conf. 214 (2019) 02044, https://doi.org/10.1051/epjconf/201921402044.
[74] S. Haykin, B.V. Veen, Signals and Systems, 2nd edition, Wiley, New Delhi, 2008.
[75] J.L. Marin, in: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), 2020, pp. 361–366.
[76] E. Fullana, et al., Optimal Filtering in the ATLAS Hadronic Tile Calorimeter, ATLAS Internal Notes ATL-TILECAL-2005-001, 2005, https://cds.cern.ch/record/816152.
[77] P. Adzic, et al., Eur. Phys. J. C 46 (2006) 23–35, https://doi.org/10.1140/epjcd/s2006-02-002-x.
[78] W. Cleland, E. Stern, Nucl. Instrum. Methods Phys. Res., Sect. A 338 (2) (1994) 467–497, https://doi.org/10.1016/0168-9002(94)91332-3.
[79] B. Peralva, J. Seixas, L. Filho, A. Cerqueira, J. Instrum. 16 (02) (2021) P02016, https://doi.org/10.1088/1748-0221/16/02/p02016.
[80] L.M. de A. Filho, B.S. Peralva, J.M. de Seixas, A.S. Cerqueira, IEEE Trans. Nucl. Sci. 62 (6) (2015) 3265–3273, https://doi.org/10.1109/TNS.2015.2481714.
[81] A. Papoulis, Probability, Random Variables, and Stochastic Processes, 4th edition, McGraw-Hill, Boston, 2002.
[82] I.T. Jolliffe, J. Cadima, Philos. Trans. R. Soc. A 374 (2065) (2016) 20150202, https://doi.org/10.1098/rsta.2015.0202.
[83] J. Seixas, L. Caloba, L. Pinto, in: Proceedings of the 39th Midwest Symposium on Circuits and Systems, vol. 2, vol. 2, 1996, pp. 843–846.
[84] J. Seixas, L. Caloba, M. Souza, A. Braga, A. Rodrigues, Comput. Phys. Commun. 95 (2) (1996) 143–157, https://doi.org/10.1016/0010-4655(96)00012-4.
[85] G. Aad, B. Abbott, et al., Eur. Phys. J. C 80 (2020) 47, https://doi.org/10.1140/epjc/s10052-019-7500-2.
[86] G. Aad, et al., J. Instrum. 14 (12) (2019) P12006, https://doi.org/10.1088/1748-0221/14/12/p12006.
[87] G. Aad, ATLAS Collaboration, J. Instrum. 15 (11) (2020) P11016, https://doi.org/10.1088/1748-0221/15/11/p11016.
[88] SingularityCE Developers, SingularityCE 3.8.2, https://doi.org/10.5281/zenodo.5564914, Aug. 2021.