



## OPEN ACCESS

## EDITED BY

Arianna Morozzi,  
Istituto Nazionale di Fisica Nucleare di Perugia,  
Italy

## REVIEWED BY

Juergen Scheins,  
Helmholtz Association of German Research  
Centres (HZ), Germany  
Jean Michel Létang,  
Institut National des Sciences Appliquées de  
Lyon (INSA Lyon), France

## \*CORRESPONDENCE

Olga Kochebina,  
✉ [olga.kochebina@universite-paris-saclay.fr](mailto:olga.kochebina@universite-paris-saclay.fr)

RECEIVED 15 September 2023

ACCEPTED 28 February 2024

PUBLISHED 21 March 2024

## CITATION

Kochebina O, Bonifacio DAB, Konstantinou G,  
Paillet A, Pommranz CM, Razdevšek G, Sharyy V,  
Yvon D and Jan S (2024), New GATE Digitizer  
Unit for versions post v9.3.  
*Front. Phys.* 12:1294916.  
doi: 10.3389/fphy.2024.1294916

## COPYRIGHT

© 2024 Kochebina, Bonifacio, Konstantinou,  
Paillet, Pommranz, Razdevšek, Sharyy, Yvon and  
Jan. This is an open-access article distributed  
under the terms of the [Creative Commons  
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,  
distribution or reproduction in other forums is  
permitted, provided the original author(s) and  
the copyright owner(s) are credited and that the  
original publication in this journal is cited, in  
accordance with accepted academic practice.  
No use, distribution or reproduction is  
permitted which does not comply with these  
terms.

# New GATE Digitizer Unit for versions post v9.3

Olga Kochebina<sup>1\*</sup>, Daniel A. B. Bonifacio<sup>2,3</sup>,  
Georgios Konstantinou<sup>4</sup>, Adrien Paillet<sup>1</sup>,  
Christian M. Pommranz<sup>5,6</sup>, Gašper Razdevšek<sup>7</sup>,  
Viatcheslav Sharyy<sup>1,8</sup>, Dominique Yvon<sup>1,8</sup> and Sebastien Jan<sup>1</sup>

<sup>1</sup>BioMaps Laboratory, Service Hospitalier Frédéric Joliot, Commissariat à l'Énergie Atomique et aux Énergies Alternatives, Université Paris-Saclay, Orsay, France, <sup>2</sup>Nuclear and Energy Research Institute (IPEN/CNEN), Radiation Metrology Center, São Paulo, Brazil, <sup>3</sup>Instituto De Instrumentacion Para Imagen Molecular (I3M), Centro Mixto CSIC-UPV, Valencia, Spain, <sup>4</sup>Multiwave Metacystal S. A, Geneva, Switzerland, <sup>5</sup>Werner Siemens Imaging Center, Department of Preclinical Imaging and Radiopharmacy, Eberhard Karls University Tuebingen, Tuebingen, Germany, <sup>6</sup>Institute for Astronomy and Astrophysics, Eberhard Karls University Tuebingen, Tuebingen, Germany, <sup>7</sup>Experimental Particle Physics Department, Jožef Stefan Institute, Ljubljana, Slovenia, <sup>8</sup>Département de Physique des Particules, Université Paris-Saclay, Commissariat à l'Énergie Atomique et aux Énergies Alternatives, Institut de Recherche sur les lois Fondamentales, Gif-sur-Yvette, France

The Digitizer Unit plays an important role in modeling using Geant4 Application for Tomographic Emission (GATE), a Geant4-based platform used for numerical simulations in medical imaging and radiotherapy. It simulates the response of the photodetection components using a sequence of analytical and semi-analytical models. The Digitizer Unit was written for the first version of GATE approximately 20 years ago. Since then, it has in parts grown in a code that can be hardly maintained. Some parts of the code were unused or duplicated; some of the functionalities were not working anymore. Therefore, the GATE Digitizer Unit update is required in order to incorporate the novelties of Geant4 to update its current version and add new features. In this article, the implementation of the new GATE Digitizer Unit (since version 9.3) is presented. Added functionalities, the impact of changes on users, the current status of the work, and perspectives are discussed.

## KEYWORDS

medical imaging, simulation software, Monte Carlo simulations, radiation detectors, simulation workflows

## 1 Introduction

The Geant4 Application for Tomographic Emission (GATE) [1] is a powerful platform used for Monte Carlo simulations in medical imaging and radiotherapy, which is based on the widely used Geant4 (G4) Monte Carlo simulation toolkit [2]. GATE offers a range of advanced features and tools to simulate interactions of particles with matter, including detailed modeling of particle transport, energy deposition, and detection. One of the key strengths of GATE is its ease of use, which makes it an ideal choice for users who need to implement complex simulations without having any knowledge in programming. This is possible as GATE, which is implemented in C++, enables users to write custom simulations in the form of *macros* with pre-defined commands. Similar alternative projects include GAMOS [3], TOPAS [4], and SimPET [5]. GATE can be downloaded from GitHub at <https://github.com/OpenGATE/Gate> or from the GATE website <http://www.opengatecollaboration.org/>. It can be installed on Linux systems and macOS, deployed as a virtual machine (the so-called vGATE) or from a Docker container image (<https://hub.docker.com/r/>

TABLE 1 Nomenclature table.

Name	Short description
Digitizer Unit	Part of the GATE code responsible for simulating realistic behaviors of the signal readout and the electronic distortions
<i>Digitizer Manager</i>	Specific class that manages all functionalities of the Digitizer Unit
<i>Digitizer Module</i>	Module allowing to incorporate the individual electronics behavior
<i>Sensitive Detector</i>	Detection element (e.g., crystal), a geometric element “sensitive” to the passage of particles
<i>Pulse</i>	GATE object related to the scintillation and/or analog pulse (used only in versions before 9.3)
<i>Hits</i>	Object corresponding to an interaction within a <i>Sensitive Detector</i> volume
<i>HitsCollection</i>	Set of <i>Hits</i> within the same generated event
<i>Digis/Singles</i>	Objects created using information about <i>Hits</i> and/or other <i>Digis</i> by a <i>Digitizer Module</i>
<i>DigiCollection</i>	Set of <i>Digis</i> within the same generated event
<i>Systems</i>	Predefined geometry template that is used to simulate scanners with its key components organized in a specific way
<i>CoincidenceDigis</i>	Objects storing couples of <i>Singles</i> in coincidence within a defined time window, corresponding to coincidence events in PET <i>Systems</i>
<i>Coincidences</i>	<i>CoincidenceDigis</i>
<i>Singles Digitizer</i>	Chain of <i>Singles Digitizer Modules</i>
<i>Coincidence Sorter</i>	GATE object combining <i>Singles</i> in Coincidence
<i>Coincidence Digitizer</i>	Chain of <i>Coincidence Digitizer Modules</i>

Specific to Gate Digitizer Unit terms are in italics.

[opengatecollaboration/gate](https://opengatecollaboration/gate)). It is also supported by a collection of proposed examples (<https://github.com/OpenGATE/GateContrib>) and tools (<https://github.com/OpenGATE/GateTools>).

This work is focused on the *GATE Digitizer Unit* (GDU), which is an essential component of the GATE workflow for imaging applications, such as positron emission tomography (PET) [6], computed tomography (CT), single-photon emission computed tomography (SPECT), and Compton camera (CC) [7]. It allows the simulation of realistic behaviors of the signal readout and the electronic distortions, ranging from the construction of single events to the production of coincidences. The GDU is widely used in medical imaging simulations. Its importance could also be illustrated by a recently added TOPAS extension [8] with a similar functionality.

This article aims to explain the GDU code modernization introduced with GATE v9.3. This update is important as the previous GDU was developed almost two decades ago, and since then, the developers of GATE have changed several times. It is also essential to follow and incorporate the new Geant4 features added since then, for example, *G4DigiManager*, a singleton class which manages the digitization processes. Moreover, the modernization of the code aims to address the issue of redundancies in the code of the previous version. Finally, we removed some bugs and inefficiencies, resulting in a more stable and reliable version. At the same time, this update also aims to minimize the discomfort to users by providing the same functionalities as before, consistent results between the old and new GDU, updated documentation, and a conversion tool for user macros. Several new features are proposed, allowing for more flexibility for simulations and expanding the range of potential applications discussed later in the article.

The secondary purpose of this article is to provide a comprehensive picture of the new GDU class organization and a general description of its architecture for future developers.

## 2 New GDU framework

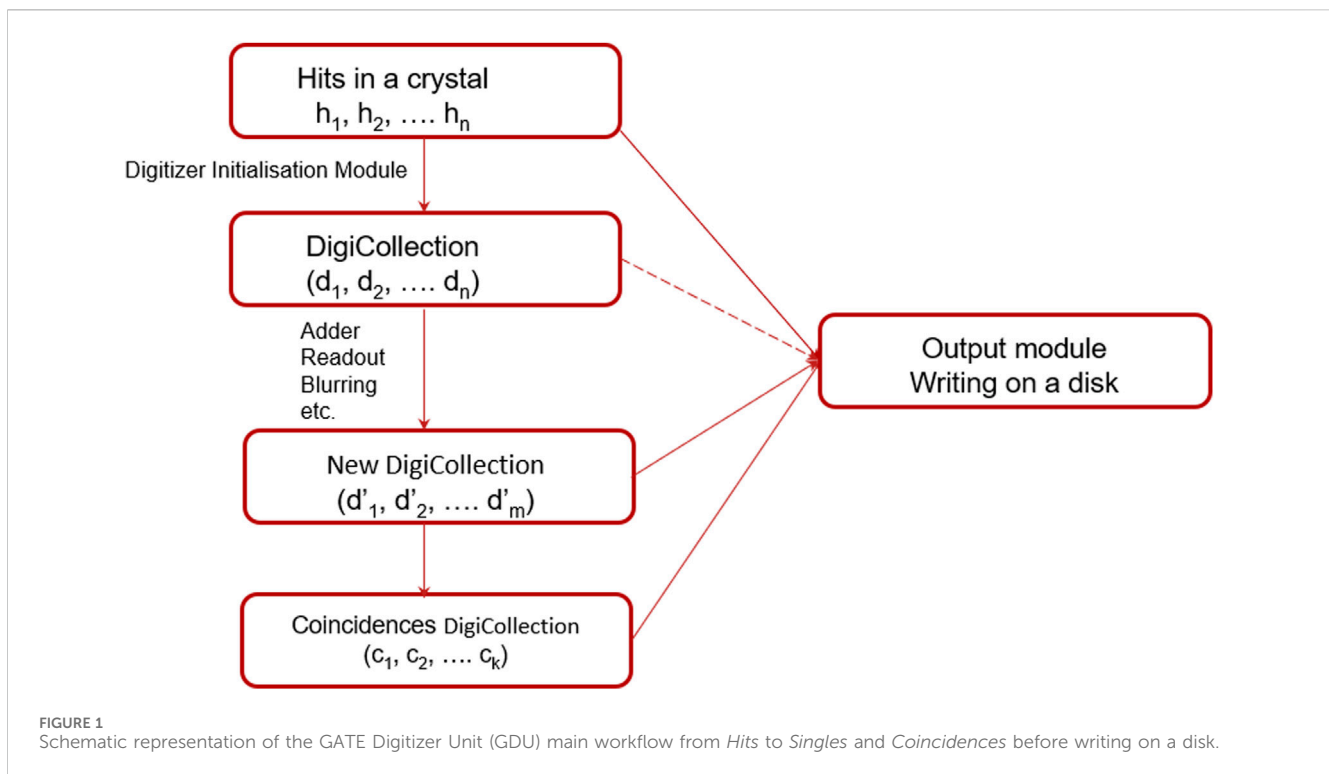
In this section, many GDU-specific names are used. Short definitions are given in Table 1.

The main purpose of the *GATE Digitizer Unit* is to reproduce a realistic behavior of the entire signal processing chain, from the interaction of particles, the so-called *Hits*, with the detector, *Sensitive Detector*, to the final digital output signal, *Digis*, that are also called *Singles* and used to construct *CoincidenceDigis* or, simply, *Coincidences*.

The GDU is based on analytic and semi-analytic models called in a specific sequence that simulates various sensor distortion effects, affecting energy, position, and time resolutions. The simulation aspects comprise transport and detection efficiencies, quantum efficiency, energy window cut, dead time, pile up, noise, and other relevant parameters.

In GATE, the GDU is built upon the terms and principles of Geant4. It uses the same fundamental concepts, inherits virtual classes, and uses template classes from G4 to implement its main functionalities. These G4 virtual classes provide a framework and predefined methods for handling the digitization process, allowing customization and integration with different detector systems. However, through the years of development of the GDU, it fell behind recent Geant4 developments. Moreover, the general architecture contained some unnecessary steps in the workflow that were implemented for planned but never realized features. All this led us to revise and modernize the GDU architecture while addressing bugs and redundancies at the same time.

It is important to note that in order to ensure the reliability and stability of the modifications made to the GDU, comprehensive testing was conducted to compare its performance and produced results against the previous version. This meticulous testing



approach instilled confidence in the robustness and functionality of the revised GDU.

This section introduces necessary terms and approaches for the explanation of the modifications, the new architecture, and some interesting *Digitizer Modules*. It is followed by focusing on the lack of flexibility in previous versions, which was addressed with the new GDU.

## 2.1 Digitizer in Geant4: general terms and approaches

One of the motivations for embracing the concepts and advancements of the Geant4 framework is its strong emphasis on long-term robustness. Geant4 is a well-established and widely adopted collaborative software toolkit that has been continuously developed and refined over time. It benefits from regular updates and ongoing follow-ups, ensuring its relevance and reliability in the scientific community.

In Geant4, the Digitizer plays a crucial role in simulating the response of detectors in the form of digital signals converted from particle interactions. Some of the important general terms and approaches are introduced below.

The Geant4 Digitizer can be managed by the G4DigiManager class, and it takes as input a list of interactions (*Hits*, G4VHit) within a crystal or detection element (*Sensitive Detector*, G4VSensitiveDetector). They refer to a snapshot of the physical interactions of a particle track or an accumulation of interactions of multiple tracks within the *Sensitive Detector*. The *Hits* represent the actual energy deposited by particles in the detector and provide information about the true interactions occurring within the detector material. They are gathered in a *HitsCollection*

(G4THitsCollection) and are stored at the end of each generated event. The G4 Digitizer converts *Hits* to *Digis* (G4VDigi), representing a detector output, such as an ADC/TDC count or a trigger signal. They are gathered in *DigiCollections* (G4TDigiCollections). Users have the flexibility to define specific classes, called *Digitizer Modules* (G4VDigitizerModule), that allow them to incorporate the desired electronics behavior and accurately reproduce the distortion effects present in real detectors during the digitization process.

## 2.2 New GDU: new GATE digitizer unit

In GATE, the first step of the Digitizer Unit is to take *Hits* (GateHit) created in a *Sensitive Detector* and to construct *Single Digis* (GateDigi) from them, as shown in Figure 1. It is done using the specific module GATE Digitizer Initialization Module. It is followed by user-selected *Digitizer Modules*, which are applied in a sequence. Currently, more than 12 *Digitizer Modules* are available in GATE, as shown in Table 2 and in the example of Figure 1. Each of them mimic one specific hardware response or detection effect.

For PET imaging, there is a dedicated GDU component for constructing *Coincidences* or *CoincidenceDigis*, a vector of *Singles Digis* (GateDigi\*). It initially passes through a *Coincidence Sorter*, where coincidence candidates are constructed. It is possible to have several types of *Coincidences*, for example, *prompt* and *delayed* coincidences for random rate estimation. Subsequently, these coincidences can be merged and processed using dedicated *Coincidence Digitizer Modules* representing the effects of coincidence processing by the electronics.

A typical data flow diagram for a PET simulation using the new GDU is shown in Figure 2. *Hits* produced in a *Sensitive Detector* are

TABLE 2 List of *Digitizer Modules* available in GATE 9.3.

<i>Digitizer Module</i>	Short description
Adder	Adds all <i>Hits</i> in one crystal
Optical adder	Adds all <i>Hits</i> generated by optical photons
Readout	Models readout by an individual photo-detector
Energy resolution	Applies Gaussian blurring on energy
Time resolution	Applies Gaussian blurring on time
Spatial resolution	Applies Gaussian blurring on 3D position
Energy framing	Selects an energy window
Efficiency	Applies detection system sensitivity
Pile-up	Models event pile-up
Dead time	Models detection systems dead time
Noise	Adds background events in a generic way for any kind of noise source
Adder Compton	Specific adder for Compton kinematics
Merger	Merges <i>Singles</i> from different <i>Sensitive Detectors</i>

processed by two *Singles Digitizers*, providing the output collections “Singles” and “High-Energy Singles.” The output of the last one is used as input for two *Coincidence Sorters*, “Coincidences” and “Delayed Coincidences.” Their outputs are combined before processing by the *Coincidence Digitizer* “Dead time” before yielding the “Final Coincidences” as the eventual output.

A full documentation of the new GDU is available on the documentation page [https://opengate.readthedocs.io/en/latest/digitizer\\_and\\_detector\\_modeling.html](https://opengate.readthedocs.io/en/latest/digitizer_and_detector_modeling.html), and several examples can be found in the GateContrib page <https://github.com/OpenGATE/GateContrib/tree/master/imaging> (see examples PET, SPECT, LUT Davis Model, CT, optical, and NumPY output).

### 2.2.1 Architecture changes

The main change in the GATE Digitizer Unit architecture is in its simplification (Figure 1). Previously, between *Hits* and *Digi*, there was an additional intermediate object *Pulse* (GatePulse). It was originally introduced as it is an important component of digitalization in real life. Historically, it was planned to go deeply into the modeling of pulse shapes; however, this was never done so far, and the intermediate step with a *Pulse* remained unused and, thus, unnecessary and parasitic.

Other improvements overcome the limitations of the GDU to handle the digitization and output of multiple *Sensitive Detectors* effectively, which was quite complex or even impossible. This lack of flexibility has been addressed and rectified in the new version of the GDU (Section 2.2.4.1).

Another limitation originating in the GATE System approach is explained below (Section 2.2.4.2). Specifically, the use of a *Sensitive Detector* was only possible if it was part of a predefined System and digitization and coincidence construction. The new GDU effectively addressed this issue as well.

### 2.2.2 Digitizer manager

The main functionality of the GDU is managed by a specific class (*Digitizer Manager*, GateDigitizerMgr), which is inherited from G4DigiManager. User parameters are set through the GateDigitizerMgrMessenger class, where one can insert and choose a name and a type of a specific part from the Digitizer Unit, i.e., *Singles Digitizer*, *Coincidence Sorter*, and *Coincidence Digitizer* explained below, as well as choose to which *Sensitive Detector* the digitization will be applied. The *Digitizer Manager* accesses the list of used *Sensitive Detectors* and supervises the components explained below that perform digitization steps.

#### 2.2.2.1 Singles digitizers

*Singles Digitizers* (inherited from GateModuleListManager) are used for modeling a digital signal from *Hits*, taken as input, and yield *Digis* as output. With a corresponding messenger, the user can select the *Digitizer Modules* (Section 2.2.2.3) listed in Table 2 and input *Digis* if the creation of *Singles* from already existing ones is needed, as shown in Figure 2, where “Singles” are used as input for “High-Energy Singles.” The result from each *Singles Digitizer* can be saved at the end of each event.

#### 2.2.2.2 Coincidence sorters

*Coincidence Sorters* are used to combine *Singles* in *Coincidences*. The *Coincidence Sorter* is inherited directly from GateVDigitizerModule and, thus, from G4VDigitizerModule. It takes *Digis* as input and creates *CoincidenceDigis*, which is a vector of *Digis* and the result of its combination in *Coincidences*. At the beginning of a simulation run, during the first several events, the *Coincidence Sorter* fills a presort buffer with a user-specified size, and once it is filled, the pair combination starts, followed by filtering according to parameters provided by the user [6]. It is possible to set a coincidence time window, offset, jitter, some geometrical parameters for data reduction (such as a minimal distance between two detector sectors), a presort buffer size, and a policy for multiple coincidences.

#### 2.2.2.3 Coincidence digitizers

Coincidence digitizers are used for the further processing of *Coincidences*. They are very similar to *Singles Digitizers* but are applied on *Coincidences* instead of *Singles*. *Coincidence Digitizers* are also inherited from the GateModuleListManager. In this case, the user can choose one or several *Coincidence* input collections and *Coincidence Digitizer Modules*: dead time, buffer, etc.

### 2.2.3 Digitizer modules

Each of the *Digitizer Modules* (inherited from GateVDigitizerModule and, thus, from G4VDigitizerModule), simulate one specific hardware response or detection effect with a semi-analytic or analytic model. For example, the energy, position, and time resolutions of a detector and its electronics can be modeled using a Gaussian function parameterized by the energy resolution, spatial resolution, and time resolution modules, respectively. The general detection efficiencies can also be modeled. Other effects such as dead time, pile-up, and noise can also be added and parameterized by the user. The full list of all available *Digitizer Modules* is given in Table 2.

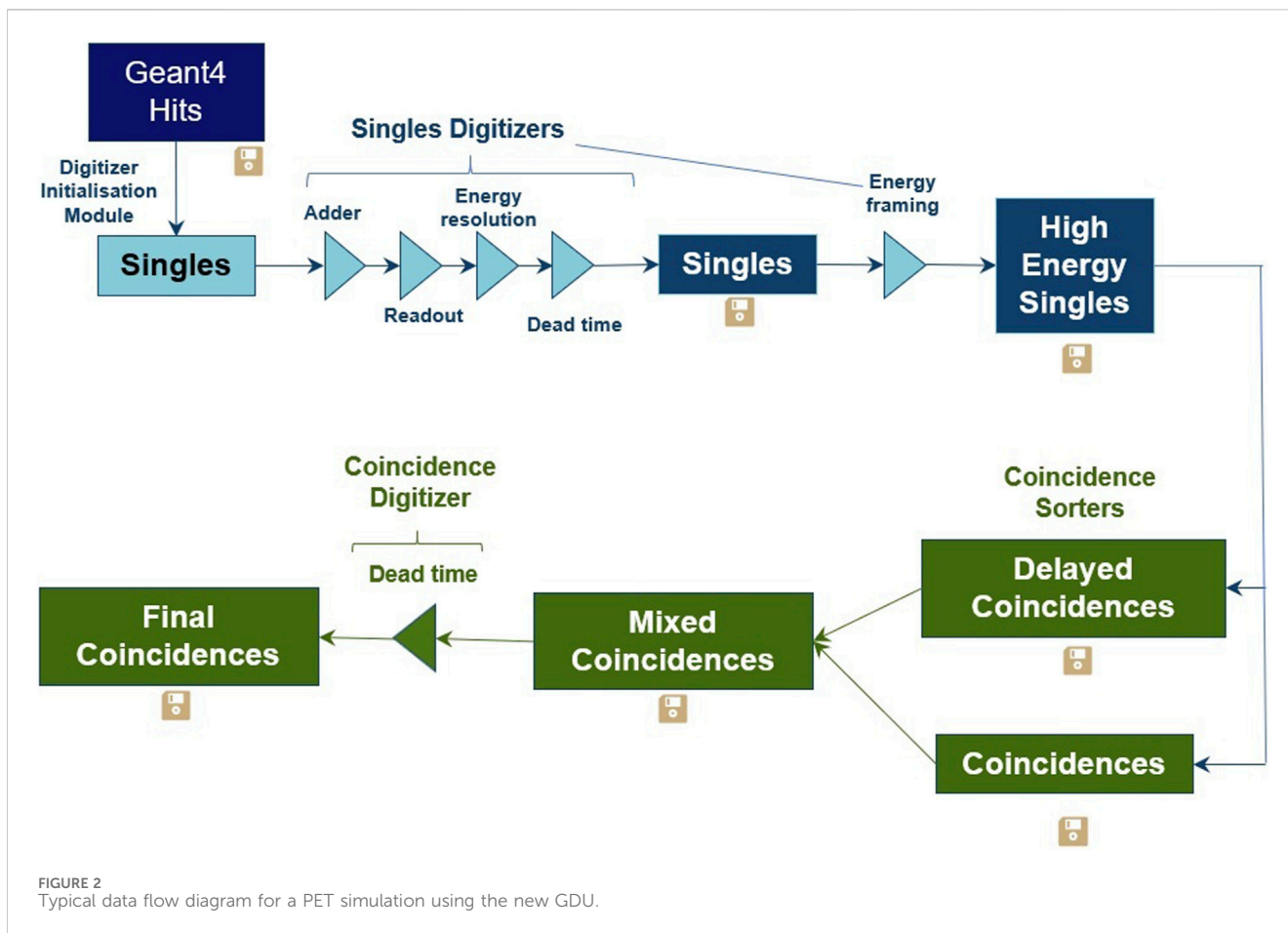


FIGURE 2 Typical data flow diagram for a PET simulation using the new GDU.

### 2.2.3.1 New digitizer module: merger

In some cases, it is useful to merge *Singles* from several different *Sensitive Detectors*, for example, as described in Sections 3.3.1 and 3.3.3. In Section 3.3.3, the *Singles* needed to be mixed and afterward combined into *Coincidences*. Another use case is a system with multi-composed scintillation materials with different characteristics read using a common photo-detector (Section 3.3.1).

## 2.2.4 Addressing the lack of flexibility in previous versions

### 2.2.4.1 Multiple sensitive detectors

Since version 9.3, GATE with the new GDU provides the possibility to have separate outputs for different *Sensitive Detectors* (see example in Section 3.3.2). This feature allows for organizing and storing data from different detector components or subsystems separately, facilitating analysis and post-processing. The simulations with only one *Sensitive Detector* remain backward compatible. Moreover, the macros for new GDU commands can be adapted automatically if needed using a tool described in Section 2.2.6.

### 2.2.4.2 Reconsideration of the GATE system approach

In GATE, a *System* is considered a predefined geometry template that is used to simulate scanners with its key components organized in a specific way as most scanners share the same general

geometrical characteristics. Each component within the *System* plays a distinct role or ordering, contributing to the overall simulation, its digitizer, and output. GATE provides nine *Systems*, including SPECT, PET, and more generic *Systems*. The detection element of a *System* has to be a repeated block of the same geometry and material, and *Coincidences* can be constructed only within the same *System*.

It is important to note that in previous versions of GATE, without a defined *System*, both digitization modeling and coincidence sorting were not possible, leading to limitations and reduced flexibility. This issue particularly affected complex systems and specific functionalities, such as modeling non-standard scanner geometries (example in Section 3.3.3) or constructing *Coincidences* between different detectors (e.g., prototype test bench simulations in Section 3.3.2).

## 2.2.5 Validation of the new GDU

Validation of the results produced by the new GDU against those produced by the old GDU is a mandatory step when transitioning between versions to ensure consistent results. In our case, we decided to validate each functionality separately, applying specific tests: for each *Digitizer Module*, several cases with one and multiple *Sensitive Detectors*, with one and multiple *Singles* collections, and for PET and SPECT (only *Singles*) *Systems*. In total, 45 tests were developed to confirm the repeatability of the previous results.

### 2.2.6 Tool for macros adaptation

One inconvenience that GATE users could encounter during the transition to the new GDU is the need for changes in macro commands. To mitigate this drawback and facilitate the transition process, we developed a conversion tool, `gt_digi_mac_converter`, available in the official GateTools repository on GitHub: <https://github.com/OpenGATE/GateTools>. It analyzes the macro scripts automatically and converts the commands to the new syntax. This conversion tool reduces the manual effort required to update macros and ensures a smoother transition for users.

## 3 Illustration of the new GDU features

### 3.1 Backward compatibility of new GDU results

One of the central requirements for the new GDU is the backward compatibility of its produced results with its previous versions. One of the key tools for ensuring backward compatibility is the continuous integration pipeline of GATE, which is automatically run for every commit in the GATE repository on GitHub. Several benchmarks, tracked in a separate Git repository (<https://github.com/OpenGATE/GateBenchmarks>) are included in the pipeline for the automatic testing of different aspects of GATE software.

To illustrate good agreement of the results between previous and new versions, the simulation studies of the expected performance of an under-development breast PET insert, originally performed with the old GDU and GATE v9.1 [9], were reconducted to compare the results. The macro files of the simulations were translated into GATE v9.3 compatible macros using the `gt_digi_mac_converter` conversion tool. The models and parameters used in the digitizer chain, the selected Geant4 physics list, and the data analysis methods were unchanged to improve comparability of the results. The digitizer chain included the Adder, Readout, Energy Resolution, Efficiency, Time Resolution, Dead Time (paralyzable), Energy Framing, and Coincidence Sorter modules, and the data were stored as ROOT [10] files.

The simulated axial sensitivity profile for one of the three breast PET insert geometries introduced by [9] (geometry C) is shown in Figure 3. The old and new GDUs produced consistent axial sensitivity profiles with a mean relative difference of 0.7% and the largest relative difference of 3.4% at the source position with an offset of 50 mm. The simulated count rates and noise equivalent count rates (NECRs) are shown in Figure 4 and Figure 5, respectively. The count rate results of the old and new GDUs were consistent, and for the count rates at the highest simulated activity of 600 MBq, which are strongly affected also by the Dead Time module and the *Coincidence Sorter* handling of multiple coincidences, the relative differences were lower than 0.7%. The relative difference of the peak NECRs was 0.2%, and the corresponding activity was slightly shifted by one data point of 10 MBq between GATE v9.1 and GATE v9.3 simulations. Overall, the results of the reconducted simulations using GATE v9.3 with the new GDU were in very good agreement with the original simulation results obtained using GATE v9.1 and the old GDU. This demonstrates the backward compatibility of the new GDU also for a realistic digitizer chain consisting of multiple *Digitizer Modules*.

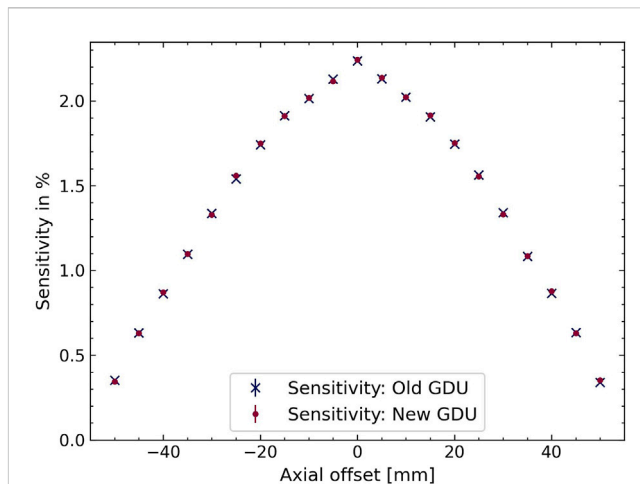


FIGURE 3 Simulated sensitivity of the future breast PET insert (geometry C) obtained using GATE v9.1 (old GDU) and GATE v9.3 (new GDU). The GATE v9.1 data were taken from [9].

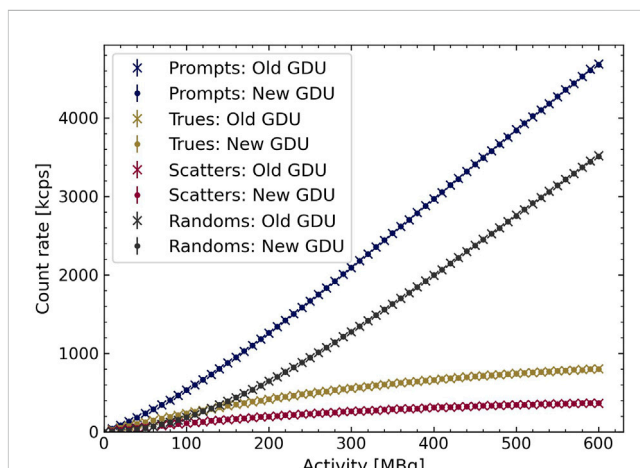
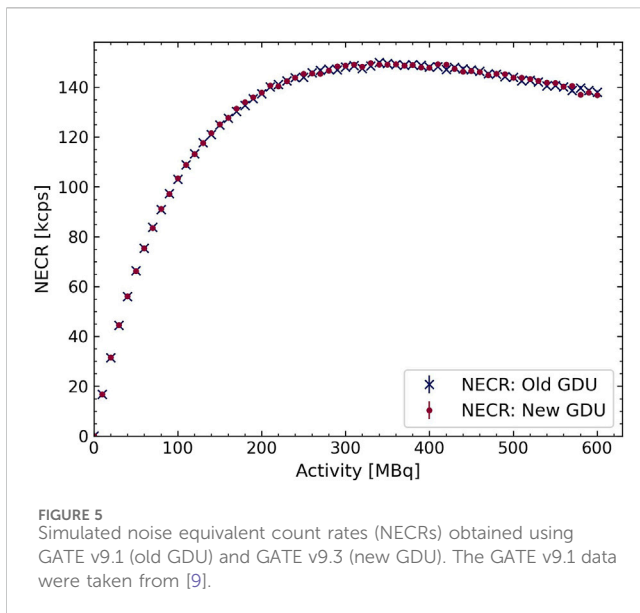


FIGURE 4 Simulated count rates obtained using GATE v9.1 (old GDU) and GATE v9.3 (new GDU). The GATE v9.1 data were taken from [9].

The macros of these tests can be found here [https://figshare.com/articles/dataset/Section3\\_1\\_zip/24981000](https://figshare.com/articles/dataset/Section3_1_zip/24981000).

### 3.2 Execution speedup

One of the key factors influencing the performance of the new GATE code is the significant acceleration achieved through bug fixes and code optimizations introduced between GATE v9.2 and GATE v9.3 with the new GDU. The speedup between these two versions was measured on an Intel® Xeon® Gold 5218R CPU @ 2.10 GHz with 10,000,000 generated primary events. A GATE simulation was performed with a *cylindricalPET System* for a GE SIGNA PET/MR scanner [11] containing 20,160 LYSO crystals. The aim of this test was to check the speedup due to architecture simplifications and the influence of the Digitizer Manager introduction, i.e., speed of the



main GDU “engine;” thus, the digitizer was set as simple as possible: the *Singles Digitizer* contained only an adder module and a *Coincidence Sorter*.

For *Hits* saving, the speedup was approximately 40%, in the case of *Singles* only saving, it was ~12%, and for *Coincidences*, the execution time improved by 6%. The macros of these tests can be found here [https://figshare.com/articles/software/Macros\\_Section3\\_2\\_zip/24981006](https://figshare.com/articles/software/Macros_Section3_2_zip/24981006).

### 3.3 Improved flexibility and application cases

The update of the GDU introduces a capability of using multiple *Sensitive Detectors*. This development allows for the independent configuration of separate digitizers, outputs, and the attachment of *Sensitive Detectors* without the need for a predefined *System*. This flexibility provides access for users to more specific imaging system simulations and even experimental setups.

Some of the application cases are considered below in this section.

#### 3.3.1 Systems with multiple detector types

One of the cases where new GDU features are beneficial is in systems with multiple detector types, such as the metascintillators [12], where detector blocks consist of thin slices of a high-Z and fast emitter scintillators placed next to each other in interleaved mode, allowing scintillation production in both materials (Figure 6). The slices are usually hundreds of micrometers thick. In other works [13, 14], the previous GDU was used to model brain PET systems based on metascintillators to analyze their performances. Due to the limitations of the previous GDU, energy and time resolutions were set from experimental data that consider the whole heterostructure. Now, the new GDU was used to generate more precise energy spectra of two metascintillator-based PET scanners by defining the specific energy resolutions of each material, which were attached to different *Sensitive Detectors* and merged into a unique readout output.

The simulated setup is the same as in [13] and is based on a sensitivity study following the preclinical NEMA guidelines [15].

The detection elements are composed of slices of high-Z material BGO, and the fast materials are BaF2 and EJ232. *Singles* energy spectra are simulated to illustrate the GDU feature explained above.

The obtained results are presented in Figure 7, illustrating the *Singles* energy spectra for BGO/BaF2 and BGO/EJ232 metascintillators. One can observe that the final energy spectra obtained using GATE v9.2 (old GDU) and GATE v9.3 (new GDU) are similar.

In metascintillators, the system response varies stochastically based on gamma energy released in different materials. Events that are shared or fully released in fast materials differ in energy and timing resolution from those in high-Z (slow) materials. The light yield and scintillation emission spectra of the materials are different, and they usually provide different SiPM responses for the same optical coupling and wavelength sensitivity. Therefore, an improved simulation of the energy spectra could be achieved with the GATE energy efficiency function. Managing two *Sensitive Detectors* pulses with the new GDU provides more resources in metascintillator signal analysis and also offers opportunities for applying advanced algorithms. The presented energy analysis is only a preliminary step, being an illustration of the new GDU potential use.

The macros of these tests can be found here [https://figshare.com/articles/software/Section3\\_3\\_1/250463458](https://figshare.com/articles/software/Section3_3_1/250463458).

#### 3.3.2 Prototype experimental setup modeling

An example of an experimental setup that is possible to simulate now with GATE is a test bench for a ClearMind PET detector prototype [16, 17], as shown in Figure 8. Most of the time, such test benches consist in placing a reference detector on one side of a positron source and the detector under the study on the opposite side of it.

In the context of the ClearMind prototype test, the reference detector is a LYSO crystal, and the ClearMind prototype installed on the opposite side is a PbWO<sub>4</sub> crystal.

Now, the GATE output in this case can contain *Hits* and *Singles* coming from the ClearMind detector and LYSO detector separately, as shown in Figure 9. This illustrates the new feature of the GDU.

In current PET detector development, a crucial aspect to investigate is the potential for time-of-flight resolution, characterized by the coincidence resolving time parameter [18]. Consequently, our attention is also directed toward this parameter, necessitating the construction of *Coincidences*. However, within the current GATE framework, this capability remained unavailable, where detection elements must be of the same geometry and material within a *System* for *Coincidences* construction. To address this limitation, we showcase a simulation involving two different *Systems* in GATE v9.3, corresponding to LYSO and ClearMind detectors. In this scenario, one more novel feature of the GDU comes into play: the ability to form *Coincidences* originating from distinct *Systems*. *Singles* from different *Sensitive Detectors* can be merged using a *Merger* (Section 2.2.3.1) and processed afterward using a *Coincidence Sorter*.

The validation of the obtained results is possible only against the experimental measurements and not the old version of the GDU, where such simulations were not possible. The obtained simulated time resolution of 384 ps FWHM is in agreement with a measured CRT of approximately 460 ps FWHM [17]. This discrepancy is expected due to the significant non-Gaussian tail in the measured distribution, which the current GATE model does not account for.

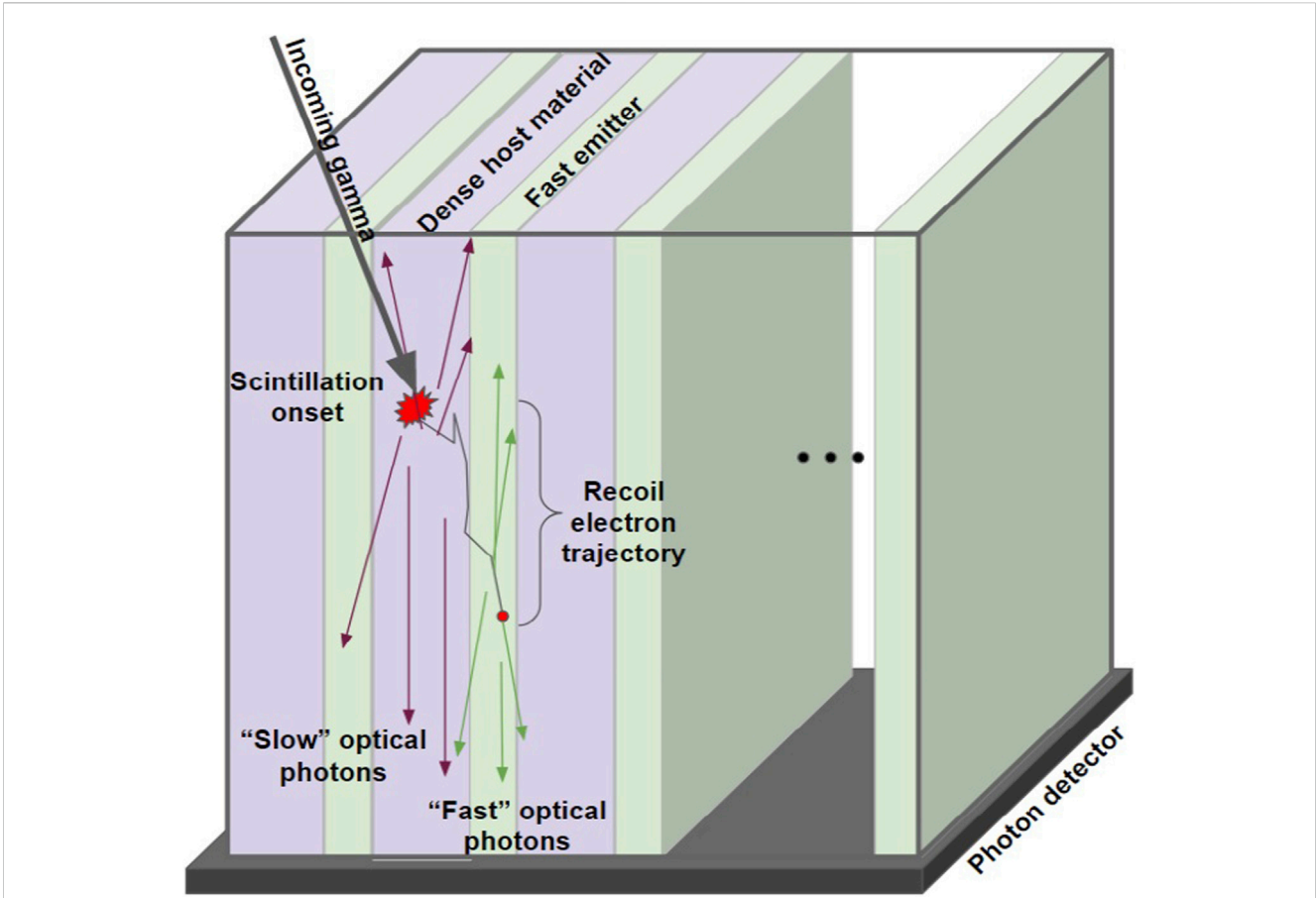


FIGURE 6 Example of a metascintillator topology: Thin slices of a dense host material and a fast emitter are placed next to each other, allowing scintillation production in both.

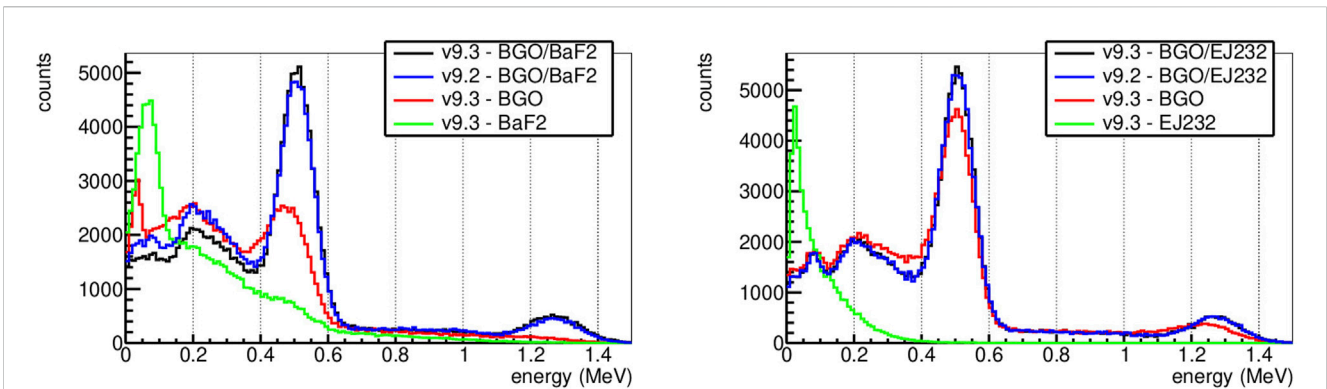


FIGURE 7 Singles energy spectra for the BGO/BaF2 and BGO/EJ232 metascintillators obtained using GATE v9.2 and v9.3.

It is important to note that with GATE, it is now possible to model such prototype setups, but GATE cannot and does not aim to replace a detailed simulation study. However, it can be used as a valuable tool for initial tests of ideas and provide insights for further improvements in the prototype design.

The macros of these tests can be found here [https://figshare.com/articles/software/Macros\\_Section3\\_3\\_2\\_zip/24981003](https://figshare.com/articles/software/Macros_Section3_3_2_zip/24981003).

### 3.3.3 Case of the multi-system setup

The following example of an application case also shows the possibility of GATE v9.3 to construct *Coincidences* coming from several different GATE *Systems*. This example is an extended case of the multipanel limited angle PET system [19], where four flat panels are proposed for PET imaging. The main motivations behind such a geometry are intraoperative surgical applications and monitoring



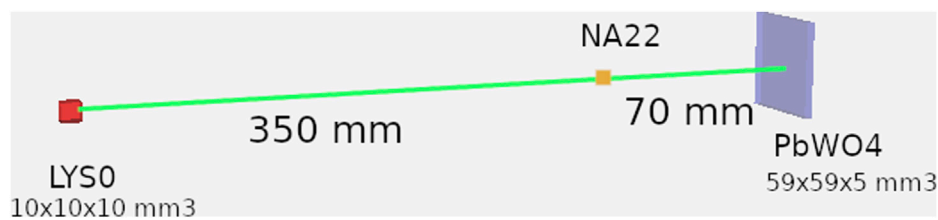


FIGURE 8 Simplified illustration of the ClearMind prototype test bench. More detailed images are given in [17].

```

TFile* test_new_good.root ROOT file with histograms
KEY: TH1D latest_event_ID;1 latest_event_ID(#)
KEY: TH1D total_nb primaries;1 total_nb primaries(#)
KEY: TTree pet_data;1 data for PET analysis
KEY: TTree Hits_crystal_LYSO;1 The root tree for hits
KEY: TTree Hits_crystal_CM;1 The root tree for hits
KEY: TTree OpticalData;1 OpticalData
KEY: TTree Coincidences;1 The root tree for coincidences
KEY: TTree Singles_crystal_LYSO;1 The root tree for singles
KEY: TTree Singles_crystal_CM;1 The root tree for singles
    
```

FIGURE 9 GATE output with the new GDU for the ClearMind prototype test bench.

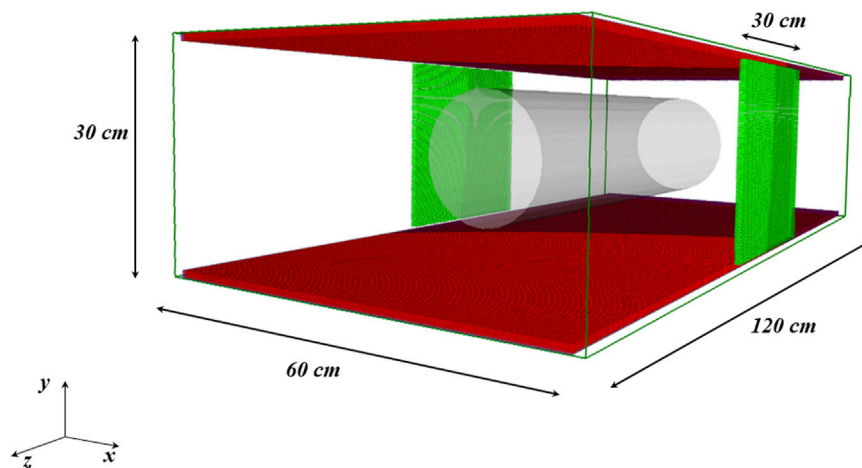
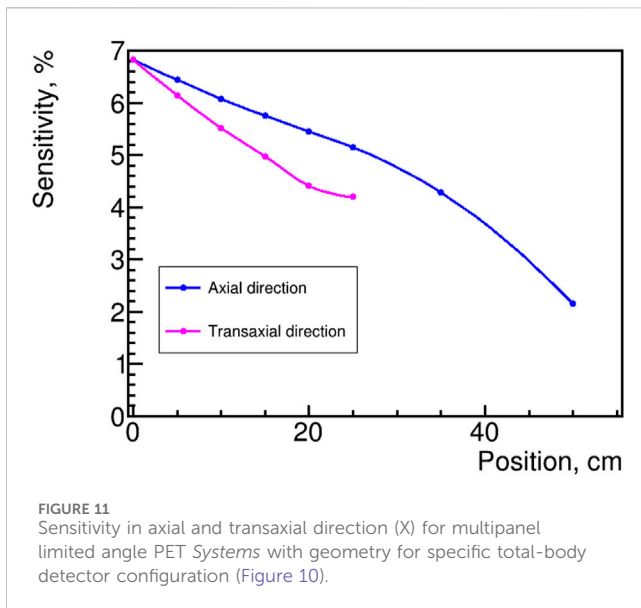


FIGURE 10 Multi-system example: multipanel limited angle PET system from [19] (red) with smaller side panels (green) added.

proton or heavy ion therapies, as well as a potential reduction in the cost and complexity of scanners. The main limitations of such scanners are in limited angular coverage, which generate distorted images with artifacts, and the loss of sensitivity. However, with improved timing resolution (e.g., 50 ps FWHM as in [19]), this limitation could be overcome as there is no need in as many projections, and the angular sampling requirement to obtain distortion-free and artefact-free PET images decreases [19].

The study [19] used GATE v8.1, where setups with two or four same panels work well. Challenges arise when simulating a specific total-body detector configuration, consisting of larger panels above and below the patient and smaller panels at the sides (Figure 10). Due to the varying sizes of these detector modules, creating them within the same GATE System was not feasible. Therefore, the main concern was to construct Coincidences in this case, which is now possible with the new GDU.



The new GDU feature can be illustrated with the sensitivity curves shown in Figure 11, which was not possible to simulate in the previous versions.

The macros of these tests can be found here [https://figshare.com/articles/software/Macros\\_Section3\\_3\\_3\\_zip/24981009](https://figshare.com/articles/software/Macros_Section3_3_3_zip/24981009).

## 4 Discussion and limitations

In this work on the new GDU with a revised architecture, we made significant improvements on the GATE platform in two key aspects: flexibility and general execution speed. First, we expanded it by introducing the ability to independently manage multiple *Sensitive Detectors* and, additionally, by removing constraints on the use of predefined *Systems*. This enhancement allows us to simulate non-standard imaging systems and experimental test benches. Moreover, we achieved a notable acceleration in the general execution speed.

Despite the significant improvements achieved, one inconvenience of the new version is the need for changes in GDU macro commands. To mitigate this drawback and facilitate the transition process, we developed a conversion tool. This tool assists users in adapting their existing macro commands to the updated version. Additionally, while efforts have been made to create more comprehensive commands, there is still room for improvement. For instance, the merger *Digitizer Module* could be further streamlined to enhance user friendliness.

Another limitation of the current GDU version comes from the need of a GATE *System* definition in order to construct *Coincidences*. This drawback will be addressed in the next version of the GDU.

Significant effort has been dedicated to improving GATE, but there are still ongoing tasks and areas that require further development and refinement. The adaptation process of rarely used *Digitizer Modules* and features is still an ongoing effort. The GATE collaboration is actively working on incorporating these modules and features.

In addition, new features are planned for future releases. One of them is the waveform generator, enabling the simulation of detailed waveforms produced by the detector readout electronics. Another planned feature is the offline GDU, providing the separation of the digitization process from the main Monte Carlo simulation.

In the future, an important step is the integration of the Digitizer Unit into GATE 10, as mentioned in [1]. GATE 10 will introduce the use of Python scripts instead of macros, providing users some advantages for specific simulations and analyses. The Digitizer Unit will be integrated into the simulation or will be available for offline use.

## 5 Conclusion

The new GATE Digitizer Unit has brought significant improvements to the GATE platform, particularly in terms of the flexibility to design complex imaging systems and test benches and general speedup. The main new features are 1) independent processing of information from multiple *Sensitive Detectors* and 2) the ability to construct coincidences for non-standard GATE *System* configurations.

The work is ongoing, and future releases of GATE are anticipated to introduce additional features such as the waveform generator, an offline Digitizer Unit, and the integration of the Digitizer Unit into the planned GATE 10 release.

## Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: [https://figshare.com/authors/Olga\\_Kochebina/16937118](https://figshare.com/authors/Olga_Kochebina/16937118).

## Author contributions

OK: writing—original draft, visualization, validation, software, project administration, methodology, investigation, formal analysis, data curation, and conceptualization. DB: writing—original draft, visualization, validation, investigation, formal analysis, and data curation. GK: writing—original draft. AP: writing—review and editing and software. CP: writing—original draft, visualization, validation, investigation, formal analysis, and data curation. GR: writing—review and editing and data curation. VS: writing—review and editing and data curation. DY: writing—review and editing and data curation. SJ: writing—review and editing and supervision.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. DB acknowledges the Brazilian funding agencies Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq grant

agreement No. 202378/2020-9) and Financiadora de Estudos e Projetos (FINEP grant agreement No. 0294/16).

## Acknowledgments

The authors thank Rok Dolenc for providing valuable input for the multipanel limited angle PET system illustration case.

## Conflict of interest

Author GK was employed by Multiwave Metacrystal SA.

## References

- Sarrut D, Arbor N, Baudier T, Borys D, Etxebeste A, Fuchs H, et al. The OpenGATE ecosystem for Monte Carlo simulation in medical physics. *Phys Med Biol* (2022) 67:184001. doi:10.1088/1361-6560/ac8c83
- Allison J, Amako K, Apostolakis J, Arce P, Asai M, Aso T, et al. Recent developments in GEANT4. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2016) 835:186–225. doi:10.1016/j.nima.2016.06.125
- Arce P, Rato P, Canadas M, Lagares JI. GAMOS: a Geant4-based easy and flexible framework for nuclear medicine applications. In: 2008 IEEE Nuclear Science Symposium Conference Record (IEEE); 19 - 25 October 2008; Dresden, Germany (2008). p. 3162–8.
- Faddegon B, Ramos-Méndez J, Schuemann J, McNamara A, Shin J, Perl J, et al. The TOPAS tool for particle simulation, a Monte Carlo simulation tool for physics, biology and clinical research. *Physica Med* (2020) 72:114–21. doi:10.1016/j.ejmp.2020.03.019
- Paredes-Pacheco J, López-González FJ, Silva-Rodríguez J, Efthimiou N, Niñerola-Baizán A, Ruibal Á, et al. SimPET—an open online platform for the Monte Carlo simulation of realistic brain PET data. Validation for 18F-FDG scans. *Med Phys* (2021) 48:2482–93. doi:10.1002/mp.14838
- Strydhorst J, Buvat I. Redesign of the GATE PET coincidence sorter. *Phys Med Biol* (2016) 61:N522–31. doi:10.1088/0031-9155/61/18/N522
- Etxebeste A, Dauvergne D, Fontana M, Létang JM, Llosá G, Munoz E, et al. CCMOD: a GATE module for Compton camera imaging simulation. *Phys Med Biol* (2020) 65:055004. doi:10.1088/1361-6560/ab6529
- Lee H, Cheon B-W, Feld JW, Grogg K, Perl J, Ramos-Méndez JA, et al. TOPAS-imaging: extensions to the TOPAS simulation toolkit for medical imaging systems. *Phys Med Biol* (2023) 68:084001. doi:10.1088/1361-6560/acc565
- Pommranz CM, Schmidt FP, Mannheim JG, Diebold SJ, Tenzer C, Santangelo A, et al. Design and performance simulation studies of a breast PET insert integrable into a clinical whole-body PET/MRI scanner. *Phys Med Biol* (2023) 68:055019. doi:10.1088/1361-6560/acba77
- Brun R, Rademakers F. ROOT — an object oriented data analysis framework. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (1997) 389:81–6. doi:10.1016/S0168-9002(97)00048-X
- Khalife M, Stute S, Wagadarikar A, Comtat C. Modeling the GE Signa PET-MR with Monte Carlo simulations using GATE. In: Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference; 10-17 November 2018; Sydney, Australia. San Diego, CA: IEEE (2016).
- Konstantinou G, Lecoq P, Benlloch JM, Gonzalez AJ. Metascintillators for ultrafast gamma detectors: a review of current state and future perspectives. *IEEE Trans Radiat Plasma Med Sci* (2022) 6:5–15. doi:10.1109/TRPMS.2021.3069624
- Bonifacio DAB, Latella R, Murata HM, Benlloch JM, Gonzalez A, Lecoq P, et al. Modeling brain-dedicated PET systems using metascintillators for optimized ToF capability (2023). doi:10.36227/techrxiv.23699694.v1
- Bonifacio DAB, Latella R, Murata HM, Benlloch JM, Gonzalez A, Lecoq P, et al. Analytical modeling of the sensitivity of cylindrical PET systems based on bulk materials and metascintillators (2023). doi:10.36227/techrxiv.22637731.v1
- NEMA. NU 4-2008. Performance measurements of small animal positron emission tomographs (2008).
- Sung C-H, Cappellugola L, Follin M, Curtioni S, Dupont M, Morel C, et al. Detailed simulation for the ClearMind prototype detection module and event reconstruction using artificial intelligence. *Nucl Instr Methods Phys Res Section A: Acc Spectrometers, Detectors Associated Equipment* (2023) 1053:168357. doi:10.1016/j.nima.2023.168357
- Sung C-H (2022). Simulation and artificial intelligence for a gamma detector for high resolution PET imaging. Phd thesis, Saclay, France; Université Paris-Saclay. Available at: <https://theses.hal.science/tel-03865521> (Accessed September 15, 2023).
- Lecoq P, Morel C, Prior JO, Visvikis D, Gundacker S, Auffray E, et al. Roadmap toward the 10 ps time-of-flight PET challenge. *Phys Med Biol* (2020) 65:21RM01. doi:10.1088/1361-6560/ab9500
- Razdevšek G, Dolenc R, Krizan P, Majewski S, Studen A, Korpar S, et al. Multipanel limited angle PET system with 50 ps FWHM coincidence time resolution: a simulation study. *IEEE Trans Radiat Plasma Med Sci* (2022) 6:721–30. doi:10.1109/TRPMS.2021.3115704

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.