White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/198834/

Version: Published Version

Article:
Abed Abud, A., Abi, B., Acciarri, R. et al. (2023) Highly-parallelized simulation of a pixelated LArTPC on a GPU. Journal of Instrumentation, 18 (04). P04034. ISSN: 1748-0221

https://doi.org/10.1088/1748-0221/18/04/p04034

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Journal of Instrumentation

PAPER • OPEN ACCESS

Highly-parallelized simulation of a pixelated LArTPC on a GPU

To cite this article: A. Abed Abud *et al* 2023 *JINST* **18** P04034

View the article online for updates and enhancements.

This content was downloaded from IP address 188.223.200.151 on 03/05/2023 at 14:10

Published by IOP Publishing for Sissa Medialab

Received: *December 21, 2022*
Accepted: *March 28, 2023*
Published: *April 26, 2023*

Highly-parallelized simulation of a pixelated LArTPC on a GPU

The DUNE Collaboration

E-mail: roberto@lbl.gov

Abstract: The rapid development of general-purpose computing on graphics processing units (GPGPU) is allowing the implementation of highly-parallelized Monte Carlo simulation chains for particle physics experiments. This technique is particularly suitable for the simulation of a pixelated charge readout for time projection chambers, given the large number of channels that this technology employs. Here we present the first implementation of a full microphysical simulator of a liquid argon time projection chamber (LArTPC) equipped with light readout and pixelated charge readout, developed for the DUNE Near Detector. The software is implemented with an end-to-end set of GPU-optimized algorithms. The algorithms have been written in Python and translated into CUDA kernels using Numba, a just-in-time compiler for a subset of Python and NumPy instructions. The GPU implementation achieves a speed up of four orders of magnitude compared with the equivalent CPU version. The simulation of the current induced on 10^3 pixels takes around 1 ms on the GPU, compared with approximately 10 s on the CPU. The results of the simulation are compared against data from a pixel-readout LArTPC prototype.

Keywords: Detector modelling and simulations II (electric fields, charge transport, multiplication and induction, pulse formation, electron emission, etc); Simulation methods and programs; Noble liquid detectors (scintillation, ionization, double-phase); Time projection Chambers (TPC)

ArXiv ePrint: 2212.09807

© 2023 CERN. Published by IOP Publishing Ltd on behalf of Sissa Medialab. Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

https://doi.org/10.1088/1748-0221/18/04/P04034

2023 JINST 18 P04034

Contents

1 Introduction **1**

2 Technical implementation **2**

3 Charge simulation **3**
- 3.1 Electron recombination 3
- 3.2 Electron transport in liquid argon 5
- 3.3 Electronic signal induction on a pixel 6
  - 3.3.1 Field response 6
  - 3.3.2 Induced current calculation 7
- 3.4 Electronics response 10

4 Light simulation **11**
- 4.1 Incident light calculation 11
- 4.2 Photocurrent simulation 13
- 4.3 Electronics response 15
- 4.4 Truth propagation 15

5 Profiling **17**

6 Simulation of a cosmic-ray sample and data comparison **18**

7 Conclusions **21**

The DUNE collaboration **25**

1 Introduction

The idea of using a liquid argon time projection chambers (LArTPC) for the detection of neutrino interactions was first proposed in 1977 [1]. The detection mechanism is the following: charged particles produced by neutrino interactions ionize the argon, leaving a trail of ionization electrons. In addition, liquid argon also produces scintillation light, which provides calorimetric information and a fast timing signal ($\mathcal{O}(10\,\mathrm{ns})$ [2]). A fraction of the ionized electrons recombine immediately with the positive argon ions, while the remaining ones drift towards the anode side of the detector in a homogeneous electric field applied to the argon volume, which is usually $\mathcal{O}(100\,\mathrm{V/cm})$. Impurities present in the LAr (e.g. O_2, H_2O, N_2) can attach a portion of the drifting electrons. The amount of drifting electrons declines as a function of the distance from the anode, since the electrons need to travel a longer path.

Typically, two or more arrays of sense wires are placed at the anode and assembled into planes. The drifting of negative charges in a constant electric field induces a signal on the wires. Each plane provides

2023 JINST 18 P04034

a two-dimensional image of the ionization: the position of the wire provides one dimension, and the time of the arrival provides the second one, since the drift velocity of the electrons in the LAr is known and is typically $O(1\,\mathrm{m/ms})$. Using multiple wire planes can help estimate the position of the ionization in three dimensions. However, ambiguities arise when drifting electrons are isochronous or parallel to a wire orientation. Unambiguous 3D imaging of LArTPC charge signals is possible using a readout system based on a pixelated array of charged-sensitive pads, which has been demonstrated in ref. [3]. Both the typical distance between adjacent wires and the typical pixel pitch are in the order of few millimeters. The position on the anode plane of the involved pads provides two spatial dimensions, and the time of the induced signal provides the third one. This truly three-dimensional readout provides better reconstruction efficiency and purity than the 2D combined wire readout, as demonstrated in ref. [4].

Pixel readout requires the channel count used to be increased by a factor of 10 to 100 with respect to wire planes. Thus, with this increased channel count and granularity simulation burden, the transport of electrons in LAr and the signal induction on the pixel pads represent an ideal use case for highly-parallelized, concurrent simulation algorithms. The development of general-purpose computing on graphics processing units (GPGPU), which went hand in hand with the advances in the machine learning and deep learning fields, has driven the design of the current generation of supercomputers, such as Perlmutter at NERSC [5], which is a heterogeneous system with both GPU-accelerated and CPU-only nodes. Implementing highly-parallelized simulation chains allows full advantage of these new systems to be taken and enables the simulation of future pixelated LArTPCs, which would otherwise not be viable with current resources.

The US-based neutrino physics program relies on present and future experiments using LArTPC technology. The flagship experiment is the Deep Underground Neutrino Experiment (DUNE), which will consist of a high-intensity accelerator neutrino beam, measured by near and far detectors [6]. The Far Detector will consist of four 17-kiloton LArTPC modules located deep underground at the Sanford Underground Research Facility in Lead, South Dakota, located 1285 km from the beam source [7]. The Near Detector will be located at Fermilab, 574 m from the beam source, and will contain a 67 t modular LArTPC called ND-LAr [8].

There are already some software toolkits for LArTPCs [9, 10] and there has been some effort towards parallelizing the reconstruction stage [11]. However, the simulation stages have remained mostly sequential and their speed up has been recognized as a priority by the community [12].

In this document we will describe the implementation of a set of highly-parallelized algorithms, organized in a module called `larnd-sim` [13], that run on GPUs. They simulate the ionized electrons recombination and drifting towards the anode, the generation of electronics signals on the pixelated readout, and the processing of the signal by the front-end electronics.

2 Technical implementation

Recent rapid developments in the field of machine learning have stimulated the creation of several toolkits for GPU-accelerated applications. In particular, the NVIDIA® CUDA platform [14] allows to use GPUs for general purpose computing via different programming languages. We opted for Numba [15], which generates CUDA computing kernels using a subset of native Python and NumPy code [16].

A CUDA kernel is a function that is executed N times in parallel by N CUDA threads. The threads can be organized in one-dimensional, two-dimensional, or three-dimensional *blocks*, which

2023 JINST 18 P04034

in turn can be organized in one-dimensional, two-dimensional, or three-dimensional *grids*. Blocks in the same grid contain the same number of threads, can run independently, and can be executed in any order, while threads in the same block can co-operate through shared memory. CUDA kernels typically store the result of the computation in a pre-allocated array passed to the kernel function. The CUDA programming model requires a careful design of the algorithm: the shape and size of the array where the result is stored must be known in advance and the threads must avoid race conditions during execution,[1] thus the result of the algorithm must not depend on the order of execution of the threads.

The software described in this document contains several CUDA kernel functions, separated into two logical categories: one for the charge simulation, described in section 3, and one for the light simulation, described in section 4. The functions simulate the detector response, including: (1) the recombination of the electrons with the argon ions, (2) the drifting of the electrons towards the anode, (3) the induction of electronic signals on the pixel pads and optical detectors, and (4) the electronics response of the charge and light readout systems.

The simulation of the passage of the initial particles through matter is performed using `edep-sim` [17], a wrapper around Geant4 [18], which is independent from the `larnd-sim` package described here. The output consists of a set of short particle track segments, in the order of few millimeters, which describe the energy deposition trail of each particle. The length of the segments depends on the derivative of the stopping power dE/dx: the portion of a particle trail where the dE/dx changes abruptly will be divided in finer segments than the portion where the dE/dx is mostly constant. Thus, in a single segment, the energy deposition per unit length is assumed to be constant. This approximation is valid when the segment length is of the same order as the detector resolution — at much shorter lengths, fluctuations from the long tail of the dE/dx distribution fall outside of the applicable region of recombination models, and at much longer lengths, correlations in the dE/dx fluctuations become significant and the dE/dx width will be under-simulated. Within these broad considerations, reducing the minimum size of the segments has not shown a significant impact on the result of the charge simulation. This set S_i is stored in a bi-dimensional NumPy array containing the energy deposition and the spatial distribution of the segments:

$$S_i = (\vec{r_s}, \vec{r_e}, E)_i \,, \tag{2.1}$$

where $\vec{r_s}$ and $\vec{r_e}$ are four-dimensional vectors containing the spatial and timing coordinates of the segment start and end points, and E is the deposited energy. This array is used as input for our module. In order to minimize the memory transfer between the host and the device (in our case the GPU), we allocate the NumPy array directly on the device memory using CuPy [19], a GPU array backend that implements a subset of the NumPy interface. The output of the `larnd-sim` simulation is then saved in a HDF5 file [20]. The entire simulation workflow is shown in figure 1.

3 Charge simulation

3.1 Electron recombination

The first step of our simulation is to calculate the number of electrons that remain after the recombination and start drifting towards the anode. We denote the initial charge ionized by the

[1] In software, a race condition can happen when the behavior of the program depends on the relative timing of multiple threads or processes.

2023 JINST 18 P04034

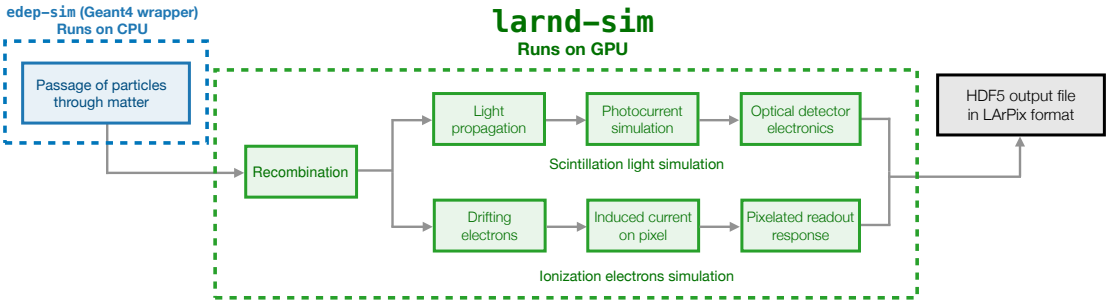


Figure 1. Diagram showing the full simulation workflow. The passage of the particle through matter is simulated by `edep-sim` on the CPU. The output is fed to `larnd-sim`, which runs entirely on GPU. Its output is finally saved in a HDF5 file.

particle as Q_0, while the charge remaining after the recombination is given by $Q_R = \mathscr{R} \cdot Q_0$, where $\mathscr{R}$ is our recombination factor.

Two different models are commonly used to describe this phenomenon: the Birks model [21], which gives spurious values when applied to high-ionization particles [22], and the modified Box model [22], which doesn't suffer from these issues but is inadequate to describe particles at low stopping power (low dE/dx).

The recombination factor for the Birks model $\mathscr{R}_{\text{Birks}}$ can be parametrized as:

$$\mathscr{R}_{\text{Birks}} = \frac{A_b}{1 + k_b/\epsilon \cdot dE/dx}, \tag{3.1}$$

where A_b and k_b are free parameters that usually depend on the detector and ϵ is the product of the electric field with the liquid argon density. The ICARUS collaboration obtained $A_b = 0.800$ and $k_b = 0.0486\,(\text{kV/MeV})\,(\text{g/cm}^3)$ [23], which are the values used in our simulation.

The recombination factor the modified Box model $\mathscr{R}_{\text{Box}}$ is defined as:

$$\mathscr{R}_{\text{Box}} = \frac{\log(\alpha + \beta/\epsilon \cdot dE/dx)}{\beta/\epsilon \cdot dE/dx}, \tag{3.2}$$

where α and β are free parameters which were measured by the ArgoNeuT collaboration to be $\alpha = 0.93$ and $\beta = 0.207\,(\text{kV/MeV})(\text{g/cm}^3)$ [22]. In both cases, the typical recombination factor for a minimum ionizing particle (MIP) is around 0.7. Our simulation assumes the Birks model by default.

The implementation of the calculation of the recombination factors $\mathscr{R}_{\text{Birks}}$ or $\mathscr{R}_{\text{Box}}$ on the GPU is trivial: the i-th thread of the $K_{\text{recomb}}(S_i, \mathscr{R})$ CUDA kernel takes as input the i-th row of the NumPy array containing the segments S and applies the recombination formula, so eq. (3.1) or eq. (3.2). The result is stored in an appropriate column of the NumPy array. Currently, the simulation treats the recombination factor as a fixed number: although this is an approximation, the level of fluctuations associated with it is significantly lower than the expected intrinsic noise of the detector, which was measured with the prototype described in section 6.

Even if this operation is computationally inexpensive, it's instructive to take a look at the performance comparison between a sequential, interpreted Python `for` loop, a loop compiled on the CPU using Numba, and the GPU implementation using a CUDA kernel. Figure 2 shows the processing time needed to calculate $\mathscr{R}_{\text{Birks}}$ of eq. (3.1) as a function of the number of GEANT4

2023 JINST 18 P04034

segments given as input. The CPU-compiled version is obviously faster than the sequential interpreted loop since the function is now translated into machine code. While the CUDA kernel processing time is initially the largest, it doesn't immediately scale with the number of input segments, so it starts being the exponentially faster implementation with more than $\mathcal{O}(10^4)$ segments, where it starts taking advantage of the massive parallelization achievable by the GPU. The NVIDIA® Tesla® V100 GPU used for this study can run more than 10^5 parallel threads. To give a figure of merit, a typical neutrino beam spill in the ND-LAr corresponds on average to $\mathcal{O}(10^5)$ segments.

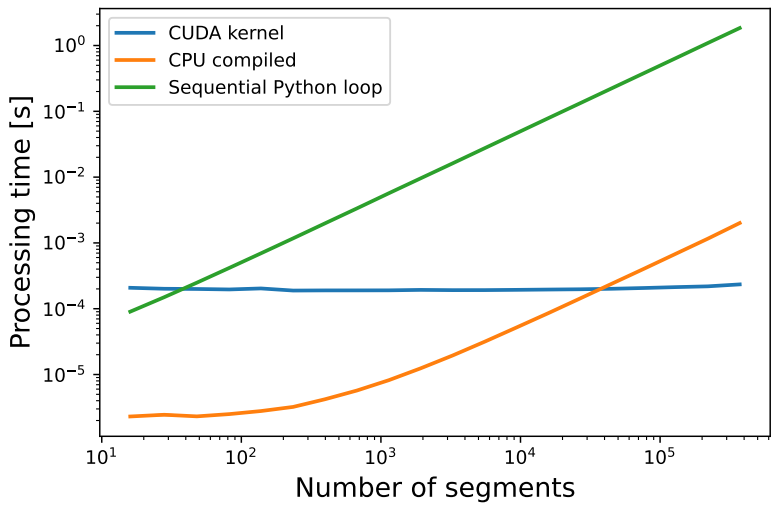


Figure 2. Processing time for the calculation of the recombination factor $\mathscr{R}_{\text{Birks}}$ of eq. (3.1) with the GPU implementation using a CUDA kernel (blue), the CPU implementation using Numba (orange), and a sequential Python `for` loop (green). For reference, a neutrino beam spill corresponds on average to $\mathcal{O}(10^5)$ energy deposition segments in the ND-LAr. This computation was performed on a node of the NERSC Cori supercomputer, which contains two sockets of 20-core Intel Xeon Gold 6148 (Skylake) at 2.40 GHz and 8 NVIDIA® Tesla® V100 (Volta) GPUs.

3.2 Electron transport in liquid argon

The electrons remaining after the recombination travel towards the anode at a constant velocity v_{drift} (assuming perfect uniformity of the electric field), which is typically $\mathcal{O}(1\ \text{mm}/\mu\text{s})$. The time they take to reach the anode is given by:

$$t_{\text{drift}} = (z - z_{\text{anode}})/v_{\text{drift}}, \tag{3.3}$$

where the electron drift direction is assumed to be along the z axis and where z_{anode} is the z coordinate of the anode.

The impurities in the liquid argon, such as O_2, N_2 and H_2O, can attach a portion of the drifting electrons, so electrons farther from the anode will have a higher chance to be attached. This effect is usually parametrized by a negative exponential, so the charge Q_a that effectively reaches the anode, assuming uniform impurities and a perfect electronics response, is:

$$Q_a = Q_R \cdot \exp(-t_{\text{drift}}/\tau), \tag{3.4}$$

2023 JINST 18 P04034

where t_{drift} is the time the charge takes to reach the anode and τ is a parameter that depends on the concentration of impurities and it is usually called *electron lifetime*, which is in the order of milliseconds for concentrations of O_2 at tens of parts per trillion.

Electrons drifting in strong electric fields do not diffuse isotropically, so it is necessary to estimate both the longitudinal and transverse components with respect to the drift direction [24]. The diffusion length is given by:

$$\sigma = \sqrt{2Dt_{\text{drift}}}, \tag{3.5}$$

where D is the longitudinal or transverse diffusion coefficient, which depends on the electric field and liquid argon temperature. In our simulation we set the longitudinal and transverse diffusion coefficients to $D_l = 4\,\text{cm}^2/\text{s}$ and $D_t = 8.8\,\text{cm}^2/\text{s}$, respectively. These values were obtained by a preliminary ProtoDUNE-SP [25] analysis.

This information is calculated and stored in appropriate columns of the NumPy array in a way analogous to the one described in section 3.1, where each thread processes a single segment independently.

3.3 Electronic signal induction on a pixel

3.3.1 Field response

The current induced by a point charge on a given pixel within the anode is calculated using the Shockley-Ramo theorem [26]:

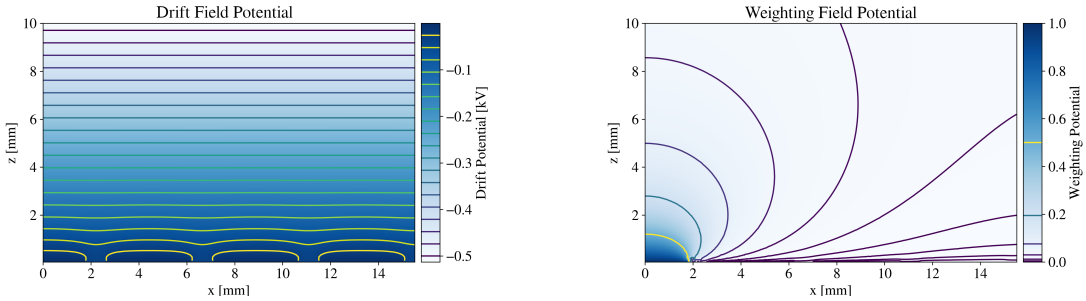
$$I_{\text{pixel}} = q\,\vec{v} \cdot \nabla W, \tag{3.6}$$

where W is the *weighting field potential*, the normalized contribution of a single electrode to the overall field, q is the particle charge, and $\vec{v}$ is the velocity of the particle. Within `larnd-sim`, the induced current is pre-calculated for a point-like charge and referenced as a $I_{\text{pixel}}[t, x, y]$ look-up table (LUT). The table contains the current induced by an electron placed at discrete (x, y) locations on the anode plane at a discrete time t, where $t = 0$ is the time when the electron is at $z = 0.5\,\text{cm}$. This value is chosen because of the observed flatness of the electric and weighting fields at this distance.

The $I_{\text{pixel}}[t, x, y]$ values are calculated as follows. In the region very close to the anode, the electric field and the weighting field are calculated numerically. The geometry of a small volume, including a central pixel with a pitch of 4.4 mm and its 8 nearest neighbor pixels, is modeled using CAD software and converted into a 3-dimensional mesh using the Gmsh package [27]. The fields are then calculated using the successive over-relaxation method as implemented in the Elmer FEM [28] software package. The electric and weighting fields share the same geometry, but differ in the boundary conditions imposed on the problem. In the electric field calculation, the pixels are grounded, with the backing plane at a small offset voltage, and the field on the cathode-facing side of the volume is set to the nominal field of 500 V/cm. In the calculation of the weighting field, all electrodes other than the central pixel are grounded, the central pixel is set to unit voltage, and the voltage on the remaining conductors is set to zero. The results of these two field calculations are shown in figure 3.

Next, the idealized drift paths (ignoring diffusion and attenuation effects) are integrated using the ICARUS and Walkowiak [29, 30] electron transport models evaluated within the calculated electric field. This model allows for the drift velocity to change as a function of the local electric

2023 JINST 18 P04034



(a) Drift electric potential field in the region near the pixel plane. Shown is a slice in y along on a pixel center. The pixel surfaces are set to 0 kV, while the gradient of the far field at the surface of $z = 10$ mm is set to 0.5 kV/cm. The color scale corresponds to the value of the drift potential.

(b) Weighting potential field in the region near a "pixel of interest". The central ($x = y = 0$ pixel is set to a unit potential, while all other surfaces are set to 0 potential. The resulting unitless field defines the susceptibility to current induction by charges moving nearby.

Figure 3. The drift (left) and weighting (right) field potentials obtained by finite-element analysis in the region near a generic pixel. Shown are slices of the potential fields along an $x - z$ plane which crosses through the centerline of a pixel.

field, as the assumption of a perfectly uniform field is not necessarily true very close to the anode. The drift paths are calculated for a grid of four hundred (x, y) positions, 0.5 cm from the anode on the z axis. The resulting granularity on the x and y axes is 0.33 mm. Finally, we compute the time-derivative of the weighting field potential along each path, which yields a charge-normalized current series for a given initial charge position. The sampling in time is 0.1 µs. A select few drift paths are shown in figure 4(a), with their corresponding current series in figure 4(b).

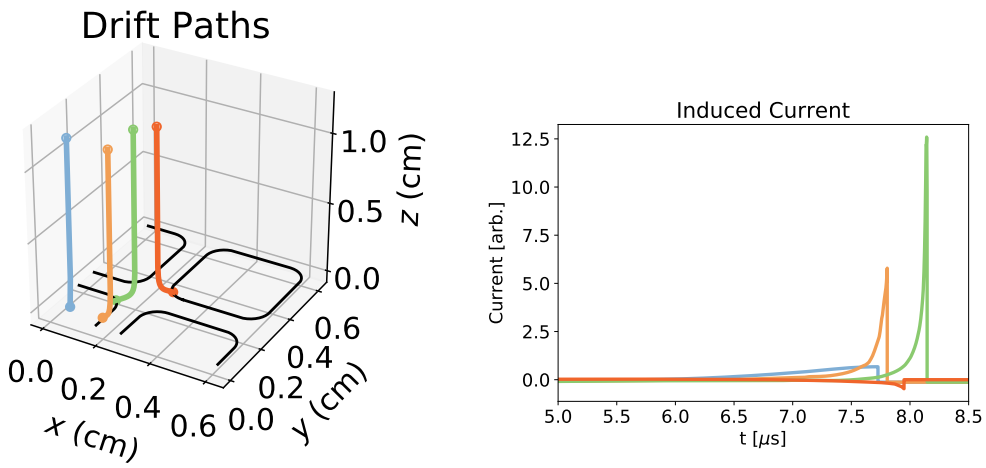
Farther from the pixel at $z > 0.5$ cm, the weighting potential is solved by treating the pixel as point-like and using a method-of-image-charges approximation to fix the boundary conditions at the anode and the cathode. The drift velocity is assumed to be uniform. The relative normalization to the near-field calculation is then fixed by enforcing continuity in the current series at the time boundary. Some re-scaling is performed on these current series to ensure they integrate to one electron charge, as error is introduced in the drift path integration, as well as numerical error present in the FEM solutions and their interpolated values. Figure 5 shows the tabulated induced current on a pixel up to 30 µs from the time of arrival of the drift electron. The agreement of the near-field FEM model and the dipole far-field approximation across this surface seen in figure 5 demonstrates that the transition surface at 0.5 cm is sufficiently far from the pixel plane.

3.3.2 Induced current calculation

The electrons drifting towards the anode are diffused in the longitudinal and transverse directions, as described in section 3.2. The ionization electrons corresponding to a Geant4 segment will then form a three-dimensional charge cloud. The induced current on a given pixel from this can be calculated by taking the convolution of the pre-calculated pixel response model (see section 3.3.1) and the charge density of the track segment:

$$I(t) = \int I_{\text{pixel}}(t - z/v_d, x, y)\rho(x, y, z)dxdydz \tag{3.7}$$

2023 JINST 18 P04034



(a) Example drift paths for electrons starting at a short distance from a pixel centered at $(0, 0)$.

(b) Induced current on the pixel at $(0, 0)$ from example drift paths

Figure 4. To determine the near-field pixel response, we simulate the paths and velocities of drift electron within a 3D FEM field model. Here we show the simulated electron drift paths (left), and the corresponding induced current on the pixel (right) for a subset of the starting points used. The color of the trajectory on the left corresponds to the color of the current induced curve on the right.

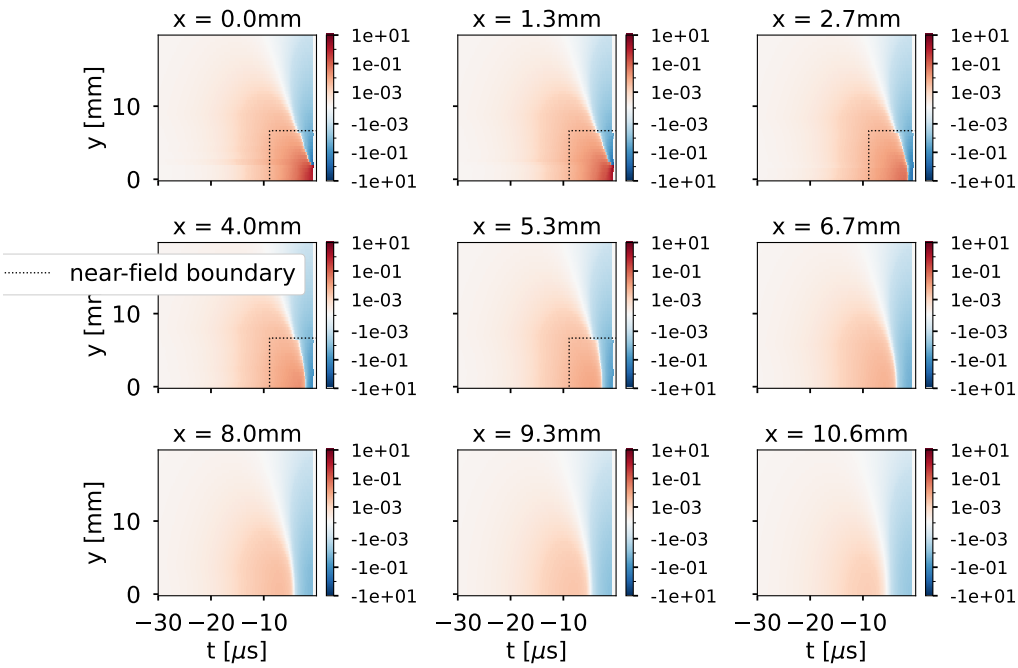


Figure 5. Induced current on a 4.4 mm pixel from a drifting electron at a given drift time and x, y position relative to the pixel, in units of electrons per 0.1 μs. The different calculations used for the near-field and far-field regions are described in the text in section 3.3.1 with the near-field boundary highlighted with a dotted line.

2023 JINST 18 P04034

where $\rho(x, y, z)$ is the 3-dimensional charge density including diffusion and $I_{\text{pixel}}(t - z/v_d, x, y)$ is the current response model at the given position and time tick, which is stored in the LUT described in section 3.3.1. The positively charged ions also produce a current on the pixel, however the magnitude of this current is more than 7 orders of magnitude smaller than the signal from the drift electrons due to the small drift velocity of the ions, and so it is neglected in the detector simulation.

The calculation of $I(t)$ is the most computational-intensive step of the detector simulation. In order to reduce its complexity, we apply two approximations to the charge density: first, we discretize the track segment along the track length into N points; and second, we approximate the effect of diffusion summing the contribution from M random 3D perturbations of the sample point. With these approximations, we can write the charge density as

$$\rho(x, y, z) \approx \frac{Q_a}{NM} \sum_{\mu=1}^{M} \sum_{\nu=1}^{N} \delta(x_\mu + \delta x_{\mu\nu} - x)\delta(y_\mu + \delta y_{\mu\nu} - y)\delta(z_\mu + \delta z_{\mu\nu} - z) \tag{3.8}$$

where Q_a is the total charge magnitude of the segment from eq. (3.4), $\delta(x)$ is the Dirac delta function, N is the number of sampled points along the track segment, M is the number of Monte Carlo iterations per sample point, (x_μ, y_μ, z_μ) is a point uniformly distributed along the track segment, and $(\delta x_{\mu\nu}, \delta y_{\mu\nu}, \delta z_{\mu\nu})$ are normally distributed in proportion to the diffusion in that dimension. This simplifies the calculation to a sum over the pixel response model at each sampled position

$$I(t) \approx \frac{Q_a}{NM} \sum_{\mu=1}^{N} \sum_{\nu=1}^{M} I_{\text{pixel}} \left(t - (z_\mu + \delta z_{\mu\nu})/v_d, x_\mu + \delta x_{\mu\nu}, y_\mu + \delta y_{\mu\nu}\right). \tag{3.9}$$

In our implementation, N is calculated for each track with a 10 µm sampling interval, with the option of increasing the resolution and the Monte Carlo sampling for higher precision. For reference, at this sampling interval each sample point contributes a charge of roughly 50 electrons for a MIP-like track, and the error introduced by the approximations is at the percent level and much smaller than the noise level.

The calculation of the induced current is particularly well-suited to be implemented on a highly-parallelized architecture, since it is necessary to perform the sum of eq. (3.9) for each time tick in the time window, each pixel, and each Geant4 detector segment. The length of the time window depends on the orientation of the segment with respect to the anode, since the arrival time is directly proportional to the z coordinate. In order to minimize the memory usage, the length of the longest signal and the largest number of pixels per segment are calculated before execution and used to allocate the array where the induced signals are going to be stored.

In our CUDA kernel $K_{\text{current}}(s_i, p_j, t_k)$, the parallel threads are organized in a three-dimensional grid, where the dimensions are the segment index s_i, the pixel index p_j, and the time tick t_k.

It is possible to compare the performances of this GPU algorithm with an equivalent CPU version, compiled with explicit parallelization of the loops using the Numba `prange` function. This function runs the loop in parallel threads, one for each core (20 in our case), and tries to merge adjacent loops together, reducing loop overhead [31]. The GPU version provides a speed-up of around four orders of magnitude for $O(10^3)$ simulated pixels, as shown in figure 6 for different values of the time sampling. Interestingly, in this case the GPU version is significantly faster than the CPU one even for only 10 pixels, since we are parallelizing also in two other dimensions (segment index and time). The processing time for the GPU implementation does not increase with the number of simulated pixels

2023 JINST 18 P04034

and with different values of the time sampling, which shows that the calculation is effectively being performed in parallel. Compilation times and memory allocation times are not taken into account here.

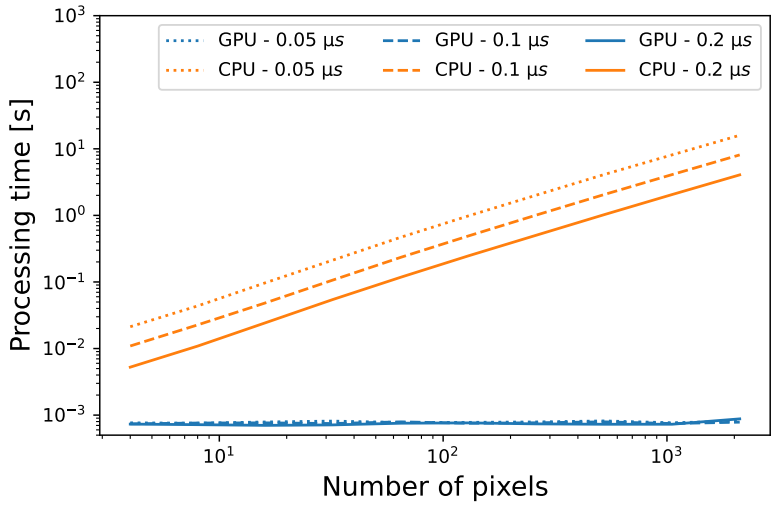


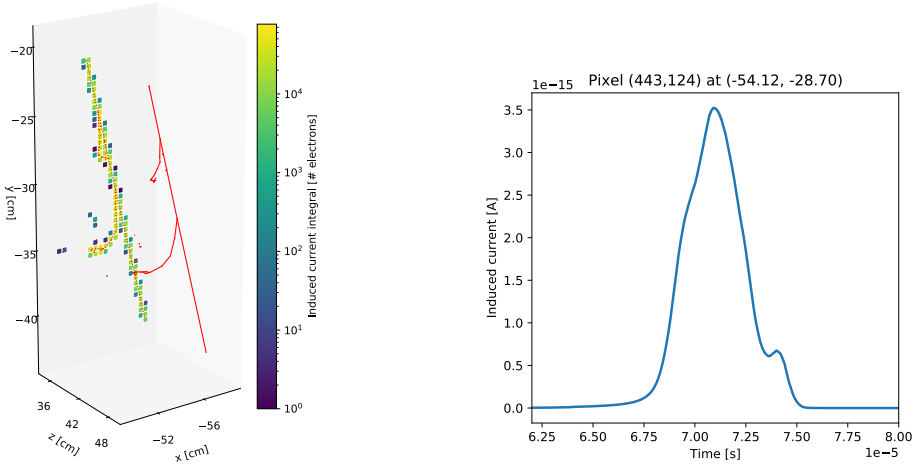
Figure 6. Processing time for the calculation of the induced current with the GPU implementation using a CUDA kernel (blue) and with a CPU implementation compiled with Numba with explicit parallelization of the loops (orange). For reference, a neutrino beam spill will induce a signal on around 5×10^4 pixels of the ND-LAr. The calculation has been performed with different values of the time sampling (0.05 µs, 0.1 µs, 0.2 µs) on a node of the NERSC Cori supercomputer, which contains two sockets of 20-core Intel Xeon Gold 6148 (Skylake) at 2.40 GHz and 8 NVIDIA® Tesla® V100 (Volta) GPUs.

The result of the simulation of a cosmic-ray muon interacting in the TPC active volume is shown in figure 7(a). Given the effect of transverse diffusion, signals are induced also on the pixels that do not lie exactly on the xy projection of the muon trajectory. The high granularity of the pixel layout shows clear imaging of features such as delta rays. Figure 7(b) shows the current induced on a pixel both by the cosmic-ray muon and by the subsequent delta ray.

3.4 Electronics response

In the LArPix system described in ref. [3] the pixel pads are uniquely instrumented by application-specific integrated circuits (ASICs), which provide a charge-sensitive amplifier (CSA) with self-triggered digitization. First, the signal from each pad is input to the CSA. Then, as signal accumulates on the pad, the voltage at the output of the CSA grows until it exceeds the discriminator threshold, which in our simulation is set to 28 mV, equivalent to the signal induced by 7×10^3 electron charges. The discriminator, in turn, triggers the output digitization through an 8-bit analog-to-digital converter (ADC). The ADC stores the CSA output voltage and then the CSA is reset, discarding the collected charge. The CSA is then ready to collect the subsequent signal. The minimum time between two consecutive ADC counts is 11 clock cycles, which with our 10 MHz clock correspond to 1.1 µs. The time between the threshold crossing and the signal digitization is tunable in the LArPix ASIC. In practice, this is 1.6 µs, which is tuned to the typical signal size. While leakage current is negligible, sub-threshold charge could collect on the channel introducing a bias in the charge measurement. A periodic reset limits the impact of this effect.

2023 JINST 18 P04034



(a) 3D event display of a simulated cosmic-ray muon.

(b) Current induced on a single pixel by the cosmic-ray muon and a delta ray.

Figure 7. The simulation of the induced current performs the calculation of eq. (3.9) for each active pixel. Here we show a 3D event display of a simulated cosmic-ray muon (left, in red) and the current induced on a pixel by the cosmic-ray muon and a delta ray (right).

Three kinds of noise are included in the simulation: (1) a reset noise, which corresponds to a random pedestal shift after every trigger reset, (2) a discriminator noise which affects the discriminator threshold, and (3) an uncorrelated noise. These are set in the simulation 900, 650, and 500 electron charges, respectively. These values have been tuned from the data described in section 6.

In our implementation, after summing the current induced by different tracks on the same pixel p_i, a CUDA kernel $K_{\text{electronics}}(p_i)$ simulates the trigger logic and the noise sources described above. The threads are organized in a one-dimensional grid, where each thread processes in parallel the current induced on a single pixel p_i. The CUDA kernel is exponentially faster than the CPU implementation and is more than three orders of magnitude faster with $O(10^3)$ pixels, as shown in figure 8.

Figure 9 shows a comparison of the integral of the induced current on each pixel and the corresponding ADC counts for a cosmic-ray muon. When the integral signal on a certain pixel does not reach the discrimination threshold no ADC count is stored, so it is possible to have pixels with a non-null induced current but no ADC count.

The resulting ADC counts and their corresponding timestamps are stored in a bi-dimensional array, where one dimension is the pixel index and the other is the ADC count index. They are then saved in the same HDF5 format used by the LArPix readout system, facilitating comparison between data and simulation.

4 Light simulation

4.1 Incident light calculation

The scintillation light produced by the excitation of the liquid argon provides additional calorimetric and topological information and a fast timing signal. Within `larnd-sim`, we include GPU algorithms to model scintillation light production, timing, and the associated electronics response.

2023 JINST 18 P04034

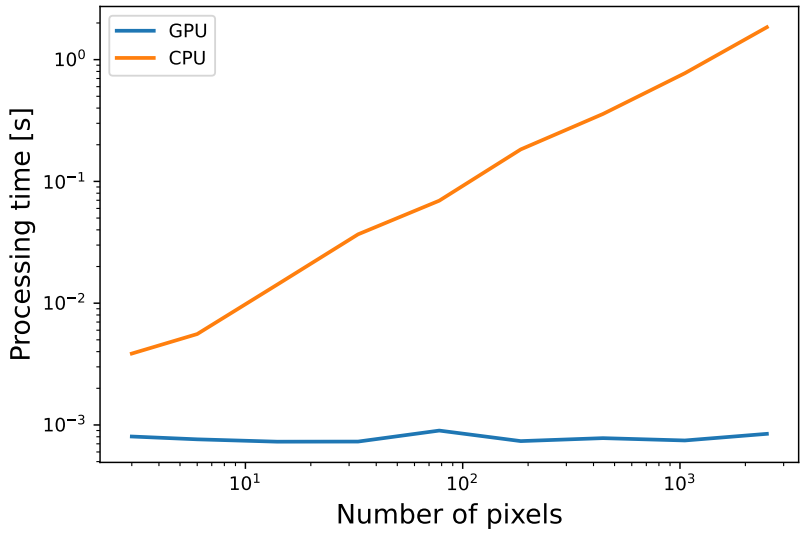


Figure 8. Processing time for the simulation of the electronics response with the GPU implementation using a CUDA kernel (blue) and with a CPU implementation compiled with Numba with explicit parallelization of the loops (orange). For reference, a neutrino beam spill will induce a signal on around 5×10^4 pixels of the ND-LAr. The simulation has been performed on a node of the NERSC Cori supercomputer, which contains two sockets of 20-core Intel Xeon Gold 6148 (Skylake) at 2.40 GHz and 8 NVIDIA® Tesla® V100 (Volta) GPUs.

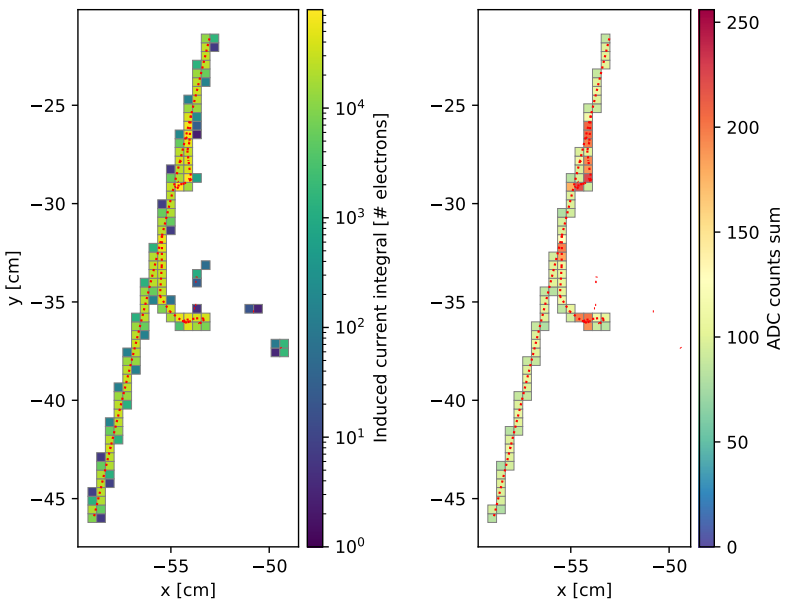


Figure 9. Bi-dimensional projection of a cosmic-ray muon interacting in the detector with the integral of the current induced on each pixel (left) and the sum of the ADC counts after the digitization stage (right).

2023 JINST 18 P04034

Light intensity is calculated using a self-consistent recombination model. Namely, each track segment dE/dx is converted to a total number of emitted photons N_γ using the same recombination parameterization used for the dQ/dx:

$$\frac{dN_\gamma}{dx} = \left(W_{\text{ph}}^{-1} - \mathscr{R} W_{\text{ion}}^{-1}\right) \frac{dE}{dx} \tag{4.1}$$

where W_{ph} is the photon production per deposited energy at zero field, and $\mathscr{R}$ is from one of the recombination models described in section 3.2.

Light propagation and material quantum efficiencies are pre-tabulated using a dedicated GEANT4 simulation into a LUT containing an average visibility $V_i\left[i_x, i_y, i_z\right]$ and relative time distribution $f_i\left[i_x, i_y, i_z, i_t\right]$ for each 3D voxel $\left(i_x, i_y, i_z\right)$ and each photosensor i in the active volume [32]. For each track segment and photosensor, the total number of photoelectrons (p.e.) is calculated as

$$N_{\text{pe},ij} = \epsilon_i V_i\left[i_x, i_y, i_z\right] \left(\frac{dN_\gamma}{dx}\right)_j dx_j \tag{4.2}$$

where ϵ_i is the overall detection efficiency of photosensor i and dx_j is the length of the track segment j. To implement this, we use a two-dimensional grid across the track segment index and detector index, such that each thread calculates the number of photoelectrons observed on a light detector from its respective track segment. In testing, we observe a speed-up factor of about 56× by using the CUDA kernel for this task.

4.2 Photocurrent simulation

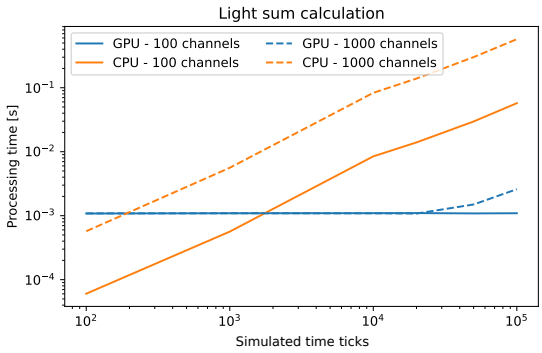
Once the total light signal is determined for each track segment, they are summed into a photocurrent time profile on each photosensor using the time distribution stored within the LUT:

$$I_{\text{pe},i}\left[i_t\right] = \frac{\Delta t_{\text{LUT}}}{\delta t} \sum_j f_i\left[i_x, i_y, i_z, i_t - t_0/\Delta t_{\text{LUT}}\right] N_{\text{pe},ij}, \tag{4.3}$$

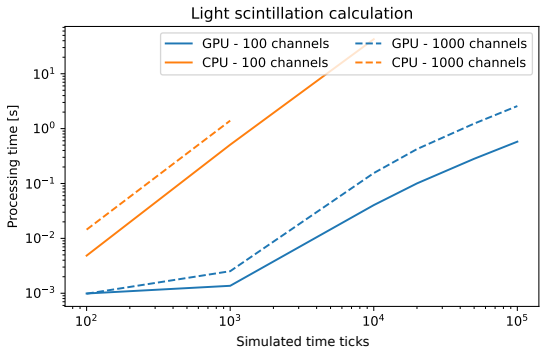
where t_0 is the segment deposition time, δt is the simulation time tick, and Δt_{LUT} is the time bin size of the LUT. The implementation of this calculation first determines the total number of ticks needed to fully simulate the event, with a configurable time pre- and post-buffer, and then pre-allocates the photocurrent time profile $I_{\text{pe},i}[i_i]$ array on the GPU. A CUDA kernel then iterates over the incident light array $N_{\text{pe},i}$ summing the contribution at each time tick. To maintain independence between threads, we use a two-dimensional grid on the dimensions of the output time profile array. Each thread is responsible for calculating the sum over all track segments for a single photodetector and time tick. We observe a factor of 45 improvement by using the GPU kernel for this calculation at scale. However below approximately 1000 channels × 1000 ticks the parallel CPU calculation becomes faster due to the copying of static configuration data to the GPU, see figure 10(a).

After the time profile has been created, the simulation calculates the smearing effect due to the scintillation light emission. LAr scintillation light consists of two components: a prompt component with a characteristic decay time of ~ 7 ns; and a slow component produced with a characteristic decay time of ~ 1600 ns [33, 34]. These values, which are assumed to be constant in our simulation, depend on the concentration of non-argon impurities in the liquid [35, 36] and on the electric

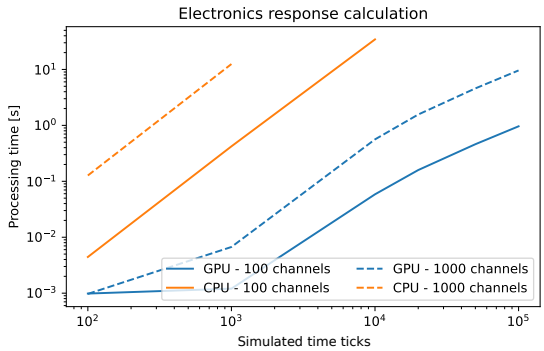
2023 JINST 18 P04034



(a) Comparison of light time profile calculation



(b) Comparison of scintillation profile convolution



(c) Comparison of electronics response convolution

Figure 10. A summary of the runtime of the CUDA-compiled light simulation kernels described in the text using 100 identical track segments in the center of a TPC. The three most computationally intensive steps of the light simulation are compared to an equivalent parallelized but CPU-based calculation. For the light scintillation calculation and the electronics response calculation, the run time does not increase with more track segments. For reference, the drift time window of ND-LAr will be around 3×10^3 time ticks.

field [37]. A two-component exponential model for the scintillation time profile

$$f_{\text{scint}}(t) = \frac{f_p}{\tau_p} e^{-t/\tau_p} + \frac{1 - f_p}{\tau_s} e^{-t/\tau_s} \tag{4.4}$$

is used to simulate the broadening of the light signals due to these processes, where f_p is the fraction of total light arising from the prompt component (~0.3 [2]), and τ_p and τ_s are the prompt and slow decay times, respectively.

The convolution is direct and truncated:

$$I_{\text{scint},i}[i_t] = \sum_{j_t=j_{t,\text{min}}}^{i_t} I_{\text{pe},i}[j_t] * f_{\text{scint}}((i_t - j_t)\delta t), \tag{4.5}$$

where $j_{t,\text{min}} = i_t - 5\frac{\tau_t}{\delta t}$. The convolution kernel is truncated at 5× the scintillation slow component lifetime. Within this interval, > 99% of the total light from an energy deposit will have been emitted, thus minimally impacting the accuracy of the simulation. However, by using a truncated convolution, we improve the scaling of the simulation from $O(N^2)$ to $O(N \times N_{\text{trunc}})$, where N_{trunc} is

2023 JINST 18 P04034

the number of truncated time ticks. This convolution is implemented in a CUDA kernel and threads are distributed in a two-dimensional grid across the light detector index i and the time tick index i_t, such that each thread calculates the weighted sum across the input array $I_{\mathrm{pe},i}$. The direct convolution of the scintillation light profile along with the electronics response convolution, discussed later, are the most computationally intense stages of the light simulation. Figure 10(b) and figure 10(c) show the performance of these kernels with the number of time ticks simulated and the number of light channels. Within the event sizes tested, regardless of the number of time ticks and light channels, a significant speed-up is realized by moving this calculation to the GPU.

Fluctuations in the number of photoelectrons produced by the incident light due to counting statistics is included for each time profile bin. An inverse transform sampling Poisson random number algorithm [38] is used to generate the number of observed photoelectrons per time tick. The average number of steps required by this algorithm scales in proportion to the mean of the distribution. To improve the runtime for large signals, at greater than 30 expected photoelectrons, a normal distribution $\sim \mathcal{N}(I_{\mathrm{scint},i}[i_t]\delta t, \sqrt{I_{\mathrm{scint},i}[i_t]\delta t})$ is used. The choice of 30 photoelectrons is selected to limit the error from this approximation to less than 1%.

4.3 Electronics response

The electronics response of the light detector readout is simulated assuming a perfectly linear ADC, a user-specified unit-normalized impulse response model, and purely uncorrelated noise:

$$V_i[i_t] = n_i[i_t] + \sum_{j_t=0}^{N_{\mathrm{resp}}} G_i I_{\mathrm{scint},i}[j_t] i_{\mathrm{resp}}[i_t - j_t] \frac{\Delta t_{\mathrm{resp}}}{\delta t} \tag{4.6}$$

where $n_i[i_t]$ is the simulated noise, $i_{\mathrm{resp}}[i_t]$ is the unit-normalized response model, N_{resp} is the number of bins in the response model, and G_i is the individual photosensor gain with units of [ADC · time / PE]. Noise is generated individually on each light detector by using a user-specified fast-fourier transform of the noise distribution to generate random phase sinusoids with appropriate amplitudes. Figure 11 demonstrates the simulation sequence on an example muon decay event with non-trivial time structure.

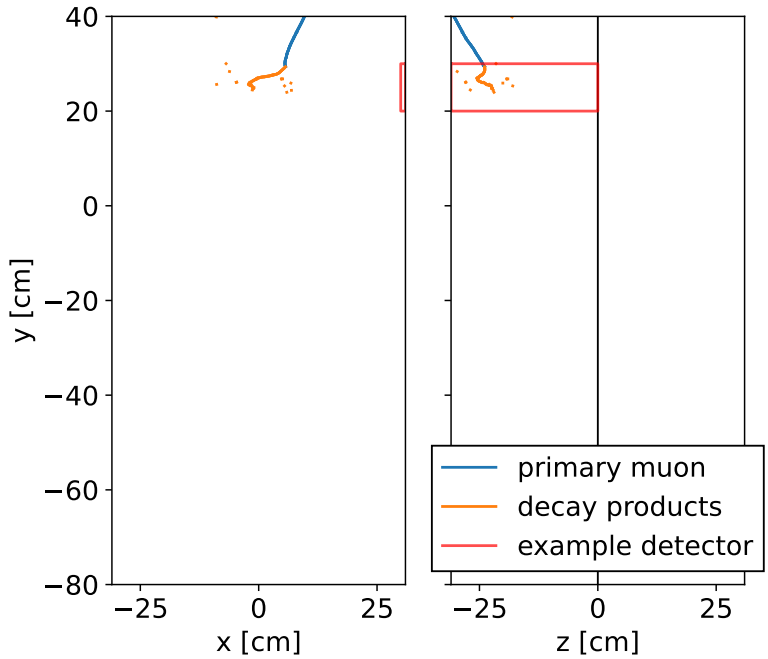
Each time profile is segmented into individual light triggers using a direct threshold on the sum of multiple light detectors. A serial algorithm loops over each light detector group and identifies each threshold crossing, accounting for dead-time between triggers. Because the timing of later triggers depends on the timing of earlier triggers, this algorithm cannot be parallelized. However, the trigger simulation only contributes $\sim 3\%$ to the overall simulation runtime and scales with the number of time ticks simulated, thus the use of a serial algorithm for the trigger generation is not a significant contributor to the overall simulation time at the DUNE Near Detector regime of < 100 triggers per spill per light detector group.

The final ADC waveform is then linearly interpolated into the desired sampling frequency and ADC resolution. Trigger information and waveforms are saved into dedicated datasets within the same output HDF5 file as the charge simulation.

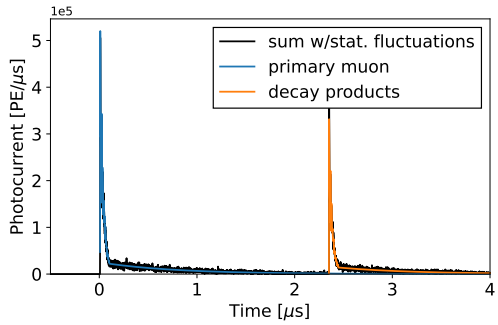
4.4 Truth propagation

True particle energy deposition information can be propagated through this chain to provide the true track segment id and incident photoelectrons contributing to each ADC sample. However for

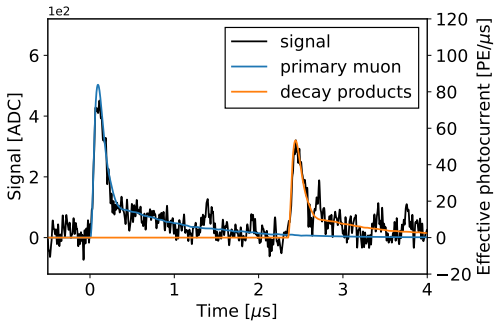
2023 JINST 18 P04034



(a) 2D projection of a simulated stopping muon and subsequent decay.



(b) Photocurrent on example photosensor after including LUT and scintillation time smearing.



(c) Photosensor output and effective photocurrent after applying response and noise models.

Figure 11. Example simulated muon decay and subsequent light simulation sequence. The region highlighted in figure 11(a) corresponds to the location of the light detector simulated in figure 11(b) and 11(c).

typical light detectors with $\mathcal{O}(\text{cm})$ spatial resolution and relatively broad response times $\mathcal{O}(\mu\text{s})$, most energy deposits originating from an interaction overlap. Thus, the number of truth entries needed scales as $(dN_{\text{segments,typ}}/dV) \times (N_{\text{samples}}) \times (N_{\text{photosensors}}) \times (\text{visible volume})$. Using typical numbers of $\mathcal{O}(500/\text{event m}^3)$, $\mathcal{O}(100)$, $\mathcal{O}(100)$, and $\mathcal{O}(10\,\text{m}^3)$, respectively, tracking the true track segment id and intensity at each sample necessitates $\mathcal{O}(5 \times 10^7)$ truth entries, using a minimum of $\mathcal{O}(400\,\text{MB/event})$ assuming each record consists of a 32-bit integer and 32-bit floating point number. Since the truth information for an event is highly correlated in time, we opt to by default only preserve the true number of photoelectrons and arrival time per track segment per light detector, reducing

2023 JINST 18 P04034

the required space by a factor of N_{samples} and runtime by a factor of N_{segments}/tick. We however maintain the option to track the full truth information as needed by setting a configuration flag.

5 Profiling

The `larnd-sim` code has been profiled using the NVIDIA Nsight™ Systems and the NVIDIA Nsight™ Compute tools v2021.3.0 with an input dataset of 250 simulated cosmic rays. The analysis shows that the algorithm is well optimized for GPUs. The charge simulation is responsible for around 70% of the processing time. Of this 70%, 97.5% of the time is used by the induced current CUDA kernel described in section 3.3.2. This kernel, in turn, spends 99.9% of the processing time for computing operations. The time spent for memory transfers, including also the initial time to allocate the dataset on the GPU memory, can be considered negligible.

A Roofline [39] analysis performed on the induced current CUDA kernel proves that the algorithm is compute-bound, so the time needed to complete its task is determined principally by the computing speed. Figure 12 shows that kernel performance for 10 cosmic-ray events with a total of 5013 active pixels, which is a typical amount of data processed for each GPU iteration, is close to the peak FLOP/s limit (in orange) and far from the memory bandwidth limit (in blue) of the GPU being used.

Overall, for a 0.5 m^3 LArTPC operated with a surface-level cosmic ray rate, the inclusion of the light simulation increases the runtime, the memory usage, and the overall data volume of the output by 30%, 15%, and 20%, respectively.

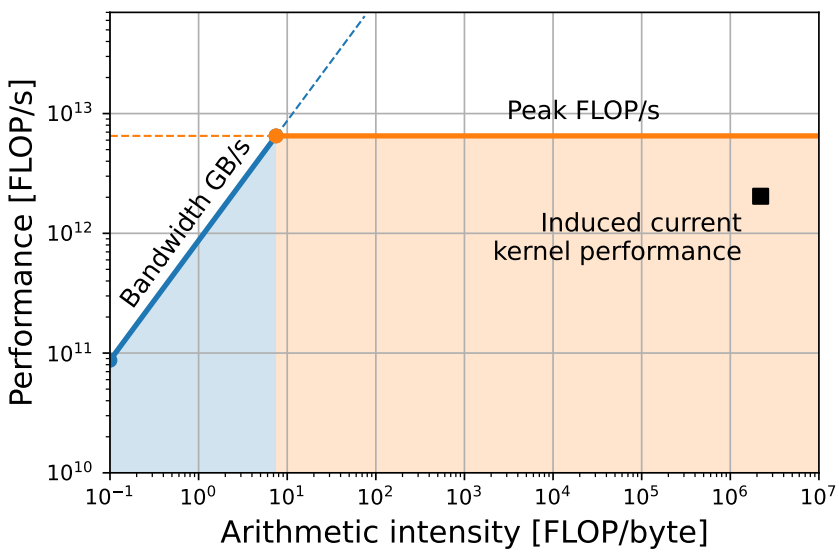


Figure 12. Roofline plot for the induced current CUDA kernel (black square) on a GPU node of the NERSC Cori supercomputer, equipped with 8 NVIDIA® Tesla® V100 (Volta) GPUs. The kernel performances are far from the memory-bound region (in light blue) and close to the peak FLOP/s limit (in orange) of the GPU being used.

2023 JINST 18 P04034

6 Simulation of a cosmic-ray sample and data comparison

A single-phase LArTPC called the *ArgonCube Module-0 Demonstrator* was operated in spring 2021 at the University of Bern as a prototype for the ND-LAr. The detector is divided into two functionally identical TPC drift regions, sharing a central high-voltage cathode that provides the drift electric field. Each anode is equipped with an array of 2×4 LArPix tiles of size $31.038\,\mathrm{cm}^2 \times 31.038\,\mathrm{cm}^2$. A total of 4900 pixel pads of size $0.4434\,\mathrm{cm}^2 \times 0.4434\,\mathrm{cm}^2$ are etched on the side of each tile facing the active volume. Each tile is instrumented by 100 ASICs placed on the opposite side of the board. The LArTPC drift length is 30.27 cm.

A dataset of cosmic-ray events was acquired with an applied electric field of 0.50 kV/cm. The detector performances are analyzed in detail in ref. [40].

Here we will compare the cosmic-ray data acquired with this experimental setup to a sample generated by the `larnd-sim` software. As a first step a dataset of cosmic rays is produced using the CORSIKA cosmic-ray generator v7.7410 [41], using FLUKA 2011 for the low-energy hadronic interaction [42]. Then, the passage of the cosmic rays through the Module-0 geometry is simulated using `edep-sim`, a C++ wrapper around GEANT4. The output of `edep-sim` is passed to `larnd-sim`, which produces a HDF5 sample in the same format of the Module-0 data.

Both the simulation and the data samples are passed to a reconstruction script, which takes as input an array of three-dimensional hits, one for each ADC count. The xy coordinates are given by the position of the pixel center on the tile, and the z coordinate is given by the corresponding clock tick, using the inverse formula of eq. (3.3). The hits are clustered together using the DBSCAN algorithm [43]. Hits grouped in the same cluster are fed to a principal component analysis (PCA), which returns three-dimensional segments that we define as *reconstructed tracks*. Figure 13 shows an example of a simulated cosmic ray crossing the cathode plane after the reconstruction stage, with the reconstructed track and the corresponding hits.

To improve the system self-trigger stability, 7.8% of the pixels in Module-0 were disabled due to an understood grounding issue, mostly near the tiles edges. The same pixels were also disabled in the simulation. Thus, reconstructed tracks may show artificial gaps due to the presence of disabled channels. Also, cathode-piercing tracks can be reconstructed as separated tracks, due to the non-null cathode thickness. In order to reconstruct the original cosmic-ray trajectories, we iteratively stitch together pairs of reconstructed tracks, based on their relative directions and positions at the cathode. To set the threshold for these metrics, a subsample of data events are reconstructed a second time after randomly translating and masking hits that fall into the disabled regions. We require that 95% of the gaps produced in this procedure are correctly identified and resolved.

Module-0 was operated principally in two channel threshold regimes, low and high, which correspond to thresholds of around 6000 and 12000 e^-, respectively. Here we compare the low-threshold data to the simulated sample, which was produced with a discrimination threshold roughly equivalent to the low regime one.

The amount of charge deposited per unit length (dQ/dx) is measured both in data and simulation using the reconstructed tracks, which are required to have at least 10 associated hits. In the simulation, the signal amplitude of the CSA is calibrated using a fixed gain of 250 e^-/mV. The value of the gain in the data is obtained by performing a template fit of the dQ/dx distribution with the simulated one, shown in figure 14. Here, the dQ is the sum of the hits charge associated to the reconstructed track

2023 JINST 18 P04034

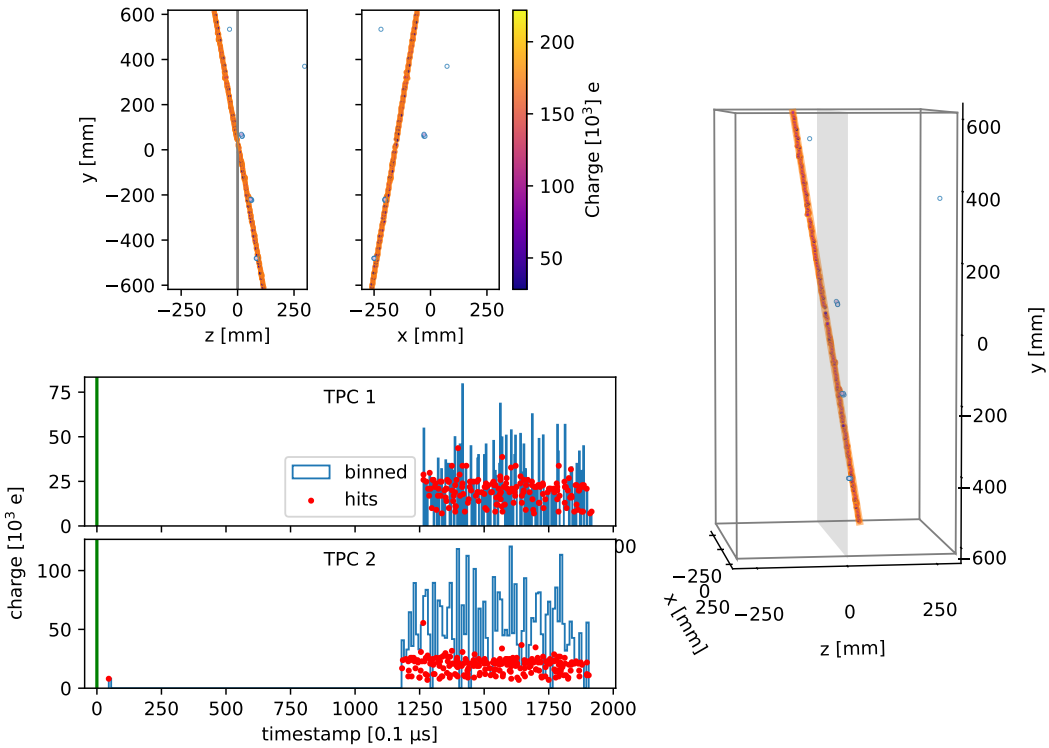


Figure 13. The ADC counts are transformed into three-dimensional hits by a reconstruction script. The hits are grouped in clusters and a fed to a principal component analysis, which returns a reconstructed track. This plot shows, clockwise from top left, the hits for a simulated cosmic ray on the zy and xy planes, a three-dimensional event display, and the hits charge as a function of time.

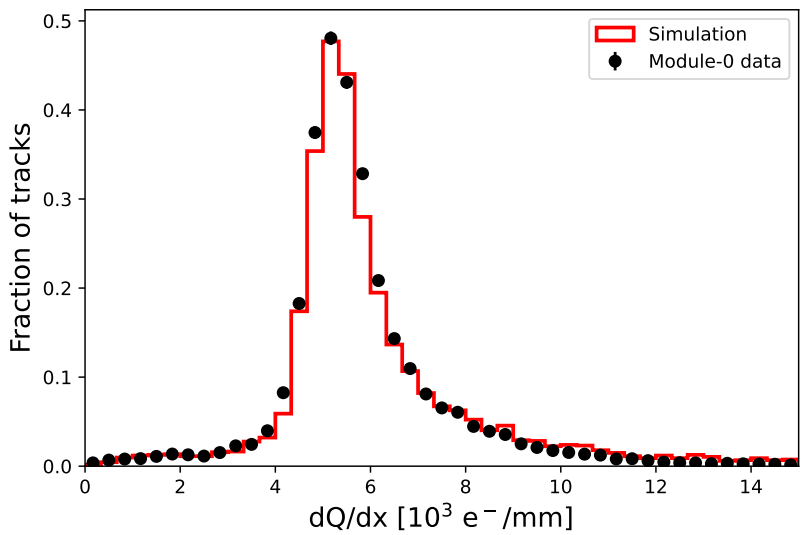


Figure 14. dQ/dx measured in data and simulation for reconstructed tracks with at least 10 associated hits. The distributions have been normalized to unity.

2023 JINST 18 P04034

and dx is the length of the reconstructed track. The best-fit value of 219 e^-/mV is compatible with the value of 221 e^-/mV, obtained with a dedicated stand-alone test of the LArPix ASIC CSA.

Reconstructed tracks can be sub-divided into segments of variable length, from 10 to 400 mm. The deposited charge dQ is the sum of the hit charges belonging to the same segment and dx is the length of the segment. The distributions have been fitted with a Gaussian-convoluted Moyal function [44].

Figure 15 shows the dQ/dx data and simulated distributions, as well as their respective fits. The dQ/dx distributions, normalized to the number of tracks, are in good agreement with data and simulation for all segment lengths.

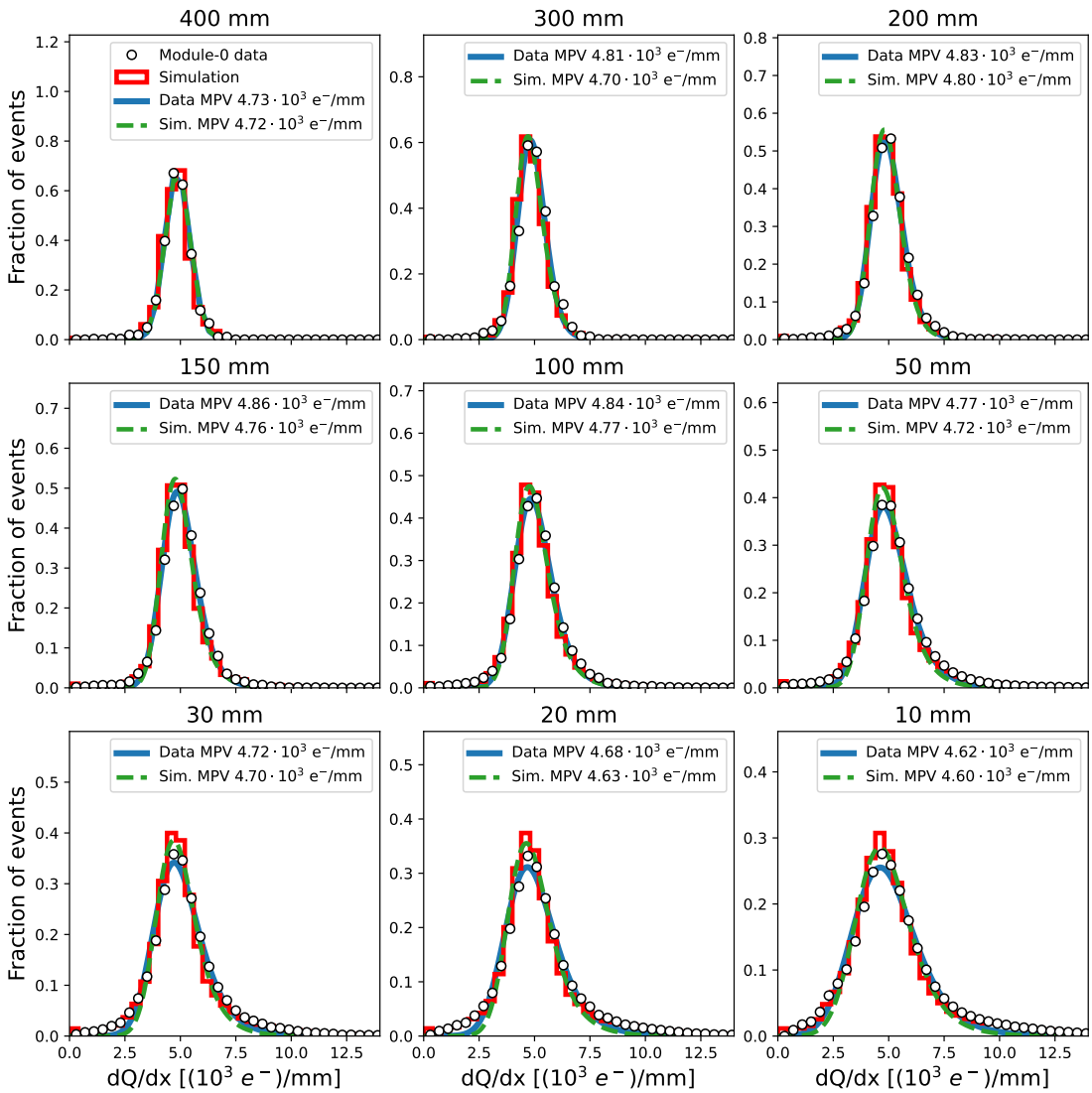


Figure 15. dQ/dx measured for segments of different lengths for low threshold runs (white dots) and a sample of simulated cosmic rays (red line). The distributions have been normalized to unity and fitted with a Gaussian-convoluted Moyal function for data (solid blue line) and simulation (dashed green line). The legend shows the most probable value (MPV) for each fit.

2023 JINST 18 P04034

7 Conclusions

We show that it is possible to implement the simulation of a pixelated LArTPC using highly-parallelized GPU algorithms. The algorithms are written in Python and they are compiled on the GPU using the Numba just-in-time compiler.

The software provides an end-to-end microphysical simulation for light readout and pixelated charge readout. Table 1 shows a summary of the speed-up achieved by the GPU implementation for various steps of the simulation.

Table 1. Summary of speed-up factors achieved by the GPU implementation for various simulation steps.

Calculation	Loop over	Quantity	GPU speed-up factor
Recombination factor	Segments	10^5	×3
Induced current	Pixels	10^3	×7314
Charge electronics response	Pixels	10^3	×985
Light time profile	Time ticks	10^5	×228
Scintillation profile	Time ticks	10^3	×568
Light electronics response	Time ticks	10^3	×1883

In general, trivial algorithms (such as the calculation of the recombination factor) require a large amount of input data before the GPU implementation starts being faster. This is expected, since there are launch and execution overheads associated with CUDA kernels. For more complex algorithms, such as the simulation of the charge readout, the GPU generally starts being may orders of magnitude faster than the CPU with only a few tens of simulated pixels in the detector. In particular, the calculation of the induced current, which is the most computationally expensive task, is around four orders of magnitude faster for $O(10^3)$ pixels. Likewise, the simulation of convolution-heavy light signals is faster on all event sizes tested, typically achieving three orders of magnitude improvement. With the acceleration achieved, we are able to simulate the charge (light) signals on $O(10^6)$ ($O(10^3)$) individual channels with sub-microsecond (nanosecond) resolution and in timescales reasonable for large Monte Carlo simulations.

As an example, the full simulation of 10^4 cosmic rays in a ton-scale LArTPC takes in total approximately 450 s, 100 s for the simulation of the passage of the initial particles through matter with `edep-sim`, and the remaining 350 s for the detector simulation with `larnd-sim`. Highly-parallelized simulation algorithms such as the one described in this document could be adapted to speed up the simulation of the DUNE Far Detector as well, which will have $O(10^6)$ readout channels [45].

We have also performed the comparison of a simulated sample of cosmic rays with the data acquired by a prototype LArTPC equipped with a large-scale pixelated readout system, called the ArgonCube Module-0 Demonstrator. Charge distributions in data and simulation are in general good agreement.

The Module-0 LArTPC was moved from Bern to Fermilab in October 2021, where it will be tested on the NuMI neutrino beamline with three other identical modules, currently under construction [40], forming the ND-LAr 2x2 prototype. The production of a dataset of accurately simulated neutrino interactions will be of fundamental importance for the analysis of this measurement. `larnd-sim` will also be used to simulate the detector response of the full ND-LAr detector and has been included

2023 JINST 18 P04034

into the DUNE Offline Computing Conceptual Design Report [46]. The DUNE collaboration is already using `larnd-sim` on the NERSC Perlmutter supercomputer, which shares a similar GPU node architecture with Cori, to perform ND-LAr 2x2 prototype simulations. A workflow chaining together `edep-sim`, `larnd-sim` and a machine learning-based reconstruction stage is currently under development.

The choice of a high-level and largely supported programming language such as Python can enable the implementation of a fully differentiable simulator in the near future, by exploiting libraries commonly used in the machine learning and artificial intelligence fields (TensorFlow [47], JAX [48]). A differentiable model will be able to use a gradient-based optimization, such as gradient descent, to automatically infer the detector simulation input and the detector physics model parameters. This technique has already been actively explored for the simulation of the photon propagation in a LArTPC [49].

Acknowledgments

This document was prepared by the DUNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359.

This work was supported by CNPq, FAPERJ, FAPEG and FAPESP, Brazil; CFI, IPP and NSERC, Canada; CERN; MŠMT, Czech Republic; ERDF, H2020-EU and MSCA, European Union; CNRS/IN2P3 and CEA, France; INFN, Italy; FCT, Portugal; NRF, South Korea; CAM, Fundación "La Caixa", Junta de Andalucía-FEDER, MICINN, and Xunta de Galicia, Spain; SERI and SNSF, Switzerland; TÜBİTAK, Turkey; The Royal Society and UKRI/STFC, United Kingdom; DOE and NSF, United States of America. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

We are also thankful to Max Katz from NVIDIA® and Daniel Margala from NERSC for the precious suggestions.

References

[1] C. Rubbia, *The Liquid Argon Time Projection Chamber: A New Concept for Neutrino Detectors*, Tech. Rep. CERN-EP-INT-77-08 (1977).

[2] A. Hitachi, T. Takahashi, N. Funayama, K. Masuda, J. Kikuchi and T. Doke, *Effect of ionization density on the time dependence of luminescence from liquid argon and xenon*, *Phys. Rev. B* **27** (1983) 5279.

[3] D.A. Dwyer et al., *LArPix: Demonstration of low-power 3D pixelated charge readout for liquid argon time projection chambers*, 2018 *JINST* **13** P10007 [arXiv:1808.02969].

[4] C. Adams, M. Del Tutto, J. Asaadi, M. Bernstein, E. Church, R. Guenette et al., *Enhancing neutrino event reconstruction with pixel-based 3D readout for liquid argon time projection chambers*, 2020 *JINST* **15** P04009 [arXiv:1912.10133].

[5] C. Yang and J. Deslippe, *Accelerate Science on Perlmutter with NERSC*, *Bull. Am. Phys. Soc.* **65** (2020) D40.00009.

2023 JINST 18 P04034

[6] DUNE collaboration, *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I Introduction to DUNE*, 2020 *JINST* **15** T08008 [arXiv:2002.02967].

[7] DUNE collaboration, *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II: DUNE Physics*, Tech. Rep. FERMILAB-PUB-20-025-ND (2020).

[8] DUNE collaboration, *Deep Underground Neutrino Experiment (DUNE) Near Detector Conceptual Design Report*, *Instruments* **5** (2021) 31 [arXiv:2103.13910].

[9] E.L. Snider and G. Petrillo, *LArSoft: Toolkit for Simulation, Reconstruction and Analysis of Liquid Argon TPC Neutrino Detectors*, *J. Phys. Conf. Ser.* **898** (2017) 042057.

[10] M. Szydagis, N. Barry, K. Kazkaz, J. Mock, D. Stolp, M. Sweany et al., *NEST: A Comprehensive Model for Scintillation Yield in Liquid Xenon*, 2011 *JINST* **6** P10002 [arXiv:1106.1613].

[11] S. Berkman, G. Cerati, K. Knoepfel, M. Mengel, A.R. Hall, M. Wang et al., *Optimizing the hit finding algorithm for liquid argon TPC neutrino detectors using parallel architectures*, 2022 *JINST* **17** P01026 [arXiv:2107.00812].

[12] P. Boyle, K. Pedro and J. Qiang, *CompF2: Theoretical Calculations and Simulation Topical Group Report*, Tech. Rep. FERMILAB-FN-1198-SCD (2022).

[13] S.R. Soleti, Peter-Madigan, , Zoya, D. Douglas, EliasTaira, D. Margala et al., *Dune/larnd-sim: v0.3.3*, 2022, https://doi.org/10.5281/zenodo.6561074.

[14] J. Nickolls, I. Buck, M. Garland and K. Skadron, *Scalable Parallel Programming with CUDA*, *Queue* **6** (2008) 40.

[15] S.K. Lam, A. Pitrou and S. Seibert, *Numba: a LLVM-based Python JIT compiler*, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, Austin, TX, U.S.A., 15 September 2015, DOI.

[16] C.R. Harris et al., *Array programming with NumPy*, *Nature* **585** (2020) 357 [arXiv:2006.10256].

[17] C. McGraw et al., *edep-sim: An Energy Deposition Simulation*, https://github.com/ClarkMcGrew/edep-sim.

[18] GEANT4 collaboration, *GEANT4–a simulation toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250.

[19] R. Okuta, Y. Unno, D. Nishino, S. Hido and C. Loomis, *CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations*, in *Proceedings of Workshop on Machine Learning Systems in The Thirty-first Annual Conference on Neural Information Processing Systems*, Long Beach, CA, U.S.A., 4–9 December 2017, http://learningsys.org/nips17/assets/papers/paper_16.pdf.

[20] HDF5 group, *HDF5*, https://github.com/HDFGroup/hdf5.

[21] J.B. Birks, *Scintillations from Organic Crystals: Specific Fluorescence and Relative Response to Different Radiations*, *Proc. Phys. Soc. A* **64** (1951) 874.

[22] ArgoNeuT collaboration, *A Study of Electron Recombination Using Highly Ionizing Particles in the ArgoNeuT Liquid Argon TPC*, 2013 *JINST* **8** P08005 [arXiv:1306.1712].

[23] ICARUS collaboration, *Study of electron recombination in liquid argon with the ICARUS TPC*, *Nucl. Instrum. Meth. A* **523** (2004) 275.

[24] Y. Li et al., *Measurement of Longitudinal Electron Diffusion in Liquid Argon*, *Nucl. Instrum. Meth. A* **816** (2016) 160 [arXiv:1508.07059].

[25] DUNE collaboration, *Design, construction and operation of the ProtoDUNE-SP Liquid Argon TPC*, 2022 *JINST* **17** P01005 [arXiv:2108.01902].

2023 JINST 18 P04034

[26] S. Ramo, *Currents induced by electron motion*, *Proc. IRE* **27** (1939) 584.

[27] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities*, *Int. J. Num. Meth. Eng.* **79** (2009) 1309.

[28] P. Ponomarev, J. Keranen and P. Pasanen, *Electromagnetic transient finite element 3D modelling of electrical machines using open-source tools*, in *Proceedings of the XXII International Conference on Electrical Machines (ICEM)*, Lausanne, Switzerland, 4–7 September 2016, pp. 1657–1661.

[29] S. Amoruso et al., *Analysis of the liquid argon purity in the ICARUS T600 TPC*, *Nucl. Instrum. Meth. A* **516** (2004) 68.

[30] W. Walkowiak, *Drift velocity of free electrons in liquid argon*, *Nucl. Instrum. Meth. A* **449** (2000) 288.

[31] M.J. Wolfe, C. Shanklin and L. Ortega, *High Performance Compilers for Parallel Computing*, Addison-Wesley Longman Publishing Co., Inc., U.S.A. (1995).

[32] P. Koller, *A Method to Improve the Neutrino Energy Reconstruction in LArTPCs*, Ph.D. thesis, Universität Bern, 2021.

[33] S. Kubota, M. Hishida and A. Nohara, *Variation of scintillation decay in liquid argon excited by electrons and alpha particles*, *Nucl. Instrum. Meth.* **150** (1978) 561.

[34] E. Morikawa, R. Reininger, P. Gürtler, V. Saile and P. Laporte, *Argon, krypton, and xenon excimer luminescence: From the dilute gas to the condensed phase*, *J. Chem. Phys.* **91** (1989) 1469.

[35] WArP collaboration, *Effects of Nitrogen contamination in liquid Argon*, 2010 *JINST* **5** P06003 [arXiv:0804.1217].

[36] WArP collaboration, *Oxygen contamination in liquid Argon: Combined effects on ionization electron charge and scintillation light*, 2010 *JINST* **5** P05003 [arXiv:0804.1222].

[37] E. Segreto, *Properties of Liquid Argon Scintillation Light Emission*, *Phys. Rev. D* **103** (2021) 043001 [arXiv:2012.06527].

[38] L. Devroye, *Discrete univariate distributions*, in *Non-Uniform Random Variate Generation*, pp. 485–553, Springer (1986).

[39] S. Williams, A. Waterman and D. Patterson, *Roofline*, *Commun. ACM* **52** (2009) 65.

[40] DUNE collaboration, *Performance of a modular ton-scale pixel-readout liquid argon Time Projection Chamber*, in preparation, 2022.

[41] D. Heck, J. Knapp, J.N. Capdevielle, G. Schatz and T. Thouw, *CORSIKA: A Monte Carlo code to simulate extensive air showers*, Tech. Rep. FZKA-6019 (1998).

[42] G. Battistoni et al., *Overview of the FLUKA code*, *Ann. Nucl. Energy* **82** (2015) 10.

[43] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR, U.S.A., 2–4 August 1996, pp. 226–331.

[44] J.E. Moyal, *Theory of ionization fluctuations*, *Phil. Mag. Ser. 7* **46** (1955) 263.

[45] DUNE collaboration, *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume IV: Far Detector Single-phase Technology*, 2020 *JINST* **15** T08010 [arXiv:2002.03010].

[46] DUNE collaboration, *DUNE Offline Computing Conceptual Design Report*, Tech. Rep. FERMILAB-DESIGN-2022-01 (2022).

2023 JINST 18 P04034

[47] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, [arXiv:1603.04467].

[48] J. Bradbury et al., *JAX: composable transformations of Python+NumPy programs*, https://github.com/google/jax, 2022.

[49] M. Lei, K.V. Tsang, S. Gasiorowski, C. Li, Y. Nashed, G. Petrillo et al., *Implicit Neural Representation as a Differentiable Surrogate for Photon Propagation in a Monolithic Neutrino Detector*, arXiv:2211.01505.

The DUNE collaboration

A. Abed Abud,[35] B. Abi,[164] R. Acciarri,[68] M.A. Acero,[11] M.R. Adames,[201] G. Adamov,[74] M. Adamowski,[68] D. Adams,[20] M. Adinolfi,[19] C. Adriano,[30] A. Aduszkiewicz,[83] J. Aguilar,[132] Z. Ahmad,[213] J. Ahmed,[216] B. Aimard,[53] F. Akbar,[182] K. Allison,[43] S. Alonso Monsalve,[35] M. Alrashed,[124] C. Alt,[62] A. Alton,[12] R. Alvarez,[39] P. Amedo,[88,87] J. Anderson,[7] D.A. Andrade,[89] C. Andreopoulos,[185,134] M. Andreotti,[96,69] M.P. Andrews,[68] F. Andrianala,[5] S. Andringa,[133] N. Anfimov,[122] W.L. Anicézio Campanelli,[64] A. Ankowski,[191] M. Antoniassi,[201] M. Antonova,[87] A. Antoshkin,[122] S. Antusch,[13] A. Aranda-Fernandez,[42] L. Arellano,[140] L.O. Arnold,[45] M.A. Arroyave,[60] J. Asaadi,[205] A. Ashkenazi,[202] L. Asquith,[199] A. Aurisano,[40] V. Aushev,[130] D. Autiero,[113] M. Ayala-Torres,[41] F. Azfar,[164] A. Back,[93] H. Back,[165] J.J. Back,[216] I. Bagaturia,[74] L. Bagby,[68] N. Balashov,[122] S. Balasubramanian,[68] P. Baldi,[24] W. Baldini,[96] B. Baller,[68] B. Bambah,[84] F. Barao,[133,115] G. Barenboim,[87] P. Barham Alzás,[35] G.J. Barker,[216] W. Barkhouse,[156] C. Barnes,[144] G. Barr,[164] J. Barranco Monarca,[79] A. Barros,[201] N. Barros,[133,63] J.L. Barrow,[141] A. Basharina-Freshville,[211] A. Bashyal,[7] V. Basque,[68] C. Batchelor,[59] J.B.R. Battat,[217] F. Battisti,[164] F. Bay,[4] M.C.Q. Bazetto,[30] J.L.L. Bazo Alba,[177] J.F. Beacom,[162] E. Bechetoille,[113] B. Behera,[44] E. Belchior,[30] L. Bellantoni,[68] G. Bellettini,[105,175] V. Bellini,[95,31] O. Beltramello,[35] N. Benekos,[35] C. Benitez Montiel,[9] D. Benjamin,[20] F. Bento Neves,[133] J. Berger,[44] S. Berkman,[68] P. Bernardini,[99,186] R.M. Berner,[14] A. Bersani,[98] S. Bertolucci,[94,17] M. Betancourt,[68] A. Betancur Rodríguez,[60] A. Bevan,[180] Y. Bezawada,[23] A.T. Bezerra,[64] T.J. Bezerra,[199] J. Bhambure,[196] A. Bhardwaj,[136] V. Bhatnagar,[167] M. Bhattacharjee,[91] M. Bhattacharya,[68] D. Bhattarai,[150] S. Bhuller,[19] B. Bhuyan,[91] S. Biagi,[107] J. Bian,[24] M. Biassoni,[100] K. Biery,[68] B. Bilki,[15,111] M. Bishai,[20] V. Bisignani,[102] A. Bitadze,[140] A. Blake,[131] F.D. Blaszczyk,[68] G.C. Blazey,[157] D. Blend,[111] E. Blucher,[37] J. Boissevain,[135] S. Bolognesi,[34] T. Bolton,[124] L. Bomben,[100,110] M. Bonesini,[100,146] C. Bonilla-Diaz,[32] F. Bonini,[20] A. Booth,[180] F. Boran,[15] S. Bordoni,[35] A. Borkum,[199] N. Bostan,[111] P. Bour,[50] D. Boyden,[157] J. Bracinik,[16] D. Braga,[68] D. Brailsford,[131] A. Branca,[100] A. Brandt,[205] M. Bravo-Moreno,[75] J. Bremer,[35] C. Brew,[185] S.J. Brice,[68] C. Brizzolari,[100,146] C. Bromberg,[145] J. Brooke,[19] A. Bross,[68] G. Brunetti,[100,146] M. Brunetti,[216] N. Buchanan,[44] H. Budd,[182] J. Buergi,[14] G. Caceres V.,[23] I. Cagnoli,[94,17] T. Cai,[223] D. Caiulo,[113] R. Calabrese,[96,69] P. Calafiura,[132] J. Calcutt,[163] M. Calin,[21] L. Calivers,[14] S. Calvez,[44] E. Calvo,[39] A. Caminata,[98] D. Caratelli,[27] D. Carber,[44] J.C. Carceller,[211] G. Carini,[20] B. Carlus,[113] M.F. Carneiro,[20] P. Carniti,[100] I. Caro Terrazas,[44] H. Carranza,[205] N. Carrara,[23] L. Carroll,[124] T. Carroll,[220] A. Carter,[183] J.F. Castaño Forero,[6] A. Castillo,[189] E. Catano-Mur,[219] C. Cattadori,[100] F. Cavalier,[168] G. Cavallaro,[100] F. Cavanna,[68] S. Centro,[166] G. Cerati,[68] A. Cervelli,[94] A. Cervera Villanueva,[87] K. Chakraborty,[174] M. Chalifour,[35] A. Chappell,[216] E. Chardonnet,[169] N. Charitonidis,[35] A. Chatterjee,[176] S. Chattopadhyay,[213] H. Chen,[20] M. Chen,[24] Y. Chen,[14,191] Z. Chen,[196] Z. Chen-Wishart,[183] Y. Cheon,[210] D. Cherdack,[83] C. Chi,[45] S. Childress,[68] R. Chirco,[89] A. Chiriacescu,[21] N. Chitirasreemadam,[105,175] K. Cho,[127] S. Choate,[157] D. Chokheli,[74] P.S. Chong,[172] B. Chowdhury,[7] A. Christensen,[44] D. Christian,[68] G. Christodoulou,[35] A. Chukanov,[122] M. Chung,[210] E. Church,[165] V. Cicero,[94,17] D. Clapa,[215] P. Clarke,[59]

2023 JINST 18 P04034

G. Cline,[132] T.E. Coan,[195] A.G. Cocco,[102] J.A.B. Coelho,[169] A. Cohen,[169] J. Collot,[78] E. Conley,[57] J.M. Conrad,[141] M. Convery,[191] S. Copello,[98] P. Cova,[101,170] C. Cox,[183] L. Cremaldi,[150] L. Cremonesi,[180] J.I. Crespo-Anadón,[39] M. Crisler,[68] E. Cristaldo,[101,9] J. Crnkovic,[68] G. Crone,[211] R. Cross,[131] A. Cudd,[43] C. Cuesta,[39] Y. Cui,[26] D. Cussans,[19] J. Dai,[78] O. Dalager,[24] R. Dallavalle,[169] H. da Motta,[33] Z.A. Dar,[219] R. Darby,[199] L. Da Silva Peres,[67] C. David,[223,68] Q. David,[113] G.S. Davies,[150] S. Davini,[98] J. Dawson,[169] K. De,[205] S. De,[2] R. De Aguiar,[30] P. De Almeida,[30] P. Debbins,[111] I. De Bonis,[53] M.P. Decowski,[154,3] A. de Gouvêa,[158] P.C. De Holanda,[30] I.L. De Icaza Astiz,[199] A. Deisting,[139] P. De Jong,[154,3] A. De la Torre,[39] A. Delbart,[34] V. De Leo,[188,106] D. Delepine,[79] M. Delgado,[100,146] A. Dell'Acqua,[35] N. Delmonte,[101,170] P. De Lurgio,[7] J.R.T. de Mello Neto,[67] D.M. DeMuth,[212] S. Dennis,[29] C. Densham,[185] P. Denton,[20] G.W. Deptuch,[20] A. De Roeck,[35] V. De Romeri,[87] G. De Souza,[30] J.P. Detje,[29] R. Devi,[119] R. Dharmapalan,[82] M. Dias,[209] J.S. Díaz,[93] F. Díaz,[177] F. Di Capua,[102,151] A. Di Domenico,[188,106] S. Di Domizio,[98,73] S. Di Falco,[105] L. Di Giulio,[35] P. Ding,[68] L. Di Noto,[98,73] E. Diociaiuti,[97] C. Distefano,[107] R. Diurba,[14] M. Diwan,[20] Z. Djurcic,[7] D. Doering,[191] S. Dolan,[35] F. Dolek,[15] M.J. Dolinski,[56] D. Domenici,[97] L. Domine,[191] S. Donati,[105,175] Y. Donon,[35] S. Doran,[112] D. Douglas,[145] A. Dragone,[191] F. Drielsma,[191] L. Duarte,[209] D. Duchesneau,[53] K. Duffy,[164,68] K. Dugas,[24] P. Dunne,[90] B. Dutta,[203] H. Duyang,[192] O. Dvornikov,[82] D.A. Dwyer,[132] A.S. Dyshkant,[157] M. Eads,[157] A. Earle,[199] D. Edmunds,[145] J. Eisch,[68] L. Emberger,[140,142] P. Englezos,[184] A. Ereditato,[221] T. Erjavec,[23] C.O. Escobar,[68] J.J. Evans,[140] E. Ewart,[93] A.C. Ezeribe,[190] K. Fahey,[68] L. Fajt,[35] A. Falcone,[100,146] M. Fani',[135] C. Farnese,[103] Y. Farzan,[114] D. Fedoseev,[122] J. Felix,[79] Y. Feng,[112] E. Fernandez-Martinez,[138] F. Ferraro,[98,73] L. Fields,[159] P. Filip,[49] A. Filkins,[200] F. Filthaut,[154,181] R. Fine,[135] G. Fiorillo,[102,151] M. Fiorini,[96,69] V. Fischer,[112] R.S. Fitzpatrick,[144] W. Flanagan,[52] B. Fleming,[37,221] R. Flight,[182] S. Fogarty,[44] W. Foreman,[89] J. Fowler,[57] J. Franc,[50] D. Franco,[221] J. Freeman,[68] J. Fried,[20] A. Friedland,[191] S. Fuess,[68] I.K. Furic,[70] K. Furman,[180] A.P. Furmanski,[149] A. Gabrielli,[94,17] A. Gago,[177] H. Gallagher,[208] A. Gallas,[168] A. Gallego-Ros,[39] N. Gallice,[101,147] V. Galymov,[113] E. Gamberini,[35] T. Gamble,[190] F. Ganacim,[201] R. Gandhi,[80] S. Ganguly,[68] F. Gao,[176] S. Gao,[20] D. Garcia-Gamez,[75] M. Á. García-Peris,[87] S. Gardiner,[68] D. Gastler,[18] A. Gauch,[14] J. Gauvreau,[161] P. Gauzzi,[188,106] G. Ge,[45] N. Geffroy,[53] B. Gelli,[30] A. Gendotti,[62] S. Gent,[194] L. Gerlach,[20] Z. Ghorbani-Moghaddam,[98] P. Giammaria,[30] T. Giammaria,[96,69] N. Giangiacomi,[207] D. Gibin,[166,103] I. Gil-Botella,[39] S. Gilligan,[163] A. Gioiosa,[105] S. Giovannella,[97] C. Girerd,[113] A.K. Giri,[92] D. Gnani,[132] O. Gogota,[130] M. Gold,[152] S. Gollapinni,[135] K. Gollwitzer,[68] R.A. Gomes,[65] L.V. Gomez Bermeo,[189] L.S. Gomez Fajardo,[189] F. Gonnella,[16] D. Gonzalez-Diaz,[88] M. Gonzalez-Lopez,[138] M.C. Goodman,[7] O. Goodwin,[140] S. Goswami,[174] C. Gotti,[100] J. Goudeau,[136] E. Goudzovski,[16] C. Grace,[132] R. Gran,[148] E. Granados,[79] P. Granger,[169] C. Grant,[18] D. Gratieri,[72] P. Green,[164] S. Greenberg,[22,132] L. Greenler,[220] J. Greer,[19] J. Grenard,[35] W.C. Griffith,[199] F.T. Groetschla,[35] M. Groh,[44] K. Grzelak,[215] W. Gu,[20] V. Guarino,[7] M. Guarise,[96,69] R. Guenette,[140] E. Guerard,[168] M. Guerzoni,[94] D. Guffanti,[100] A. Guglielmi,[103] B. Guo,[192] A. Gupta,[191] V. Gupta,[154,3] K.K. Guthikonda,[128] P. Guzowski,[140] M.M. Guzzo,[30] S. Gwon,[38] C. Ha,[38] K. Haaf,[68] A. Habig,[148] H. Hadavand,[205] A. Hadef,[139] R. Haenni,[14] L. Hagaman,[221] A. Hahn,[68] J. Haiston,[193] P. Hamacher-Baumann,[164] T. Hamernik,[68] P. Hamilton,[90] J. Han,[176] J. Hancock,[16] F. Happacher,[97] D.A. Harris,[223,68] J. Hartnell,[199] T. Hartnett,[185] J. Harton,[44] T. Hasegawa,[126] C. Hasnip,[164] R. Hatcher,[68] K.W. Hatfield,[24] A. Hatzikoutelis,[187] C. Hayes,[93] K. Hayrapetyan,[180] J. Hays,[180] E. Hazen,[18] M. He,[83] A. Heavey,[68] K.M. Heeger,[221] J. Heise,[198] S. Henry,[182] M.A. Hernandez Morquecho,[89] K. Herner,[68] V. Hewes,[40] C. Hilgenberg,[149] T. Hill,[85] S.J. Hillier,[16] A. Himmel,[68] E. Hinkle,[37] L.R. Hirsch,[201] J. Ho,[55] J. Hoff,[68] A. Holin,[185] T. Holvey,[164] E. Hoppe,[165] G.A. Horton-Smith,[124] M. Hostert,[149] T. Houdy,[168] A. Hourlier,[141] B. Howard,[68] R. Howell,[182] J. Hoyos Barrios,[143] I. Hristova,[185] M.S. Hronek,[68] J. Huang,[23] R.G. Huang,[132] Z. Hulcher,[191] G. Iles,[90] N. Ilic,[207] A.M. Iliescu,[94] R. Illingworth,[68] G. Ingratta,[94,17] A. Ioannisian,[222] B. Irwin,[149] L. Isenhower,[1] M. Ismerio Oliveira,[67] R. Itay,[191] C.M. Jackson,[165] V. Jain,[2] E. James,[68] W. Jang,[205] B. Jargowsky,[24] F. Jediny,[50] D. Jena,[68] Y.S. Jeong,[38]

2023 JINST 18 P04034

C. Jesús-Valls,[86] X. Ji,[20] J. Jiang,[196] L. Jiang,[214] A. Jipa,[21] J.H. Jo,[221] F.R. Joaquim,[133,115] W. Johnson,[193] B. Jones,[205] R. Jones,[190] N. Jovancevic,[160] M. Judah,[176] C.K. Jung,[196] T. Junk,[68] Y. Jwa,[45] M. Kabirnezhad,[90] A. Kaboth,[183,185] I. Kadenko,[130] I. Kakorin,[122] A. Kalitkina,[122] D. Kalra,[45] O. Kamer Koseyan,[111] F. Kamiya,[66] D.M. Kaplan,[89] G. Karagiorgi,[45] G. Karaman,[111] A. Karcher,[132] Y. Karyotakis,[53] S. Kasai,[129] S.P. Kasetti,[136] L. Kashur,[44] I. Katsioulas,[16] N. Kazaryan,[222] E. Kearns,[18] P. Keener,[172] K.J. Kelly,[35] E. Kemp,[30] O. Kemularia,[74] Y. Kermaidic,[168] W. Ketchum,[68] S.H. Kettell,[20] M. Khabibullin,[109] A. Khotjantsev,[109] A. Khvedelidze,[74] D. Kim,[203] B. King,[68] B. Kirby,[45] M. Kirby,[68] J. Klein,[172] J. Kleykamp,[150] A. Klustova,[90] T. Kobilarcik,[68] K. Koehler,[220] L.W. Koerner,[83] D.H. Koh,[191] S. Kohn,[22,132] P.P. Koller,[14] L. Kolupaeva,[122] D. Korablev,[122] M. Kordosky,[219] T. Kosc,[78] U. Kose,[35] V.A. Kostelecký,[93] K. Kothekar,[19] I. Kotler,[56] V. Kozhukalov,[122] R. Kralik,[199] L. Kreczko,[19] F. Krennrich,[112] I. Kreslo,[14] W. Kropp,[24] T. Kroupova,[172] M. Kubu,[35] Y. Kudenko,[109] V.A. Kudryavtsev,[190] S. Kuhlmann,[7] S. Kulagin,[109] J. Kumar,[82] P. Kumar,[190] P. Kunze,[53] R. Kuravi,[132] N. Kurita,[191] C. Kuruppu,[192] V. Kus,[50] T. Kutter,[136] J. Kvasnicka,[49] D. Kwak,[210] A. Lambert,[132] B.J. Land,[172] C.E. Lane,[56] K. Lang,[206] T. Langford,[221] M. Langstaff,[140] F. Lanni,[35] O. Lantwin,[53] J. Larkin,[20] P. Lasorak,[90] D. Last,[172] A. Laundrie,[220] G. Laurenti,[94] A. Lawrence,[132] P. Laycock,[20] I. Lazanu,[21] M. Lazzaroni,[101,147] T. Le,[208] S. Leardini,[88] J. Learned,[82] P. LeBrun,[113] T. LeCompte,[191] C. Lee,[68] V. Legin,[130] G. Lehmann Miotto,[35] R. Lehnert,[93] M.A. Leigui de Oliveira,[66] M. Leitner,[132] L.M. Lepin,[140] S.W. Li,[191] Y. Li,[20] H. Liao,[124] C.S. Lin,[132] S. Lin,[136] D. Lindebaum,[19] R.A. Lineros,[32] J. Ling,[197] A. Lister,[220] B.R. Littlejohn,[89] J. Liu,[24] Y. Liu,[37] S. Lockwitz,[68] T. Loew,[132] M. Lokajicek,[49] I. Lomidze,[74] K. Long,[90] T. Lord,[216] J.M. LoSecco,[159] W.C. Louis,[135] X.-G. Lu,[216] K.B. Luk,[22,132] B. Lunday,[172] X. Luo,[27] E. Luppi,[96,69] T. Lux,[86] V.P. Luzio,[66] J. Maalmi,[168] D. MacFarlane,[191] A.A. Machado,[30] P. Machado,[68] C.T. Macias,[93] J.R. Macier,[68] M. MacMahon,[211] A. Maddalena,[77] A. Madera,[35] P. Madigan,[22,132] S. Magill,[7] K. Mahn,[145] A. Maio,[133,63] A. Major,[57] K. Majumdar,[134] J.A. Maloney,[51] M. Man,[207] G. Mandrioli,[94] R.C. Mandujano,[24] J. Maneira,[133,63] L. Manenti,[211] S. Manly,[182] A. Mann,[208] K. Manolopoulos,[185] M. Manrique Plata,[93] V.N. Manyam,[20] M. Marchan,[68] A. Marchionni,[68] W. Marciano,[20] D. Marfatia,[82] C. Mariani,[214] J. Maricic,[82] F. Marinho,[116] A.D. Marino,[43] T. Markiewicz,[191] D. Marsden,[140] M. Marshak,[149] C.M. Marshall,[182] J. Marshall,[216] J. Marteau,[113] J. Martín-Albo,[87] N. Martinez,[124] D.A. Martinez Caicedo ,[193] F. Martínez López,[180] P. Martínez Miravé,[87] S. Martynenko,[20] V. Mascagna,[100,110] K. Mason,[208] A. Mastbaum,[184] F. Matichard,[132] S. Matsuno,[82] J. Matthews,[136] C. Mauger,[172] N. Mauri,[94,17] K. Mavrokoridis,[134] I. Mawby,[216] R. Mazza,[100] A. Mazzacane,[68] T. McAskill,[217] E. McCluskey,[68] N. McConkey,[211] K.S. McFarland,[182] C. McGrew,[196] A. McNab,[140] A. Mefodiev,[109] P. Mehta,[120] P. Melas,[10] O. Mena,[87] H. Mendez,[178] P. Mendez,[35] D.P. Méndez,[20] A. Menegolli,[104,171] G. Meng,[103] M.D. Messier,[93] W. Metcalf,[136] M. Mewes,[93] H. Meyer,[218] T. Miao,[68] G. Michna,[194] V. Mikola,[211] R. Milincic,[82] G. Miller,[140] W. Miller,[149] J. Mills,[208] O. Mineev,[109] A. Minotti,[100,146] O.G. Miranda,[41] S. Miryala,[20] S. Miscetti,[97] C.S. Mishra,[68] S.R. Mishra,[192] A. Mislivec,[149] M. Mitchell,[136] D. Mladenov,[35] I. Mocioiu,[173] K. Moffat,[58] A. Mogan,[44] N. Moggi,[94,17] R. Mohanta,[84] T.A. Mohayai,[68] N. Mokhov,[68] J. Molina,[9] L. Molina Bueno,[87] E. Montagna,[94,17] A. Montanari,[94] C. Montanari,[104,68,171] D. Montanari,[68] D. Montanino,[99,186] L.M. Montaño Zetina,[41] S.H. Moon,[210] M. Mooney,[44] A.F. Moor,[29] D. Moreno,[6] L. Morescalchi,[105] D. Moretti,[100] C. Morris,[83] C. Mossey,[68] M. Mote,[136] E. Motuk,[211] C.A. Moura,[66] J. Mousseau,[144] G. Mouster,[131] W. Mu,[68] L. Mualem,[28] J. Mueller,[44] M. Muether,[218] F. Muheim,[59] A. Muir,[54] M. Mulhearn,[23] D. Munford,[83] L.J. Munteanu,[35] H. Muramatsu,[149] J. Muraz,[53] M. Murphy,[214] S. Murphy,[62] J. Musser,[93] J. Nachtman,[111] Y. Nagai,[61] S. Nagu,[137] M. Nalbandyan,[222] R. Nandakumar,[185] D. Naples,[176] S. Narita,[117] A. Nath,[91] A. Navrer-Agasson,[140] N. Nayak,[20] M. Nebot-Guinot,[59] K. Negishi,[117] J.K. Nelson,[219] M. Nelson,[111] J. Nesbit,[220] M. Nessi,[68,35] D. Newbold,[185] M. Newcomer,[172] H. Newton,[54] R. Nichol,[211] F. Nicolas-Arnaldos,[75] A. Nikolica,[172] J. Nikolov,[160] E. Niner,[68] K. Nishimura,[82] A. Norman,[68] A. Norrick,[68] P. Novella,[87] J.A. Nowak,[131] M. Oberling,[7] J.P. Ochoa-Ricoux,[24] A. Olivier,[182] A. Olshevskiy,[122] Y. Onel,[111] Y. Onishchuk,[130] J. Ott,[24] L. Pagani,[23]

2023 JINST 18 P04034

G. Palacio,[60] O. Palamara,[68] S. Palestini,[35] J.M. Paley,[68] M. Pallavicini,[98,73] C. Palomares,[39] S. Pan,[174] W. Panduro Vazquez,[183] E. Pantic,[23] V. Paolone,[176] V. Papadimitriou,[68] R. Papaleo,[107] A. Papanestis,[185] S. Paramesvaran,[19] S. Parke,[68] E. Parozzi,[100,146] S. Parsa,[14] Z. Parsa,[20] S. Parveen,[120] M. Parvu,[21] D. Pasciuto,[105] S. Pascoli,[58,17] L. Pasqualini,[94,17] J. Pasternak,[90] J. Pater,[140] C. Patrick,[59,211] L. Patrizii,[94] R.B. Patterson,[28] S.J. Patton,[132] T. Patzak,[169] A. Paudel,[68] L. Paulucci,[66] Z. Pavlovic,[68] G. Pawloski,[149] D. Payne,[134] V. Pec,[49] S.J.M. Peeters,[199] A. Pena Perez,[191] E. Pennacchio,[113] A. Penzo,[111] O.L.G. Peres,[30] Y.F. Perez Gonzalez,[58] L. Pérez-Molina,[39] C. Pernas,[219] J. Perry,[59] D. Pershey,[57] G. Pessina,[100] G. Petrillo,[191] C. Petta,[95,31] R. Petti,[192] V. Pia,[94,17] F. Piastra,[14] L. Pickering,[183] F. Pietropaolo,[35,103] V.L. Pimentel,[47,30] G. Pinaroli,[20] K. Plows,[164] R. Plunkett,[68] T. Pollman,[154,3] F. Pompa,[87] X. Pons,[35] N. Poonthottathil,[112] F. Poppi,[94,17] S. Pordes,[68] J. Porter,[199] M. Potekhin,[20] R. Potenza,[95,31] B.V.K.S. Potukuchi,[119] J. Pozimski,[90] M. Pozzato,[94,17] S. Prakash,[30] T. Prakash,[132] C. Pratt,[23] M. Prest,[100] F. Psihas,[68] D. Pugnere,[113] X. Qian,[20] J.L. Raaf,[68] V. Radeka,[20] J. Rademacker,[19] R. Radev,[35] B. Radics,[223] A. Rafique,[7] E. Raguzin,[20] M. Rai,[216] M. Rajaoalisoa,[40] I. Rakhno,[68] A. Rakotonandrasana,[5] L. Rakotondravohitra,[5] R. Rameika,[68] M.A. Ramirez Delgado,[172] B. Ramson,[68] A. Rappoldi,[104,171] G. Raselli,[104,171] P. Ratoff,[131] S. Raut,[196] H. Razafinime,[40] R.F. Razakamiandra,[5] E.M. Rea,[149] J.S. Real,[78] B. Rebel,[220,68] R. Rechenmacher,[68] M. Reggiani-Guzzo,[140] J. Reichenbacher,[193] S.D. Reitzner,[68] H. Rejeb Sfar,[35] A. Renshaw,[83] S. Rescia,[20] F. Resnati,[35] M. Ribas,[201] S. Riboldi,[101] C. Riccio,[196] G. Riccobene,[107] L.C.J. Rice,[176] J.S. Ricol,[78] A. Rigamonti,[35] Y. Rigaut,[62] E.V. Rincón,[60] A. Ritchie-Yates,[183] D. Rivera,[135] R. Rivera,[68] A. Robert,[78] J.L. Rocabado Rocha,[87] L. Rochester,[191] M. Roda,[134] P. Rodrigues,[164] M.J. Rodriguez Alonso,[35] J. Rodriguez Rondon,[193] E. Romeo,[102] S. Rosauro-Alcaraz,[168] P. Rosier,[168] M. Rossella,[104,171] M. Rossi,[35] M. Ross-Lonergan,[135] J. Rout,[120] P. Roy,[218] A. Rubbia,[62] C. Rubbia,[76] G. Ruiz Ferreira,[140] B. Russell,[132] D. Ruterbories,[182] A. Rybnikov,[122] A. Saa-Hernandez,[88] R. Saakyan,[211] S. Sacerdoti,[169] N. Sahu,[92] P. Sala,[101,35] N. Samios,[20] O. Samoylov,[122] M.C. Sanchez,[71] V. Sandberg,[135] D.A. Sanders,[150] D. Sankey,[185] D. Santoro,[101] N. Saoulidou,[10] P. Sapienza,[107] C. Sarasty,[40] I. Sarcevic,[8] I. Sarra,[97] G. Savage,[68] V. Savinov,[176] G. Scanavini,[221] A. Scaramelli,[104] A. Scarff,[190] A. Scarpelli,[20] T. Schefke,[136] H. Schellman,[163,68] S. Schifano,[96,69] P. Schlabach,[68] D. Schmitz,[37] A.W. Schneider,[141] K. Scholberg,[57] A. Schukraft,[68] E. Segreto,[30] A. Selyunin,[122] C.R. Senise,[209] J. Sensenig,[172] D. Sgalaberna,[62] M.H. Shaevitz,[45] S. Shafaq,[120] F. Shaker,[223] M. Shamma,[26] P. Shanahan,[68] R. Sharankova,[208] H.R. Sharma,[119] R. Sharma,[20] R. Kumar,[179] K. Shaw,[199] T. Shaw,[68] K. Shchablo,[113] C. Shepherd-Themistocleous,[185] A. Sheshukov,[122] W. Shi,[196] S. Shin,[121] I. Shoemaker,[214] D. Shooltz,[145] R. Shrock,[196] J. Silber,[132] L. Simard,[168] J. Sinclair,[191] G. Sinev,[193] Jaydip Singh,[137] J. Singh,[137] L. Singh,[48] P. Singh,[180] V. Singh,[48] S. Singh Chauhan,[167] R. Sipos,[35] G. Sirri,[94] A. Sitraka,[193] K. Siyeon,[38] K. Skarpaas,[191] E. Smith,[93] P. Smith,[93] J. Smolik,[50] M. Smy,[24] E.L. Snider,[68] P. Snopok,[89] D. Snowden-Ifft,[161] M. Soares Nunes,[200] H. Sobel,[24] M. Soderberg,[200] S. Sokolov,[122] C.J. Solano Salinas,[108] S. Söldner-Rembold,[140] S.R. Soleti,[132,*] N. Solomey,[218] V. Solovov,[133] W.E. Sondheim,[135] M. Sorel,[87] A. Sotnikov,[122] J. Soto-Oton,[39] A. Sousa,[40] K. Soustruznik,[36] F. Spagliardi,[164] M. Spanu,[100,146] J. Spitz,[144] N.J.C. Spooner,[190] K. Spurgeon,[200] D. Stalder,[9] M. Stancari,[68] L. Stanco,[103,166] J. Steenis,[23] R. Stein,[19] H.M. Steiner,[132] A.F. Steklain Lisbôa,[201] A. Stepanova,[122] J. Stewart,[20] B. Stillwell,[37] J. Stock,[193] F. Stocker,[35] T. Stokes,[136] M. Strait,[149] T. Strauss,[68] L. Strigari,[203] A. Stuart,[42] J.G. Suarez,[60] J. Subash,[16] A. Surdo,[99] V. Susic,[13] L. Suter,[68] C.M. Sutera,[95,31] K. Sutton,[28] Y. Suvorov,[102,151] R. Svoboda,[23] S.K. Swain,[155] B. Szczerbinska,[204] A.M. Szelc,[59] A. Taffara,[105] N. Talukdar,[192] J. Tamara,[6] H.A. Tanaka,[191] S. Tang,[20] N. Taniuchi,[29] B. Tapia Oregui,[206] A. Tapper,[90] S. Tariq,[68] E. Tarpara,[20] N. Tata,[81] E. Tatar,[85] R. Tayloe,[93] A.M. Teklu,[196] P. Tennessen,[132,4] M. Tenti,[94] K. Terao,[191] F. Terranova,[100,146] G. Testera,[98] T. Thakore,[40] A. Thea,[185] A. Thompson,[203] C. Thorn,[20] S.C. Timm,[68] V. Tishchenko,[20] N. Todorović,[160] L. Tomassetti,[96,69] A. Tonazzo,[169] D. Torbunov,[20] M. Torti,[100,146] M. Tortola,[87] F. Tortorici,[95,31] N. Tosi,[94] D. Totani,[27] M. Toups,[68] C. Touramanis,[134] R. Travaglini,[94] J. Trevor,[28] S. Trilov,[19] W.H. Trzaska,[123] Y. Tsai,[24] Y.-T. Tsai,[191] Z. Tsamalaidze,[74] K.V. Tsang,[191]

2023 JINST 18 P04034

N. Tsverava,[74] S.Z. Tu,[118] S. Tufanli,[35] C. Tull,[132] J. Turner,[58] M. Tuzi,[87] J. Tyler,[124] E. Tyley,[190] M. Tzanov,[136] L. Uboldi,[35] M.A. Uchida,[29] J. Urheim,[93] T. Usher,[191] H. Utaegbulam,[200] S. Uzunyan,[157] M.R. Vagins,[125,24] P. Vahle,[219] S. Valder,[199] G.A. Valdiviesso,[64] E. Valencia,[79] R. Valentim,[209] Z. Vallari,[28] E. Vallazza,[100] J.W.F. Valle,[87] S. Vallecorsa,[35] R. Van Berg,[172] R.G. Van de Water,[135] D. Vanegas Forero,[143] D. Vannerom,[141] F. Varanini,[103] D. Vargas Oliva,[207] G. Varner,[82] S. Vasina,[122] N. Vaughan,[163] K. Vaziri,[68] J. Vega,[46] S. Ventura,[103] A. Verdugo,[39] S. Vergani,[29] M.A. Vermeulen,[154] M. Verzocchi,[68] M. Vicenzi,[98,73] H. Vieira de Souza,[169] C. Vignoli,[77] C. Vilela,[35] B. Viren,[20] A. Vizcaya-Hernandez,[44] T. Vrba,[50] Q. Vuong,[182] T. Wachala,[153] A.V. Waldron,[180] M. Wallbank,[40] T. Walton,[68] H. Wang,[25] J. Wang,[193] L. Wang,[132] M.H.L.S. Wang,[68] X. Wang,[68] Y. Wang,[25] Y. Wang,[196] K. Warburton,[112] D. Warner,[44] M.O. Wascko,[90] D. Waters,[211] A. Watson,[16] K. Wawrowska,[185,199] P. Weatherly,[56] A. Weber,[139,68] M. Weber,[14] H. Wei,[136] A. Weinstein,[112] D. Wenman,[220] M. Wetstein,[112] J. Whilhelmi,[221] A. White,[205] A. White,[221] L.H. Whitehead,[29] D. Whittington,[200] M.J. Wilking,[196] A. Wilkinson,[211] C. Wilkinson,[132] Z. Williams,[205] F. Wilson,[185] R.J. Wilson,[44] W. Wisniewski,[191] J. Wolcott,[208] J. Wolfs,[182] T. Wongjirad,[208] A. Wood,[83] K. Wood,[132] E. Worcester,[20] M. Worcester,[20] M. Wospakrik,[68] K. Wresilo,[29] C. Wret,[182] S. Wu,[149] W. Wu,[68] W. Wu,[24] M. Wurm,[139] J. Wyenberg,[55] Y. Xiao,[24] I. Xiotidis,[90] B. Yaeggy,[40] N. Yahlali,[87] E. Yandel,[27] G. Yang,[196] K. Yang,[164] T. Yang,[68] A. Yankelevich,[24] N. Yershov,[109] K. Yonehara,[68] Y.S. Yoon,[38] T. Young,[156] B. Yu,[20] H. Yu,[20] H. Yu,[197] J. Yu,[205] Y. Yu,[89] W. Yuan,[59] R. Zaki,[223] J. Zalesak,[49] L. Zambelli,[53] B. Zamorano,[75] A. Zani,[101] L. Zazueta,[219] G.P. Zeller,[68] J. Zennamo,[68] K. Zeug,[220] C. Zhang,[20] S. Zhang,[93] Y. Zhang,[176] M. Zhao,[20] E. Zhivun,[20] E.D. Zimmerman,[43] S. Zucchelli,[94,17] J. Zuklin,[49] V. Zutshi[157] and R. Zwaska[68]

[*] *Corresponding author*
[1] *Abilene Christian University, Abilene, TX 79601, U.S.A.*
[2] *University of Albany, SUNY, Albany, NY 12222, U.S.A.*
[3] *University of Amsterdam, NL-1098 XG Amsterdam, The Netherlands*
[4] *Antalya Bilim University, 07190 Döşemealtı/Antalya, Turkey*
[5] *University of Antananarivo, Antananarivo 101, Madagascar*
[6] *Universidad Antonio Nariño, Bogotá, Colombia*
[7] *Argonne National Laboratory, Argonne, IL 60439, U.S.A.*
[8] *University of Arizona, Tucson, AZ 85721, U.S.A.*
[9] *Universidad Nacional de Asunción, San Lorenzo, Paraguay*
[10] *University of Athens, Zografou GR 157 84, Greece*
[11] *Universidad del Atlántico, Barranquilla, Atlántico, Colombia*
[12] *Augustana University, Sioux Falls, SD 57197, U.S.A.*
[13] *University of Basel, CH-4056 Basel, Switzerland*
[14] *University of Bern, CH-3012 Bern, Switzerland*
[15] *Beykent University, Istanbul, Turkey*
[16] *University of Birmingham, Birmingham B15 2TT, United Kingdom*
[17] *Università del Bologna, 40127 Bologna, Italy*
[18] *Boston University, Boston, MA 02215, U.S.A.*
[19] *University of Bristol, Bristol BS8 1TL, United Kingdom*
[20] *Brookhaven National Laboratory, Upton, NY 11973, U.S.A.*
[21] *University of Bucharest, Bucharest, Romania*
[22] *University of California Berkeley, Berkeley, CA 94720, U.S.A.*
[23] *University of California Davis, Davis, CA 95616, U.S.A.*
[24] *University of California Irvine, Irvine, CA 92697, U.S.A.*
[25] *University of California Los Angeles, Los Angeles, CA 90095, U.S.A.*
[26] *University of California Riverside, Riverside CA 92521, U.S.A.*
[27] *University of California Santa Barbara, Santa Barbara, California 93106 U.S.A.*
[28] *California Institute of Technology, Pasadena, CA 91125, U.S.A.*
[29] *University of Cambridge, Cambridge CB3 0HE, United Kingdom*

2023 JINST 18 P04034

[30] *Universidade Estadual de Campinas, Campinas — SP, 13083-970, Brazil*
[31] *Università di Catania, 2 — 95131 Catania, Italy*
[32] *Universidad Católica del Norte, Antofagasta, Chile*
[33] *Centro Brasileiro de Pesquisas Físicas, Rio de Janeiro, RJ 22290-180, Brazil*
[34] *IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France*
[35] *CERN, The European Organization for Nuclear Research, 1211 Meyrin, Switzerland*
[36] *Institute of Particle and Nuclear Physics of the Faculty of Mathematics and Physics of the Charles University, 180 00 Prague 8, Czech Republic*
[37] *University of Chicago, Chicago, IL 60637, U.S.A.*
[38] *Chung-Ang University, Seoul 06974, South Korea*
[39] *CIEMAT, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, E-28040 Madrid, Spain*
[40] *University of Cincinnati, Cincinnati, OH 45221, U.S.A.*
[41] *Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Mexico City, Mexico*
[42] *Universidad de Colima, Colima, Mexico*
[43] *University of Colorado Boulder, Boulder, CO 80309, U.S.A.*
[44] *Colorado State University, Fort Collins, CO 80523, U.S.A.*
[45] *Columbia University, New York, NY 10027, U.S.A.*
[46] *Comisión Nacional de Investigación y Desarrollo Aeroespacial, Lima, Peru*
[47] *Centro de Tecnologia da Informacao Renato Archer, Amarais — Campinas, SP — CEP 13069-901, Brazil*
[48] *Central University of South Bihar, Gaya, 824236, India*
[49] *Institute of Physics, Czech Academy of Sciences, 182 00 Prague 8, Czech Republic*
[50] *Czech Technical University, 115 19 Prague 1, Czech Republic*
[51] *Dakota State University, Madison, SD 57042, U.S.A.*
[52] *University of Dallas, Irving, TX 75062-4736, U.S.A.*
[53] *Laboratoire d'Annecy de Physique des Particules, Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAPP-IN2P3, 74000 Annecy, France*
[54] *Daresbury Laboratory, Cheshire WA4 4AD, United Kingdom*
[55] *Dordt University, 700 7th St NE, Sioux Center, IA 51250, U.S.A.*
[56] *Drexel University, Philadelphia, PA 19104, U.S.A.*
[57] *Duke University, Durham, NC 27708, U.S.A.*
[58] *Durham University, Durham DH1 3LE, United Kingdom*
[59] *University of Edinburgh, Edinburgh EH8 9YL, United Kingdom*
[60] *Universidad EIA, Envigado, Antioquia, Colombia*
[61] *Eötvös Loránd University, 1053 Budapest, Hungary*
[62] *ETH Zurich, Zurich, Switzerland*
[63] *Faculdade de Ciências da Universidade de Lisboa — FCUL, 1749-016 Lisboa, Portugal*
[64] *Universidade Federal de Alfenas, Poços de Caldas — MG, 37715-400, Brazil*
[65] *Universidade Federal de Goias, Goiania, GO 74690-900, Brazil*
[66] *Universidade Federal do ABC, Santo André — SP, 09210-580, Brazil*
[67] *Universidade Federal do Rio de Janeiro, Rio de Janeiro — RJ, 21941-901, Brazil*
[68] *Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.*
[69] *University of Ferrara, Ferrara, Italy*
[70] *University of Florida, Gainesville, FL 32611-8440, U.S.A.*
[71] *Florida State University, Tallahassee, FL, U.S.A.*
[72] *Fluminense Federal University, 9 Icaraí Niterói — RJ, 24220-900, Brazil*
[73] *Università degli Studi di Genova, Genova, Italy*
[74] *Georgian Technical University, Tbilisi, Georgia*
[75] *University of Granada & CAFPE, 18002 Granada, Spain*
[76] *Gran Sasso Science Institute, L'Aquila, Italy*
[77] *Laboratori Nazionali del Gran Sasso, L'Aquila AQ, Italy*
[78] *University Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, 38000 Grenoble, France*

2023 JINST 18 P04034

[79] *Universidad de Guanajuato, Guanajuato, C.P. 37000, Mexico*
[80] *Harish-Chandra Research Institute, Jhunsi, Allahabad 211 019, India*
[81] *Harvard University, Cambridge, MA 02138, U.S.A.*
[82] *University of Hawaii, Honolulu, HI 96822, U.S.A.*
[83] *University of Houston, Houston, TX 77204, U.S.A.*
[84] *University of Hyderabad, Gachibowli, Hyderabad — 500 046, India*
[85] *Idaho State University, Pocatello, ID 83209, U.S.A.*
[86] *Institut de Física d'Altes Energies (IFAE)—Barcelona Institute of Science and Technology (BIST), Barcelona, Spain*
[87] *Instituto de Física Corpuscular, CSIC and Universitat de València, 46980 Paterna, Valencia, Spain*
[88] *Instituto Galego de Física de Altas Enerxías, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain*
[89] *Illinois Institute of Technology, Chicago, IL 60616, U.S.A.*
[90] *Imperial College of Science Technology and Medicine, London SW7 2BZ, United Kingdom*
[91] *Indian Institute of Technology Guwahati, Guwahati, 781 039, India*
[92] *Indian Institute of Technology Hyderabad, Hyderabad, 502285, India*
[93] *Indiana University, Bloomington, IN 47405, U.S.A.*
[94] *Istituto Nazionale di Fisica Nucleare Sezione di Bologna, 40127 Bologna BO, Italy*
[95] *Istituto Nazionale di Fisica Nucleare Sezione di Catania, I-95123 Catania, Italy*
[96] *Istituto Nazionale di Fisica Nucleare Sezione di Ferrara, I-44122 Ferrara, Italy*
[97] *Istituto Nazionale di Fisica Nucleare Laboratori Nazionali di Frascati, Frascati, Roma, Italy*
[98] *Istituto Nazionale di Fisica Nucleare Sezione di Genova, 16146 Genova GE, Italy*
[99] *Istituto Nazionale di Fisica Nucleare Sezione di Lecce, 73100 — Lecce, Italy*
[100] *Istituto Nazionale di Fisica Nucleare Sezione di Milano Bicocca, 3 — I-20126 Milano, Italy*
[101] *Istituto Nazionale di Fisica Nucleare Sezione di Milano, 20133 Milano, Italy*
[102] *Istituto Nazionale di Fisica Nucleare Sezione di Napoli, I-80126 Napoli, Italy*
[103] *Istituto Nazionale di Fisica Nucleare Sezione di Padova, 35131 Padova, Italy*
[104] *Istituto Nazionale di Fisica Nucleare Sezione di Pavia, I-27100 Pavia, Italy*
[105] *Istituto Nazionale di Fisica Nucleare Laboratori Nazionali di Pisa, Pisa PI, Italy*
[106] *Istituto Nazionale di Fisica Nucleare Sezione di Roma, 00185 Roma RM, Italy*
[107] *Istituto Nazionale di Fisica Nucleare Laboratori Nazionali del Sud, 95123 Catania, Italy*
[108] *Universidad Nacional de Ingeniería, Lima 25, Perú*
[109] *Institute for Nuclear Research of the Russian Academy of Sciences, Moscow 117312, Russia*
[110] *University of Insubria, Via Ravasi, 2, 21100 Varese VA, Italy*
[111] *University of Iowa, Iowa City, IA 52242, U.S.A.*
[112] *Iowa State University, Ames, Iowa 50011, U.S.A.*
[113] *Institut de Physique des 2 Infinis de Lyon, 69622 Villeurbanne, France*
[114] *Institute for Research in Fundamental Sciences, Tehran, Iran*
[115] *Instituto Superior Técnico — IST, Universidade de Lisboa, Portugal*
[116] *Instituto Tecnológico de Aeronáutica, Sao Jose dos Campos, Brazil*
[117] *Iwate University, Morioka, Iwate 020-8551, Japan*
[118] *Jackson State University, Jackson, MS 39217, U.S.A.*
[119] *University of Jammu, Jammu-180006, India*
[120] *Jawaharlal Nehru University, New Delhi 110067, India*
[121] *Jeonbuk National University, Jeonrabuk-do 54896, South Korea*
[122] *Joint Institute for Nuclear Research, Dzhelepov Laboratory of Nuclear Problems 6 Joliot-Curie, Dubna, Moscow Region, 141980 RU, Russia*
[123] *University of Jyvaskyla, FI-40014, Finland*
[124] *Kansas State University, Manhattan, KS 66506, U.S.A.*
[125] *Kavli Institute for the Physics and Mathematics of the Universe, Kashiwa, Chiba 277-8583, Japan*
[126] *High Energy Accelerator Research Organization (KEK), Ibaraki, 305-0801, Japan*
[127] *Korea Institute of Science and Technology Information, Daejeon, 34141, South Korea*

2023 JINST 18 P04034

[128] *K L University, Vaddeswaram, Andhra Pradesh 522502, India*
[129] *National Institute of Technology, Kure College, Hiroshima, 737-8506, Japan*
[130] *Taras Shevchenko National University of Kyiv, 01601 Kyiv, Ukraine*
[131] *Lancaster University, Lancaster LA1 4YB, United Kingdom*
[132] *Lawrence Berkeley National Laboratory, Berkeley, CA 94720, U.S.A.*
[133] *Laboratório de Instrumentação e Física Experimental de Partículas, 1649-003 Lisboa and 3004-516 Coimbra, Portugal*
[134] *University of Liverpool, L69 7ZE, Liverpool, United Kingdom*
[135] *Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.*
[136] *Louisiana State University, Baton Rouge, LA 70803, U.S.A.*
[137] *University of Lucknow, Uttar Pradesh 226007, India*
[138] *Madrid Autonoma University and IFT UAM/CSIC, 28049 Madrid, Spain*
[139] *Johannes Gutenberg-Universität Mainz, 55122 Mainz, Germany*
[140] *University of Manchester, Manchester M13 9PL, United Kingdom*
[141] *Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
[142] *Max-Planck-Institut, Munich, 80805, Germany*
[143] *University of Medellín, Medellín, 050026 Colombia*
[144] *University of Michigan, Ann Arbor, MI 48109, U.S.A.*
[145] *Michigan State University, East Lansing, MI 48824, U.S.A.*
[146] *Università del Milano-Bicocca, 20126 Milano, Italy*
[147] *Università degli Studi di Milano, I-20133 Milano, Italy*
[148] *University of Minnesota Duluth, Duluth, MN 55812, U.S.A.*
[149] *University of Minnesota Twin Cities, Minneapolis, MN 55455, U.S.A.*
[150] *University of Mississippi, University, MS 38677 U.S.A.*
[151] *Università degli Studi di Napoli Federico II , 80138 Napoli NA, Italy*
[152] *University of New Mexico, Albuquerque, NM 87131, U.S.A.*
[153] *H. Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, Cracow, Poland*
[154] *Nikhef National Institute of Subatomic Physics, 1098 XG Amsterdam, Netherlands*
[155] *National Institute of Science Education and Research (NISER), Odisha 752050, India*
[156] *University of North Dakota, Grand Forks, ND 58202-8357, U.S.A.*
[157] *Northern Illinois University, DeKalb, IL 60115, U.S.A.*
[158] *Northwestern University, Evanston, Il 60208, U.S.A.*
[159] *University of Notre Dame, Notre Dame, IN 46556, U.S.A.*
[160] *University of Novi Sad, 21102 Novi Sad, Serbia*
[161] *Occidental College, Los Angeles, CA 90041, U.S.A.*
[162] *Ohio State University, Columbus, OH 43210, U.S.A.*
[163] *Oregon State University, Corvallis, OR 97331, U.S.A.*
[164] *University of Oxford, Oxford, OX1 3RH, United Kingdom*
[165] *Pacific Northwest National Laboratory, Richland, WA 99352, U.S.A.*
[166] *Universtà degli Studi di Padova, I-35131 Padova, Italy*
[167] *Panjab University, Chandigarh, 160014 U.T., India*
[168] *Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France*
[169] *Université Paris Cité, CNRS, Astroparticule et Cosmologie, Paris, France*
[170] *University of Parma, 43121 Parma PR, Italy*
[171] *Università degli Studi di Pavia, 27100 Pavia PV, Italy*
[172] *University of Pennsylvania, Philadelphia, PA 19104, U.S.A.*
[173] *Pennsylvania State University, University Park, PA 16802, U.S.A.*
[174] *Physical Research Laboratory, Ahmedabad 380 009, India*
[175] *Università di Pisa, I-56127 Pisa, Italy*
[176] *University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.*
[177] *Pontificia Universidad Católica del Perú, Lima, Perú*
[178] *University of Puerto Rico, Mayaguez 00681, Puerto Rico, U.S.A.*

2023 JINST 18 P04034

[179] *Punjab Agricultural University, Ludhiana 141004, India*
[180] *Queen Mary University of London, London E1 4NS, United Kingdom*
[181] *Radboud University, NL-6525 AJ Nijmegen, Netherlands*
[182] *University of Rochester, Rochester, NY 14627, U.S.A.*
[183] *Royal Holloway College London, TW20 0EX, United Kingdom*
[184] *Rutgers University, Piscataway, NJ, 08854, U.S.A.*
[185] *STFC Rutherford Appleton Laboratory, Didcot OX11 0QX, United Kingdom*
[186] *Università del Salento, 73100 Lecce, Italy*
[187] *San Jose State University, San José, CA 95192-0106, U.S.A.*
[188] *Sapienza University of Rome, 00185 Roma RM, Italy*
[189] *Universidad Sergio Arboleda, 11022 Bogotá, Colombia*
[190] *University of Sheffield, Sheffield S3 7RH, United Kingdom*
[191] *SLAC National Accelerator Laboratory, Menlo Park, CA 94025, U.S.A.*
[192] *University of South Carolina, Columbia, SC 29208, U.S.A.*
[193] *South Dakota School of Mines and Technology, Rapid City, SD 57701, U.S.A.*
[194] *South Dakota State University, Brookings, SD 57007, U.S.A.*
[195] *Southern Methodist University, Dallas, TX 75275, U.S.A.*
[196] *Stony Brook University, SUNY, Stony Brook, NY 11794, U.S.A.*
[197] *Sun Yat-Sen University, Guangzhou, 510275, China*
[198] *Sanford Underground Research Facility, Lead, SD, 57754, U.S.A.*
[199] *University of Sussex, Brighton, BN1 9RH, United Kingdom*
[200] *Syracuse University, Syracuse, NY 13244, U.S.A.*
[201] *Universidade Tecnológica Federal do Paraná, Curitiba, Brazil*
[202] *Tel Aviv University, Tel Aviv-Yafo, Israel*
[203] *Texas A&M University, College Station, Texas 77840, U.S.A.*
[204] *Texas A&M University — Corpus Christi, Corpus Christi, TX 78412, U.S.A.*
[205] *University of Texas at Arlington, Arlington, TX 76019, U.S.A.*
[206] *University of Texas at Austin, Austin, TX 78712, U.S.A.*
[207] *University of Toronto, Toronto, Ontario M5S 1A1, Canada*
[208] *Tufts University, Medford, MA 02155, U.S.A.*
[209] *Universidade Federal de São Paulo, 09913-030, São Paulo, Brazil*
[210] *Ulsan National Institute of Science and Technology, Ulsan 689-798, South Korea*
[211] *University College London, London, WC1E 6BT, United Kingdom*
[212] *Valley City State University, Valley City, ND 58072, U.S.A.*
[213] *Variable Energy Cyclotron Centre, 700 064 West Bengal, India*
[214] *Virginia Tech, Blacksburg, VA 24060, U.S.A.*
[215] *University of Warsaw, 02-093 Warsaw, Poland*
[216] *University of Warwick, Coventry CV4 7AL, United Kingdom*
[217] *Wellesley College, Wellesley, MA 02481, U.S.A.*
[218] *Wichita State University, Wichita, KS 67260, U.S.A.*
[219] *William and Mary, Williamsburg, VA 23187, U.S.A.*
[220] *University of Wisconsin Madison, Madison, WI 53706, U.S.A.*
[221] *Yale University, New Haven, CT 06520, U.S.A.*
[222] *Yerevan Institute for Theoretical Physics and Modeling, Yerevan 0036, Armenia*
[223] *York University, Toronto M3J 1P3, Canada*

2023 JINST 18 P04034