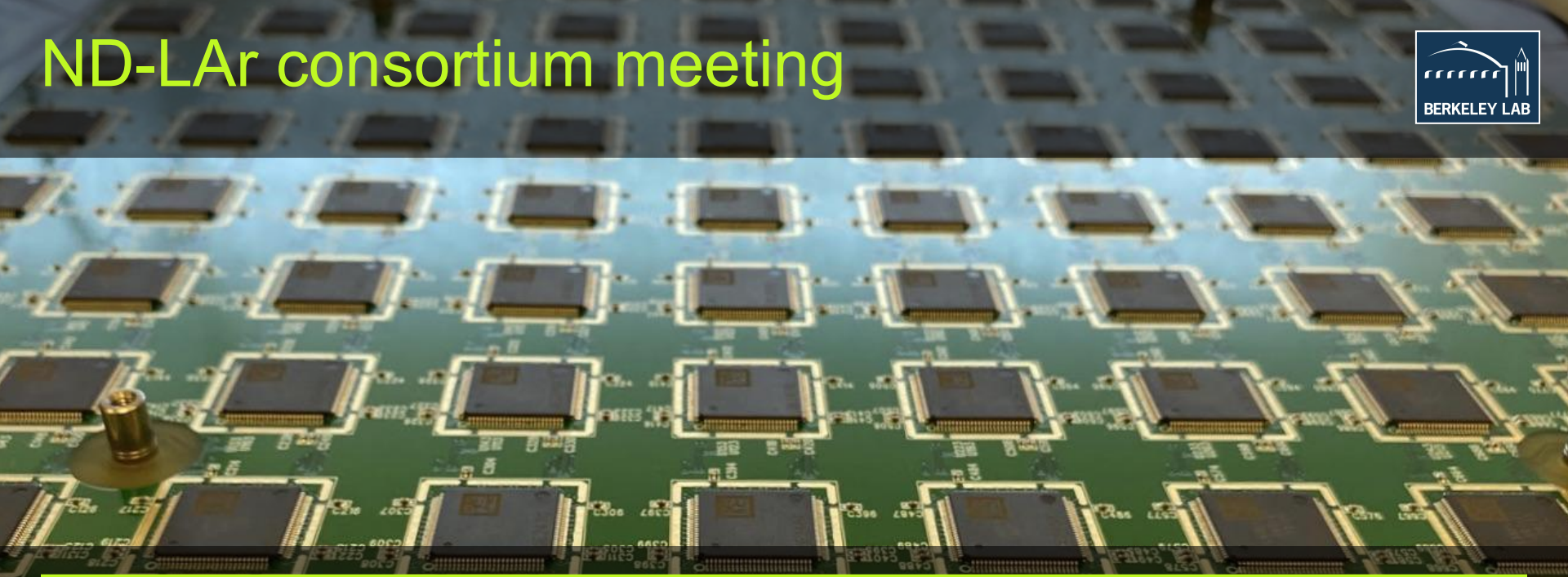
ND-LAr consortium meeting



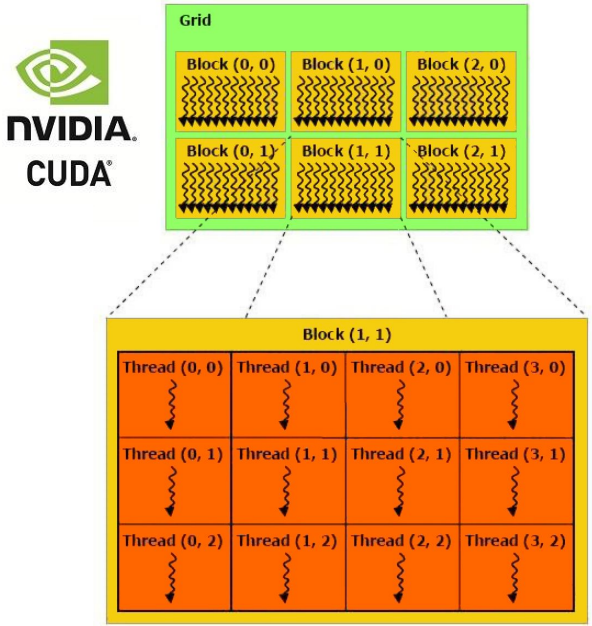
Highly-parallelized simulation of a pixelated LArTPC on a GPU

STEFANO ROBERTO SOLETI
LAWRENCE BERKELEY NATIONAL LABORATORY

13 OCTOBER 2022

Simulation of a pixelated LArTPC

- The **DUNE ND-LAr** detector will be equipped with 14M pixels and detect around 50 neutrino interactions per beam spill, stressing the need for a highly-scalable simulation software.
- We implemented a detector simulation software for pixelated LArTPCs called `larnd-sim`.
- It is implemented as a set of GPU algorithms that use **Numba**, a just-in-time compiler that allows to **speed-up pure Python code** both on CPU and on GPU, using CUDA libraries.
- The **CUDA platform** lets you run your function (the *kernel*) in large number of *threads*, that run in parallel on the GPU and are organized in *blocks.* It comes with a natural C++ extension, but can be used also with other languages (Java, Fortran, Python).
- The advantage is that the CUDA *hides* the specific underlying architecture, and allows to compile the **same code on different GPUs** with automatic scalability.
- The simulations shown here have been performed shown here on the GPU nodes of the **NERSC Cori** supercomputer.



Proposed publication

Highly-parallelized simulation of a pixelated LArTPC on a GPU

The DUNE Collaboration

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 liquid argon 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 project chamber (LArTPC) equipped with light readout and pixelated charge readout. 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 four orders of magnitude speed-up 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 around 10 s on the CPU. The results of the simulation are compared against data from a pixel-readout LArTPC prototype.

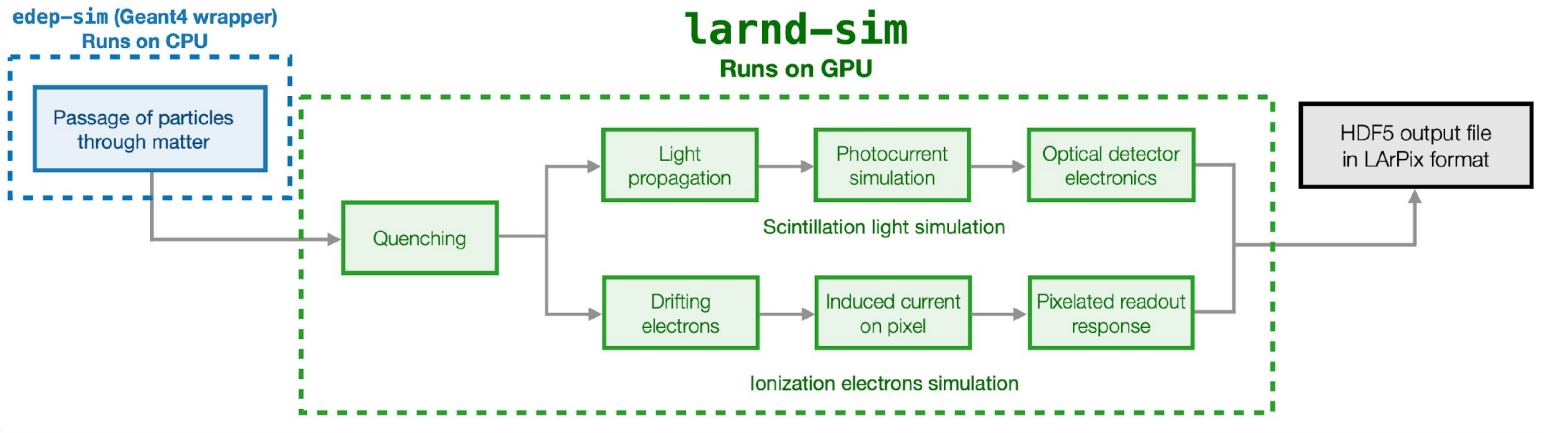
KEYWORDS: Computing, Time projection chambers, Simulation methods and programs

- **Journal targeted**: JINST
- **Main authors**: D. Douglas (MSU), D. Dwyer (LBL), P. Madigan (UC Berkeley, LBL), B. Russell (LBL), S. R. Soleti (LBL), Z. Vallari (Caltech).
- **Draft**: DocDB 26864
- **GitHub repository**
- **Code documentation**
- **Comments sheet**

Paper structure

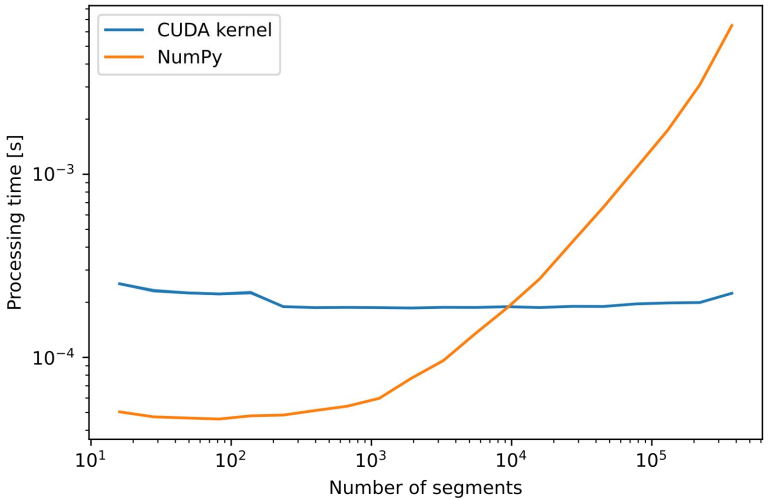
Simulation chain

- The software described in the paper contains **several CUDA kernel functions**, separated into two logical categories: one for the charge simulation and one for the light simulation.
- The functions simulates the detector response, including:
  - the recombination of the electrons with the argon ions
  - the drifting of the electrons towards
  - the induction of electronic signals on the pixel pads and optical detectors
  - the electronics response of the charge and light readouts.



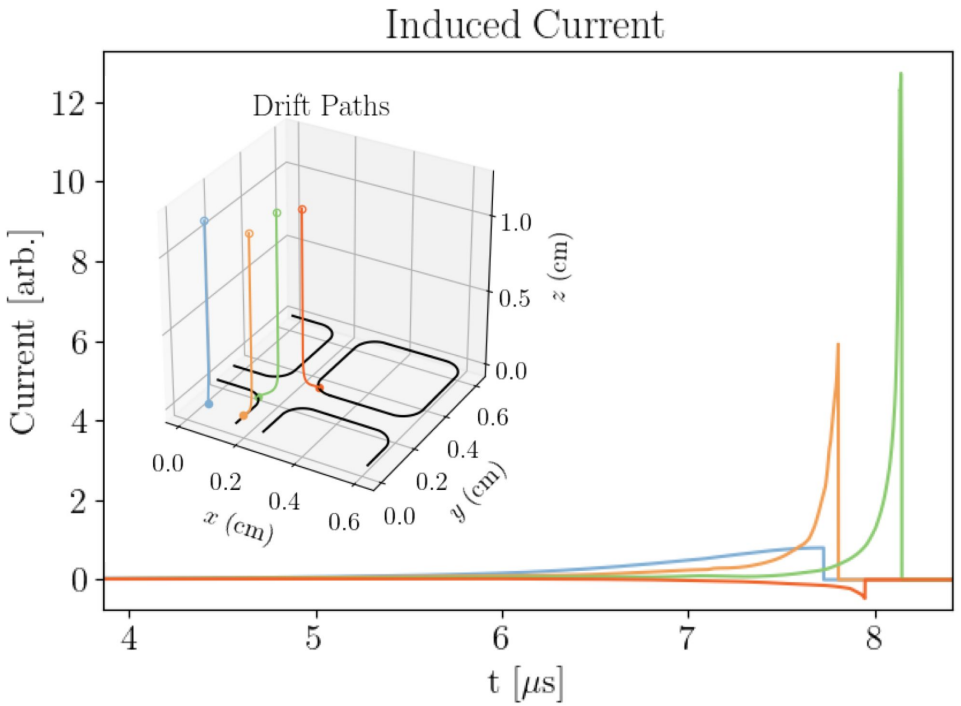
Electron recombination CUDA kernel

- As a first example, in the paper we compare the performance between CPU and GPU for the function that calculates the recombination factor in LAr, given as input an array of deposited energy segments, as generated by edep-sim.
- While the CUDA kernel processing time is initially larger than the NumPy one, the former doesn't scale with the number of input segments, so it starts being the **exponentially faster implementation** with O(10^4) segments.



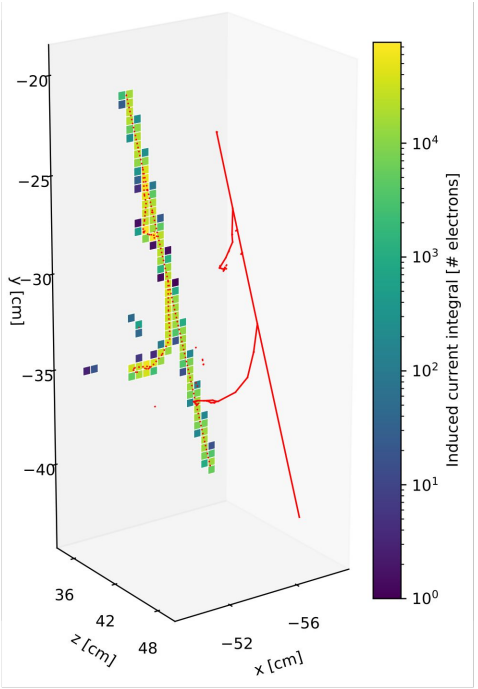
Induced current calculation

- Once we know the number of electrons that effectively reach the anode (after recombination and drifting), we need to **calculate the current induced on each pixel**.
- Traveling through the liquid argon, the **trail of ionization electrons is diffused** longitudinally and transversely w.r.t the drift direction, forming a charge cloud around the original trail.
- We have analytically calculated the field response for an electron starting at 0.5 cm from the anode for several points on the pixel.
- The induced current on a given pixel from this can be calculated by taking the **convolution** of the pre-calculated pixel response model and the charge density of the track segment.



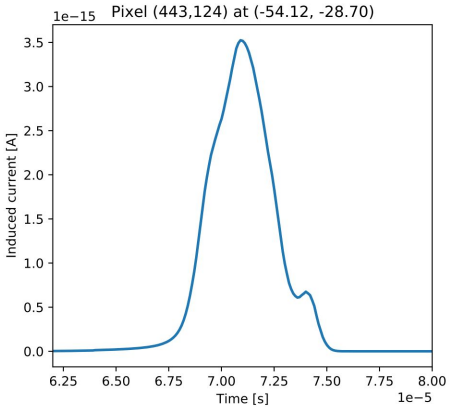
Induced current implementation

3D event display of a simulated cosmic muon

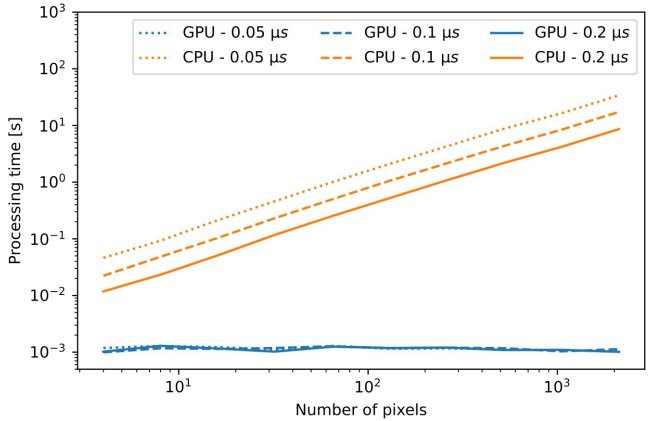


- We implemented this calculation on a **3D CUDA grid**, where each thread calculates the current induced by a single particle on a single pixel at a single instant in time.
- Then, the currents induced on the same pixel are summed in order to obtain the full waveform.
- Comparing the GPU and the CPU implementation shows a **speed-up of four orders of magnitude for O(103) pixels**.

Full waveform on a single pixel

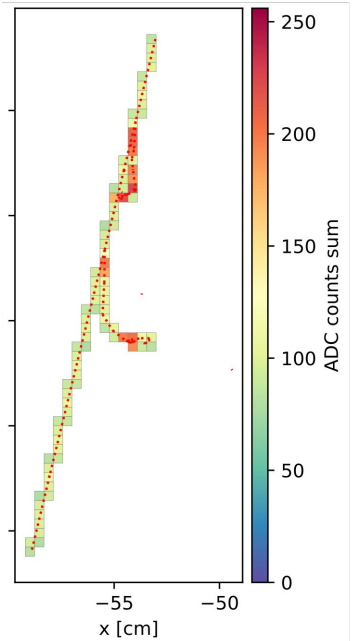


Performance comparison

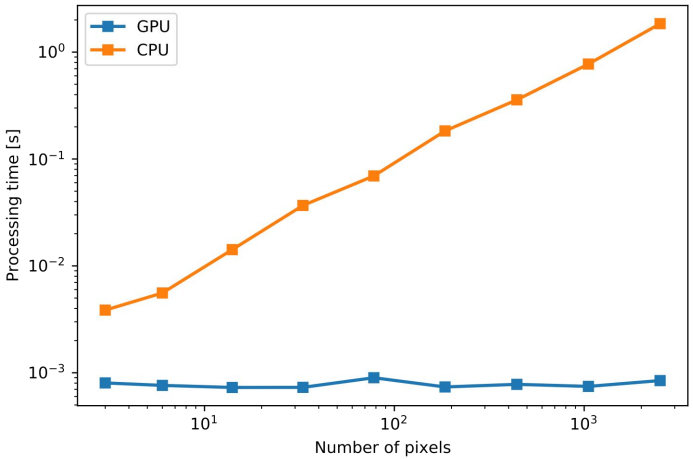


Charge electronics simulation

2D event display of a simulated cosmic muon after digitization



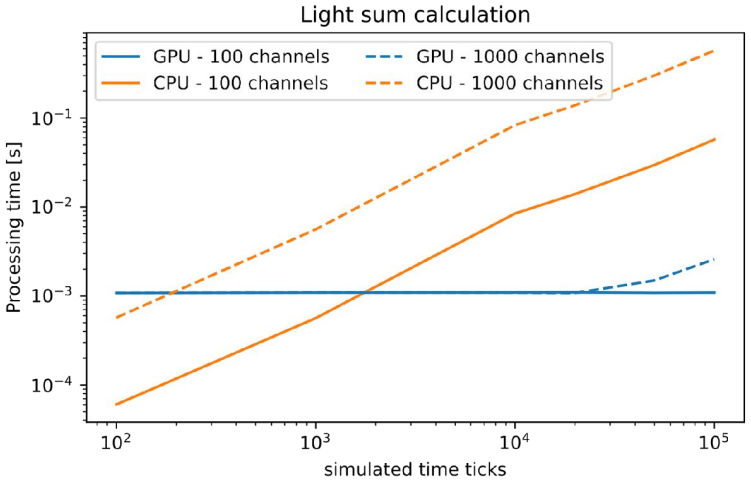
- In the LArPix system the pixel pads are uniquely instrumented by **application-specific integrated circuits** (ASICs).
- The signal on each pixel pad is input to a charge-sensitive amplifier and then digitized through an **8-bit ADC**.
- We perform the simulation of this system with a **1D CUDA grid**, where each thread simulates the electronics response of a single pixel.

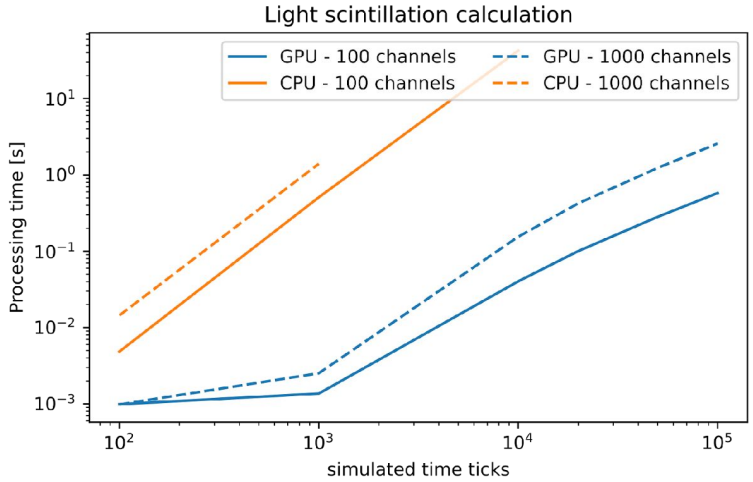


- The GPU implementation is **more than three orders of magnitude faster** with tracks.

Light simulation

- Light propagation and material quantum efficiencies are **pre-tabulated** using a dedicated Geant4 simulation into a look-up table.
- 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.
- After the time profile has been created, the simulation calculates the smearing effect due to the scintillation light emission.

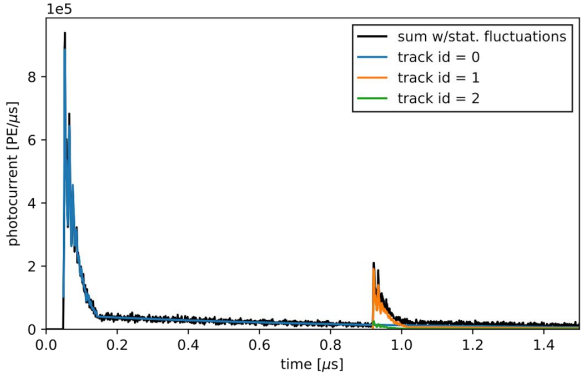




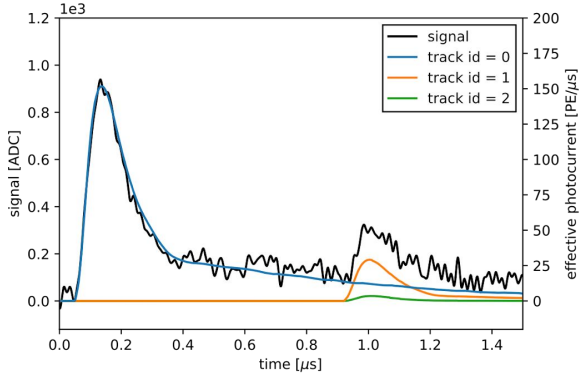
Light electronics simulation

- 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**.
- 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.

Michel decay



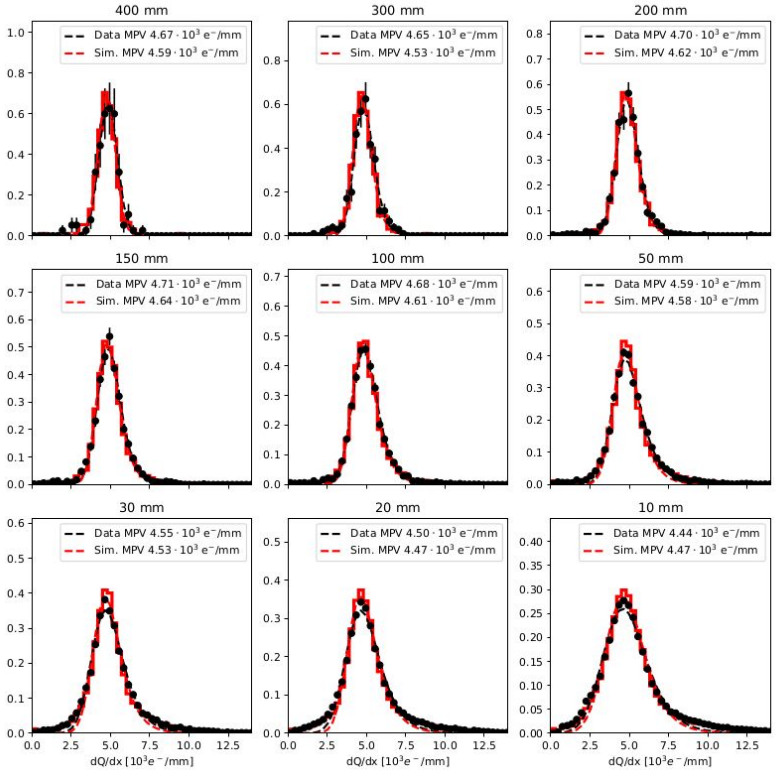
(b) Photocurrent on example photosensor after including LUT and scintillation time smearing.



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

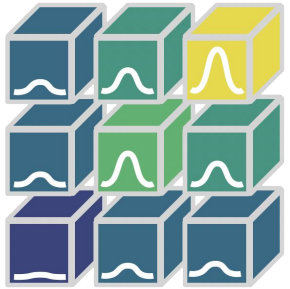
Data-simulation comparison

- The result of the simulation is exported in the same **HDF5 format** used by the LArPix system. This allows a direct comparison of the data with the simulation, using the same analysis scripts.
- A single-phase LArTPC called the **ArgonCube Module-0 Demonstrator** was operated in spring 2021 at the University of Bern.
- Here, we compare the Module-0 data with the a **sample of cosmic rays** simulated with `edep-sim` + `larnd-sim`.
- Both the simulation and the data samples are passed to a reconstruction script, which groups contiguous hits into tracks.
- Then, we subdivide each track into segments of different length (from 10 mm to 400 mm) and we calculate the dQ/dx for each segment.
- The data and the simulation are in **good agreement**.



Conclusions

- We have shown that it is possible to implement the simulation of a pixelated LArTPC using **highly-parallelized GPU algorithms**.
- Running the software on the GPU achieves **three to four orders of magnitude** speedup, both for the charge and the light simulation.
- The software can be used to produce simulations for **several detectors**: we have produced samples for SingleCube, Module-0 and 2x2.
- **dQ/dx distributions** show a good agreement between data and simulation.
- We plan to submit the paper draft to the Publication Board **next week**. Please feel free to add your questions and comments to https://docs.google.com/spreadsheets/d/17IZbbAkMMvT2RLF2YhNE3_oD3t38O5XZrnzaE7cgKMY/edit?usp=sharing. The link will be available also on DocDB 26864.



larnd-sim