

Pre-Learning a Geometry Using Machine Learning to Accelerate High Energy Physics Detector Simulations

E. Kourlitis, W. Hopkins, D. Benjamin

Argonne National Laboratory

The Large Hadron Collider (LHC) [1] at CERN, in Switzerland, has planned a major upgrade to extend its operation for more than a decade, starting from 2027. The upgraded project is called High-Luminosity LHC (HL-LHC) [2] and it will deliver an increase in luminosity by a factor of 5 to 7. Paramount importance to the physics program is the simulation of physics processes. Already, the simulation of the passage of particles through the ATLAS detector of the LHC [3] occupies more than a third of the available computing resources [4]. To maintain a low uncertainties for future measurements, the amount of simulations needed scales with the luminosity. Recent studies predict the computing capacity required to produce adequate simulations to exceed the planned computing budget after 2026 [5]. This still holds even if most of the detector simulation is done with fast simulation methods, such as response parametrization.

For the simulation of the passage of particles through and their interaction with a complex detector apparatus, the software toolkit Geant4 [6] is widely used. Detailed profiling of the toolkit, using as a paradigm the ATLAS detector, indicate that significant time is spent on parts of the software that explore the geometry of the detector volume. The execution time of such methods can take up to 50% of the total simulation time of complex geometries, such as the electromagnetic calorimeters. The main task of the geometry exploration is the calculation of the fly path of a particle instance (i.e. time snapshot) over a simulation step in time. Of particular importance, is the length of the distance from the initial step point to the geometry volume boundary, along the direction of the particle momentum. This length eventually determines the physics process the particle can undertake at the simulation step. Due to the serial implementation of the simulation process this calculation occurs in every simulation step – $O(10^4)$ for the passage of a 1 GeV photon through the end-cap calorimeter of the ATLAS detector.

This LoI introduces an approach to learn a geometry beforehand in order to reduce the computational demand while the actual simulation is being produced. This is achieved by the construction of a map that corresponds Euclidean space-points, described by their position and direction, to ray lengths until the facing boundary of the volume. The map to achieve this task has high dimensional inputs (at least five, three for the position and two for the direction) and should be based on a general enough approach to be applicable to arbitrary complex geometries. For these reasons, Machine Learning (ML) techniques are utilized in order to approximate the function that maps the space-points inputs to the lengths of interest. The use

of ML comes with two significant advantages. First, the portability of the method is assured among current and future computing architectures. Examples of these are (non-)x86 CPUs or accelerator hardware, such as GPUs or TPUs. This is achieved by the usage of industrial ML frameworks that abstract hardware-specific implementation out. Second, the method is efficiently parallelizable. Although the current version of Geant4 is executed in a serial fashion, extensive R&D projects within the simulation community have published studies and plans to extend the High Energy Physics (HEP) simulations in a parallel manner, i.e. simulate the passage and interaction of multiple particles at the same time using a Single Instruction Multiple Data/Thread model [7]. Our ML approach can be integrated in this simulation paradigm and especially benefit by the large parallel computation capabilities of novel accelerator hardware.

In order to train and optimize the ML algorithm the production of a sufficiently large and accurate dataset is needed. This is generated by running the original Geant4 simulation in the particular geometry of interest. First, the geometry is sampled to create space-points with random position and direction. From these points idealistic particles are shot and the distance, along their directions, to the volume boundary is calculated and stored by the application. These idealistic particles do not interact with the material volume, as this information is not needed for the calculation. Thus, the production of the training dataset is computationally much lighter than the complete detector simulation and only takes place once.

These datasets are used to train and optimize the ML algorithm for the distance prediction. In particular, a Deep Neural Network (DNN) with fully connected layers is used for the distance regression task. The inputs describe the position and direction of the particle instance and the output describes its distance to the volume boundary. It is important the predicted distance is not larger than the true distance, as then the application might be forced to propagate the particle outside of the physical volume. In order to enforce this constraint to the DNN training, a custom loss function is used that penalizes over-predictions more severely than under-predictions. A more sophisticated DNN architecture, based on the PointNet++ concept [8], is also investigated. The benefit of the latter lies in the fact that it can directly conduct 3D convolution operations on the unordered and irregular dataset at hand, in order to extract local features in a hierarchical manner. These features can be then interpolated to any point to extract point-wise predictions. The predictions of the ML algorithm are validated by comparing with truth distances calculated by the Geant4 application.

The purpose of the studies described in this LoI is to investigate whether the evaluation time of the ML algorithm scales weaker than the corresponding calculation of Geant4 with increasing geometry complexity. Thus, the potential of the geometry pre-learning approach to accelerate HEP simulations. To demonstrate this an automated pipeline is set up and executed at Argonne computing facilities. First, geometries of increasing complexity are constructed (e.g. nested twisted trapezoids) and space-points are randomly sampled within them. The Geant4 application is evaluating the corresponding distance of interest for each space-point and the output dataset is used to train adequate ML algorithms that predicts the distances for each geometry. A scalable hyperparameter

optimization package, called DeepHyper [9], is utilized to distribute the search for optimal DNNs for each geometry across multiple computing nodes. Finally, for each geometry the inference time of the ML algorithm is compared to the calculation time of the Geant4 application.

Upon successfully determining whether the pre-learned geometry approach can accelerate the whole simulation process, the ML algorithm should be integrated with the Geant4 toolkit. This can be achieved by using the VecGeom plugin [7] which is modular enough to allow various acceleration structures for geometry navigation. In this way our approach would be broad enough to be used for various HEP detectors with arbitrary complexity. This approach is an addition to the HEP community efforts to accelerate the simulation of physics processes and reduce the computing resources required to achievable levels in the coming years.

References

1. O. S. Brüning et al. “LHC Design Report”. CERN Yellow Reports: Monographs (2004). doi: 10.5170/CERN-2004-003-V-1
2. G. Apollinari et al. “High-Luminosity Large Hadron Collider (HL-LHC)”. CERN Yellow Rep. Monogr. 4 (2017), p. 1. doi: 10.23731/CYRM-2017-004
3. The ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. *Journal of Instrumentation* 3 (2008), p. 8003. doi: 10.1088/1748-0221/3/08/S08003
4. The ATLAS collaboration. “Fast simulation methods in ATLAS: from classical to generative models”. 24th International Conference on Computing in High Energy and Nuclear Physics, 2019. <https://indico.cern.ch/event/773049/contributions/3563145/>
5. The ATLAS collaboration. “Computing and Software - Public Results” https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults#Recent_Public_Plots
6. S. Agostinelli et al. “GEANT4: A Simulation Toolkit”. *Nucl. Instrum. Meth. A* 506 (2003), p. 250. doi: 10.1016/S0168-9002(03)01368-8
7. G. Amadio et al. “GeantV: Results from the prototype of concurrent vector particle transport simulation in HEP”. arXiv:2005.00949
8. R. Charles et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. arXiv:1706.02413
9. P. Balaprakash et al. DeepHyper: “Asynchronous Hyperparameter Search for Deep Neural Networks”. In 25th IEEE International Conference on High Performance Computing, Data, and Analytics. IEEE, 2018.