The TrackML high-energy physics tracking challenge on Kaggle

Moritz Kiehn[1,*], *Sabrina* Amrouche[1], *Paolo* Calafiura[2], *Victor* Estrade[3], *Steven* Farrell[2], *Cécile* Germain[3], *Vava* Gligorov[4], *Tobias* Golling[1], *Heather* Gray[2], *Isabelle* Guyon[5,6], *Mikhail* Hushchyn[7,8], *Vincenzo* Innocente[9], *Edward* Moyse[10], *David* Rousseau[11], *Andreas* Salzburger[9], *Andrey* Ustyuzhanin[7,8], *Jean-Roch* Vlimant[12], and *Yetkin* Yilnaz[11]

[1]Département de physique nucléaire et corpusculaire, Université de Genève, Genève, Switzerland
[2]Physics Division, Lawrence Berkeley National Laboratory and University of California, Berkeley CA, USA
[3]LRI/TAU, Université Paris-Sud/INRIA/CNRS, Université Paris-Saclay, Gif-sur-Yvette, France
[4]LPNHE, Sorbonne Université, Paris Diderot Sorbonne Paris Cité, CNRS/IN2P3, Paris, France
[5]UPSud/INRIA, Université Paris-Saclay, Orsay, France
[6]ChaLearn, California, USA
[7]National Research University Higher School of Economics, Moscow, Russia
[8]Yandex School of Data Analysis, Moscow, Russia
[9]CERN, Geneva, Switzerland
[10]Department of Physics, University of Massachusetts, Amherst MA, U.S.A.
[11]LAL, Université Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France
[12]California Institute of Technology, Pasadena CA, USA

Abstract. The High-Luminosity LHC (HL-LHC) is expected to reach unprecedented collision intensities, which in turn will greatly increase the complexity of tracking within the event reconstruction. To reach out to computer science specialists, a tracking machine learning challenge (TrackML) was set up on Kaggle by a team of ATLAS, CMS, and LHCb physicists tracking experts and computer scientists building on the experience of the successful Higgs Machine Learning challenge in 2014. A training dataset based on a simulation of a generic HL-LHC experiment tracker has been created, listing for each event the measured 3D points, and the list of 3D points associated to a true track.The participants to the challenge should find the tracks in the test dataset, which means building the list of 3D points belonging to each track.The emphasis is to expose innovative approaches, rather than hyper-optimising known approaches. A metric reflecting the accuracy of a model at finding the proper associations that matter most to physics analysis will allow to select good candidates to augment or replace existing algorithms.

*e-mail: moritz.kiehn@unige.ch

1 Introduction

Successful particles physics measurements are dependent on an efficient and performant reconstruction of the physics objects from the detector measurements. As a community we are always striving to run at the edge of the available detector and computation capabilities to extract the maximum amount of information from the experiments. Tracking detectors are at the core of most collider experiments and track reconstruction is one of the crucial tasks in every reconstruction chain.

Track reconstruction at its heart is a combinatorial problem, i.e. to find the measurements that originate from the same initial particle from a set of possible combinations. The created particles have a wide range of possible properties, especially different creation vertices and momenta, and their particle trajectories have non-deterministic contributions from material interactions. In combination with inhomogeneous magnetic fields and detector inefficiencies, this leads to the possibility of confusion, fake tracks, etc. . All of these effects depend strongly on the density of the measurements and thus on the collision pile-up. Consequently, with the upcoming upgrade to the High-Luminosity Large Hadron Collider (HL-LHC) the track reconstruction complexity will increase significantly.

Existing solutions usually rely on well-established algorithms based on track following and combinatorial Kalman filters [1, 2] that are often implemented specifically for each experiment. While these type of algorithms have proven to be quite powerful in the past, they do not scale favorably and the question arises whether new algorithms and different approaches exists that might be better suited to handle the conditions at the HL-LHC.

The TrackML project uses the format of a machine learning challenge in order to expose the tracking problem to non-physicist, e.g. data scientists, computer scientist, or algorithm experts, with the aim to find new types of algorithms and approaches. The TrackML challenge [3] is split into two phases: the accuracy phase [4] that considers only the accuracy of proposed algorithms and the throughput phase [5] that considers their accuracy and computational performance. The accuracy phase ran from May 2018 until August 2018 on the Kaggle platform [6].

2 Setup

To run a machine learning challenge a few components are required: a well-defined problem, a dataset that contains the relevant features, a scoring metric to rank different solutions, and a platform to run the challenge. The problem design has to be reduced as much as possible to be as accessible as possible to a lay-person. At the same time it also has to retain the necessary complexity inherent in the tracking challenge at HL-LHC experiments. In this section, these components will be presented and the chosen configuration and the motivation for particular choices are discussed.

2.1 TrackML detector

The basis of the challenge is a realistic detector model to simulate measured particle hits similar to what is expected for an HL-LHC experiment. Instead of using the proposed upgrade tracker designs of either ATLAS or CMS we opted to design a generic tracker design that takes the key concepts from the existing proposals. This avoids issues of private collaboration information or artefacts and allows a challenge that is more separated from the particular design choices made by each experiment.

Both ATLAS and CMS upgrade tracker designs [7, 8] are based on large surface all-silicon detectors with an extended η coverage. While both experiments are expected to use an optimized geometry with barrels, disks, and inclined sections, here a classical design is chosen. In the central regions, a barrel-like geometry is used, and in the forward regions, a disk-like geometry is employed.

The full detector geometry is shown in figure 1. The detector is split into three separate sub-detectors that differ in spatial resolution and material budget. The inner-most sub-detector is a pixel detector with a spatial segmentation of 50 µm × 50 µm. Further out two different strip detectors with short 80 µm × 1200 µm and long strips 0.12 mm × 10.8 mm are placed. Each detector is build up from realistic module geometries with placement and overlap chosen to yield a hermetic coverage up to $\eta = 3$.

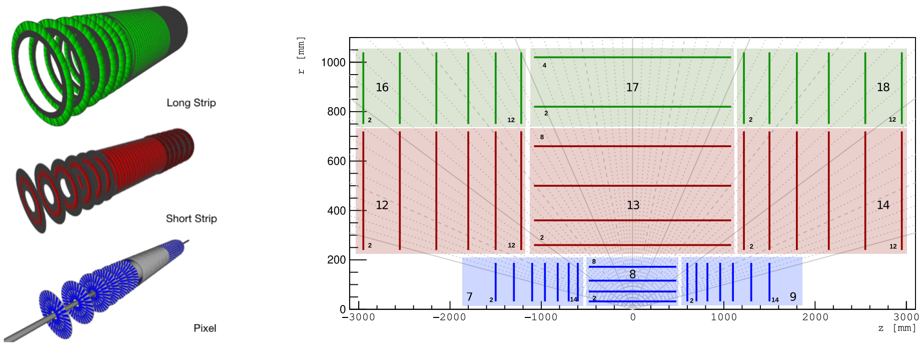

Figure 1. Detector layout for the virtual TrackML detector. On the left the three major sub-detectors, pixel, short strips, and long strips, are shown separately. On the right, a schematic of the full layout and its coverage along the radial and longitudinal dimensions as well as in the η direction is shown. The different colors represent the different sub-detectors while the marked numbers are the internal volume and layer identifiers.

2.2 Fast simulation

The particle content of the collisions is generated using the Pythia 8 event generator [9]. A hard QCD interaction that generates a $t\bar{t}$-pair is used as the signal. An additional 200 soft QCD interactions are overlayed to simulate the expected pile-up conditions at the HL-LHC. The interaction vertices are spread out over a luminous region with a width of 5.5 mm along the beam axis.

Charged particles are propagated through the detector using a fast detector simulation based on the ACTS software [10, 11]. An inhomogeneous magnetic field similar to the one in the ATLAS experiment is used and material interactions, i.e. multiple scattering, energy loss, or hadronic interactions, are simulated using parametric models. Only tracks with a transverse momentum above 150 MeV are propagated. Tracks below this momentum threshold are typically not considered by the HL-LHC experiments and the minimum momentum threshold for reconstruction is often even higher. Inefficient sensors and additional hits from noise or particles below the threshold are also considered.

2.3 Dataset format

Within the HEP community ROOT [12] is the default data format of choice for a large variety of applications. While established and widely used within HEP it is virtually unknown in the wider data science field. To reduce the entrance barrier to the challenge and allow a wide acceptance a different format had to be used. Text files based on comma-separated values (CSV) were chosen as it is a well-established data exchange format within in the larger data science community.

Particle physics events contain a multitude of different types of information that are usually represented in a nested structures of variable length. Here, we aim to provide the data in a flattened structure to avoid the necessity for specialised tools or formats. Since events are statistically independent, data is stored separately for each event, organized by a numerical event identifier. For each event in the training dataset the following four files are provided:

Hits: The hits file provides the simulated hit information that are the core input for the challenge. Each entry corresponds to one hit generated by one particle in a detector module. Hits are identified by an event-unique numerical identifier. The position of the hit in the global coordinate system (x, y, z) as well as its origin in the detector hierarchy, i.e. volume, layer, and module identifier, is given.

Cells: Each hit originates from one or more active detector cells, e.g. clusters of active pixels in the innermost sub-detectors. The cells file contains this information to enable participants to extract additional information, e.g. directional information from the cluster shape. Each entry contains the hit identifier to which the cell contributes too, information on the active channels, and possible charge measurement.

Particles truth: This file contains the generated particle truth information. Each entry represents one generated, charged, final state particle and is identified by a event-unique numerical particle identifier. The numerical identifier can be further decoded to provide detailed generator information, e.g. a vertex identifier. In addition, the particle momentum (p_x, p_y, p_z), its vertex position (v_x, v_y, v_z), its charge, and the number of hits generated by this particle are available. The information on the number of generated hits can in principle be derived from the other files, but is provided here for convenience.

Hit truth: This file contains the mapping between particles and hits and the true particle state at every hit. Each entry contains the hit identifier from the hits file. Depending on the subdetector the true particle state is given either before or after the interaction with the detector material.

Training and test data-sets only differ by the type of files made available. In the test data set only the hits and cells files are available, as would be the case in the real detector reconstruction. Internally, all events were generated in exactly the same way.

Solutions provided by the participants were also provided in CSV format. Due to technical requirements by the Kaggle platform a single file had to be provided for all events in the test dataset. Each entry therefore had to comprise three values: the event identifier as provided, the event-unique hit identifier as provided, and a track identifier generated by the participant. Hits reconstructed to belong to the same track must have the same track identifier.

The chosen file format allowed the data to be read directly by a variety of different analysis tools, programming languages, and processing frameworks, e.g. [13] or [14].

2.4 Scoring

Reconstruction performance and its impact on the physics measurement is measured by a multitude of metrics. High efficiency and low ghost or fake rate have a high priority. The achievable momentum and vertex resolution also plays an important role and additional requirements might be present for particular event topologies. However, for the challenge a single scoring function has to be selected that allows us to rank different solutions based on their expected physics performance.

As a simplification, hits have to be uniquely associated to a single reconstructed track. Only reconstructable tracks with four hits or more are considered. As a minimal requirement a reconstructed track is scored only if more than 50 % of its hits originate from the same truth particle. Using the truth information from the simulation, all hits originating from this majority particle are selected for each track. Each hit has an per-hit scoring weight and the total score for a given solution is given by the following multiple sum

$$S \sim \sum_{\{\text{events}\}} \sum_{\{\text{tracks}\}} \begin{cases} 0 & \#\text{good hits} < 50\,\%, \#\text{hits} < 3 \\ \sum_{\{\text{good hits}\}} w_i & \text{else} \end{cases} \tag{1}$$

$$S_{perfect} = 1 \tag{2}$$

$$w_i = w_i\,(\text{hit order, particle } p_\perp)\ . \tag{3}$$

With this definition no penalty for incorrect hits, i.e. hits associated to a track that do not originate from its majority particle, is necessary, since a wrongly associated hit will automatically reduce the score for the track it should have been associated to.

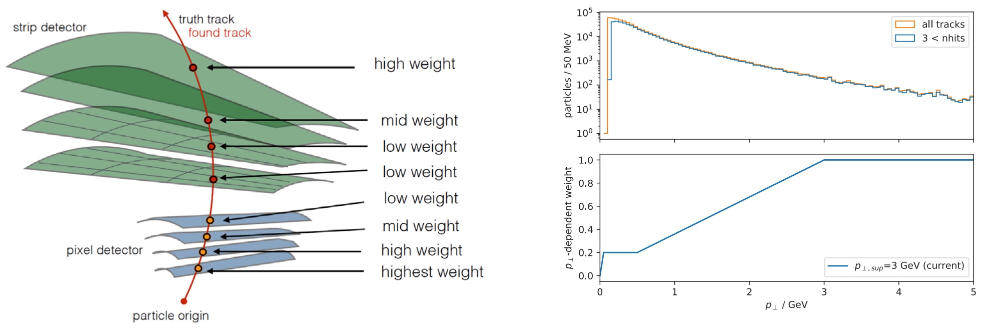

Figure 2. On the left: illustration of the order-dependent hit weight. On the right: the simulated $p_\perp$ spectrum and the $p_\perp$-dependent hit weight.

The contributions to the per-hit weight are illustrated in figure 2. The order-dependent weight penalizes missed hits at the inner and outer-most part of the detector. Missing an hit on the inner-most layer will strongly influence the vertex resolution, while a missed hit on the outer layers reduces the lever arm for the momentum measurement and thus the momentum resolution. The $p_\perp$-dependent part favors reconstructing high-momentum tracks over low-momentum tracks without excluding either region completely.

The overall score is normalized such that a random solution scores zero and a score of one is a perfect reconstruction of all events in the dataset.

2.5 The Kaggle platform

The accuracy phase uses the Kaggle platform [4, 6]. This platform hosts the dataset and provides the scoring and the leader-board for the participants. Participants can download the training and test dataset, train on the former and prepare a solution for the latter, and upload the solution.

While a variety of solution metrics already existed, the scoring metric discussed in section 2.4 was not one of them. It was implemented by Kaggle on their platform specifically for this challenge. For the test dataset, 125 simulated events were selected. Participants had to reconstruct all events on their own machines and upload their solution to Kaggle where the score was then computed.

3 Experience

The accuracy phase started in May 2018 and ran until August 2018. Over the course of the challenge more than 650 participants, either individuals or groups, participated in the challenge and submitted solutions. The score progressing for the 100 best teams is shown in figure 3 and clearly shows the continuous improvements by the participants during the challenge. The lowest score of ~0.2 corresponds to the score generated by an example solution provided by the organizers.

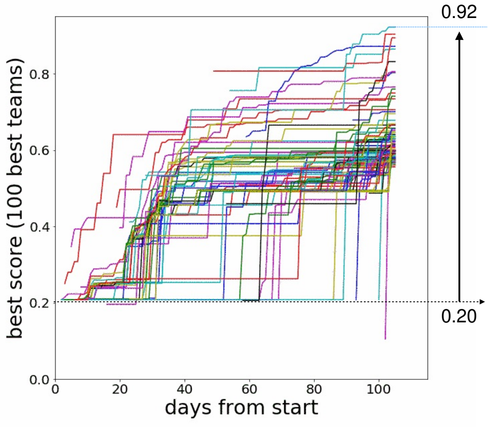

Figure 3. Progression of the submission score for the best 100 teams over the course of the challenge. The lowest score of 0.20 corresponds to the score generated by the starting kit. The highest score of 0.92 is the winner score by the “Top Quarks“ team.

The winning solution at the end of the challenge (Top Quarks) had a score of 0.921, followed by 0.903 for the second place (outrunner), and the third place with 0.893 (Sergey Gorbunov). The chosen scoring metric revealed clear differences in algorithm accuracy at the closing time of the competition. From figure 3 it is not evident that a plateau in the solutions had been reached. This suggests that a longer competition duration could have potentially lead to even better results.

4 Outlook

With the accuracy phase closed, we saw a successful first phase of this challenge. Thanks to the great number of participants a set of new interesting solutions is now available. We are currently conducting a detailed analysis of the best-performing solutions to see if and how they can be applied to existing and future experiments.

For the throughput phase both the accuracy and the run time need to be measured. For this the throughput phase is hosted on the Codalab platform [5]. Here, participants publish not the output of their algorithm but the software itself. It is then run in a consistent environment to measure its runtime in addition to its accuracy.

In the future, the challenge dataset is expected to be published as open data, e.g. on [15]. We expect such a dataset to be useful as a common reference for algorithmic research, software validation, and to allow future work after the challenge has finished.

Acknowledgement

The team would like to thank CERN for allowing the use of the dataset, and Kaggle for hosting it. We are very grateful to our generous sponsors without which the challenges would not have been possible. Platinum sponsors: Kaggle, Nvidia, and Université de Genève. Gold sponsors: Chalearn and DataIA. Silver sponsors: CERN Openlab, Paris-Saclay CDS, INRIA, ERC mPP, ERC RECEPT, Common Ground, Université Paris-Sud, INQNET, Fermilab, and pyTorch. TG acknowledges the support of the Swiss National Science Foundation under the grant 200020_181984. This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 724777 "RECEPT", No 772369 "mPP" and No 654168 "AIDA-2020".

References

[1] R. Frühwirth, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **262**, 444 (1987)

[2] P. Billoir, S. Qian, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **294**, 219 (1990)

[3] D. Rousseau, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon et al., *TrackML Challenge*, https://sites.google.com/site/trackmlparticle/home (2018)

[4] D. Rousseau, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon et al., *Accuracy phase of the TrackML Challenge on Kaggle* (2018)

[5] D. Rousseau, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon et al., *Throughput phase of the TrackML Challenge on Codalab*, https://competitions.codalab.org/competitions/20112 (2018)

[6] Kaggle Inc., *Kaggle: Your Home for Data Science*, https://www.kaggle.com/

[7] A. Collaboration, Tech. Rep. ATLAS-TDR-030, CERN, Geneva (2017)

[8] CMS Collaboration, Tech. Rep. CMS-TDR-014, CERN, Geneva (2017)

[9] T. Sjöstrand, S. Mrenna, P. Skands, Computer Physics Communications **178**, 852 (2008)

[10] C. Gumpert, A. Salzburger, M. Kiehn, J. Hrdinka, N. Calace, A. Collaboration, J. Phys.: Conf. Ser. **898**, 042011 (2017)

[11] N. Calace, C. Gumpert, J. Hdrinka, M. Kiehn, A. Salzburger, *A Common Tracking Software (Acts)*

[12] R. Brun, F. Rademakers, others, *ROOT — an object oriented data analysis framework*

[13] W. McKinney, others, *Pandas: Python Data Analysis Library*

[14] E. Jones, T. Oliphant, P.P. Peterson, others, *SciPy: Open source scientific tools for Python* (2001)

[15] *CERN Open Data Portal*, http://opendata.cern.ch/