Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC

Sylvain Caillou[1,*], *Christophe* Collard[1], *Charline* Rougier[1], *Jan* Stark[1], *Heberth* Torres[1], and *Alexis* Vallier[1]

[1]Laboratoire des 2 Infinis - Toulouse (L2IT-IN2P3), Université de Toulouse, CNRS, Université Toulouse III - Paul Sabatier, Toulouse, France

Abstract. Data from the LHC detectors are not easily represented using regular data structures. These detectors are comprised of several species of subdetectors and therefore produce heterogeneous data. LHC detectors are highly granular by design so that nearby particles may be distinguished. As a consequence, LHC data are sparse, in that many detector channels are not active during a given collision event. Graphs offer a flexible and efficient alternative to rectilinear data structures for representing LHC data. Accordingly, graph-based machine learning algorithms are becoming increasingly popular for a large number of LHC physics tasks [1, 2]. This popularity, and the corresponding potential for substantial increase in physics output, are illustrated on the cover of a recent issue [3] of the CERN Courier magazine. The graphs used in almost all practical applications at the LHC so far are homogeneous, i.e. each node is assigned the same features, and each edge is assigned the same features. In other words, the power of graphs to represent sparse data has been exploited in applications for the LHC, but the potential of graphs to represent heterogeneous data has not. The pink graph on the cover of the CERN Courier [3] can be seen as an illustration of this limitation: all nodes are pink, regardless of their position in the detector. We present novel fully-heterogeneous GNN designs and apply them to simulated data from a tracking detector that resembles the trackers that will be used at the HL-LHC. It contains a pixel subsystem that provides 3D hits and a strip subsystem that provides 2D hits. We present a new design which solves the degraded performance observed in the strip detector in the first GNN-based tracking studies presented by the ATLAS Collaboration [4].

1 Introduction

During the upcoming High Luminosity phase [5] of the Large Hadron Collider [6] (HL-LHC) the peak instantaneous luminosity of the accelerator will be increased to an unprecedented 7.5 times the nominal luminosity. The bunch crossing time (25 ns) will remain unchanged: higher instantaneous luminosity will be achieved by increasing the number of protons per bunch. This leads to an increase of the average number of inelastic proton-proton collisions per bunch crossing (pile-up) expected to reach 200 during the HL-LHC. As a result, the ATLAS experiment [7] will be implementing significant upgrades to the detector and to the

*e-mail: sylvain.caillou@l2it.in2p3.fr

data acquisition system in order to cope with the increased data rates and pile-up, including construction of a new silicon-only Inner Tracker with higher granularity, ITk [8].

The HL-LHC will lead to a steep increase in computing and software resources needed to process and analyse the expected deluge of data. Tracking is one of the most important parts of event reconstruction. Hits left by a particle in a given collision event have to be associated together to build a track candidates. During HL-LHC, current algorithms based on a Combinatoric Kalman Filter (CKF) will not be able to cope with the complexity and rate of the data recorded. Solutions based on geometric learning methods are explored, especially Graph Neural Network (GNN)-based algorithms. A first track reconstruction study on ATLAS HL-LHC conditions simulated data in ITk has been presented by the ATLAS Collaboration [4]. In this study the reconstruction has been processed via the complete GNN-based pipeline, so-called GNN4ITk [4] [9], as shown in Figure 1.

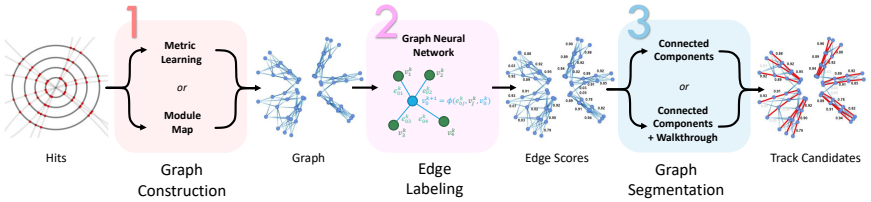
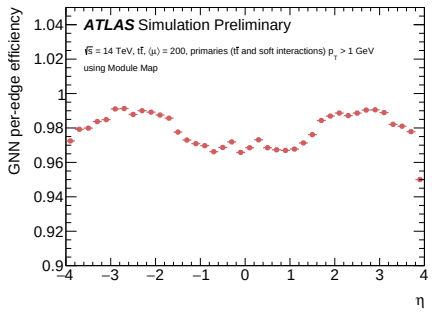


Figure 1: Schematic overview of the GNN-based track finding pipeline (GNN4ITk).

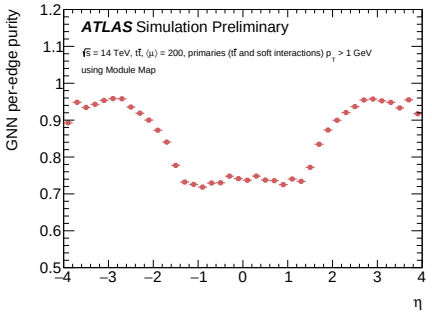
In the first stage of the pipeline, the graph construction, a collision event is represented as a graph. Nodes represent Space Point of hits in the detector and edges are possible connections between nodes. True edges are connections between two successive hits from the same charged particle. In the second stage, a GNN is trained to learn deep geometric patterns of the particles tracks and classifies edges between true and fake edges. Finally, in the third stage, a post-processing algorithm takes the scored graph as input to build track candidates.

2 Problem statement

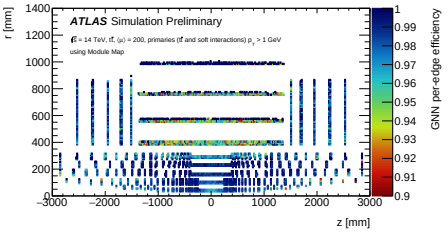
In the results presented in [4] the input events were simulated $pp \rightarrow t\bar{t}$ events where at least one of the top quarks decays leptonically, produced at $\sqrt{s} = 14$ TeV with a pile-up of $\langle\mu\rangle = 200$. The training dataset contains 400 events represented as graphs constructed with a *module map* (a map of possible connections between modules in ITk, [4]) built from 90k events. The node features were the euclidian geometric coordinates of the reconstructed Space Point of the hit in polar coordinates r, ϕ , z. The edge features were preprocessed features computed from the features of the source and destination nodes of the edge Δr, $\Delta\phi$, Δz, $\Delta\eta$ (where η is the pseudo-rapidity). The training used 600k iterations. The GNN trained model was evaluated on a test dataset of 100 graphs disjoint from the train dataset. The edge classification performance of the model was evaluated by two metrics: the edge-wise efficiency and the edge-wise purity. The efficiency is defined as the number of true edges passing a threshold on the edges scores divided by the number of the true edges. The purity is defined as the number of true edges passing a given threshold on the edges scores divide the number of true edges passing the threshold plus the number of fake edges passing the threshold.



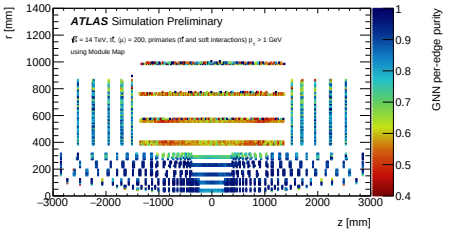
(a) GNN per-edge efficiency vs η



(b) GNN per-edge purity vs η



(c) GNN per-edge efficiency in (r, z) plane



(d) GNN per-edge purity in (r, z) plane

Figure 2: GNN per-edge classification performance (efficiency and purity) along η and in the (z,r) plane from [4]. The (z, r) position of an edge is defined by the one of its source node.

The results are shown in figure 2. For a threshold on the edge score of 0.5, a high global efficiency of ~98% and a global purity ~85% are obtained. An important drop of purity is observed in the central region which comes more specifically from the barrel strip region where the purity goes down to ~50%. In order to understand the reason of the degraded performance of the GNN in this specific region it is necessary to understand how data are reconstructed in ITk.

As shown in schematic view in figure 3, ITk is composed of two sub-systems which are based on two different technologies: The Pixel detector and the Strip Detector. The Pixel detector is a set of more than five billion silicon sensors (pixels), most of which have a width of 50 x 50 μm^2 and the rest 25 x 100 μm^2. The Strip detector comprises almost 100 million long and skinny silicon sensors (75.5 μm x 24.1 or 48.2 mm). Because these technologies are different the data in ITk are heterogeneous by nature. In the Pixel detector, Space Points of the hits are reconstructed by clustering the pixels which have been fired by the particle. Pixels have a small size (50 μm) and it led to a high spatial resolution O(15 μm). In the Strip detector, Space Points are reconstructed from the two clusters from the two sides of the strip modules which have been fired by the particle. As shown in Figure 3, low p_{T} particles can be significantly deviated while traveling inside one strip module. It is Impossible to exactly reconstruct space point position without knowing curvature of the track. The straight line approximation used in the ATLAS space point reconstruction leads to poor resolution (O(1 cm) at p_{T} = 1 GeV) in the z-axis which is the main reason of the GNN poor performance in the strip barrel region.

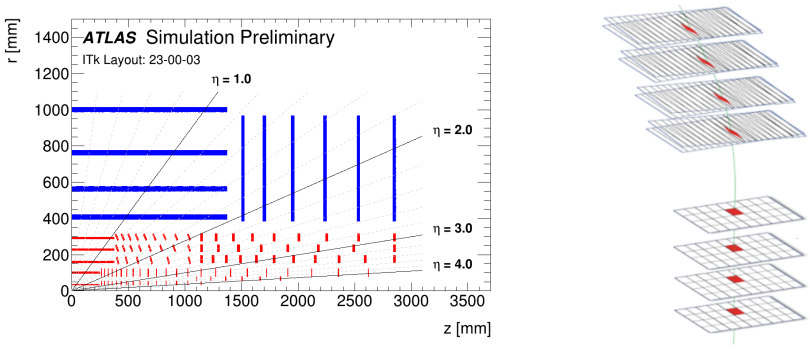


Figure 3: Schematic view of the ITk detector [10] with its two subsystems, the pixel detector (in red) and the strip detector (in blue). A schematic view of a particle traveling through these two subdetectors can be seen on the right.

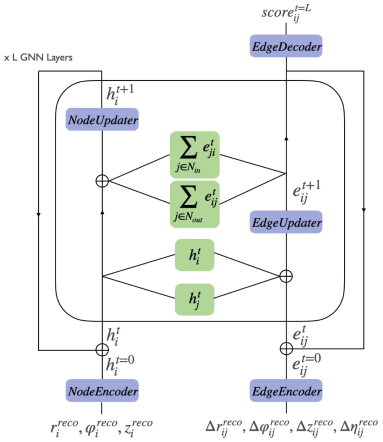


Figure 4: GNN model design with homogeneous data in ITk used in [4]

3 Methods

The GNN model used in [4], described in figure 4, has an encode-process-decode architecture.. At the encode stage, nodes and edges euclidian geometric features are projected with two encoders (one encoder for the nodes and one encoder for the edges), typically Multi Layer Perceptrons (MLPs), into their embeddings representations into $\mathcal{D}$-dimensional latent spaces.

At the process stage an Interaction Network [11] layer is applied iteratively. The Interaction Network proceed in two steps. First, each edge is updated as the output of a MLP (i.e. called the *EdgeUpdater*) taking as input the concatenation of the current edge latent, the source node latent and the destination node latent. Each node is then updated as the output of a MLP (i.e. called the *NodeUpdater*) taking as input the concatenation of the current node latent with a separate aggregation of incoming and outcoming edges latent. Each iteration of the Interaction Network layer propagates information through the graph via the so called message passing mechanism allowing the model to capture deep geometric patterns of the particle tracks. The number of iteration L is called the level of message passing. At the de-

code stage each edge latent is projected from the latent space to an edge classification score (a scalar) giving the pseudo-probability that the edge is a segment of a target particle track (connect two successive hits of the same target particle).

The Space Point reconstruction in the strip is based on the two clusters on both side of the strip. The key idea to improve the performance of the model is to give as node features in the strip barrel region not only the reconstructed space point but also the two clusters coordinates: $\left[r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{c1}, \varphi_i^{c1}, z_i^{c1}, \eta_i^{c1}, r_i^{c2}, \varphi_i^{c2}, z_i^{c2}, \eta_i^{c2}\right]$. The GNN combines space points into tracks, i.e. has access to curvature. The idea is that the GNN model would be able to learn a better latent representation of the Space Point, leading to a better classification of edges in this region. In the rest of the detector where there is a high spatial reconstruction it is given as input to the model only the reconstructed space point There is two possible designs to handle this heterogeneous data.

The first one is to use Heteregenous GNN as shown in figure 5a. We can imagine several level of heterogeneity with a MLP dedicated by region at the encode stage but also at the update stage and at the decode stage. The advantage of this approach is that the MLPs are specialized by heterogeneous regions. A disadvantage is that it will become harder to synchronize speed of training and to share common latent representation between heteregenous regions.

An other approach is to handle heteregenous data with a homogeneous GNN model as shown in Figure 5b. In the strip barrel region the number of node features has been multiplied by three (the reconstructed Space Point with the two clusters coordinates). In order to have the same number of node features in all the detector, one give as input in the rest of the detector a three times concatenation of the space point reconstructed coordinates: $\left[r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}\right]$ (like three channels which give the same information). The advantage of this approach is that we keep a common learning which allow an efficient message passing between region, but on the other hand the MLP are not specialized by region.

In the rest of this proceeding we present results obtained with the homogeneous model approach. Study of the heteregenous model approach is still work in progress.

4 Experiments

The samples events used in this study was the same than in [4] and the event are represented as graphs using the same module map.

The training dataset used in the study presented here was composed of 1600 graphs. The node features are set as described in section 3. The preprocessed edges features are $\Delta r_{ij}^{reco}, \Delta\varphi_{ij}^{reco}, \Delta z_{ij}^{reco}, \Delta\eta_{ij}^{reco}, \varphi slope_{ij}^{reco}, r\varphi slope_{ij}^{reco}$. Compared to [4] φ_{slope} ($\Delta r_{ij}^{reco}/\Delta\varphi_{ij}^{reco}$) and $r\varphi_{slope}$ have been added as they are closely related to p_T (so informative for the curvature of the tracks). The model used is an homogeneous GNN model described at the begin of this section but with 3 layers per MLP and a non recurrent architecture for the Interaction Network layers (i.e. each Interaction Network layer have its own instance of $NodeUpdater$ and $EdgeUpdater$). BatchNormalization layers have been added after each linear layers in MLPs to improve the stability and speed of the training. The model forward pass is summarized in algorithm 1. The optimizer used was the same than in [4]. The training has last ~90k iterations.

The model is evaluated on a test set of 100 graphs. The results are shown in figures 6. For the same threshold (0.5) on edge score, a similar efficiency is observed with a slight improvement in the central region. A Significant improvement in purity is observed in the central region and more specifically in the the strip barrel region (from ~50% to ~80%).

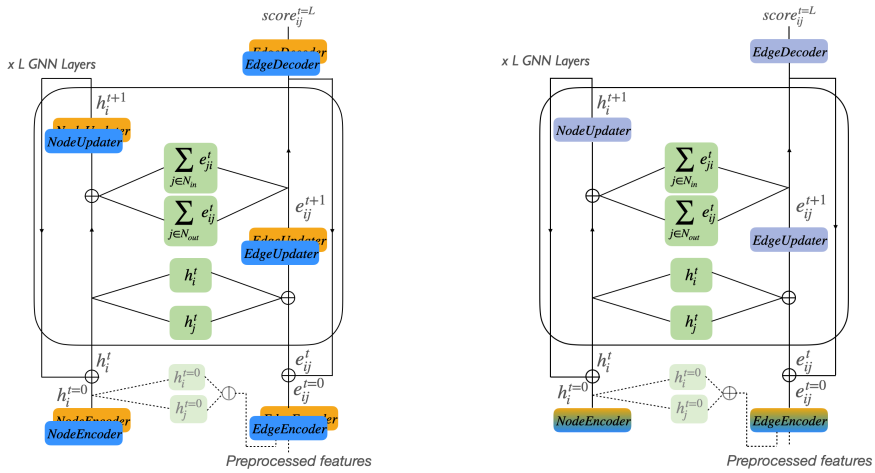
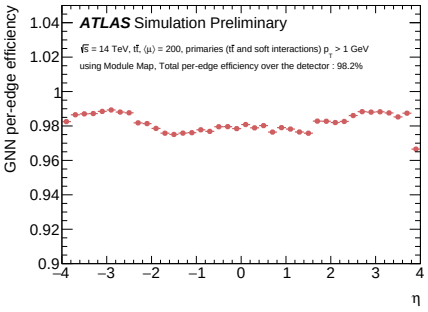


Figure 5: GNN designs handling heterogeneous data. (a) Heterogeneous GNN design with dedicated MLPs per region. (b) Homogeneous GNN design with common MLPs per region.

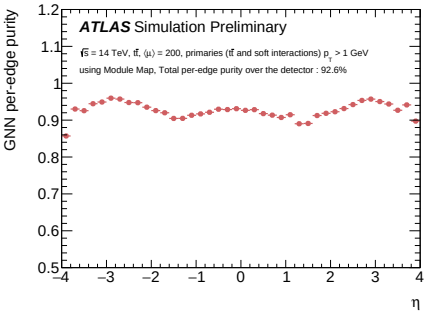
Algorithm 1 Forward pass of Homogeneous GNN handling heterogeneous data in ITk

Input : $G(V, E)$
$\forall i (v_i \in V)$:
if $region(v_i)$ == STRIP BARREL **then**
    $x_i \leftarrow \left[r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{c1}, \varphi_i^{c1}, z_i^{c1}, \eta_i^{c1}, r_i^{c2}, \varphi_i^{c2}, z_i^{c2}, \eta_i^{c2} \right]$
else
    $x_i \leftarrow \left[r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco}, r_i^{reco}, \varphi_i^{reco}, z_i^{reco}, \eta_i^{reco} \right]$
end if
$\forall i (v_i \in V), h_i^{t=0} \leftarrow NodeEncoder(x_i)$
$\forall i, j (e_{i,j} \in E), e_{ij}^{t=0} \leftarrow EdgeEncoder(\Delta r_{ij}^{reco}, \Delta \varphi_{ij}^{reco}, \Delta z_{ij}^{reco}, \Delta \eta_{ij}^{reco}, \varphi slope_{ij}^{reco}, r\varphi slope_{ij}^{reco})$
for $t \leftarrow 0$ to $L - 1$ **do**
    $\forall i, j (e_{i,j} \in E), e_{ij}^{t+1} \leftarrow \psi^t(e_{ij}^t | h_i^t | h_j^t)$
    $\forall i (v_i \in V), h_i^{t+1} \leftarrow \phi^t(h_i^t | \sum_{j \in N_{in}^i} e_{ji}^t | \sum_{j \in N_{out}^i} e_{ij}^t)$
end for
$\forall i, j (e_{i,j} \in E), score_{ij}^{t=L} \leftarrow EdgeDecoder(e_{ij}^{t=L})$

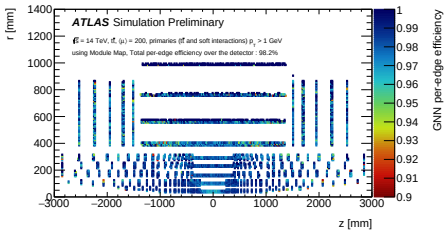
The track reconstruction is processed on the scored graphs of the test set after GNN inferences using a combination of a Connected Components algorithm with a loose requirement on the classification scores followed by a Walk-through algorithm. Figure 7 displays the new results and the ones of [4]. The track efficiency is similar in both studies. In [4], even with the poor GNN purity in the strip barrel region it was possible to get excellent track reconstruction performance *but* at the cost of the computation time of the Walk-through algorithm. In the present study, with high level of GNN edge-wise efficiency and edge-wise purity it is possible to get ~80% of perfect tracks and ~90% of standard matching tracks with a simple Connected Components algorithm. This point is very important in term of computation time as Connected Components algorithm can be easily accelerated on GPU (unlike the Walk-



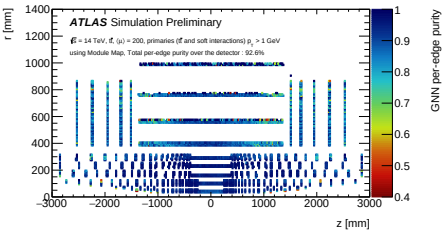
(a) GNN per-edge efficiency vs η



(b) GNN per-edge purity vs η



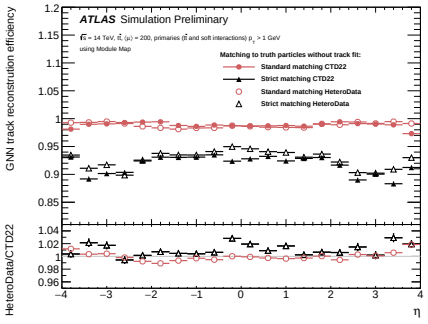
(c) GNN per-edge efficiency in (r, z) plane



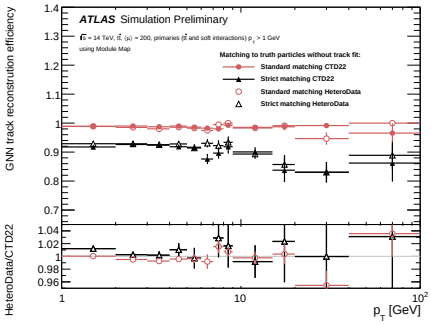
(d) GNN per-edge purity in (r, z) plane

Figure 6: GNN per-edge classification performance (efficiency and purity) along η and in the (z,r) plane as obtained in the present study. The (z, r) position of an edge is defined by the one of its source node.

through algorithm). The Walk-through algorithm can be applied only on a small subset of the graph to solve residual ambiguities and retrieve the same level of track reconstruction efficiency than [4].



(a) Track reconstruction efficiency vs η



(b) Track reconstruction efficiency vs p_T

Figure 7: Track reconstruction efficiency obtained in [4] and in the present study.

5 Conclusion

We present here novel GNN designs able to handle heteregeneous data in ITk. A complete study adding clusters data in the strip barrel region shows a significant improvement of the GNN performance compared to the first tracking GNN-based study presented by the ATLAS collaboration [4]. The new GNN model presented is a major upgrade to the GNN4ITk pipeline. The homogeneous performance and the high level of efficiency and purity allows an easier and faster track reconstruction at the final stage of the pipeline. This is an important step towards the deployment at production-level of a GNN-based track reconstruction algorithm for HL-LHC.

Acknowledgements

We thank our colleagues at the IN2P3 computing centre (CC-IN2P3) in Lyon (Villeurbanne) for the smooth operation of their GPU production platform, and for the successful deployment of a new experimental platform dedicated to machine learning developments that require large amounts of memory. Without these resources, the present studies would not have been possible.

References

[1] J. Shlomi, P. Battaglia, J.R. Vlimant, Machine Learning: Science and Technology **2**, 021001 (2020)

[2] G. DeZoort, P.W. Battaglia, C. Biscarat, J.R. Vlimant, Nature Rev. Phys. **5**, 281 (2023)

[3] CERN Courier Volume 61, Number 5, September/October 2021 (2021), https://cds.cern.ch/record/2782568

[4] S. Caillou, P. Calafiura, S.A. Farrell, X. Ju, D.T. Murnane, C. Rougier, J. Stark, A. Vallier (ATLAS Collaboration), Tech. rep., CERN, Geneva (2022), presented at Connecting the Dots Workshop (CTD 2022), http://cds.cern.ch/record/2815578, http://cds.cern.ch/record/2815578

[5] G. Apollinari et al., *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1*, CERN Yellow Reports: Monographs (CERN, Geneva, 2017), https://cds.cern.ch/record/2284929

[6] L. Evans and P. Bryant, JINST **3**, S08001 (2008)

[7] ATLAS Collaboration, JINST **3**, S08003 (2008)

[8] Tech. rep., CERN, Geneva (2017), https://cds.cern.ch/record/2285585

[9] GNN4ITk Team, *ACORN*, {https://github.com/GNN4ITkTeam/CommonFramework}

[10] Tech. rep., CERN, Geneva (2021), https://cds.cern.ch/record/2776651

[11] P.W. Battaglia et al. (2016), 1612.00222