

Reconstructing jets in the Phase-2 upgrade of the CMS Level-1 Trigger with a seeded cone algorithm

Sioni Summers^{1,*}, *Ioannis Bestintzanos*^{2,1}, and *Giovanni Petrucciani*¹
on behalf of the CMS Collaboration

¹CERN (Geneva, Switzerland)

²University of Ioannina (Ioannina, Greece)

Abstract. The Phase-2 Upgrade of the CMS Level-1 Trigger (L1T) will reconstruct particles using the Particle Flow algorithm, connecting information from the tracker, muon, and calorimeter detectors, and enabling fine-grained reconstruction of high level physics objects like jets. We have developed a jet reconstruction algorithm using a cone centred on an energetic seed from these Particle Flow candidates. The implementation is designed to find up to 16 jets in each Xilinx Ultrascale+ FPGA, with a latency of less than 1 μ s, and event throughput of 6.7 MHz to fit within the L1T system constraints. Pipelined processing enables reconstruction of jet collections with different cone sizes for little additional resource cost. The design of the algorithm also provides a platform for additional computation using the jet constituents, such as jet tagging using neural networks. We will describe the implementation, its jet reconstruction performance, computational metrics, and the developments towards jet tagging.

1 Introduction

The Phase-2 Upgrade of the CMS Level-1 Trigger (L1T) [1] will have access to new, highly-granular, data not available in the Phase-1 system. Tracks will be reconstructed for every collision event from modules in the silicon outer tracker for charged particles with transverse momentum (p_T) above 2 GeV in the L1T Track Finder [2]. The new endcap high-granularity calorimeter, a sampling calorimeter with high transverse and longitudinal granularity will send clusters to the L1T [3]. In addition, the barrel calorimeter will provide more granular data to the L1T than during Phase-1. The upgraded L1T has a latency restriction of 12.5 μ s to return its select/reject decision back to the detector for event readout. Of this total, approximately 5 μ s are allocated to the reconstruction of charged particle tracks in the outer tracker, calorimeter clusters, and muon tracks, and around 1 μ s to each of two ‘correlator’ layers that are described in Section 2. Jets will be reconstructed in the second of these correlator layers, implying a 1 μ s latency budget for the jet reconstruction. The baseline design comprises Xilinx Ultrascale+ FPGAs and high speed optical transceivers for data transmission between devices.

The correlator subsystem of the L1T will carry out Particle Flow (PF) [4] reconstruction and Pileup per Particle Identification (PUPPI) [5] pileup suppression. PF reconstructs particle candidates from the charged particle and muon tracks, and calorimeter clusters by linking compatible measurements. PUPPI suppresses the contribution from the 200 pileup

*e-mail: sioni@cern.ch

(PU) vertices planned for the High Luminosity LHC (HL-LHC). Charged PF particles are removed by PUPPI if they are not associated to the reconstructed Primary Vertex, and the p_T of neutral particles is weighted according to the local neighbourhood of other particles. The implementation and performance of PF and PUPPI for the L1T are described in [6–8].

Jets will be reconstructed from the PUPPI candidates in the L1T. In the CMS High Level Trigger (HLT) and offline reconstruction, jets are reconstructed with the anti-kt [9] algorithm. This algorithm is used as the benchmark of performance in Section 4, but due to the large number of iterations required, it is deemed unsuitable for implementation within the $1\mu\text{s}$ constraint. Cone algorithms for reconstructing jets have previously been used at CMS, but are disfavoured due to lack of colinear and infra-red safety [10]. However the computational simplicity makes them a good candidate for fast implementation in the online domain.

We implemented a simple jet finding algorithm using a cone centred on an energetic seed particle. Our implementation uses the full granularity of information available from the reconstructed particles, and fits within the L1T system constraints. The implementation provides a platform for development of further sophisticated jet processing, that will take the capabilities of the L1T beyond what is possible with the Phase-1 system. Here we describe the concurrent reconstruction of jets with both $R = 0.4$ and $R = 0.8$ radius parameters: ‘SC4’ and ‘SC8’ respectively. SC4 jets are used for studies with the baseline trigger selections, while the SC8 jets are under investigation for improving the performance of the L1T in particular for boosted topologies. This may have applications to extend the physics reach of the CMS Phase-2 L1T, for example in searches for low-mass resonances decaying to boosted jets, that have required some initial- or final-state radiation in order to pass a trigger selection in previous analyses [11, 12].

In addition, reconstructing jets from particles (as opposed to calorimeter tiles as in the Phase 1 system) enables further processing of the jets based on their constituents. This has already been used for the development of b-tagging in the L1T, using a Neural Network to classify jets originating from bottom quarks using the information of each constituent [13].

2 Algorithm

2.1 Deregonizer

Before finding jets, the data arriving into each Correlator Layer 2 FPGA are reorganised with the objective of simplifying the implementation of the jet algorithm itself. In the baseline design, Correlator Layer 1 comprises 36 FPGAs, divided into 6 event processors (through time multiplexing), each further divided into 6 regional processors in (η, ϕ) space. Each Layer 1 board transmits the PUPPI candidates over 3–6 links operating at 25 Gbs^{-1} to Layer 2, depending on the detector region. Within the ‘macro-region’ processed by one board, the PF and PUPPI candidates are reconstructed in smaller ‘PF-regions’ to simplify the implementation. The objects from each PF-region are transmitted from Correlator Layer 1 to Layer 2 across several clock cycles, and PF-regions are processed and transmitted sequentially. The total transmission period for the PUPPI objects equates to the time multiplexing subdivision of 6 LHC bunch crossings, or 150 ns, after which time the next event is transmitted.

At the receiving end in Correlator Layer 2, a ‘deregonizer’ is placed that receives and aggregates the PUPPI candidates from each Layer 1 macro-region, and outputs one ‘flattened’ list of particles once per event. The deregonizer removes null objects from the input packet and aligns all valid particles to one end of the list, with no gaps. The output of the deregonizer is a fixed-sized list, with the size chosen from balancing the number of L1T PUPPI candidates observed in Phase-2 simulations with the cost of processing those candidates in the jet finding.

Figure 1 show the event multiplicity of PUPPI candidates in different simulated processes, with the deregionizer output size of 128 particles, after which particles would be truncated.

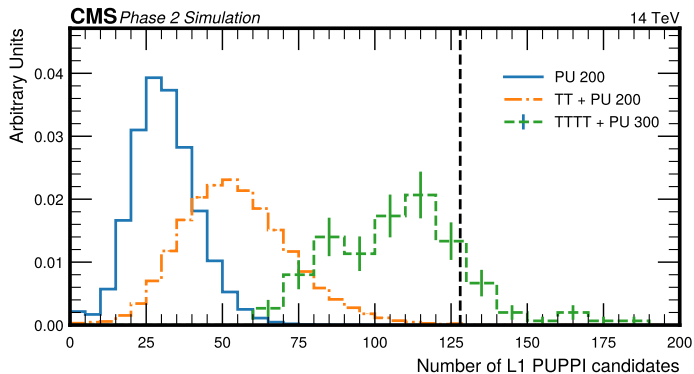


Figure 1. Distributions of multiplicity of L1 PUPPI particles in different simulated signal samples. The dashed line shows the deregionizer module limit, after which particles are truncated.

2.2 Jet finding

From the flattened list of particles, the following steps are repeated until either no particles remain in the list, or the maximum number of jets are found:

1. find the particle with the highest p_T : the jet seed
2. find particles within the jet radius, R , of the seed: the constituents
3. define the jet axis as the sum over the constituents' p_T , and the p_T -weighted average of η and ϕ
4. correct the jet p_T using a table of correction factors derived from matching reconstructed jets to simulated jets
5. remove the jet constituents from the list of particles

The algorithm is simple, considering an efficient, low latency implementation in FPGAs. Nonetheless, using the full granularity of information from constituent particles, reconstructed from measurement in multiple subsystems, represents a development beyond the Phase-1 capability and the Phase-1-inspired algorithm described in [1].

3 Implementation

As described, the simple jet algorithm was designed as such in order to reach a low latency and fit within the resource constraints of the Virtex Ultrascale+ FPGAs of the Phase 2 trigger system. Even so, the implementation makes use of several techniques in order to fit the performance requirements. As a design principle, we seek to maximise use of the two types of parallelism afforded by FPGAs: resource-level and pipeline-level. Resource-level parallelism enables the concurrent execution of separate tasks, while pipeline-level parallelism enables one task to process multiple data simultaneously.

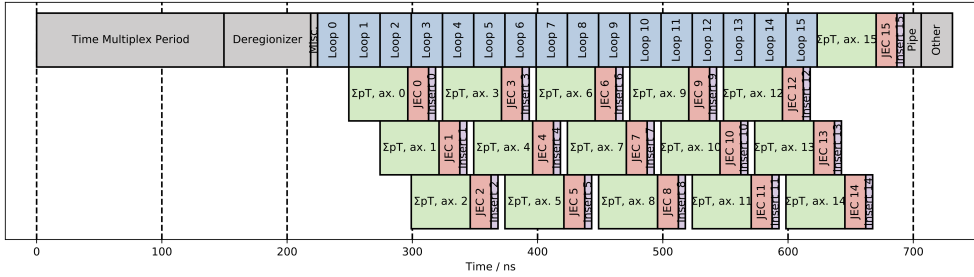


Figure 2. Processing timeline of one event, showing concurrent execution of tasks. The tasks contributing to the latency critical path are grouped along the top row.

We note that the algorithm described in 2.2 has a ‘loop-carried dependence’ [14]. That is, since the constituents of a jet are removed from the particle list before finding the next seed, the processing of iteration ‘ i ’ cannot commence until the completion of iteration ‘ $i - 1$ ’. Consequently, the total latency comprises a component proportional to the latency of the loop. In order to minimize the latency of one loop iteration, we identify the shortest path creating the loop-carried dependence. Referring to the algorithm steps in 2.2, steps 1, 2, and 5 comprise this path. As soon as the constituents of jet ‘ $i - 1$ ’ are found and removed from the list, the seed-finding of jet ‘ i ’ can commence simultaneously with the computation of the jet axis of jet ‘ $i - 1$ ’ (steps 3 and 4).

The processing steps are implemented as modules in the FPGA using Xilinx High Level Synthesis (HLS) technology. Each module is pipelined with an ‘initiation interval’ (II) of one clock cycle, such that they can accept new data every cycle. A control module orchestrates the injection of new event data, and the movement of data between the modules, including the iteration of the loop, implemented in VHDL. Jets are found in order of the p_T of the seed, which does not equate to ordering by p_T of the jet, although they are correlated. After the p_T of each jet is corrected, the jet is inserted into a module that orders them by p_T .

Figure 2 shows a timeline view of the module processing. Each iteration of a block is executed in the same FPGA logic module (e.g. ‘Loop 0’ and ‘Loop 1’ refer to the same module processing different data). The top row of the timeline depicts the critical path of the latency. From the start of the event at $t = 0$, the Time Multiplex Period of 150 ns and the Deregonizer latency are spent receiving and reorganising the particles from Correlator Layer 1. At 225 ns the jet finding iteration begins, with 16 iterations of finding a seed and the in-cone constituents. The latency of all but one of the iterations of jet axis computation and correction is hidden due to the overlap of processing with the jet finding loop. The total latency from the first input particle to the first output jet (FIFO) is 720 ns, with an additional 138 ns to transmit the output jets serially on one link.

Given the previously described loop carried dependence, and the II=1 implementation of the processing modules, only every 8th pipeline slot of the modules as shown in Figure 2 is utilised. This facilitates a concurrent processing of events in the same modules with pipeline parallelism. Since events exit the Deregonizer at intervals of 150 ns, and the latency of the 16 jet iterations – the duration that an event occupies one slot in the loop pipeline – is 400 ns, up to three events will be circulating at any one time. The control module that orchestrates the jet loop tracks the iteration of each event, sending the jets of each event into separate sorting modules. Without this mechanism, three parallel copies of the jet algorithm would be needed to keep up with the event throughput.

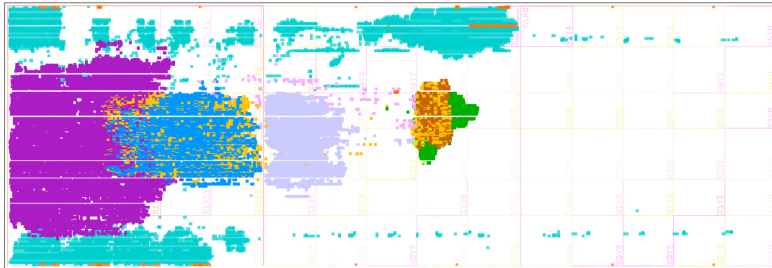


Figure 3. Floorplan of the Seeded Cone Jet algorithm in a VU9P FPGA.

With event concurrency, up to three of every eight pipeline slots are utilised. Since more than half of the possible pipeline slots remain unused, we are able to process each event twice without significant extra resource cost. The radius of the cone used in the constituent finding loop is configurable from the orchestration module, so each event can be processed with two different radius parameters. We therefore produce jets reconstructed with a radius of $R = 0.4$ and $R = 0.8$ for each event. As for event concurrency, the jet collections for each radius setting are sorted in separate modules.

Up to 12 reconstructed jets of each collection ($R = 0.4, 0.8$) are finally transmitted to the Global Trigger, which makes the final event selection decision by comparing all upstream computed high-level objects (jets, τ s, electrons, muons, photons, missing transverse energy, and more) against a ‘menu’ of conditions on which to trigger [15].

Figure 3 shows the floorplan of placement of jet finding modules in a VU9P FPGA on a Serenity ATCA card [16]. Inputs arrive on multi-gigabit transceivers in the leftmost Super Logic Region (SLR) tile. The placement of modules is loosely constrained to flow from left to right through the floorplan as shown. One SLR remains completely unutilised, leaving space for the development of more sophisticated processing.

The resource usage of the algorithm and its modules in the VU9P FPGA is shown in Table 1. The overall resource usage is very small, reaching at most 7.6% of an individual resource type, namely the DSPs. These are used primarily for multiplications in the computation of ΔR between the seed and each particle.

Module	LUT (%)	FF (%)	DSP (%)	BRAM (%)
Total, of which:	7.4	6.2	7.6	0.1
Deregonizer	3.5	2.3	0.0	0.0
Jet Finding	3.5	3.6	7.5	0.0
Jet Corrections	0.0	0.0	0.0	0.0
Hadronic Sums	0.3	0.1	0.1	0.0

Table 1. Resource usage of the jet finding algorithm as a percentage of the Xilinx VU9P FPGA, for Look Up Tables (LUT), Flip Flops (FF), Digital Signal Multipliers (DSP) and Block RAMs (BRAM).

4 Performance

The performance of the LIT seeded-cone jet reconstruction algorithm has been evaluated on simulated LHC events of different physics processes, propagated through the simulation of the response of the CMS detector. Where possible all LIT processing algorithms are emulated in C++ as part of the CMS experiment software framework CMSSW. These emulators are designed to match the behaviour of the FPGA implementations at a bit-exact level.

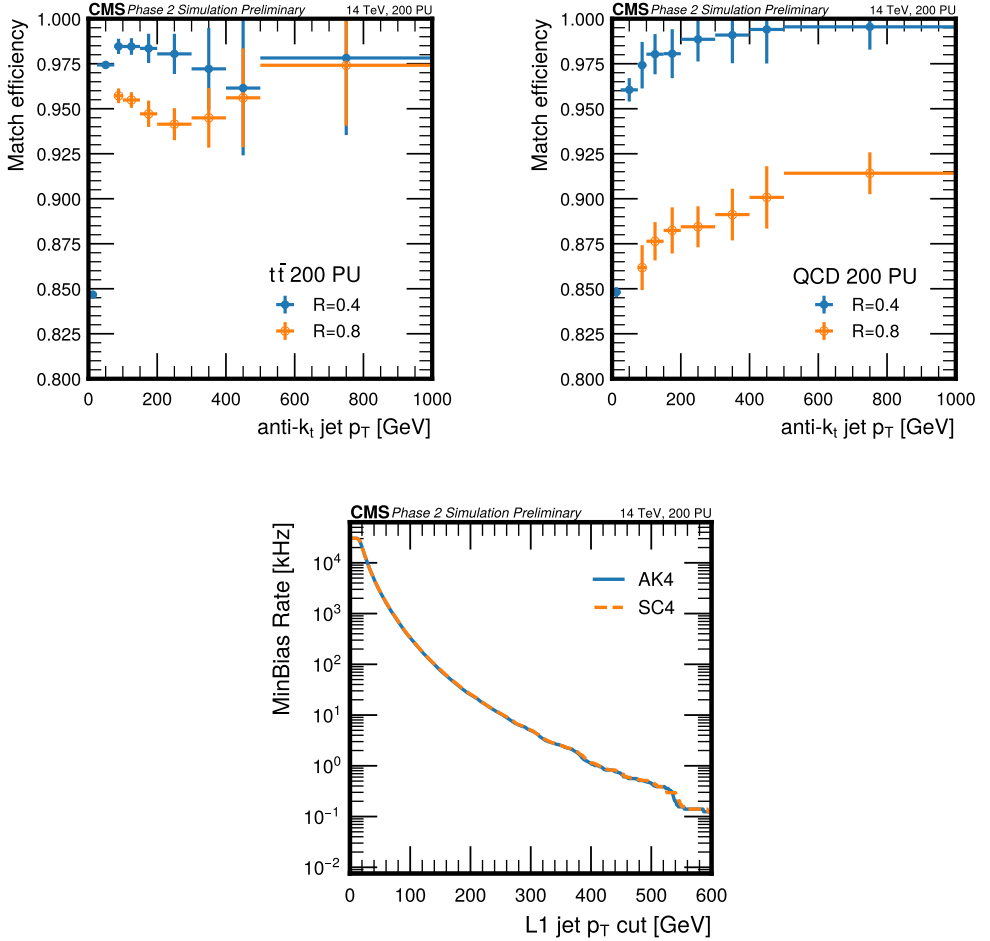


Figure 4. Top row: matching efficiency of seeded-cone jet algorithm with respect to anti- k_t jets of the same radius parameters, for both $R = 0.4$ and $R = 0.8$, in simulated events of $t\bar{t}$ (left) and QCD (right), both with 200 pileup. Bottom row: trigger rate of a selection on the highest p_T jet of each event, for seeded-cone (SC4) and anti- k_t (AK4) with radius parameter $R = 0.4$, in simulated ‘minimum bias’ (MinBias) events with no hard physics process, and 200 pileup.

From the emulated L1T PUPPI particles, jets are reconstructed using both the seeded-cone and anti- k_t jet algorithms. The seeded-cone implementation is fully emulated, while the anti- k_t algorithm is the offline implementation in the FastJet package [17]. The emulation matches the hardware implementation with bit-exact agreement on 99.9% of jets. Jet p_T corrections are derived from matching of SC4 L1T PUPPI jets to AK4 jets reconstructed from simulated particles in bins of η and p_T , using a sample of simulated $t\bar{t}$ events with 200 pileup. In each bin a correction factor is derived that is applied to the SC4 L1T PUPPI jets.

The matching efficiency of seeded-cone to anti- k_t jets is shown in Figure 4 (top row), with radius parameters of $R = 0.4$ and $R = 0.8$ used for both algorithms. Each anti- k_t jet is considered matched when a seeded-cone jet is found within $\Delta R \leq 0.2$ and $0.8 \leq p_{T,AK}/p_{T,SC} \leq 1.2$.

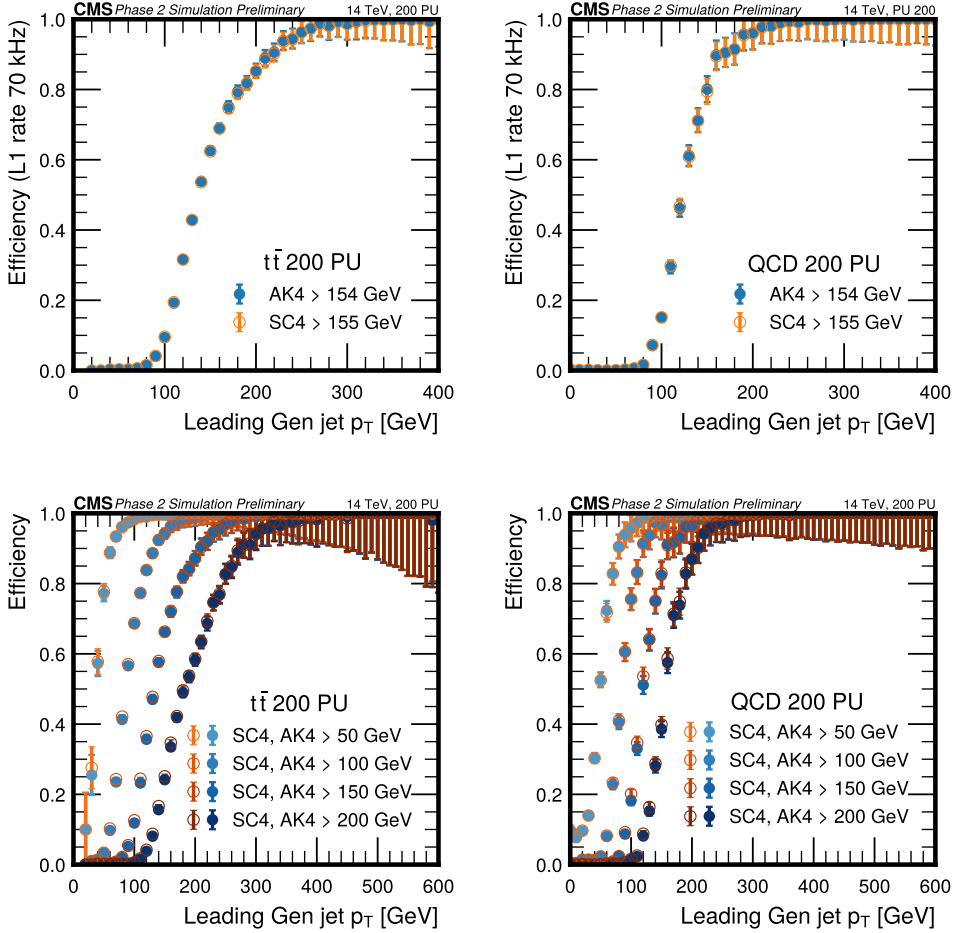


Figure 5. Top: turn-on curve at a fixed rate of 70 kHz, for seeded-cone (SC4) and anti-kt (AK4), with radius parameter $R = 0.4$, in simulated events of $t\bar{t}$ (left) and QCD (right), both with 200 pileup. Bottom: turn-on curves with different thresholds applied to the leading jet p_T , for SC4 and AK4 in simulated events of $t\bar{t}$ (left) and QCD (right), both with 200 pileup.

We observe excellent matching of SC4 to AK4 in events with simulated QCD jets and 200 pileup. The matching efficiency of SC4 to AK4 is very high in events of $t\bar{t}$ and 200 pileup, though somewhat worse than in the QCD jets. Matching of SC8 to AK8 is better in $t\bar{t}$ events than QCD. We understand these differences to be a consequence of the seeded-cone algorithm seeding on the single highest p_T constituent, and the presence or absence of boosted jets in the events.

Figure 4 (bottom row) shows the rate of a trigger selection on the p_T of the leading L1T jet of each event on ‘minimum bias’ data – events with only pileup vertices and no hard physics process which are representative of the majority of LHC collisions. The seeded-cone and anti-kt performance is very similar.

Figure 5 shows the efficiency turn-on curves for a leading jet trigger at a fixed rate of 70 kHz (top row), with the thresholds derived from Figure 4, and turn-on curves for different thresholds from 50 to 200 GeV (bottom row). The seeded cone performance is typically very close to the anti-kt performance in all cases. At every threshold studied the efficiency reaches a plateau at 100%. The steepness of turn-on varies between the different thresholds and simulated samples. At the higher p_T thresholds the turn-on in $t\bar{t}$ with 200 pileup is noticeably slower than at the same threshold in QCD with 200 pileup. The seemingly lost efficiency in $t\bar{t}$ is understood to arise from cases where the system is insufficiently boosted to produce a single narrow jet that is completely captured by the $R=0.4$ reconstruction. These events may still be triggered by selections on multiple jets, hadronic energy sums, or the large radius jets.

5 Conclusions

We have described the implementation of a seeded cone jet algorithm, a lightweight jet reconstruction designed for the tight latency constraint of the CMS Phase-2 Level-1 Trigger system. The implementation fits within one of the Virtex Ultrascale+ FPGAs of the system, with adequate resources still available for further jet processing such as jet-tagging.

The performance of jet triggers on the Seeded Cone jets matches closely to that reached with anti-kt jets reconstructed from the same L1 PUPPI particles, showing that the simple algorithm sufficiently utilises the information available from its inputs.

In addition to reconstructing jets for use in a baseline menu complementary to the CMS Phase-1 physics program, the seeded cone jet algorithm provides a platform to extend to a rich set of jet triggers. Tagging of b-jets using the seeded cone constituents has already been demonstrated in [13]. Here we described the extension to jets reconstructed with a radius $R = 0.8$, which we anticipate will be used to develop triggers, and possibly taggers, of boosted jet topologies.

References

- [1] CMS Collaboration, Tech. rep. (2020), <https://cds.cern.ch/record/2714892>
- [2] CMS Collaboration, Tech. rep. (2017), <https://cds.cern.ch/record/2272264>
- [3] CMS Collaboration, Tech. rep. (2017), <https://cds.cern.ch/record/2293646>
- [4] CMS Collaboration, JINST **12**, P10003 (2017)
- [5] D. Bertolini, P. Harris, M. Low, N. Tran, JHEP **2014**, 59 (2014)
- [6] B. Kreis, Tech. rep. (2018), 1808.02094
- [7] G. Petrucciani, EPJ Web Conf. **214**, 01019 (2019)
- [8] C. Herwig, JINST **18**, C01037 (2023)
- [9] M. Cacciari, G.P. Salam, G. Soyez, JHEP **2008**, 063 (2008)
- [10] R. Atkin, Journal of Physics: Conference Series **645**, 012008 (2015)
- [11] CMS Collaboration, Phys. Rev. D **99**, 012005 (2019)
- [12] CMS Collaboration, Phys. Rev. Lett. **123**, 231803 (2019)
- [13] A. Chambers, D. Rankin, *A neural network-based tagger for the identification of bottom quarks in the CMS Level-1 trigger*, in *APS April Meeting Abstracts (2022)*, Vol. 2022 of *APS Meeting Abstracts*, p. H09.007
- [14] J. Licht, M. Besta, S. Meierhans, T. Hoefer, IEEE TPDS **32**, 1014 (2021)
- [15] G. Bortolato, C. Deldicque, D. Gigi, B. Huber, E. Leutgeb, A. Lobanov, D. Rabaday, A. Racz, H. Sakulin, JINST **18**, C01034 (2023)
- [16] A.W. Rose, D. Parker, G. Iles, O. Sahin, P.A. Bausson, A. Tsirou, G. Fedi, P.G. Verdini, L. Ardilla, M. Balzer et al., PoS **TWEP2018**, 115 (2019)
- [17] M. Cacciari, G.P. Salam, G. Soyez, Eur. Phys. J. C **72**, 1896 (2012), 1111.6097