

# 1 Track-based Algorithms

Standalone track algorithms represent an important part of the Phase-II trigger menu, since they provide the ability to robustly use an independent sub-detector. The tracker is the most granular layer in the CMS L1 decision, so is key for delivering pile-up resilient measurements at L1 which include the primary vertex z-position, track-based jets, and track based missing energy. The harsh pile-up conditions of the Phase-2 LHC make it paramount to separate L1tracks originating from the primary collision from spurious track hits caused by the large occupancy in the tracking volume. Track purity requirements become a key component to achieve manageable data-recording rates while maintaining low trigger thresholds. This section outlines firmware implemented track based algorithms for reconstructing MET,  $H_T$  and clustering tracks into jets.

## 1.1 Track-based Missing Energy

The vector sum of the transverse momenta (MET) of all particles produced in the primary collision is a key input for BSM triggers at Level-1. For the track-based algorithm described in this section, one of the main challenges is excluding tracks from bad combinations that give high transverse momentum. Though these tracks are rare in pileup after requiring a tight window around the primary vertex z position, events containing these tracks would dominate an L1 MET trigger. The algorithm makes use of track purity requirements in addition to the  $\Delta z(PV_z, trk_z)$  requirement to reduce the number of events with poorly measured momentum balance.

The track purity selection is based mainly on the confines of the detector, the track reconstruction algorithm, and the available track fit quality parameters. The track  $p_T$  and  $\eta$  requirements shown in Table 1 are based on the minimal  $p_T$  of tracks that can be reconstructed reliably and the tracking acceptance. The minimal number of track layers is also based on the minimal requirement for the track fit. There is a further requirement that 4 of the track layers must also have stubs to remove any 4-stub tracks that were created with only hits in 3 layers. Requirements on the track fit quality measured as  $\chi_{n.dof}^2$  is set to a minimum of 50 to reduce tracks that are poorly reconstructed. To further reject high momentum tracks from poorly reconstructed track hit combinations, a new variable, bend consistency,  $\chi_{bend}^2$  is measured. This measure of the bend of the track is uncorrelated to the  $\chi_{n.dof}^2$  from the track fit.  $\chi_{bend}^2$  is calculated based on the horizontal distance between the two consecutive track hits in a track module, the stubs. The bend consistency quantifies how compatible the bend of the track hits measured in

the detector is compatible with the reconstructed track  $p_T$ . A bad combination of track hits tend to have a large value of bend consistency compared to well-reconstructed tracks. The cut on  $\chi_{bend}^2$  is optimized to lower the fraction of poorly reconstructed tracks that are included in the MET calculation as well as maintaining a reasonable L1 data-rate for a track-based MET trigger. [Here you would reference the fake-rate and the L1 Rate Reduction plot]

Track Variable	Cut
$N_{stubs}$ per track layer	$\geq 4$
$p_T$	2 GeV
$\chi_{n.dof}^2$	50
$\chi_{bend}^2$	1.75

Table 1: L1 track purity requirements for tracks input to both the MET computation and the L1 track-based jet clustering. Cuts are optimized to preserve a sharp-on turn on in track-based MET,  $H_T$  and  $H_T^{miss}$  rejecting the bulk of "fake" tracks from bad combination of track hits.

The L1 vertex described in the TSA firmware algorithm in Section ?? is input to the L1 track-based MET algorithm. A window of  $\Delta z(PV_z, trk_z)$  is required for pseudo-rapidity regions within the tracking acceptance. Table 2 shows the range of values for the allowed range of track-z around the measured primary vertex z. Central tracks allow for a tight window of 0.4cm while more forward tracks need a wider z-window ( as large as 2.2cm) to include tracks within  $3\sigma$  of the gaussian track z-resolution in the forward regions.

$\eta$ range	$ \Delta z(PV_z, trk_z) $ [cm]
$0 \leq \eta < 0.7$	0.4
$0.7 \leq \eta < 1.0$	0.6
$1.0 \leq \eta < 1.2$	0.76
$1.2 \leq \eta < 1.6$	1.0
$1.6 \leq \eta < 2.0$	1.7
$2.0 \leq \eta < 2.4$	2.2

Table 2: Minimum  $\Delta z$  requirements between the primary vertex and track  $z_0$  for each pseudorapidity region for to select tracks for the MET algorithm.

Including these criteria for the optimized track MET measurement reduces the threshold at at data-taking rate of 35kHz from 235GeV to 60GeV. This threshold corresponds close to the simulated true track MET performance. The threshold of 60 GeV gives 95% efficiency for the trigger turn at 300GeV in generator level MET.

## 1.2 Jet Algorithm with Tracks

A rapid firmware implemented track clustering algorithm allows to combine an integrated collection of tracks and output a collection of track-based jets. The track-based jets can then be used for global trigger quantities like  $H_T$  and  $H_T^{miss}$ . As described in Section 1.1, the same set of track purity requirements are applied to the input selection of tracks to keep the L1 trigger objects resilient to pileup. In this section, we describe the simple track clustering algorithm that is implemented in firmware. Though the algorithm does not make use of an input L1 primary vertex, it can be used as input to the algorithm to make the algorithm more robust against contaminating L1 tracks from pileup.

The clustering of tracks in the  $\eta$ - $\phi$  plane is done using a nearest-neighbor approach in two 1-dimensional steps. The first phase of the algorithm parses tracks into 27  $\phi$  regions and 24  $\eta$  regions, which divides the region of tracking acceptance into cells of that are  $\approx 0.2 \times 0.2$ . The  $trk_{z0}$  is also used to assign the track to a z-cell along the beam spot. The clustering in  $\eta$ - $\phi$  is done in z-regions that span the beamspot. In between every two z-cells is a third cell to cover the overlapping region and eliminate edge cases where the primary vertex is close to the edge of a given bin. This gives a maximal jet size that is approximately a 3x3 square of cells with a half-side equal to  $\Delta R = 0.3$ . The width of the z-cell used for this study is 1cm.

Only tracks passing the purity requirements in Table 1 are clustered. In the first layer of track clustering in  $\phi$ , the sum of track  $p_T$ ,  $\sum p_T^{trk}$ , in each cell is compared to its two neighbors in  $\phi$  and the  $p_T$  is summed into the local maximum cell. The result of this step is to create a list of 1-dimensional track  $\sum p_T^{trk}$  clusters based on the local maxima in the  $\phi$  dimension, so that the minimum distance between any two  $\phi$ -clusters is one cell. The next step is to take the  $\phi$ -clusters and check if cells neighbor each other in  $\eta$ , and if they do they are merged into the  $\phi$ -cluster with larger  $\sum p_T^{trk}$ . The track-based jets are then the list of cells from the local maximum in the  $\eta$ - $\phi$  plane for a given z-cell. Jets from the primary interaction are found by summing the track-based jet  $p_T$ ,  $H_T^{trk}$ , in each z-cell and taking the jets from the z-bin with the largest  $H_T^{trk}$ . In addition, to the  $\sum p_T^{trk}$ , the track multiplicity is also summed in the same procedure to provide an additional handle for the output collection of track-based jets. This approach gives a list of track-based jet positions in  $\eta, \phi, z_0, p_T$ , and  $N_{tracks}$ .

Figure 1 shows the reconstruction efficiency for this firmware implementable track clustering compared to a full anti- $k_T$  clustering with FAST-JET. The same set of input L1 reconstructed tracks is used that pass the track purity requirements. The efficiencies are similar across the range of

tracking acceptance and the generator level  $p_T$  range, so the approach of 1D nearest neighbor clustering in two steps gives similar efficiency to full 2D jet reconstruction in an algorithm similar to offline jet reconstruction.

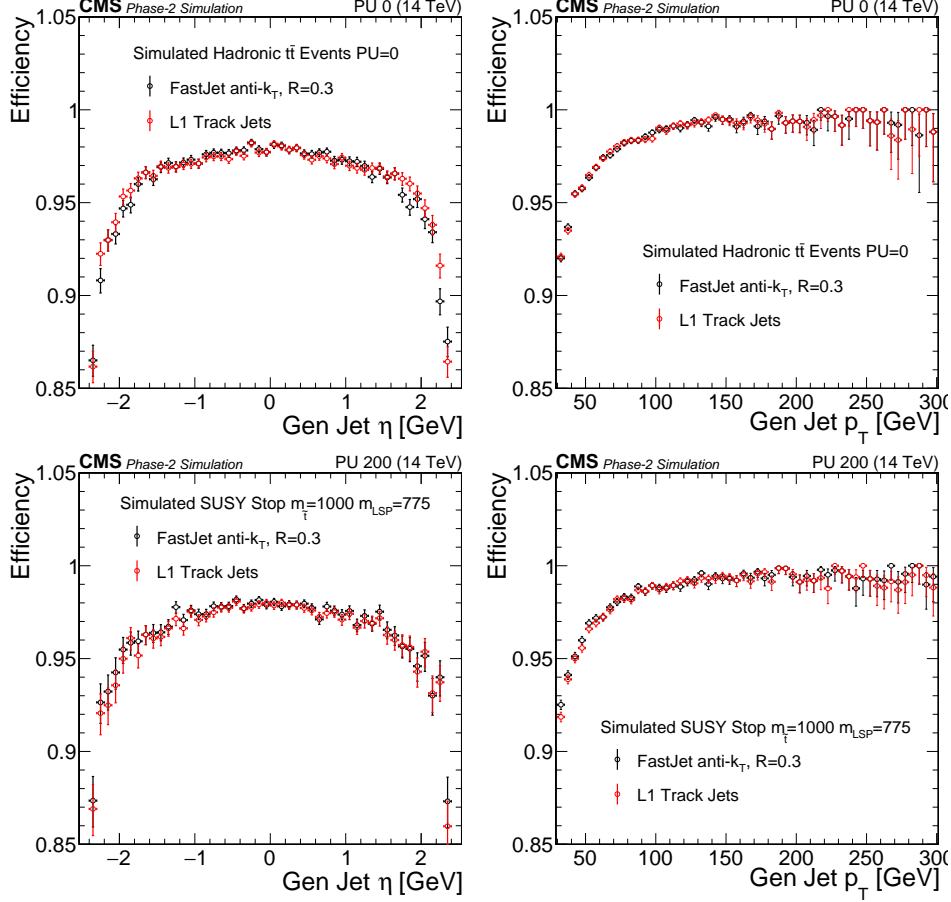


Figure 1: Comparison of the L1 track jet clustering and anti- $k_T$  clustering using FASTJET with  $R = 0.3$  with the same input collection of L1 reconstructed tracks. (top) The efficiency in fully hadronic  $t\bar{t}$  with no pileup events as a function of generator  $\eta$ (left) and  $p_T$  (right). The efficiency is computed by matching the generator-level jet to the reconstructed within  $\Delta R < 0.4$ .